ORTHOGONAL COLLOCATION ON FINITE ELEMENTS USING QUADRATIC AND CUBIC B-SPLINES

UNIVERSITY OF
KWAZULU-NATAL

Rasheed Ayodeji Adetona

October 2024

ORTHOGONAL COLLOCATION ON FINITE ELEMENTS USING QUADRATIC AND CUBIC B-SPLINES

by

Rasheed Ayodeji Adetona

Submitted in fulfilment of the

academic requirements for the degree of

Doctor of Philosophy (PhD)

in the

School of

Mathematics, Statistics and Computer Science

University of KwaZulu-Natal

October 2024

As the candidate's supervisors, we have approved this thesis for submission.

Signed: Name: Prof. N. Parumasur Date: 18^{th} October, 2024

Signed: Name: Prof. P. Singh Date: 18^{th} October, 2024

Published Articles

1. Adetona, R. A., Parumasur, N., & Singh, P. (2024). Solution of the Space Fractional Diffusion Equation Using Quadratic B-Splines and Collocation on Finite Elements. Contemporary Mathematics, 1232-1256.

2. Adetona, R. A., Parumasur, N., & Singh, P. (2024). Solution of the Schrödinger equation using quadratic B-Spline collocation on non-uniform grids. Partial Differential Equations in Applied Mathematics, 9, 100621.

3. Parumasur, N., Adetona, R. A., & Singh, P. (2023). Efficient solution of Burgers', modified Burgers' and KdV–Burgers' equations using B-spline approximation functions. Mathematics, 11(8), 1847.

Preface

The work reported in this thesis was carried out in the School of Mathematical Sciences, University of KwaZulu-Natal, Durban, from February 2022 to January 2024, under the supervision of Prof N. Parumasur and co-supervised by Prof P. Singh. I confirm that the work submitted in this study is my original work and has not otherwise been submitted in any form for any previous degree or professional qualifcation to any other tertiary institution. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

Signed: 

Rasheed Ayodeji Adetona

FACULTY OF SCIENCE AND AGRICULTURE

DECLARATION 1 - PLAGIARISM

I, Rasheed Ayodeji Adetona, declare that

1. The research/report presented in this thesis is my own original work, except where due reference is made.

2. This thesis has not previously been submitted to any institution (academic or otherwise) for any degree.

3. In this thesis I have not copied in part or whole or otherwise plagiarised the work of other students and/or persons in a form a data, pictures, graphs or other information, unless specifically acknowledged as being sourced from the said persons.

4. I have acknowledged all material and sources used in it's preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. Where other written sources have been quoted, then:

   (a) Their words have been re-written and have referenced the general information attributed to them.

   (b) Their writing have been placed in italics within quotation marks and referenced, where their exact words have been used.

5. This thesis has no text, graphics or tables copied and pasted from the Internet,unless specifically acknowledged, and the source being detailed in the thesis and in the Bibliography.

Signed: 

Rasheed Ayodeji Adetona

Dedication

This thesis is dedicated to Allah, the most gracious and the most merciful.

Acknowledgements

In the name of Allah, the most gracious and the most merciful. Alihamdu-li-Lahi for the successful completion of my Doctor of Philosophy (Ph.D) in Mathematics in University of KwaZulu-Natal, South Africa.

My sincere gratitude goes to my supervisors Prof. N. Parumasur and Prof. P. Singh for their support, cooperation, understanding and guidance throughout the programme. I am very grateful to the members of staff of the department of mathematics, faculty of science and Obafemi Awolowo university (OAU), Ile-Ife, Nigeria, for the opportunity given to me to pursue my doctoral degree in university of KwaZulu-Natal. My gratitude also go to TETFUND for their support throughout the completion of my doctoral degree programme.

This piece of work would be incomplete if I fail to acknowledge the support of Prof. B.A. Taleat and Dr. A. T. Raji for their contributions to the success of this programme.

I also express my sincere appreciation to my parents, wife, children, brother, sisters and friends who contribute to my success in one-way or the other. May Allah continue to shower his blessings upon you all.

Abstract

We discuss the application of orthogonal collocation on finite elements (OCFE) method using quadratic and cubic B-spline basis functions to ordinary, partial and fractional differential equations. Collocation is performed at Gaussian points to obtain optimal solutions, hence the name orthogonal collocation. The OCFE method based on quadratic spline is used to solve Burgers' equation, modified Burgers' equation and the nonlinear Schrödinger equation in non-uniform subintervals with different cases of soliton solutions. It is also extended to obtain numerical solutions of fractional differential equations taking the fractional diffusion equation as a case study.

In addition, we apply orthogonal collocation based on modified quadratic B-spline functions to solve time-dependent and time independent two-dimensional partial differential equations. The numerical results are in good agreement with previous ones in the literature.

Furthermore, KdV-Burgers' equation that does not have an exact solution and the one with exact solution, Burgers' equation, KdV equation and fractional differential equations are considered as test cases for the OCFE method using cubic B-spline basis functions. The results compare favourably with exact solutions and existing results in the literature.

Generally, the numerical schemes are proven to be convergent, unconditionally stable and utilizes minimal memory storage due to the sparse matrix systems associated with B-spline basis functions.

Table of Contents

List of Tables

Table **Page**

Table **Page**

Table **Page**

List of Figures

Figure **Page**

Figure **Page**

Figure **Page**

Figure **Page**

Figure **Page**

CHAPTER ONE

INTRODUCTION

In this thesis we use quadratic and cubic B-splines as trial functions for solving ordinary and partial differential equations using the method of orthogonal collocation on finite elements (OCFE) [17, 63]. Essentially, the OCFE method combines the features of the classical finite element and orthogonal collocation methods. It was developed in the chemical engineering community to solve problems having rapidly changing solution behaviours [17, 26, 72]. It is a robust numerical method which inherits the flexibilty of dynamic grid choices and the superior stability and convergence features of the orthogonal collocation method. We elaborate on further details of the method below. We also demonstrate the capability of our method to solve Burgers' equation, modified Burgers' equation, Schrödinger equation, KdV-Burger's equation and KdV equation. In addition we extend our method to solving two dimensional partial differential equations and fractional order partial differential equations.

1.1 Method of weighted residual

The method of weighted residuals (MWR) forms the basis for collocation methods. They are used to approximate solutions of differential equations by assuming the solution is a linear combination of basis functions and force the residual to be equal to zero. The method is described as follows:

Given a second order differential equation

$$\frac{d^2u}{dx^2} = f(x, u, u'(x)), \quad a < x < b,\ u(a) = u_a \text{ and } u(b) = u_b. \tag{1.1}$$

Suppose we approximate the exact solution by $u(x) = \sum_{i=0}^{n} \alpha_i \sigma_i(x)$, where $\sigma_i(x)$ are trial functions. Then the residual is written as

$$r(x, \alpha) = \sum_{i=0}^{n} \alpha_i \sigma_i''(x) - f\left(x, \sum_{i=0}^{n} \alpha_i \sigma_i(x), \sum_{i=0}^{n} \alpha_i \sigma_i'(x)\right), \tag{1.2}$$

and at the boundaries,

$$u_a = \sum_{i=0}^{n} \alpha_i \sigma_i(a), \tag{1.3}$$

$$u_b = \sum_{i=0}^{n} \alpha_i \sigma_i(b). \tag{1.4}$$

In the method of weighted residuals we set

$$\int_a^b w_i(x) r(x, \alpha)\, dx = 0, \quad i = 1, 2, \ldots, n. \tag{1.5}$$

where $w_i(x)$ are some chosen set of independent functions called the test functions. Other choices for $w_i(x)$ lead to methods, such as, Galerkin method and method of least squares. In the collocation method, $w_i(x) = \delta(x - x_i)$, where $\delta(x)$ is the Dirac delta function.

1.2 Orthogonal Collocation on finite elements (OCFE)

In the method of orthogonal collocation the solution of a differential equation is approximated by a linear combination of known functions called the trial functions [26]. It is called orthogonal because roots of orthogonal polynomials are chosen as the collocation points. If a continuous (trial) function is used to approximate the solution throughout a given domain then the approximation is said to be global.

However, if the domain is split into subintervals and continuous piecewise functions are used as basis functions then the resulting approximation is local and referred to as a finite element approximation. The method of OCFE combines the orthogonal collocation and finite element approach. It is useful for solving problems having rapid changes in their solution profiles.

1.3 Literature review

Collocation belongs to the class of numerical methods for solving ordinary differential equations called method of weighted residuals. It interpolates the residual to zero. It can also be used to solve partial differential equations, integral equations, integro-differential equations and fractional differential equations. This method eliminates some challenges associated with other numerical methods. It does not involve integrations and the resulting matrix has a small band width [4]. Other advantages of collocation method include flexibility, strong stability properties, high order of convergence and ability to recover missing values when step size is changed [16].

The works of the authors of [17, 20,23,27,35,41] are some of the research that laid the foundation for recent advances on orthogonal collocation method on finite elements. A research work on solution of boundary value problems using orthogonal collocation was presented in [72]. They chose appropriate weight functions and zeros of Chebyshev polynomials as collocation points since it minimizes the magnitude of the residual. This approach led to a more intuitive, finite difference like method and facilitated automation of the method for general applications. It was noted that the orthogonal collocation method is a discrete analog of the Galerkin's

method. In [46] a procedure for discretization matrices for first and second derivatives, roots of orthogonal polynomials and weight functions for orthogonal collocation was examined. The author in [26] applied orthogonal collocation to solve models of chemical reactor for heat and mass transfer in an SO_2 (sulphur (iv) oxide) oxidation reactor, countercurrent heat transfer in ammonia reactor and a transient model on reduction of nitric oxide in automobile exhaust. The method compared favourably with finite difference and required less time for computation. It is worthy to note that high rate of convergence exhibited by OCFE is due to continuous representation of the approximate solution in the domain of interest. A comparison of spectral and pseudospectral methods for some simple model problems was made in [51] and it was found that both methods gave similar errors.

The authors in [6] were the first to use the word differential quadrature since quadrature is a discrete evaluation of an integral. They proposed that the method could be used to solve differential and integral equations. The method was described and demonstrated for some problems in [7] using Legendre polynomials as test functions. In [10], a review of the differential quadrature method was carried out. It was applied to solve heat and diffusion problem, integro-differential equation and free vibrating cantilever beam problem. The author in [49] used polynomial differential quadrature to solve Burgers' type nonlinear differential equations. It was concluded in [75] that the differential quadrature and pseudo-spectral methods are just other names for the collocation method. The main drawbacks of these methods are the use of dense matrices in their computations.

The flexibility of collocation method is not only it's independence on integration

but also the choice of basis functions and collocation points. Hermite polynomials of various degrees have been used as trial functions for collocation method. A cubic Hermite collocation method was used to solve nonlinear Burgers' equation in [28]. In [53], a collocation method for solving system of fractional differential equations with variable coefficients using Hermite polynomials as trial functions was presented. In [73] collocation method on Hermite polynomials was used to obtain solutions of pantograph equations with variable coefficients. A septic Hermite collocation method for Kuramoto-Sivashinsky equation was studied in [39].

An orthogonal collocation method based on global basis function was used in [13] to solve optimization problems. In [69] a model of a tubular isothermal catalytic-wall reactor was solved by orthogonal collocation method. A Gauss' pseudospectral method was used to solve a continuous-time optimal control problem in [9]. The accuracy of orthogonal collocation method was improved in [19] by using piecewise basis functions instead of the global ones. The authors of [29] also used a modified piecewise polynomial as basis function for Galerkin method to obtain the numerical solution of boundary value problems. An extensive discussion on the properties, error estimates and application of orthogonal collocation using piecewise B-splines to ordinary differential equations were done in [20]. [22] extended the work in [20] to PDEs. These methods are referred to as orthogonal spline collocation (OSC) method. The main drawback of OSC methods are that they are limited to the class of C^1 approximations.

OCFE method is equivalent to OSC although originally using Lagrange basis functions and piecewise Hermite basis functions.. A study of orthogonal collocation on finite elements using quintic Hermite polynomials was carried out in [63] to solve

partial differential equations. Numerical solution of time dependent singularly perturbed problems were also obtained by [5] using quintic Hermite collocation method.

Furthermore differential quadrature and splines are discussed in [8]. The author defined spline interpolating functions as piecewise interpolating polynomial functions that satisfy some continuity properties at the interpolating points. Recently a review on developments on collocation methods for solving differential equations was made in [64]. Quartic B-spline collocation method was also employed in [60] for solving the Burgers' equation. Quintic B-spline collocation method was used to solve Kuramoto-Sivashinski equation by [47]. A modified cubic B-spline collocation method that gave a diagonally dominant system was examined and applied to Burgers' equation without using linearization in a paper by [48]. The authors in [34] obtained the solution of singularly perturbed boundary value problems with cubic B-spline collocation method. The method was found to be second order accurate.

Moreover, a decatic spline collocation was used to solve Burgers' equation in [32]. Trigonometric [3,21,62,74] and exponential [30,55,77] spline collocation methods have also been considered for solving differential equations in literature. A compact spline collocation using quadratic basis functions for the Helmholtz equation was studied by [25]. Numerical solution of the fractional Black-Schole's equation using a compact quadratic spline collocation method was examined in [70]. The Dirichlet biharmonic problem was solved by [12] with quadratic spline collocation method. The application of quadratic spline methods to shallow water equations in spherical coordinate was discussed in [42]. Quadratic non-polynomial spline was used to

solve a system of second order boundary value problems in [65]. It was noted in [37] that cubic spline is less accurate than quadratic splines in some cases as demonstrated with examples in their work. Based on the above, we develop the quadratic and cubic OCFE methods. The merits of each method is included within each chapter. In addition a list of proposed works are recommended for future research.

1.4 Organization of the thesis

This thesis is organized as follows: In chapter 2 we derive the method of orthogonal collocation on finite elements using quadratic B-spline functions. We apply the method to ordinary differential equations and discussed it's convergence. Chapter 3 is dedicated to the application of OCFE based on quadratic splines to partial differential equations. We consider Burgers' and Modified Burgers' equations as examples. Chapter 4 deals with the solution of partial differential equations on non-uniform intervals with emphasis on the Schrödinger equation. In chapter 5, we present the application of OCFE using quadratic splines to solve space fractional differential equations with some illustrations. In Chapter 6, we applied quadratic B-spline OCFE to solve two-dimensional partial differential equations. Chapter 7 deals with OCFE using cubic B-splines to solve the KdV-Burgers', Burgers' and KdV and equations. We also use the method to approximate the numerical solutions of time and space fractional partial differential equations of orders $0 < \alpha < 1$ and $1 < \alpha < 2$, then Chapter 8 concludes our work with proposed future research.

CHAPTER TWO

QUADRATIC B-SPLINE ORTHOGONAL COLLOCATION ON FINITE ELEMENTS

In this chapter we discuss the derivation of quadratic B-spline basis functions for orthogonal collocation on finite elements, it's error estimates, convergence and it's application to ordinary differential equations. We consider cases of linear and nonlinear ordinary differential equations. Our results are are consistent with known results in the literature.

2.1 Quadratic B-Spline Basis

Consider a non-decreasing sequence of knots,

$$z_1 \leq z_2 \leq \cdots \leq z_k \leq z_{k+1} \cdots \leq z_{N+1}.$$

Each B-spline curve has the form $p(z) = \sum_{i=1}^{N+1} \beta_i B_{i,k}(z)$ where β_i are constants and $B_{i,k}(z)$ are normalized basis functions. The order of the basis function is k, and the degree of the polynomial is $k-1$. Some properties of B-spline curves are listed in [14] as follows:

1. $p(z)$ is a polynomial of degree $k-1$ on $z_i \leq z < z_{i+1}$.

2. $p(z) \in C^{k-2}[z_1, z_{N+1}]$.

3. $\sum_{i=1}^{k} B_{i,k}(z) = 1$.

4. Each $B_{i,k}(z) > 0$ on $[z_i, z_{i+k}]$.

5. Each basis function has one maximum value, except in the case of $k = 1$.

In order to calculate the basis functions, we need the knot vectors, which are usually written as $[z_i, z_{i+1}, \cdots, z_{i+p}]$ where $p + 1$ denotes the number of knots. There are three types of knot vectors: (1) uniform knots, which are evenly spaced; (2) non-uniform knots, which are irregularly spaced; and (3) open uniform knots. For the latter case, the multiplicities of the knots at the ends are equal to the order of the basis. We shall consider only the third type and only two distinct knots. Once the knots have been chosen, the basis is calculated using the Cox–de Boor recursion formula [20],

$$B_{i,1}(z) = \begin{cases} 1, & [z_i, z_{i+1}), \\ 0, & \text{otherwise}, \end{cases} \tag{2.1}$$

$$B_{i,k}(z) = \frac{z - z_i}{z_{i+k-1} - z_i}\, B_{i,k-1}(z) - \frac{z - z_{i+k}}{z_{i+k} - z_{i+1}}\, B_{i+1,k-1}(z). \tag{2.2}$$

Here, for the recursion to work, we adopt the notation $0/0 = 0$.

In this case, $k = 3$, and we use the knots $[z_1, z_2, z_3, z_4, z_5, z_6] = [0, 0, 0, 1, 1, 1]$. The multiplicity at the end points 0 and 1 is three, and there are two distinct knots. From the definition of $B_{i,1}(z)$, it follows that $B_{1,1}(z) = B_{2,1}(z) = B_{4,1}(z) = 0$ and $B_{3,1}(z) = 1$ in $[0, 1]$. One can also confirm from the recurrence relation that $B_{1,2}(z) = B_{4,2}(z) = 0, B_{2,2}(z) = 1 - z$ and $B_{3,2}(z) = z$. Hence, the recurrence relation gives

$$B_{1,3}(z) = (1 - z)^2,\ B_{2,3} = 2z(1 - z),\ B_{3,3}(z) = z^2.$$

From the recursion Formula (2.2), $B_{i,k}(z)$ can be represented as entries of the

matrix given by

$$\begin{pmatrix} 0 & 0 & (1-z)^2 \\ 0 & 1-z & 2z(1-z) \\ 1 & z & z^2 \\ 0 & 0 & 0 \end{pmatrix}. \tag{2.3}$$

We observe that if we expand $[z+(1-z)]^{k-1} = \sum_{i=0}^{k-1} \binom{k-1}{i} z^i (1-z)^{k-i-1}$ using the binomial theorem, then we can recover the quadratic basis $B_{i,k}$, $i = 1, 2, 3$, where k is the order of the spline. Therefore.

$$B_{i+1,k}(z) = \binom{k-1}{i} z^i (1-z)^{k-i-1}, \quad i = 0, 1, \ldots k-1. \tag{2.4}$$

Hence

$$B_1(z) = (1-z)^2, \; B_2(z) = 2z(1-z), \; B_3(z) = z^2, \tag{2.5}$$

where we have dropped the second subscript. The B-spline basis functions on $[0, 1]$ are graphically shown in Figure 2.1.

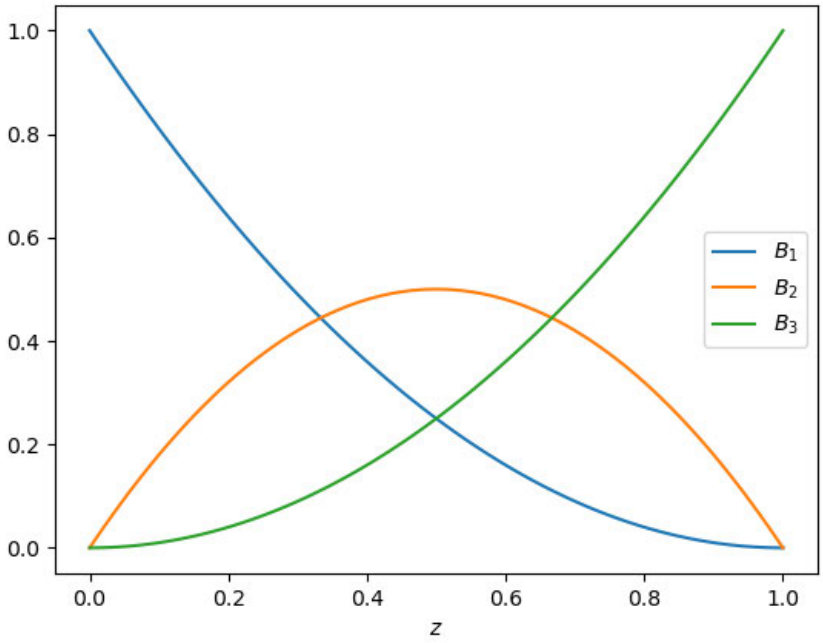

Figure 2.1: Graph of quadratic B-spline basis functions on $[0, 1]$.

Figures 2.2 and 2.3 show the continuities of the quadratic B-spline basis functions and their first derivatives across the boundaries x_{i+1} to x_{i+2}.

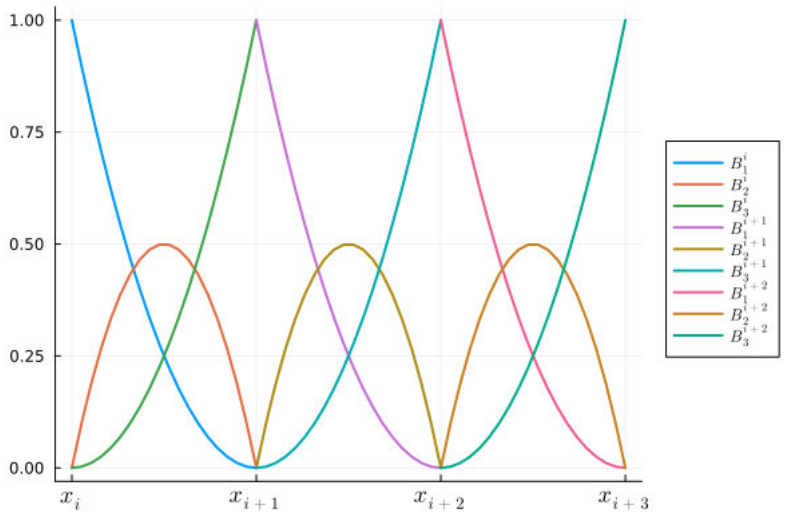

Figure 2.2: Continuity at the boundaries.

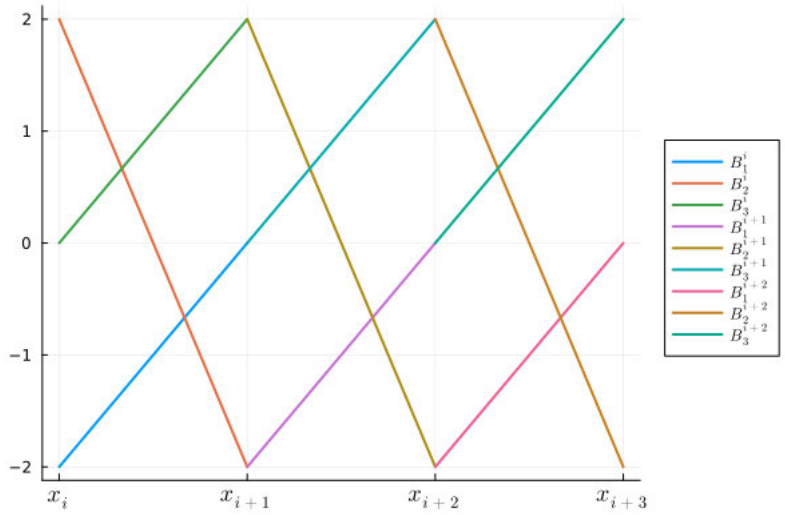

Figure 2.3: First derivative continuity at the boundaries.

2.2 Quadratic B-spline OCFE for second order ODE

For ease of explanation of the quadratic B-spline OCFE method, we solve a second-order ODE given by

$$a_2(x)y''(x) + a_1(x)y'(x) + a_0(x)y(x) = f(x), \quad x \in (a, b), \tag{2.6}$$

with the boundary conditions

$$y(a) = \theta, \quad y(b) = \Theta, \tag{2.7}$$

where $a_2(x)$, $a_1(x)$, $a_0(x)$ are coefficients and $f(x)$ is the source term.

We partition the interval $[a,\ b]$ into N elements with nodes given by $x_i = a + (i - 1)h$, $i = 1, 2, \ldots, N+1$, where $h = \frac{b-a}{N}$ is the uniform step size. The transformation $z = \frac{x-x_i}{x_{i+1}-x_i} = \frac{x-x_i}{h}$ maps the i_{th} interval $[x_i\ x_{i+1}]$ to $[0,\ 1]$. The collocation solution in interval i and interval $(i + 1)$ is respectively assumed to be

$$Y^i(x) = \sum_{k=1}^{3} b_k^i B_k(x) \tag{2.8}$$

and

$$Y^{i+1}(x) = \sum_{k=1}^{3} b_k^{i+1} B_k(x). \tag{2.9}$$

To enforce continuity at x_{i+1} boundaries, we write

$$Y^{i+1}(x_{i+1}) = Y^i(x_{i+1}), \tag{2.10}$$

$$\left.\frac{dY^{i+1}}{dx}\right|_{x_{i+1}} = \left.\frac{dY^i}{dx}\right|_{x_{i+1}}. \tag{2.11}$$

This in the z variable implies

$$Y^{i+1}(0) = Y^i(1), \tag{2.12}$$

$$\left.\frac{dY^{i+1}}{dz}\right|_{z=0} = \left.\frac{dY^i}{dz}\right|_{z=1}, \quad i = 1, 2, \ldots, N-1. \tag{2.13}$$

Equations (2.12) and (2.13) together with (2.5) give

$$b_1^{i+1} = b_3^i, \tag{2.14}$$

$$b_2^{i+1} = 2b_3^i - b_2^i, \tag{2.15}$$

respectively.

Then, for each element i, we may write

$$Y^i(x) = \sum_{k=1}^{3} b_{k+2(i-1)} B_k(x), \tag{2.16}$$

which implies

$$Y(z) = \sum_{k=1}^{3} b_{k+2(i-1)} B_k(z), \quad i = 1, 2, \ldots, N. \tag{2.17}$$

We satisfy the ODE (2.6)on each element by substituting (2.17) in (2.6). The result, together with (2.7), yields

$$\sum_{k=1}^{3} \left[\frac{a_2(z)}{h^2} B_k''(z) + \frac{a_1(z)}{h} B_k'(z) + a_0(z) B_k(z) \right] b_{k+2(i-1)} = f(x_i + hz),\ i = 1, 2, \ldots, N \tag{2.18}$$

and

$$y(0) = b_1 = \theta, \quad y(1) = b_{2N+1} = \Theta. \tag{2.19}$$

Equation (2.18) contains $2N + 1$ unknowns which implies that we require $2N + 1$ conditions. In order to achieve this, we use one collocation point per element, namely $z = 0.5$ in (2.18) , which, together with the two boundary conditions and $N - 1$ continuity equations from (2.15), results in a closed linear system of equations of the form

$$A\mathbf{b} = \mathbf{f}. \tag{2.20}$$

The coefficient matrix A is given by

$$A = \begin{bmatrix}
M_1 & M_2 & M_3 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & M_1 & M_2 & M_3 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & M_1 & M_2 & M_3 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & M_1 & M_2 & M_3 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & M_1 & M_2 & M_3 \\
0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 2 & -1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1
\end{bmatrix}, \tag{2.21}$$

where

$$M_j = \frac{a_2(0.5)}{h^2} B_j''(0.5) + \frac{a_1(0.5)}{h} B_j'(0.5) + a_0(0.5) B_j(0.5), \quad j = 1, 2, 3 \tag{2.22}$$

and the vector of unknowns $\mathbf{b} = [b_1, b_2, b_3, \cdots, b_{2N+1}]^T$, where T represents the

transpose and

$$f_i = \begin{cases} f(x_i + 0.5h) & i = 1, \ldots, N, \\ 0 & i = N+1, \ldots, 2N-1, \\ \theta & i = 2N, \\ \Theta & i = 2N+1. \end{cases} \tag{2.23}$$

The last two rows of the matrix A represent the transformed boundary conditions. Once the b values are obtained, (2.16) is used to reconstruct the solution. We now show that the collocation matrix system (2.21) has a unique solution for any number of finite elements N. The resulting coefficient matrix is a square matrix of order $2N+1$. Expand the determinant $\det(A)$ about the last and second-to-last rows, successively, to obtain $\det(A)$=-$\det(A')$, where A' is the $(2N-1)\times(2N-1)$ submatrix with the first column, last column and last two rows deleted.

Let P be the even permutation matrix given via $P = [\mathbf{e}_1\ \mathbf{e}_3\ \ldots \mathbf{e}_{2N-1}\ \mathbf{e}_2 \ldots \mathbf{e}_{2N-2}]$, where $\mathbf{e}_j$ are the standard basis vectors in $\mathbb{R}^{2N-1}$.

Let

$$A'' = \begin{bmatrix} I_N & \mathbf{0} \\ \mathbf{0} & M_2 I_{N-1} \end{bmatrix} A'P,$$

where I_j is the $j \times j$ identity matrix.

Row reduce A'' by letting $R_{N+j} \to R_{N+j} + R_j$, $j = 1, 2, \ldots, N-1$ to obtain

$$A'' = \begin{bmatrix} I_N & E \\ \mathbf{0} & T \end{bmatrix},$$

where T is the $(N-1)\times(N-1)$ tridiagonal matrix with M_3 on the super diagonal, $\alpha = 2M_2 + M_1 + M_3$ on the diagonal, and M_1 on the subdiagonal.

Let $d_{N-1} = \det(T)$; then following [24] and [71], the determinant of a tridiagonal matrix satisfies the recurrence formula

$$d_k = \alpha d_{k-1} - M_1 M_3 \; d_{k-2}, \;\; k = N-1, \;\; N-2, \ldots, 4, 3, 2, \tag{2.24}$$

with $d_1 = \alpha$ and $d_0 = 1$.

Let $d_k = x^k$ in (2.24), then

$$x^k - \alpha x^{k-1} + M_1 M_3 \; x^{k-2} = 0. \tag{2.25}$$

Divide both sides of equation (2.25) by x^{k-2} gives

$$x^2 - \alpha x + M_1 M_3 = 0$$

and the roots are

$$x_1 = \frac{\alpha + \sqrt{\alpha^2 - 4M_1 M_3}}{2},$$
$$x_2 = \frac{\alpha - \sqrt{\alpha^2 - 4M_1 M_3}}{2}.$$

Thus the solution can be expressed as

$$d_k = c_1 x_1^k + c_2 x_2^k, \quad k = 0, 1, \tag{2.26}$$

with

$$d_0 = 1 = c_1 + c_2, \tag{2.27}$$

$$d_1 = \alpha = c_1 x_1 + c_2 x_2. \tag{2.28}$$

Solution of (2.27) and (2.28) using Cramer's rule yields

$$c_1 = \frac{x_2 - \alpha}{(x_2 - x_1)},$$
$$c_2 = \frac{\alpha - x_1}{(x_2 - x_1)}.$$

Equation (2.26) becomes

$$d_k = \frac{(x_2 - \alpha) \; x_1^k + (\alpha - x_1) \; x_2^k}{x_2 - x_1}. \tag{2.29}$$

Since $x_1 + x_2 = \alpha$ and $x_2 - x_1 = -\sqrt{\alpha^2 - 4M_1M_3}$, it follows from (2.29) that

$$\begin{aligned} d_k &= \frac{-x_1^{k+1} + x_2^{k+1}}{x_2 - x_1}, \\ &= \frac{(\alpha + \sqrt{\alpha^2 - 4M_1M_3})^{k+1} - (\alpha - \sqrt{\alpha^2 - 4M_1M_3})^{k+1}}{2^{k+1}\sqrt{\alpha^2 - 4M_1M_3}}. \end{aligned}$$

Thus $\det(T) = d_{N-1}$.

Now

$$\begin{aligned} \det(A'') &= \det(A'''), \\ M_2^{N-1}\det(A') &= M_2^N \det(T), \\ \det(A') &= M_2 \det(T), \\ \det(A) &= \frac{-M_2\left(\left[\alpha + \sqrt{\alpha^2 - 4M_1M_3}\right]^N - \left[\alpha - \sqrt{\alpha^2 - 4M_1M_3}\right]^N\right)}{2^N\sqrt{\alpha^2 - 4M_1M_3}}. \end{aligned}$$

Hence the matrix A is nonsingular provided $\alpha^2 \neq 4M_1M_3$ and $M_2 \neq 0$.

2.3 Error estimates

In this section, we show the error estimates numerically.

Theorem 2.1: *Suppose the coefficients $a_i(x)$ in (2.6) satisfies $a_i(x) \in C^4[a, b]$. If Gauss' points are chosen as the collocation points then $\exists$ K_1 and K_2 such that for $j \in \{0, 1, 2\}$ [20]*

$$|D^j(y - Y)(x_i)|^h \leq K_1 h^2, \tag{2.30}$$

and

$$||D^j(y - Y)(x_i)||_\infty^h \leq K_2 h^2. \tag{2.31}$$

Suppose the errors at the node x_i is $O(h^m)$ then

$$|D^j(y - Y)(x_i)|^h = O(h^m), \tag{2.32}$$

and

$$|D^j(y - Y)(x_i)|^{(\frac{h}{2})} = O\left(\frac{h}{2}\right)^m. \tag{2.33}$$

Using (2.32) and (2.33), we compute the ratio

$$\begin{aligned}\beta_1 &= \frac{|D^j(y-Y)(x_i)|^h}{|D^j(y-Y)(x_i)|^{\frac{h}{2}}} \\ &\approx 2^m.\end{aligned} \tag{2.34}$$

This implies that the order of convergence at the node is

$$m(h) = \frac{\ln \beta_1}{\ln 2}. \tag{2.35}$$

Similarly, we compute the ratio

$$\begin{aligned}\beta_2 &= \frac{||D^j(y-Y)(x)||_\infty^h}{||D^j(y-Y)(x)||_\infty^{\frac{h}{2}}} \\ &\approx 2^m.\end{aligned} \tag{2.36}$$

This implies that the global order of convergence is

$$m_2(h) = \frac{\ln \beta_2}{\ln 2}. \tag{2.37}$$

2.4 Application of quadratic OCFE to ODEs

In this section, we demonstrate the applicability of OCFE based on quadratic B-spline functions to linear and nonlinear ordinary differential equations. The approximate solution to the differential equation (2.6) can be obtained by substituting $b_i,\ i = 1, 2,\ \ldots,\ 2N+1$ into equation (2.17).

2.4.1 Linear ODE

Example 2.1: As a test case, we consider the linear second order boundary value problem (BVP)

$$\frac{d^2y}{dx^2} - 100y = 0, \quad y(0) = y(1) = 1. \tag{2.38}$$

The exact solution is

$$y(x) = \frac{\cosh(10x-5)}{\cosh(5)}. \tag{2.39}$$

We set $a_2(x) = 1$, $a_1(x) = 0$, $a_0(x) = -100$, $f(x) = 0$ in equation (2.18) and $\theta = \Theta = 1$ in equation (2.19). We then solve the resulting linear system (2.20). The rate of convergence is calculated as $\log_2 \frac{||\mathbf{e}||_\infty^h}{||\mathbf{e}||_\infty^{h/2}}$ where the components of $\mathbf{e}$ are given by $e_i = y_i(x) - Y_i(x)$, $y_i(x)$ and $Y_i(x)$ are exact and approximate solution at the nodes x_i respectively.

The approximate solution and errors for this example are depicted in Figures 2.4 and 2.5 respectively. Figure 2.4 shows that the approximate solutions at the nodes are very close to the exact solution.

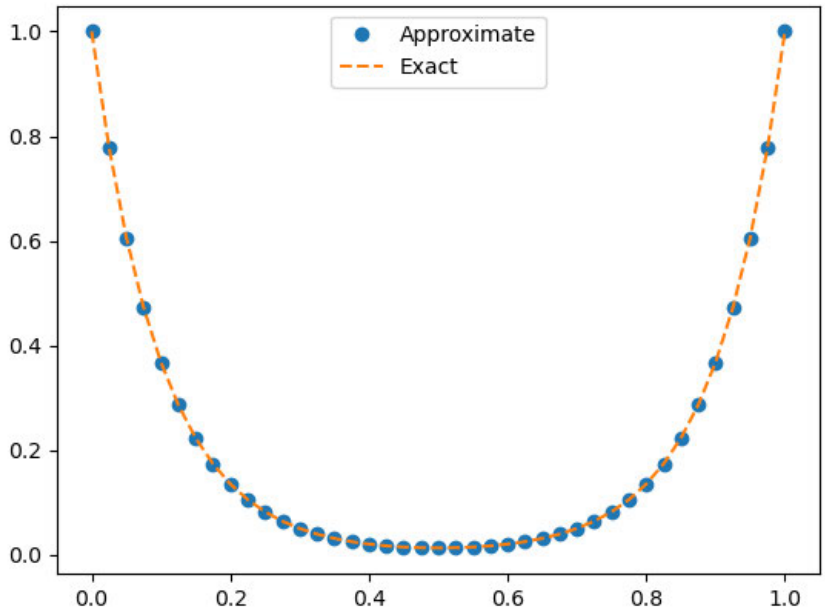

Figure 2.4: Plot of the approximate and exact solutions of $y(x)$ when $N = 40$.

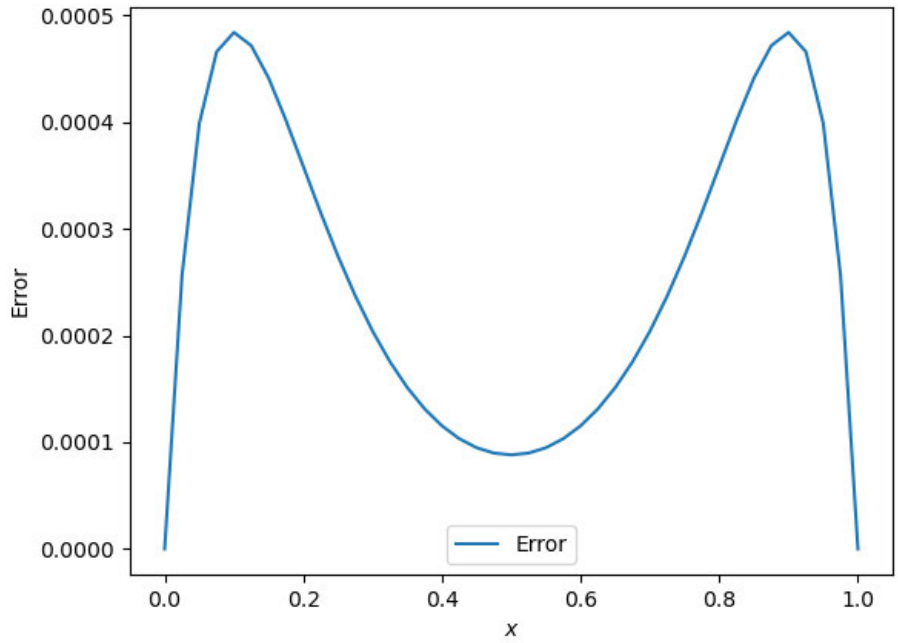

Figure 2.5: Plot of the error for example 2.1 when $N = 40$.

The convergence rate at the nodes is approximately 2 as displayed in Table 2.2. Table 2.1 below shows the order of convergence using the OCFE method and the classical collocation method using quadratic splines studied in [**13**]. As expected, the OCFE method converges faster than the classical quadratic spline collocation method. Hence quadratic B-spline OCFE can solve linear ordinary differential equations.

Table 2.1: Convergence rates at different step sizes

h	Khalifa [**13**]	OCFE
$\frac{1}{20}$	1.76	2.12
$\frac{1}{40}$	1.93	2.03
$\frac{1}{80}$	1.96	2.00

Table 2.2: Convergence rates $m(h)$ at the nodes when $N = 40$.

x	$j = 0$	$j = 1$	x	$j = 0$	$j = 1$
0.025	2.0087	1.9945	0.475	2.0054	1.9984
0.050	2.0085	1.9952	0.525	2.0054	1.9984
0.075	2.0083	1.9958	0.550	2.0055	1.9984
0.100	2.0081	1.9963	0.575	2.0056	1.9984
0.125	2.0080	1.9967	0.600	2.0058	1.9984
0.150	2.0078	1.9970	0.625	2.0060	1.9983
0.175	2.0076	1.9973	0.650	2.0062	1.9983
0.200	2.0074	1.9975	0.675	2.0064	1.9982
0.225	2.0072	1.9977	0.700	2.0066	1.9981
0.250	2.0070	1.9979	0.725	2.0068	1.9980
0.275	2.0068	1.9980	0.750	2.0070	1.9979
0.300	2.0066	1.9981	0.775	2.0072	1.9977
0.325	2.0064	1.9982	0.800	2.0074	1.9975
0.350	2.0062	1.9983	0.825	2.0076	1.9973
0.375	2.0060	1.9983	0.850	2.0078	1.9970
0.400	2.0058	1.9984	0.875	2.0080	1.9967
0.425	2.0056	1.9984	0.900	2.0081	1.9963
0.450	2.0055	1.9984	0.925	2.0083	1.9958

Table 2.3: Global convergence rates

N	$j = 0$	$j = 1$
10	2.1283	1.4244
20	2.0324	1.7282
30	2.0145	1.8245
40	2.0081	1.8708
50	2.0052	1.8979
100	2.0013	1.9504

The global order of convergence in Table 2.3 approaches 2 as the number of intervals N increases. Hence the global order and convergence rates at the nodes are the same for the linear ordinary differential equation.

2.4.2 Nonlinear ODE

Example 2.2: We take the nonlinear boundary value problem

$$y''(x) - (y'(x))^2 - y(x) = -e^{-2x}, \quad y(0) = 1, y(1) = e^{-1}. \tag{2.40}$$

The exact solution is $y(x) = e^{-x}$.

The approximate solutions and errors for this problem are plotted in Figures 2.6 and 2.7 respectively. Figure 2.6 shows that the approximate solutions at the nodes are very close to the exact solutions.

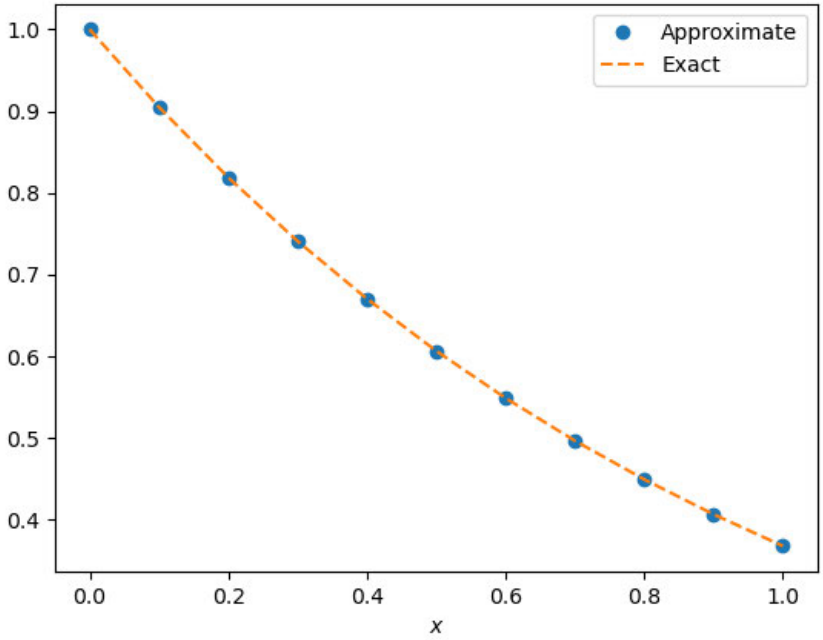

Figure 2.6: Plot of the solution for example 2.2 when $N = 10$.

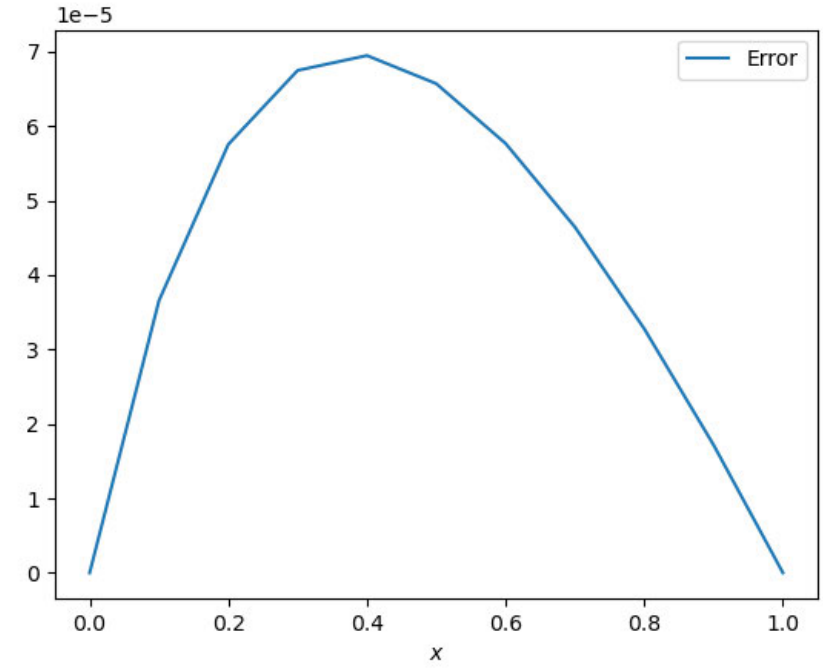

Figure 2.7: Plot of the error for example 2.2 when $N = 10$.

