

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**SOLVING DIFFERENTIAL EQUATIONS USING
NUMERICAL METHODS: DIFFERENTIAL QUADRATURE**



Gülsemay YİĞİT

DOCTOR OF PHILOSOPHY THESIS
Department of Mathematical Engineering
Mathematical Engineering Program

Advisor
Prof. Dr. Mustafa BAYRAM

July, 2019

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**SOLVING DIFFERENTIAL EQUATIONS USING NUMERICAL
METHODS: DIFFERENTIAL QUADRATURE**

A thesis submitted by GÜLSEMAY YİĞİT in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY** is approved by the committee on 09.07.2019 in Department of Mathematical Engineering, Mathematical Engineering Program.

Prof. Dr. Mustafa BAYRAM
Biruni University
Advisor

Approved By the Examining Committee

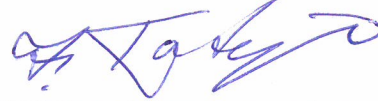
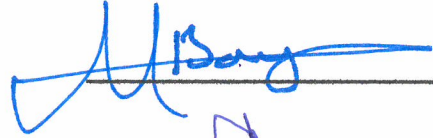
Prof. Dr. Mustafa BAYRAM, Advisor
Biruni University

Prof. Dr. Mustafa SİVRİ, Member
Yıldız Technical University

Prof. Dr. Ünsal TEKİR, Member
Marmara University

Prof. Dr. Samet Yücel KADIOĞLU, Member
Istanbul Technical University

Assoc. Prof. Dr. Aydın SEÇER, Member
Yıldız Technical University



I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Solving Differential Equations using numerical methods: Differential Quadrature, supervised by my supervisor, Prof. Dr. Mustafa BAYRAM. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Gülsemay YİĞİT

Dedicated to my mother



ACKNOWLEDGEMENTS

I would like to thank my respected supervisor Prof. Dr. Mustafa BAYRAM for his invaluable guidance and support.

I want to thank to the members of the examining committee for taking time out to assess the quality of this work.

I would like to express my special thanks to my colleagues from the Department of Basic Sciences of Altınbaş University for their helpfulness and supports.

I express thanks to all my friends for their encouragement and emotional supports, during the course of this research, especially Deniz Mukaddes TÜRET.

Last but not least, I wish to express my deepest gratitude to all my family members for their supportive encouragement during the entire period my education. Without their supports, this work would not have become possible.

Gülsemay YİĞİT

TABLE OF CONTENTS

LIST OF SYMBOLS	vii
LIST OF ABBREVIATIONS	viii
LIST OF FIGURES	ix
LIST OF TABLES	xi
ABSTRACT	xiii
ÖZET	xv
1 INTRODUCTION	1
1.1 Literature Review	1
1.2 Objective of the Thesis	3
1.3 Hypothesis	3
1.4 Thesis Plan	4
2 METHODOLOGY	5
2.1 The Differential Quadrature Method	5
2.2 Chebyshev Polynomials	7
2.3 Polynomial based Differential Quadrature (PDQ)	9
2.3.1 Computation of Weighting Coefficients for First Order Derivative	9
2.3.2 Weighting Coefficients for Second and Higher Order Derivatives	14
2.3.3 Modification of Weighting Coefficients	16
2.3.4 Matrix Multiplication Approach	17
2.4 Fourth Order Runge-Kutta Algorithm	18
2.5 Error Norms	20
3 APPLICATIONS ON QUADRATURE DISCRETIZATION FOR SOME ORDINARY DIFFERENTIAL EQUATIONS	22
3.1 Quadrature Solution of the First Order Equations	23
3.2 Quadrature Solution of the Second Order Equations	25
3.3 Quadrature Solution of the Higher Order Boundary Value Problems . .	30

3.4	Quadrature Solutions of the Third and Fourth Order Singularly Perturbed Problems	38
3.4.1	Quadrature Discretization of Third Order Perturbation Problems	38
3.4.2	Quadrature Discretization of Fourth Order Perturbation Problems	40
4	APPLICATIONS ON QUADRATURE DISCRETIZATION FOR SOME PARTIAL DIFFERENTIAL EQUATIONS	53
4.1	Quadrature Solution of Linear and Nonlinear Diffusion Equations . . .	53
4.1.1	Discretization of Linear Diffusion Equation	54
4.1.2	Discretization of Nonlinear Diffusion Equation	55
4.2	Quadrature Solution of Reaction-Diffusion Equation in Non-Uniform Media	64
4.3	Quadrature Solution of Kuramoto-Sivashinsky Equation	75
4.4	Quadrature Solution of Nonlinear Advection Equation	84
4.5	Quadrature Solution of Ginzburg Landau Equation	92
5	Conclusion and Recommendations	98
	References	100
	Publications From the Thesis	109

LIST OF SYMBOLS

w_{ij}	First order derivative weighting coefficient
α	Non-homogeneity parameter
$w_{ij}^{(n)}$	n -th order derivative weighting coefficient
ϵ	Perturbation parameter
Δt	Time increment

LIST OF ABBREVIATIONS

BVT	Boundary Value Technique
BVP	Boundary Value Problem
DQ	Differential Quadrature
DQM	Differential Quadrature Method
FGM	Functionally Graded Material
GRE	Global Relative Error
IVT	Initial Value Technique
KS	Kuramoto-Sivashinsky
PDQ	Polynomial Differential Quadrature

LIST OF FIGURES

Figure 2.1	Stability region	20
Figure 3.1	Comparison of solutions for Example 3.4.1, numerical solution (dotted line), exact solution (solid line).	43
Figure 3.2	Comparison of solutions for Example 3.4.2, numerical solution (dotted line), exact solution (solid line).	46
Figure 3.3	Comparison of solutions for Example 3.4.3, numerical solution (dotted line), exact solution (solid line).	48
Figure 3.4	Comparison of solutions for Example 3.4.4, numerical solution (dotted line), exact solution (solid line).	50
Figure 3.5	Comparison of solutions for Example 3.4.5, numerical solution (dotted line), exact solution (solid line).	52
Figure 4.1	Eigenvalue distribution when (a) $N = 20$, (b) $N = 40$	60
Figure 4.2	Eigenvalue distribution when (c) $N = 80$ and (d) $N = 120$	60
Figure 4.3	Numerical solutions and projection on xt - plane of Example 4.1.1	61
Figure 4.4	Numerical solutions and projection on xt - plane of Example 4.1.2 (a)	61
Figure 4.5	Numerical solutions and projection on xt - plane of Example 4.1.2 (b)	61
Figure 4.6	Comparison of numerical and exact solutions for constant time levels of Example 4.1.1	62
Figure 4.7	Comparison of numerical and exact solutions for constant time levels of Example 4.1.2 (a)	62
Figure 4.8	Comparison of numerical and exact solutions for constant time levels of Example 4.1.2 (b)	63
Figure 4.9	Numerical solutions for constant time levels of Example 4.1.3 . . .	63
Figure 4.10	Eigenvalue distribution when (a) $N = 20$, (b) $N = 40$	68
Figure 4.11	Eigenvalue distribution when (c) $N = 80$, (d) $N = 120$	68
Figure 4.12	Numerical solution of Example 4.2.1 for $\alpha = 0.001$	69
Figure 4.13	Numerical solution of Example 4.2.1 for $\alpha = 1$	69
Figure 4.14	Numerical solution of Example 4.2.1 for $\alpha = 3$	70

Figure 4.15 Comparison between numerical and exact solutions for different time levels of Example 4.2.1 for $\Delta t = 0.01$ and $N = 30$	70
Figure 4.16 Eigenvalue distribution when (a) $N = 20$, (b) $N = 40$	72
Figure 4.17 Eigenvalue distribution when (c) $N = 80$, (d) $N = 120$	73
Figure 4.18 Numerical solutions for different time levels of Example 4.2.2 (a) for $\Delta t = 0.001$ and $N = 40$	73
Figure 4.19 Numerical solutions for different time levels of Example 4.2.2 (b) for $\Delta t = 0.001$ and $N = 40$	74
Figure 4.20 Graph of numerical solutions for Example 4.2.2 (a)	74
Figure 4.21 Graph of numerical solutions for Example 4.2.2 (b)	75
Figure 4.22 Eigenvalue distribution when (a) $N = 20$, (b) $N = 40$	79
Figure 4.23 Eigenvalue distribution when (c) $N = 80$, (d) $N = 120$	80
Figure 4.24 Solutions of Example 4.3.1 (b) for different values of constant time	80
Figure 4.25 Numerical solutions for different time levels of Example 4.3.2 when $\nu = 0.4/\pi^2$	81
Figure 4.26 Numerical solutions for different time levels of Example 4.3.2 when $\nu = 0.8/\pi^2$	81
Figure 4.27 The chaotic simulation of the KS equation with Gaussian initial condition at $t = 5$	82
Figure 4.28 The chaotic simulation of the KS equation with Gaussian initial condition at $t = 10$	82
Figure 4.29 The chaotic solution of the KS equation for Example 4.3.3 (a)	83
Figure 4.30 The chaotic solution of the KS equation for Example 4.3.3 (b)	83
Figure 4.31 Eigenvalue distribution when (a) $N = 20$, (b) $N = 40$	88
Figure 4.32 Eigenvalue distribution when (c) $N = 80$, (d) $N = 120$	88
Figure 4.33 Comparison between numerical and exact solutions of Example 4.4.1	89
Figure 4.34 Comparison between numerical and exact solutions of Example 4.4.2	89
Figure 4.35 Numerical solutions of Example 4.4.1	90
Figure 4.36 Numerical solutions of Example 4.4.2	90
Figure 4.37 Numerical simulation of Example 4.4.3 (a) at various time steps with sinusoid initial condition	91
Figure 4.38 Numerical simulation of Example 4.4.3 (b) at various time steps with Gaussian initial condition	91
Figure 4.39 Eigenvalue distribution for (a) $N = 20$, (b) $N = 40$	95
Figure 4.40 Eigenvalue distribution for (c) $N = 80$, (d) $N = 120$	95
Figure 4.41 Exact solution of Ginzburg Landau Equation	96
Figure 4.42 Numerical solution of Ginzburg Landau Equation	96
Figure 4.43 Numerical solutions of Ginzburg Landau equation for different values of constant time	97

LIST OF TABLES

Table 3.1	Numerical solutions and absolute errors for $N = 5$ of Example 3.1 .	24
Table 3.2	Numerical solutions and absolute errors for $N = 9$ of Example 3.1 .	25
Table 3.3	Numerical solutions and absolute errors for $N = 13$ of Example 3.1	25
Table 3.4	Numerical solutions and absolute errors for $N = 5$ of Example 3.2 .	27
Table 3.5	Numerical solutions and absolute errors for $N = 9$ of Example 3.2 .	27
Table 3.6	Numerical solutions and absolute errors for $N = 13$ of Example 3.2	28
Table 3.7	Numerical solutions and absolute errors for $N = 5$ of Example 3.3 .	29
Table 3.8	Numerical solutions and absolute errors for $N = 9$ of Example 3.3 .	29
Table 3.9	Numerical solutions and absolute errors for $N = 13$ of Example 3.3	30
Table 3.10	Numerical solutions and absolute errors for $N = 5$ of Example 3.4 .	32
Table 3.11	Numerical solutions and absolute errors for $N = 9$ of Example 3.4 .	32
Table 3.12	Numerical solutions and absolute errors for $N = 13$ of Example 3.4	33
Table 3.13	Comparison of maximum absolute errors for Example 3.4	33
Table 3.14	Numerical solutions and absolute errors for $N = 5$ of Example 3.5 .	34
Table 3.15	Numerical solutions and absolute errors for $N = 9$ of Example 3.5 .	35
Table 3.16	Numerical solutions and absolute errors for $N = 13$ of Example 3.5	35
Table 3.17	Comparison of maximum absolute errors of Example 3.5	35
Table 3.18	Numerical solutions and absolute errors for $N = 5$ of Example 3.6 .	37
Table 3.19	Numerical solutions and absolute errors for $N = 9$ of Example 3.6 .	37
Table 3.20	Numerical solutions and absolute errors for $N = 13$ of Example 3.6	38
Table 3.21	Comparison of absolute errors of Example 3.4.1 for $\epsilon = 2^{-3}$	42
Table 3.22	Comparison of absolute errors of Example 3.4.1 for $\epsilon = 2^{-6}$	42
Table 3.23	Comparison of absolute errors of Example 3.4.1 for $\epsilon = 2^{-9}$	43
Table 3.24	Comparison of absolute errors of Example 3.4.2 for $\epsilon = 2^{-3}$	44
Table 3.25	Comparison of absolute errors of Example 3.4.2 for $\epsilon = 2^{-6}$	45
Table 3.26	Comparison of absolute errors of Example 3.4.2 for $\epsilon = 2^{-9}$	45
Table 3.27	Error Norms for Example 3.4.3	47
Table 3.28	Comparison of absolute errors between different methods for Example 3.4.3 when $\epsilon = 10^{-3}$	47
Table 3.29	Absolute errors for $N = 10$	49
Table 3.30	Error Norms for Example 3.4.4	49

Table 3.31 Absolute errors for $N = 10$	51
Table 3.32 Error Norms for Example 3.4.5	51
Table 4.1 Errors for different time levels of Example 4.1.1 for $N = 40$	58
Table 4.2 Maximum eigenvalues for different number of grid points	58
Table 4.3 Errors for different time levels of Example 4.1.2 (a) for $N = 80$	58
Table 4.4 Comparison of absolute errors of Example 4.1.2 (a) when $t = 0.1$	58
Table 4.5 Errors for different time levels of Example 4.1.2 (b) for $N = 80$	59
Table 4.6 Comparison of absolute errors of Example 4.1.3 when $t = 0.1$	59
Table 4.7 Maximum absolute errors for non-homogeneity parameters when $t = 1$ and $N = 30$	67
Table 4.8 Error Norms for different time levels when $\Delta t = 0.01$, $\alpha = 1$ and $N = 30$	67
Table 4.9 Maximum eigenvalues of the discretized system for different number of grid points	68
Table 4.10 Error Norms for constant time levels when $\Delta t = 0.001$, and $N = 40$ for Example 4.2.2 (a)	72
Table 4.11 Error Norms for constant time levels when $\Delta t = 0.001$, and $N = 40$ for Example 4.2.2 (b)	72
Table 4.12 Comparison of Global Relative Errors of Example 4.3.1 (a) for Different Methods	79
Table 4.13 Errors for different time levels of Example 4.3.1 (b) for $N = 30$	79
Table 4.14 Numerical solutions of Example 4.4.1 at $t = 0.5$	87
Table 4.15 Numerical solutions of Example 4.4.2 at $t = 0.5$	87
Table 4.16 Error Norms for different time levels when $\Delta t = 0.001$ for Example 4.4.1	87
Table 4.17 Error Norms for different time levels when $\Delta t = 0.001$, for Example 4.4.2	87
Table 4.18 Comparison of methods of Example 4.5 at various time levels	94

ABSTRACT

SOLVING DIFFERENTIAL EQUATIONS USING NUMERICAL METHODS: DIFFERENTIAL QUADRATURE

Gülsemay YİĞİT

Department of Mathematical Engineering
Doctor of Philosophy Thesis

Advisor: Prof. Dr. Mustafa BAYRAM

This thesis presents numerical schemes to solve differential equations using the method of polynomial differential quadrature. Differential quadrature (DQ) is a discrete derivative approximation method which provides highly accurate numerical solutions for differential equations. Determination of the weighting coefficients is investigated in detail, which is essential for the implementation of the present method. Simplified formulations of these coefficients are also stated by using zeros of the Chebyshev polynomials. The implementation of DQ to linear or nonlinear ordinary differential equations is established by producing the set of algebraic equations.