In Table 2.4 the convergence rates at the nodes are approximately 2.

Table 2.4: Convergence rates at the nodes when $N = 10$.

x	$j = 0$	$j = 1$
0.1	2.0088	2.0038
0.2	2.0080	2.0033
0.3	2.0073	2.0028
0.4	2.0068	2.0025
0.5	2.0063	2.0022
0.6	2.0059	2.0019
0.7	2.0056	2.0017
0.8	2.0053	2.0016
0.9	2.0050	2.0015

Suppose the global error $|D^j(y - Y)(x)|_\infty$ is $O(h^{-m}),\ j = 0,\ 1$. We define the ratio

$$\begin{aligned} \beta_3 &= \frac{|D^j(y - Y)(x)|_\infty^N}{|D^j(y - Y)(x)|_\infty^{N+1}} \\ &\approx 2^m \end{aligned} \tag{2.41}$$

This implies that the global order of convergence is

$$m_3(h) = \frac{\ln \beta_3}{\ln\left(\frac{N+1}{N}\right)}. \tag{2.42}$$

We compute the global order using equation (2.4.2). The results are shown in Table 2.5. Hence the global order of convergence is approximately 2.

Table 2.5: Global convergence rates for different values of N.

N	$j = 0$	$j = 1$
10	2.0068	2.0024
20	2.0001	2.0009
30	1.9986	2.0004
40	2.0001	2.0002
50	2.0007	2.0002
60	2.0007	1.9996
70	2.0002	2.0001

2.5 Discussion of Chapter 2

We have successfully used the quadratic B-spline OCFE to solve linear and nonlinear ordinary differential equations. We also showed that it's order of convergence is 2. This is due to the fact that B-splines with the continuity conditions yield the smoothest spline, has minimum compact support [14], shape preserving and flexible to apply to differential equations. Moreover, orthogonal collocation on finite elements (OCFE) is more accurate than the classical collocation. We leveraged on this property of OCFE and that of B-splines to solve one-dimensional problems that has solutions with steep gradients. The resulting coefficient matrix is sparse and the boundary conditions do not require special treatment. First derivative also has order of convergence 2. In the next chapter we shall extend our method to solving partial differential equations.

CHAPTER THREE

APPLICATION OF QUADRATIC B-SPLINE OCFE TO PDEs: BURGERS' EQUATION

We present numerical solutions to partial differential equations using quadratic B-spline OCFE. For ease of exposition we illustrate the OCFE method on Burgers' equation. A complete analysis of the stability and convergence of the method is provided. Various numerical simulations for Burgers' and the modified Burgers' equations are also presented. We found that the quadratic OCFE method outperforms the classical quadratic spline collocation method.

3.1 OCFE applied to Burgers' equation

Consider the viscous Burgers' equation [[1], [2]]

$$u_t = \nu u_{xx} - u u_x, \quad a < x < b, \quad t \geq t_0, \tag{3.1}$$

with the boundary conditions

$$u(a,t) = \theta,\ u(b,t) = \Theta, \tag{3.2}$$

and initial condition

$$u_0(x) = u(x,t_0). \tag{3.3}$$

For a partial differential equation (PDE) in space and time, we write the trial solution as

$$u(z,t) = \sum_{k=1}^{3} b_{k+2(i-1)}(t)\ B_k(z). \tag{3.4}$$

Discretization of (3.1) into N finite elements, similar to the ODE case, and substitution of (3.4) in (3.1) yields

$$\begin{aligned}\sum_{k=1}^{3} b'_{k+2(i-1)}(t)\ B_k(z) &= \frac{\nu}{h^2}\sum_{k=1}^{3} b_{k+2(i-1)}(t)\ B''_k(z)\\ &- \frac{1}{h}\left(\sum_{k=1}^{3} b_{k+2(i-1)}(t)\ B_k(z)\right)\left(\sum_{k=1}^{3} b_{k+2(i-1)}(t)\ B'_k(z)\right),\ i=1,2,\ldots,N.\end{aligned} \tag{3.5}$$

The boundary conditions (3.2) (for simplicity, choosing $a=0,\ b=1$) yield

$$b_1(t)=\theta,\ b_{2N+1}(t)=\Theta. \tag{3.6}$$

Equations (3.5) and (3.6) give a system of differential-algebraic equations (DAEs) which has to be solved in time. Unfortunately, for large N, solving this system could prove to be computationally challenging. A common alternative used in the literature, which avoids dealing with DAEs, is to use the Crank–Nicolson method with quasi-linearization to accomplish time integration. Furthermore, the stability and convergence of the method is easily established. In the next section, we apply the quasilinearization method to Burgers' equation.

Now we apply Trapezoid method to equation (3.1).

Given $y'=f(t,y)$, let $y(t_j)=y_j$ then

$$y_{j+1}=y_j+\frac{\Delta t}{2}\left(f(t_{j+1},y_{j+1})+f(t_j,y_j)\right),\quad j=1,..,N. \tag{3.7}$$

Applying equations (3.7) in (3.1),

$$\begin{aligned}u(x,t_{j+1}) = u(x,t_j)+\frac{\Delta t}{2}\left[\nu u_{xx}(x,t_{j+1})-u_x(x,t_{j+1})u(x,t_{j+1})\right.\\ \left.+\nu u_{xx}(x,t_j)-u_x(x,t_j)u(x,t_j)\right].\end{aligned} \tag{3.8}$$

Expand $u(x,t_{j+1})u_x(x,t_{j+1})$ by Taylor's theorem to linearize it. Let

$$\phi(t_{j+1})=u(x,t_{j+1})u_x(x,t_{j+1})=\phi(t_j+\Delta t), \tag{3.9}$$

$$\begin{aligned}\phi(t_{j+1}) &= \phi(t_j) + \phi'(t_j)\Delta t + O((\Delta t)^2) \\ &= u(x,t_j)u_x(x,t_j) + \left[\frac{\partial \phi}{\partial u}u_t + \frac{\partial \phi}{\partial u_x}u_{xt}\right]\Delta t + O((\Delta t)^2) \\ &= u(x,t_j)u_x(x,t_j) + \left[u_x(x,t_j)u_t(x,t_j) + u(x,t_j)u_{xt}(x,t_j)\right]\Delta t + O((\Delta t)^2).\end{aligned} \tag{3.10}$$

Now use the forward difference approximations

$$\begin{aligned}u_t &= \frac{u(x,t_{j+1}) - u(x,t_j)}{\Delta t}, \\ u_{xt} &= \frac{u_x(x,t_{j+1}) - u_x(x,t_j)}{\Delta t},\end{aligned} \tag{3.11}$$

and substitute equation(3.11) in (3.10) to obtain

$$\begin{aligned}\phi(t_{j+1}) = u(x,t_j)u_x(x,t_j) + u(x,t_{j+1})u_x(x,t_j) - u(x,t_j)u_x(x,t_j) \\ + u(x,t_j)u_x(x,t_{j+1}) - u(x,t_j)u_x(x,t_j) + O((\Delta t)^2).\end{aligned} \tag{3.12}$$

This implies

$$\phi(t_{j+1}) = u(x,t_{j+1})u_x(x,t_j) - u(x,t_j)u_x(x,t_j) + u(x,t_j)u_x(x,t_{j+1}) + O((\Delta t)^2). \tag{3.13}$$

Substitute equation(3.13) in (3.8), to get

$$\begin{aligned}u(x,t_{j+1}) &= u(x,t_j) + \frac{\Delta t}{2}\left(\nu u_{xx}(x,t_{j+1}) - \left[u(x,t_{j+1})u_x(x,t_j) - u(x,t_j)u_x(x,t_j)\right.\right. \\ &\quad \left.\left. + u(x,t_j)u_x(x,t_{j+1}) + O((\Delta t)^2)\right] + \nu u_{xx}(x,t_j) - u_x(x,t_j)u(x,t_j)\right) \\ &= u(x,t_j) + \frac{\Delta t}{2}\left[\nu u_{xx}(x,t_{j+1}) - u(x,t_{j+1})u_x(x,t_j) + u(x,t_j)u_x(x,t_j)\right. \\ &\quad \left. - u(x,t_j)u_x(x,t_{j+1}) + \nu u_{xx}(x,t_j) - u_x(x,t_j)u(x,t_j)\right] + O((\Delta t)^3) \\ &= u(x,t_j) + \frac{\Delta t}{2}\left[\nu u_{xx}(x,t_{j+1}) - u(x,t_{j+1})u_x(x,t_j) - u(x,t_j)u_x(x,t_{j+1})\right. \\ &\quad \left. + \nu u_{xx}(x,t_j)\right] + O((\Delta t)^3).\end{aligned} \tag{3.14}$$

Ignoring the error term, the linearized Burgers' equation takes the form

$$\begin{aligned}\left[1 + \frac{\Delta t}{2}u_x(x,t_j)\right]u(x,t_{j+1}) - \frac{\nu \Delta t}{2}u_{xx}(x,t_{j+1}) + \frac{\Delta t}{2}u(x,t_j)u_x(x,t_{j+1}) \\ = u(x,t_j) + \frac{\nu \Delta t}{2}u_{xx}(x,t_j).\end{aligned} \tag{3.15}$$

Since

$$u(x,t_j)=\sum_{k=1}^{3} b_{k+2(i-1)}(t_j)B_k(x), \tag{3.16}$$

using the transformation $z=\frac{x-x_i}{h}$ in (3.15), we have

$$\begin{aligned}
&\sum_{k=1}^{3}\left(\left[1+\frac{\Delta t}{2h}\sum_{k=1}^{3} b_{k+2(i-1)}(t_j)\ B_k'(z)\right]B_k(z)-\frac{\Delta t}{2h^2}\nu\ B_k''(z)\right.\\
&\left.+\frac{\Delta t}{2h}\left[\sum_{k=1}^{3} b_{k+2(i-1)}(t_j)B_k(z)\right]B_k'(z)\right)b_{k+2(i-1)}(t_{j+1})\\
&=\sum_{k=1}^{3}\left[B_k(z)+\frac{\Delta t}{2h^2}\nu\ B_k''(z)\right]b_{k+2(i-1)}(t_j),\quad i=1,2,\ldots,N.
\end{aligned} \tag{3.17}$$

Substituting one collocation point per interval for z in equation (3.17) and using the boundary conditions will give a system of equations of the form $Q\ \mathbf{b}(t_{j+1})=P\ \mathbf{b}(t_j)$, where the entries of P and Q are defined as follows:

$$P_{i,2(i-1)+k}=B_k(0.5)+\frac{\Delta t\nu}{2h^2}B_k''(0.5),\quad k=1,2,3\quad i=1,2,\ldots,N, \tag{3.18}$$

$$P_{2N,1}=1, \tag{3.19}$$

$$P_{2N+1,2N+1}=1, \tag{3.20}$$

$$P_{i,j}=0,\quad \text{otherwise.} \tag{3.21}$$

Define

$$S_i^1=\sum_{k=1}^{3} b_{k+2(i-1)}(t_j)B_k'(0.5), \tag{3.22}$$

$$S_i^0=\sum_{k=1}^{3} b_{k+2(i-1)}(t_j)B_k(0.5). \tag{3.23}$$

then

$$\begin{aligned}
Q_{i,2(i-1)+k}&=\left(1+\frac{\Delta t}{2h}S_i^1\right)B_k(0.5)-\frac{\Delta t\nu}{2h^2}B_k''(0.5)+\frac{\Delta t}{2h}S_i^0B_k'(0.5),\\
&k=1,2,3,\quad i=1,2\ldots,N.
\end{aligned} \tag{3.24}$$

$$\left.\begin{aligned} Q_{N+i,2i} &= -1 \\ Q_{N+i,2i+1} &= 2 \\ Q_{N+i,2i+2} &= -1 \end{aligned}\right\}, \quad i = 1, 2, \ldots, N-1 \quad \text{(for continuity conditions).} \tag{3.25}$$

$$Q_{2N,1} = 1, \tag{3.26}$$

$$Q_{2N+1,2N+1} = 1, \tag{3.27}$$

$$Q_{i,j} = 0, \quad \text{otherwise.} \tag{3.28}$$

Note that $Q \sim A$ in equation (2.21) and $P\mathbf{b} \sim \mathbf{f}$ in (2.23).

3.2 Stability of the Quadratic B-Spline OCFE Method

The von Neumann analysis technique is used to determine the stability of a numerical method for linear initial value problems and linearized nonlinear boundary value problems.

Let $K = \max\{u\}$ be a local constant representing u in the nonlinear term of equation (3.1), and use the Crank–Nicolson method for discretization. We have

$$u^{n+1} + \frac{\Delta t K}{2} u_x^{n+1} - \frac{\Delta t}{2} \nu u_{xx}^{n+1} = u^n - \frac{\Delta t K}{2} u_x^n + \frac{\Delta t}{2} \nu u_{xx}^n, \tag{3.29}$$

where Δt is the time step. Transforming to variable z, equation (3.29) becomes

$$u^{n+1} + \frac{\alpha}{2} u_z^{n+1} - \frac{\beta}{2} u_{zz}^{n+1} = u^n - \frac{\alpha}{2} u_z^n + \frac{\beta}{2} u_{zz}^n, \tag{3.30}$$

where $\alpha = \frac{\Delta t K}{h}$ and $\beta = \frac{\Delta t \nu}{h^2}$. Since

$$u^n(z) = \sum_{l=1}^{3} b_{l+2(m-1)}^n B_l(z), \tag{3.31}$$

Equation (3.30) can be written as

$$\sum_{l=1}^{3} b_{l+2(m-1)}^{n+1} \left[B_l(z) + \frac{\alpha}{2} B_l'(z) - \frac{\beta}{2} B_l''(z) \right] = \sum_{l=1}^{3} b_{l+2(m-1)}^{n} \left[B_l(z) - \frac{\alpha}{2} B_l'(z) + \frac{\beta}{2} B_l''(z) \right]. \tag{3.32}$$

Then, at the collocation point $z = \frac{1}{2}$,

$$b_{2m-1}^{n+1}\sigma_1 + b_{2m}^{n+1}\sigma_2 + b_{2m+1}^{n+1}\sigma_3 = b_{2m-1}^{n}\rho_1 + b_{2m}^{n}\rho_2 + b_{2m+1}^{n}\rho_3, \tag{3.33}$$

where

$$\begin{aligned} \rho_1 &= \tfrac{1}{4} - \tfrac{\alpha}{2} + \beta, & \sigma_1 &= \tfrac{1}{4} + \tfrac{\alpha}{2} - \beta, \\ \rho_2 &= \tfrac{1}{2} - 2\beta, & \sigma_2 &= \tfrac{1}{2} + 2\beta, \\ \rho_3 &= \tfrac{1}{4} + \tfrac{\alpha}{2} + \beta, & \sigma_3 &= \tfrac{1}{4} - \tfrac{\alpha}{2} - \beta. \end{aligned}$$

Let $b_l^n = \lambda^n e^{ilhk}$, k= mode and $i = \sqrt{-1}$, then (3.33) gives

$$\begin{aligned} &\lambda^{n+1}\left[e^{i(2m-1)hk}\sigma_1 + e^{i2mhk}\sigma_2 + e^{i(2m+1)hk}\sigma_3\right] \\ &= \lambda^{n}\left[e^{i(2m-1)hk}\rho_1 + e^{i2mhk}\rho_2 + e^{i(2m+1)hk}\rho_3\right], \end{aligned} \tag{3.34}$$

$$\implies \lambda = \frac{E_1 + iF_1}{E_2 + iF_2}, \tag{3.35}$$

where $E_1 = 2\rho_1 \cos hk + \rho_2 + \alpha\cos(hk)$, $E_2 = 2\sigma_1 \cos hk + \sigma_2 - \alpha\cos(hk)$, $F_1 = \alpha\sin(hk)$, and $F_2 = -\alpha\sin(hk)$. Therefore

$$|\lambda|^2 = \frac{E_1^2 + F_1^2}{E_2^2 + F_2^2} = \frac{N}{D}, \tag{3.36}$$

$$\begin{aligned} N - D = {} & 4(\rho_1^2 - \sigma_1^2)\cos^2(hk) + (\rho_2^2 - \sigma_2^2) + 4(\rho_1\rho_2 - \sigma_1\sigma_2)\cos(hk) \\ & + 2\alpha(\rho_2 + \sigma_2)\cos(hk) + 4\alpha(\rho_1 + \sigma_1)\cos^2(hk), \end{aligned} \tag{3.37}$$

where

$$\begin{aligned} \rho_1 - \sigma_1 &= -\alpha + 2\beta, & \rho_2 - \sigma_2 &= -4\beta, \\ \rho_1 + \sigma_1 &= \tfrac{1}{2}, & \rho_2 + \sigma_2 &= 1, \\ \rho_2^2 - \sigma_1^2 &= \beta - \tfrac{\alpha}{2}, & \rho_2^2 - \sigma_2^2 &= -4\beta, \\ \rho_1\rho_2 - \sigma_1\sigma_2 &= \tfrac{-\alpha}{2}. \end{aligned}$$

This gives

$$\begin{aligned} N - D &= (4\beta - 2\alpha)\cos^2(hk) - 4\beta - 2\alpha\cos(hk) + 2\alpha\cos(hk) + 2\alpha\cos^2(hk) \\ &= -4\beta(1 - \cos^2(hk)) = -4\beta\sin^2(hk) \leq 0. \end{aligned} \tag{3.38}$$

Therefore $N \leq D \implies \frac{N}{D} \leq 1$. Hence $|\lambda| = \sqrt{\frac{N}{D}} \leq 1$.

This shows that orthogonal quadratic spline collocation on finite elements using Gauss points for Burgers' equation is unconditionally stable.

3.3 Convergence of the Method

Suppose the partial differential equation

$$u_t = u_{xx} + q_1(x)u_x + q_0(x)u, \quad t_0 < t < t_f, \tag{3.39}$$

with boundary conditions

$$u(a,t) = \beta, \quad u(b,t) = \gamma, \quad a < x < b, \tag{3.40}$$

where $q_1(x)$ and $q_0(x)$ are coefficients of u_x and u respectively, has the initial condition $u(x,t_0) = \alpha(x)$. We assume that the exact solution of equation (3.39) is $\bar{u}(x,t) \in C^4[a,b] \times C^2[t_0,t_f]$, where t_0 and t_f are initial and final times, respectively. For time, we use the discretization $t_j = t_0 + j\Delta t$ and the trapezoidal rule to effect time integration. From the trapezoid rule, the local error is given by $\frac{\Delta t^3}{12}\ u_{tt}(x,\xi_j), \xi_j \in [t_j, t_{j+1}]$.

Let $u_L(x,t_{j+1})$ be the solution of the time discretized form of equation (3.39). At time t_{j+1}, the global error is $t_{j+1}K'\Delta t^2$ for constant K'. Hence

$$||\bar{u}(x,t_{j+1}) - u_L(x,t_{j+1})||_\infty \leq K\Delta t^2 \tag{3.41}$$

for some constants K.

We let $u(x,t_j)=u^j(x)$, then the time discretized form of equation (3.39) is

$$u_{xx}^{j+1}(x)+a_1(x)u_x^{j+1}(x)+a_0(x)u^{j+1}(x)=f(x). \tag{3.42}$$

where $f(x)=-\left(\left[1+\frac{\Delta t}{2}q_0(x)\right]u^j(x)+\frac{\Delta t}{2}q_1(x)u_x^j(x)+\frac{\Delta t}{2}u_{xx}(x)\right)$, $a_0(x)=\frac{\Delta t}{2}q_0(x)-1$ and $a_1(x)=\frac{\Delta t}{2}q_1(x)$. We assume that $a_0(x),\ a_1(x)\in C^2[a,b]$. Let $u_c(x,t)\in P_3\cap C^1[a,b]$ be the collocation solution to (3.42), where P_3 denotes the space of piecewise polynomials of degree <3. From equation 2.30,

$$|D^{(i)}(u_L^{j+1}-u_c^{j+1})(x_p)|^h\le C_1h^2, \tag{3.43}$$

and

$$||D^{(i)}(u_L^{j+1}-u_c^{j+1})(x_p)||_\infty^h\le C_1h^{1+\min(1,2-i)},\quad i=0,1,2. \tag{3.44}$$

Inequalities (3.41) and (3.43) imply

$$||\bar{u}(x,t_{j+1})-u_c(x,t_{j+1})||_\infty^h\le K\Delta t^2+C_1h^2. \tag{3.45}$$

This establishes the rate of convergence of the method is second order in both time and space.

3.4 Numerical Examples and Simulations for Burgers' Equation

In this section, we consider various examples and present various numerical simulations to demonstrate the efficiency of the OCFE method.

Example 3.1: We consider the Burgers' Equation (3.1). The exact solution to (3.1) is

$$u(x,t)=\frac{x}{t\left(1+\sqrt{\frac{t}{\tau}}e^{\frac{x^2}{4\nu t}}\right)}, \tag{3.46}$$

where $\tau = \exp(\frac{1}{8\nu})$, and the initial condition is

$$u_0(x) = u(x, 1). \tag{3.47}$$

The boundary conditions are $u(0,t) = 0$ and $u(1,t) = 0$. The result of our method is presented when the number of partitions of the space interval [0,1] is $N = 50$, the number of time steps is $Nt = 50$ and the final time is $t_f = 2$. The 3D plot of the solution is shown in Figure 3.1 and error in Figure 3.2.

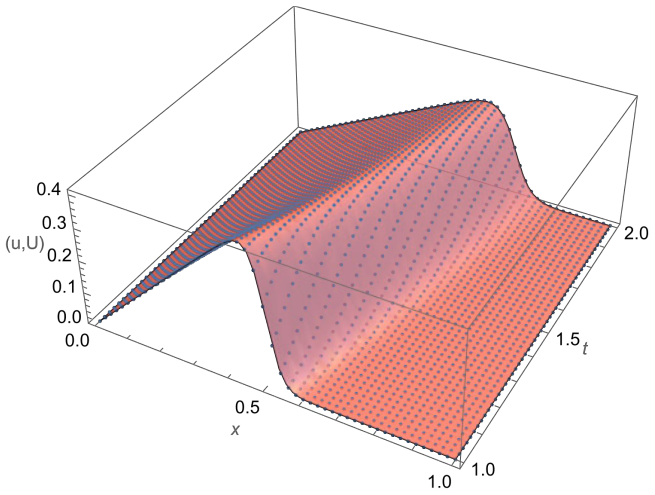

Figure 3.1: Approximate solution of the Burgers' equation when $N = Nt = 50$.

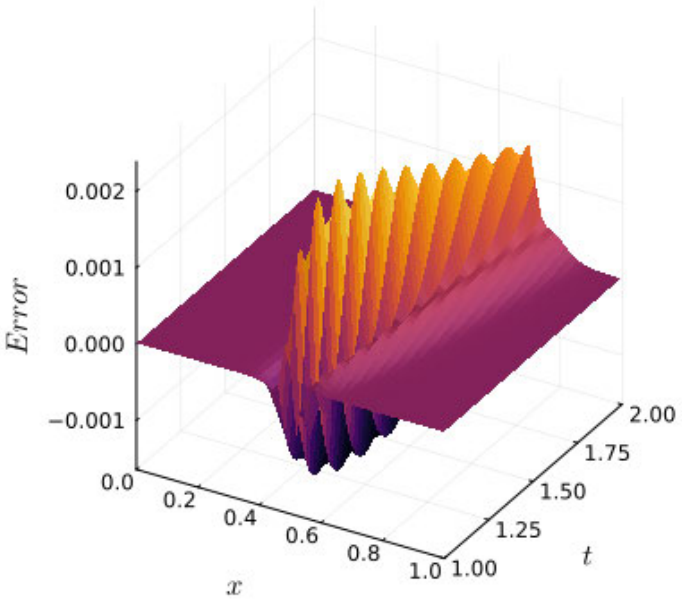

Figure 3.2: 3D plot of the error for example 3.1 when $N = Nt = 50$ and $\nu = 0.005$.

Table 3.1 shows that the convergence rate is approximately 2.

Table 3.1: Convergence rates for example 3.1 when $\nu = 0.005$, $N = Nt = 50$.

x	Rate	x	Rate
0.1	1.9319	0.60	1.8457
0.2	1.8753	0.72	1.4772
0.3	1.8352	0.84	2.0333
0.4	1.8409	0.90	2.2622
0.5	1.8740	0.96	1.9796

We compare the results of our method with the work of Raslan [56] on Burgers' equation (3.1). Table 3.2 indicates that our results are better than those of Raslan [56], since both the L_∞ and the L_2 norm values are smaller. The table also show that the L_∞ norm of the present work is better than that of the Crank–Nicolson method (CN) based on [56].

The CN based on [56] performed better than Raslan [56], as shown in the table. Hence, both CN and the present method, quadratic B-spline collocation using Gauss points on finite elements, compare favourably.

Table 3.2: Errors for example 3.1 at $N = Nt = 50,\ T_f = 2$.

	L_∞			L_2		
t	Present	Raslan [56]	CN	Present	Raslan [56]	CN
1.2	0.002389	0.009445	0.002455	0.003594	0.014613	0.003630
1.4	0.002140	0.009192	0.002201	0.003779	0.019857	0.003762
1.6	0.001868	0.008531	0.001905	0.003589	0.024209	0.003546
1.8	0.001653	0.010477	0.001667	0.003310	0.027808	0.003258
2.0	0.001431	0.012058	0.001426	0.003025	0.030687	0.002972

In Table 3.3, the invariants for the present work, Raslan [56] and CN based on [56] are compared and found to agree with one another.

Table 3.3: Invariant for example 3.1 at $N = Nt = 50,\ T_f = 2$.

t	Present	Raslan [56]	CN
1.0	0.124967	0.124817	0.124967
1.2	0.124059	0.124317	0.124059
1.4	0.123291	0.123954	0.123291
1.6	0.122626	0.123602	0.122626
1.8	0.122039	0.123264	0.122039
2.0	0.121513	0.122940	0.000000

Exact solutions $u(x, 2)$ and approximate solutions with the present method and that of [56] are compared for some values of x in Table 3.4. The Table shows that the present method converges to the exact solution faster than that of Raslan [56].

Table 3.4: Comparison of $u(x, 2)$ and approximate values at some points x.

$\boldsymbol{x}$	Present	Exact	Raslan [56]
0.365	0.182978	0.182473	0.185924
0.415	0.206938	0.207419	0.215954
0.465	0.232783	0.232227	0.235308
0.605	0.289358	0.288192	0.293358
0.645	0.274683	0.274869	0.279174
0.695	0.181949	0.180506	0.150321
0.725	0.095053	0.098406	0.096996
0.765	0.028298	0.029633	0.030028
0.805	0.006413	0.006911	0.007348
0.845	0.001254	0.001413	0.001598
0.915	6.11×10^{-5}	7.05×10^{-5}	0.000140

Furthermore, in Figure 3.3, the convergence of the different methods and the CPU times for Raslan [56], the Crank–Nicolson method based on Raslan, and quadratic spline collocation on Gauss points for computation of this example are shown. The OCFE method has order-two convergence, and the present method is the fastest of the three methods and nearly five times faster than Raslan [56] and four times faster than CN.

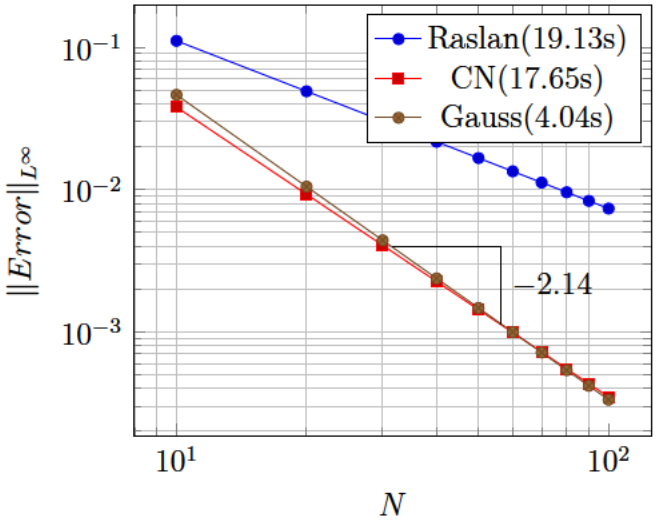

Figure 3.3: Convergence plot for example 3.1.

Example 3.2: A travelling wave solution of Burgers' Equation (3.1), which is of the form

$$u(x,t) = \frac{\mu + \alpha + (\mu - \alpha)\, e^{\frac{\alpha\,\xi}{\nu}}}{1 + e^{\frac{\alpha\,\xi}{\nu}}}, \tag{3.48}$$

with the initial condition

$$u_0(x) = u(x,0) \tag{3.49}$$

and the boundary conditions $u(0,t) = 1,\ u(1,t) = 0.2$, where $\xi = x - \mu\, t - \beta$, μ and α are constants, is considered.

We compute the solution at various times when the number of partitions of the space interval [0,1] is $N = 50$, the number of time steps is $Nt = 50$ and the final time is $T_f = 2$. Figure 3.4 shows the profiles at some values of t for all x. The graph of the exact solution is overlaid onto that of the approximate solution in Figure 3.5. The exact and approximate solutions match perfectly on the computational domain. Figure 3.6 shows that our method has convergence order two.

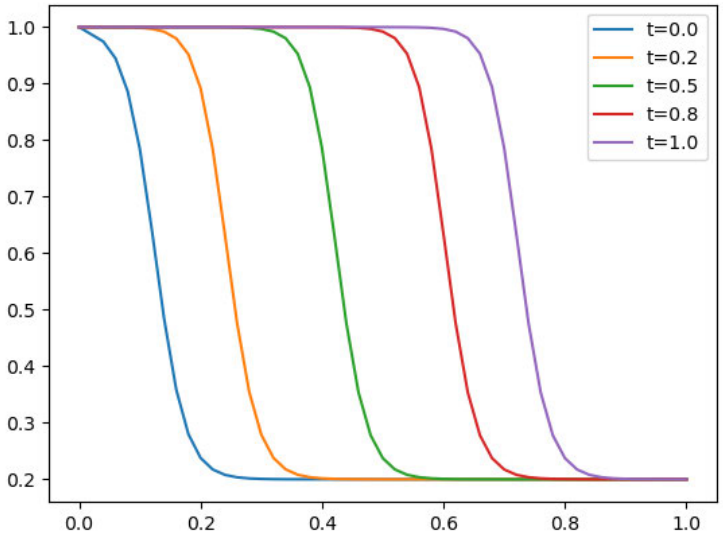

Figure 3.4: Final solution profiles for example 3.2 at different times when $N = Nt = 50$.

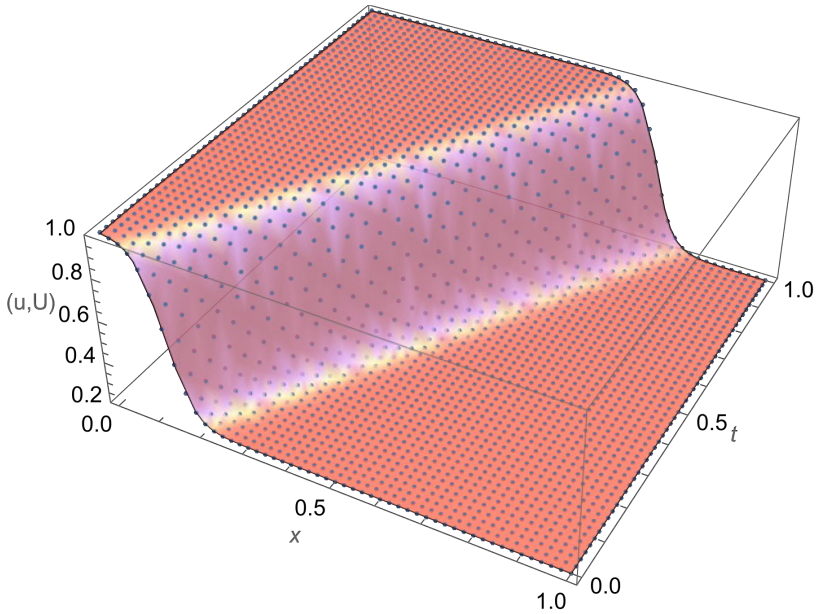

Figure 3.5: Approximate travelling wave solution of the Burgers' equation when $N = Nt = 50$.

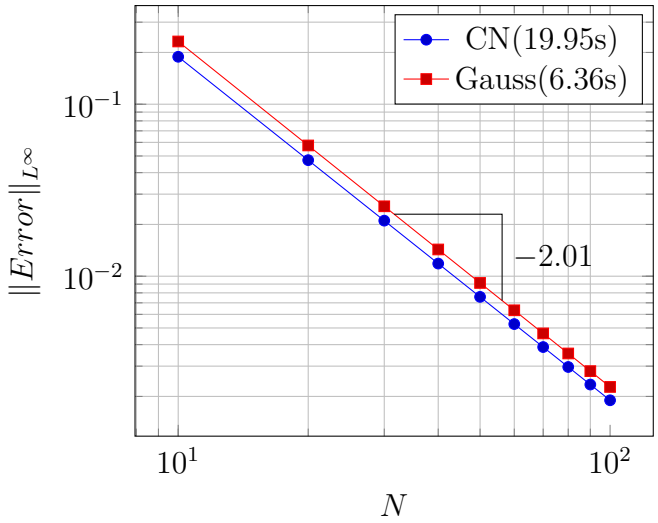

Figure 3.6: Convergence plot for CN and Gauss.

3.5 Modified Burgers' Equation

Example 3.3: We apply the OCFE method with a quadratic basis to the modified Burgers' equation [[15,18,31,38,40,61]].

$$u_t + u^\delta u_x - \nu u_{xx} = 0, \quad \delta \geq 2. \tag{3.50}$$

Consider the case of $\delta = 2$, $\nu = 0.005$ with the initial condition

$$u(x,1) = \frac{x}{1 + 2\ e^{\frac{x^2}{4\nu}}} \tag{3.51}$$

and boundary conditions

$$u(0,t) = 0, \quad u(1,t) = 0. \tag{3.52}$$

The exact solution is

$$u(x,t) = \frac{x}{t\left(1 + 2\sqrt{t}\ e^{\frac{x^2}{4\nu t}}\right)}. \tag{3.53}$$

Substituting (3.4) into (3.50), we get the nonlinear system of equations

$$\sum_{k=1}^{3} b'_{k+2(i-1)}(t)\ B_k(x) = \frac{\nu}{h^2} \sum_{k=1}^{3} b_{k+2(i-1)}(t)\ B''_k(x) - \frac{1}{h}\left(\sum_{k=1}^{3} b_{k+2(i-1)}(t)\ B_k(x)\right)^\delta \left(\sum_{k=1}^{3} b_{k+2(i-1)}(t)\ B'_k(x)\right). \tag{3.54}$$

Similar to the Burgers' equation case, we apply the trapezoidal rule in time, linearize the nonlinear term of equation (3.50) and substitute (3.16) in (3.50):

$$\sum_{k=0}^{3}\left(\left[1+\frac{\Delta t\delta}{2h}\left(\sum_{k=0}^{3}b_{k+2(i-1)}(t_j)\ B_k(z)\right)^{\delta-1}\sum_{k=0}^{3}b_{k+2(i-1)}(t_j)\ B_k'(z)\right]B_k(z)\right.$$
$$\left.-\frac{\Delta t}{2h^2}\nu\ B_k''(z)+\frac{\Delta t}{2h}\left[\sum_{k=0}^{3}b_{k+2(i-1)}(t_j)B_k(z)\right]^{\delta}B_k'(z)\right)b_{k+2(i-1)}(t_{j+1})$$
$$=\sum_{k=0}^{3}\left[B_k(z)+\frac{\Delta t}{2h^2}\nu\ B_k''(z)+\frac{\Delta t}{2h}(\delta-1)\left(\sum_{k=0}^{3}b_{k+2(i-1)}(t_j)\ B_k(z)\right)^{\delta}B_k'(z)\right]b_{k+2(i-1)}(t_j). \tag{3.55}$$

The boundary conditions become

$$\left.\begin{aligned}u(0,t)&=b_1=0,\\ u(1,t)&=b_{2N+1}=0.\end{aligned}\right\} \tag{3.56}$$

The system comprising (2.15), (3.55) and (3.56) can now be solved to obtain $u(x,t)$. The stability and convergence analysis are very similar to the previous case of the classical Burgers' equation and are not repeated.

A 3D plot of the numerical solution and error for $N=50$ are given in Figures 3.7 and 3.8, respectively. The orthogonal collocation of finite elements based on quadratic B-spline basis functions gives good results when compared with the exact solution of the modified Burgers' equation as shown in Figure 3.8.

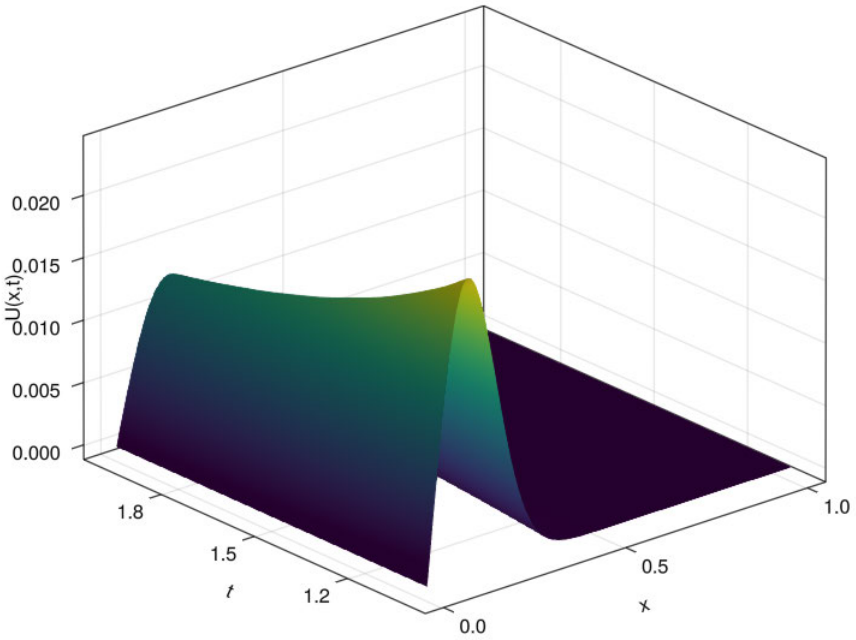

Figure 3.7: 3D plot of approximate solution of the modified Burgers' equation.

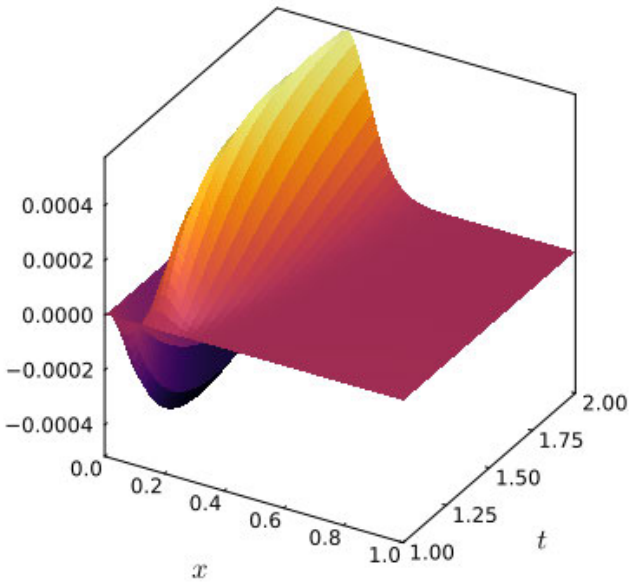

Figure 3.8: 3D plot of error for the modified Burgers' equation.

In Figures 3.9–3.11, time profiles of the solution are shown for various values of ν and N. It is seen that the OCFE method is capable of tracking the steep profiles of the solution for extremely small values of ν and moderate values of N.

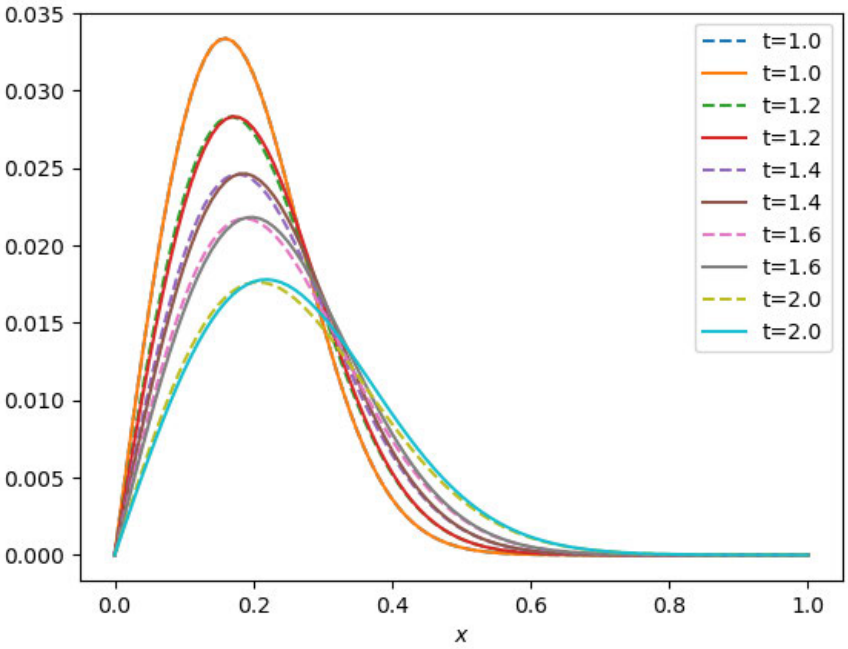

Figure 3.9: Shock propagation $\nu = 0.01$ and $N = 100$.

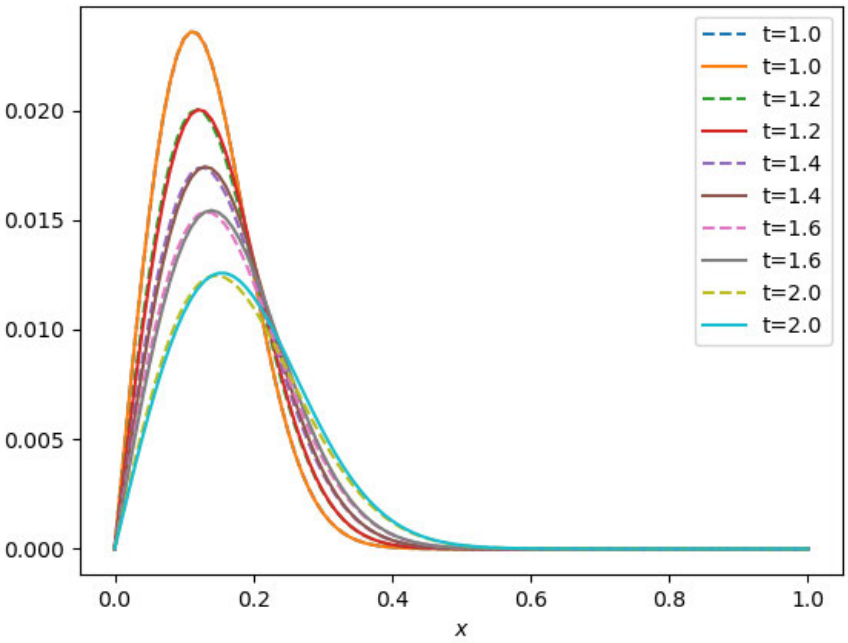

Figure 3.10: Shock propagation $\nu = 0.005$ and $N = 100$.

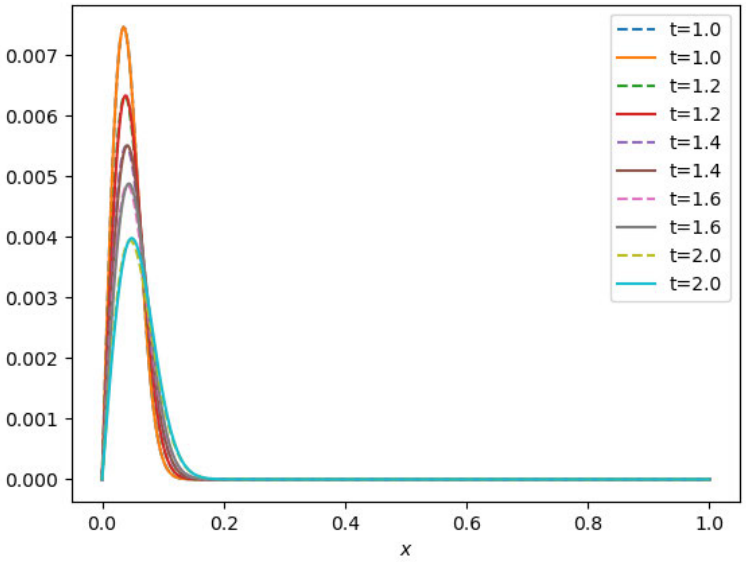

Figure 3.11: Shock propagation $\nu = 0.0005$ and $N = 400$.

3.6 Discussion of Chapter 3

In the previous sections, we applied quadratic B-splines in conjunction with finite elements (OCFE method) to solve second-order PDEs. It is seen that the OCFE method using quasilinearization is optimal in efficiency for solving Burgers' and modified Burgers' equations. In particular, there is no need to impose additional boundary conditions, and the method matches the second-order method used for time integration. Hence, the overall method is second-order in space and time.

We have shown that quadratic B-spline OCFE can effectively solve second order partial differential equations in uniform intervals using the Burgers' equation as a case study. Moreover, we showed that it is faster and gives smaller error than the classical B-spline collocation methods. In the next chapter we shall investigate the performance of quadratic B-spline OCFE to Schrödinger equation on non-uniform intervals.

CHAPTER FOUR

QUADRATIC B-SPLINE OCFE FOR NON-UNIFORM INTERVALS AND APPLICATION TO SCHRÖDINGER EQUATION

In this chapter, we examine the applicability of the quadratic B-spline OCFE to partial differential equations defined on non-uniform intervals. We shall illustrate our method with the Schrödinger equation considering different soliton cases. We also discuss the stability of the quadratic B-spline OCFE for the Schrödinger equation. In the next section, we shall derive the first derivative condition when the given interval is partitioned into N non-uniform subintervals.

4.1 First derivative continuity condition for non-uniform intervals

Suppose the interval $[a, b]$ is divided into N non-uniform intervals $[x_i, x_{i+1}]$ such that given the points

$$a = x_1 < x_2 < x_3 < \cdots < x_i < x_{i+1} < \cdots < x_{n+1} = b, \tag{4.1}$$

the interval length h_i for $[x_i, x_{i+1}]$ are not equal, where $h_i = x_{i+1} - x_i,\ i = 1, 2, \cdots, N$. We transform $x \in [x_i, x_{i+1}]$ to $[0, 1]$ using

$$z = \frac{x - x_i}{h_i}. \tag{4.2}$$

Using the continuity condition in equation (2.13), we have

$$\left(\frac{1}{h_{i+1}}\right) \frac{dY^{i+1}}{dz}\bigg|_{z=0} = \left(\frac{1}{h_i}\right) \frac{dY^{i}}{dz}\bigg|_{z=1}. \tag{4.3}$$

If we combine (4.3) with (2.12), we get a system of linear equations

$$b_{2k-3}^{i+1} = b_{2k-1}^{i}, \tag{4.4}$$

$$b_{2k}^{i+1} = \left(1 + \frac{h_{i+1}}{h_i}\right) b_{2k-1}^{i} - \left(\frac{h_{i+1}}{h_i}\right) b_{2k-2}^{i},\ k = 2, 3, \cdots, N. \tag{4.5}$$

The approximate solution $Y_i(x)$ in an element $[x_i, x_{i+1}]$ is also of the form (2.16).

4.2 Application to the Schrödinger equation

As an application we take an example from [58], the Schrödinger equation [[45], [50]]

$$iu_t + u_{xx} + q|u|^2 u = 0,\ x \in (-\infty, \infty),\ t \in [0, T], \tag{4.6}$$

where $i = \sqrt{-1}$ with the initial condition

$$u(x, 0) = g(x),\ |g(x)| \to 0 \text{ as } |x| \to \infty. \tag{4.7}$$

We truncate the infinite spatial domain into a finite domain $[a, b]$, and impose the boundary conditions

$$u(a, t) = 0,\ u(b, t) = 0. \tag{4.8}$$

It is convenient to rewrite (4.6) as a system of PDEs (without complex quantities) by defining $u(x, t) = v(x, t) + iw(x, t)$ then $iu = -w + iv$.

$$-w_t + v_{xx} + q(v^2 + w^2)v = 0, \tag{4.9}$$

$$v_t + w_{xx} + q(v^2 + w^2)w = 0. \tag{4.10}$$

We apply the trapezoid rule in the time variable and linearize equations (4.9) and (4.10) which gives

$$\begin{aligned} &\left[\frac{-\Delta t}{2}\ v''_{j+1} - \frac{q\Delta t}{2}\ (3v_j^2 + w_j^2)\ v_{j+1}\right] + [1 - q\Delta t\ w_j v_j]\, w_{j+1} \\ &= w_j + \frac{\Delta t}{2}\left[v''_j - q(v_j^2 + w_j^2)v_j\right] \end{aligned} \tag{4.11}$$

and

$$\begin{aligned}&\left[1+q\Delta t\ v_jw_j\right]v_{j+1}+\left[\frac{\Delta t}{2}\ w''_{j+1}+\frac{q\Delta t}{2}\ (v_j^2+3w_j^2)\ w_{j+1}\right]\\&=v_j+\frac{\Delta t}{2}\left[-w''_j+q(w_j^2+v_j^2)w_j\right],\end{aligned}\tag{4.12}$$

respectively. The approximate solution in the i^{th} interval at time t_j is assumed to be of the form

$$U^i(x,t_j)=\sum_{k=1}^{3}c_{k+2(i-1)}(t_j)\ B_k(x),\tag{4.13}$$

where

$$c_{k+2(i-1)}(t_j)=a_{k+2(i-1)}(t_j)+ib_{k+2(i-1)}(t_j),\quad i=\sqrt{-1}.\tag{4.14}$$

Therefore at the collocation point z, $U(z,t_j)=\sum_{k=1}^{3}c_{k+2(i-1)}(t_j)\ B_k(z)$. This implies that

$$v_j=v(z,t_j)=\sum_{k=1}^{3}a_{k+2(i-1)}(t_j)\ B_k(z),\tag{4.15}$$

$$w_j=w(z,t_j)=\sum_{k=1}^{3}b_{k+2(i-1)}(t_j)\ B_k(z).\tag{4.16}$$

We substitute equations (4.5), (4.15) and (4.16) into equations (4.11) and (4.12). Therefore, equations (4.11) and (4.12) can be expressed as

$$\begin{aligned}&\frac{\Delta t}{2}\left[-\frac{1}{h_i^2}\sum_{k=1}^{3}a_{k+2(i-1)}(t_{j+1})\ B''_k(z)-q\left(\left[3\sum_{k=1}^{3}a_{k+2(i-1)}(t_j)\ B_k(z)\right]^2+\left[\sum_{k=1}^{3}b_{k+2(i-1)}(t_j)\ B_k(z)\right]^2\right)\right.\\&\left.\times\sum_{k=1}^{3}a_{k+2(i-1)}(t_{j+1})\ B_k(z)\right]\\&+\sum_{k=1}^{3}\left[1-q\Delta t\left(\sum_{k=1}^{3}a_{k+2(i-1)}(t_j)\ B_k(z)\right)\left(\sum_{k=1}^{3}b_{k+2(i-1)}(t_j)\ B_k(z)\right)\right]b_{k+2(i-1)}(t_{j+1})\ B_k(z)\\&=\sum_{k=1}^{3}b_{k+2(i-1)}(t_j)\ B_k(z)+\frac{\Delta t}{2}\left[\frac{1}{h_i^2}\sum_{k=1}^{3}a_{k+2(i-1)}(t_j)\ B''_k(z)\right.\\&\left.-q\left(\left[\sum_{k=1}^{3}a_{k+2(i-1)}(t_j)\ B_k(z)\right]^2+\left[\sum_{k=1}^{3}b_{k+2(i-1)}(t_j)\ B_k(z)\right]^2\right)\sum_{k=1}^{3}a_{k+2(i-1)}(t_j)B_k(z)\right]\end{aligned}\tag{4.17}$$

and

$$\sum_{k=1}^{3}\left[1+q\Delta t\left(\sum_{k=1}^{3}a_{k+2(i-1)}(t_j)\ B_k(z)\right)\left(\sum_{k=1}^{3}b_{k+2(i-1)}(t_j)\ B_k(z)\right)\right]a_{k+2(i-1)}(t_{j+1})\ B_k(z)$$
$$+\frac{\Delta t}{2}\left[\frac{1}{h_i^2}\sum_{k=1}^{3}b_{k+2(i-1)}(t_{j+1})\ B_k''(z)+q\left(\left[\sum_{k=1}^{3}a_{k+2(i-1)}(t_j)\ B_k(z)\right]^2+3\left[\sum_{k=1}^{3}b_{k+2(i-1)}(t_j)\ B_k(z)\right]^2\right)\right.$$
$$\left.\sum_{k=1}^{3}b_{k+2(i-1)}(t_{j+1})\ B_k(z)\right]=\sum_{k=1}^{3}a_{k+2(i-1)}(t_j)\ B_k(z)+\frac{\Delta t}{2}\left[-\frac{1}{h_i^2}\sum_{k=1}^{3}b_{k+2(i-1)}(t_j)\ B_k''(z)\right.$$
$$\left.+q\left(\left[\sum_{k=1}^{3}a_{k+2(i-1)}(t_j)\ B_k(z)\right]^2+\left[\sum_{k=1}^{3}b_{k+2(i-1)}(t_j)\ B_k(z)\right]^2\right)\sum_{k=1}^{3}b_{k+2(i-1)}(t_j)B_k(z)\right], \tag{4.18}$$

$i=1,2,\cdots,N.$, respectively. The boundary conditions yield

$$a_1=a_{2N+1},\ b_1=b_{2N+1}=0. \tag{4.19}$$

Equations (4.17), (4.18), (4.5) and the boundary conditions (4.19) form a linear system of equations $A\mathbf{p}=\mathbf{q}$ where

$$A=\begin{pmatrix}A_{1,1} & A_{1,2}\\ \\ A_{2,1} & A_{2,2}\end{pmatrix}, \tag{4.20}$$

$\mathbf{p}=[\mathbf{a},\mathbf{b}]^T$, $\mathbf{a}=[a_1,a_2,\ldots,a_{2N+1}]$, $\mathbf{b}=[b_1,b_2,\ldots,b_{2N+1}]$, $\mathbf{q}=[\mathbf{q}_1,\mathbf{q}_2]^T$,

$$q_{1,i}=\begin{cases}f_1(x_i+0.5h) & i=1,\ldots,N,\\ \\ 0 & i=N+1,\ldots,2N+1,\end{cases} \tag{4.21}$$

$$q_{2,i}=\begin{cases}f_2(x_i+0.5h) & i=1,\ldots,N,\\ \\ 0 & i=N+1,\ldots,2N+1,\end{cases} \tag{4.22}$$

$f_1(z)$ is the right hand side of (4.17), $f_2(z)$ is the right hand side of (4.18), $A_{i,1}$ and $A_{i,2},\ (i=1,2)$ are the coefficient matrices of the unknowns a_j and b_j, respectively for $j=1,2,\ldots,2N+1$.

Using the initial condition at the collocation point $z=0.5$, the initial coefficient matrices are such that $A_{1,1}=A_{2,2}$, $A_{1,2}=A_{2,1}=\mathbf{0}$, where $A_{1,1}$ has the form

$$A_{1,1} = \begin{bmatrix} B_1 & B_2 & B_3 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & B_1 & B_2 & B_3 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & B_1 & B_2 & B_3 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & B_1 & B_2 & B_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & B_2 & B_3 \\ 0 & \frac{h_2}{h_1} & -\frac{h_2}{h_1}-1 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \frac{h_3}{h_2} & -\frac{h_3}{h_2}-1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{h_4}{h_3} & -\frac{h_4}{h_3}-1 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \quad (4.23)$$

$$q_{1,i} = \begin{cases} g(x_i + 0.5h) & i = 1, \ldots, N, \\ 0 & i = N+1, \ldots, 2N+1, \end{cases} \quad (4.24)$$

$$B_j = B_j(0.5) \quad j = 1, 2, 3, \quad (4.25)$$

and

$$q_{2,i} = 0, \quad i = 1, \ldots, 2N+1 \quad (4.26)$$

are used to get the initial solution.

At subsequent times, we compute sub-matrices $A_{1,1}$, $A_{1,2}$, $A_{2,1}$ and $A_{2,2}$ from equations (4.17) and (4.18) to get the matrix A and finally obtain $\mathbf{p}$. The numerical solution to the Schrödinger equation can now be obtained by substituting $\mathbf{p}$ into (4.14) and then into (4.13).

The invariants at various times are given in [58] as

$$L = \int_{-\infty}^{\infty} |u|^2 dx = 2, \tag{4.27}$$

$$H = \int_{-\infty}^{\infty} \left(|u_x|^2 dx - \frac{1}{2} q|u|^4 \right) dx = \frac{2}{3}(1-q), \tag{4.28}$$

$$L' = \int_{-20}^{20} |U|^2 dx, \quad \text{and} \quad H' = \int_{-20}^{20} \left(|U_x|^2 dx - \frac{1}{2} q|U|^4 \right) dx, \tag{4.29}$$

where $u = u(x,t)$ is the exact solution and $U = U(x,t)$ is the approximate solution.

4.3 Stability Analysis

In this section, we apply the von Neumman stability analysis to the Schrödinger equation for equal spacing h. Let the non-linear term $q|u|^2$ be bounded by a local constant K [44]. Then the Schrödinger equation becomes

$$iu_t = -u_{xx} - Ku. \tag{4.30}$$

Applying Crank-Nicolson method to equation (4.30), we get

$$\left(1 - \frac{iK\Delta t}{2}\right) u^{j+1} - \frac{i\Delta t}{2h^2} u_{zz}^{j+1} = \left(1 + \frac{iK\Delta t}{2}\right) u^j + \frac{i\Delta t}{2h^2} u_{zz}^j. \tag{4.31}$$

Since $u^j = \sum_{k=1}^{3} c_{k+2(m-1)}^j \; B_k(z)$ in the m_{th} interval, equation (4.31) becomes

$$\begin{aligned} &\sum_{k=1}^{3} c_{k+2(m-1)}^{j+1} \; \left[\left(1 - \frac{iK\Delta t}{2}\right) B_k(z) - \frac{i\Delta t}{2h^2} B_k''(z) \right] = \\ &\sum_{k=1}^{3} c_{k+2(m-1)}^{j} \; \left[\left(1 + \frac{iK\Delta t}{2}\right) B_k(z) + \frac{i\Delta t}{2h^2} B_k''(z) \right], \end{aligned} \tag{4.32}$$

where $\alpha = \frac{K\Delta t}{2}$ and $\beta = \frac{\Delta t}{2h^2}$.

$$\begin{aligned} &\sum_{k=1}^{3} c_{k+2(m-1)}^{j+1} \; \left[(1 - i\alpha) B_k(z) - i\beta B_k''(z) \right] = \\ &\sum_{k=1}^{3} c_{k+2(m-1)}^{j} \; \left[(1 + i\alpha) B_k(z) + i\beta B_k''(z) \right]. \end{aligned} \tag{4.33}$$

At $z = \frac{1}{2}$ (collocation point),

$$c_{2m-1}^{j+1}\sigma_1 + c_{2m}^{j+1}\sigma_2 + c_{2m+1}^{j+1}\sigma_1 = c_{2m-1}^{j}\bar{\sigma_1} + c_{2m}^{j}\bar{\sigma_2} + c_{2m+1}^{j}\bar{\sigma_1}, \tag{4.34}$$

where

$$\sigma_1 = \frac{1}{4} - i\frac{\alpha}{4} - 2i\beta, \quad \sigma_2 = \frac{1}{2} - i\frac{\alpha}{2} + 4i\beta, \tag{4.35}$$

and $c_m^j = \lambda^j \mathrm{e}^{imhk}$ for the k_{th} mode. Equation (4.34) is now expressed as

$$\begin{aligned} &\lambda^{j+1}\left[e^{i(2m-1)hk}\sigma_1 + e^{i2mhk}\sigma_2 + e^{i(2m+1)hk}\sigma_1\right] \\ &= \lambda^{j}\left[e^{i(2m-1)hk}\bar{\sigma_1} + e^{i2mhk}\bar{\sigma_2} + e^{i(2m+1)hk}\bar{\sigma_1}\right], \end{aligned} \tag{4.36}$$

$$\lambda^{j+1}[\sigma_1\cos(hk) + \sigma_2] = \lambda^{j}[2\bar{\sigma_1}\cos(hk) + \bar{\sigma_2}], \tag{4.37}$$

$\Longrightarrow$

$$\lambda = \frac{2\bar{\sigma_1}\cos(hk) + \bar{\sigma_2}}{2\sigma_1\cos(hk) + \sigma_2} = \frac{\cos(hk) + \frac{1}{2} + i\left[\left(\frac{\alpha}{2} + 4\beta\right)\cos(hk) + \left(\frac{\alpha}{2} - 4\beta\right)\right]}{\cos(hk) + \frac{1}{2} - i\left[\left(\frac{\alpha}{2} + 4\beta\right)\cos(hk) + \left(\frac{\alpha}{2} - 4\beta\right)\right]} = \frac{M}{\bar{M}}, \tag{4.38}$$

where $\bar{M}$ is the conjugate of M.

The fact that $|M| = |\bar{M}|$ means that $|\lambda| = 1$. Therefore the orthogonal quadratic OCFE method using Gauss' points for the Schrödinger equation is unconditionally stable.

4.4 Numerical examples

We consider a bound state of n solitons of the Schrödinger equation

$$iu_t + u_{xx} + q|u|^2u = 0, \ x \in (-\infty, \infty), \ t \in [0, T], \tag{4.39}$$

where $q = 2n^2$ and n is the number of solitons, with the initial condition $u(x, 0)$ on the interval [-20, 20] and $t \in [0, 2.5]$ [58]. We consider the interval $[a, b]$ with a

non-uniform grid given by

$$x_0 = 0,\ x_j = x_{-j} = 20\left(\frac{2j}{N}\right)^k,\ k \geq 1,\ j = 1, 2, \cdots, \frac{N}{2}, \tag{4.40}$$

where $k \in \mathbb{R}$ and N is the number of intervals. The results for the approximate solutions of the Schrödinger equation on uniform and non-uniform grids for the one soliton ($q = 2$), two solitons ($q = 8$) and three solitons ($q = 18$) cases are presented below:

Example 4.1: We take $q = 2$ and $N = Nt = 100$ for the one soliton case. The exact solution is

$$u(x, t) = e^{it}\text{sech}(x). \tag{4.41}$$

The numerical results are presented graphically in the Figures 4.1 to 4.4. The figure in the upper part of the right column shows the 3D plot of the approximate solution and the one in the lower right column displays error for example 4.1 when $k = 1$. In the left column, the upper and lower figures depict the approximate solution and the error when $k = 1.2$. We can deduce that the error is smaller when $k = 1.2$ than when $k = 1$. Hence the infinity norm of errors for case $k = 1.2$ is smaller than that of case $k = 1$.

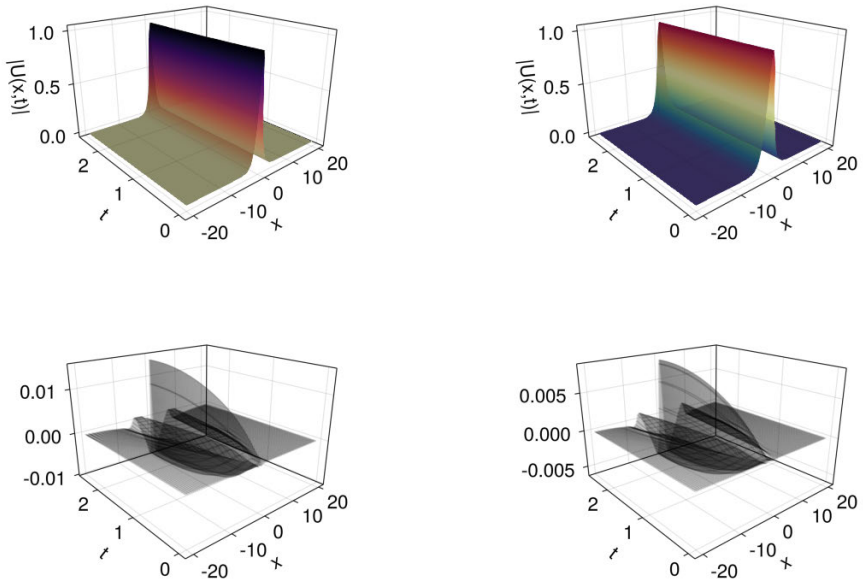

Figure 4.1: Approximate solution and the error when $k = 1$ (Left column) and $k = 1.2$ (Right column).

In Figures 4.2 and 4.3, the infinity norm of the errors of the real and imaginary parts of the approximate solution is oscillatory with respect to time but decreases as k increases.

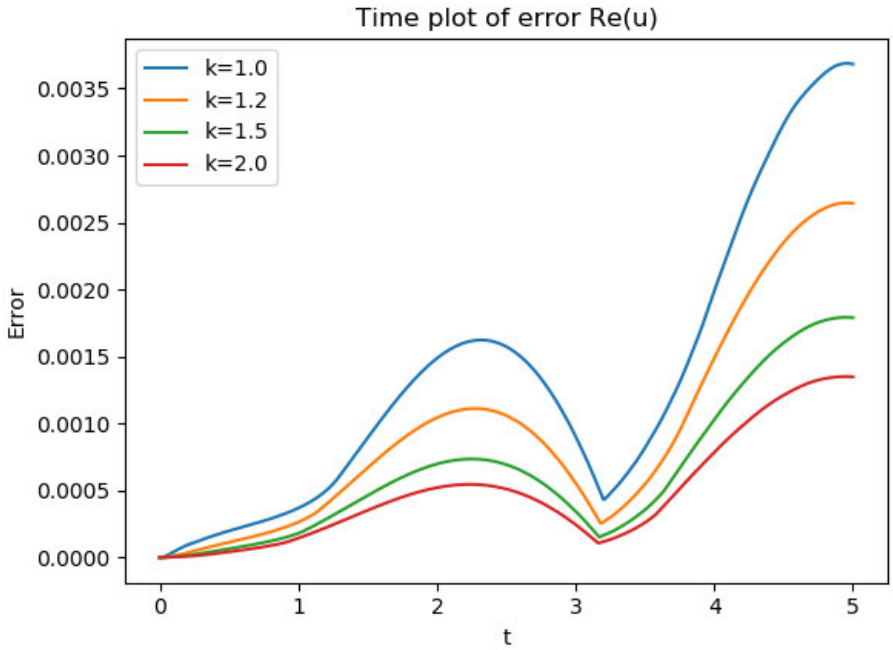

Figure 4.2: L_∞ error of $Re(U)$ against time (t) when $q = 2$ and $N = 500$ for various values of k.

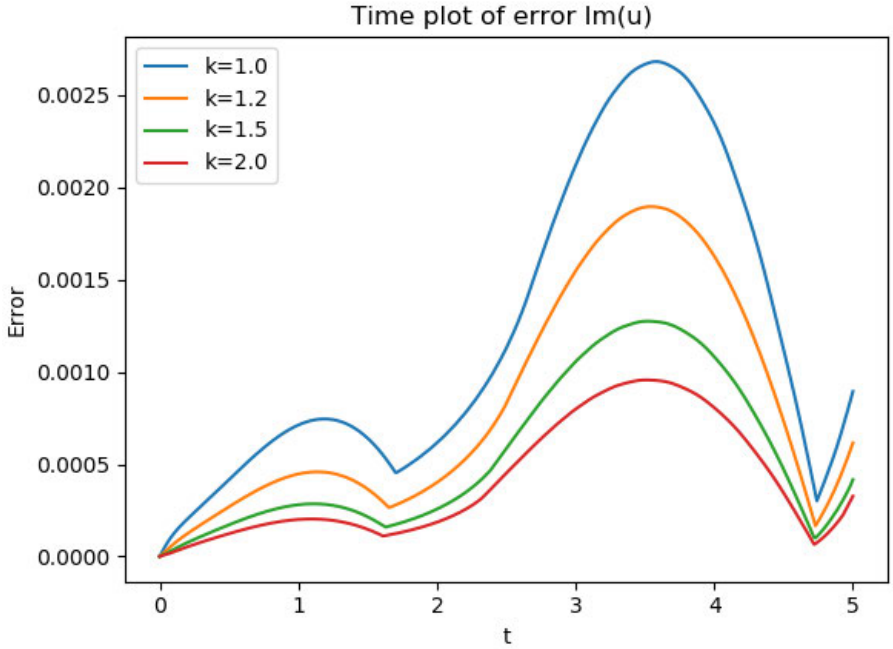

Figure 4.3: L_∞ error of $Im(U)$ against time (t) when $q = 2$ and $N = 500$ for various values of k.

Also Figure 4.4 shows that the invariants L' is approximately the same as the

exact value of $L = 2$ but as k increases, the absolute error for L' does not exceed 3×10^{-4}. Similarly Figure 4.5 shows that the invariant H' almost the same for the values of k considered except for $k = 1.2$. The absolute error for H' is less than 3×10^{-4}.

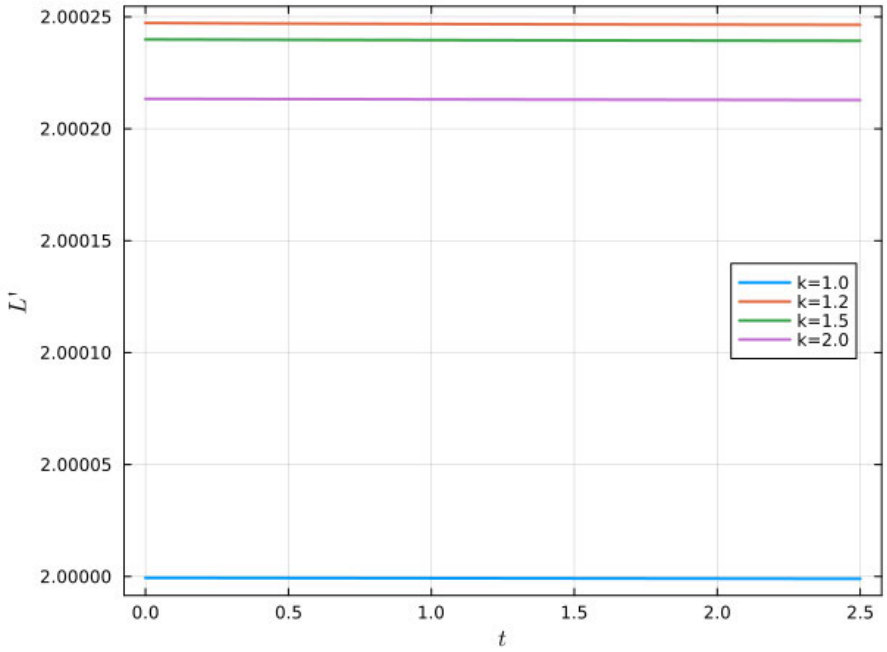

Figure 4.4: Invariant L' and H' for $N = 500$ $q = 2$.

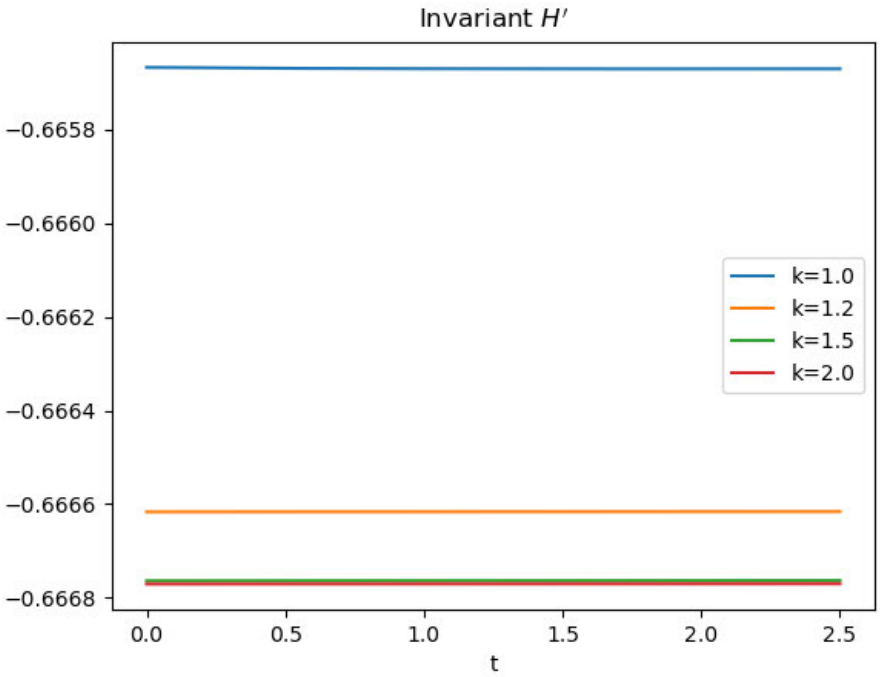

Figure 4.5: Invariant L' and H' for $N = 500,\ q = 2$.

Figures 4.6 and 4.7 further confirm that as k increases, the infinity norm of errors of the real and imaginary parts solutions decreases. Hence the results are closer to the exact solution as k increases.

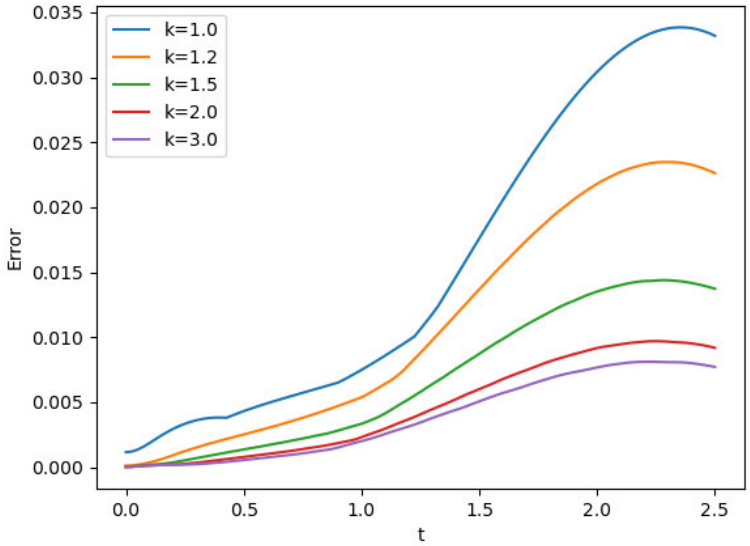

Figure 4.6: Plot of L_∞ norm of error of $Re(U)$ and $Im(U)$ against time when $N = Nt = 100$ $q = 2$.

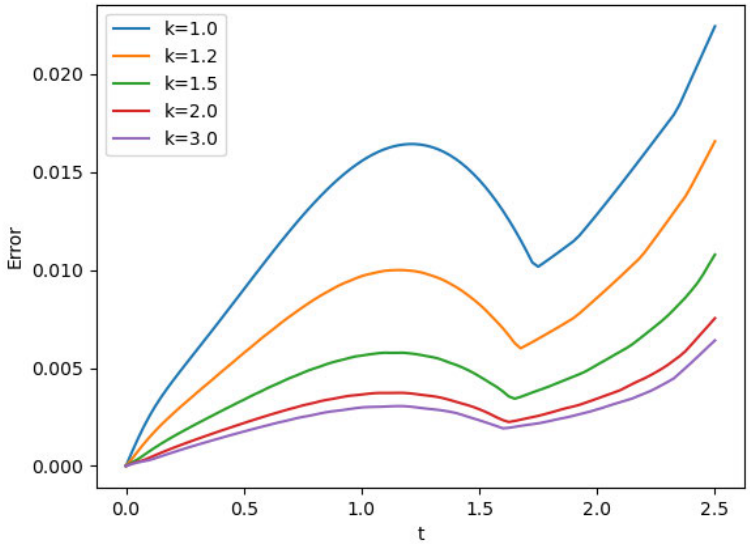

Figure 4.7: Plot of L_∞ norm of error of $Re(U)$ and $Im(U)$ against time when $N = Nt = 100$ $q = 2$.

In this case, $q = 2$ the invariants $L = 2$ and $H = -\frac{2}{3}$. The approximate results are shown in Table 4.1 . In Table 4.1, the invariant $L' \approx L$ such that the absolute errors when $k = 1,\ 1.2$ and 2 are less than or equal to 1.1×10^{-6}, 2.5×10^{-4} and 2.2×10^{-4}, respectively.

Table 4.1: Invariant L' for $q = 2$, $N = 500$.

t	$k = 1$	$k = 1.2$	$k = 2$
0.5	1.9999993140704686	2.000246967311435	2.0002132431685116
1.0	1.9999992255589547	2.0002467675324036	2.0002131357895463
1.2	1.9999991906558545	2.0002467024427304	2.0002130884933638
1.5	1.9999991388513283	2.0002466164730652	2.0002130315117610
2.0	1.9999990539225080	2.0002465013137947	2.0002129394045010
2.2	1.9999990204151104	2.0002464639068678	2.0002129044476304
2.5	1.9999989706002652	2.0002464173233556	2.0002128557823440

Figure 4.8 illustrates the numerical convergence of the method for $k = 1.2$. The method is approximately of order two in space. The theoretical convergence analysis can be obtained along the same lines as [52].

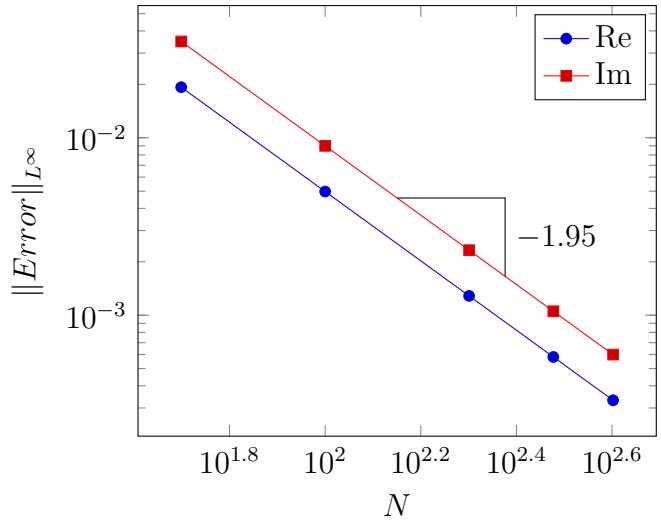

Figure 4.8: Convergence plot for example 4.1 when $k = 1.2$.

Example 4.2: We take $q = 8$ in equation (4.6) for the case of two solitons. The exact solution is given by

$$u(x,t) = \mathrm{e}^{it}\mathrm{sech}(x)\frac{(1 + 0.75\,\mathrm{sech}^2(x)(\mathrm{e}^{8\,it} - 1))}{1 - 0.75\,\mathrm{sech}^4(x)\sin^2(4\,t)}. \tag{4.42}$$

The results are presented graphically in Figure 4.9 for $k = 1.2$ and $N = Nt = 500$. The solutions for the formation of two solitons are depicted in the figure. We observe that the absolute error for $|U(x,t)|$ is less than 3×10^{-2}. The density plot in Figure 4.9 shows the distribution of the modulus $|U(x,t)|$ in the region under consideration.

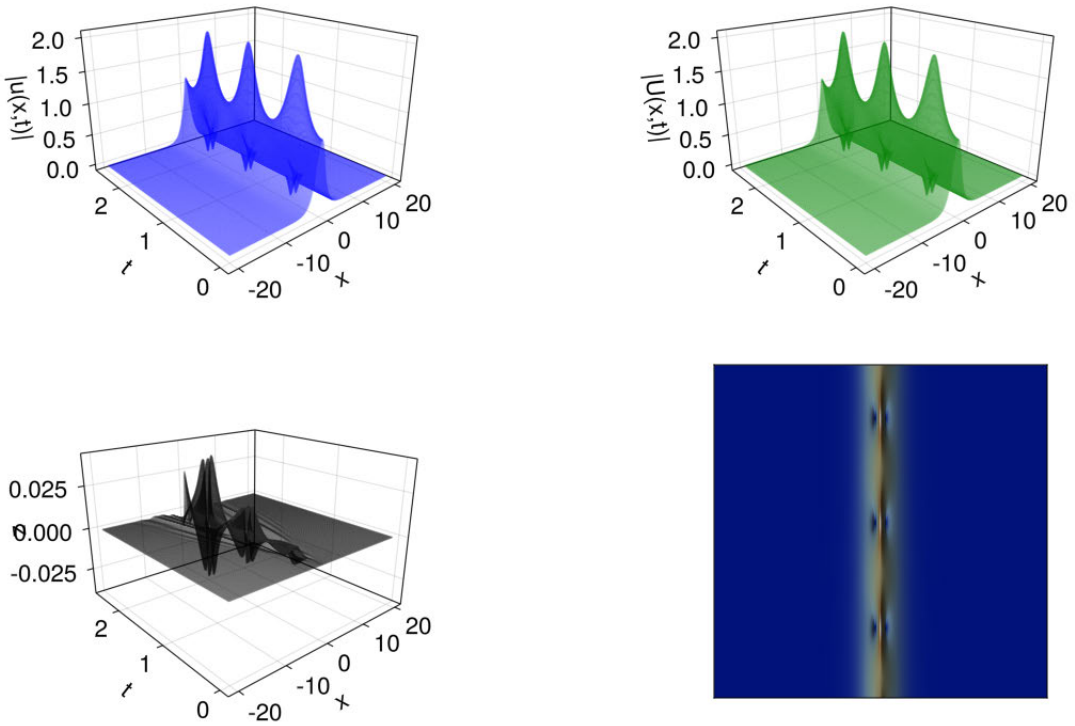

Figure 4.9: Top left: Exact solution . Top right: Approximate solution. Bottom left: Error. Bottom right: Density plot

In Figure 4.10 and 4.11, the L_∞ error of the real and imaginary part solutions decreases as k increases.

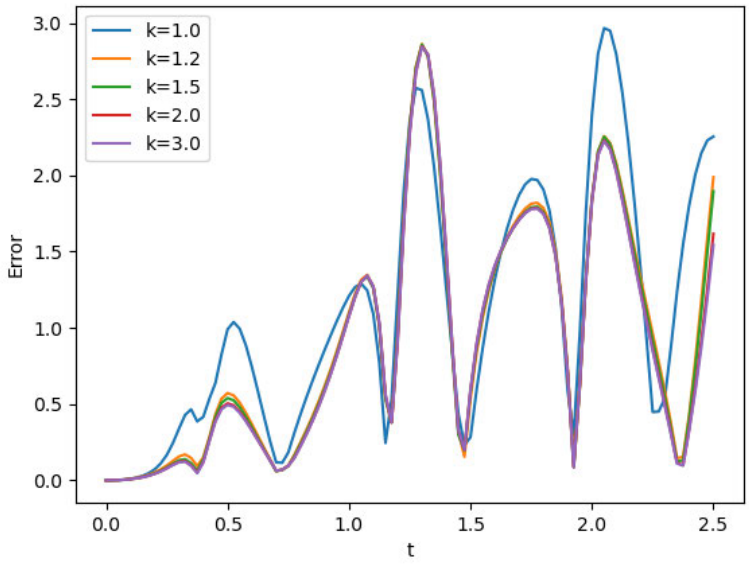

Figure 4.10: Plot of L_∞ norm of error of $Re(U)$ against time when $N = Nt = 100$.

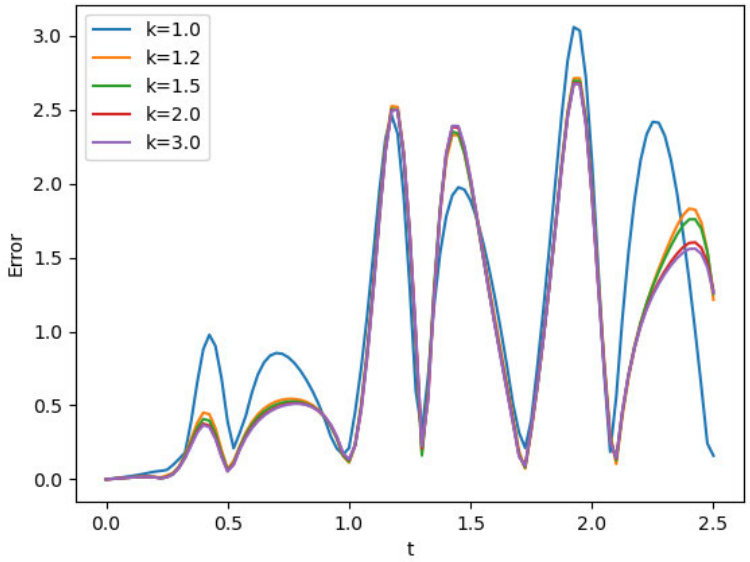

Figure 4.11: Plot of L_∞ norm of error of $Im(U)$ against time when $N = Nt = 100$.

Moreover the exact values of the invariants are $L = 2$ and $H = -\frac{14}{3}$. The approximate results for L' and H' are shown in the Tables 4.2 and 4.3, respectively. The absolute errors for L' in Table 4.2 when $k = 1$, 1.2 and 2 are less than or equal to 4×10^{-3}, 3×10^{-3} and 3×10^{-3} respectively. This shows that as k increases the absolute error for L' decreases.

Furthermore, in Table 4.3 the absolute errors for H' when $k = 1$, 1.2 and 2 are less than or equal to 1.7×10^{-1}, 4×10^{-2} and 4×10^{-2} respectively. Hence as k increases the absolute error for H' decreases. Therefore the approximate results are consistent with the exact values of the invariants L and H.

Table 4.2: Invariant L' for $q = 8$, $N = 500$.

t	$k = 1$	$k = 1.2$	$k = 2$
0.5	1.9997860736288362	2.0018511811360410	2.0005647365874095
1.0	1.9991639559418688	2.0001139413317444	1.9996056704823670
1.2	1.9999478538782265	2.0039065138798340	2.0012834291274677
1.5	1.9979781887606685	1.9982537645416570	1.9981634131928380
2.0	1.9988043891305036	2.0025937977783066	2.0000636330174470
2.2	1.9970867529391771	1.9976414358241767	1.9973464431795530
2.5	1.9969780960005120	1.9973784496338522	1.9971894528931016

Table 4.3: Invariant H' for $q = 8$, $N = 500$.

t	$k = 1$	$k = 1.2$	$k = 2$
0.5	-4.591078095033748	-4.679429403766860	-4.675583794247313
1.0	-4.638954540953396	-4.662140176598175	-4.660707149685754
1.2	-4.437170115041884	-4.707311934773099	-4.698417405662838
1.5	-4.640863387307565	-4.643435607752973	-4.642913738260561
2.0	-4.438513861316172	-4.691039939310799	-4.682415168541428
2.2	-4.624185813998357	-4.634236501508632	-4.633074582893552
2.5	-4.626840469901180	-4.632320126916547	-4.631457470965063

Example 4.3: We take $q = 18$ for the three solitons case. Here, the exact solution is unknown. The results are shown in Figure 4.12 when $N = Nt = 100$ for different values of k. Figure 4.12 shows an improvement in the solution when the interval is less uniform ($k = 1.2$) compared to that of the uniform case ($k = 1$). The

density plot in Figure 4.12 depict the distribution of the modulus in the region. The concentration at the middle for three solitons (case $q = 18$) is higher than that of two solitons ($q = 8$) in Figure 4.9.

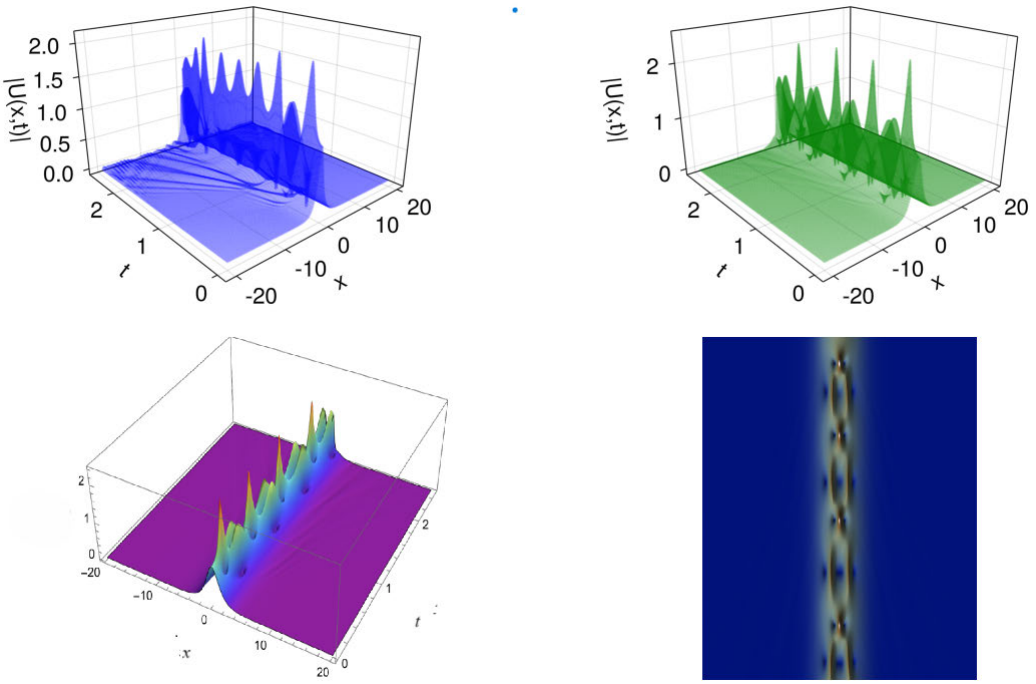

Figure 4.12: Top left: $k = 1$. Top right: k=1.2. Bottom left: Surface plot. Bottom right: Density plot

Similarly in Figure 4.13 and 4.14, L_∞ error of the real and imaginary part solutions decreases as k increases.