The application of DQ to partial differential equations results in systems of time-dependent differential equations. These time dependent differential equations systems are integrated by the Runge-Kutta method. The DQ method is then applied to solve five different time-dependent partial differential equations. Each of the five equation is solved numerically by giving illustrative test problems with the stability analysis. Linear and nonlinear diffusion equations, reaction-diffusion equation in non-homogeneous media, the Kuramoto-Sivashinsky equation, the inviscid Burgers' equation and the Ginzburg-Landau equation are studied by using the proposed method. Solutions accomplished by the present scheme are compared to the analytical or well-known numerical methods.

Keywords: Differential quadrature, weighting coefficients matrix, Chebyshev polynomials, stability.



DİFERENSİYEL DENKLEMLERİN SAYISAL ÇÖZÜMLERİ: DİFERENSİYEL QUADRATURE

Gülsemay YİĞİT

Matematik Mühendisliği Bölümü

Doktora Tezi

Danışman: Prof. Dr. Mustafa BAYRAM

Bu tez çalışmasında, polinom tabanlı diferensiyel quadrature metodu kullanılarak diferensiyel denklemlerin nümerik şemaları sunulmuştur. Diferensiyel quadrature (DQ), diferensiyel denklemlerin çözümüne yüksek doğruluklu nümerik çözümler veren, bir ayrık türev yaklaşımı metodudur. Ele alınan bu metodun uygulanabilmesinde gereken ağırlıklı katsayıların nümerik hesabı ayrıntılı bir şekilde incelenmiştir. Bu katsayıların, Chebyshev polinomlarının köklerinden elde edilen sadeleştirilmiş formülasyonları, ayrıca verilmiştir. DQ metodunun lineer veya lineer olmayan adi diferensiyel denklemlere uygulanması cebirsel denklem sistemlerinin oluşturulması ile yapılmıştır.

DQ metodunun kısmi diferensiyel denklemlere uygulanması ile zamana bağlı adi diferensiyel denklem sistemleri elde edilir. Bu diferensiyel denklem sistemlerinin zaman integrasyonu Runge-Kutta metodu ile yapılmıştır. Daha sonra, DQ metodu, beş farklı zamana bağlı kısmi diferensiyel denkleme uygulanmıştır. Ele alınan tüm kısmi türevli diferensiyel denklemler, açıklayıcı test problemler ve stabilite analizleri verilerek nümerik olarak çözülmüştür. Lineer ve lineer olmayan difüzyon denklemleri, homojen olmayan malzemelerde reaksiyon-difüzyon denklemi, Kuramoto-Sivashinsky denklemi, viskozitesiz Burgers' denklemi ve lineer olmayan Ginzburg-Landau denklemi önerilen metotla çalışılmıştır. Sunulan metot ile edilen nümerik sonuçlar, analitik veya iyi bilinen nümerik metotlarla karşılaştırılmıştır.

Anahtar Kelimeler: Diferensiyel quadrature, ağırlıklı katsayılar matrisi, Chebyshev polinomları, kararlılık.



1

INTRODUCTION

Scientists and engineers try to formalize schemes of events in nature using mathematical models. These mathematical models can be in the forms of algebraic, differential or other types of equations. Most mathematical models that give rise to differential equations may not have analytical solutions. Alternatively, numerical simulations of such models are suggested for the solution of governing equations by employing numerical methods [1]. It is also well-known that numerical discretization techniques play a fundamental role for the solutions of most engineering problems due to their fast and effective outcomes especially when analytical solutions fail to provide practical way forward [2].

Differential quadrature (DQ) is a discrete derivative process which gives highly accurate solutions to a single or system of differential equations. The current method approximates the derivative of the function by a weighted sum of function values on the whole domain. It provides efficient numerical solutions with relative ease to code the implementation of the required algorithms [3].

1.1 Literature Review

The differential quadrature method (DQM) was introduced by Bellman [4] and his associates to propose an alternative form for the existing numerical methods such as finite difference and finite element methods. The scheme has been widely applied to numerous equations in engineering and physical disciplines [2].

The essential part of the present discretization scheme is to determine the weighting coefficients. Bellman et al [4, 5] introduced two approaches to build up the weighting coefficients for the derivative of order one. First approach solves an algebraic system. The second method uses shifted Legendre polynomials as test function and grid points are preferred as zeros of shifted Legendre polynomials. First approach of Bellman does not provide practical solutions when the grid points are getting larger. This difficulty

was resolved by Quan and Chang [6, 7] who proposed the application of Lagrange polynomials to compute the weighting coefficients of order one and two. Shu [8] generalized the idea of previous forms of weighting coefficients using polynomial approximation theory in association with the properties of linear vector space and concluded that weighting coefficients can be determined by appropriate choice of the basis functions. Useful formulas are proposed in [8] to compute derivatives of any order. Stability analysis based on eigenvalue distribution is also explored in [8] together with different time integration schemes. Shu also presented the relationships between finite difference and collocation methods with the differential quadrature method [8].

Several independent formulations are presented in literature to determine weighting coefficients which is necessary to yield numerical solutions. To name a few, Korkmaz and Dag [9] used sinc functions, Cheng et al [10] employed Hermite polynomials, Shu and Wu [11] used radial basis functions, Mittal [12] proposed Bernstein polynomials based differential quadrature, O'Mahoney [13] applied Laguerre polynomials, Dag et al [14] employed cosine expansion situated differential quadrature. Korkmaz and Dag [15] applied cosine expansion for solutions of Schrödinger equation and Korkmaz and Dag [16, 17] also solved same model using polynomial based differential quadrature. Korkmaz [18] solved Korteweg–de Vries (KdVE) equation with Lagrange and Fourier basis. Korkmaz and Dag [19] also solved Burgers' equation using polynomial differential quadrature.

One of the popular approach to choose basis functions, is B-splines to acquire numerical solutions. A wide range of applications in literature are based on B-spline differential quadrature. For example, Zhong [20] studied quintic B-spline basis differential quadrature for solutions of fourth order differential equations. Arora and Singh [21] used modified B-splines to generate weighting coefficients. Korkmaz [22] employed different types of B-spline functions as base and applied the present method to one-dimensional physical models. Korkmaz et al [23–27] implemented B-spline generated differential quadrature to various equations. They used Runge-Kutta scheme for time discretization with stability analysis of the presented equations. Başhan et al [28–33] also used B-splines for solutions of many different equations. Mittal and Dahiya [34–37] studied on various equations using quadrature methods generated by B-splines.

DQ method has been extensively used in areas such as material science, thermal and structural mechanical analysis as well as physics and biology. Some applications on mechanics can be found in [2, 3, 8, 38–42]. Mittal and Jiwari [43] studied some nonlinear differential equations which models various processes in biology.

Akman [44] solved diffusion equations in two-dimensions with different types of boundary conditions with Runge-Kutta time integration. Meral and Tezer-Sezgin [45] solved nonlinear reaction-diffusion equation in one and two dimensional space using polynomials as base functions. In a subsequent paper, Meral and Tezer-Sezgin [46] employed differential quadrature to nonlinear reaction-diffusion and nonlinear wave equations using three different time integration schemes. Tomasiello [47–49] proposed iterative differential quadrature method with stability analysis. Tomasiello [50] introduced differential quadrature least square method and employed to buckling problem with elastic supports.

The DQM is generally used to discretize space derivatives. A number of studies show that DQM gives effective results in temporal discretization as well. Chen [38] proposed DQ based time integration scheme. Fung [51, 52] presented DQ based time discretization algorithms which are unconditionally stable with the analysis of grid point distributions. Shu et al [53] explored block marching technique with differential quadrature and concluded that scheme obtained by block-marching gives better solutions when compared to classical Runge-Kutta. Wang [54] studied DQ based time integration and applied it to nonlinear dynamic equations.

1.2 Objective of the Thesis

The work of the current thesis aims to achieve numerical solutions for some important differential equations using polynomial differential quadrature method. Differential quadrature algorithm is applied to one-dimensional time dependent problems together with sample cases to show the accuracy of the method. We also aim to present an approach for computing the weighting coefficients using Chebyshev polynomials. The present scheme is demonstrated to provide effective numerical results for model equations in all cases.

1.3 Hypothesis

Polynomial approximation is used to generate weighting coefficients to discretize spatial derivatives. First order weighting coefficients are determined and used to evaluate higher coefficients which are used in procedures of solving the model problems. Time integration of the partial differential equation is performed by Runge-Kutta method of fourth order which has strong stability properties. Matrix stability analysis is implemented related to the eigenvalues of space discretization matrix. Efficiency is shown on sample equations by computing error norms.

1.4 Thesis Plan

This thesis contains five chapters. First chapter consists of detailed literature survey and motivation for the study. Second chapter introduces the procedure of implementation for the differential quadrature method together with the computation of weighting coefficients matrix based on Chebyshev polynomials. Runge-Kutta method of order four and its stability criteria are also presented. In the third chapter, application of the current method regarding for some ordinary differential equations is studied. A number of models with higher order singularly perturbed problems are investigated by the proposed technique.

Chapter four constitutes the main results of this thesis. In this chapter, five different time-dependent model problems are numerically solved. These consists of linear and nonlinear diffusion equations, reaction-diffusion equations in non-homogenous media, the Kuramoto-Sivashinsky equation, the inviscid Burgers' equation and the Ginzburg-Landau equation. In the fourth chapter, all problems are analyzed with stability results.

In the final chapter, all results are discussed with the advantages and disadvantages of the proposed schemes with possible future extensions of the work.

2 METHODOLOGY

Differential quadrature method (DQM) incorporates higher order polynomial approximation to provide numerical solutions. This approximation is used to compute the weighting coefficients matrix which is essential to assemble numerical schemes for solving ordinary or partial differential equations.

This chapter provides definitions related with DQM and computing of weighting coefficients with the relevant description of polynomial differential quadrature strategies. Throughout the thesis time-dependent problems are discretized using Runge-Kutta time integration scheme where the application of DQ is used to approximate spatial derivatives. We combine the differential quadrature and Runge-Kutta method of order four towards the final part of this chapter.

2.1 The Differential Quadrature Method

Let $f(x)$ be a continuous and differentiable function of x defined in $[a, b] \subset \mathbb{R}$. Numerical derivative of f at each nodal point x_i is indicated as the weighted sum of function values on the entire domain. Quadrature description of first derivative at each nodes is formulated as [8],

$$f_x(x_i) = \left. \frac{df}{dx} \right|_{x_i} = \sum_{j=1}^N w_{ij} f(x_j) \quad i = 1, 2, \dots, N. \quad (2.1)$$

Here, w_{ij} is called as weighting coefficients and N is the number of spatial grid on the problem domain. The n -th order derivative formulation is defined using same idea,

$$f_x^{(n)}(x_i) = \left. \frac{d^n f}{dx^n} \right|_{x_i} = \sum_{j=1}^N w_{ij}^{(n)} f(x_j) \quad i = 1, 2, \dots, N, \quad (2.2)$$

where $w_{ij}^{(n)}$ is n -th order coefficients. The primary aspect of the procedure is to determine weighting coefficients $w_{ij}^{(n)}$. The differential quadrature method (DQM)

mainly depends on high order polynomial approximation of a smooth function. The coefficients for first order derivative are calculated, then higher order derivatives are obtained by recurrent formulas [8].

Weierstrass' first theorem [8]: If $f(x)$ is a real valued continuous function on $[a, b]$ with any $\epsilon > 0$. Then, there exists a polynomial p with the degree n such that,

$$|f(x) - p_n(x)| \leq \epsilon, \quad (2.3)$$

maintains throughout the domain $[a, b]$.

Assume that u is the unknown function of a differential equation, then u is represented as,

$$u(x) \approx p_N(x) = \sum_{k=0}^{N-1} c_k x^k, \quad (2.4)$$

where c_k are constants corresponding to the index k . Here, p denotes a linear vector space V_N where N represents its dimension. Thus, set of polynomials $1, x, x^2, x^3$, in such vector space are linearly independent. It is therefore that, $e_k = x^{k-1}$ is a basis of V_N .