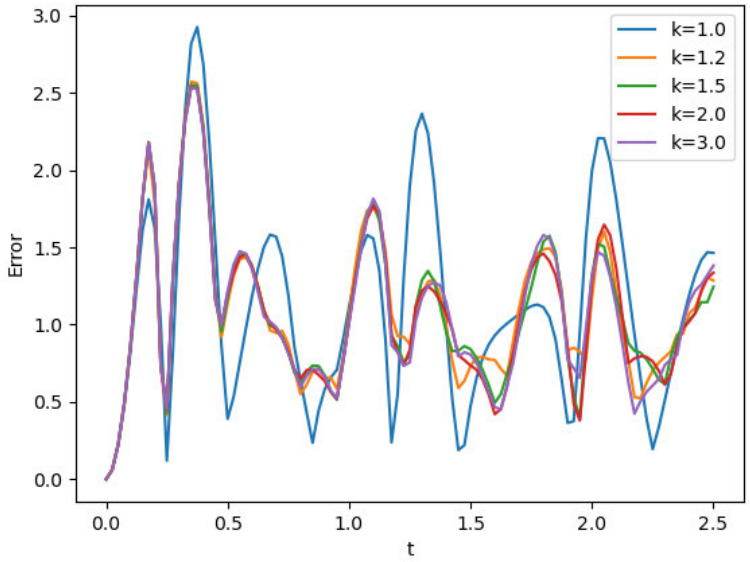

Figure 4.13: Plot of L_∞ norm of error of $Re(U)$ against time when $N = Nt = 100$.

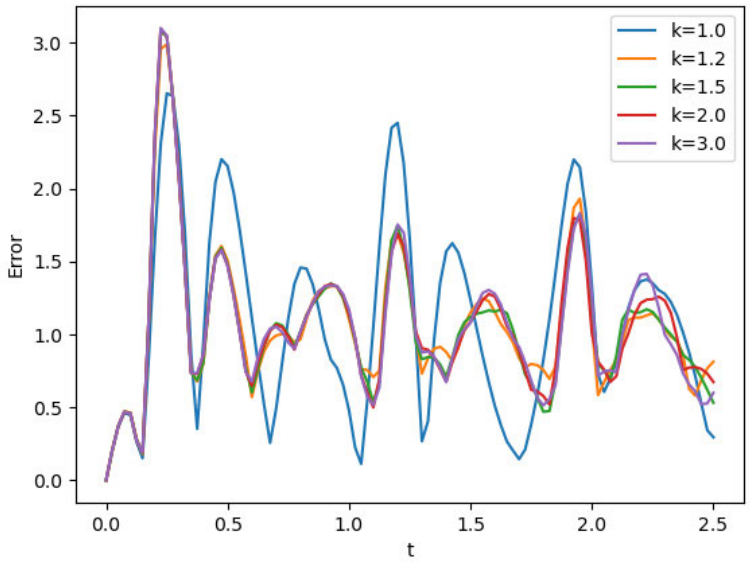

Figure 4.14: Plot of L_∞ norm of error of $Im(U)$ against time when $N = Nt = 100$.

The approximate results for invariants L' and H' are shown in Tables 4.4 and 4.5, respectively when $q = 18$. In Table 4.4, the values of L' decreases gradually as k increases and generally $L' \approx 1.9$. However in Table 4.5, H' increases gradually as k increases but in general $H' \approx -9$.

Table 4.4: invariant L' for $q = 18,\ N = 500$.

t	$k = 1$	$k = 1.2$	$k = 2$
0.5	1.9716347189462933	1.9701328044896482	1.9700315096238600
1.0	1.9536882683277290	1.9521142630388197	1.9520900110377186
1.2	1.9496199006870800	1.9482745207398677	1.9479056524406426
1.5	1.9329236584048455	1.9314952912676382	1.9311104479246304
2.0	1.9302098765860340	1.9288105118104828	1.9278253869036210
2.2	1.9157331912116036	1.9140079543809243	1.9129958271366290
2.5	1.9130255707307150	1.9102995775787281	1.9093871035211032

Table 4.5: invariant H' for $q = 18,\ N = 500$.

t	$k = 1$	$k = 1.2$	$k = 2$
0.5	-10.20335493227373	-10.20306913753222	-10.21290813500802
1.0	-9.466406734489020	-9.533985328334023	-9.596214446558621
1.2	-9.589608183446568	-9.545004918903630	-9.531327742520370
1.5	-9.063897102819032	-8.991606269061817	-8.970671762172701
2.0	-8.988052636484749	-8.955082573439789	-8.921529953368218
2.2	-8.551664812991596	-8.461102002161958	-8.424026080706097
2.5	-8.459921555578845	-8.375491781024992	-8.340442854468023

4.5 Discussion of Chapter 4

We have shown that quadratic B-spline OCFE can effectively solve a system of second order partial differential equations arising from the Schrödinger equation in a non-uniform interval. We also found that it's accuracy increases as the non-uniform parameter increases for the soliton cases considered. The results for the examples in this chapter agree favourably with the exact solutions and previous results in the literature. The next chapter deals with the application of quadratic B-spline OCFE to fractional differential equations.

CHAPTER FIVE

APPLICATION OF OCFE TO FRACTIONAL DIFFERENTIAL EQUATIONS

We demonstrate the capability of the quadratic B-spline OCFE to handle fractional differential equations using the Caputo fractional derivative. We considered it's application to ordinary and partial fractional differential equations. We discuss it's stability and carry out convergence analysis of the method. Our numerical simulations showed accuracy of order $2-\alpha$, $1<\alpha<2$, where α is the order of the fractional derivative. Our results compared favourably with previous ones in the literature.

5.1 Solution of fractional ODE

Let a fractional ordinary differential equation

$$D_x^{\alpha}u(x)=\frac{d^{\alpha}u(x)}{dx^{\alpha}}=f(x),\quad n-1<\alpha<n, \tag{5.1}$$

with boundary conditions

$$u(a)=\delta_1,\ u(b)=\delta_2, \tag{5.2}$$

be given. We shall make use of the Caputo fractional derivative together with the quadratic B-spline basis functions to solve equation (5.1). The Caputo derivative [59] is defined as

$$D_x^{\alpha}u(x)=\frac{1}{\Gamma(n-\alpha)}\int_a^x\frac{d^n u(\xi)}{d\xi^n}(x-\xi)^{n-1-\alpha}d\xi,\quad n-1<\alpha<n, \tag{5.3}$$

where $\alpha \in \mathbb{R}$ and $n \in \mathbb{N}$. Due to the fact that quadratic B-spline basis functions are of degree 2, therefore n cannot be greater than 2. Thus we consider the case of $n = 2$ only. Equation (5.3) is now

$$D_x^\alpha u(x) = \frac{1}{\Gamma(2-\alpha)} \int_a^x (x-\xi)^{1-\alpha} u''(\xi)\, d\xi, \quad 1 < \alpha < 2. \tag{5.4}$$

Let $I_j = [x_j, x_{j+1}] \subset [a, b]$ be an interval. Suppose the exact and approximate solutions of equation (5.1) are $u(x)$ and $u_c(x)$ respectively. In the interval I_j, we assumed that

$$u(x) \approx u_c^j(x) = \sum_{k=1}^{3} b_{k+2(j-1)} B_k(x), \tag{5.5}$$

defined in equation (2.16). Let x be in the j^{th} interval. Therefore equation (5.4) can be written as

$$\begin{aligned} D_x^\alpha u(x) = \frac{1}{\Gamma(2-\alpha)} \sum_{l=1}^{j-1} \int_{x_l}^{x_{l+1}} (x-\xi)^{1-\alpha} \sum_{k=1}^{3} b_{k+2(l-1)} B_k''(\xi)\, d\xi \\ + \frac{1}{\Gamma(2-\alpha)} \int_{x_j}^{x} (x-\xi)^{1-\alpha} \sum_{k=1}^{3} b_{k+2(j-1)} B_k''(\xi)\, d\xi, \end{aligned} \tag{5.6}$$

where $x_j = a+(j-1)h$, $x_{j+\frac{h}{2}} = x_j + \frac{h}{2}$, $h = x_{j+1} - x_j$. Because of the transformation $z = \frac{x-x_j}{h}$, it follows that

$$\left.\begin{aligned} B_1''(\xi) &= \tfrac{2}{h^2}, \\ B_2''(\xi) &= -\tfrac{4}{h^2}, \\ B_3''(\xi) &= \tfrac{2}{h^2}. \end{aligned}\right\} \tag{5.7}$$

Thus at the collocation point $x_j + \frac{h}{2}$ (5.6) becomes

$$\begin{aligned} D_x^\alpha u\left(x_j + \frac{h}{2}\right) = \frac{1}{\Gamma(2-\alpha)} \sum_{l=1}^{j-1} \int_{x_l}^{x_{l+1}} (x_j + \frac{h}{2} - \xi)^{1-\alpha} \sum_{k=1}^{3} b_{k+2(l-1)} B_k''(\xi)\, d\xi \\ + \frac{1}{\Gamma(2-\alpha)} \int_{x_j}^{x} (x_j + \frac{h}{2} - \xi)^{1-\alpha} \sum_{k=1}^{3} b_{k+2(j-1)} B_k''(\xi)\, d\xi. \end{aligned} \tag{5.8}$$

Therefore

$$
\begin{aligned}
D_x^\alpha u\left(x_j+\frac{h}{2}\right) = \frac{2}{h^2\Gamma(2-\alpha)}\Bigg[&(b_1-2b_2+b_3)\int_{x_1}^{x_2}\left(x_j+\frac{h}{2}-\xi\right)^{1-\alpha}\,d\xi\\
&+(b_3-2b_4+b_5)\int_{x_2}^{x_3}\left(x_j+\frac{h}{2}-\xi\right)^{1-\alpha}\,d\xi+\cdots\\
&+(b_{2j-3}-2b_{2j-2}+b_{2j-1})\int_{x_{j-1}}^{x_j}\left(x_j+\frac{h}{2}-\xi\right)^{1-\alpha}\,d\xi\\
&+(b_{2j-1}-2b_{2j}+b_{2j+1})\int_{x_j}^{x_j+\frac{h}{2}}\left(x_j+\frac{h}{2}-\xi\right)^{1-\alpha}\,d\xi\Bigg].
\end{aligned}
\tag{5.9}
$$

$$
\begin{aligned}
D_x^\alpha u\left(x_j+\frac{h}{2}\right) = \frac{2}{h^2\Gamma(2-\alpha)}\Bigg[&\sum_{l=1}^{j-1}(b_{2l-1}-2b_{2l}+b_{2l+1})\int_{x_l}^{x_{l+1}}\left(x_j+\frac{h}{2}-\xi\right)^{1-\alpha}d\xi\\
&+(b_{2j-1}-2b_{2j}+b_{2j+1})\int_{x_j}^{x_j+\frac{h}{2}}\left(x_j+\frac{h}{2}-\xi\right)^{1-\alpha}\,d\xi\Bigg].
\end{aligned}
\tag{5.10}
$$

We ignore the first term in the square bracket of equation (5.10) when $j=1$. The integral

$$
\int_{x_l}^{x_{l+1}}\left(x_j+\frac{h}{2}-\xi\right)^{1-\alpha}d\xi = \frac{h^{2-\alpha}}{2^{2-\alpha}(2-\alpha)}\left[(2j-2l+1)^{2-\alpha}-(2j-2l-1)^{2-\alpha}\right].
\tag{5.11}
$$

Similarly,

$$
\int_{x_j}^{x_j+\frac{h}{2}}\left(x_j+\frac{h}{2}-\xi\right)^{1-\alpha}\,d\xi = \frac{h^{2-\alpha}}{2^{2-\alpha}(2-\alpha)}.
\tag{5.12}
$$

Hence equation (5.10) becomes

$$
\begin{aligned}
D_x^\alpha u\left(x_j+\frac{h}{2}\right) = C_\alpha\Bigg(&\sum_{l=1}^{j-1}(b_{2l-1}-2b_{2l}+b_{2l+1})\left[(2j-2l+1)^\beta-(2j-2l-1)^\beta\right]\\
&+(b_{2j-1}-2b_{2j}+b_{2j+1})\Bigg),\quad j=1,2,\ldots,N.
\end{aligned}
\tag{5.13}
$$

where $\beta=2-\alpha$ and $C_\alpha=\frac{2^{\alpha-1}}{h^\alpha\Gamma(3-\alpha)}$. The boundary conditions in (5.2) yield

$$
b_1=\delta_1,\ b_{2N+1}=\delta_2.
\tag{5.14}
$$

Thus substituting the collocation points into (5.1) and using (5.13) we obtain N equations. These equations together with the continuity equation (2.15) and

boundary conditions in (5.14) give a square linear system of size $(2N+1)$ given by

$$C_\alpha \; P \; \mathbf{b} = \mathbf{f}, \tag{5.15}$$

where $\mathbf{b} = [b_1, b_2, \ldots, b_{2N+1}]^T$, $\mathbf{f} = [f_1, f_2, \ldots, f_N, 0, \ldots, 0, \delta_1, \delta_2]^T$,

$$P = \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 3^\beta - 1 & -2(3^\beta - 1) & 3^\beta & -2 & 1 & \cdots & \cdots & 0 & 0 \\ 5^\beta - 3^\beta & -2(5^\beta - 3^\beta) & 5^\beta - 1 & -2(3^\beta - 1) & 3^\beta & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & -2 & 1 \\ 0 & a & -2a & a & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & 0 & a & -2a & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & a & 0 \\ a & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & a \end{pmatrix}, \tag{5.16}$$

and $a = (C_\alpha)^{-1}$. The solution to the fractional differential equation (5.1) is then obtained via equation (5.5).

We now illustrate the application of our method to fractional ordinary and partial differential equations with examples.

Example 5.1: Consider the fractional ordinary differential equation

$$\frac{d^\alpha u(x)}{dx^\alpha} = \frac{x^{3-\alpha}}{\Gamma(4-\alpha)}, \quad 1 < \alpha < 2, \tag{5.17}$$

with the exact solution $u(x) = \frac{1}{6}x^3$ has boundary conditions $u(0) = 0$ and $u(1) = \frac{1}{6}$.

We shall use our method to obtain it's numerical solution.

Figure 5.1 shows the numerical and exact solutions to the fractional differential equation (5.17) when $\alpha = 1.5$ and $N = 50$. The error is depicted in Figure (5.2).

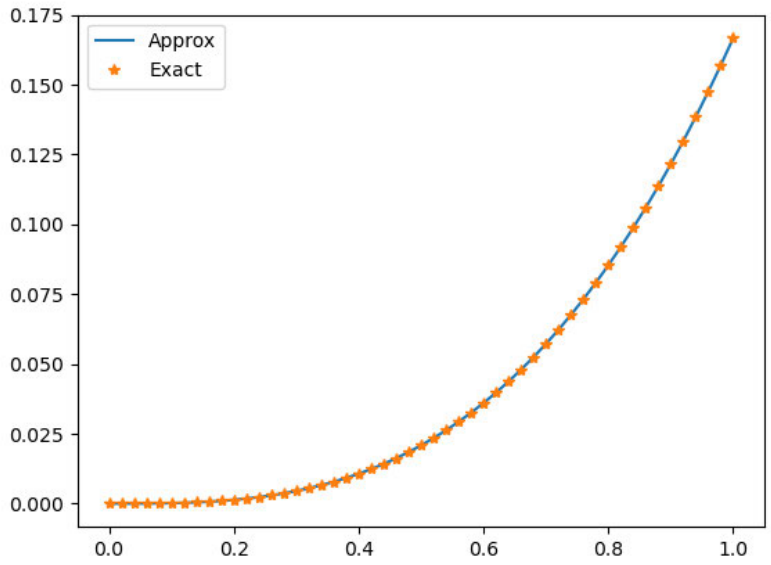

Figure 5.1: Solution of $u(x)$ for example 5.1 when $N = 50$, and $\alpha = 1.5$.

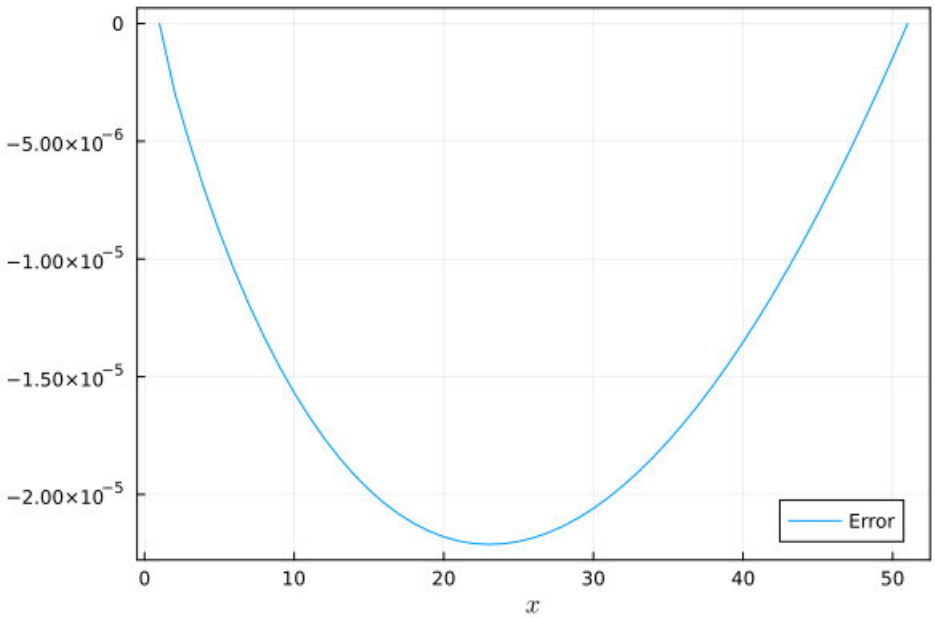

Figure 5.2: Error plot for example 5.1 when $N = 50$, and $\alpha = 1.5$.

Table 5.1 show the convergence rates when $\alpha = 1.5$ and $N = 50$. It is approximately 1.5.

Table 5.1: Convergence rates for example 5.1 when $\alpha = 1.5$ and $N = 50$.

x	Rate	x	Rate
0.02	1.6212	0.5	1.5147
0.08	1.5395	0.6	1.5134
0.1	1.5348	0.7	1.5123
0.2	1.5238	0.8	1.5115
0.3	1.5192	0.9	1.5108
0.4	1.5165	0.96	1.5105

This shows that our method is suitable for solving fractional differential equations.

5.2 Application to space-fractional partial differential equation

In this case, we re-write the approximate representation of $u_c(x)$ in equation (5.5) as

$$u_c(x,t) = \sum_{k=1}^{3} b_{k+2(i-1)}(t) B_k(x) \tag{5.18}$$

to accommodate the variable t. As an example we consider the fractional diffusion equation

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^\alpha u(x,t)}{\partial x^\alpha} + f(x), \tag{5.19}$$

subject to the initial conditions

$$u(a,t_0) = \omega(x) \tag{5.20}$$

and boundary condition

$$u(a,t) = \delta_3,\ u(b,t) = \delta_4. \tag{5.21}$$

To solve this problem, we apply the trapezoidal rule to equation (5.19) to get

$$u_i^{j+1} - \frac{\Delta t}{2} D_\alpha u_i^{j+1} = u_i^j + \frac{\Delta t}{2} \left(D_\alpha u_i^j + f_i^{j+1} + f_i^j \right), \tag{5.22}$$

where $u_i^j = u(x_i, t_j)$, j is the index for discrete time t, i is index for discrete x, Δt is the time step, t_f is the final time, t_0 is the initial time and $D_\alpha u = \frac{\partial^\alpha u(x,t)}{\partial x^\alpha}$.

We use equation (5.13) in (5.22) and evaluate at the collocation point in intervals $1, 2, \ldots, N$. The resulting system coupled with the boundary and continuity conditions form a linear system represented in the form

$$C_\alpha Q \mathbf{b} = \mathbf{R}, \tag{5.23}$$

where Q is a $(2N+1)$ square matrix, $\mathbf{b} = [b_1, b_2, \ldots, b_{2N+1}]^T$ and

$\mathbf{R} = [R_1, R_2, \ldots, R_N, 0, \ldots, 0, \delta_3, \delta_4]^T$.

5.3 Stability

Consider now the fractional diffusion equation given by

$$u_t - d(x,t) \frac{\partial^\alpha}{\partial x^\alpha} u(x,t) = p(x,t), \tag{5.24}$$

where $d(x,t)$ and $p(x,t)$ are coefficients of the fractional derivative and the source term, respectively. Consider the homogeneous case of (5.24) i.e $p(x,t) = 0$ and put $d = 1$ is a constant.

Apply the trapezoidal rule to get

$$u(x, t_{n+1}) - u(x, t_n) = \frac{\Delta t}{2} \left(D^\alpha u(x, t_{n+1}) + D^\alpha u(x, t_n) \right), \tag{5.25}$$

which we write as

$$u^{n+1}(x) - u^n(x) = \frac{\Delta t}{2} \left(D^\alpha u^{n+1}(x) + D^\alpha u^n(x) \right). \tag{5.26}$$

If x is in the m^{th} interval then we write equation (5.26) as

$$u_m^{n+1}(x) - u_m^n(x) = \frac{\Delta t}{2}\left(D^\alpha u_m^{n+1}(x) + D^\alpha u_m^n(x)\right), \tag{5.27}$$

where $u_m^n(x) = \sum_{k=1}^{3} b_{k+2(m-1)}^n B_k(x)$.

Evaluate (5.27) at the collocation point $x_m + \frac{1}{2}h$

$$u_m^{n+1}\left(x_m+\frac{1}{2}h\right) - u_m^n\left(x_m+\frac{1}{2}h\right) = \frac{\Delta t}{2}\left(D^\alpha u_m^{n+1}\left(x_m+\frac{1}{2}h\right) + D^\alpha u_m^n\left(x_m+\frac{1}{2}h\right)\right), \tag{5.28}$$

$$\begin{aligned} D^\alpha u_m^n\left(x_m+\frac{1}{2}h\right) &= \frac{1}{\Gamma(2-\alpha)}\sum_{j=1}^{m-1}\int_{x_j}^{x_{j+1}} \frac{(u_j^n)''(\xi)}{\left(x_m+\frac{1}{2}h-\xi\right)^{\alpha-1}}d\xi \\ &\quad + \frac{1}{\Gamma(2-\alpha)}\int_{x_m}^{x_m+\frac{1}{2}h} \frac{(u_m^n)''(\xi)}{\left(x_m+\frac{1}{2}h-\xi\right)^{\alpha-1}}d\xi \\ &= \frac{2^{\alpha-1}}{h^\alpha\Gamma(3-\alpha)}\left[\sum_{j=1}^{m-1}\left[(2m-2j+1)^{2-\alpha}-(2m-2j-1)^{2-\alpha}\right]\right. \\ &\quad \left.\times(b_{2j-1}^n - 2b_{2j}^n + b_{2j+1}^n) + b_{2m-1}^n - 2b_{2m}^n + b_{2m+1}^n\right] \\ &= \frac{2^{\alpha-1}}{h^\alpha\Gamma(3-\alpha)}\left[\sum_{j=1}^{m-1} c_j(b_{2j-1}^n - 2b_{2j}^n + b_{2j+1}^n) + b_{2m-1}^n - 2b_{2m}^n + b_{2m+1}^n\right], \end{aligned} \tag{5.29}$$

where $c_j = (2m-2j+1)^{2-\alpha} - (2m-2j-1)^{2-\alpha}$,

$$u_m^n\left(x_m+\frac{1}{2}h\right) = \frac{1}{4}b_{2m-1}^n + \frac{1}{2}b_{2m}^n + \frac{1}{4}b_{2m+1}^n. \tag{5.30}$$

Substitute (5.29) and (5.30) into 5.28 to get

$$\begin{aligned} &b_{2m-1}^{n+1} + 2b_{2m}^{n+1} + b_{2m+1}^{n+1} - C\sum_{j=1}^{m-1} c_j(b_{2j-1}^{n+1} - 2b_{2j}^{n+1} + b_{2j+1}^{n+1}) - C\left(b_{2m-1}^{n+1} - 2b_{2m}^{n+1} + b_{2m+1}^{n+1}\right) \\ &= b_{2m-1}^n + 2b_{2m}^n + b_{2m+1}^n + C\sum_{j=1}^{m-1} c_j(b_{2j-1}^n - 2b_{2j}^n + b_{2j+1}^n) + C\left(b_{2m-1}^n - 2b_{2m}^n + b_{2m+1}^n\right), \end{aligned} \tag{5.31}$$

where $C = \frac{2^\alpha \Delta t}{h^\alpha \Gamma(3-\alpha)}$. Let $b_j^n = \lambda^n e^{ijhk}$,

$$\begin{aligned}
&\lambda^{n+1}\Bigg[(1-C)e^{i(2m-1)hk} + 2(1+C)e^{i2mhk} + (1-C)e^{i(2m+1)hk} \\
&-C\sum_{j=1}^{m-1} c_j\left(e^{i(2j-1)hk} - 2e^{i2jhk} + e^{i(2j+1)hk}\right)\Bigg] \\
&= \lambda^n\Bigg[(1-C)e^{i(2m-1)hk} + 2(1+C)e^{i2mhk} + (1-C)e^{i(2m+1)hk} \\
&+C\sum_{j=1}^{m-1} c_j\left(e^{i(2j-1)hk} - 2e^{i2jhk} + e^{i(2j+1)hk}\right)\Bigg].
\end{aligned} \tag{5.32}$$

Let $X = \sum_{j=1}^{m-1} c_j(e^{i(2j-1)hk} - 2e^{i2jhk} + e^{i(2j+1)hk})$. Divide (5.32) by e^{i2mhk} to get

$$\begin{aligned}
\lambda &= \frac{(1-C)(e^{ihk}+e^{-ihk}) + 2(1+C) + CXe^{-i2mhk}}{(1-C)(e^{ihk}+e^{-ihk}) + 2(1+C) - CXe^{-i2mhk}}, \\
&= \frac{2(1-C)\cos(hk) + 2(1+C) + CXe^{-i2mhk}}{2(1-C)\cos(hk) + 2(1+C) - CXe^{-i2mhk}}, \\
&= \frac{R+A}{R-A},
\end{aligned} \tag{5.33}$$

where $R = 2(1-C)\cos(hk) + 2(1+C)$ is real and $A = CXe^{-i2mhk}$ is complex.

$$|\lambda|^2 = \frac{R^2 + |A|^2 + 2R\ Re(A)}{R^2 + |A|^2 + 2R\ Re(A)} = \frac{N}{D}, \tag{5.34}$$

$$N - D = 4R\ Re(A), \tag{5.35}$$

since $R = 2(1+\cos(hk)) + 2C(1-\cos(hk)) \geq 0$. If $Re(A) \leq 0$ then $|\lambda|^2 \leq 1 \implies$ $|\lambda| \leq 1$ and conditional stability.

However if Δt is chosen small enough so that $C << 1$ then (5.33) implies

$$|\lambda| = \frac{2\cos(hk)+2}{2\cos(hk)+2} = 1, \tag{5.36}$$

and conditional stability with condition

$$\Delta t << \frac{h^\alpha \Gamma(3-\alpha)}{2^\alpha}. \tag{5.37}$$

5.4 Convergence Analysis

Having shown that our method is stable in the previous section, we now show that our method is consistent. The determinant of the coefficient matrix $C_\alpha Q$ is computed as $det(C_\alpha Q) = (-1)^{\lfloor N/2 \rfloor}\ 2N(C_\alpha)^N \neq 0$ for all values of α. The coefficient matrix is nonsingular. This implies that our method is consistent. Hence our method is convergent. Suppose

$$D^\alpha u = \lambda u,$$

where λ is a constant. This simplifies to

$$\begin{aligned} \int_a^x \frac{u''(s)}{(x-s)^{\alpha-1}} ds &= \Gamma(2-\alpha)\lambda u \\ &= f(x). \end{aligned} \tag{5.38}$$

Let $x_i = x_1 + (i-1)h, \quad i = 1, 2, \ldots, N+1$, the collocation points are at

$$\bar{x}_i = x_i + \frac{h}{2} = x_1 + \left(i - \frac{1}{2}\right) h.$$

The collocation solution in the i_{th} interval is

$$\begin{aligned} u_c^i &= \sum_{j=1}^{3} b_{j+2(i-1)} B_j(z) \\ &= b_{2i-1} B_1(z) + b_{2i} B_2(z) + b_{2i+1} B_3(z). \end{aligned}$$

Then

$$u_c^{''(i)}(x) = \frac{2}{h^2} \left[b_{2i-1} - 2b_{2i} + b_{2i+1} \right].$$

If x is in the i_{th} interval then

$$\int_{x_1}^x \frac{u''(s)}{(x-s)^{\alpha-1}} ds = \sum_{k=1}^{i-1} I_k + I_i, \tag{5.39}$$

with I_k defined as

$$\begin{aligned} I_k &= \int_{x_k}^{x_{k+1}} \frac{u^{(k)''}(s)}{(x-s)^{\alpha-1}} ds \\ &= \frac{2}{h^2(2-\alpha)} \left[(x-x_k)^{2-\alpha} - (x-x_{k+1})^{2-\alpha}\right] (b_{2k-1} - 2b_{2k} + b_{2k+1}), \end{aligned} \tag{5.40}$$

$k = 1, 2, , \ldots, i-1.$

Therefore

$$\begin{aligned} I_i(x) &= \int_{x_i}^{x} \frac{u^{(i)''}(s)}{(x-s)^{\alpha-1}} ds, \\ &= \frac{2(x-x_i)^{2-\alpha}}{h^2(2-\alpha)} \left[b_{2i-1} - 2b_{2i} + b_{2i+1}\right]. \end{aligned} \tag{5.41}$$

Substitute the collocation points $\bar{x}_i, i = 1, 2, \ldots, N$ into (5.40) and (5.41) to get

$$\begin{aligned} I_k(\bar{x}_i) = \frac{2^{\alpha-1}}{2-\alpha} h^{-\alpha} &\left[(2i-2k+1)^{2-\alpha} - (2i-2k-1)^{2-\alpha}\right] \\ &\times (b_{2k-1} - 2b_{2k} + b_{2k+1}), \quad k < i. \end{aligned} \tag{5.42}$$

$$I_i(\bar{x}_i) = \frac{2^{\alpha-1}}{2-\alpha} h^{-\alpha} \left[b_{2i-1} - 2b_{2i} + b_{2i+1}\right]. \tag{5.43}$$

Hence, substituting the collocation points $\bar{x}_i$ into (5.38) gives

$$\begin{aligned} c_\alpha \Bigg[\sum_{k=1}^{i-1} &\left[(2i-2k+1)^{2-\alpha} - (2i-2k-1)^{2-\alpha}\right] \\ &\times (b_{2k-1} - 2b_{2k} + b_{2k+1}) + (b_{2i-1} - 2b_{2i} + b_{2i+1}) \Bigg] = f(\bar{x}_i), \end{aligned} \tag{5.44}$$

where $c_\alpha = \frac{2^{\alpha-1}h^{-\alpha}}{2-\alpha}$ and $i = 1, 2, \ldots, N$.

We now define a linear operator L by $(Lu)(x) = \int_a^x \frac{u''(s)}{(x-s)^{\alpha-1}} ds$. Let $x \in [x_i, x_{i+1}]$ and $\bar{x}_i$ be the midpoint of $[x_i, x]$ so that the length of this interval is given by $w_i h$ for $w_i \in (0, 1]$. Define $e(x) = u(x) - u_c(x)$

$$Le(x) = \sum_{k=1}^{i-1} \int_{x_k}^{x_{k+1}} \frac{e''(s)}{(x-s)^{\alpha-1}} ds + \int_{x_i}^{x} \frac{e''(s)}{(x-s)^{\alpha-1}} ds, \tag{5.45}$$

$$= \sum_{k=1}^{i-1} e''(\xi_k) \int_{x_k}^{x_{k+1}} \frac{1}{(x-s)^{\alpha-1}} ds + e''(\xi_i) \int_{x_i}^{x} \frac{1}{(x-s)^{\alpha-1}} ds, \tag{5.46}$$

where we have used the integral mean value theorem, $\xi_k \in (x_k, x_{k+1})$ and $\xi_i \in (x_i, x)$. Now

$$\begin{aligned} e''(\xi_k) &= u''(\xi_k) - u_c''(\xi_k), \\ &= u''(\xi_k) - u_c''(\bar{x}_k), \end{aligned}$$

since $u_c''(x)$ is constant in $[x_k, x_{k+1}]$.

We assume that for $h << 1$, ξ_k can be approximated by $\bar{x}_k$ so that $e''(\xi_k) = e''(\bar{x}_k) = u''(\bar{x}_k) - u_c''(\bar{x}_k)$. By Taylor's theorem,

$u(x_{k+1}) = u(\bar{x}_k + \frac{h}{2}) = \sum_{p=0}^{3} \frac{u^{(p)}(\bar{x}_k)}{p!} \left(\frac{h}{2}\right)^p + u^4(\beta_1) \left(\frac{h}{2}\right)^4, \quad \beta_1 \in (\bar{x}_k, x_{k+1}).$

$u(x_k) = u(\bar{x}_k - \frac{h}{2}) = \sum_{p=0}^{3} (-1)^p \frac{u^{(p)}(\bar{x}_k)}{p!} \left(\frac{h}{2}\right)^p + u^4(\beta_2) \left(\frac{h}{2}\right)^4, \quad \beta_2 \in (x_{k-1}, \bar{x}_k).$

Thus $u(x_{k+1}) + u(x_k) = 2u(\bar{x}_k) + u''(\bar{x}_k)\frac{h^2}{2} + O(h^4)$,

$$u''(\bar{x}_k) = \frac{2}{h^2} \left[u(x_{k+1}) - 2u(\bar{x}_k) + u(x_k)\right] + O(h^2), \tag{5.47}$$

$$u_c''(\bar{x}_k) = \frac{2}{h^2} \left[b_{2k-1} - b_{2k} + b_{2k+1}\right]. \tag{5.48}$$

Now $b_{2k-1} = u_c(x_k)$ and $b_{2k+1} = u_c(x_{2k+1})$,

$$\begin{aligned} u_c(\bar{x}_k) &= \frac{1}{4}\left(b_{2k-1} + b_{2k+1}\right) + \frac{1}{2} b_{2k}, \\ &= \frac{1}{2}\left(\frac{u_c(x_k) + u_c(x_{k+1})}{2}\right) + \frac{1}{2} b_{2k}, \\ &= \frac{1}{2} u_c(\bar{x}_k) + \frac{1}{2} b_{2k} + O(h^2). \end{aligned} \tag{5.49}$$

We have assumed that the collocation solution at the midpoint of $[x_k, x_{k+1}]$ is approximately the average of the solutions at the endpoints.

Hence $b_{2k} = u_c(\bar{x}_k) + O(h^2)$.

$$u_c''(x_k) = \frac{2}{h^2} \left[u_c(x_{k+1}) - 2u_c(\bar{x}_k) + u_c(x_k)\right]. \tag{5.50}$$

We assume that the approximation of the second derivative in (5.47) agrees to within $O(h^2)$ with that in (5.50), thus

$$u''(\bar{x}_k) - u_c''(\bar{x}_k) = O(h^2). \tag{5.51}$$

Hence

$$|e''(\xi_k)| \leq M_k h^2, \tag{5.52}$$

for some constant M_k.

Equation (5.51) has been verified numerically. We shall approximate the integral using the midpoint rule.

$$\int_{x_k}^{x_{k+1}} \frac{1}{(x-s)^{\alpha-1}} ds = \frac{h}{(x-\bar{x}_k)^{\alpha-1}} + g''(\bar{\xi}_k)\frac{h^3}{24}, \tag{5.53}$$

where $g(s) = \frac{1}{(x-s)^{\alpha-1}}$ and $\bar{\xi}_k \in (x_k, x_{k+1})$. Hence

$$\begin{aligned}
\sum_{k=1}^{i-1} \int_{x_k}^{x_{k+1}} \frac{1}{(x-s)^{\alpha-1}} ds &= h \sum_{k=1}^{i-1} \frac{1}{(x-\bar{x}_k)^{\alpha-1}} + \frac{h^3}{24} \sum_{k=1}^{i-1} g''(\bar{\xi}_k) \\
&\leq \frac{h(i-1)}{(x-\bar{x}_i)^{\alpha-1}} + \frac{\alpha(\alpha-1)h^3}{24} \sum_{k=1}^{i-1} \frac{1}{(x-\bar{\xi}_k)^{\alpha+1}} \\
&\leq \frac{h(i-1)}{(x-x_i)^{\alpha-1}} + \frac{\alpha(\alpha-1)h^3}{24}(i-1)\frac{1}{(x-x_i)^{\alpha+1}}.
\end{aligned} \tag{5.54}$$

Similarly, the second term in (5.45) can be simplified to

$$\begin{aligned}
\int_{x_i}^{x} \frac{e''(s)}{(x-s)^{\alpha-1}} ds &\leq M_i h^2 \left[\frac{w_1 h}{(x-\bar{x}_i)^{\alpha-1}} + g''(\bar{\xi}_i)\frac{w_1^3 h^3}{24} \right], \quad \bar{\xi}_i \in (x_i, x) \\
&\leq M_i h^2 \left[\frac{w_1 h}{(x-\bar{x}_i)^{\alpha-1}} + \frac{\alpha(\alpha-1)w_1^3 h^3}{24} \frac{1}{(x-\bar{\xi}_i)^{\alpha+1}} \right].
\end{aligned} \tag{5.55}$$

If $\bar{x}_i = x_i + w_2 h$, $\bar{\xi}_i = x_i + w_3 h$, $w_2, w_3 \in (0,1)$ and $M_i = \max_{k \leq i-1} M_k$ then from (5.46), (5.52), (5.54) and (5.55) it follows that

$$\begin{aligned}
|Le(x)| \leq\ & M_1 h^3 (i-1) \left[\frac{1}{(x-x_i)^{\alpha-1}} + \frac{\alpha(\alpha-1)h^2}{24} \frac{1}{(x-x_i)^{\alpha+1}} \right] \\
& + M_i h^3 \left[\frac{w_1}{(x-x_i-w_2 h)^{\alpha-1}} + \frac{\alpha(\alpha-1)}{24} \frac{w_1^3 h^2}{(x-x_i-w_3 h)^{\alpha+1}} \right].
\end{aligned} \tag{5.56}$$

The function on the RHS of (5.56) is an upper bound for $|L\, e(x)|\ \forall x$. In particular, when $x = x_{N+1}$, $i = N$, we have

$$\begin{aligned}
||L\ e(x)||_\infty =\ & M_1 h^{4-\alpha} (N-1) \left[1 + \frac{\alpha(\alpha-1)}{24} \right] \\
& + M_N h^{4-\alpha} \left[\frac{w_1}{1-w_2} + \frac{\alpha(\alpha-1)}{24} \frac{w_1^3}{1-w_3} \right].
\end{aligned} \tag{5.57}$$

But $N-1 \approx N \propto h^{-1}$ for large N. Thus there exists constants k_1 and k_2 such that

$$||L\ e(x)||_\infty \leq k_1 h^{3-\alpha} + k_2 h^{4-\alpha}. \tag{5.58}$$

Now assuming that L^{-1} is bounded above, we have

$$\begin{aligned} |(u-u_c)(x)| &\leq ||(u-u_c)(x)||_\infty, \\ &= ||L^{-1}L(u-u_c)(x)||_\infty, \\ &\leq ||L^{-1}||\ ||L\ e(x)||_\infty, \\ &= O(h^{3-\alpha}) \end{aligned} \tag{5.59}$$

from (5.58). This shows that the fractional differential equation (5.1) is of order $(3-\alpha)$ in space.

We then infer from the trapezoidal rule that the local error is given by $\frac{\Delta t^3}{12} u_{tt}(x, \xi_n)$, $\xi_n \in [t_n, t_{n+1}]$.

Theorem 5.1: *Let $u(x, t_{n+1})$ be the exact solution of (5.1), and assume that $u(x, t_{n+1}) \in C^4[a, b]$. At time t_{n+1}, the global error is $t_{n+1} K \Delta t^2$ for constant K. Hence*

$$\begin{aligned} ||u_c(x, t_{n+1}) - u(x, t_{n+1})||_\infty &= O(\Delta t^2 + h^{3-\alpha}), \\ &= O(\Delta t^2 + \Delta x^{3-\alpha}), \end{aligned} \tag{5.60}$$

for some constants K.

This shows that the quadratic B-spline OCFE for fractional differential equations is second order in time and order $3-\alpha$ in space.

5.5 Numerical examples

The rate of convergence is calculated as $\log_2 \frac{||\mathbf{e}||_\infty^h}{||\mathbf{e}||_\infty^{h/2}}$ where the components of $\mathbf{e}$ are given by $e_i = u_i(x) - U_i(x)$, $u_i(x)$ and $U_i(x)$ are exact and approximate solution

at the nodes respectively.

Example 5.2: We consider the equation

$$\frac{\partial u}{\partial t} - \frac{24x^{\alpha}}{\Gamma(5+\alpha)} \frac{\partial^{\alpha} u}{\partial x^{\alpha}} + 2u = 0, \tag{5.61}$$

with initial condition

$$u(x,0) = x^{4+\alpha}, \tag{5.62}$$

and boundary conditions

$$\left.\begin{aligned} u(0,t) &= 0, \\ u(1,t) &= e^{-t}. \end{aligned}\right\} \tag{5.63}$$

The exact solution is $u(x,t) = x^{\alpha+4}e^{-t}$.

The discretized form of equation (5.61) is

$$(1+\Delta t)u_i^{j+1} - \frac{\Delta t}{2} m_i D_{\alpha} u_i^{j+1} = (1-\Delta t)u_i^{j} + \frac{\Delta t}{2} m_i D_{\alpha} u_i^{j}, \tag{5.64}$$

where $m(x) = \frac{24x^{\alpha}}{\Gamma(5+\alpha)}$, i and j represents discretization in x and t, respectively.

Figures 5.3 and 5.4 display the efficiency of the quadratic OCFE method to handle the fractional diffusion equation using a small number of intervals.

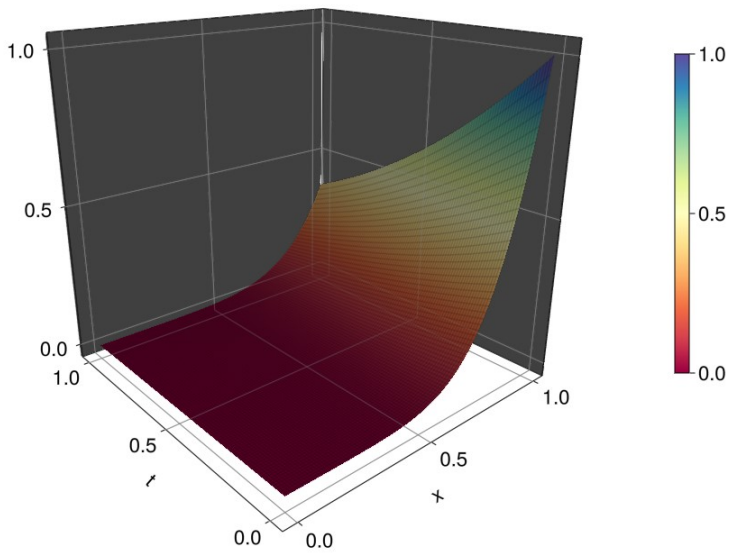

Figure 5.3: 3D plot of solution for example 5.2 at $N = Nt = 100,\ \alpha = 1.5$.

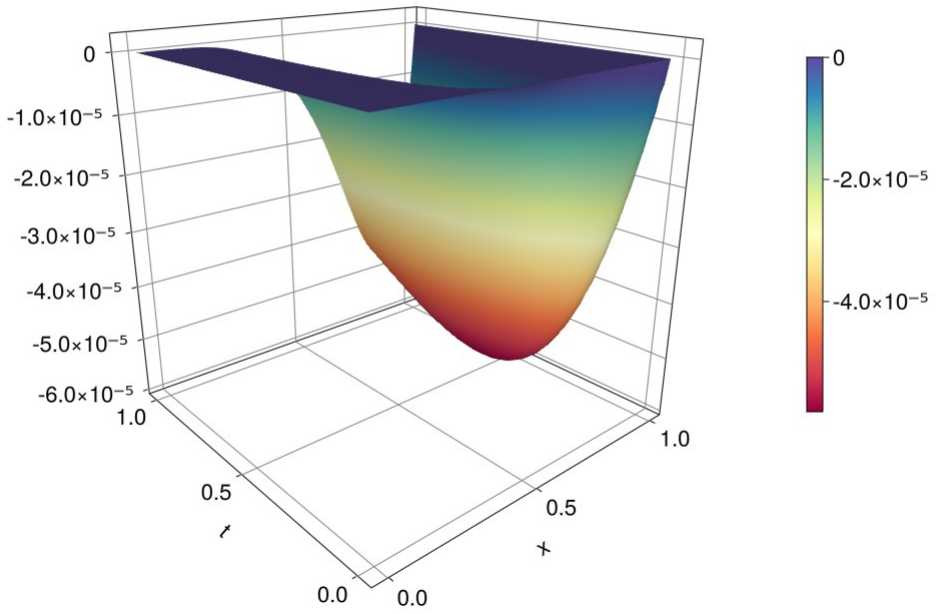

Figure 5.4: Error plot for example 5.2 at $N = Nt = 50,\ \alpha = 1.5$.

Table 5.2 shows the absolute error for various values of α at the time $t = 1$. Table 5.3 shows that the convergence order is $(3-\alpha)$. The global convergence rate

is represented by Ord_∞.

Table 5.2: Table of Absolute errors at for different values of α when $N = Nt = 100$.

x	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 1.8$
0.10	1.560×10^{-8}	1.590×10^{-8}	1.390×10^{-8}	6.300×10^{-9}
0.30	5.330×10^{-7}	8.570×10^{-7}	1.190×10^{-6}	1.140×10^{-6}
0.50	2.680×10^{-6}	5.380×10^{-6}	9.230×10^{-6}	1.220×10^{-5}
0.70	7.650×10^{-6}	1.760×10^{-5}	3.300×10^{-5}	4.930×10^{-5}
0.90	8.830×10^{-6}	2.030×10^{-5}	3.960×10^{-5}	6.450×10^{-5}
0.98	2.420×10^{-6}	5.820×10^{-6}	1.180×10^{-5}	2.000×10^{-5}

Table 5.3: Convergence rates for example 5.2 for different values of α when $N = Nt = 50$.

x	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 1.8$
0.10	1.7746	1.5451	1.2872	0.5882
0.30	1.8834	1.6746	1.4615	1.0995
0.50	1.8900	1.6834	1.4760	1.1411
0.70	1.8892	1.6814	1.4768	1.1522
0.90	1.8872	1.6808	1.4771	1.1561
0.98	1.8860	1.6802	1.4771	1.1570
Ord_∞	1.8649	1.6599	1.4545	1.1060

Example 5.3: We consider the equation

$$\frac{\partial u}{\partial t} - \frac{\Gamma(5-\alpha)}{24} x^\alpha \frac{\partial^\alpha u}{\partial x^\alpha} + 2u = 0, \tag{5.65}$$

with initial condition

$$u(x,0) = x^4, \tag{5.66}$$

and boundary conditions

$$\left.\begin{aligned} u(0,t) &= 0, \\ u(1,t) &= e^{-t}. \end{aligned}\right\} \tag{5.67}$$

The exact solution is $u(x,t) = x^4 e^{-t}$.

The discretized form of (5.65) is

$$(1+\Delta t)u_i^{j+1} - \frac{\Delta t}{2} m_i D_\alpha u_i^{j+1} = (1-\Delta t)u_i^j + \frac{\Delta t}{2} m_i D_\alpha u_i^j, \tag{5.68}$$

where $m(x) = \frac{\Gamma(5-\alpha)}{24} x^\alpha$.

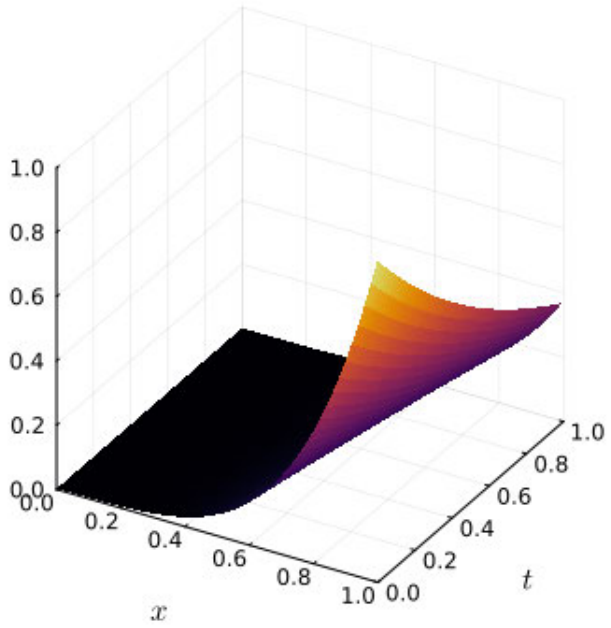

Figure 5.5: 3D plot of approximate solution for example 5.3 when $N = Nt = 50$, $\alpha = 1.5$.

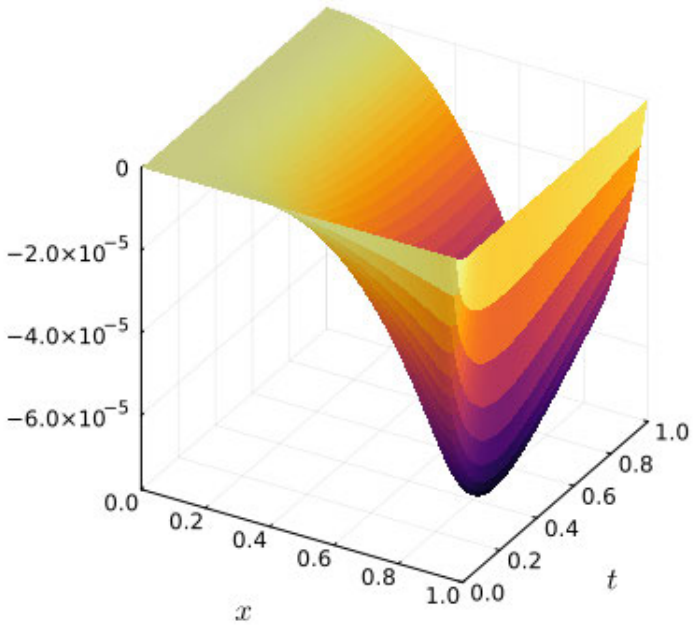

Figure 5.6: 3D plot of errors for example 5.3 when $N = Nt = 50$, $\alpha = 1.5$.

Figures 5.5 and 5.6 depicts the approximate solution and error for this problem respectively. Tables 5.4 and 5.5 show the comparison between the present work and that of [68] for different values of α. The present results are better than that of [68] due to lower infinity norm of errors. Table 5.6 shows the convergence order for this example to be approximately $3 - \alpha$ for different values of α.

Table 5.4: Comparison of L_∞ errors when $h = \Delta t$ at $t = 1,\ \alpha = \{1.2,\ 1.4\}$.

	$\alpha = 1.2$		$\alpha = 1.4$	
h	[68]	Present	[68]	Present
1/20	7.122×10^{-4}	1.387×10^{-4}	5.162×10^{-4}	2.050×10^{-4}
1/25	4.801×10^{-4}	9.303×10^{-5}	3.401×10^{-4}	1.445×10^{-4}
1/30	3.432×10^{-4}	6.758×10^{-5}	2.411×10^{-4}	1.090×10^{-4}

Table 5.5: Comparison of L_∞ errors when $h = \Delta t$ at $t = 1,\ \alpha = \{1.5,\ 1.8\}$.

	$\alpha = 1.5$		$\alpha = 1.8$	
h	[68]	Present	[68]	Present
1/15	7.439×10^{-4}	3.584×10^{-4}	4.361×10^{-4}	3.186×10^{-4}
1/20	4.399×10^{-4}	2.361×10^{-4}	2.532×10^{-4}	2.323×10^{-4}
1/25	2.880×10^{-4}	1.704×10^{-4}	1.645×10^{-4}	1.808×10^{-4}
1/30	2.040×10^{-4}	1.309×10^{-4}	1.158×10^{-4}	1.470×10^{-4}

Table 5.6: Convergence rates for example 5.3 when $N = Nt = 50$.

x	$\alpha = 1.2$	$\alpha = 1.4$	$\alpha = 1.5$	$\alpha = 1.8$
0.10	1.8109	1.5839	1.4714	1.1045
0.30	1.7997	1.5896	1.4858	1.1657
0.50	1.7907	1.5841	1.4821	1.1705
0.70	1.7789	1.5783	1.4786	1.1710
0.90	1.7763	1.5768	1.4778	1.1713
0.98	1.7739	1.5757	1.4771	1.1713
Ord_∞	1.7794	1.5783	1.4786	1.1711

Example 5.4: We consider the fractional diffusion equation

$$\frac{\partial u}{\partial t} - \Gamma(3-\alpha)\ x^{\alpha-1}\ \frac{\partial^\alpha u}{\partial x^\alpha} = (x^2+1)\cos(t+1) - 2x\sin(t+1), \tag{5.69}$$

with initial condition

$$u(x,0) = (x^2+1)\sin(1), \tag{5.70}$$

and boundary conditions

$$\left.\begin{aligned} u(0,t) &= \sin(t+1), \\ u(1,t) &= 2\sin(t+1). \end{aligned}\right\} \tag{5.71}$$

The exact solution is $u(x,t) = (x^2+1)\sin(t+1)$.

The discretized form of (5.69) is

$$u_i^{j+1} - \frac{\Delta t}{2} m_i D_\alpha u_i^{j+1} = u_i^j + \frac{\Delta t}{2} m_i D_\alpha u_i^j + \frac{\Delta t}{2}(r_i^{j+1} + r_i^j), \tag{5.72}$$

where $m(x) = \Gamma(3-\alpha)\ x^{\alpha-1}$ and $r(x,t) = (x^2+1)\cos(t+1) - 2x\sin(t+1)$.

Figure 5.7 shows that the approximate solution overlaid with a mesh plot of the exact solution. There is a perfect match between the exact and approximate solutions. The error plot for this example is represented by Figure 5.8.

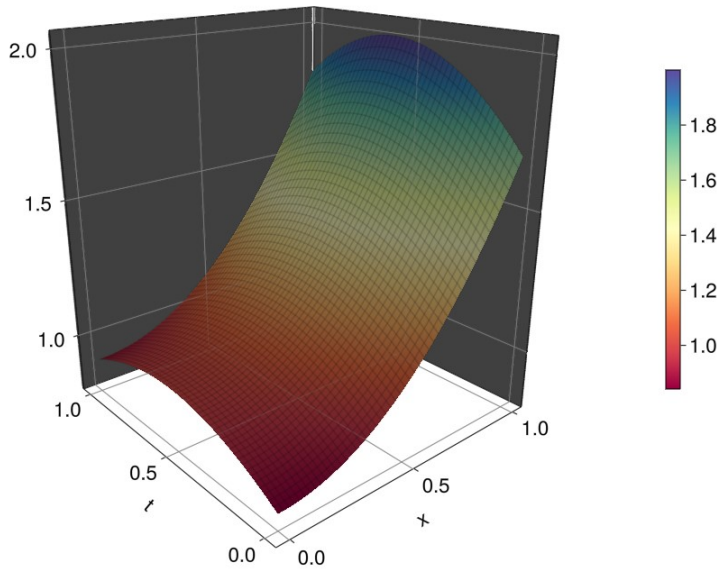

Figure 5.7: 3D plot of solution for example 5.4 at $N = Nt = 50,\ \alpha = 1.5$.

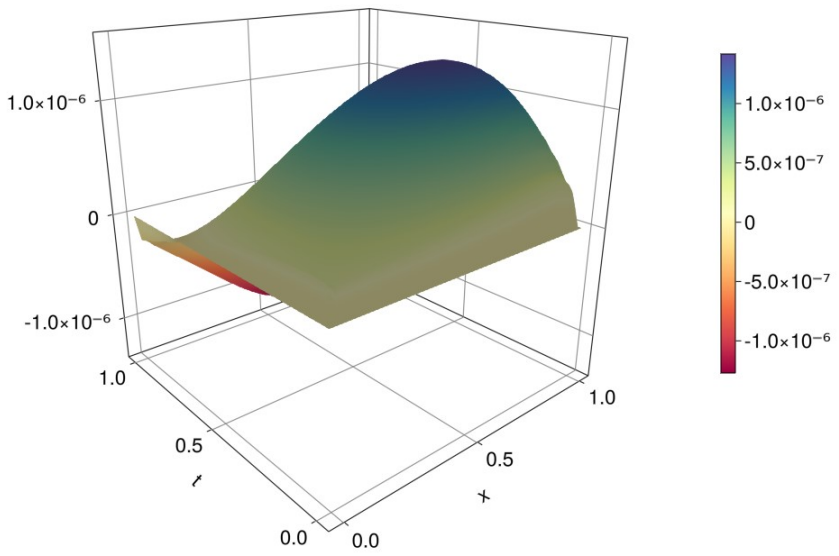

Figure 5.8: Error plot for example 5.4 at $N = Nt = 50,\ \alpha = 1.5$.

Table 5.7 shows that absolute error for different values of α. The convergence order is shown in Table 5.8. Ord_∞ represents the global order. The nodal order is better than the global order.

Table 5.7: Table of Absolute errors at for different values of α when $N = Nt = 100$.

x	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 1.8$
0.10	9.290×10^{-8}	5.650×10^{-8}	7.730×10^{-8}	2.140×10^{-7}
0.30	3.590×10^{-7}	1.930×10^{-7}	1.790×10^{-7}	3.260×10^{-7}
0.50	5.600×10^{-7}	3.190×10^{-7}	2.720×10^{-7}	3.780×10^{-7}
0.70	5.310×10^{-7}	3.440×10^{-7}	2.890×10^{-7}	3.440×10^{-7}
0.90	2.310×10^{-7}	1.820×10^{-7}	1.530×10^{-7}	1.650×10^{-7}
0.98	5.110×10^{-8}	4.260×10^{-8}	3.590×10^{-8}	3.780×10^{-8}

Table 5.8: Convergence rates for example 5.4 with different values of α when $N = Nt = 50$.

x	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 1.8$
0.10	2.0591	2.2555	2.3884	2.1976
0.30	2.0094	2.0823	2.2073	2.1587
0.50	2.0047	2.0439	2.1293	2.1302
0.70	2.0084	2.0285	2.0903	2.1096
0.90	2.0066	2.0204	2.0678	2.0948
0.98	2.0039	2.0182	2.0614	2.0925
Ord_∞	1.5247	1.5753	1.6783	1.6419

Example 5.5: We consider the linear fractional diffusion equation

$$\frac{\partial u}{\partial t} - \Gamma(4-\alpha)\ x^\alpha\ \frac{\partial^\alpha u}{\partial x^\alpha} + 7u = 2\alpha x^2 e^{-t}, \tag{5.73}$$

with initial condition

$$u(x,0) = x^2 - x^3, \tag{5.74}$$

and boundary conditions

$$\left.\begin{aligned} u_x(0,t) &= 0, \\ u_x(2,t) &= -8e^{-t}. \end{aligned}\right\} \tag{5.75}$$

The exact solution is $u(x,t) = (x^2 - x^3)e^{-t}$.

The discretized form of (5.73) is

$$\left[1 + \frac{7\Delta t}{2}\right] u_i^{j+1} - \frac{\Delta t}{2} m_i D_\alpha u_i^{j+1} = \left[1 - \frac{7\Delta t}{2}\right] u_i^j + \frac{\Delta t}{2} m_i D_\alpha u_i^j + \frac{\Delta t}{2}(r_i^{j+1} + r_i^j), \tag{5.76}$$

where $m(x) = \Gamma(4-\alpha)\ x^\alpha$ and $r(x,t) = 2\alpha x^2 e^{-t}$.

Figures 5.9 and 5.10 show the graph of the approximate solution superimposed on the mesh plot of the exact solution and the error respectively when $\alpha = 1.5$.

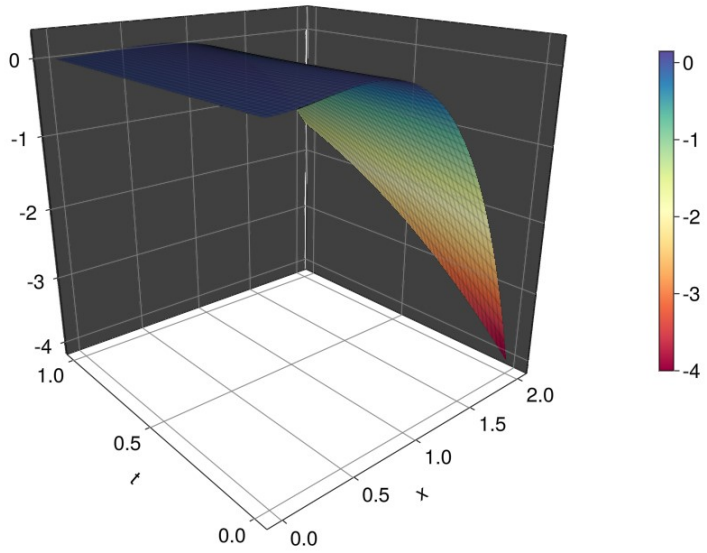

Figure 5.9: 3D plot of solution for example 5.5 at $N = Nt = 50,\ \alpha = 1.5$.

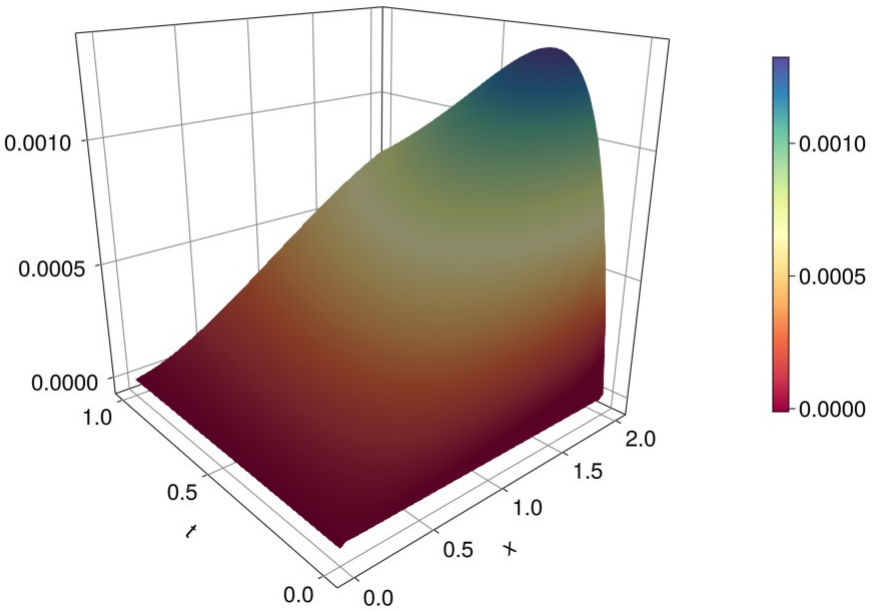

Figure 5.10: Error plot for example 5.5 at $N = Nt = 50,\ \alpha = 1.5$.

In Table 5.9, the absolute error at $t = 1$ is illustrated for different values of α. The convergence order in Table 5.10 is approximately $(3 - \alpha)$.

Table 5.9: Table of Absolute errors at for different values of α when $N = Nt = 100$.

x	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 1.8$
0.20	4.790×10^{-6}	7.910×10^{-6}	1.180×10^{-5}	1.400×10^{-5}
0.60	1.680×10^{-5}	3.270×10^{-5}	5.860×10^{-5}	9.150×10^{-5}
0.80	2.360×10^{-5}	4.700×10^{-5}	8.700×10^{-5}	1.440×10^{-4}
1.00	3.120×10^{-5}	6.190×10^{-5}	1.170×10^{-4}	2.010×10^{-4}
1.20	3.960×10^{-5}	7.720×10^{-5}	1.460×10^{-4}	2.570×10^{-4}
1.40	4.910×10^{-5}	9.260×10^{-5}	1.740×10^{-4}	3.110×10^{-4}
1.96	8.290×10^{-5}	1.350×10^{-4}	2.350×10^{-4}	4.120×10^{-4}

Table 5.10: Convergence rates for example 5.5 using different values of α.

x	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 1.8$
0.20	1.8995	1.6960	1.4959	1.1996
0.60	1.9066	1.7113	1.5117	1.2143
0.80	1.9118	1.7197	1.5195	1.2218
1.00	1.9180	1.7298	1.5289	1.2307
1.20	1.9249	1.7417	1.5402	1.2415
1.40	1.9323	1.7557	1.5539	1.2547
1.96	1.9542	1.8060	1.6097	1.3129
Ord_∞	1.9027	1.7797	1.6103	1.0966

Example 5.6: We take a nonlinear fractional partial differential equation

$$\frac{\partial u}{\partial t} - \Gamma(3-\alpha)\ x^{2+\alpha} t\ \frac{\partial^\alpha u}{\partial x^\alpha} + 2u^2 = x^2, \tag{5.77}$$

with initial condition

$$u(x,0) = 0, \tag{5.78}$$

and boundary conditions

$$\left.\begin{aligned} u(0,t) + u_x(0,t) &= 0, \\ u(1,t) + u_x(1,t) &= (2e^{-t}+1)t. \end{aligned}\right\} \tag{5.79}$$

The exact solution is $u(x,t) = tx^2$.

The linearized form of (5.77) is

$$(1+2u^j\Delta t)u_i^{j+1} - \frac{\Delta t}{2} m_i^{j+1} D_\alpha u_i^{j+1} = u_i^j + \frac{\Delta t}{2} m_i^j D_\alpha u_i^j + \Delta t\ r_i, \tag{5.80}$$

where $m(x,t) = \Gamma(3-\alpha)\ x^{2+\alpha}t$ and $r(x) = x^2$.

Figure 5.11 illustrates a 3D plot of the exact solution overlaid on the approximate solution. Figure 5.12 depicts the error.

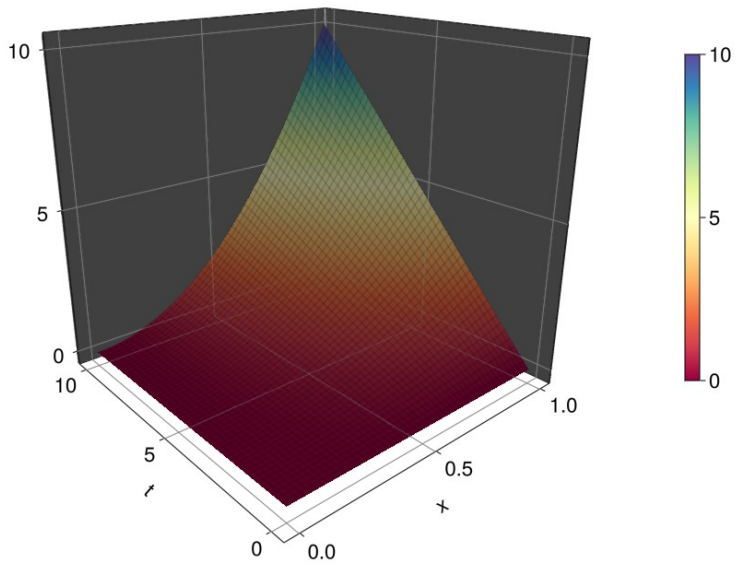

Figure 5.11: 3D plot of solution for example 5.6 at $N = Nt = 50,\ \alpha = 1.5$.

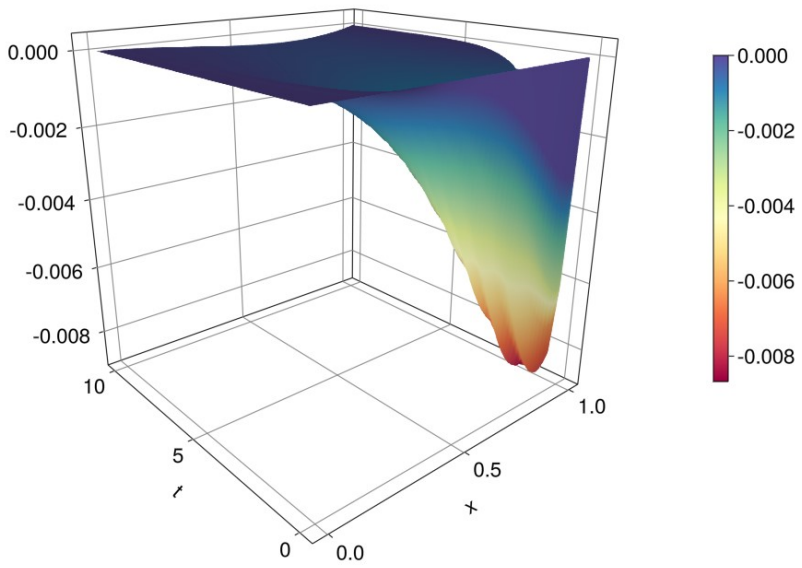

Figure 5.12: Error plot for example 5.6 at $N = Nt = 50,\ \alpha = 1.5$.

Table 5.11 gives the absolute error for various values of α and Table 5.12 gives the order of convergence.

Table 5.11: Table of Absolute errors at for different values of α when $N = Nt = 100$.

x	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 1.8$
0.10	7.540×10^{-6}	6.440×10^{-6}	5.810×10^{-6}	4.920×10^{-6}
0.30	4.020×10^{-5}	3.630×10^{-5}	3.230×10^{-5}	2.650×10^{-5}
0.50	8.730×10^{-5}	7.430×10^{-5}	6.300×10^{-5}	4.880×10^{-5}
0.70	1.120×10^{-4}	9.040×10^{-5}	7.380×10^{-5}	5.480×10^{-5}
0.90	6.670×10^{-5}	5.140×10^{-5}	4.060×10^{-5}	2.910×10^{-5}
0.98	1.630×10^{-5}	1.240×10^{-5}	9.640×10^{-6}	6.830×10^{-6}

Table 5.12: Convergence rates for example 5.6 with different values of α when $N = Nt = 50$.

x	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 1.8$
0.10	2.0060	2.0247	2.0043	1.9986
0.30	1.9991	1.9991	1.9989	1.9989
0.50	2.0002	1.9993	1.9992	1.9991
0.70	1.9978	1.9991	1.9992	1.9994
0.90	1.9972	2.0002	1.9981	2.0017
0.98	1.9973	2.0017	2.0018	1.9976
Ord_∞	1.2139	1.4982	1.4987	1.4945

Example 5.7: We consider the Fisher's equation

$$\frac{\partial u(x,t)}{\partial t} - \frac{1}{10}\,\frac{\partial^\alpha u(x,t)}{\partial x^\alpha} - \frac{1}{4}u(x,t) + \frac{1}{4}u^2(x,t) = 0, \tag{5.81}$$

which does not have an exact solution with the initial condition

$$u(x,0) = e^{-10|x|}, \tag{5.82}$$

and boundary conditions

$$\left.\begin{aligned} u_x(0,t) &= 0, \\ u_x(1,t) &= 0. \end{aligned}\right\} \tag{5.83}$$

We make use of the Crank-Nicholson method and linearize equation (5.81). We have

$$\left[1 + \frac{\Delta t}{4}u_i^j - \frac{\Delta t}{8}\right] u_i^{j+1} - \frac{\Delta t}{2} d D_\alpha u_i^{j+1} = \left[1 + \frac{\Delta t}{8}\right] u_i^j + \frac{\Delta t}{2} d D_\alpha u_i^j, \tag{5.84}$$

where $d = \frac{1}{10}$.

Figure 5.13 shows the approximate solution. Figure 5.14 shows the final profiles for various values of α.

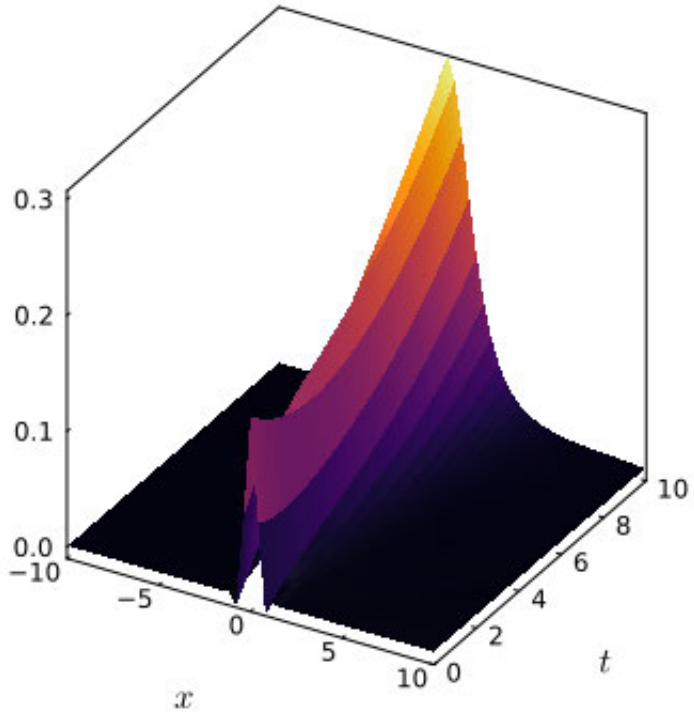

Figure 5.13: 3D plot of the approximate solution to example 5.7.

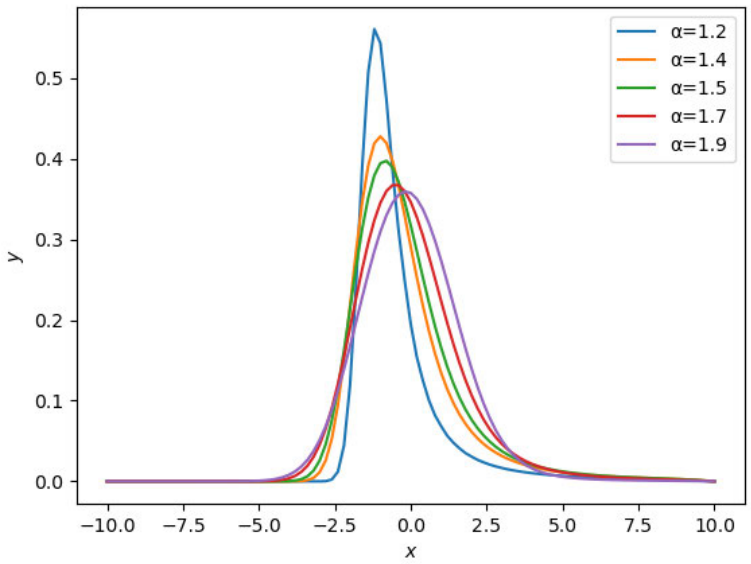

Figure 5.14: Final profiles for example 5.7 at different values of α

5.6 Discussion of Chapter 5

We have used the quadratic B-spline OCFE to solve fractional partial differential equations. We solved various diffusion type equations including Fisher's equation as case studies to illustrate our method. A complete stability and convergence analysis was carried out. It was shown that the method is unconditionally stable and the spatial order of convergence was $3 - \alpha$. The numerical results validate the theoretical results. In the next chapter we extend quadratic B-spline to solve two-dimensional partial differential equations.

CHAPTER SIX

APPLICATION OF OCFE TO TWO-DIMENSIONAL PARTIAL DIFFERENTIAL EQUATIONS

In this chapter, we show the versatility of the quadratic B-spline OCFE in solving two-dimensional partial differential equations. We used a modified form of the quadratic B-spline basis functions as trial functions. We also show that the convergence order for our numerical results is 2. Our results are presented graphically and in tabular form to compare with existing work. We find that our method is better than the existing ones in the literature.

6.1 Derivation of 2-D quadratic B-spline function

Suppose the B-spline numerical solution of the second order ordinary differential equation

$$b_2\frac{d^2u}{dx^2} + b_1\frac{du}{dx} + b_0u = g(x) \tag{6.1}$$

with boundary conditions

$$u(a) = \alpha, \quad u(b) = \beta, \tag{6.2}$$

in the interval $a \leq x \leq b$ is $u_c(x)$ and the analytical solution be represented by $u(x)$. We define the modified B-spline basis functions in $[0, 1]$ as $\mathcal{B}_k(z), k = 1, 2, 3,$

$$\begin{aligned} \mathcal{B}_1(z) &= (1-z)^2, \\ \mathcal{B}_2(z) &= 1 + 2z(1-z), \\ \mathcal{B}_3(z) &= z^2. \end{aligned} \tag{6.3}$$

Using the continuity conditions in (2.12) and (2.13), the collocation solution $u_c(z)$ in an interval $[x_i, x_{i+1}]$ can be written in the following alternate form

$$u_c^i(z) = \sum_{k=1}^{3} a_{k+i-1}\mathcal{B}_k(z). \tag{6.4}$$

The coefficients $a_{k+i-1},\ k = 1, 2, 3,\ i = 1, 2, \ldots, N$, form a regular pattern across the intervals. In the first interval, the coefficients are $a_1,\ a_2,\ a_3$. In the second interval, they are $a_2,\ a_3,\ a_4$ and $a_3,\ a_4,\ a_5$ in the third interval e.t.c.

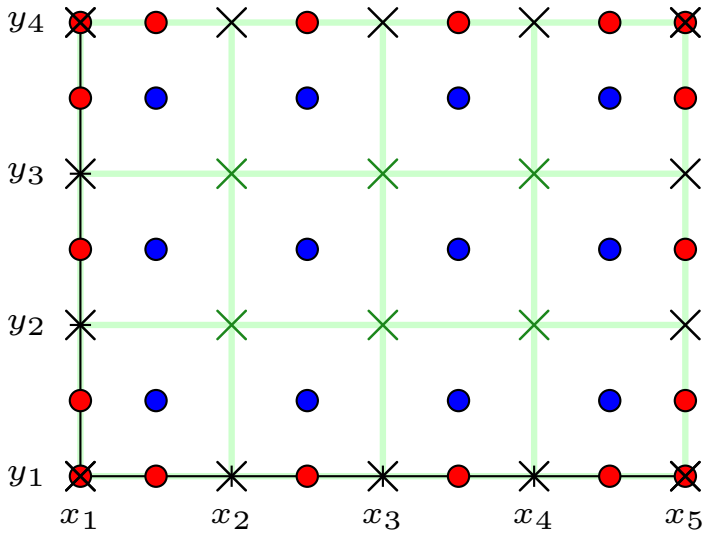

Figure 6.1: Grid for $N = 4,\ M = 3$.

The blue and red dots in Figure 6.1 represent the collocation and boundary points respectively. The black crosses represent the boundary node while the green ones represent internal nodes where discrete errors are evaluated. Let $u(x, y)$ be a function of two variables in x and y on the grid in Figure 6.1 such that $x \in [a, b]$ and $y \in [c, d]$. Consider the partial differential equation

$$\begin{aligned} q_5(x,y)\frac{\partial^2 u}{\partial x^2} + q_4(x,y)\ \frac{\partial^2 u}{\partial y^2} + q_3(x,y)\ \frac{\partial^2 u}{\partial x \partial y} + q_2(x,y)\ \frac{\partial u}{\partial x} \\ + q_1(x,y)\ \frac{\partial u}{\partial y} + q_0(x,y)\ u(x,y) = g(x,y) \end{aligned} \tag{6.5}$$

where $q_0(x, y),\ q_1(x, y),\ q_2(x, y),\ q_3(x, y),\ q_4(x, y)$ and $q_5(x, y)$ are coefficient func-

tions, and boundary conditions

$$u(a,y)=\alpha_1(y),\quad u(b,y)=\beta_1(y),\quad u(x,c)=\alpha_2(x),\quad u(x,d)=\beta_2(x). \tag{6.6}$$

Define the collocation solution in the rectangle (i,j) to be of the form

$$u_c^{i,j}(x,y)=\sum_{p=1}^{3}\sum_{k=1}^{3}a_{kp}^{ij}\mathcal{B}_k(x)\mathcal{B}_p(y), \tag{6.7}$$

We discretize in the y variable as $c=y_1<y_2<\ldots<y_{M+1}=d$. Each rectangle $[x_i,x_{i+1}]\times[y_j,y_{j+1}]$ is mapped to $[0,1]\times[0,1]$ by

$$z_1=\frac{x-x_i}{h_1}\quad\text{and}\quad z_2=\frac{y-y_j}{h_2}, \tag{6.8}$$

where h_1 and h_2 are the uniform spacing in the x and y directions, respectively. We can write the collocation solution (6.7) as

$$u_c^{i,j}(z_1,z_2)=\sum_{k=1}^{3}\sum_{p=1}^{3}a_{k+i-1}^{p+j-1}\ \mathcal{B}_k(z_1)\ \mathcal{B}_p(z_2), \tag{6.9}$$

where a_{k+i-1}^{p+j-1} are the coefficients to be determined. We note that our modification to the basis functions used in [52] ensures that continuity is automatically embedded in the collocation solution and avoid difficulties in indexing the coefficients a_{k+i-1}^{p+j-1}. Hence the computational cost is significantly reduced since the number of unknowns is $(M+2)(N+2)$. Therefore equation (6.5) can be re-written as

$$\begin{aligned}
&\frac{q_5}{h_1^2}\ \sum_{k=1}^{3}\sum_{p=1}^{3}a_{k+i-1}^{p+j-1}\ \mathcal{B}_k''(z_1)\ \mathcal{B}_p(z_2)+\frac{q_4}{h_2^2}\ \sum_{k=1}^{3}\sum_{p=1}^{3}a_{k+i-1}^{p+j-1}\ B_k(z_1)\ \mathcal{B}_p''(z_2)\\
&+\frac{q_3}{h_1h_2}\ \sum_{k=1}^{3}\sum_{p=1}^{3}a_{k+i-1}^{p+j-1}\ \mathcal{B}_k'(z_1)\ \mathcal{B}_p'(z_2)+\frac{q_2}{h_1}\ \sum_{k=1}^{3}\sum_{p=1}^{3}a_{k+i-1}^{p+j-1}\ \mathcal{B}_k'(z_1)\ \mathcal{B}_p(z_1)\\
&+\frac{q_1}{h_2}\ \sum_{k=1}^{3}\sum_{p=1}^{3}a_{k+i-1}^{p+j-1}\ \mathcal{B}_k(z_1)\ \mathcal{B}_p'(z_2)+q_0\ \sum_{k=1}^{3}\sum_{p=1}^{3}a_{k+i-1}^{p+j-1}\ \mathcal{B}_k(z_1)\ \mathcal{B}_p(z_2)\\
&=g(x_i+h_1z_1,y_j+h_2z_2),\quad q_r=q_r(x_i+h_1z_1,y_j+h_2z_2),\ r=0,1,2,\ldots,5
\end{aligned} \tag{6.10}$$

and the boundary conditions are transformed into

$$\left.\begin{aligned}
&\sum_{k=1}^{3}\sum_{p=1}^{3} a_k^{p+j-1}\ \mathcal{B}_k(0)\ \mathcal{B}_p(z_2) = \alpha_{1,j}(y+h_2z_2),\\
&\sum_{k=1}^{3}\sum_{p=1}^{3} a_{k+N-1}^{p+j-1}\ \mathcal{B}_k(1)\ \mathcal{B}_p(z_2) = \beta_{1,j}(y+h_2z_2),\\
&\sum_{k=1}^{3}\sum_{p=1}^{3} a_{k+i-1}^{p}\ \mathcal{B}_k(z_1)\ \mathcal{B}_p(0), = \alpha_{2,i}(x+h_1z_1),\\
&\sum_{k=1}^{3}\sum_{p=1}^{3} a_{k+i-1}^{p+M-1}\ \mathcal{B}_k(z_1)\ \mathcal{B}_p(1) = \beta_{2,i}(x+h_1z_1).
\end{aligned}\right\} \tag{6.11}$$

Equation (6.10) is then evaluated at the collocation points $(0.5, 0.5)$ coupled with the boundary conditions to get a linear system of $(M+2)(N+2)$ equations in $(M+2)(N+2)$ unknowns. The solution to the differential equation is obtained by substituting the coefficients a_{k+i-1}^{p+j-1} back into equation (6.9).

6.2 Convergence of the Method

The order of convergence is checked numerically using $\log_2\left[\frac{||\mathbf{e}||_\infty^{[h_1,h_2]}}{||\mathbf{e}||_\infty^{[\frac{h_1}{2},\frac{h_2}{2}]}}\right]$ where the components of $\mathbf{e}$ are given by

$$e_{i,j} = u_{i,j} - U_{i,j},\ i = 1, 2, \ldots, N,\ j = 1, 2, \ldots, M,$$

$u_{i,j}$ and $U_{i,j}$ are exact and approximate solutions at the nodes, respectively, and

$$||e||_\infty^{[h_1,h_2]} = \max_{\substack{i=1,2,\ldots,N\\ j=1,2,\ldots,M}} |e_{i,j}|, \tag{6.12}$$

using h_1 for spacing in x direction and h_2 for spacing in y direction. The nodes are the vertices of the grid.

6.3 Numerical examples

In this section, we illustrate our method with examples on partial differential equations in two spatial variables. We shall take $\Delta x = h_1$ and $\Delta y = h_2$ along x and y directions, respectively.

Example 6.1: We consider the partial differential equation [57]

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = g(x, y), \tag{6.13}$$

with the boundary conditions

$$u(0, y) = 0, \quad u(1, y) = 0, \quad u(x, 0) = 0, \quad u(x, 1) = 0, \tag{6.14}$$

where the source term

$$g(x, y) = 3e^{2x+3y} \left(x^2(18y^2 - 4y - 5) + x(5 - 8y^2 - 6y) - (3y^2 - 3y) \right).$$

The exact solution to this problem is

$$u(x, y) = 3xy\ e^{2x+3y}(1 - x)(1 - y). \tag{6.15}$$

We obtain the discretization of equation (6.13) when $q_5 = q_4 = q_2 = q_1 = 1$, $q_3 = q_0 = 0$ in (6.10) and that of the boundary conditions (6.14) when $\alpha_1 = \beta_1 = \alpha_2 = \beta_2 = 0$ in equation (6.11). The source term $g(x, y)$ is evaluated at points $(x_i + 0.5h_1, y_j + 0.5h_2)$, $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M$.

Figure 6.2 shows a 3D surface plot of the exact solution (shaded) and approximate solution (mesh) superimposed on each other. In Figure 6.3, the contour curves for the approximate solution (coloured dots) are overlaid on that of the exact solution (solid curves). It shows that both solutions match very well at all points in the $x - y$ plane. In Figure 6.4, the error is presented.

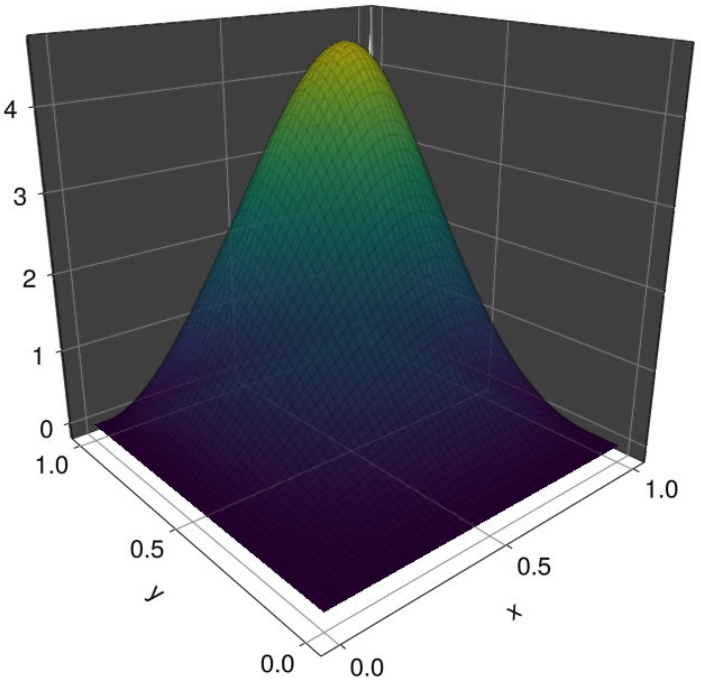

Figure 6.2: 3D plot of the approximate solution when $\Delta x = \Delta y = 0.02$.

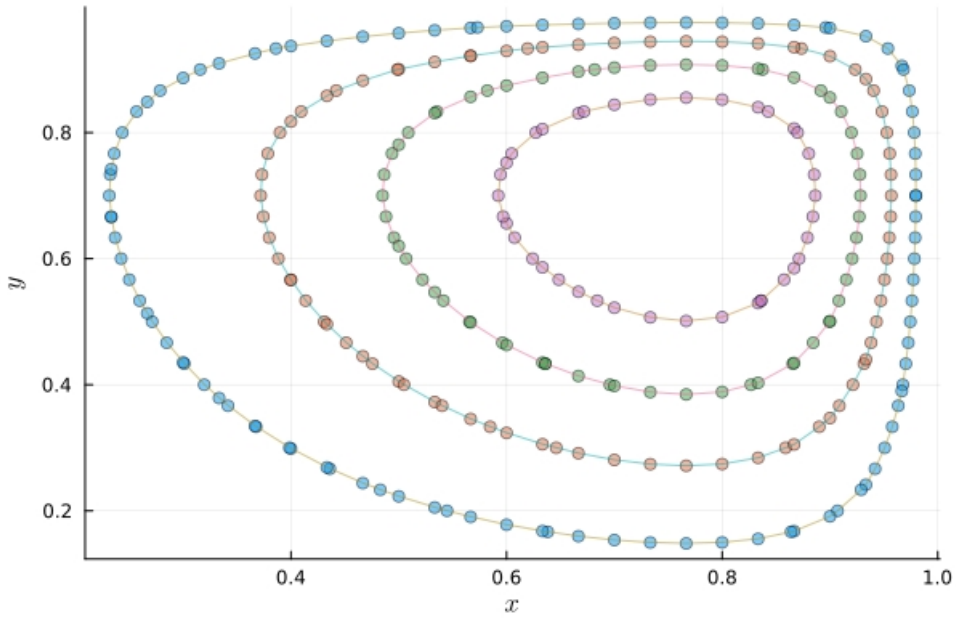

Figure 6.3: Contour plots for the exact and approximate solutions when $\Delta x = \Delta y = 0.03$.