The function values at specific nodal points should be determined to achieve numerical solution of a differential equation. Assume that, there are N nodes on $[a, b]$ in the form of $a = x_1, x_2, \dots, x_N = b$ and the function value at a discrete point x_i is given as $u(x_i)$. Then the constants c_k are obtained from the solution of the following system of equations,

$$\begin{aligned} u(x_1) &= c_0 + c_1 x_1 + c_2 x_1^2 + \dots + c_{N-1} x_1^{N-1}, \\ u(x_2) &= c_0 + c_1 x_2 + c_2 x_2^2 + \dots + c_{N-1} x_2^{N-1}, \\ &\vdots \\ u(x_N) &= c_0 + c_1 x_N + c_2 x_N^2 + \dots + c_{N-1} x_N^{N-1}. \end{aligned} \quad (2.5)$$

The matrix in (2.5) is in the form of a Vandermonde matrix which is non-singular. Hence, the matrix is invertible and provides a unique solution for each constants c_k . When constants are determined, the approximated polynomial is achieved. However, it is more difficult to determine these coefficients $c_0, c_1, c_2, \dots, c_N$ since the system becomes ill-conditioned with large number of grid points N .

Linear vector space can be spanned by many different sets of base function. Therefore, the difficulty caused by high grid spacing is tackled using Lagrange interpolation polynomial as the base polynomial in this case.

The k -th order Lagrange interpolating polynomial is of the form,

$$P_N(x) = \sum_{k=1}^N f(x_k) l_k(x), \quad (2.6)$$

where

$$l_k(x) = \frac{M(x)}{M^{(1)}(x_k)(x - x_k)}, \quad (2.7)$$

with

$$M(x) = (x - x_1)(x - x_2) \cdots (x - x_N),$$

$$M^{(1)}(x_k) = (x_k - x_1) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_N) = \prod_{i=1, i \neq k}^N (x_k - x_i). \quad (2.8)$$

The k -th Lagrange polynomial $l_k(x)$ in (2.7) satisfies the property of Kronecker delta, equivalently written as,

$$l_k(x_i) = \begin{cases} 1, & \text{when } k = i \\ 0, & \text{otherwise.} \end{cases} \quad (2.9)$$

2.2 Chebyshev Polynomials

The Chebyshev polynomials are obtained by the eigenfunctions of the following singular Sturm-Liouville problem [55],

$$\left(\sqrt{1-x^2} T'_k(x) \right)' + \frac{k^2}{\sqrt{1-x^2}} T_k(x) = 0, \quad (2.10)$$

where $T'_k(x)$ represents the first order derivative of $T_k(x)$. The Chebyshev polynomial of the first kind $T_k(x)$ is the polynomial of degree k for $x \in [-1, 1]$, defined as,

$$T_k(x) = \cos(k \arccos x), \quad k = 0, 1, 2, \dots. \quad (2.11)$$

Substituting for $x = \cos \theta$ allows to write (2.11) as

$$T_k(x) = \cos(k\theta). \quad (2.12)$$

Thus, the first Chebyshev polynomials are obtained as,

$$\begin{aligned} T_0 &= 1, \\ T_1 &= \cos \theta = x, \\ T_2 &= \cos 2\theta = 2 \cos^2 \theta - 1 = 2x^2 - 1, \\ &\vdots \end{aligned} \tag{2.13}$$

Using the trigonometric identity,

$$\cos(k+1)\theta + \cos(k-1)\theta = 2 \cos \theta \cos k\theta, \tag{2.14}$$

gives rise to the recurrence relationship,

$$T_{k+2}(x) - 2xT_{k+1}(x) + T_k(x) = 0, \quad k \in N. \tag{2.15}$$

It means that T_{k+2} can be computed in terms of T_k and T_{k+1} for all $k \in N$. The graphs of the first Chebyshev polynomials can be found in [55]. The values of $T_k(x)$ and T'_k which represent first order derivative of $T_k(x)$ have the properties given by,

$$T_k(\pm 1) = (\pm 1)^k, \text{ and } T'_k(\pm 1) = (\pm 1)^{k+1}k^2, \tag{2.16}$$

$$T_k(-x) = (-1)^k T_k(x) = 0. \tag{2.17}$$

Chebyshev polynomials are orthogonal [55] with the respect to the Chebyshev weight $w(x) = \frac{1}{\sqrt{1-x^2}}$ on the interval $[-1, 1]$. Thus,

$$\int_{-1}^1 T_k(x) T_l(x) w(x) dx = \frac{\pi}{2} c_k \delta_{k,l} \tag{2.18}$$

where, $\delta_{k,l}$ is Kronecker delta, and

$$c_k = \begin{cases} 2, & \text{when } k = 0 \\ 1, & \text{when } k \geq 1. \end{cases} \tag{2.19}$$

Using the definition,

$$\delta_{k,l} = \begin{cases} 0, & \text{when } k \neq l \\ 1, & \text{when } k = l. \end{cases} \tag{2.20}$$

Equation (2.18) becomes,

$$\int_{-1}^1 T_k(x)^2 \frac{dx}{\sqrt{1-x^2}} = c_k \frac{\pi}{2}. \quad (2.21)$$

With DQ discretization, generally nonuniform grid point distribution is preferred since it gives more stable solutions [8]. In this thesis, we use the Chebyshev-Gauss-Lobatto (CGL) grid point distribution which is refined near boundaries on space domain. This approach provides the opportunity to choose less number of grid points compared to uniform distribution. In many physical model problems, choosing grid points as CGL provides to control the behavior of the approach, near the endpoints of the domain.

Chebyshev-Gauss-Lobatto (CGL) grid points are defined [55] as,

$$x_i = \frac{\cos(\pi i)}{N}, \quad i = 0, 1, \dots, N. \quad (2.22)$$

These points are the roots of polynomial $(1-x^2)T'_N(x)$ and the polynomial T_N gets its extreme values at ± 1 at the points x_i . Also these points are defined on the interval $[-1, 1]$ [55]. If the problem is given on interval $[a, b]$, the following transformation is used to obtain x_i on $[a, b]$,

$$x_i = \frac{b-a}{2}(1-\xi_i) + a. \quad (2.23)$$

2.3 Polynomial based Differential Quadrature (PDQ)

In this section, we examine the details of how to determine the weighting coefficients by choosing test functions as polynomials. Polynomial based differential quadrature gives highly accurate numerical solutions for most differential equations. We combine the idea of Bellman's approach, Quan and Chang's treatment, with Shu's general approach and analysis of the modification of the weighting coefficients using roots of Chebyshev polynomials.

2.3.1 Computation of Weighting Coefficients for First Order Derivative

Consider a one-dimensional problem on a closed interval $[a, b]$ and coordinates with $a = x_1, x_2, \dots, x_N = b$ of N grid points. $f(x)$ is a smooth function on the interval $[a, b]$ and its first order derivative $f_x(x_i)$ is defined as,

$$f_x(x_i) = \sum_{j=1}^N a_{ij} f(x_j) \quad i = 1, 2, \dots, N. \quad (2.24)$$

Recall the conventional form of polynomial approximation,

$$f(x_i) = \sum_{k=1}^N d_k e_k(x_i), \quad i = 1, 2, \dots, N \quad (2.25)$$

where d_k are constant corresponding to index k and $e_k(x_i)$ is the base polynomial. Taking first derivative of both sides of (2.25),

$$f_x(x_i) = \sum_{k=1}^N d_k e'_k(x_i), \quad i = 1, 2, \dots, N. \quad (2.26)$$

Substituting (2.25) and (2.26) in (2.24) yields,

$$\sum_{k=1}^N d_k e'_k(x_i) = \sum_{j=1}^N a_{ij} f(x_j), \quad i = 1, 2, \dots, N \quad (2.27)$$

or equivalently written as,

$$\sum_{k=1}^N d_k e'_k(x_i) = \sum_{j=1}^N a_{ij} \left(\sum_{k=1}^N d_k e_k(x_j) \right), \quad i = 1, 2, \dots, N. \quad (2.28)$$

The right side of (2.28) can be written as,

$$\sum_{k=1}^N d_k e'_k(x_i) = \sum_{j=1}^N \sum_{k=1}^N a_{ij} d_k e_k(x_j), \quad i = 1, 2, \dots, N. \quad (2.29)$$

From (2.29), we obtain the formula for calculating the weighting coefficients as,

$$e'_k(x_i) = \sum_{j=1}^N a_{ij} e_k(x_j), \quad i = 1, 2, \dots, N. \quad (2.30)$$

Obtaining the coefficients a_{ij} results in weighted sum of the function value approximation for spatial derivatives at all grid points.

2.3.1.1 Bellman's Approaches

For the calculation of weighting coefficients, we begin with the Bellman's two approaches. In the first approach, Bellman suggested test functions as,

$$e_k(x) = x^k, \quad k = 0, 1, \dots, N-1, \quad (2.31)$$

which gives N trial functions. In (2.24), the indexes i and j are chosen such that $1 \leq i, j \leq N$, where the number of coefficients are $N \times N$. On the other hand, N test

functions are used at N nodal points. Thus, $N \times N$ algebraic equations are obtained in the form of,

$$\begin{aligned} \sum_{j=1}^N a_{ij} &= 0, \quad k = 0, \\ \sum_{j=1}^N a_{ij} x_j^k &= k x_i^{k-1}, \quad k = 1, 2, \dots, N-1, \quad i = 1, 2, \dots, N. \end{aligned} \quad (2.32)$$

System (2.32) is in the form of Vandermonde matrix and provides unique solutions. However, the drawback of this approach is that, grid points are restricted to 13. For larger N , condition number of the matrix get higher and instabilities may occur in applications [8].

In the second approach, test function is used as,

$$e_k(x) = \frac{L_N(x)}{L_N^{(1)}(x_k) \cdot (x - x_k)}, \quad k = 1, 2, \dots, N \quad (2.33)$$

where $L_N(x)$ and $L_N^{(1)}(x_k)$ represents Legendre polynomial of degree N and its derivative. Here, x_k is chosen as roots of Legendre polynomial and applied at N grid points $x_1, x_2, \dots, x_N$. Thus, following formulations are obtained to calculate the weighting coefficients,

$$\begin{aligned} a_{ij} &= \frac{L_N^{(1)}(x_i)}{(x_i - x_j) L_N^{(1)}(x_j)} \quad j \neq i, \\ a_{ii} &= \frac{1 - 2x_i}{2x_i(x_i - 1)}. \end{aligned} \quad (2.34)$$

Thus, the weighting coefficients are determined by a simple formulation unlike the first approach. The coordinates of grid points are chosen as roots of Legendre polynomials. However, this approach places some restriction on choosing grid points [8].

2.3.1.2 Quan and Chang's Approach

Quan and Chang suggested new formulations to improve Bellman's approaches based on Lagrange interpolation polynomials as test functions,

$$e_k(x) = \frac{M(x)}{M^{(1)}(x_k) \cdot (x - x_k)}, \quad k = 1, 2, \dots, N \quad (2.35)$$

where

$$M(x) = (x - x_1)(x - x_2) \cdots (x - x_N), \quad (2.36)$$

and

$$M^{(1)}(x_i) = \prod_{k=1, k \neq i}^N (x_i - x_k). \quad (2.37)$$

Using (2.36) and (2.37), explicit formulations are obtained in agreement with those in [8],

$$\begin{aligned} a_{ij} &= \frac{1}{x_j - x_i} \prod_{k=1, k \neq i}^N \frac{x_i - x_k}{x_j - x_k} \quad j \neq i, \\ a_{ii} &= \sum_{k=1, k \neq i}^N \frac{1}{x_i - x_k}. \end{aligned} \quad (2.38)$$

2.3.1.3 Shu's General Approach

Shu has suggested a generalized algorithm by combining previously mentioned approaches based on the properties of vector space and high order polynomial approximation. From the analysis of vector space, it is known that, if one set of base polynomials satisfy a linear operator, so does any arbitrary base polynomials. This explains that, each bases gives the same weighting coefficients. Thus, the weighting coefficients are independent from the choice of test functions. Since there are various sets of base polynomials, new computations can be generated to compute the weighting coefficients.

The solution of a differential equation is approximated by a high degree of polynomial. Let V_N denotes the N dimensional vector space spanned by $N-1$ degree of polynomial. Therefore V_N provides to reconstruct all polynomials of degree $N-1$ in terms of linear combination of weighted sum of all basis of V_N . The properties of linear vector space enables us to apply weighting coefficients to discretize a partial differential equation.