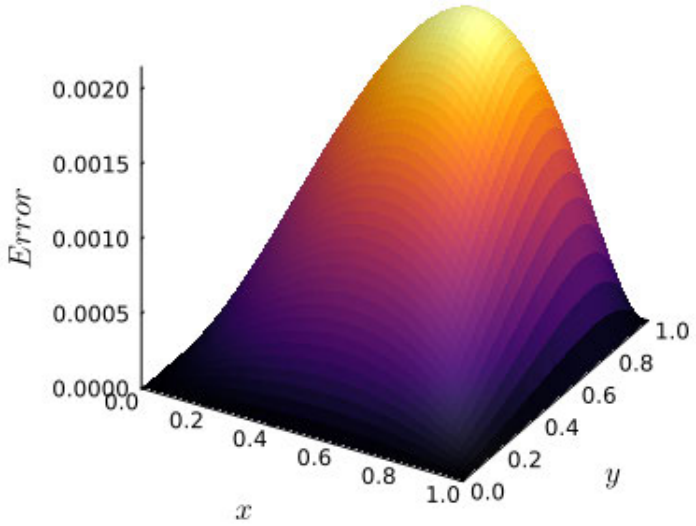

Figure 6.4: 3D plot of the error when $\Delta x = \Delta y = 0.02$.

In Table 6.1, we compared our results with that of [57] and show that our method gives less absolute error. Hence the present work is better than that of [57]. Table 6.2 shows that the convergence rate is 2 and Table 6.3depicts the computational times in Julia v1.11 [11].

Table 6.1: Comparison of Absolute errors with that of [57] when $y = 0.5$, $\Delta x = \Delta y = 0.02$.

x	Present	[57]
0.1	0.000482	0.001070
0.3	0.001269	0.003409
0.4	0.001550	0.004316
0.5	0.001727	0.005043
0.6	0.001766	0.005607
0.8	0.001299	0.006468
0.9	0.000750	0.006931

Table 6.2: Convergence rates for example 6.1 when $\Delta x = \Delta y = 0.02$.

	x								
y	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	1.9993	1.9993	1.9994	1.9994	1.9994	1.9994	1.9994	1.9993	1.9992
0.2	1.9993	1.9994	1.9994	1.9994	1.9995	1.9995	1.9995	1.9995	1.9994
0.3	1.9993	1.9994	1.9994	1.9995	1.9995	1.9996	1.9997	1.9997	1.9997
0.4	1.9993	1.9994	1.9994	1.9995	1.9996	1.9997	1.9998	1.9999	2.0000
0.5	1.9993	1.9994	1.9994	1.9996	1.9997	1.9998	2.0000	2.0002	2.0004
0.6	1.9992	1.9993	1.9994	1.9996	1.9998	2.0000	2.0002	2.0005	2.0008
0.7	1.9991	1.9993	1.9994	1.9996	1.9998	2.0001	2.0004	2.0009	2.0014
0.8	1.9990	1.9992	1.9994	1.9996	1.9999	2.0002	2.0007	2.0013	2.0021
0.9	1.9988	1.9990	1.9993	1.9996	1.9999	2.0003	2.0009	2.0018	2.0032

Table 6.3: CPU computation times for example 6.1 in seconds

N	CPU time (s)
10	0.194180
20	0.238544
30	0.237030
40	0.369534
50	0.852913

Example 6.2: We consider the Poisson equation (example 4 of [57]),

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = g(x, y), \tag{6.16}$$

with the boundary conditions

$$u(0, y) = 0, \quad u(1, y) = 0, \quad u(x, 0) = 0, \quad u(x, 1) = 0, \tag{6.17}$$

where the source term $g(x, y) = 0.5xy(xy + x + y - 3)e^{(x+y+0.5)}$.

The exact solution to this problem is

$$u(x, y) = 0.25xy(1 - x)(1 - y)e^{(x+y+0.5)}. \tag{6.18}$$

We obtain the discretization of equation (6.16) when $q_0 = q_1 = q_2 = q_3 = 0$, $q_4 = 1$ and $q_5 = 1$ in (6.10). The boundary conditions and the source term are discretized in a similar way as in example 6.1.

Figure 6.5 shows the mesh plot for the approximate solution together with the 3D plot of the exact solution. In Figure 6.6, the contour curves for the approximate solution (coloured dots) are overlaid on that of the exact solution (solid curves). The figure shows that both solutions match very well. Figure 6.7 displays the 3D plot of the error.

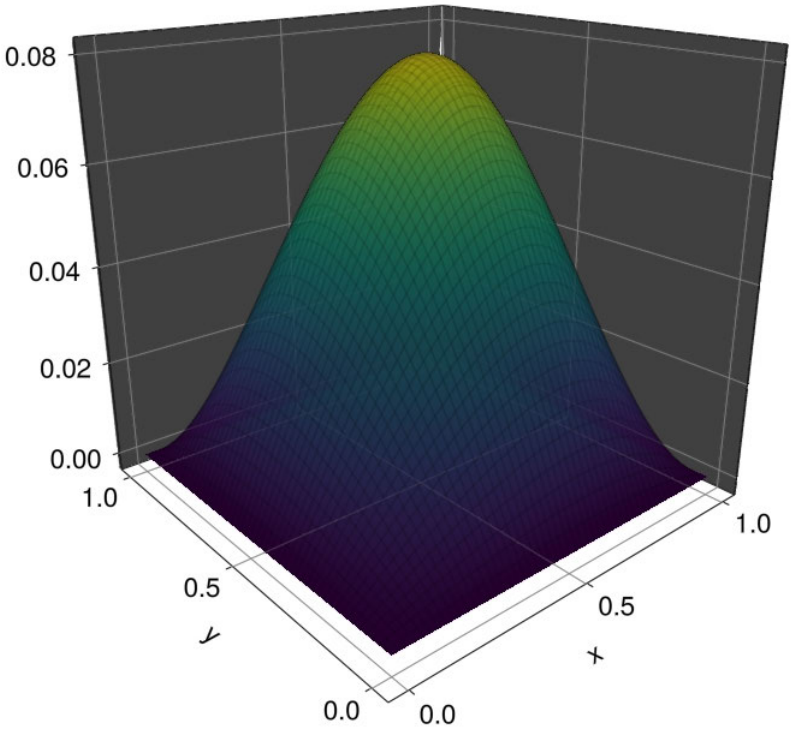

Figure 6.5: 3D plot of the approximate solution to example 6.2 when $\Delta x = \Delta y = 0.02$.

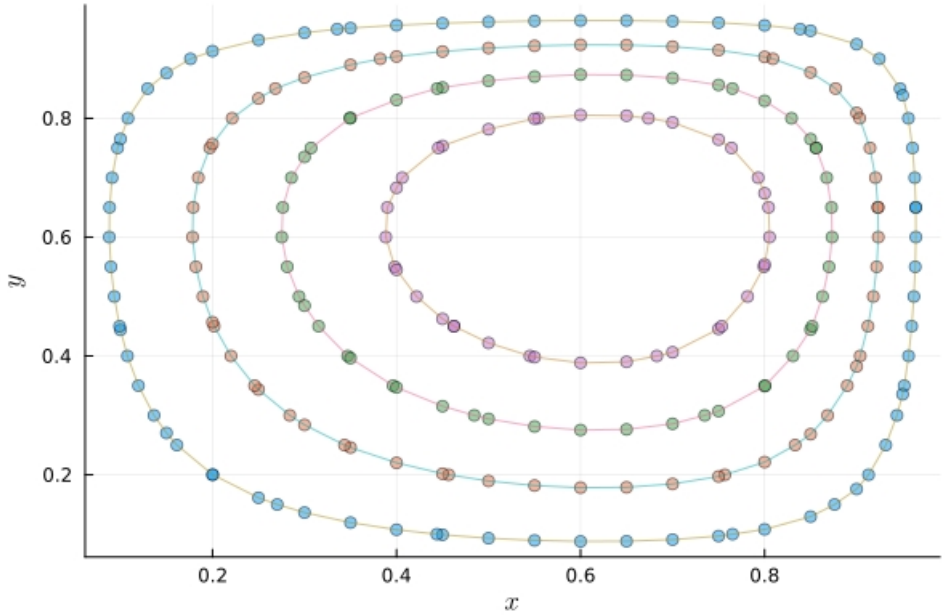

Figure 6.6: Contour plots for the exact and approximate solutions to example 6.2 when $\Delta x = \Delta y = 0.05$.

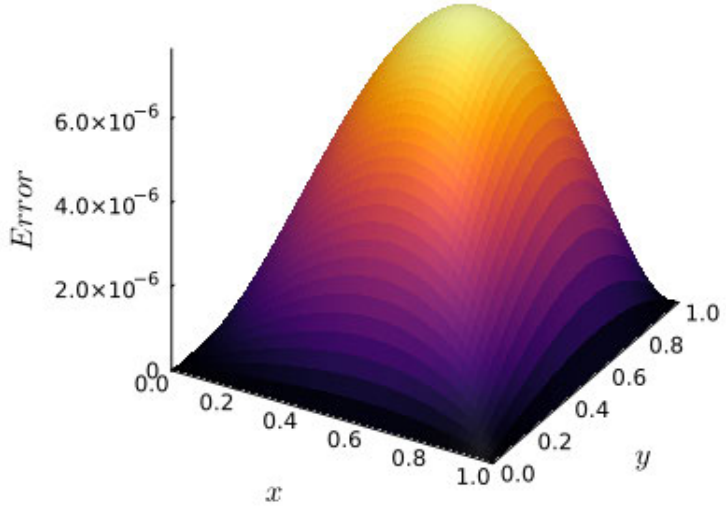

Figure 6.7: 3D plot of the error for example 6.2 when $\Delta x = \Delta y = 0.02$.

Table 6.4 shows the computation times for different number of number of partitions of the computational domain. Time increases as N increases. Table 6.5 shows that the convergence rate is approximately 2.

Table 6.4: CPU computation times for example 6.2 in seconds

N	CPU time (s)
10	0.163356
20	0.199027
30	0.240362
40	0.299560
50	0.599808

Table 6.5: Convergence rates for example 6.2 when $\Delta x = \Delta y = 0.02$.

	x								
y	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	2.0005	2.0005	2.0005	2.0006	2.0006	2.0006	2.0007	2.0007	2.0008
0.2	2.0005	2.0005	2.0005	2.0006	2.0006	2.0006	2.0007	2.0007	2.0008
0.3	2.0005	2.0005	2.0006	2.0006	2.0006	2.0006	2.0007	2.0007	2.0008
0.4	2.0006	2.0006	2.0006	2.0006	2.0006	2.0007	2.0007	2.0007	2.0008
0.5	2.0006	2.0006	2.0006	2.0006	2.0007	2.0007	2.0007	2.0008	2.0009
0.6	2.0006	2.0006	2.0006	2.0007	2.0007	2.0007	2.0008	2.0009	2.0010
0.7	2.0007	2.0007	2.0007	2.0007	2.0007	2.0008	2.0009	2.0010	2.0011
0.8	2.0007	2.0007	2.0007	2.0007	2.0008	2.0009	2.0010	2.0011	2.0013
0.9	2.0008	2.0008	2.0008	2.0008	2.0009	2.0010	2.0011	2.0013	2.0017

Example 6.3: In this case we consider the equation [43]

$$\frac{\partial u}{\partial t} - (x^2+1)\,\frac{\partial^2 u}{\partial x^2} - (y^2+1)\frac{\partial^2 u}{\partial y^2} - y\,\frac{\partial u}{\partial y} - (x+y)\,u(x,y) = g(x,y,t), \quad (6.19)$$

with the initial condition

$$u(x,y,0) = 2\sin(\pi x)\sin(\pi y) \quad (6.20)$$

and boundary conditions

$$u(0,y,t) = 0, \quad u(1,y,t) = 0, \quad u(x,0,t) = 0, \quad u(x,1,t) = 0. \quad (6.21)$$

The exact solution is

$$u(x,y,t) = (1+e^{-t})\sin(\pi x)\sin(\pi y). \quad (6.22)$$

The function $g(x, y, t)$ can be obtained by substituting the exact solution into equation (6.19). We apply the Crank-Nicolson technique to discretize equation (6.19). This gives

$$
\begin{aligned}
&\left[1 - \frac{\Delta t}{2}(x_i + h_1 z_1 + y_j + h_2 z_2)\right] \sum_{k=1}^{3}\sum_{p=1}^{3} a_{k+i-1}^{p+j-1,n+1}\ \mathcal{B}_k(z_1)\ \mathcal{B}_p(z_2) \\
&- \frac{\Delta t}{2h_2}(y_j + h_2 z_2) \sum_{k=1}^{3}\sum_{p=1}^{3} a_{k+i-1}^{p+j-1,n+1}\ \mathcal{B}_k(z_1)\ \mathcal{B}'_p(z_2) \\
&- \frac{\Delta t}{2h_1^2}((x_i + h_1 z_1)^2 + 1) \sum_{k=1}^{3}\sum_{p=1}^{3} a_{k+i-1}^{p+j-1,n+1}\ \mathcal{B}''_k(z_1)\ \mathcal{B}_p(z_2) \\
&- \frac{\Delta t}{2h_2^2}((y_j + h_2 z_2)^2 + 1) \sum_{k=1}^{3}\sum_{p=1}^{3} a_{k+i-1}^{p+j-1,n+1}\ \mathcal{B}_k(z_1)\ \mathcal{B}''_p(z_2) \\
&= \left[1 + \frac{\Delta t}{2}(x_i + h_1 z_1 + y_j + h_2 z_2)\right] \sum_{k=1}^{3}\sum_{p=1}^{3} a_{k+i-1}^{p+j-1,n}\ \mathcal{B}_k(z_1)\ \mathcal{B}_p(z_2) \\
&+ \frac{\Delta t}{2h_2}(y_j + h_2 z_2) \sum_{k=1}^{3}\sum_{p=1}^{3} a_{k+i-1}^{p+j-1,n}\ \mathcal{B}_k(z_1)\ \mathcal{B}'_p(z_2) \\
&+ \frac{\Delta t}{2h_1^2}((x_i + h_1 z_1)^2 + 1) \sum_{k=1}^{3}\sum_{p=1}^{3} a_{k+i-1}^{p+j-1,n}\ \mathcal{B}''_k(z_1)\ \mathcal{B}_p(z_2) \\
&+ \frac{\Delta t}{2h_2^2}((y_j + h_2 z_2)^2 + 1) \sum_{k=1}^{3}\sum_{p=1}^{3} a_{k+i-1}^{p+j-1,n}\ \mathcal{B}_k(z_1)\ \mathcal{B}''_p(z_2) \\
&+ \frac{\Delta t}{2}[g(x_i + h_1 z_1, y_j + h_2 z_2, t_n) + g(x_i + h_1 z_1, y_j + h_2 z_2, t_{n+1})], \\
&i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, M.
\end{aligned}
\tag{6.23}
$$

The boundary conditions in (6.21) and the source term $g(x, y)$ are treated in a similar way as in example 6.1.

The approximate and the exact solutions are displayed together as a mesh and 3D surface respectively in Figure 6.8. In Figure 6.9, we show that both solutions match very well on the $x-y$ plane with coloured dots representing the approximate solution and solid curves representing the exact solution. Figure 6.10 shows the 3D plot of the error. In Table 6.6 the convergence rate is 2. Furthermore, Table 6.7 shows that the time-dependent two-dimensional partial differential equation takes

longer time than the time-independent ones.

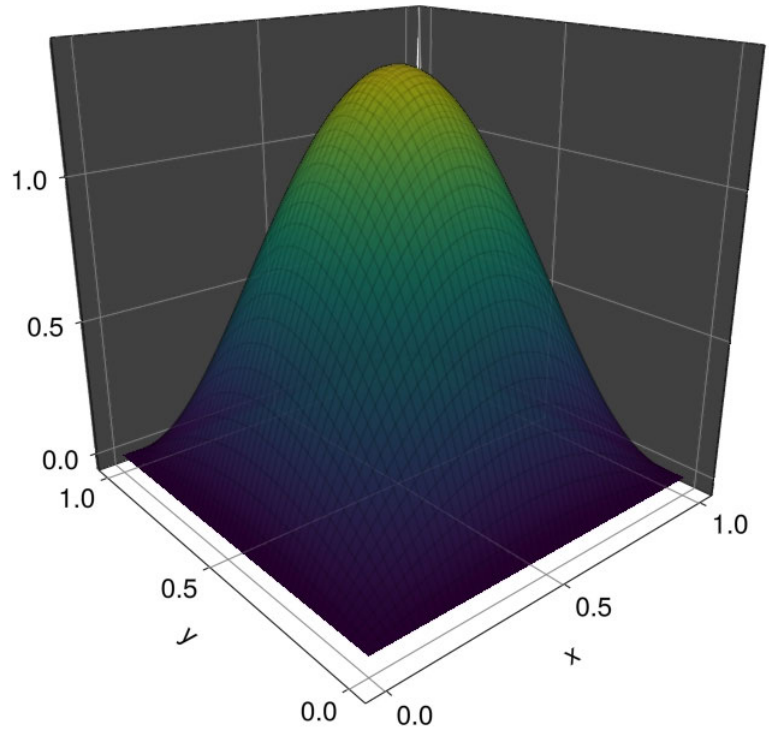

Figure 6.8: 3D plot of the approximate solution to example 6.3 at $t = 1$ when $\Delta x = \Delta y = \Delta t = 0.02$.

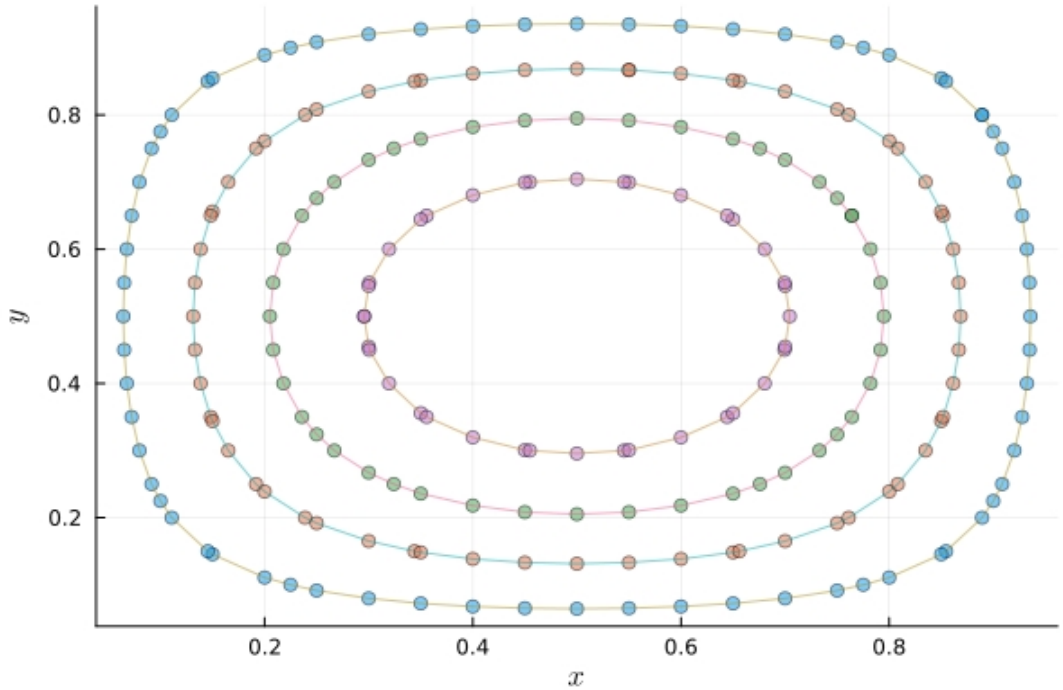

Figure 6.9: Contour plots for example 6.3 at $t = 0.5$ when $\Delta t = 0.02$, $\Delta x = \Delta y = 0.05$.

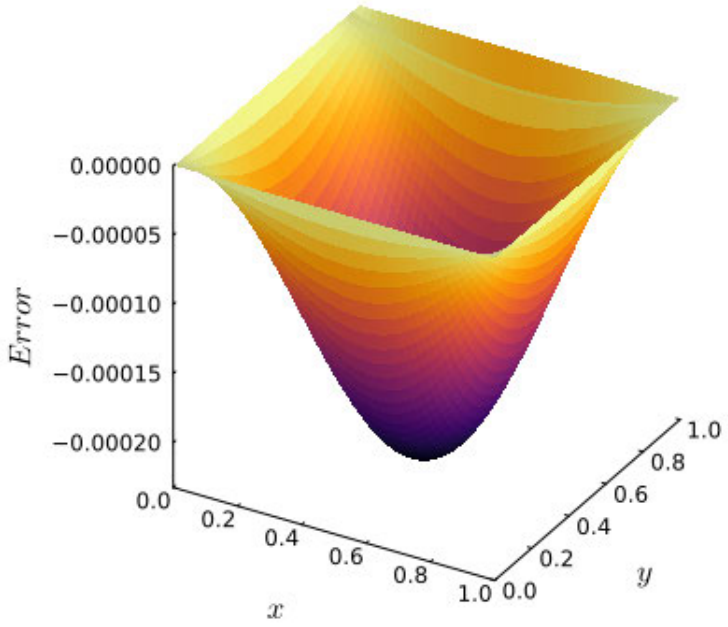

Figure 6.10: 3D plot of error for example 6.3 at $t = 1$ when $\Delta x = \Delta y = 0.02$.

Table 6.6: Convergence rate at $t = 1$ when $\Delta x = \Delta y = \Delta t = 0.025$.

	x								
y	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	1.9911	1.9912	1.9913	1.9914	1.9916	1.9918	1.9919	1.9920	1.9921
0.3	1.9912	1.9913	1.9914	1.9916	1.9917	1.9919	1.9920	1.9922	1.9923
0.4	1.9913	1.9914	1.9915	1.9916	1.9918	1.9920	1.9921	1.9923	1.9924
0.5	1.9914	1.9915	1.9916	1.9917	1.9919	1.9921	1.9922	1.9924	1.9925
0.6	1.9916	1.9916	1.9917	1.9919	1.9920	1.9922	1.9923	1.9925	1.9926
0.8	1.9917	1.9917	1.9919	1.9920	1.9921	1.9923	1.9925	1.9926	1.9927
0.9	1.9917	1.9917	1.9918	1.9920	1.9921	1.9923	1.9924	1.9926	1.9927

Table 6.7: CPU computation times for example 6.3 in seconds

N	CPU Time (s)
10	1.335024
20	1.514066
30	2.314410
40	5.632195
50	14.934014

6.4 Discussion of Chapter 6

We have shown that partial differential equations involving two spatial variables, and ones that also include a temporal variable can be solved efficiently using the modified quadratic B-spline OCFE. This simplified version of B-spline basis function has the advantage that continuity is built in and generates a smaller linear system. Hence it makes our work easier and computationally faster than the original B-spline basis functions, We also showed that our results compared favourably with exact solutions and existing work in the literature. In the next chapter, we discuss cubic B-spline OCFE and it's application to differential equations.

CHAPTER SEVEN

CUBIC B-SPLINES ORTHOGONAL COLLOCATION ON FINITE ELEMENTS

In this chapter, we present the cubic B-splines OCFE method and demonstrate it's efficiency for solving third-order PDEs. We consider the KdV-Burgers' equation, the Burgers' equation using Gauss' points, cases of KdV equation with exact solution and without a known solution, and fractional differential equations. Various numerical experiments are performed to demonstrate the suitability of the cubic B-splines OCFE method.

7.1 Cubic B-spline basis

A spline of order $k = 4$, which is a linear combination of third-degree polynomial bases, is called a cubic B-spline. Hence following equation (2.4), the required cubic B-spline basis on $[0, 1]$ are

$$B_1 = (1-z)^3, \quad B_2 = 3z(1-z)^2, \quad B_3 = 3z^2(1-z) \quad \text{and} \quad B_4 = z^3. \tag{7.1}$$

The cubic B-spline basis functions on $[0, 1]$ are graphically shown in Figure 7.1.

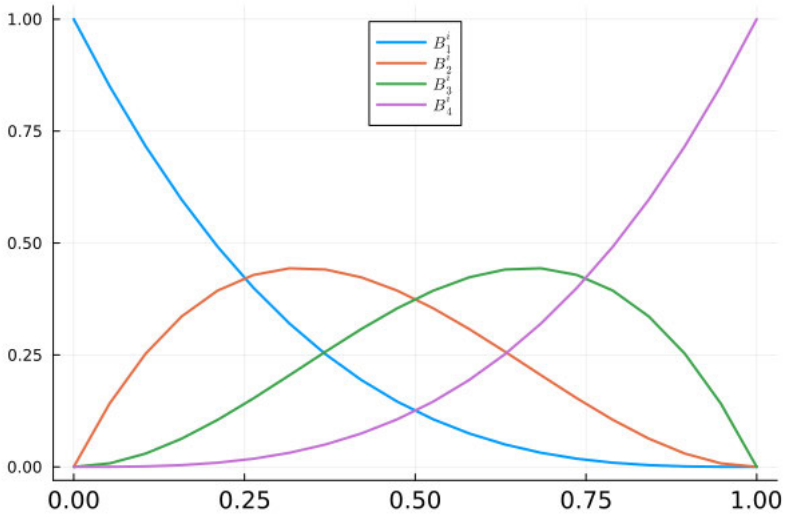

Figure 7.1: Cubic B-spline basis functions on $[0, 1]$.

Figures 7.2, 7.3 and 7.4 show the continuities of the cubic B-spline basis functions, their first and second derivatives across the boundaries x_{i+1} to x_{i+2}, respectively.

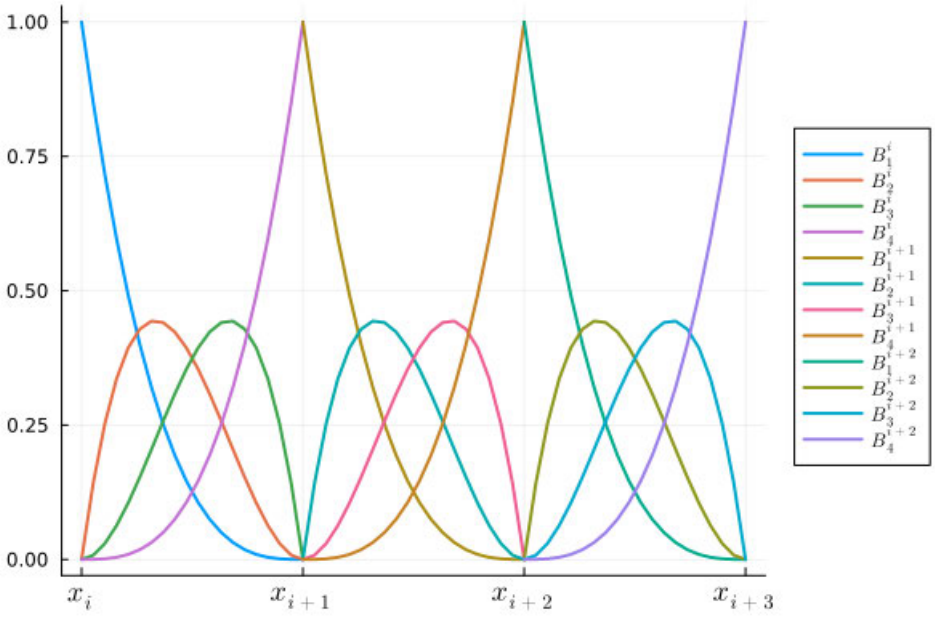

Figure 7.2: Continuity at the boundaries.

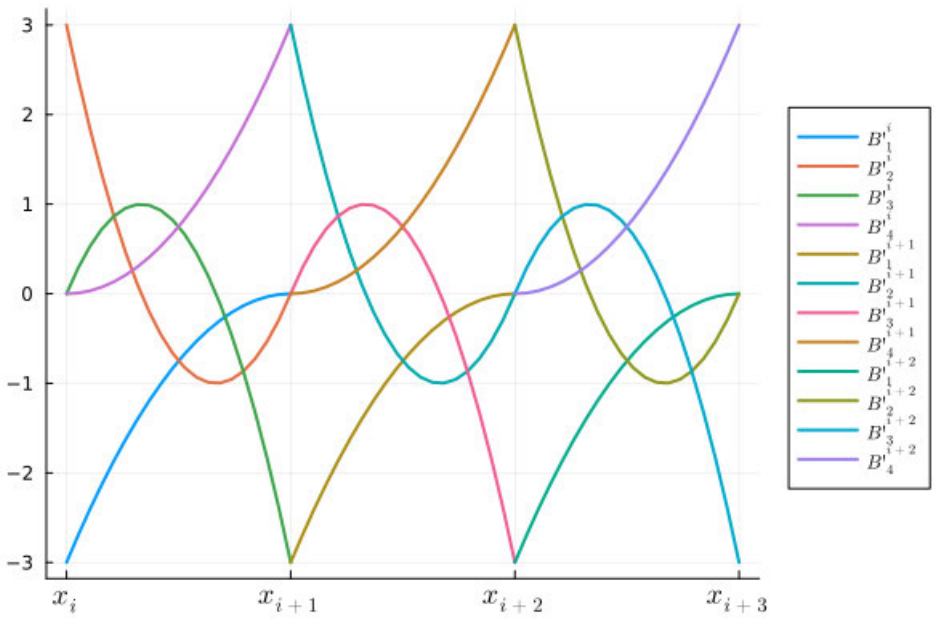

Figure 7.3: First derivative continuity at the boundaries.

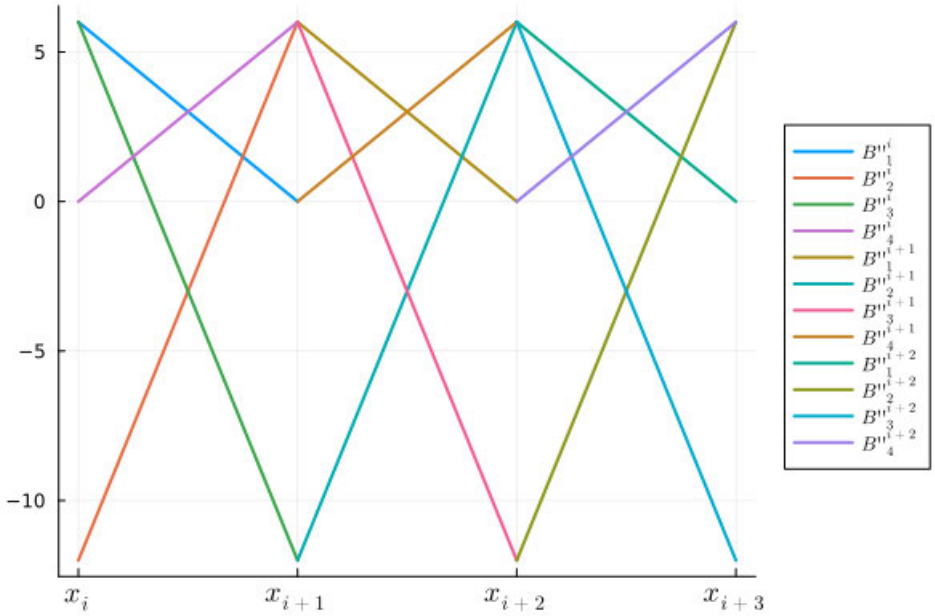

Figure 7.4: Second derivative continuity at the boundaries.

7.2 Derivation of the cubic B-spline OCFE method

Consider solving a third order linear differential equation in one spatial variable x on (a, b)

$$Y'''(x) + \lambda Y''(x) + \omega Y'(x) + \eta Y(x) = f(x), \tag{7.2}$$

subject to the boundary conditions

$$Y(a) = \alpha, \quad Y'(a) = \gamma, \quad Y(b) = \beta. \tag{7.3}$$

Suppose the interval $[a, b]$ is partitioned into N finite elements with nodes given by $x_i = a + (i-1)h$, $i = 1, 2, ...N+1$, where h is the uniform spacing. For every $x \in [x_i, x_{i+1}]$, we transform x into $z \in [0, 1]$ via

$$z = \frac{x - x_i}{x_{i+1} - x_i} = \frac{x - x_i}{h}, \tag{7.4}$$

such that the collocation solution denoted by Y^i in the i^{th} element is given by

$$Y^i(x) = Y^i(z) = \sum_{j=1}^{4} b_j B_j^i(z), \quad i = 1, 2, \ldots, N+1. \tag{7.5}$$

We now require continuity at the internal boundaries such that

$$Y^i(x_{i+1}) = Y^{i+1}(x_{i+1}), \tag{7.6}$$

$$\left.\frac{dY^i}{dx}\right|_{x_{i+1}} = \left.\frac{Y^{i+1}}{dx}\right|_{x_{i+1}}, \tag{7.7}$$

$$\left.\frac{d^2Y^i}{dx^2}\right|_{x_{i+1}} = \left.\frac{d^2Y^{i+1}}{dx^2}\right|_{x_{i+1}}. \tag{7.8}$$

Equations (7.6)–(7.8) lead to

$$b_{4j-4} = b_{4j-3}, \tag{7.9}$$

$$b_{3j-1} = 2b_{3j-2} - b_{3j-3}, \tag{7.10}$$

$$b_{3j} = b_{3j-4} - 4b_{3j-3} + 4b_{3j-2}, \tag{7.11}$$

where $j = 2, 3, \ldots, N$. Therefore, discarding the superscript j, we write the collocation solution as

$$Y(z) = \sum_{k=1}^{4} b_{k+3(j-1)} B_k(z). \tag{7.12}$$

The boundary conditions in (7.3) become

$$b_1 = \alpha, \tag{7.13}$$

$$b_2 - b_1 = \frac{h\gamma}{3}, \tag{7.14}$$

$$b_{3N+1} = \beta. \tag{7.15}$$

Substitute (7.12) into the differential equation (7.2), evaluate at the collocation points $z = 0.5$ on each interval together with equations (7.10), (7.11), (7.13), (7.14) and (7.15) to get a $3N + 1$ linear system of equations that contain $3N + 1$ unknowns. The solution of this system will be used to compute the solution to the differential equation. The solution at nodes x_i is given via $Y(x_i) = b_{3i-2}$.

We now illustrate our method with some applications to solving partial differential equations in the next section.

7.3 Application of the Cubic B-Spline OCFE to KdV–Burgers' equation

The third-order KdV–Burgers' equation [67] combines KdV [54] and Burgers' equations. It is given by

$$u_t = -\epsilon u u_x + \nu u_{xx} - \mu u_{xxx}. \tag{7.16}$$

We substitute a modified version of (7.12), where the coefficients are functions of t, into (7.16) and obtain

$$\sum_{k=1}^{4} b'_{k+3(i-1)}(t)\ B_k(z) + \frac{\mu}{h^3}\sum_{k=1}^{4} b_{k+3(i-1)}(t)\ B'''_k(z) - \frac{\nu}{h^2}\sum_{k=1}^{4} b_{k+3(i-1)}(t)\ B''_k(z) + \frac{\epsilon}{h}\left(\sum_{k=1}^{4} b_{k+3(i-1)}(t)\ B_k(z)\right)\left(\sum_{k=1}^{4} b_{k+3(i-1)}(t)\ B'_k(z)\right) = 0, \quad i = 1, 2, \ldots, N. \tag{7.17}$$

Applying Crank-Nicolson method together with quasilinearization (similar to section 3.1) yields

$$\begin{aligned}&\sum_{k=1}^{4}\left(\left[1 + \frac{\epsilon\Delta t}{2h}\sum_{k=1}^{4} b_{k+3(i-1)}(t_j)\ B'_k(z)\right] B_k(z) + \frac{\mu\Delta t}{2h^3}\ B'''_k(z)\right.\\ &\left. + \frac{\epsilon\Delta t}{2h}\left[\sum_{k=1}^{4} b_{k+3(i-1)}(t_j) B_k(z)\right] B'_k(z) - \frac{\nu\Delta t}{2h^2}\ B''_k(z)\right) b_{k+3(i-1)}(t_{j+1})\\ &= \sum_{k=1}^{4}\left[B_k(z) - \frac{\mu\Delta t}{2h^3}\ B'''_k(z) + \frac{\nu\Delta t}{2h^2}\ B''_k(z)\right] b_{k+3(i-1)}(t_j), \quad i = 1, 2, \ldots, N.\end{aligned} \tag{7.18}$$

We now solve equation (7.16) with different values of the parameters ε, μ and ν.

Example 7.1: We consider the case of equation (7.16) when $\varepsilon = 1$ with the exact solution

$$u(x,t) = -\frac{6\nu^2}{25\mu}\left(1 + \tanh\left(\frac{\nu}{10\mu}\left(x + \frac{6\nu^2 t}{25\mu}\right)\right) - \frac{1}{2}\left(\operatorname{sech}\left(\frac{\nu}{10\mu}\left(x + \frac{6\nu^2 t}{25\mu}\right)\right)\right)^2\right), \tag{7.19}$$

the initial condition

$$u(x,0) = -\frac{6\,\nu^2}{25\,\mu}\left(1 + \tanh\left(\frac{\nu\,x}{10\mu}\right) - \frac{1}{2}\ \left(\operatorname{sech}\left(\frac{\nu\,x}{10\mu}\right)\right)^2\right), \tag{7.20}$$

and the boundary conditions

$$u(-50,t) = 0,\ u(50,t) = 0 \text{ and } u_x(50,t) = 0. \tag{7.21}$$

A 3D plot of the approximate solution and error are depicted in Figure 7.5 and Figure 7.6, respectively when $\mu = \nu = 1$. The error is shown for various values of

$N = 50, 100, 200, 300, 400$ and 500. The time step for each N is $\Delta t = \frac{10}{N}$. It is seen that the OCFE method using cubic splines produces a highly accurate solution for moderate values of N. This shows that this method gives a good approximation of the exact solution.

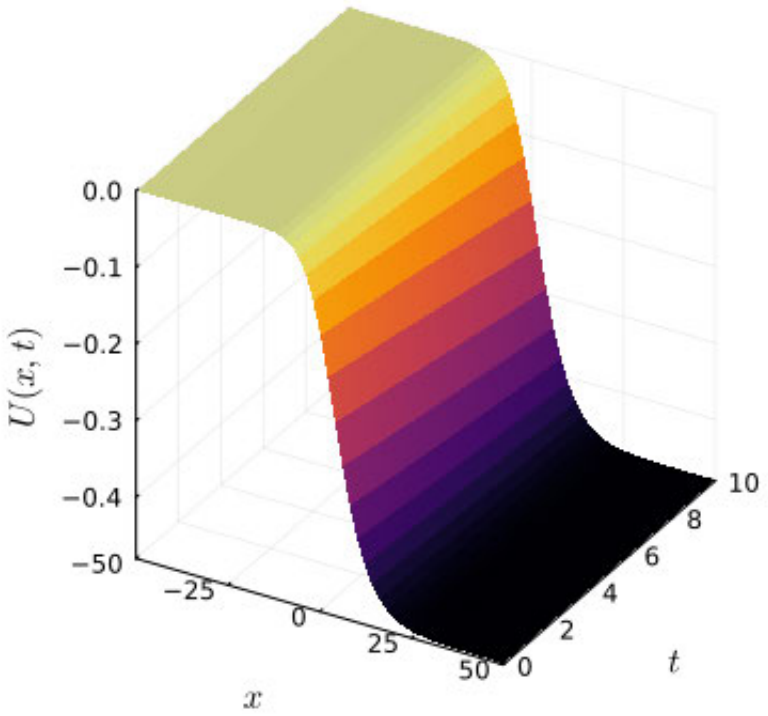

Figure 7.5: 3D plot of the approximate solution for example 7.1 when $N = 50$.

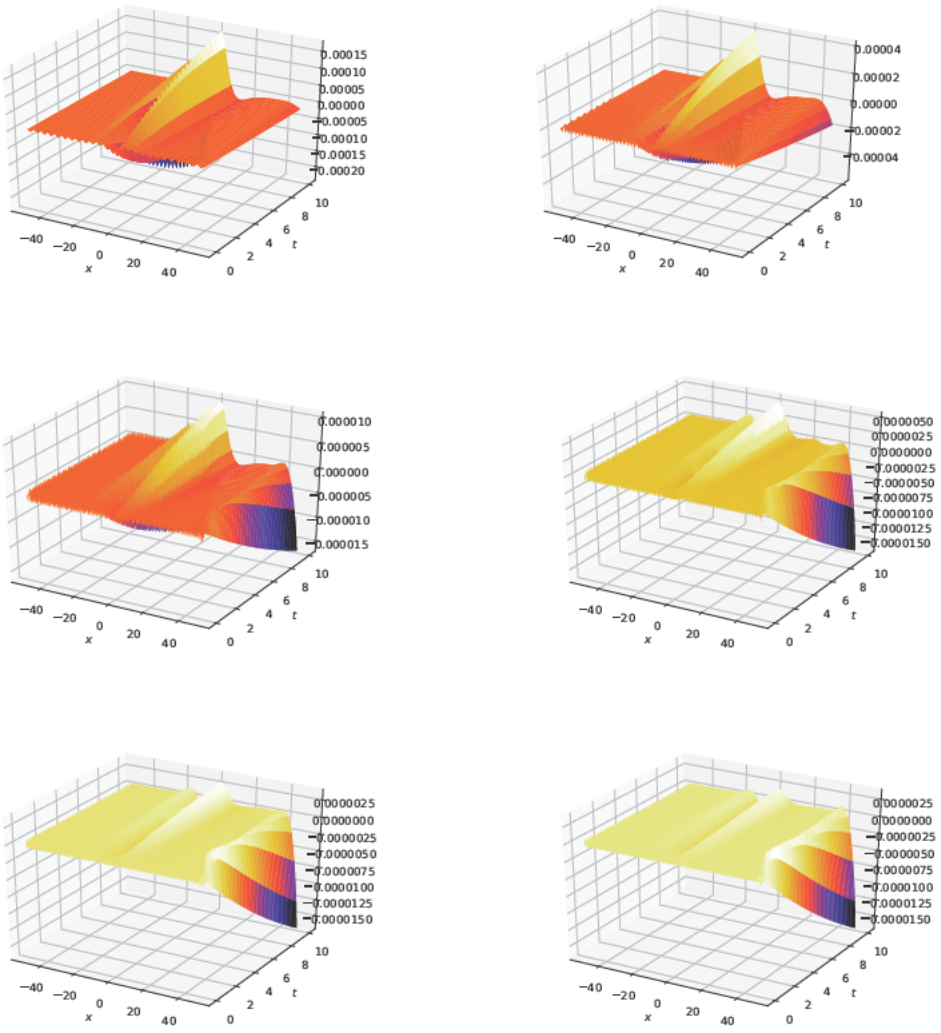

Figure 7.6: 3D plot of error of KdV–Burgers; equation when $N = 50,\ 100$ (top: left to right) $N = 200,\ 300$ (middle: left to right) and $N = 400,\ 500$ (bottom: left to right).

Example 7.2: A case of the KdV–Burgers' equation whose exact solution is not known in the literature is considered. We use $\varepsilon = 0.2$, $\nu = 0.2$ and $\mu = 0.1$ with the initial condition [76]

$$u(x,0) = \frac{1}{2}\left(1 - \tanh\left(\frac{|x| - x_0}{d}\right)\right), \tag{7.22}$$

and boundary conditions

$$u_x(-50,t) = u_x(150,t) = u_{xx}(150,t) = 0. \tag{7.23}$$

We solve the problem using the OCFE method with $N = 300$ and compare the solution with the results obtained using the Mathematica v13.1 NDSolve built-in solver. We note that the value of $N = 300$ is much smaller than the value of $N = 4000$ used in [76]. The results are presented graphically in Figures 7.7 and 7.8.

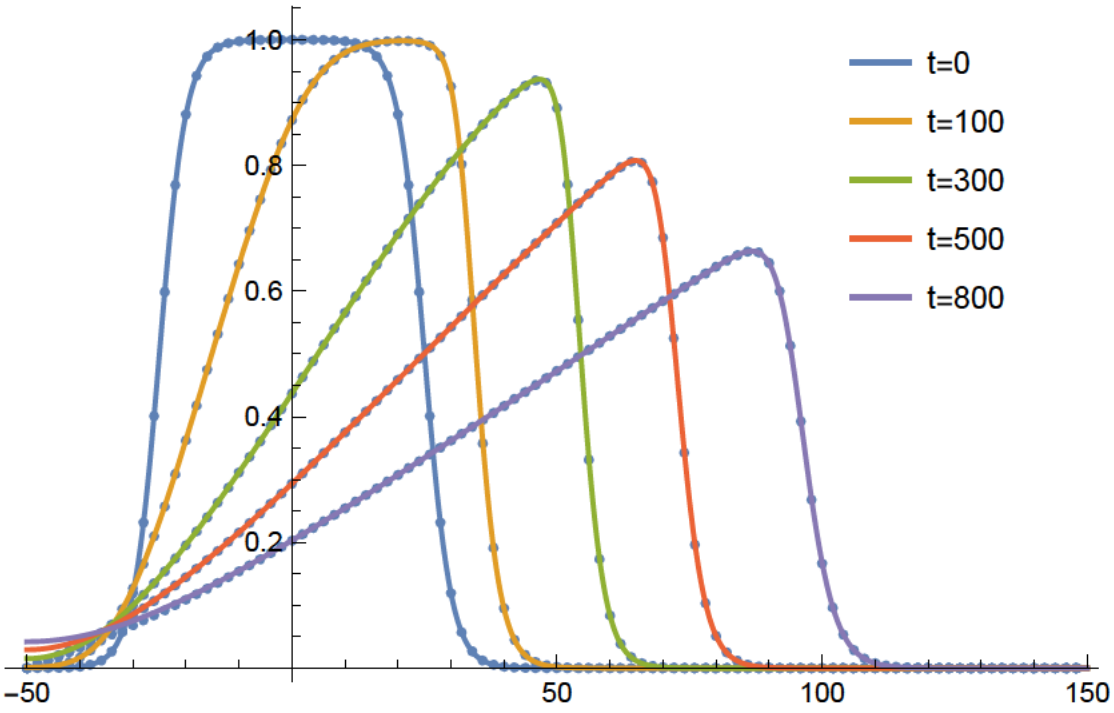

Figure 7.7: Comparison of OCFE solution (dots) and Mathematica built-in solver (solid line) at different times.

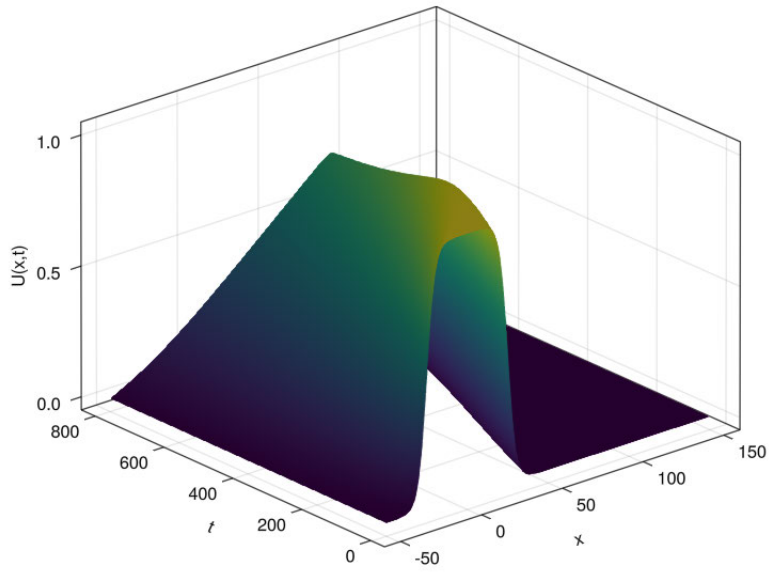

Figure 7.8: 3D plot for the KdV-Burgers' equation when $N = 300,\ t = 800$.

The results in Figure 7.7 demonstrate that the OCFE method produces results which match the solution produced with Mathematica.

7.4 Burgers' equation

In this section, we use the cubic B-spline OCFE method to solve the Burgers' equation

$$u_t = \nu u_{xx} - u u_x, \tag{7.24}$$

with the boundary conditions

$$u(a,t) = 0,\ u(b,t) = 0, \tag{7.25}$$

and the initial condition

$$u_0(x) = u(x,1). \tag{7.26}$$

The exact solution is

$$u(x,t) = \frac{x}{t\left(1 + \sqrt{\frac{t}{\tau}} e^{\frac{x^2}{4\nu t}}\right)}, \tag{7.27}$$

where $\tau = \exp(\frac{1}{8\nu})$.

The discretization of equation (7.24) is done along similar lines of (7.16) by setting $\mu = 0$ in (7.18).

In order to use cubic B-spline OCFE to solve the Burgers equation, we require evaluation at an additional collocation point to ensure a linear system of equations that has a unique solution. We choose roots of the shifted Legendre polynomials as our collocation points. We map $[-1, 1]$ to $[0, 1]$ using the linear transformation $T : z \to 2z - 1$. Thus we consider the roots of the shifted Legendre polynomial of degree 2 given as

$$\bar{P}_2(z) = 6z^2 - 6z + 1. \tag{7.28}$$

The roots are $\frac{1}{6}(3 - \sqrt{3})$ and $\frac{1}{6}(3 + \sqrt{3})$. We use $z = 0$ in the first element and $z = 1$ in the last element, which corresponds to the global domain, to satisfy the boundary conditions. In addition we use $z = \frac{1}{6}(3 - \sqrt(3))$ in each element as collocation points. This yields $N + 2$ equations. These coupled with the first and second derivative continuity conditions at the boundary of the elements give an additional $2(N - 1)$ equations. We choose the additional collocation point $\frac{1}{6}(3 + \sqrt(3))$ in the first element for a total of $3N + 1$ equations.

A 3D plot of the approximate solution is shown in Figure 7.9 and that of the error is shown in Figure 7.10. Furthermore, Table 7.1 displays the convergence rates for the cubic B-spline solution of the Burgers' equation.

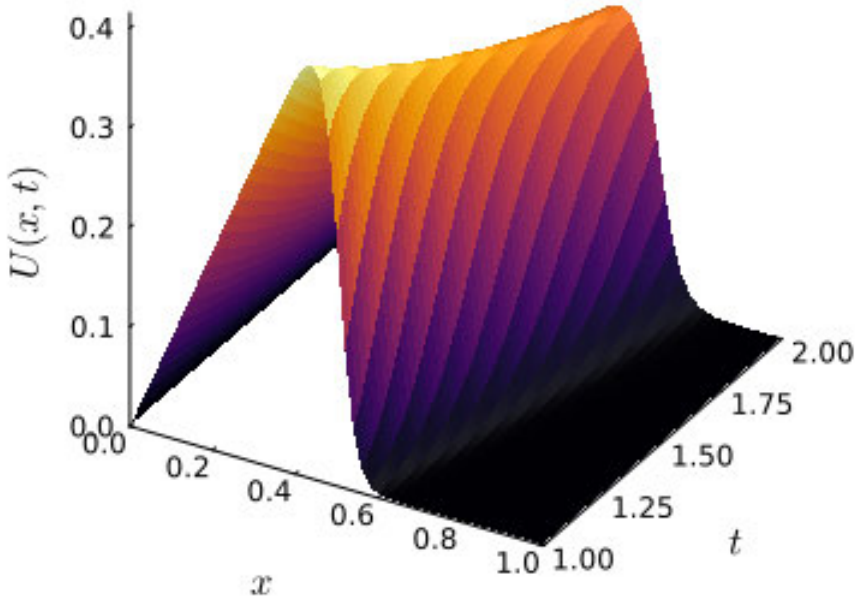

Figure 7.9: 3D plot of the approximate solution to the Burgers' equation at $N = 50$.

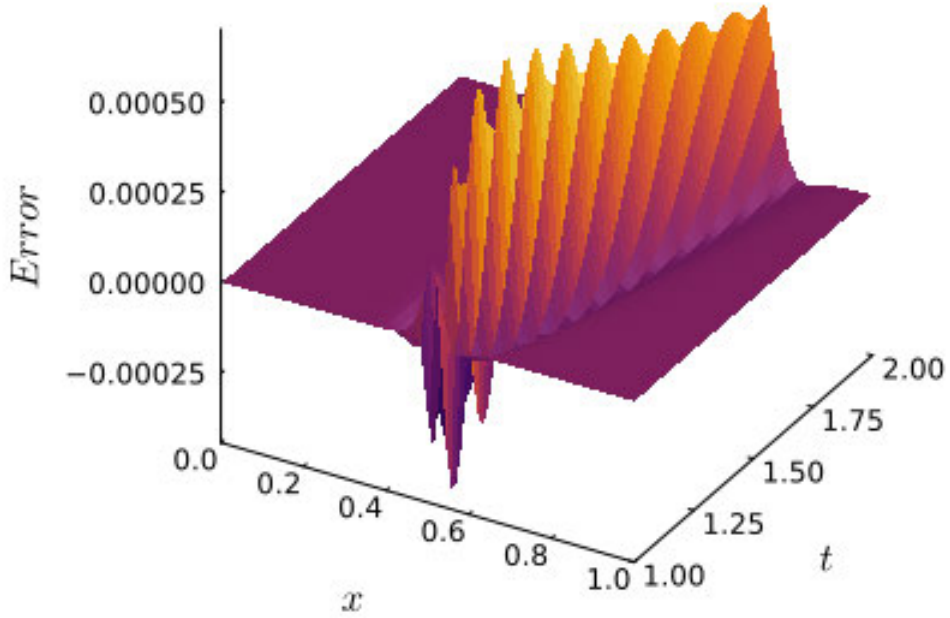

Figure 7.10: 3D plot of error for the Burgers' equation when $N = 50$.

Table 7.1: Convergence rates for the Burgers' equation with cubic B-spline when $\nu = 0.005$, $N = 50$ and $\Delta t = 0.02$.

x	Rate	x	Rate
0.10	3.6130	0.66	1.5175
0.20	3.2835	0.70	2.0226
0.30	2.8957	0.84	3.5850
0.40	2.4770	0.90	1.9268
0.50	1.9545	0.96	3.2883

Figure 7.10 shows that the cubic B-spline OCFE gives less error and more accurate than the quadratic B-spline version in Figure 3.2 at the Gauss points. Also the convergence rates for the cubic Bspline case in Table 7.1 are higher than that of the quadratic case in Table 3.1. Hence the cubic B-spline performs better than quadratic OCFE for the Burgers' equation.

Note 7.1: Greville points [33] can also be used in place of Gauss points as collocation points for the cubic B-spline solution of the Burgers equation.

7.5 KdV equation

In this section, we use the cubic B-spline collocation to solve the KdV equation

$$u_t = -\epsilon u u_x - \mu u_{xxx}, \tag{7.29}$$

with initial condition

$$u(x, 0) = \phi(x), \tag{7.30}$$

and boundary conditions

$$u(a,t) = 0,\ u(b,t) = 0 \text{ and } u_x(b,t) = 0. \tag{7.31}$$

The discretization of equation (7.29) is done along similar lines of (7.16) by setting $\nu = 0$ in (7.18).

In order to use cubic B-spline OCFE to solve the KdV equation, we require evaluation at a collocation point per interval together with the continuity and boundary conditions to make a linear system of equations that has a unique solution. We consider the root of the shifted Legendre polynomial of degree 1 given as

$$\bar{P}_1(z) = 2z - 1. \tag{7.32}$$

We choose the point $z = 0.5$ as our collocation point.

Example 7.3: The third order KdV equation obtained when $\epsilon = 1$, $\mu = 4.84 \times 10^{-4}$ [66] with initial condition

$$u(x,0) = 3c_1\text{sech}^2(xc_2 + c_4), \tag{7.33}$$

and boundary conditions

$$u(0,t) = 0,\ u(2,t) = 0 \text{ and } u_x(2,t) = 0, \tag{7.34}$$

has the exact solution

$$3c_1\text{sech}^2(xc_2 - tc_3 + c_4), \tag{7.35}$$

where $c_1 = 0.3,\ c_2 = 0.5\sqrt{\frac{c_1\varepsilon}{\mu}},\ c_3 = c_1\varepsilon c_2$ and $c_4 = -6.0$.

The graphs of the approximate solution, profiles at different times and the error are shown in Figures 7.11, 7.12 and 7.13 when $N = 500$ and $\Delta t = 0.002$.

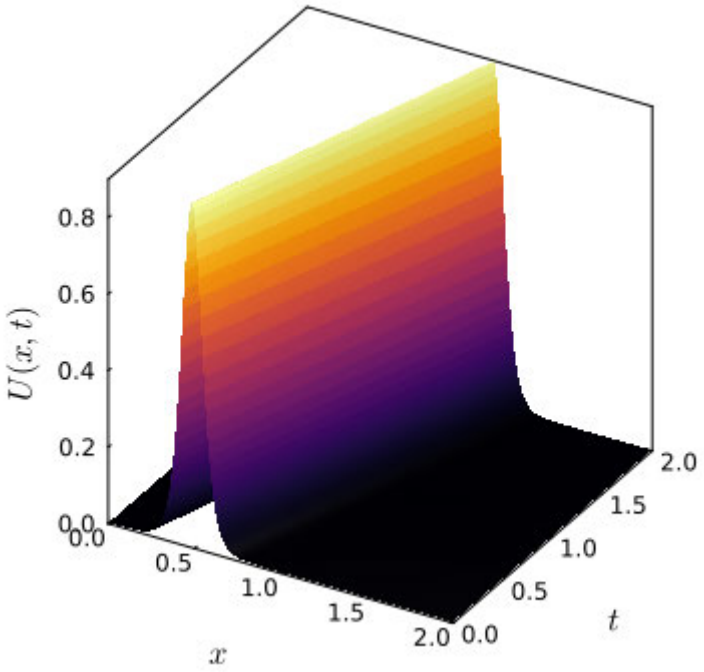

Figure 7.11: Plot of the approximate solution of KdV equation

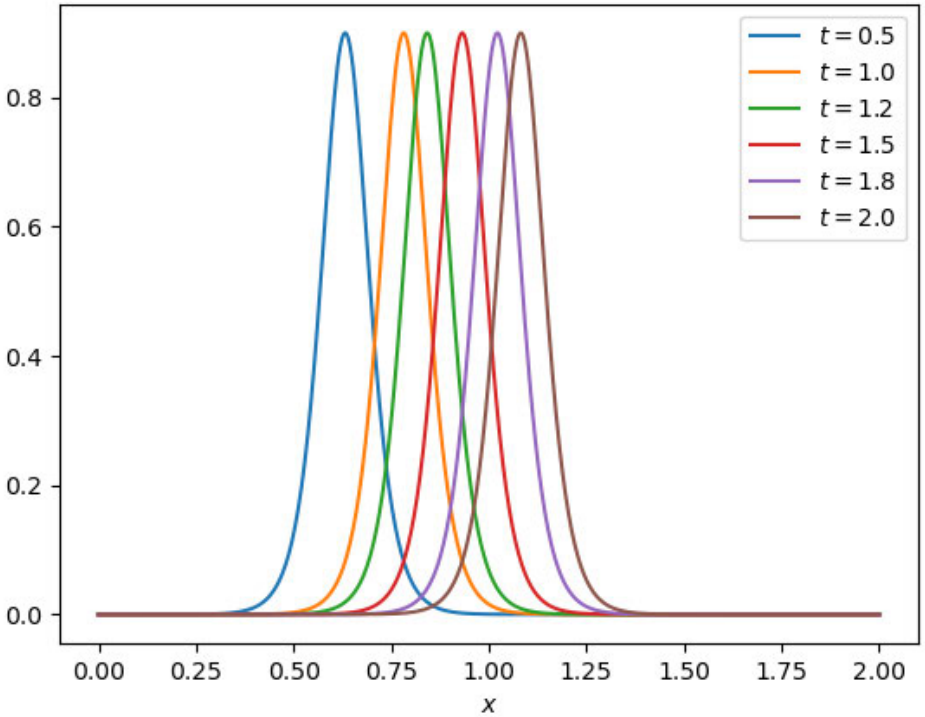

Figure 7.12: Profiles of the KdV equation at different times.

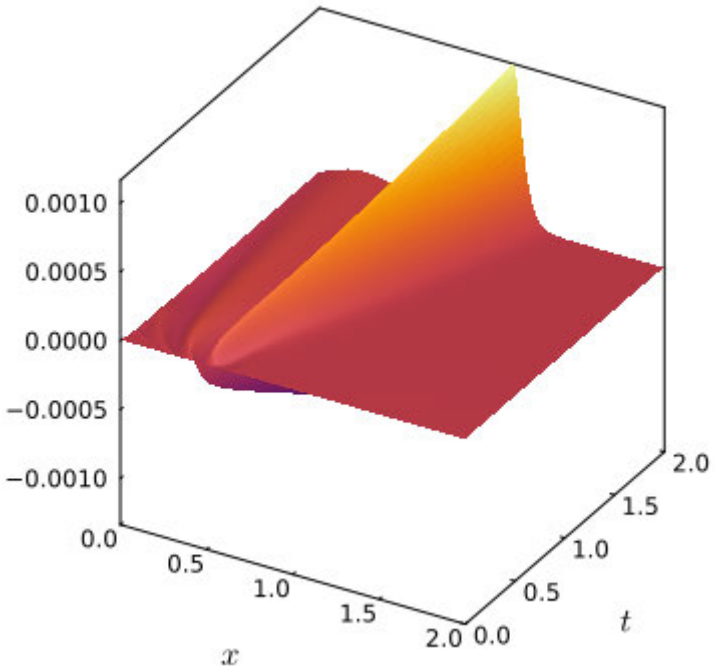

Figure 7.13: Plot of error for KdV equation

The convergence rate at time $t = 1$ when $N = 100$ is shown in Table 7.2. As expected the convergence rate is of order 2, which is in agreement with [20].

Table 7.2: Convergence rates for the KdV equation when $N = 100$, $t = 1$ and $\Delta t = 0.01$.

x	Rate	x	Rate
0.02	1.9045	1.08	1.9032
0.08	1.8808	1.20	1.9377
0.16	2.0459	1.38	1.9554
0.24	1.9187	1.58	1.9559
0.32	1.8978	1.70	1.9518
0.72	1.8564	1.86	2.2457

7.6 Application of cubic B-spline to fractional differential equations

In this section we approximate the solution of the fractional ordinary differential equation

$$D_x^\alpha u(x) = \frac{d^\alpha u(x)}{dx^\alpha} = f(x), \quad n-1 < \alpha < n, \tag{7.36}$$

with boundary conditions

$$u(a) = \delta_3, \ u(b) = \delta_4, \tag{7.37}$$

by cubic B-spline functions. We use the Caputo fractional derivative defined by

$$D_x^\alpha u(x) = \frac{1}{\Gamma(n-\alpha)} \int_a^x \frac{d^n u(\xi)}{d\xi^n} (x-\xi)^{n-1-\alpha} d\xi, \quad n-1 < \alpha < n, \tag{7.38}$$

where $\alpha \in \mathbb{R}$ and $n \in \mathbb{N}$. We shall consider the case $n = 2$ only. This gives

$$D_x^\alpha u(x) = \frac{1}{\Gamma(2-\alpha)} \int_a^x (x-\xi)^{1-\alpha} u''(\xi) d\xi, \quad 1 < \alpha < 2, \tag{7.39}$$

Let $I_i = [x_i, x_{i+1}] \subset [a, b]$ be an interval. Suppose the exact and approximate solutions of equation (7.36) are $u(x)$ and $u_c(x)$ respectively. In the interval I_i, we assumed that

$$u(x) \approx u_c^i(x) = \sum_{k=1}^{4} b_{k+3(i-1)} B_k(x). \tag{7.40}$$

Therefore equation (7.39) can be written as

$$D_x^\alpha u(x) = \frac{1}{\Gamma(2-\alpha)} \int_a^x (x-\xi)^{1-\alpha} \sum_{k=1}^{4} b_{k+3(i-1)} B_k''(\xi) \ d\xi, \quad 1 < \alpha < 2, \tag{7.41}$$

where $x_i = a + (i-1)h$, $h = x_{i+1} - x_i$,

$$B_1(\xi) = \left(1 - \frac{\xi - x_j}{h}\right)^3 \implies B_1''(\xi) = \frac{6}{h^2}\left(1 - \frac{\xi - x_j}{h}\right), \tag{7.42}$$

$$B_2(\xi) = 3\left(\frac{\xi - x_j}{h}\right)\left(1 - \frac{\xi - x_j}{h}\right)^2 \implies B_2''(\xi) = \frac{6}{h^2}\left(\frac{3(\xi - x_j)}{h} - 2\right), \tag{7.43}$$

$$B_3(\xi) = 3\left(\frac{\xi - x_j}{h}\right)^2\left(1 - \frac{\xi - x_j}{h}\right) \quad \Longrightarrow B_3''(\xi) = \frac{6}{h^2}\left(1 - \frac{3(\xi - x_j)}{h}\right), \tag{7.44}$$

and

$$B_4(\xi) = \left(\frac{\xi - x_j}{h}\right)^3 \quad \Longrightarrow B_4''(\xi) = \frac{6}{h^2}\left(\frac{\xi - x_j}{h}\right). \tag{7.45}$$

$$\begin{aligned} D_x^\alpha u(x) = \frac{1}{\Gamma(2-\alpha)}\Bigg[&\sum_{j=1}^{i-1}\sum_{k=1}^{4} b_{k+3(j-1)}\int_{x_j}^{x_{j+1}} B_k''(\xi)(x-\xi)^{1-\alpha}\,d\xi \\ &+ \sum_{k=1}^{4} b_{k+3(i-1)}\int_{x_i}^{x}(x-\xi)^{1-\alpha}B_k''(\xi)\,d\xi\Bigg]. \end{aligned} \tag{7.46}$$

Therefore, equation (7.46) becomes

$$\begin{aligned} D_x^\alpha u(x) = \frac{1}{\Gamma(2-\alpha)}\Bigg[&\sum_{j=1}^{i-1}\left[b_{3j-2}J_1 + b_{3j-1}J_2 + b_{3j}J_3 + b_{3j+1}J_4\right] \\ &+ b_{3i-2}I_1 + b_{3i-1}I_2 + b_{3i}I_3 + b_{3i+1}I_4\Bigg], \end{aligned} \tag{7.47}$$

where

$$J_k = \int_{x_j}^{x_{j+1}} B_k''(\xi)(x-\xi)^{1-\alpha}\,d\xi, \tag{7.48}$$

$$I_k = \int_{x_i}^{x} B_k''(\xi)(x-\xi)^{1-\alpha}\,d\xi, \quad k = 1,2,3,4. \tag{7.49}$$

Using

$$\begin{aligned} g_1(x) &= \int_{x_j}^{x_{j+1}}(x-\xi)^{1-\alpha}\,d\xi, \\ &= \frac{1}{(2-\alpha)(3-\alpha)}\left[(3-\alpha)(x-x_j)^{2-\alpha} - (3-\alpha)(x-x_{j+1})^{2-\alpha}\right], \end{aligned} \tag{7.50}$$

$$\begin{aligned} g_2(x) &= \frac{1}{h}\int_{x_j}^{x_{j+1}}(x-\xi)^{1-\alpha}(\xi - x_j)\,d\xi, \\ &= \frac{1}{(2-\alpha)(3-\alpha)h}\left[(x-x_j)^{3-\alpha} - (x-x_{j+1})^{3-\alpha} - (3-\alpha)h(x-x_{j+1})^{2-\alpha}\right]. \end{aligned} \tag{7.51}$$

We have

$$J_1 = \frac{6}{h^2}(g_1(x) - g_2(x)), \tag{7.52}$$

$$J_2 = \frac{6}{h^2}(3g_2(x) - 2g_1(x)), \tag{7.53}$$

$$J_3 = \frac{6}{h^2}(g_1(x) - 3g_2(x)), \tag{7.54}$$

$$J_4 = \frac{6}{h^2}g_2(x). \tag{7.55}$$

Similarly,

$$I_1 = \frac{6}{h^2}(\bar{g}_1(x) - \bar{g}_2(x)), \tag{7.56}$$

$$I_2 = \frac{6}{h^2}(3\bar{g}_2(x) - 2\bar{g}_1(x)), \tag{7.57}$$

$$I_3 = \frac{6}{h^2}(\bar{g}_1(x) - 3\bar{g}_2(x)), \tag{7.58}$$

$$I_4 = \frac{6}{h^2}\bar{g}_2(x), \tag{7.59}$$

where

$$\bar{g}_1(x) = \int_{x_i}^{x} (x-\xi)^{1-\alpha}\, d\xi = \frac{1}{(2-\alpha)(3-\alpha)}\left[(x-x_i)^{2-\alpha}(3-\alpha)\right], \tag{7.60}$$

$$\bar{g}_2(x) = \frac{1}{h}\int_{x_i}^{x} (x-\xi)^{1-\alpha}(\xi - x_j)\, d\xi = \frac{1}{(2-\alpha)(3-\alpha)h}(x-x_i)^{3-\alpha}. \tag{7.61}$$

Let $x \in [x_i, x_{i+1}]$ where for $i = 1, 2, \ldots, N$.

$$x - x_j = (i + z - j)h, \tag{7.62}$$

and

$$x - x_{j+1} = (i + z - j - 1)h. \tag{7.63}$$

Therefore,

$$g_1(z) = P_\alpha\left[(3-\alpha)((i+z-j)^{2-\alpha} - (i+z-j-1)^{2-\alpha})\right], \tag{7.64}$$

$$g_2(z) = P_\alpha \left[(i+z-j)^{3-\alpha} - (i+z-j-1)^{3-\alpha} \right. \\ \left. -(3-\alpha)(i+z-j-1)^{2-\alpha}\right], \tag{7.65}$$

$$\bar{g}_1(z) = P_\alpha(3-\alpha)z^{2-\alpha}, \tag{7.66}$$

$$\bar{g}_2(z) = P_\alpha z^{3-\alpha}, \tag{7.67}$$

where $P_\alpha = \frac{h^{2-\alpha}}{(2-\alpha)(3-\alpha)}$. We can now say that

$$J_1(z) = R_\alpha \left[(3-\alpha)(i+z-j)^{2-\alpha} - (i+z-j)^{3-\alpha} + (i+z-j-1)^{3-\alpha}\right], \tag{7.68}$$

$$J_2(z) = R_\alpha \left[3(i+z-j)^{3-\alpha} - 3(i+z-j-1)^{3-\alpha} - 2(3-\alpha)(i+z-j)^{2-\alpha} \right. \\ \left. -(3-\alpha)(i+z-j-1)^{2-\alpha}\right], \tag{7.69}$$

$$J_3(z) = R_\alpha \left[(3-\alpha)(i+z-j)^{2-\alpha} + 2(3-\alpha)(i+z-j-1)^{2-\alpha} \right. \\ \left. +3(i+z-j-1)^{3-\alpha} - 3(i+z-j)^{3-\alpha}\right], \tag{7.70}$$

$$J_4(z) = R_\alpha \left[(i+z-j)^{3-\alpha} - (i+z-j-1)^{3-\alpha} \right. \\ \left. -(3-\alpha)(i+z-j-1)^{2-\alpha}\right]. \tag{7.71}$$

Similarly,

$$I_1(z) = R_\alpha \left[(3-\alpha)z^{2-\alpha} - z^{3-\alpha}\right], \tag{7.72}$$

$$I_2(z) = R_\alpha \left[3z^{3-\alpha} - 2(3-\alpha)z^{2-\alpha}\right], \tag{7.73}$$

$$I_3(z) = R_\alpha \left[(3-\alpha)z^{2-\alpha} - 3z^{3-\alpha}\right], \tag{7.74}$$

$$I_4(z) = R_\alpha z^{3-\alpha}, \tag{7.75}$$

where $R_\alpha = \frac{6h^{-\alpha}}{(2-\alpha)(3-\alpha)}$. Substitute equations (7.68) - (7.75) into (7.47), we have

$$D_x^\alpha u(x_i + hz) = \frac{1}{\Gamma(2-\alpha)} \left[\sum_{j=1}^{i-1}\sum_{k=1}^{4} b_{k+3(j-1)} J_k(z) + \sum_{k=1}^{4} b_{k+3(i-1)} I_k(z)\right], \\ i = 1, 2, 3 \ldots, N. \tag{7.76}$$

Let

$$\begin{aligned} J_k^* &= \frac{1}{R_\alpha} J_k, \\ I_k^* &= \frac{1}{R_\alpha} I_k \quad k = 1, 2, 3, 4. \end{aligned} \tag{7.77}$$

Therefore

$$\begin{aligned} D_x^\alpha u(x_i + hz) = &\frac{R_\alpha}{\Gamma(2-\alpha)} \left[\sum_{j=1}^{i-1} \sum_{k=1}^{4} b_{k+3(j-1)} J_k^*(z) + \sum_{k=1}^{4} b_{k+3(i-1)} I_k^*(z) \right], \\ &\frac{6h^{-\alpha}}{\Gamma(4-\alpha)} \left[\sum_{j=1}^{i-1} \sum_{k=1}^{4} b_{k+3(j-1)} J_k^*(z) + \sum_{k=1}^{4} b_{k+3(i-1)} I_k^*(z) \right], \\ &i = 1, 2, 3 \ldots, N. \end{aligned} \tag{7.78}$$

The boundary conditions in (7.37) become

$$b_1 = \delta_3, \ b_{3N+1} = \delta_4. \tag{7.79}$$

We then incorporate equation (7.78) into (7.36) by substituting it into the left hand side. Thus we evaluate the resulting equation at point $z = z_1$ for all intervals and another point $z = z_2$ in the last interval to get $N + 1$ equations. These equations together with the continuity equations (7.10), (7.11) and boundary conditions in (7.79) give a square linear system of size $(3N + 1)$ given by

$$M_\alpha \ S \ \mathbf{b} = \mathbf{f}, \tag{7.80}$$

where S is a square matrix, $\mathbf{b} = [b_1, b_2, \ldots, b_{3N+1}]^T$, $\mathbf{f} = [f_1, f_2, \ldots, f_{N+1}, 0, \ldots, 0, \delta_3, \delta_4]^T$ and $M_\alpha = \frac{6h^{-\alpha}}{\Gamma(4-\alpha)}$. When $z_1 = \frac{1}{2}$ and $z_2 = \frac{1}{3}$, the matrix S for $N = 2$ is $S = 6^{\alpha-3}T$ where T is

$$\begin{bmatrix}
\frac{2^{2\alpha-3}(135-54\alpha)}{6^{\alpha}} & \frac{2^{2\alpha-3}(108\alpha-243)}{6^{\alpha}} & \frac{2^{2\alpha-3}(81-54\alpha)}{6^{\alpha}} & \frac{272^{2\alpha-3}}{6^{\alpha}} & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{272^{2\alpha-3}(3^{\alpha}-18\alpha+27)}{18^{\alpha}} & \frac{272^{2\alpha-3}\left(23^{\alpha}\alpha-3^{\alpha+2}+36\alpha-27\right)}{18^{\alpha}} & \frac{2^{2\alpha-3}\left(-1083^{\alpha}\alpha+1353^{\alpha+1}-486\alpha-729\right)}{18^{\alpha}} & \frac{2^{2\alpha-3}(729-543^{\alpha})}{18^{\alpha}} & \frac{2^{2\alpha-3}(108\alpha-243)}{6^{\alpha}} & \frac{2^{2\alpha-3}(81-54\alpha)}{6^{\alpha}} & \frac{272^{2\alpha-3}}{6^{\alpha}} & 0 & 0 & 0 \\
\frac{2^{2\alpha-3}(-135045^{\alpha}\alpha+67545^{\alpha}+72975^{\alpha})}{1350^{\alpha}} & \frac{2^{2\alpha-3}(270045^{\alpha}\alpha+202545^{\alpha}+48675^{\alpha}\alpha-364575^{\alpha})}{1350^{\alpha}} & \frac{2^{2\alpha-3}\left(-135045^{\alpha}\alpha-13545^{\alpha+1}-97275^{\alpha}\alpha+510375^{\alpha}\right)}{1350^{\alpha}} & \frac{272^{2\alpha-3}(15^{\alpha}+1253^{\alpha}-545^{\alpha})}{90^{\alpha}} & \frac{272^{2\alpha-3}\left(23^{\alpha}\alpha-3^{\alpha+2}+36\alpha-27\right)}{18^{\alpha}} & \frac{2^{2\alpha-3}\left(-1083^{\alpha}\alpha+1353^{\alpha+1}-486\alpha-729\right)}{18^{\alpha}} & \frac{2^{2\alpha-3}(729-543^{\alpha})}{18^{\alpha}} & \frac{2^{2\alpha-3}(108\alpha-243)}{6^{\alpha}} & \frac{2^{2\alpha-3}(81-54\alpha)}{6^{\alpha}} & \frac{272^{2\alpha-3}}{6^{\alpha}} \\
0 & 0 & w & -2w & w & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & w & -2w & w & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w & -2w \\
0 & -w & 4w & -4w & 0 & w & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -w & 4w & -4w & 0 & w & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{7.81}$$

and $w = \frac{6^{3-\alpha}h^{\alpha}\Gamma(4-\alpha)}{6}$. The solution to the fractional differential equation (7.36) is then obtained via equation (7.40).

Example 7.4: Consider the fractional ordinary differential equation

$$\frac{d^{\alpha}u(x)}{dx^{\alpha}} = \frac{x^{3-\alpha}}{\Gamma(4-\alpha)}, \quad 1 < \alpha < 2, \tag{7.82}$$

with the exact solution $u(x) = \frac{1}{6}x^3$ and boundary conditions $u(0) = 0$ and $u(1) = \frac{1}{6}$.

We shall use the cubic B-spline OCFE to obtain it's numerical solution. Figure 7.14 shows that the exact and the approximate solutions are in good agreement. In Figure 7.15, the error is shown when $\alpha = 1.5$ and $N = 50$. The collocation point used are $z_1 = \frac{1}{6}(3 - \sqrt{3})$ and $z_2 = \frac{1}{2}$.

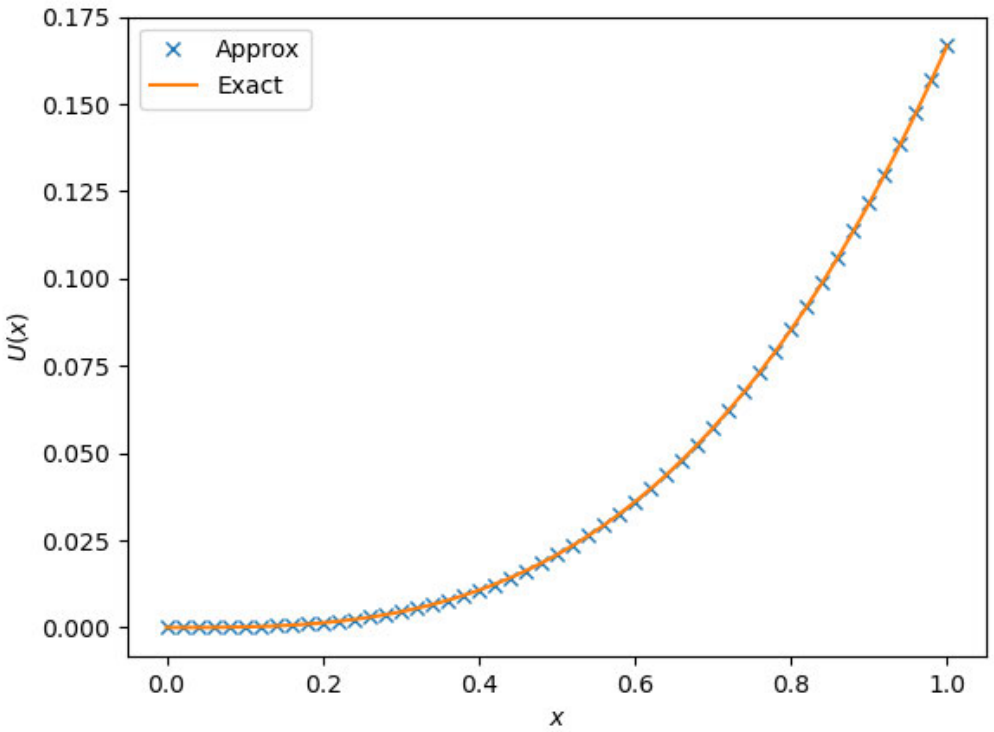

Figure 7.14: Numerical solution example for 7.4 at $N = 50$ and $\alpha = 1.5$.

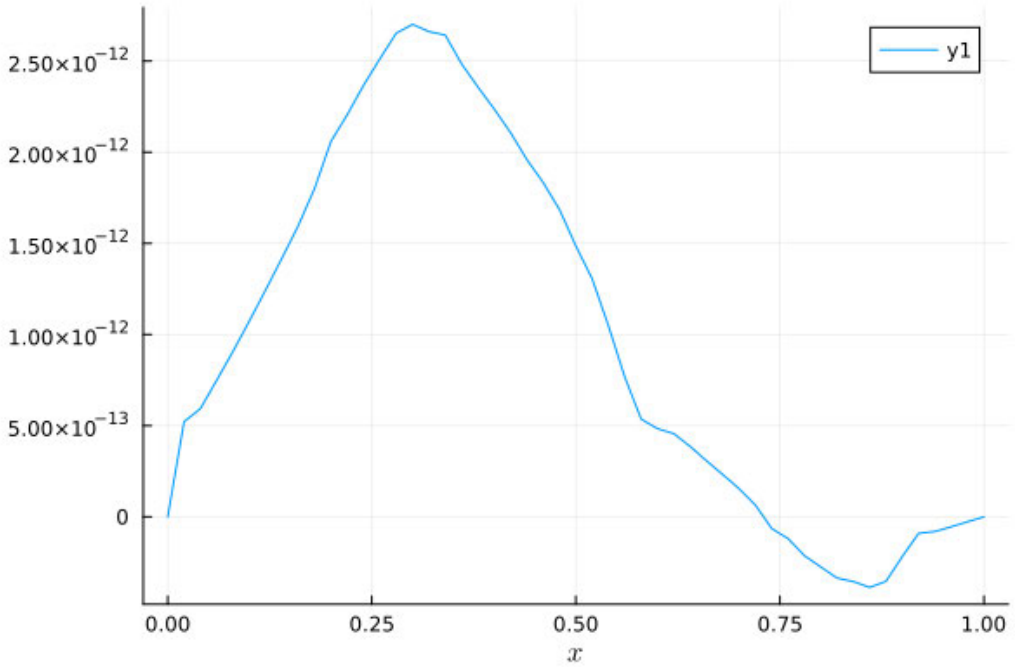

Figure 7.15: Error for example 7.4 at $N = 50$ and $\alpha = 1.5$.

The numerical solution in Figure 7.14 and the error in Figure 7.15 shows that the cubic B-spline OCFE can solve partial differential equations of fractional order in space when the order is $1 < \alpha < 2$.

Example 7.5: We consider the linear fractional diffusion equation

$$\frac{\partial u}{\partial t} - \Gamma(4-\alpha)\; x^{\alpha}\; \frac{\partial^{\alpha} u}{\partial x^{\alpha}} + 7u = 2\alpha x^2 e^{-t}, \quad 1 < \alpha < 2, \tag{7.83}$$

with initial condition

$$u(x,0) = x^2 - x^3 \tag{7.84}$$

and boundary conditions

$$\left.\begin{aligned} u_x(0,t) &= 0, \\ u_x(2,t) &= -8e^{-t}. \end{aligned}\right\} \tag{7.85}$$

The exact solution is $u(x,t) = (x^2 - x^3)e^{-t}$.

The discretized form of (7.83) is

$$\left[1 + \frac{7\Delta t}{2}\right] u_i^{j+1} - \frac{\Delta t}{2} m_i D_\alpha u_i^{j+1} = \left[1 - \frac{7\Delta t}{2}\right] u_i^{j} + \frac{\Delta t}{2} m_i D_\alpha u_i^{j} + \frac{\Delta t}{2}(r_i^{j+1} + r_i^{j}), \tag{7.86}$$

where $m(x) = \Gamma(4-\alpha)\ x^{\alpha}$ and $r(x,t) = 2\alpha x^2 e^{-t}$. The 3D plot of the approximate solution to this example is shown in Figure 7.16 and that of the error is shown in Figure 7.17. The collocation point used are $z_1 = \frac{1}{6}(3-\sqrt{3})$ and $z_2 = \frac{1}{2}$. This means that the cubic B-spline OCFE can solve fractional differential equations efficiently.

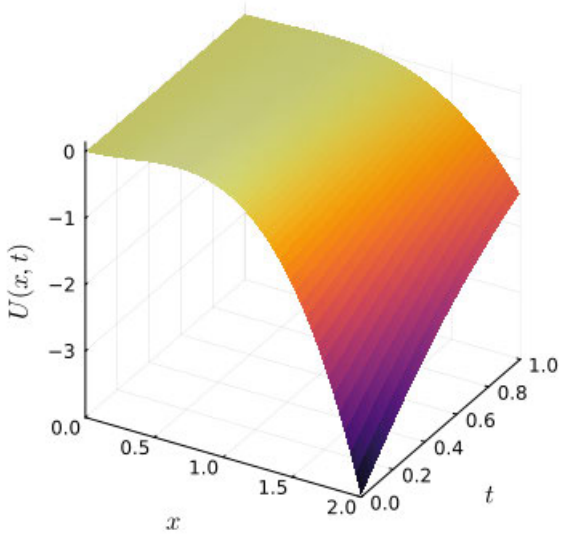

Figure 7.16: 3D plot of the approximate solution for example 7.5 when $\alpha = 1.5$, $N = 50$ and $\Delta t = 0.02$.

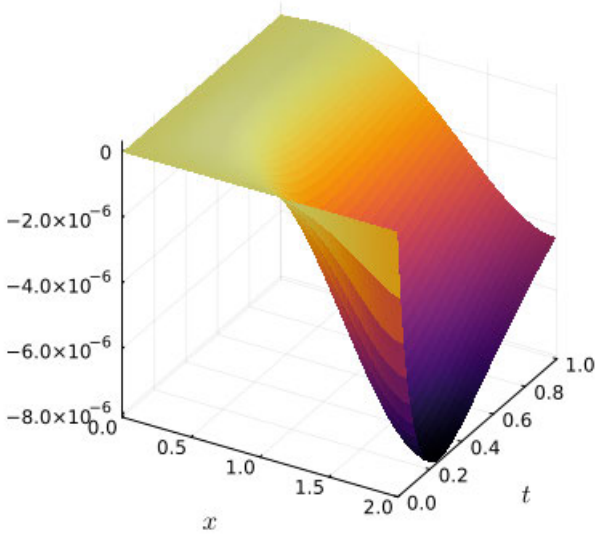

Figure 7.17: Error for example 7.5 plot when $\alpha = 1.5$, $N = 50$ and $\Delta t = 0.02$.

We compare the absolute errors and convergence rates of the cubic and quadratic B-spline solutions of fractional differential equation of order $1 < \alpha < 2$. The absolute errors for the cubic B-spline cases in Figures 7.15 (ODE) and 7.17 (PDE) are less than that of the quadratic B-spline cases in Figure 5.2 and 5.10 respectively. Hence the numerical solution of the fractional differential equations of order $1 < \alpha < 2$ using cubic B-spline OCFE gives better accuracy than the quadratic B-spline OCFE.