In general approach presented in [8], there are two choices of the base functions. First base function is chosen as Lagrange polynomial given in (2.35). To obtain an efficient formulation, the polynomial is indicated as,

$$M(x) = N(x, x_k)(x - x_k), \quad k = 1, 2, \dots, N, \quad (2.39)$$

along with $N(x_i, x_j) = M^{(1)}(x_i)\delta_{ij}$, where δ_{ij} is Kronecker operator. Substituting (2.39) into (2.35), base polynomial can be rewritten as,

$$e_k(x) = \frac{M(x)}{M^{(1)}(x_k)(x - x_k)} = \frac{N(x, x_k)}{M^{(1)}(x_k)}, \quad k = 1, 2, \dots, N. \quad (2.40)$$

Differentiation both sides of (2.40) with respect to x , we have,

$$e_k^{(1)}(x) = \frac{N^{(1)}(x, x_k)}{M^{(1)}(x_k)}, \quad k = 1, 2, \dots, N. \quad (2.41)$$

Substituting (2.40) and (2.41) into (2.30) results in,

$$\frac{N^{(1)}(x_i, x_k)}{M^{(1)}(x_k)} = \sum_{j=1}^N a_{ij} \frac{N(x, x_k)}{M^{(1)}(x_k)}, \quad k = 1, 2, \dots, N, \quad (2.42)$$

or equivalently written as,

$$N^{(1)}(x_i, x_k) = \sum_{j=1}^N a_{ij} N(x_j, x_k), \quad k = 1, 2, \dots, N. \quad (2.43)$$

Simplifying (2.43) by adapting the definition of Kronecker operator leads to formulas for coefficients in the form,

$$\begin{aligned} a_{ij} &= \frac{N^{(1)}(x_i, x_j)}{N(x_j, x_j)}, \\ a_{ij} &= \frac{N^{(1)}(x_i, x_j)}{M^{(1)}(x_j)}, \end{aligned} \quad (2.44)$$

for $k = j \neq i$. To evaluate $M^{(1)}(x_j)$, (2.37) is used and $N^{(1)}(x_i, x_j)$ is evaluated by taking the derivative of both sides (2.39). Therefore, following expressions for approximating first and higher order derivatives are obtained,

$$\begin{aligned} M^{(1)}(x) &= N^{(1)}(x, x_k) \cdot (x - x_k) + N(x, x_k), \\ M^{(2)}(x) &= N^{(2)}(x, x_k) \cdot (x - x_k) + 2N^{(1)}(x, x_k), \\ &\vdots \\ M^{(m)}(x) &= N^{(m)}(x, x_k) \cdot (x - x_k) + mN^{(m-1)}(x, x_k), \end{aligned} \quad (2.45)$$

for $m = 1, 2, \dots, N-1$, and $k = 1, 2, \dots, N$. Here, $M^{(m)}(x)$ and $N^{(m)}(x, x_k)$ represents m -th order derivatives of $M(x)$ and $N(x, x_k)$. Relation (2.45) indicates,

$$N^{(1)}(x_i, x_j) = \frac{M^{(1)}(x_i) - N(x_i, x_j)}{x_i - x_j}. \quad (2.46)$$

We substitute the expression $N(x_i, x_j) = M^{(1)}(x_i)\delta_{ij}$ for $i \neq j$ into (2.46) to obtain,

$$N^{(1)}(x_i, x_j) = \frac{M^{(1)}(x_i)}{x_i - x_j}, \quad i \neq j. \quad (2.47)$$

Similarly, $M^{(2)}$ in (2.45) is used for the case $i = j$ as,

$$N^{(1)}(x_i, x_j) = \frac{M^{(2)}(x_i)}{2}. \quad (2.48)$$

Substituting (2.47) and (2.48), weighting coefficients are determined as,

$$\begin{aligned} a_{ij} &= \frac{N^{(1)}(x_i, x_j)}{M^{(1)}(x_j)} = \frac{M^{(1)}(x_i)}{(x_i - x_j)M^{(1)}(x_j)}, \quad i \neq j, \\ a_{ii} &= \frac{N^{(1)}(x_i, x_i)}{M^{(1)}(x_i)} = \frac{M^{(2)}(x_i)}{2M^{(1)}(x_i)}. \end{aligned} \quad (2.49)$$

According to the simplified formulations of weighting coefficients, for $i \neq j$ can be computed using $M^{(1)}(x_i)$ when x_i is given. In the same manner, if $i = j$ second derivative $M^{(2)}(x_i)$ needs to be computed. However, calculating $M^{(2)}(x_i)$ is not trivial the same way as calculation of $M^{(1)}(x_i)$. Thus, using the properties of linear vector space, a_{ij} obtained from Lagrange interpolating polynomial is the same with set of base polynomials x^{k-1} , $k = 1, 2, \dots, N$. Hence, a_{ij} can also be computed for $k = 1$ as,

$$\sum_{j=1}^N a_{ij} = 0, \quad \text{or,} \quad a_{ii} = - \sum_{j=1, j \neq i}^N a_{ij}. \quad (2.50)$$

Formulations given by (2.49) and (2.50) are the two common ways to determine the weighting coefficients of order one.

2.3.2 Weighting Coefficients for Second and Higher Order Derivatives

DQ discretization of order two is given by,

$$f_x^{(2)}(x_i) = \sum_{j=1}^N a_{ij}^{(2)} f(x_j) \quad i = 1, 2, \dots, N, \quad (2.51)$$

where $a_{ij}^{(2)}$ is the second order weighting coefficient. Here, we consider the general approach from [8], to compute the weighting coefficients of order two and more. This approach is based on polynomial approximation and linear vector spaces like the approach of the first order derivative. This approach deals with the following two base polynomials,

$$\begin{aligned} e_k(x) &= x^{k-1}, \quad k = 1, 2, \dots, N, \\ e_k(x) &= \frac{M(x)}{M^{(1)}(x_k) \cdot (x - x_k)}, \quad k = 1, 2, \dots, N. \end{aligned} \quad (2.52)$$

Recall the expression,

$$e_k(x) = \frac{N(x, x_k)}{M^{(1)}(x_k)}. \quad (2.53)$$

Combining (2.53) with (2.51) provides,

$$a_{ij}^{(2)} = \frac{N^{(2)}(x_i, x_j)}{M^{(1)}(x_j)}. \quad (2.54)$$

Equation (2.45) can be expanded as,

$$\begin{aligned} M^{(2)}(x) &= N^{(2)}(x, x_k).(x - x_k) + 2N^{(1)}(x, x_k), \\ M^{(3)}(x) &= N^{(3)}(x, x_k).(x - x_k) + 3N^{(2)}(x, x_k). \end{aligned} \quad (2.55)$$

Then, using (2.54), the values $N^{(2)}$ are evaluated as,

$$\begin{aligned} N^{(2)}(x_i, x_j) &= \frac{M^{(2)}(x_i) - 2N^{(1)}(x_i, x_j)}{x_i - x_j}, \quad i \neq j, \\ N^{(2)}(x_i, x_i) &= \frac{M^{(3)}(x_i)}{3}. \end{aligned} \quad (2.56)$$

Substituting (2.56) into (2.54) leads to the expression for a_{ij} in the form,

$$a_{ij}^{(2)} = \frac{M^{(2)}(x_i) - 2N^{(1)}(x_i, x_j)}{M^{(1)}(x_j)}, \quad (2.57)$$

and

$$a_{ii}^{(2)} = \frac{M^{(3)}(x_i)}{3M^{(1)}(x_i)}. \quad (2.58)$$

Then, using (2.49) into (2.57), weighting coefficients are determined as,

$$a_{ij}^{(2)} = 2a_{ij} \left(a_{ii} - \frac{1}{x_i - x_j} \right), \quad i \neq j. \quad (2.59)$$

For the case $i \neq j$, and provided that $a_{ij}^{(2)}$ is computable, then for $i = j$, the third order derivative $M^{(3)}(x_i)$ should be calculated. Instead, an approach similar to the first order derivative can be used to determine weighting coefficients. Thus, the following recurrence formula can be used, when $k = 1$,

$$\sum_{j=1}^N a_{ij}^{(2)} = 0, \quad \text{or,} \quad a_{ii}^{(2)} = - \sum_{j=1, j \neq i}^N a_{ij}^{(2)}. \quad (2.60)$$

Shu's general approach can be expanded to find the formulation for general derivatives of any order,

$$\begin{aligned} w_{ij}^{(n)} &= n \left(w_{ij}^{(1)} w_{ii}^{(n-1)} - \frac{w_{ij}^{(n-1)}}{x_i - x_j} \right) \quad i, j = 1, 2, \dots, N, \\ w_{ii}^{(n)} &= - \sum_{j=1, j \neq i}^N w_{ij}^{(n)}, \end{aligned} \quad (2.61)$$

where $w_{ij}^{(n)}$ represents n -th order derivative.

2.3.3 Modification of Weighting Coefficients

As mentioned in previous sections, calculation of weighting coefficients does not depend on choice of test functions. In other words, every sets of base polynomials give the same weighting coefficients. Here, we consider modification of weighting coefficients obtained by Lagrange interpolating polynomial,

$$e_k(x) = \frac{C(x)}{C^{(1)}(x_k)(x - x_k)}, \quad k = 1, 2, \dots, N. \quad (2.62)$$

We replaced $C(x)$ by $M(x)$ in (2.35). Assuming that grid points are chosen as Chebyshev collocation points,

$$x_i = \cos(\theta_i), \quad \theta_i = \frac{i\pi}{N}, \quad i = 0, 1, \dots, N. \quad (2.63)$$

Recall the Chebyshev collocation points x_i are the zeros of polynomial $(1-x^2)T_N^{(1)}(x)$. So the function $C(x)$ in (2.62) is rewritten as,

$$C(x) = (1-x^2)T_N^{(1)}(x), \quad (2.64)$$

where $T_N^{(1)}(x)$ is the first derivative of $T_N(x)$. Using the equations,

$$x = \cos(\theta), \quad T_N(x) = \cos(N\theta), \quad (2.65)$$

$$T_N^{(1)}(x) = (\cos(N\theta))' = -N \sin(N\theta) d\theta, \quad (2.66)$$

where

$$d\theta = -\frac{1}{\sqrt{1-x^2}} = -\frac{1}{\sin(\theta)}, \quad (2.67)$$

Then, (2.64) becomes,

$$C(x) = C(\theta) = N \sin(\theta) \sin(N\theta). \quad (2.68)$$

By taking the derivative of Equation (2.68)

$$C^{(1)}(x) = (N \sin(\theta) \sin(N\theta))', \quad (2.69)$$

or

$$C^{(1)}(x) = -\frac{N \cos(\theta) \sin(N\theta) + N^2 \sin(\theta) \cos(N\theta)}{\sin(\theta)}. \quad (2.70)$$

Since $N\theta_i = i\pi$ when $\sin(\theta_i) \neq 0$, for $i \neq 0, N$, (2.70) becomes

$$C^{(1)}(x_i) = (-1)^{i+1} N^2. \quad (2.71)$$

If $\sin(\theta_i) = 0$, L'Hospital's rule is used, then,

$$\begin{aligned} C^{(1)}(x_0) &= -2N^2, \\ C^{(1)}(x_N) &= (-1)^{N+1} 2N^2. \end{aligned} \quad (2.72)$$

Using Equations (2.71) and (2.72) explicit matrix formula for the first derivative is established as, [8, 55],

$$\begin{aligned} w_{ij} &= \frac{\bar{c}_i (-1)^{i+j}}{\bar{c}_j (x_i - x_j)}, \quad 0 \leq i, j \leq N, i \neq j, \\ w_{ii} &= -\frac{x_i}{2(1 - x_i^2)}, \quad 1 \leq i \leq N-1, \\ w_{00} &= -w_{NN} = \frac{2N^2 + 1}{6}, \end{aligned} \quad (2.73)$$

where $\bar{c}_0 = \bar{c}_N = 2$ and $\bar{c}_j = 1$, $1 \leq j \leq N-1$.

DQ discretization enables to define the higher order derivatives by a recurrence relation using first order weighting coefficients. Matrix multiplication idea is now used to compute the related higher order derivatives which will be used in following chapters.

2.3.4 Matrix Multiplication Approach

Let

$$\frac{\partial^2 U}{\partial x^2} = \frac{\partial}{\partial x} \left(\frac{\partial U}{\partial x} \right) \quad (2.74)$$

, be the second order derivative operator. Application of the DQ algorithm for the left side of (2.74) gives the following,

$$U_x^{(2)}(x_i) = \sum_{j=1}^N w_{ij}^{(2)} U(x_j) \quad i = 1, 2, \dots, N. \quad (2.75)$$

Using similar idea for the right side of (2.74) gives,

$$U_x^{(2)}(x_i) = \sum_{k=1}^N w_{ik}^{(1)} U_x^{(1)}(x_k) = \sum_{k=1}^N w_{ik}^{(1)} \sum_{j=1}^N w_{kj}^{(1)} U(x_j),$$

$$\sum_{j=1}^N w_{ij}^{(2)} U(x_j) = \sum_{j=1}^N \left[\sum_{k=1}^N w_{ik}^{(1)} w_{kj}^{(1)} \right] U(x_j), \quad i = 1, 2, \dots, N. \quad (2.76)$$

Combining (2.75) and (2.76) gives the following,

$$w_{ij}^{(2)} = \sum_{k=1}^N w_{ik}^{(1)} w_{kj}^{(1)}, \quad (2.77)$$

which corresponds to definition of matrix multiplication [8]. Matrices given in Equation (2.77) is defined as,

$$[A^{(1)}] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}, \quad [A^{(2)}] = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1N} \\ b_{21} & b_{22} & \cdots & b_{2N} \\ \vdots & & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NN} \end{bmatrix}. \quad (2.78)$$

So, the equation (2.77) is written in matrix form,

$$A^{(2)} = A^{(1)} A^{(1)}, \quad (2.79)$$

where $A^{(2)} = w_{ij}^{(2)}$ and $A^{(1)} = w_{ij}^{(1)}$. The last expression shows that second order derivative is evaluated by matrix multiplication of first order derivatives. If the idea is generalized, the m -th order derivative is determined as [8],

$$A^{(m)} = A^{(1)} A^{(m-1)}, \quad m = 2, 3, \dots, N-1. \quad (2.80)$$

In this thesis, matrix multiplication approach is used to obtain related higher order derivatives depends on the feature of the differential equation.