7.7 Application of the cubic b-spline for space fractional differential equation of order $0 < \alpha < 1$

In the case of space fractional derivative of order $0 < \alpha < 1$, we employ a similar approach used in the previous section. The Caputo space fractional derivative of order $0 < \alpha < 1$ is

$$D_x^{\alpha} u(x) = \frac{1}{\Gamma(1-\alpha)} \int_a^x (x-\xi)^{-\alpha} u'(\xi) d\xi, \quad 0 < \alpha < 1. \tag{7.87}$$

This can be approximated by

$$\begin{aligned} D_x^{\alpha} u(x) = \frac{1}{\Gamma(1-\alpha)} \Bigg[& \sum_{j=1}^{i-1} [b_{3j-2} J_1 + b_{3j-1} J_2 + b_{3j} J_3 + b_{3j+1} J_4] \\ & + b_{3i-2} I_1 + b_{3i-1} I_2 + b_{3i} I_3 + b_{3i+1} I_4 \Bigg], \end{aligned} \tag{7.88}$$

where

$$J_k = \int_{x_j}^{x_{j+1}} B_k'(\xi)(x-\xi)^{-\alpha} \, d\xi, \tag{7.89}$$

$$I_k = \int_{x_i}^{x} B_k'(\xi)(x-\xi)^{-\alpha} \, d\xi, \quad k = 1, 2, 3, 4. \tag{7.90}$$

Let

$$\begin{aligned} g_1(x) &= \int_{x_j}^{x_{j+1}} (x-\xi)^{-\alpha}\, d\xi, \\ &= \frac{(\alpha-2)(\alpha-3)}{(\alpha-1)(\alpha-2)(\alpha-3)} \left[(x-x_{j+1})^{1-\alpha} - (x-x_j)^{1-\alpha} \right], \end{aligned} \tag{7.91}$$

$$\begin{aligned} g_2(x) &= \frac{1}{h} \int_{x_j}^{x_{j+1}} (x-\xi)^{-\alpha} (\xi - x_j)\, d\xi, \\ &= \frac{(\alpha-3)}{(\alpha-1)(\alpha-2)(\alpha-3)h} \left[(x-x_{j+1})^{1-\alpha} \left((\alpha-1)h - (x-x_j) \right) \right. \\ &\left. + (x-x_j)^{2-\alpha} \right]. \end{aligned} \tag{7.92}$$

$$\begin{aligned} g_3(x) &= \frac{1}{h^2} \int_{x_j}^{x_{j+1}} (x-\xi)^{-\alpha} (\xi - x_j)^2\, d\xi, \\ &= \frac{1}{(\alpha-1)(\alpha-2)(\alpha-3)h^2} \left[(x-x_{j+1})^{1-\alpha} \left[(x_j - x_{j+1})^2 \alpha^2 \right. \right. \\ &+ \alpha(x_j - x_{j+1})(3(x_{j+1} - x_j) + 2(x - x_j)) + 6(x - x_j)(x_{j+1} - x_j) \\ &\left. \left. + 2(x - x_{j+1})^2 \right] - 2(x - x_j)^{3-\alpha} \right]. \end{aligned} \tag{7.93}$$

We can now compute J_k, $k = 1, 2, 3, 4$ as follows:

$$J_1(x) = \frac{3}{h} \left[-g_1(x) + 2g_2(x) - g_3(x) \right], \tag{7.94}$$

$$J_2(x) = \frac{3}{h} \left[g_1(x) - 4g_2(x) + 3g_3(x) \right], \tag{7.95}$$

$$J_3(x) = \frac{3}{h} \left[2g_2(x) - 3g_3(x) \right], \tag{7.96}$$

$$J_4(x) = \frac{3}{h} \left[g_3(x) \right]. \tag{7.97}$$

Similarly,

$$\begin{aligned} \bar{g}_1(x) &= \int_{x_i}^{x} (x-\xi)^{-\alpha}\, d\xi, \\ &= \frac{(\alpha-2)(\alpha-3)}{(\alpha-1)(\alpha-2)(\alpha-3)} \left[-(x-x_i)^{1-\alpha} \right], \end{aligned} \tag{7.98}$$

$$\begin{aligned} \bar{g}_2(x) &= \frac{1}{h} \int_{x_i}^{x} (x-\xi)^{-\alpha} (\xi - x_j)\, d\xi, \\ &= \frac{(\alpha-3)}{(\alpha-1)(\alpha-2)(\alpha-3)h} \left[(x-x_j)^{2-\alpha} \right]. \end{aligned} \tag{7.99}$$

$$\begin{aligned}\bar{g}_3(x) &= \frac{1}{h^2}\int_{x_i}^{x}(x-\xi)^{-\alpha}(\xi-x_i)^2\,d\xi,\\ &= \frac{1}{(\alpha-1)(\alpha-2)(\alpha-3)h^2}\left[-2(x-x_i)^{3-\alpha}\right].\end{aligned} \tag{7.100}$$

Then $I_k,\ k=1,2,3,4$ are also obtained as follows:

$$I_1(x) = \frac{3}{h}\left[-\bar{g}_1(x) + 2\bar{g}_2(x) - \bar{g}_3(x)\right], \tag{7.101}$$

$$I_2(x) = \frac{3}{h}\left[\bar{g}_1(x) - 4\bar{g}_2(x) + 3\bar{g}_3(x)\right], \tag{7.102}$$

$$I_3(x) = \frac{3}{h}\left[2\bar{g}_2(x) - 3\bar{g}_3(x)\right], \tag{7.103}$$

$$I_4(x) = \frac{3}{h}\left[\bar{g}_3(x)\right]. \tag{7.104}$$

Therefore,

$$g_1(z) = P_\alpha\,(\alpha-3)\,(\alpha-2)\left((i+z-j-1)^{1-\alpha} - (i+z-j)^{1-\alpha}\right), \tag{7.105}$$

$$g_2(z) = P_\alpha\left((i+z-j-1)^{1-\alpha}\,(\alpha-1-i-z+j) + (i+z-j)^{2-\alpha}\right)(\alpha-3), \tag{7.106}$$

$$\begin{aligned}g_3(z) = P_\alpha\big[&\left(\alpha^2 - \alpha\,(3+2\,i+2\,z-2\,j) + 6\,i + 6\,z - 6\,j + 2\,(i+z-j-1)^2\right)\\ &(i+z-j-1)^{1-\alpha} - 2\,(i+z-j)^{3-\alpha}\big].\end{aligned} \tag{7.107}$$

$$\bar{g}_1(z) = P_\alpha(-\,(\alpha-2)\,(\alpha-3)\,z^{1-\alpha}), \tag{7.108}$$

$$\bar{g}_2(z) = P_\alpha((\alpha-3)\,z^{2-\alpha}), \tag{7.109}$$

$$\bar{g}_3(z) = P_\alpha(-2\,z^{3-\alpha}). \tag{7.110}$$

where $P_\alpha = \frac{h^{1-\alpha}}{(\alpha-1)(\alpha-2)(\alpha-3)}$.

$$\begin{aligned}J_1(z) = R_\alpha\big[&-(\alpha-3)\,(\alpha-2)\left((i+z-j-1)^{1-\alpha} - (i+z-j)^{1-\alpha}\right)\\ &+2\left((i+z-j-1)^{1-\alpha}\,(\alpha-1-i-z+j) + (i+z-j)^{2-\alpha}\right)(\alpha-3)\\ &-\left(\alpha^2 - \alpha\,(3+2\,i+2\,z-2\,j) + 6\,i + 6\,z - 6\,j + 2\,(i+z-j-1)^2\right)\\ &\times(i+z-j-1)^{1-\alpha} + 2\,(i+z-j)^{3-\alpha}\big],\end{aligned} \tag{7.111}$$

$$J_2(z) = R_\alpha \left[(\alpha-3)(\alpha-2)\left((i+z-j-1)^{1-\alpha} - (i+z-j)^{1-\alpha}\right) \right.$$
$$-4\left((i+z-j-1)^{1-\alpha}(\alpha-1-i-z+j) + (i+z-j)^{2-\alpha}\right)(\alpha-3)$$
$$+3\left(\alpha^2 - \alpha\,(3+2\,i+2\,z-2\,j) + 6\,i+6\,z-6\,j+2\,(i+z-j-1)^2\right)$$
$$\left. \times (i+z-j-1)^{1-\alpha} - 6\,(i+z-j)^{3-\alpha}\right], \tag{7.112}$$

$$J_3(z) = R_\alpha \left[2\left((i+z-j-1)^{1-\alpha}(\alpha-1-i-z+j) + (i+z-j)^{2-\alpha}\right)(\alpha-3)\right.$$
$$-3\left(\alpha^2 - \alpha\,(3+2\,i+2\,z-2\,j) + 6\,i+6\,z-6\,j+2\,(i+z-j-1)^2\right)$$
$$\left. \times (i+z-j-1)^{1-\alpha} + 6\,(i+z-j)^{3-\alpha}\right], \tag{7.113}$$

$$J_4(z) = R_\alpha \left[\left(\alpha^2 - \alpha\,(3+2\,i+2\,z-2\,j) + 6\,i+6\,z-6\,j\right.\right.$$
$$\left.\left. +2\,(i+z-j-1)^2\right)(i+z-j-1)^{1-\alpha} - 2\,(i+z-j)^{3-\alpha}\right]. \tag{7.114}$$

$$I_1(z) = R_\alpha \left[(\alpha-2)(\alpha-3)\,z^{1-\alpha} + 2\,(\alpha-3)\,z^{2-\alpha} + 2\,z^{3-\alpha}\right], \tag{7.115}$$

$$I_2(z) = R_\alpha \left[-(\alpha-2)(\alpha-3)\,z^{1-\alpha} - 4\,(\alpha-3)\,z^{2-\alpha} - 6\,z^{3-\alpha}\right], \tag{7.116}$$

$$I_3(z) = R_\alpha \left[2\,(\alpha-3)\,z^{2-\alpha} + 6\,z^{3-\alpha}\right], \tag{7.117}$$

$$I_4(z) = R_\alpha \left[-2\,z^{3-\alpha}\right], \tag{7.118}$$

where $R_\alpha = \frac{3h^{-\alpha}}{(\alpha-1)(\alpha-2)(\alpha-3)}$.

$$D_x^\alpha u(x_i + hz) = \frac{1}{\Gamma(1-\alpha)} \left[\sum_{j=1}^{i-1} \sum_{k=1}^{4} b_{k+3(j-1)} J_k(z) + \sum_{k=1}^{4} b_{k+3(i-1)} I_k(z) \right], \tag{7.119}$$
$$i = 1, 2, 3 \ldots, N.$$

Let

$$J_k^* = \frac{1}{R_\alpha} J_k,$$
$$I_k^* = \frac{1}{R_\alpha} I_k \quad k = 1, 2, 3, 4. \tag{7.120}$$

Hence

$$D_x^\alpha u(x_i + hz) = \frac{R_\alpha}{\Gamma(1-\alpha)} \left[\sum_{j=1}^{i-1} \sum_{k=1}^{4} b_{k+3(j-1)} J_k^*(z) + \sum_{k=1}^{4} b_{k+3(i-1)} I_k^*(z) \right],$$
$$= \frac{-3h^{-\alpha}}{\Gamma(4-\alpha)} \left[\sum_{j=1}^{i-1} \sum_{k=1}^{4} b_{k+3(j-1)} J_k^*(z) + \sum_{k=1}^{4} b_{k+3(i-1)} I_k^*(z) \right], \quad (7.121)$$
$$i = 1, 2, 3 \ldots, N.$$

We shall now use the result of this derivation in the example below.

Example 7.6: Consider the space fractional KdV equation

$$\frac{\partial u}{\partial t} = -\epsilon u \frac{\partial^\alpha u}{\partial x^\alpha} - \mu \frac{\partial^3 u}{\partial x^3} + r(x,t), \quad 0 < \alpha < 1, \quad (7.122)$$

with $\epsilon = 1$, $\mu = 1$, $r(x,t) = 6t + x^3 + \frac{6t^2 x^{6-\alpha}}{\Gamma(4-\alpha)}$, the initial condition

$$u(x,0) = 0, \quad (7.123)$$

and boundary conditions

$$u(0,t) = 0, \quad u(1,t) = t, \quad u_x(1,t) = 3t. \quad (7.124)$$

The exact solution to the fractional KdV equation is

$$u(x,t) = tx^3. \quad (7.125)$$

Integrating equation (7.122) from t_n to t_{n+1}, applying the trapezoidal rule and linearizing results in

$$u_i^{n+1} + \frac{\epsilon \Delta t}{2} D_x^\alpha u_i^n u_i^{n+1} + \frac{\epsilon \Delta t}{2} u_i^n D_x^\alpha u_i^{n+1} + \frac{\mu \Delta t}{2} u_{xxx,i}^{n+1} = u_i^n - \frac{\mu \Delta t}{2} u_{xxx,i}^n + \frac{\Delta t}{2}(r_i^n + r_i^{n+1}), \quad (7.126)$$

where n represents discretization in time and i represents discretization in space. We replace the fractional derivatives in equation (7.126) with (7.121). We then evaluate the resulting equation at $z = 0.5$ for $i = 1, 2, \ldots, N$. These equations

combined with continuity equations (7.10) and (7.11) and the boundary conditions in (7.124) give a linear system for each time interval that has a unique solution.

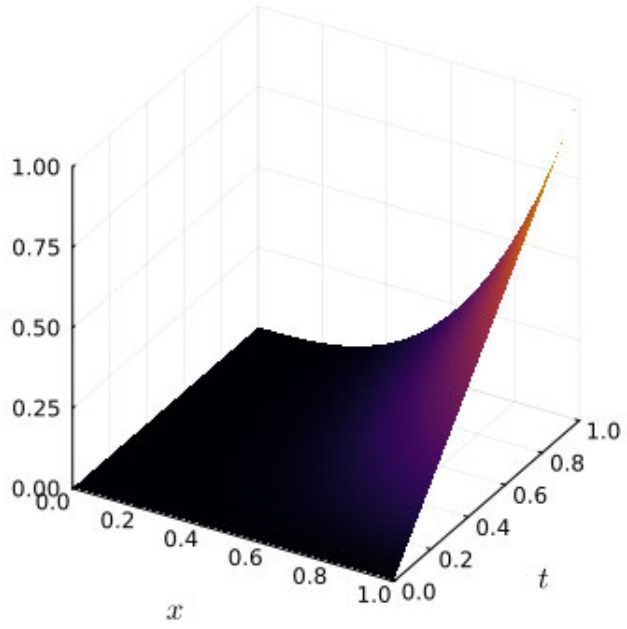

Figure 7.18: 3D plot of the approximate solution for example 7.6 when $\alpha = 0.5,\ N = 50$ and $\Delta t = 0.02$.

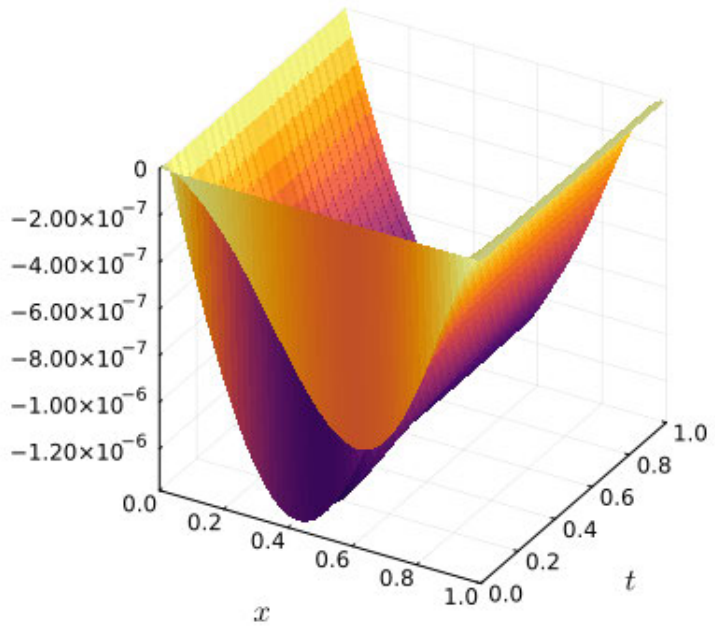

Figure 7.19: 3D plot of error for example 7.6 when $\alpha = 0.5,\ N = 50$ and $\Delta t = 0.02$.

Figures 7.18 and 7.19 show the approximate solution and error respectively, when $\alpha = 0.5$.

7.8 Discretization of the time fractional differential equation of order $0 < \beta < 1$

Suppose the time interval $[t_0, t_f]$ is divided into N partitions such that $t_0 = t_1 < t_2 < \ldots < t_{N+1}$ with $\Delta t = \frac{t_f - t_0}{N}$. The Caputo time derivative of order β is

$$D_t^\beta u(x,t) = \frac{\partial^\beta u(x,t)}{\partial t^\beta} = \frac{1}{\Gamma(m-\beta)} \int_{t_1}^{t} (t-\eta)^{m-\beta-1} \frac{\partial^m u(x,\eta)}{\partial \eta^m} d\eta, \tag{7.127}$$

where $m-1 < \beta < m$, $m \in \mathbb{N}$. We now follow the approach of [36]. At the point t_{i+1}, $i = 1, 2 \ldots, N$, equation (7.127) becomes

$$\begin{aligned} D_t^\beta u(x,t_{i+1}) &= \frac{1}{\Gamma(m-\beta)} \int_{t_1}^{t_{i+1}} (t_{i+1}-\eta)^{m-\beta-1} \frac{\partial^m u(x,\eta)}{\partial \eta^m} d\eta, \quad m-1 < \beta < m, \\ &= \frac{1}{\Gamma(m-\beta)} \sum_{j=1}^{i} \int_{t_j}^{t_{j+1}} (t_{i+1}-\eta)^{m-\beta-1} \frac{\partial^m u(x,\eta)}{\partial \eta^m} d\eta. \end{aligned} \tag{7.128}$$

When $m = 1$, we have

$$D_t^\beta u(x,t_{i+1}) = \frac{1}{\Gamma(1-\beta)} \sum_{j=1}^{i} \int_{t_j}^{t_{j+1}} (t_{i+1}-\eta)^{-\beta} \frac{\partial u(x,\eta)}{\partial \eta} d\eta, \quad 0 < \beta < 1. \tag{7.129}$$

We replace the derivative inside the integral with its forward difference formula.

This gives

$$
\begin{aligned}
D_t^{\beta}u(x,t_{i+1}) &= \frac{1}{\Gamma(1-\beta)}\sum_{j=1}^{i}\int_{t_j}^{t_{j+1}}(t_{i+1}-\eta)^{-\beta}\frac{u(x,t_{j+1})-u(x,t_j)}{\Delta t}d\eta,\\
&= \frac{1}{\Gamma(1-\beta)}\left[\sum_{j=1}^{i-1}\frac{u(x,t_{j+1})-u(x,t_j)}{\Delta t}\int_{t_j}^{t_{j+1}}(t_{i+1}-\eta)^{-\beta}d\eta\right.\\
&\left.+\frac{u(x,t_{i+1})-u(x,t_i)}{\Delta t}\int_{t_i}^{t_{i+1}}(t_{i+1}-\eta)^{-\beta}d\eta\right],\\
&= \sum_{j=1}^{i-1}\frac{u(x,t_{j+1})-u(x,t_j)}{\Delta t\Gamma(1-\beta)}\left(\frac{(t_{i+1}-t_{j+1})^{-\beta+1}-(t_{i+1}-t_j)^{-\beta+1}}{\beta-1}\right)\\
&+\frac{u(x,t_{i+1})-u(x,t_i)}{\Delta t\Gamma(1-\beta)}\left(\frac{(t_{i+1}-t_i)^{-\beta+1}}{1-\beta}\right),
\end{aligned}
\tag{7.130}
$$

$$
\begin{aligned}
D_t^{\beta}u(x,t_{i+1}) &= \sum_{j=1}^{i-1}\left[\frac{u(x,t_{j+1})-u(x,t_j)}{\Gamma(2-\beta)}\right]\left(\frac{(i-j+1)^{1-\beta}-(i-j)^{1-\beta}}{\Delta t^{\beta}}\right)\\
&+\left[\frac{u(x,t_{i+1})-u(x,t_i)}{\Delta t^{\beta}\Gamma(2-\beta)}\right].
\end{aligned}
\tag{7.131}
$$

Therefore equation (7.131) can be concisely written as

$$
\begin{aligned}
D_t^{\beta}u(x,t_{i+1}) &= \frac{\Delta t^{-\beta}}{\Gamma(2-\beta)}\left[u(x,t_{i+1})-u(x,t_i)\right.\\
&\left.+\sum_{j=1}^{i-1}\left[u(x,t_{j+1})-u(x,t_j)\right]\left((i-j+1)^{1-\beta}-(i-j)^{1-\beta}\right)\right],
\end{aligned}
\tag{7.132}
$$

$$i = 1, 2\ldots, N.$$

This means that when $i = 1$, we ignore the term of equation (7.132) that contains the summation sign. We shall now apply the result of this derivation in the next example.

Example 7.7: Consider the time fractional KdV equation

$$\frac{\partial^{\beta}u}{\partial t^{\beta}} = -\epsilon u\frac{\partial u}{\partial x} - \mu\frac{\partial^3 u}{\partial x^3} + f(x,t), \quad 0 < \beta < 1, \tag{7.133}$$

with $\epsilon = 1$, $\mu = 1$, $f(x,t) = 6t + 3t^2x^5 + \frac{t^{1-\alpha}x^3}{\Gamma(2-\alpha)}$ the initial condition

$$u(x,0) = 0 \tag{7.134}$$

and boundary conditions

$$u(0,t) = 0, \quad u(1,t) = t, \quad u_x(1,t) = 3t. \tag{7.135}$$

The exact solution to this problem is

$$u(x,t) = tx^3. \tag{7.136}$$

Integrate (7.133) over $[t_n, t_{n+1}]$ to obtain

$$(D_t^\beta u)^{n+1}\Delta t = \Delta t\left[-\frac{\epsilon}{2}u_x^n u^{n+1} - \frac{\epsilon}{2}u^n u_x^{n+1} - \frac{\mu}{2}u_{xxx}^{n+1} - \frac{\mu}{2}u_{xxx}^n\right] + f^{n+1}\Delta t, \tag{7.137}$$

Here we have used the right hand rectangular rule for the first and last terms and trapezoidal rule for the rest which is $O(\Delta t^2)$. Further linearizing the non-linear term in time at x_i results in

$$D_t^\beta u(x_i, t_{n+1}) + \frac{\epsilon}{2}u_{x,i}^n u_i^{n+1} + \frac{\epsilon}{2}u_i^n u_{x,i}^{n+1} + \frac{\mu}{2}u_{xxx,i}^{n+1} = -\frac{\mu}{2}u_{xxx,i}^n + f_i^{n+1}, \tag{7.138}$$

where n and i represent discretization in time and space, respectively. We now replace the fractional derivative on the left hand side of (7.138) with

$$\begin{aligned} D_t^\beta u(x_i, t_{n+1}) = \frac{\Delta t^{-\beta}}{\Gamma(2-\beta)}\Bigg[&u(x,t_{n+1}) - u(x,t_n) \\ &+ \sum_{j=1}^{n-1}\left[u(x_i,t_{j+1}) - u(x_i,t_j)\right]\left((n-j+1)^{1-\beta} - (n-j)^{1-\beta}\right)\Bigg], \\ &n = 1, 2\ldots, Nt, \quad i = 1, 2\ldots, N. \end{aligned} \tag{7.139}$$

We obtain

$$\begin{aligned} \frac{\Delta t^{-\beta}}{\Gamma(2-\beta)}u_i^{n+1} + \frac{\epsilon}{2}u_{x,i}^n u_i^{n+1} + \frac{\epsilon}{2}u_i^n u_{x,i}^{n+1} + \frac{\mu}{2}u_{xxx,i}^{n+1} &= \frac{\Delta t^{-\beta}}{\Gamma(2-\beta)}u_i^n - \frac{\mu}{2}u_{xxx,i}^n \\ &- \frac{\Delta t^{-\beta}}{\Gamma(2-\beta)}\sum_{j=1}^{n-1}\left(u_i^{j+1} - u_i^j\right)\left[(n-j+1)^{1-\beta} - (n-j)^{1-\beta}\right] \\ &+ f(x_i + hz, t_{n+1}), \\ n = 1, 2\ldots, Nt, \quad i = 1, 2\ldots, N. \end{aligned} \tag{7.140}$$

Using the transformation $z = \frac{x-x_i}{h}$, we have

$$\begin{aligned}
&\sum_{k=1}^{4}\left(\left[\frac{\Delta t^{-\beta}}{\Gamma(2-\beta)}+\frac{\epsilon}{2h}\sum_{k=1}^{4}b_{k+3(i-1)}(t_n)\ B_k'(z)\right]B_k(z)+\frac{\mu}{2h^3}\ B_k'''(z)\right.\\
&\left.+\frac{\epsilon}{2h}\left[\sum_{k=1}^{4}b_{k+3(i-1)}(t_n)B_k(z)\right]\ B_k'(z)\right)b_{k+3(i-1)}(t_{n+1}),\\
&=\sum_{k=1}^{4}\left[\frac{\Delta t^{-\beta}}{\Gamma(2-\beta)}B_k(z)-\frac{\mu}{2h^3}\ B_k'''(z)\right]b_{k+3(i-1)}(t_n)\\
&-\frac{\Delta t^{-\beta}}{\Gamma(2-\beta)}\sum_{j=1}^{n-1}\left(\sum_{k=1}^{4}b_{k+3(i-1)}(t_{j+1})\ B_k(z)-\sum_{k=1}^{4}b_{k+3(i-1)}(t_j)\ B_k(z)\right)\\
&\times\left[(n-j+1)^{1-\beta}-(n-j)^{1-\beta}\right]+f(x_i+hz,t_{n+1}),\\
&n=1,2\ldots,Nt,\quad i=1,2,\ldots,N,
\end{aligned}\tag{7.141}$$

where $h = x_{i+1}-x_i$, N and Nt are number of intervals on x and t, respectively. We evaluate equation (7.141) at $z = 0.5$ for $i = 1, 2, \ldots, N$. These equations combined with the continuity equations (7.10) and (7.11) and the boundary conditions in equation (7.135) give a linear system for each time interval that has a unique solution.

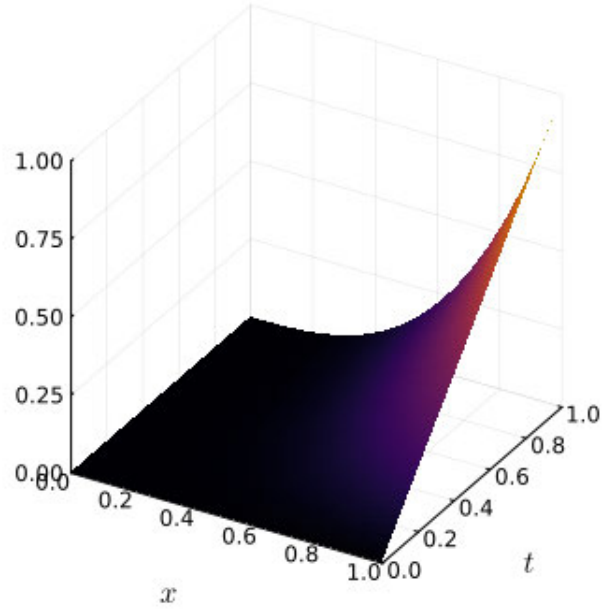

Figure 7.20: 3D plot of solution for 7.7 when $N = 50,\ \beta = 0.5$ and $\Delta t = 0.02$.

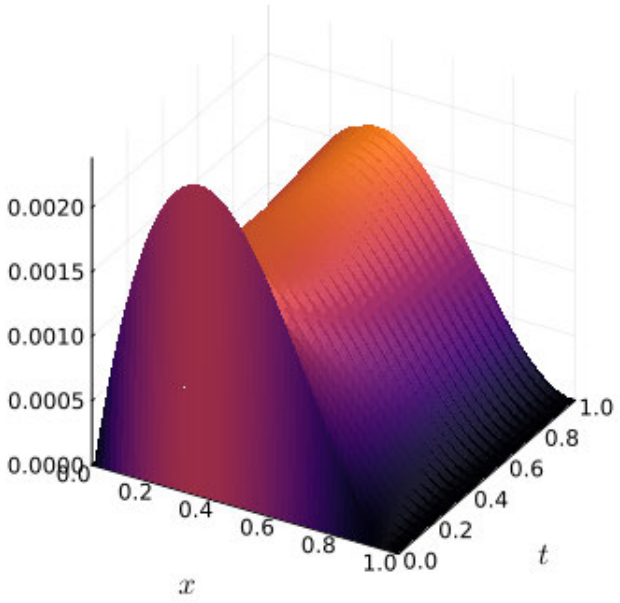

Figure 7.21: 3D plot of error for 7.7 when $N = 50,\ \beta = 0.5$ and $\Delta t = 0.02$.

Figures 7.20 and 7.21 display the approximate solution and errors respectively, when $\beta = 0.5$. The fractional time derivative becomes the classical KdV equation when $\beta = 1$. The absolute errors are less than or equal to 8×10^{-4} in Figure 7.21. This shows that cubic B-spline OCFE can be used to solve time fractional partial differential equations.

7.9 Discussion of Chapter 7

In this chapter we have demonstrated that the cubic B-spline OCFE can solve ordinary and partial and fractional differential equations. We also showed that cubic B-spline OCFE with Gauss' points as collocation point is better than the quadratic B-spline OCFE. To the best of our knowledge, this is the first time that cubic B-splines OCFE method has been used to solve fractional ODEs and PDEs. The next chapter concludes the thesis.

CHAPTER EIGHT

CONCLUSION

In this thesis, we derive and apply the orthogonal collocation on finite elements method (OCFE) using the quadratic and cubic B-spline basis to obtain the numerical solution of both ordinary and fractional ordinary and partial differential equations. We solved linear and nonlinear ordinary differential equations with the quadratic OCFE. The nonlinear ODEs were linearized and integrated via the Crank-Nicolson technique. We applied the quadratic OCFE to solve Burgers' equation and the modified Burgers' equation.

Moreover, we extend the quadratic OCFE to accommodate non-uniform intervals and use the Schrödinger equation as a case study. This leads to a coupled system of nonlinear ODEs. We obtain results that are consistent with the exact solutions for different soliton cases and show that the quadratic OCFE is suitable for coupled ordinary differential equations. We discuss the consistency, stability and convergence of the quadratic B-spline OCFE with respect to the aforementioned applications.

Furthermore, we derive the fractional order derivative matrix based on the quadratic B-spline OCFE to solve fractional differential equations of order α, $1 < \alpha < 2$. We demonstrate its applicability to linear and nonlinear fractional ODEs and fractional diffusion equations including the Fisher's equation. We also discuss the consistency, stability and convergence of the fractional case of the quadratic B-spline OCFE. In this case we find that the solution is $O(h^{3-\alpha})$.

In addition, we apply a modified form of the quadratic B-spline basis functions to

solve two-dimensional partial differential equations. This approach minimizes the number of unknowns in $2D$. We obtain results that are in agreement with exact solutions and previous ones in the literature for examples on time-dependent and time-independent two-dimensional partial differential equations. We also show that the OCFE method based on quadratic B-splines has convergence order of two for both one-dimensional and two-dimensional partial differential equations.

We also extend our work to orthogonal collocation on finite elements using cubic B-spline basis functions. We applied it to linear and nonlinear ODEs, the Burgers', KdV, KdV-Burgers' equations and a version of KdV-Burgers' equation whose exact solution is not known. We demonstrated the efficiency of cubic B-splines OCFE to solve space fractional partial differential equations of orders $0 < \alpha < 1$ and $1 < \alpha < 2$, and time fractional partial differential equation of order $0 < \beta < 1$ effectively.

The main features and merits of the OCFE method is it's simplicity in implementation, computational efficiency (enhanced memory storage and CPU time) due to the sparse structure of the matrices arising from the application of the B-spline basis functions, adaptability and flexibilty due to the implementation using finite elements. This makes the OCFE method useful for solving problems whose solution involve steep gradients. We used the Julia programming language v1.11 [11] for numerical computations. The numerical results compared favourably with the existing ones in the literature.

8.1 Future Research

The following are recommended for future research:

1. Cubic B-spline orthogonal collocation on finite elements method for space and time fractional partial differential equations.

2. Numerical solution of two-dimensional partial differential equations with cubic B-spline orthogonal collocation on finite elements method.

3. Solution of time fractional diffusion equation using quadratic B-splines and compact finite differences.

4. Solution of time fractional diffusion equation using cubic B-splines and compact finite differences.

REFERENCES

[1] Abdullah M., Yaseen M., and De la Sen M. (2022), *An efficient collocation method based on Hermite formula and cubic B-splines for numerical solution of the Burgers' equation*, Mathematics and Computers in Simulation, Vol. 197, , pp. 166–184.

[2] Adel W., Rezazadeh H., and Inc M. (2024), *On numerical solutions of telegraph, viscous, and modified Burgers equations via bernoulli collocation method*, Scientia Iranica, Vol. 31, No. 1, pp. 43–54.

[3] Alam M. P., Kumar D., and Khan A. (2021), *Trigonometric quintic B-spline collocation method for singularly perturbed turning point boundary value problems*, International Journal of Computer Mathematics, Vol. 98, No. 5, pp. 1029-1048.

[4] Ali A. H. A., Gardner G. A., and Gardner L. R. T. (1992), *A collocation solution for Burgers' equation using cubic B-spline finite elements*, Computer Methods in Applied Mechanics and Engineering, Vol. 100, No. 3, pp. 325-337.

[5] Arora S., and Kaur I. (2018), *Applications of Quintic Hermite collocation with time discretization to singularly perturbed problems*, Applied Mathematics and Computation, Vol. 316, , pp. 409-421.

[6] Bellman R., and Casti J. (1971), *Differential quadrature and long-term integration*, Journal of Mathematical Analysis and Applications, Vol. 34, No. 2, pp. 235-238.

[7] Bellman R., Kashef B. G., and Casti J. (1972), *Differential quadrature: A technique for the rapid solution of nonlinear partial differential equations*, Journal of Computational Physics, Vol. 10, No. 1, pp. 40-52.

[8] Bellman R., Kashef B., Lee E. S., and Vasudevan R. (1975), *Differential quadrature and splines*, Computers & Mathematics with Applications, Vol. 1, No. 3–4, pp. 371-376.

[9] Benson D. A., Huntington G. T., Thorvaldsen T. P., and Rao A. V. (2006), *Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method*, Journal of Guidance, Control, and Dynamics, Vol. 29, No. 6, pp. 1435-1440.

[10] Bert C. W., and Malik M. (1996), *Differential quadrature method in computational mechanics: A review*, Applied Mechanics Reviews, Vol. 49, No. 1, pp. 1-28.

[11] Bezanson J., Edelman A., Karpinski S., and Shah V. B. (2017), *Julia: A Fresh Approach to Numerical Computing*, SIAM Review, Vol. 59, No. 1, pp. 65–98.

[12] Bialecki B., Fairweather G., Karageorghis A., and Maack J. (2020), *A quadratic spline collocation method for the Dirichlet biharmonic problem*, Numerical Algorithms, Vol. 83, No. 1, pp. 165-199.

[13] Biegler L. T. (1984), *Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation*, Computers & Chemical Engineering, Vol. 8, No. 3, pp. 243-247.

[14] Botella O., and Shariff K. (2003), *B-spline methods in fluid dynamics*, International Journal of Computational Fluid Dynamics, Vol. 17, No. 2, pp. 133-149.

[15] Bratsos A. G., and Khaliq A. Q. M. (2018), *An exponential time differencing method of lines for the Burgers' and the modified Burgers' equations*, Numerical Methods for Partial Differential Equations, Vol. 34, No. 6, pp. 2024-2039.

[16] Cardone A., Conte D., D'Ambrosio R., and Paternoster B. (2018), *Collocation Methods for Volterra Integral and Integro-Differential Equations: A Review*, Axioms, Vol. 7, No. 3, pp. 45.

[17] Carey G. F., and Finlayson B. A. (1975), *Orthogonal collocation on finite elements*, Chemical Engineering Science, Vol. 30, No. 5, pp. 587-596.

[18] Celikten G. (2021), *Numerical solutions of the modified Burgers' equation by explicit logarithmic finite difference schemes*, Sohag Journal of Mathematics, Vol. 8, No. 3, pp. 73–79.

[19] Ciarlet P. G., Schultz M. H., and Varga R. S. (1967), *Numerical methods of high-order accuracy for nonlinear boundary value Problems*, Numerische Mathematik, Vol. 9, No. 5, pp. 394-430.

[20] De Boor C., and Swartz B. (1973), *Collocation at Gaussian points*, SIAM Journal on numerical analysis, Vol. 10, No. 4, pp. 582-606.

[21] Dhiman N., Huntul M. J., and Tamsir M. (2021), *A modified trigonometric cubic B-spline collocation technique for solving the time-fractional diffusion*

equation, Engineering Computations, Vol. 38, No. 7, pp. 2921-2936.

[22] Douglas J. Jr., Dupont T., and Wheeler M. F. (1974), *H1-Galerkin methods for the Laplace and heat equations*, In: Mathematical aspects of finite elements in partial differential equations; Proceedings of the Symposium. p. 383-416.

[23] Douglas J., and Dupont T. (1973), *A finite element collocation method for quasilinear parabolic equations*, Mathematics of Computation, Vol. 27, No. 121, pp. 17-28.

[24] El-Mikkawy M. (2003), *A note on a three-term recurrence for a tridiagonal matrix*, Applied Mathematics and Computation, Vol. 139, No. 2, pp. 503-511.

[25] Fairweather G., Karageorghis A., and Maack J. (2011), *Compact optimal quadratic spline collocation methods for the Helmholtz equation*, Journal of Computational Physics, Vol. 230, No. 8, pp. 2880-2895.

[26] Finlayson B. A. (1974), *Orthogonal Collocation in Chemical Reaction Engineering*, Catalysis Reviews, Vol. 10, No. 1, pp. 69-138.

[27] Frazer R. A., Jones W. P., and Skan S. W. (1937), *Approximations to functions and to the solutions of differential equations*, Britain Aerospace Research Council London. Report and Memo, No. 1799.

[28] Ganaie I. A., and Kukreja V. K. (2014), *Numerical solution of Burgers' equation by cubic Hermite collocation method*, Applied Mathematics and Computation, Vol. 237, , pp. 571-581.

[29] Hallet P., Hennart J. P., and Mund E. H. (1976), *A Galerkin method with mod-*

ified piecewise polynomials for solving a second-order boundary value problem, Numerische Mathematik, Vol. 27, No. 1, pp. 11-20.

[30] Hepson O. E., Korkmaz A., and Dag I. (2020), *Exponential B-spline collocation solutions to the Gardner equation*, International Journal of Computer Mathematics, Vol. 97, No. 4, pp. 837-850.

[31] Irk D. (2009), *Sextic B-spline collocation method for the modified Burgers' equation*, Kybernetes. The International Journal of Cybernetics, Systems and Management Sciences, Vol. 38, No. 9, pp. 1599-1620.

[32] Jena S. R., and Gebremedhin G. S. (2023), *Decatic B-spline collocation scheme for approximate solution of Burgers' equation*, Numerical Methods for Partial Differential Equations, Vol. 39, No. 3, pp. 1851-1869.

[33] Johnson R. W. (2005), *Higher order B-spline collocation at the Greville abscissae*, Applied Numerical Mathematics, Vol. 52, No. 1, pp. 63-75.

[34] Kadalbajoo M. K., and Arora P. (2009), *B-spline collocation method for the singular-perturbation problem using artificial viscosity*, Computers & Mathematics with Applications, Vol. 57, No. 4, pp. 650-663.

[35] Kantorovich L. V., and Krylov V. I. (1934), *On a method for the approximate solution of partial differential equations*, Reports of the USSR Academy of Sciences, Vol. 2, No. 9, pp. 532-534.

[36] Khalid N., Abbas M., Iqbal M. K., Singh J., and Md. Ismail A. I. (2020), *A computational approach for solving time fractional differential equation via*

spline functions, Alexandria Engineering Journal, Vol. 59, No. 5, pp. 3061-3078.

[37] Khalifa A. K. A., and Eilbeck J. C. (1982), *Collocation with Quadratic and Cubic Splines*, IMA Journal of Numerical Analysis, Vol. 2, No. 1, pp. 111-121.

[38] Kumari A., and Kukreja V. K. (2022), *Error bounds for septic Hermite interpolation and its implementation to study modified Burgers' equation*, Numerical Algorithms, Vol. 89, No. 4, pp. 1799-1821.

[39] Kumari A., and Kukreja V. K. (2023), *Study of 4th order Kuramoto-Sivashinsky equation by septic Hermite collocation method*, Applied Numerical Mathematics, Vol. 188, , pp. 88-105.

[40] Lakshmi C., and Awasthi A. (2018), *Robust numerical scheme for nonlinear modified Burgers' equation*, International Journal of Computer Mathematics, Vol. 95, No. 9, pp. 1910-1926.

[41] Lanczos C. (1938), *Trigonometric interpolation of empirical and analytical functions*, Journal of Mathematics and Physics, Vol. 17, No. {1-4}, pp. 123-199.

[42] Layton A. T., Christara C. C., and Jackson K. R. (2006), *Quadratic spline methods for the shallow water equations on the sphere: Collocation*, Mathematics and Computers in Simulation, Vol. 71, No. 3, pp. 187-205.

[43] Li Z. (2012), *B-spline collocation for two dimensional, time-dependent, parabolic PDEs*, [Halifax, Nova Scotia]: Saint Mary's University.

[44] Lin B. (2015), *Septic spline function method for nonlinear Schrödinger equations*, Applicable Analysis, Vol. 94, No. 2, pp. 279-293.

[45] Manzhos S., Ihara M., and Carrington T. (2023), *Using collocation to solve the Schrödinger equation*, Journal of Chemical Theory and Computation, Vol. 19, No. 6, pp. 1641–1656.

[46] Michelsen M. L., and Villadsen J. (1972), *A convenient computational procedure for collocation constants*, The Chemical Engineering Journal, Vol. 4, No. 1, pp. 64-68.

[47] Mittal R. C., and Arora G. (2010), *Quintic B-spline collocation method for numerical solution of the Kuramoto–Sivashinsky equation*, Communications in Nonlinear Science and Numerical Simulation, Vol. 15, No. 10, pp. 2798-2808.

[48] Mittal R. C., and Jain R. K. (2012), *Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method*, Applied Mathematics and Computation, Vol. 218, No. 15, pp. 7839-7855.

[49] Mittal R. C., and Jiwari R. (2012), *A differential quadrature method for numerical solutions of Burgers'-type equations*, International Journal of Numerical Methods for Heat & Fluid Flow, Vol. 22, No. 7, pp. 880-895.

[50] Mostafa D., Zaky M. A., Hafez R. M., Hendy A. S., Abdelkawy M. A., and Aldraiweesh A. A. (2023), *Tanh Jacobi spectral collocation method for the numerical simulation of nonlinear Schrödinger equations on unbounded domain*, Mathematical Methods in the Applied Sciences, Vol. 46, No. 1, pp. 656–674.

[51] Orszag S. A. (1972), *Comparison of Pseudospectral and Spectral Approximation*, Studies in Applied Mathematics, Vol. 51, No. 3, pp. 253-259.

[52] Parumasur N., Adetona R. A., and Singh P. (2023), *Efficient solution of Burgers', modified Burgers' and KdV-Burgers' equations using B-spline approximation functions*, Mathematics, Vol. 11, No. 8, pp. 1–21.

[53] Pirim N. A., and Ayaz F. (2016), *A new technique for solving fractional order systems: Hermite collocation method*, Applied Mathematics, Vol. 07, No. 18, pp. 2307-2323.

[54] Ramadan M. A., and El-Danaf T. S. (2005), *Numerical treatment for the modified Burgers' equation*, Mathematics and Computers in Simulation, Vol. 70, No. 2, pp. 90-98.

[55] Rao S. C. S., and Kumar M. (2008), *Exponential B-spline collocation method for self-adjoint singularly perturbed boundary value problems*, Applied Numerical Mathematics, Vol. 58, No. 10, pp. 1572-1581.

[56] Raslan K. R. (2003), *A collocation solution for Burgers' equation using quadratic B-spline finite elements*, International Journal of Computer Mathematics, Vol. 80, No. 7, pp. 931-938.

[57] Raslan K. R., and Ali K. K. (2021), *On n-dimensional quadratic B-splines*, Numerical Methods for Partial Differential Equations, Vol. 37, No. 2, pp. 1057-1071.

[58] Robinson M. P., and Fairweather G. (1994), *Orthogonal spline collocation*

methods for Schrödinger-type equations in one space variable, Numerische Mathematik, Vol. 68, No. 3, pp. 355-376.

[59] Saadatmandi A., and Dehghan M. (2011), *A tau approach for solution of the space fractional diffusion equation*, Computers & Mathematics with Applications, Vol. 62, No. 3, pp. 1135-1142.

[60] Saka B., and Dağ Î. (2007), *Quartic B-spline collocation method to the numerical solutions of the Burgers' equation*, Chaos, Solitons & Fractals, Vol. 32, No. 3, pp. 1125-1137.

[61] Shyaman V. P., Sreelakshmi A., and Awasthi A. (2022), *An adaptive tailored finite point method for the generalized Burgers' equations*, Journal of Computational Science, Vol. 62, , pp. 101744.

[62] Singh B. K., and Gupta M. (2022), *Trigonometric tension B-spline collocation approximations for time fractional Burgers' equation*, Journal of Ocean Engineering and Science, Vol. 9, No. 5, pp. 508–516.

[63] Singh P., Parumasur N., and Bansilal C. (2021), *Orthogonal collocation on finite elements using quintic Hermite basis*, Australian Journal Of Mathematical Analysis And Applications, Vol. 18, No. 2, pp. 1–12.

[64] Singh P., Parumasur N., and Singh S. (2022), *A review of collocation approximations to solutions of differential equations*, Mathematics, Vol. 10, No. 23, pp. 1–22.

[65] Siraj-ul-Islam, Noor M. A., Tirmizi I. A., and Khan M. A. (2006), *Quadratic*

non-polynomial spline approach to the solution of a system of second-order boundary-value problems, Applied Mathematics and Computation, Vol. 179, No. 1, pp. 153-160.

[66] Soliman A. A. (2004), *Collocation solution of the Korteweg–de Vries equation using septic splines*, International Journal of Computer Mathematics, Vol. 81, No. 3, pp. 325-331.

[67] Soliman A. A. (2006), *A numerical simulation and explicit solutions of KdV-Burgers' and Lax's seventh-order KdV equations*, Chaos, Solitons & Fractals, Vol. 29, No. 2, pp. 294-302.

[68] Sousa E. (2011), *Numerical approximations for fractional diffusion equations via splines*, Computers & Mathematics with Applications, Vol. 62, No. 3, pp. 938-944.

[69] Stewart W. E. (1984), *Simulation and estimation by orthogonal collocation*, Chemical Engineering Education, Vol. 18, No. 4, pp. 204-212.

[70] Tian Z., Zhai S., Ji H., and Weng Z. (2021), *A compact quadratic spline collocation method for the time-fractional Black-Scholes model*, Journal of Applied Mathematics and Computing, Vol. 66, No. 1, pp. 327-350.

[71] user17762 (2015), *How to compute the determinant of a tridiagonal matrix with constant diagonals?*, https://math.stackexchange.com/q/267466

[72] Villadsen J. V., and Stewart W. E. (1967), *Solution of boundary-value problems by orthogonal collocation*, Chemical Engineering Science, Vol. 22, No. 11,

pp. 1483-1501.

[73] Yalçinbaş S., Aynigül M., and Sezer M. (2011), *A collocation method using Hermite polynomials for approximate solution of pantograph equations*, Journal of the Franklin Institute, Vol. 348, No. 6, pp. 1128-1139.

[74] Yaseen M., Abbas M., Ismail A. I., and Nazir T. (2017), *A cubic trigonometric B-spline collocation approach for the fractional sub-diffusion equations*, Applied Mathematics and Computation, Vol. 293, , pp. 311-319.

[75] Young L. C. (2019), *Orthogonal collocation revisited*, Computer Methods in Applied Mechanics and Engineering, Vol. 345, , pp. 1033-1076.

[76] Zaki S. I. (2000), *A quintic B-spline finite elements scheme for the KdVB equation*, Computer Methods in Applied Mechanics and Engineering, Vol. 188, No. 1, pp. 121-134.

[77] Zhu X., Nie Y., Yuan Z., Wang J., and Yang Z. (2017), *An exponential B-spline collocation method for the fractional sub-diffusion equation*, Advances in Difference Equations, Vol. 2017, No. 285, pp. 1–17.