2.4 Fourth Order Runge-Kutta Algorithm

There are several numerical methods to solve time dependent partial differential equations. One of the common way is discretization of spatial derivatives using a numerical method such as finite difference, finite element, etc. Then time dependent linear equations get reduced to system of ordinary differential equations [22]. This system can be integrated by using explicitly or implicitly. In this thesis, present method

is combined with the Runge-Kutta method which preserves strong stability and gives high accuracy. After DQ discretization, model equations are transformed into a time dependent ordinary differential equation as,

$$\frac{dU}{dt} = f(U), \quad U(t_0) = 0, \quad t \in [t_0, T], \quad (2.81)$$

Time integration of the system is established using Runge-Kutta algorithm given by,

$$U_{n+1} = U_n + \frac{1}{6}[K_1 + 2K_2 + 2K_3 + K_4], \quad n = 0, 1, 2, \dots, N-1, \quad (2.82)$$

where

$$\begin{aligned} K_1 &= \Delta t f(U_n), \\ K_2 &= \Delta t f\left(U_n + \frac{1}{2}K_1\right), \\ K_3 &= \Delta t f\left(U_n + \frac{1}{2}K_2\right), \\ K_4 &= \Delta t f(U_n + K_3). \end{aligned} \quad (2.83)$$

Here, Δt represents time step.

A time dependent problem with proper initial and boundary conditions is expressed as

$$\frac{\partial U}{\partial t} = l(U), \quad U(t_0) = 0, \quad t \in [t_0, T], \quad (2.84)$$

where l represents spatial differential operator and it is generally nonlinear. After DQ discretization and proper linearizations, Equation (2.84) transforms into ordinary differential equation. Then, application of boundary conditions, following matrix-vector form is obtained,

$$\frac{d}{dt}\{U\} = [A]\{U\} + \{B\}, \quad (2.85)$$

where $\{U\}$ is the unknown vector at interior points and $\{B\}$ represents the vector consists of boundary conditions and inhomogeneous part of the equation. $[A]$ is the weighting coefficient matrix.

Stability of the constructed scheme for the numerical integration of Equation (2.84) depends on the stability of the differential equation. If (2.84) is unstable then the stable numerical scheme may not produce the converged solution for the temporal discretization [8]. Let λ_i be the eigenvalues of the matrix A . Stability conditions of (2.84) are determined based on real and imaginary parts of the eigenvalue distribution. Let $\Re(\lambda_i)$ be the real part and $Im(\lambda_i)$ be the complex part of the

eigenvalues of A . Then,

- If all eigenvalues are real, $-2.78 < \Delta t \lambda_i < 0$
- If eigenvalues consist of complex part only, $-2\sqrt{2} < \Delta t \lambda_i < -2\sqrt{2}$
- If all eigenvalues are both real and complex, $\Delta t \lambda_i$ should be inside the region given by Figure 2.1.

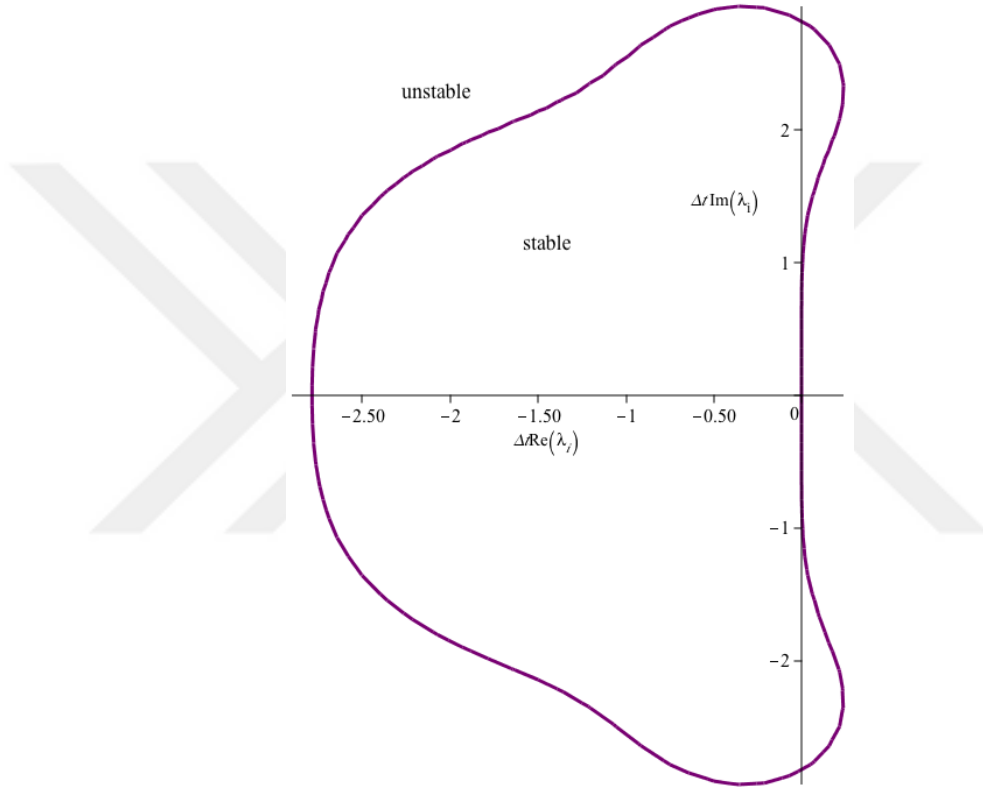


Figure 2.1 Stability region

2.5 Error Norms

To see the validity of the numerical method, there are several error norms. If the analytical solution exists, L_2 , L_∞ and global relative errors are measured since we deal with smooth problems. Let $u(x_i, t)$ represents analytical solution and $U(x_i, t)$ is the approximate solution at a grid point x_i and at time t .

$$L_2 = \sqrt{\sum_{i=1}^N |(u(x_i, t)) - (U(x_i, t))|^2}, \quad (2.86)$$

$$L_{\infty} = \max_{1 \leq i \leq N} |(u(x_i, t)) - (U(x_i, t))|, \quad (2.87)$$

$$GRE = \frac{\sum_{i=1}^N |(u(x_i, t)) - (U(x_i, t))|}{\sum_{i=1}^N |(u(x_i, t))|}. \quad (2.88)$$



APPLICATIONS ON QUADRATURE DISCRETIZATION FOR SOME ORDINARY DIFFERENTIAL EQUATIONS

This section explores numerical solutions of some differential equations by differential quadrature method (DQM). Polynomials are considered as basis functions to establish the weighting coefficients mentioned in Chapter 2. In the beginning, boundary and initial value problems of order one and two were studied. After, higher order linear and nonlinear problems are solved numerically. In the last part of this chapter, third and fourth order singular perturbation problems are solved. To show the effectiveness of the presented scheme, linear and nonlinear perturbation problems are solved. All results are tabulated and results are compared with existing numerical or perturbation methods.

Singular perturbation problems which arise in fluid mechanics, various physical and chemical processes, have been extensively studied [56]. The perturbation equations are given by ordinary differential equations with a small perturbation coefficient denoted by ϵ . Solutions exhibit non-uniform behavior as the small parameter $\epsilon \rightarrow 0$ [57, 58].

The classes of perturbation problems are determined by assuming $\epsilon = 0$. If the order of the problem is reduced by one, the problem becomes convection-diffusion type, on the other hand if the order is reduced by two, it gets reaction-diffusion equation [59]. The accuracy of the method is tested using five examples and quadrature solutions are compared with analytical or existing solutions for various ϵ parameter.

Theoretical and numerical approaches on perturbation problems are extensively studied. Howes [60] studied existence of third order singularly perturbation problem using asymptotic techniques. Yao and Feng [61] also considered lower and upper solutions method for existence of third order two point boundary value problems. Numerical methods such as boundary or initial value techniques are studied for perturbation problems. Results of perturbation equation is obtained by transforming

the model into a system of a first and a second order singularly perturbed equation [62]. Phaneendra et al [63] solved such kind of problems by using Fitted Numerov Method reducing the model to the second order equation. Finite element method [64] is also used to solve such problems numerically including the case of discontinuous source. Cui and Geng [65] investigated third order singular perturbation equation using an analytical method with asymptotic expansion. Kumar and Tiwari [59] studied initial value technique on third order problems. Shanthi and Valarmathi [66] have investigated singular perturbation problems of order four adapting boundary value technique. The DQ is applied to first and second equations in Section 3.1 and Section 3.2 respectively. Higher order BVPs are studied numerically in Section 3.3. Section 3.4 deals with the numerical study of third and fourth order singular perturbation problems.

3.1 Quadrature Solution of the First Order Equations

Let us consider the approximated solution for first order differential equation given by,

$$a(x)y'(x) + b(x)y(x) = f(x), \quad y(0) = \alpha, \quad x \in [0, l]. \quad (3.1)$$

Equation (3.1) can be rewritten by applying grid points x_i , $i = 1, 2, \dots, N$,

$$a(x_i)y'(x_i) + b(x_i)y(x_i) = f(x_i), \quad y(x_1) = y_1 = \alpha \quad (3.2)$$

First derivative is replaced by DQ equality,

$$a(x_i) \sum_{j=1}^N w_{ij}^{(1)} y(x_j) + b(x_i)y(x_i) = f(x_i). \quad (3.3)$$

Grid point distribution is employed using roots of Chebyshev polynomials as explained in Chapter 2. Equation (3.3) can be expanded as,

$$\begin{aligned} a(x_1)\{w_{11}^{(1)}y_1 + w_{12}^{(1)}y_2 + \dots + w_{1N}^{(1)}y_N\} + b(x_1)y_1 &= f_1, \\ a(x_2)\{w_{21}^{(1)}y_1 + w_{22}^{(1)}y_2 + \dots + w_{2N}^{(1)}y_N\} + b(x_2)y_2 &= f_2, \\ &\vdots \\ a(x_N)\{w_{N1}^{(1)}y_1 + w_{N2}^{(1)}y_2 + \dots + w_{NN}^{(1)}y_N\} + b(x_N)y_N &= f_N, \end{aligned} \quad (3.4)$$

where y_i means $y(x_i)$ and f_i means $f(x_i)$. Numerical solutions are obtained after applying initial condition to the system. Thus, following matrix form is obtained,

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ a(x_2)w_{21}^{(1)} & a(x_2)w_{22}^{(1)} + b(x_2) & \cdots & a(x_2)w_{2N}^{(1)} \\ \vdots & \vdots & \vdots & \ddots \\ a(x_N)w_{N1}^{(1)} & a(x_N)w_{N2}^{(1)} & \cdots & a(x_N)w_{NN}^{(1)} + b(x_N) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} \alpha \\ f_2 \\ \vdots \\ f_N \end{bmatrix}.$$

Example 3.1 Consider the first order ordinary differential equation on $0 \leq x \leq 1$,

$$y'(x) + y(x) = 4xe^{2x}, \quad y(0) = 0, \quad (3.5)$$

Analytical solution of the problem is given as,

$$y(x) = \left(\frac{4}{9}(-1 + 3x)e^{3x} + \frac{13}{9} \right) e^{-x}. \quad (3.6)$$

First derivative definition is applied to the equation (3.5),

$$\sum_{j=1}^N w_{ij}^{(1)} y(x_j) + y(x_i) = 4x_i e^{2x_i}. \quad (3.7)$$

Equation (3.7) can be rewritten in the form of system of equations as,

$$[A_{ij}][y(x_j)] = [f(x_i)] \quad (3.8)$$

where $[A_{ij}]$ is first order weighting coefficient matrix. Absolute error is measured for various grid points such as 5, 9 and 13. Numerical results are given by Table 3.1, Table 3.2 and Table 3.3. As the number of grid points increase, accurateness increases as expected.

Table 3.1 Numerical solutions and absolute errors for $N = 5$ of Example 3.1

x_i	y_e	y_i	Absolute Error
0.00000	1.00000	1.00000	0.00000
0.14645	0.91369	0.87220	4.149E-02
0.50000	1.48016	0.42254	3.214E-03
0.85355	4.43914	4.42254	1.660E-02
1.00000	7.09943	7.08869	1.073E-02

Table 3.2 Numerical solutions and absolute errors for $N = 9$ of Example 3.1

x_i	y_e	y_i	Absolute Error
0.00000	1.00000	1.00000	0.00000
0.03806	0.965665	0.965664	5.320E-06
0.14644	0.913691	0.913691	3.002E-08
0.30865	0.999850	0.999849	1.194E-06
0.50000	1.480162	1.480162	7.797E-10
0.69124	2.626023	2.626024	7.133E-07
0.85355	4.439143	4.439143	1.785E-07
0.96193	6.291148	6.291148	3.375E-07
1.00000	7.099431	7.099430	2.996E-07

Table 3.3 Numerical solutions and absolute errors for $N = 13$ of Example 3.1

x_i	y_e	y_i	Absolute Error
0.00000	1.00000	1.00000	0.00000
0.01704	0.98369777	0.98369777	3.854E-10
0.06699	0.94481413	0.94481413	3.750E-10
0.14645	0.91369150	0.91369150	3.434E-10
0.25000	0.94174321	0.94174321	3.278E-10
0.37059	1.10137953	1.10137953	3.117E-10
0.50000	1.48016225	1.48016225	3.298E-10
0.62941	2.15983554	2.15983554	3.719E-10
0.75000	3.17221345	3.17221345	4.582E-10
0.85355	4.43914321	4.43914321	5.692E-10
0.93301	5.73548110	5.73548110	6.903E-10
0.98296	6.72630350	6.72630350	7.833E-10
1.00000	7.09943128	7.09943128	8.188E-10

3.2 Quadrature Solution of the Second Order Equations

We consider the numerical solution for second order boundary value problem on $a \leq x \leq b$ given by,

$$a_2(x)y''(x) + a_1(x)y'(x) + a_0(x)y(x) = f(x), \quad y(a) = \alpha, y(b) = \beta. \quad (3.9)$$

By applying grid points x_i , $i = 1, 2, \dots, N$, (3.9) can be rewritten as,

$$a_2(x_i)y''(x_i) + a_1(x_i)y'(x_i) + a_0(x_i)y(x_i) = f(x_i). \quad (3.10)$$

Derivative expressions are replaced by the DQ form as,

$$a_2(x_i) \sum_{j=1}^N w_{ij}^{(2)} y(x_j) + a_1(x_i) \sum_{j=1}^N w_{ij}^{(1)} y(x_j) + a_0(x_i)y(x_i) = f(x_i). \quad (3.11)$$

By assuming, $y(x_i) = y_i$ and $f(x_i) = f_i$, equation (3.11) can be expanded as,

$$\begin{aligned} a_2(x_1)\{w_{11}^{(2)}y_1 + \dots + w_{1N}^{(2)}y_N\} + a_1(x_1)\{w_{11}^{(1)}y_1 + \dots + w_{1N}^{(1)}y_N\} + a_0(x_1)y_1 &= f_1, \\ a_2(x_2)\{w_{21}^{(2)}y_1 + \dots + w_{2N}^{(2)}y_N\} + a_1(x_2)\{w_{21}^{(1)}y_1 + \dots + w_{2N}^{(1)}y_N\} + a_0(x_2)y_2 &= f_2, \\ \vdots \\ a_2(x_N)\{w_{N1}^{(2)}y_1 + \dots + w_{NN}^{(2)}y_N\} + a_1(x_N)\{w_{N1}^{(1)}y_1 + \dots + w_{NN}^{(1)}y_N\} + a_0(x_N)y_N &= f_N. \end{aligned} \quad (3.12)$$

To obtain accurate numerical solutions, boundary conditions are implemented to the system directly as,

$$y(x_1) = \alpha, y(x_N) = \beta.$$

Example 3.2 Consider the second order boundary value problem on $0 \leq x \leq 1$,

$$y''(x) - y(x) = x, \quad y(0) = 0, \quad y(1) = 0. \quad (3.13)$$

Analytical solution of the problem is given by,

$$y(x) = \frac{e(e^x - e^{-x})}{e^2 - 1}, \quad (3.14)$$

Derivative definition is applied to the equation (3.13),

$$\sum_{j=1}^N w_{ij}^{(2)} y(x_j) - y(x_i) = x_i. \quad (3.15)$$

Equation (3.15) can be rewritten as,

$$[A_{ij}][y(x_j)] = [f(x_i)] \quad (3.16)$$

where $[A_{ij}]$ consists of second order weighting coefficient matrix. After adding two constraints to the system numerical solutions y_i are obtained. Absolute error is

measured for different number of grid points such as 5, 9 and 13. Numerical results are given by Table 3.4, Table 3.5 and Table 3.6. Based on the solutions, it can be concluded that as the number of grid points increase, accurateness increases.

Table 3.4 Numerical solutions and absolute errors for $N = 5$ of Example 3.2

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.14645	-0.02137	-0.02138	1.3909E-05
0.50000	-0.05659	-0.05659	2.2074E-06
0.85355	-0.03580	-0.03578	1.4936E-05
1.00000	0.00000	0.00000	0.00000

Table 3.5 Numerical solutions and absolute errors for $N = 9$ of Example 3.2

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.03806	-0.005666	-0.005666	1.0416E-11
0.14644	-0.021386	-0.021386	2.8140E-13
0.30865	-0.041825	-0.041825	4.4289E-12
0.50000	-0.056590	-0.056590	5.1530E-13
0.69124	-0.055072	-0.055072	4.2204E-12
0.85355	-0.035788	-0.035788	2.3990E-13
0.96193	-0.011201	-0.011201	1.0991E-11
1.00000	0.00000	0.00000	0.00000

Table 3.6 Numerical solutions and absolute errors for $N = 13$ of Example 3.2

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.01704	-0.002539	-0.002539	2.080E-15
0.06699	-0.009943	-0.009943	1.310E-15
0.14645	-0.0213866	-0.021386	1.000E-15
0.25000	-0.035047	-0.035047	00000
0.37059	-0.047980	-0.047980	2.000E-16
0.50000	-0.056590	-0.056590	3.300E-15
0.62941	-0.057764	-0.057764	7.000E-16
0.75000	-0.050275	-0.050275	2.000E-15
0.85355	-0.035788	-0.035788	4.400E-15
0.93301	-0.018790	-0.018790	4.000E-15
0.98296	-0.005189	-0.005189	1.840E-15
1.00000	0.00000	0.00000	0.00000

Example 3.3 Now, we consider the second order initial value problem on $0 \leq x \leq 1$,

$$2y''(x) - 3y' + y(x) = x, \quad y(0) = 2, \quad y'(0) = \frac{1}{2}. \quad (3.17)$$

Analytical solution of the problem is given as,

$$y(x) = 3e^{\frac{1}{2}x} - e^x.$$

Quadrature definition is applied to the equation (3.17),

$$2 \sum_{j=1}^N w_{ij}^{(2)} y(x_j) - 3 \sum_{j=1}^N w_{ij}^{(1)} y(x_j) + y(x_i) = x_i, \quad (3.18)$$

or (3.18) is rewritten equivalently as a system of equations,

$$[2w_{ij}^{(2)} - 3w_{ij}^{(1)} + I][y(x_j)] = [x_i], \quad (3.19)$$

or

$$[A_{ij}][y(x_j)] = [f(x_i)] \quad (3.20)$$

DQ method is applied for the given conditions as,

$$\begin{aligned} y(0) &= y(x_1) = 2, \\ y'(0) &= y'(x_1) = \sum_{j=1}^N w_{1j}^{(1)} y(x_j) = \frac{1}{2}. \end{aligned} \quad (3.21)$$

Here the matrix $[A_{ij}]$ consists of second and first order derivatives. Solution of the following matrix equation gives the numerical solutions y_i .

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2N} \\ \vdots & & & \ddots & \\ A_{(N-1)1} & A_{(N-1)2} & A_{(N-1)3} & \cdots & A_{(N-1)N} \\ w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} & \cdots & w_{1N}^{(1)} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} 2 \\ f_2 \\ \vdots \\ f_{N-1} \\ \frac{1}{2} \end{bmatrix}.$$

Absolute error is measured for different number of grid points such as 5, 9 and 13. Numerical results are given by Table 3.7, Table 3.8 and Table 3.9. We can conclude that as the number of grid points increase, errors get decrease and present approach and exact solutions are in good agreement.

Table 3.7 Numerical solutions and absolute errors for $N = 5$ of Example 3.3

x_i	y_e	y_i	Absolute Error
0.00000	2.000000	2.000000	0.00000
0.14645	2.070079	2.070199	1.1959E-04
0.50000	2.202895	2.203354	4.5992E-04
0.85355	2.248013	2.248956	9.4247E-04
1.00000	2.226618	2.227881	1.2630E-03

Table 3.8 Numerical solutions and absolute errors for $N = 9$ of Example 3.3

x_i	y_e	y_i	Absolute Error
0.00000	2.00000	2.00000	0.00000
0.03806	2.018843	2.018843	7.05420E-11
0.14644	2.070199	2.070199	2.22108E-10
0.30865	2.139027	2.139027	5.17417E-10
0.50000	2.203354	2.203354	9.74500E-10
0.69124	2.242420	2.242420	1.52802E-09
0.85355	2.248956	2.248956	2.15697E-09
0.96193	2.236160	2.236160	2.61652E-09
1.00000	2.227881	2.227881	2.81982E-09

Table 3.9 Numerical solutions and absolute errors for $N = 13$ of Example 3.3

x_i	y_e	y_i	Absolute Error
0.00000	2.00000	2.00000	000000
0.01704	2.008481	2.008481	2.000E-14
0.06699	2.032900	2.032900	1.800E-14
0.14645	2.070199	2.070199	1.400E-14
0.25000	2.115419	2.115419	1.200E-14
0.37059	2.162131	2.162131	1.000E-15
0.50000	2.203354	2.203354	1.600E-14
0.62941	2.233062	2.233062	4.200E-14
0.75000	2.247974	2.247974	6.600E-14
0.85355	2.248956	2.248956	9.500E-13
0.93301	2.241085	2.241085	1.180E-13
0.98296	2.231846	2.231846	1.380E-13
1.00000	2.227881	2.227881	000000

3.3 Quadrature Solution of the Higher Order Boundary Value Problems

We consider the numerical solutions for higher order boundary value problem on $a \leq x \leq l$ given by,

$$p_n(x)y^{(n)}(x) + p_{n-1}(x)y^{(n-1)}(x) + \dots + p_1(x)y'(x) + p_0y(x) = f(x), \quad (3.22)$$

Adapting the nodes x_i , $i = 1, 2, \dots, N$, (3.22) can be written as,

$$p_n(x_i)y^{(n)}(x_i) + p_{n-1}(x_i)y^{(n-1)}(x_i) + \dots + p_1(x_i)y'(x_i) + p_0y(x_i) = f(x_i), \quad (3.23)$$

Equation (3.22) is discretized as,

$$\begin{aligned} p_n(x_i) \sum_{j=1}^N w_{ij}^{(n)} y(x_j) + p_{n-1}(x_i) \sum_{j=1}^N w_{ij}^{(n-1)} y(x_j) + \dots \\ + p_1(x_i) \sum_{j=1}^N w_{ij}^{(1)} y(x_j) + p_0(x_i)y(x_i) = f(x_i), \end{aligned} \quad (3.24)$$

or,

$$\sum_{j=1}^N \{p_n(x_i)w_{ij}^{(n)} + p_{n-1}(x_i)w_{ij}^{(n-1)} + \dots + p_1(x_i)w_{ij}^{(1)} + p_0(x_i)\}y(x_j) = f(x_i) \quad (3.25)$$

where $w_{ij}^{(n)}$ represents n -th order weighting coefficient which is obtained by Chebyshev polynomials. Then, the equation is represented by system of equations as,

$$[p_n(x_i)w_{ij}^{(n)} + p_{n-1}(x_i)w_{ij}^{(n-1)} + \dots + p_1(x_i)w_{ij}^{(1)} + p_0(x_i)][y(x_j)] = [f(x_i)]. \quad (3.26)$$

Numerical solutions are obtained after adding boundary conditions to the system. Here, we consider one linear and one nonlinear fourth order and one fifth order boundary value problems to show the efficiency of the present method.

Example 3.4 Consider the fourth order boundary value problem,

$$y^{(4)}(x) - y(x) = -4(2x \cos x + 3 \sin x), \quad 0 \leq x \leq 1 \quad (3.27)$$

with boundary conditions

$$y(0) = y(1) = 0, \quad y''(0) = 0, \quad y''(1) = 2 \sin 1 + \cos 1. \quad (3.28)$$

Analytical solution of the problem is given in [67] as,

$$y(x) = (x^2 - 1) \sin x.$$

Derivative approximation is applied to (3.28),

$$\sum_{j=1}^N w_{ij}^{(4)} y(x_j) - y(x_i) = -4(2x_i \cos x + 3 \sin x_i). \quad (3.29)$$

Then, (3.29) can be given in matrix form,

$$[-w_{ij}^{(4)} - I][y(x_j)] = [-4(2x_i \cos x + 3 \sin x_i)] \quad (3.30)$$

or in simplified form as,

$$[A_{ij}][y(x_j)] = [f(x_i)]. \quad (3.31)$$

Boundary conditions are discretized as,

$$\begin{aligned} y(0) &= y(x_1) = 0, \quad y(1) = y(x_N) = 0, \\ y''(0) &= y''(x_1) = \sum_{j=1}^N w_{1j}^{(2)} y(x_j) = 0, \\ y''(1) &= y''(x_N) = \sum_{j=1}^N w_{Nj}^{(2)} y(x_j) = 2 \sin 1 + 4 \cos 1. \end{aligned} \quad (3.32)$$

The matrix, $[A_{ij}]$ in (3.31) is the fourth-order weighting coefficient matrix. Adding

four constraints to the system gives the following resultant matrix equation,

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} & \cdots & w_{1N}^{(2)} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3N} \\ \vdots & & & \ddots & \\ w_{N1}^{(2)} & w_{N2}^{(2)} & w_{N3}^{(2)} & \cdots & w_{NN}^{(2)} \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f_3 \\ \vdots \\ 2 \sin 1 + 4 \cos 1 \\ 0 \end{bmatrix}.$$

Solution of the last expression gives the desired solutions y_i . Absolute error is measured for different number of grid points such as 5, 9 and 13. Numerical results are given by Table 3.10, Table 3.11 and Table 3.12. As the number of grid points increase, accurateness increases. Choosing grid points more than 10 gives effective results. Numerical results are also compared with a previous numerical work and presented in Table 3.13. It is concluded that current study gives more accurate results for linear problems.

Table 3.10 Numerical solutions and absolute errors for $N = 5$ of Example 3.4

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.14645	-0.172445	-0.142794	2.9651E-02
0.50000	-0.419438	-0.359569	5.9868E-02
0.85355	-0.229074	-0.204567	2.4507E-02
1.00000	0.00000	0.00000	000000

Table 3.11 Numerical solutions and absolute errors for $N = 9$ of Example 3.4

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.03806	-0.037995	-0.037995	2.6302E-06
0.14644	-0.142794	-0.142794	9.0962E-06
0.30865	-0.274839	-0.274839	1.4597E-05
0.50000	-0.359569	-0.359569	1.5319E-05
0.69124	-0.332842	-0.332842	1.1460E-05
0.85355	-0.204567	-0.204567	5.8914E-06
0.96193	-0.061253	-0.061253	1.5632E-06
1.00000	0.00000	0.00000	000000

Table 3.12 Numerical solutions and absolute errors for $N = 13$ of Example 3.4

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.01704	-0.017031	-0.017031	5.4398E-12
0.06699	-0.066636	-0.066636	2.0429E-11
0.14645	-0.142794	-0.142794	3.9727E-11
0.25000	-0.231941	-0.231941	5.6552E-11
0.37059	-0.312427	-0.312427	6.5702E-11
0.50000	-0.359569	-0.359569	6.4950E-11
0.62941	-0.355463	-0.355463	5.5444E-11
0.75000	-0.298216	-0.298216	4.0679E-11
0.85355	-0.204567	-0.204567	2.4883E-11
0.93301	-0.104032	-0.104032	1.1581E-11
0.98296	-0.028113	-0.028113	2.9581E-12
1.00000	0.00000	0.00000	0.00000

Table 3.13 Comparison of maximum absolute errors for Example 3.4

N	DQ	Galerkin Method [67]
10	7.558E-07	5.275E-06

Example 3.5 Consider the fourth order nonlinear boundary value problem on $0 \leq x \leq 1$,

$$y^{(4)}(x) + (y''(x))^2 = \sin x + \sin^2 x, \quad (3.33)$$

with boundary conditions

$$y(0) = 0, \quad y(1) = \sin 1, \quad y'(0) = 1, \quad y'(1) = \cos 1. \quad (3.34)$$

Analytical solution of the problem is given in [67] as,

$$y(x) = \sin x. \quad (3.35)$$

Application of derivative approach of DQ to (3.28) gives,

$$\sum_{j=1}^N w_{ij}^{(4)} y(x_j) + \left(\sum_{j=1}^N w_{ij}^{(2)} y(x_j) \right)^2 = \sin x_i + \sin^2 x_i. \quad (3.36)$$

Then, (3.36) is represented as,

$$\left[w_{ij}^{(4)} + w_{ij}^{(2)} w_{ij}^{(2)} \right]_{N \times N} [y(x_j)]_{N \times 1} = [\sin x_i + \sin^2 x_i]_{N \times 1} \quad (3.37)$$

Equation (3.37) is rewritten as,

$$[A_{ij}] [y(x_j)] = [f(x_i)] \quad (3.38)$$

Boundary conditions are discretized as,

$$\begin{aligned} y(0) = y(x_1) = 0, \quad y(1) = y(x_N) = \sin 1, \\ y'(0) = y'(x_1) = \sum_{j=1}^N w_{1j}^{(1)} y(x_j) = 1, \\ y'(1) = y'(x_N) = \sum_{j=1}^N w_{Nj}^{(1)} y(x_j) = \cos 1. \end{aligned} \quad (3.39)$$

The matrix, $[A_{ij}]$ in (3.38) consists of including fourth and second order derivatives. After implementing four constraints to the system numerical solutions are obtained. The resultant matrix form is given by,

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} & \cdots & w_{1N}^{(1)} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3N} \\ \vdots & & & \ddots & \\ w_{N1}^{(1)} & w_{N2}^{(1)} & w_{N3}^{(1)} & \cdots & w_{NN}^{(1)} \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ f_3 \\ \vdots \\ \cos 1 \\ \sin 1 \end{bmatrix}.$$

To see the accuracy absolute error is measured for different number of grid points such as 5, 9 and 13. Numerical results are given by Table 3.14, Table 3.15 and Table 3.16. Results are also compared with a previous work related with Galerkin method with quintic B-splines. Both methods are seen to be effective and given by Table 3.17.

Table 3.14 Numerical solutions and absolute errors for $N = 5$ of Example 3.5

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.14645	0.146088	0.145923	1.6432E-04
0.50000	0.479918	0.479425	4.9295E-04
0.85355	0.753705	0.753620	8.4722E-05
1.00000	0.841470	0.841470	0.00000

Table 3.15 Numerical solutions and absolute errors for $N = 9$ of Example 3.5

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.03806	0.038045	0.038051	5.7461E-06
0.14644	0.145854	0.145923	6.9126E-05
0.30865	0.303572	0.303780	2.0819E-04
0.50000	0.479133	0.479425	2.9160E-04
0.69124	0.637358	0.637571	2.1316E-04
0.85355	0.753548	0.753620	7.2534E-05
0.96193	0.820296	0.820302	6.1565E-06
1.00000	0.841470	0.841470	0.00000

Table 3.16 Numerical solutions and absolute errors for $N = 13$ of Example 3.5

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.01704	0.017035	0.017036	1.1972E-06
0.06699	0.066920	0.066937	1.6941E-05
0.14645	0.145854	0.145923	6.9247E-05
0.25000	0.247244	0.247403	1.5932E-04
0.37059	0.361914	0.362165	2.5112E-04
0.50000	0.479133	0.479425	2.9180E-04
0.62941	0.588412	0.588667	2.5499E-04
0.75000	0.681474	0.681638	1.6434E-04
0.85355	0.753548	0.753620	7.2564E-05
0.93301	0.803399	0.803417	1.8006E-05
0.98296	0.832142	0.832144	1.2854E-06
1.00000	0.84147	0.841470	0.00000

Table 3.17 Comparison of maximum absolute errors of Example 3.5

N	DQ	Galerkin Method [67]
10	2.753E-04	1.358E-05

Example 3.6 Consider the fifth order linear boundary value problem on $0 \leq x \leq 1$,

$$y^{(v)}(x) - y(x) = -15e^x - 10xe^x \quad (3.40)$$

with boundary conditions

$$y(0) = 0, y(1) = 0, y'(0) = 1, y''(0) = 0, y'(1) = -e. \quad (3.41)$$

Analytical solution of the problem is given in [68] as,

$$y(x) = x(1 - x)e^x. \quad (3.42)$$

Derivative approximation is applied to (3.40),

$$\sum_{j=1}^N w_{ij}^{(5)} y(x_j) - y(x_j) = -15e^{x_i} - 10xe^{x_i}. \quad (3.43)$$

Then, Equation (3.43) is represented by system of equations as,

$$[A_{ij}][y(x_j)] = [f(x_i)] \quad (3.44)$$

Boundary conditions are discretized as,

$$\begin{aligned} y(0) = y(x_1) = 0, \quad y(1) = y(x_N) = 0, \\ y'(0) = y'(x_1) = \sum_{j=1}^N w_{1j}^{(1)} y(x_j) = 1, \\ y''(0) = y''(x_1) = \sum_{j=1}^N w_{1j}^{(2)} y(x_j) = 0, \\ y'(1) = y'(x_N) = \sum_{j=1}^N w_{1j}^{(1)} y(x_j) = -e. \end{aligned} \quad (3.45)$$

The matrix, $[A_{ij}]$ in (3.44) is the weighting coefficient matrix, including fifth order derivative. To obtain numerical solution, boundary conditions are implemented to

the system. Hence, the resultant matrix equation becomes,

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} & \cdots & w_{1N}^{(1)} \\ w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} & \cdots & w_{1N}^{(2)} \\ A_{41} & A_{42} & A_{43} & \cdots & A_{4N} \\ \vdots & & & \ddots & \\ w_{N1}^{(1)} & w_{N2}^{(1)} & w_{N3}^{(1)} & \cdots & w_{NN}^{(1)} \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ f_4 \\ \vdots \\ -e \\ 0 \end{bmatrix}.$$

Absolute error is measured for different number of grid points and presented in Table 3.18, Table 3.19 and Table 3.20. As seen in previous examples, method gives better solutions if the grid spacing is chosen as 13 so we did not consider grid points more than 13.

Table 3.18 Numerical solutions and absolute errors for $N = 5$ of Example 3.6

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.14645	0.145231	0.144714	5.1727E-04
0.50000	0.419892	0.412180	7.7122E-03
0.85355	0.297107	0.293496	3.6105E-03
1.00000	0.00000	0.00000	0.00000

Table 3.19 Numerical solutions and absolute errors for $N = 9$ of Example 3.6

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.03806	0.038032	0.038031	1.1556E-07
0.14644	0.144718	0.144714	4.3846E-06
0.30865	0.290570	0.290548	2.1245E-05
0.50000	0.412217	0.412180	3.6725E-05
0.69124	0.426036	0.426006	2.9796E-05
0.85355	0.293507	0.293496	1.0787E-05
0.96193	0.095805	0.095804	9.4951E-07
1.00000	0.00000	0.00000	0.00000

Table 3.20 Numerical solutions and absolute errors for $N = 13$ of Example 3.6

x_i	y_e	y_i	Absolute Error
0.00000	0.00000	0.00000	0.00000
0.01704	0.017034	0.017034	1.9586E-13
0.06699	0.066830	0.066830	8.7157E-12
0.14645	0.144714	0.144714	5.6828E-11
0.25000	0.240754	0.240754	1.5911E-10
0.37059	0.337888	0.337888	2.7336E-10
0.50000	0.412180	0.412180	3.3220E-10
0.62941	0.437700	0.437700	2.9891E-10
0.75000	0.396937	0.396937	1.9715E-10
0.85355	0.293496	0.293496	8.8771E-11
0.93301	0.158884	0.158884	2.2371E-11
0.98296	0.044753	0.044753	1.6132E-12
1.00000	0.00000	0.00000	0.00000

3.4 Quadrature Solutions of the Third and Fourth Order Singularly Perturbed Problems

In this section, a number of classes of higher order perturbation problems are studied. Third and fourth order problems are discretized by present method. These models are solved in [69] using current scheme. To see the validity, five examples are solved. All results are tabulated and figures for different perturbation parameters are presented.

3.4.1 Quadrature Discretization of Third Order Perturbation Problems

Third order perturbation problem is given by the following ordinary differential equation, which is classified as reaction diffusion type [59],

$$\epsilon y'''(x) + p(x)y'(x) + q(x)y(x) = f(x), \quad x \in [a, b], \quad (3.46)$$

with boundary conditions,

$$y(a) = a_0, \quad y(b) = b_0 \quad y'(a) = c_0, \quad (3.47)$$

where $\epsilon > 0$ is the small perturbation parameter. Coefficients a_0, b_0, c_0 are constants and $p(x), q(x)$ and $f(x)$ are smooth functions. Domain of the problem is given on $a \leq x \leq b$. To discretize the model problem, (3.47) is rewritten as,

$$\epsilon y'''(x_i) + p(x_i)y'(x_i) + q(x_i)y(x_i) = f(x_i). \quad (3.48)$$

The DQ method is applied to (3.48)

$$\epsilon \sum_{j=1}^N w_{ij}^{(3)} y(x_j) + p(x_i) \sum_{j=1}^N w_{ij}^{(1)} y(x_j) + q(x_i)y(x_i) = f(x_i). \quad (3.49)$$

Equation (3.49) can be rewritten in a simplified form, where $w_{ij}^{(k)}$ represents k -th order weighting coefficient, (3.49) is rewritten in vector-matrix form,

$$[\epsilon w_{ij}^{(3)} + p(x_i)w_{ij}^{(1)} + q(x_i)I][y(x_j)] = [f(x_i)], \quad (3.50)$$

or

$$[A_{ij}][y(x_j)] = [f(x_i)]. \quad (3.51)$$

The DQ method is implemented to the boundary conditions,

$$\begin{aligned} y(a) = y(x_1) = a_0, \quad y(b) = y(x_N) = b_0, \\ y'(a) = y'(x_1) = \sum_{j=1}^N w_{1j}^{(1)} y(x_j) = c_0. \end{aligned} \quad (3.52)$$

After using the given constraints given in (3.52), we obtain the following system,

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} & \cdots & w_{1N}^{(1)} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3N} \\ \vdots & & & \ddots & \\ A_{(N-1)1} & A_{(N-1)2} & A_{(N-1)3} & \cdots & A_{(N-1)N} \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \\ f_3 \\ \vdots \\ f_{N-1} \\ b_0 \end{bmatrix}.$$

The resultant matrix equation is solved to get approximated solutions. Here, LU decomposition is used to solve the system.

3.4.2 Quadrature Discretization of Fourth Order Perturbation Problems

In this section, fourth order singularly perturbed BVP is investigated. This model is given by,

$$-\epsilon y''''(x) - p(x)y''' + q(x)y''(x) - r(x)y(x) = -f(x), \quad x \in (0, 1), \quad (3.53)$$

with boundary conditions,

$$y(0) = a_0, \quad y(1) = b_0, \quad y''(0) = -c_0, \quad y''(1) = -d_0,$$

where $\epsilon > 0$ is perturbation parameter and $p(x), q(x), r(x)$ and $f(x)$ are functions which are differentiable on $(0, 1)$. DQ method is applied to Equation (3.53) as,

$$-\epsilon \sum_{j=1}^N w_{ij}^{(4)} y(x_j) - p(x_i) \sum_{j=1}^N w_{ij}^{(3)} y(x_j) + q(x_i) \sum_{j=1}^N w_{ij}^{(2)} y(x_j) - r(x_i) y(x_i) = -f(x_i). \quad (3.54)$$

Equation (3.54) can be represented as,

$$[A_{ij}][y(x_j)] = [f(x_i)], \quad (3.55)$$

where $[A_{ij}]$ is the weighting coefficient matrix. The method is employed for the boundary conditions,

$$\begin{aligned} y(0) &= y(x_1) = a_0, \quad y(1) = y(x_N) = b_0, \\ y''(0) &= y''(x_1) = \sum_{j=1}^N w_{2j}^{(2)} y(x_j) = -c_0, \\ y''(1) &= y''(x_N) = \sum_{j=1}^N w_{Nj}^{(2)} y(x_j) = -d_0. \end{aligned} \quad (3.56)$$

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} & \cdots & w_{1N}^{(2)} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3N} \\ \vdots & & & \ddots & \\ w_{N1}^{(2)} & w_{N2}^{(2)} & w_{N3}^{(2)} & \cdots & w_{NN}^{(2)} \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} a_0 \\ -c_0 \\ f_3 \\ \vdots \\ -d_0 \\ b_0 \end{bmatrix}.$$

The resultant matrix equation is solved to get approximated solutions. LU decomposition is used to solve the system. Now, five illustrative examples are presented to test the validity of the DQ method.

Example 3.4.1

$$\begin{aligned}\epsilon y''' + 4y' - 4y &= x^2, \\ y(0) = 0.5, \quad y(1) = 1.47 \quad y'(0) &= 0.5.\end{aligned}\tag{3.57}$$

Example 3.4.1 is solved by Initial Value Technique (IVT) [59]. The problem is discretized using differential quadrature algorithm as,

$$\epsilon \sum_{j=1}^N w_{ij}^{(3)} y(x_j) + 4 \sum_{j=1}^N w_{ij}^{(1)} y(x_j) - 4y(x_i) = x_i^2 \tag{3.58}$$

or in matrix form

$$[\epsilon w_{ij}^{(3)} + 4w_{ij}^{(1)} - 4I][y(x_j)] = [x_i^2]. \tag{3.59}$$

Boundary conditions are implemented as,

$$\begin{aligned}y(0) = y(x_1) &= 0.5, \quad y(1) = y(x_N) = 1.47, \\ y'(0) = y'(x_1) &= \sum_{j=1}^N w_{1j}^{(1)} y(x_j) = 0.5.\end{aligned}\tag{3.60}$$

Errors between analytical and numerical methods are computed using different perturbation parameters. Quadrature and analytical solutions are depicted by Figure 3.1. As small perturbation parameter increase, more accurate results are obtained. The results are compared with IVT and observed that presented method gives more accurate results, Tables 3.21-3.23. In addition, solutions are more accurate as $x \rightarrow 0$ where boundary layer occurs.

Table 3.21 Comparison of absolute errors of Example 3.4.1 for $\epsilon = 2^{-3}$

x	DQ	IVT [59]
0.0000	0.0000	0.0000
0.0001	4.0803E-12	5.9049E-09
0.0010	2.9995E-12	5.9033E-07
0.0100	1.3356E-12	5.8867E-05
0.0300	2.5316E-12	5.2566E-04
0.1000	3.1185E-12	5.5841E-03
0.3000	4.2729E-11	0.0379E-02
0.5000	7.2330E-12	6.2041E-02
0.7000	3.1879E-12	5.0860E-02
0.9000	2.0270E-12	1.7090E-02
1.0000	0.0000	0.0000

Table 3.22 Comparison of absolute errors of Example 3.4.1 for $\epsilon = 2^{-6}$

x	DQ	IVT [59]
0.0000	0.0000	0.0000
0.0001	5.5546E-13	4.7400E-09
0.0010	3.8232E-12	4.7384E-07
0.0100	2.5430E-11	4.7134E-05
0.0300	5.6183E-11	4.1405E-04
0.1000	1.0460E-11	3.6784E-03
0.3000	8.1553E-11	3.2214E-03
0.5000	9.0121E-11	3.2902E-03
0.7000	6.3508E-11	1.5055E-03
0.9000	7.9107E-11	1.1480E-03
1.0000	0.0000	0.0000

Table 3.23 Comparison of absolute errors of Example 3.4.1 for $\epsilon = 2^{-9}$

x	DQ	IVT [59]
0.0000	0.0000	0.0000
0.0001	1.4086E-09	1.2330E-09
0.0010	1.0533E-08	1.2326E-07
0.0100	2.2326E-08	1.2103E-05
0.0300	3.2396E-08	9.4622E-05
0.1000	5.2096E-08	1.5146E-03
0.3000	6.9652E-09	8.5658E-04
0.5000	5.9590E-09	2.3461E-05
0.7000	7.5398E-09	9.1170E-04
0.9000	4.9060E-08	2.6400E-04
1.0000	0.0000	0.0000

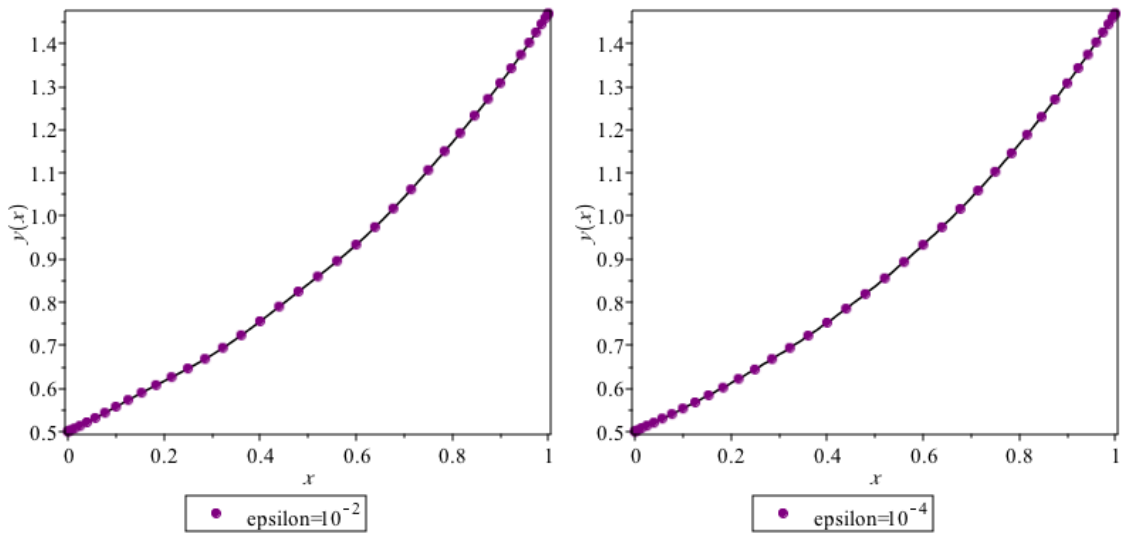


Figure 3.1 Comparison of solutions for Example 3.4.1, numerical solution (dotted line), exact solution (solid line).

Example 3.4.2

$$\begin{aligned} \epsilon y''' + \left(1 + \frac{x}{2}\right) y' - \frac{1}{2} y &= 0, \\ y(0) &= 0.6, \quad y(1) = 0.9 \quad y'(0) = 0.23. \end{aligned} \quad (3.61)$$

The example given in (3.61) is examined by IVT in [59]. The problem is discretized

using differential quadrature algorithm as,

$$\epsilon \sum_{j=1}^N w_{ij}^{(3)} y(x_j) + \left(1 + \frac{x_i}{2}\right) \sum_{j=1}^N w_{ij}^{(1)} y(x_j) - \frac{1}{2} y(x_j) = 0. \quad (3.62)$$

Equation (3.52) is written in matrix form

$$\left[\epsilon w_{ij}^{(3)} + \left(1 + \frac{x_i}{2}\right) w_{ij}^{(1)} - \frac{1}{2} I \right] [y(x_j)] = 0. \quad (3.63)$$

Boundary conditions are implemented as follows:

$$\begin{aligned} y(0) &= y(x_1) = 0.6, \quad y(1) = y(x_N) = 0.9, \\ y'(0) &= y'(x_1) = \sum_{j=1}^N w_{1j}^{(1)} y(x_j) = 0.23. \end{aligned} \quad (3.64)$$

Errors between analytical and numerical methods are computed using different perturbation parameters and presented in Figure 3.2. The results are compared with initial value technique, Tables 3.24-3.26. As small perturbation parameter increase, more accurate results obtained. It is observed that the presented method gives more accurate results. Also, solutions are more accurate as $x \rightarrow 0$ where boundary layer occurs.

Table 3.24 Comparison of absolute errors of Example 3.4.2 for $\epsilon = 2^{-3}$

x	DQ	IVT [59]
0.0000	0.0000	0.0000
0.0001	1.7570E-12	2.9556E-08
0.0010	1.9643E-12	4.1225E-06
0.0100	2.5038E-12	5.1848E-05
0.0300	3.2534E-12	1.7770E-03
0.1000	4.5989E-12	1.9112E-03
0.3000	4.8178E-12	3.1617E-03
0.5000	3.1264E-12	1.1798E-02
0.7000	1.5297E-12	5.0860E-02
0.9000	2.3540E-13	1.7090E-02
1.0000	0.0000	0.0000

Table 3.25 Comparison of absolute errors of Example 3.4.2 for $\epsilon = 2^{-6}$

x	DQ	IVT [59]
0.0000	0.0000	0.0000
0.0001	1.2089E-12	2.5234E-08
0.0010	1.1556E-12	4.1225E-06
0.0100	9.3170E-12	3.2961E-05
0.0300	5.4370E-13	1.3783E-04
0.1000	6.7030E-13	9.6782E-03
0.3000	6.7770E-14	3.0945E-03
0.5000	3.2022E-12	7.7235E-03
0.7000	2.2141E-12	9.7253E-03
0.9000	3.3340E-13	1.5692E-02
1.0000	0.0000	0.0000

Table 3.26 Comparison of absolute errors of Example 3.4.2 for $\epsilon = 2^{-9}$

x	DQ	IVT [59]
0.0000	0.0000	0.0000
0.0001	6.3730E-12	1.2330E-08
0.0010	1.8984E-08	4.0725E-06
0.0100	1.4160E-07	2.9034E-05
0.0300	2.9550E-07	9.5372E-04
0.1000	2.5050E-07	7.9146E-03
0.3000	2.7559E-07	2.8450E-03
0.5000	2.7559E-07	5.6283E-03
0.7000	3.1095E-07	6.8244E-03
0.9000	1.5620E-07	2.6436E-03
1.0000	0.0000	0.0000

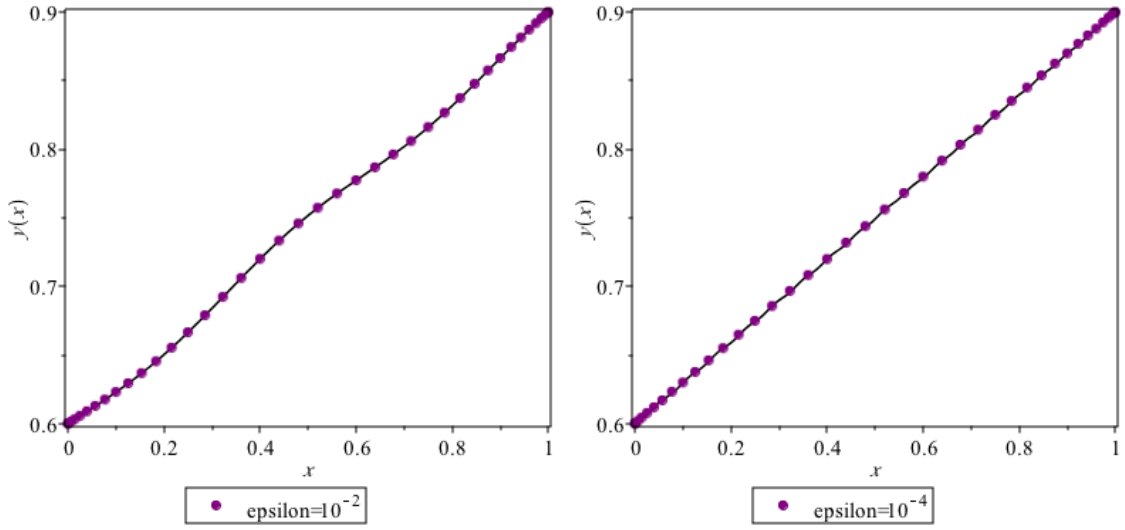


Figure 3.2 Comparison of solutions for Example 3.4.2, numerical solution (dotted line), exact solution (solid line).

Example 3.4.3

$$\begin{aligned} -\epsilon y'''' - 4y''' &= 1, \quad 0 < x < 1, \\ y(0) &= 1, \quad y(1) = 1 \quad y''(0) = -1, \quad y''(1) = -1. \end{aligned} \quad (3.65)$$

Equation (3.65) is studied by Boundary Value Technique (BVT) in [66]. After DQ application, present result is compared with BVT.

Now, DQ algorithm is applied as,

$$-\epsilon \sum_{j=1}^N w_{ij}^{(4)} y(x_j) - 4 \sum_{j=1}^N w_{ij}^{(3)} y(x_j) = 1 \quad (3.66)$$

or in matrix form

$$\left[-\epsilon w_{ij}^{(4)} - 4w_{ij}^{(3)} \right] [y(x_j)] = 1. \quad (3.67)$$

Boundary conditions are discretized as,

$$\begin{aligned} y(0) &= y(x_1) = 1, \quad y(1) = y(x_N) = 1, \\ y''(0) &= y''(x_1) = \sum_{j=1}^N w_{1j}^{(2)} y(x_j) = -1, \\ y''(1) &= y''(x_N) = \sum_{j=1}^N w_{Nj}^{(2)} y(x_j) = -1. \end{aligned} \quad (3.68)$$

Errors between analytical and numerical results are computed using different perturbation parameters with different number of space grids Table 3.27, Figure 3.3. The results are also compared with boundary value technique and given by Table 3.28. Both methods give highly effective results.

Table 3.27 Error Norms for Example 3.4.3

Norms	N	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$
L_2	20	8.03036E-05	7.89303E-05	7.86227E-05
	50	1.36130E-06	3.11759E-06	3.05975E-06
	100	9.66820E-13	2.51022E-07	2.68298E-07
L_∞	20	3.15926E-05	3.10204E-05	3.08603E-05
	50	3.29883E-07	7.44577E-07	7.23180E-07
	100	1.64111E-13	4.20483E-08	4.41987E-08

Table 3.28 Comparison of absolute errors between different methods for Example 3.4.3 when $\epsilon = 10^{-3}$

x	DQ	BVT [66]
0.0000	0.0000	0.0000
0.0003	4.703905E-08	4.127016E-09
0.2000	8.719068E-08	2.370865E-08
0.3000	7.553848E-08	1.621495E-09
0.4000	5.836873E-08	7.137186E-09
0.5000	5.396490E-08	2.543137E-08
0.6000	4.306175E-08	8.116636E-09
0.7000	3.265890E-08	2.852202E-08
0.8000	2.139622E-08	4.563018E-10
0.9000	1.072529E-08	3.535062E-09
1.0000	0.0000	0.0000

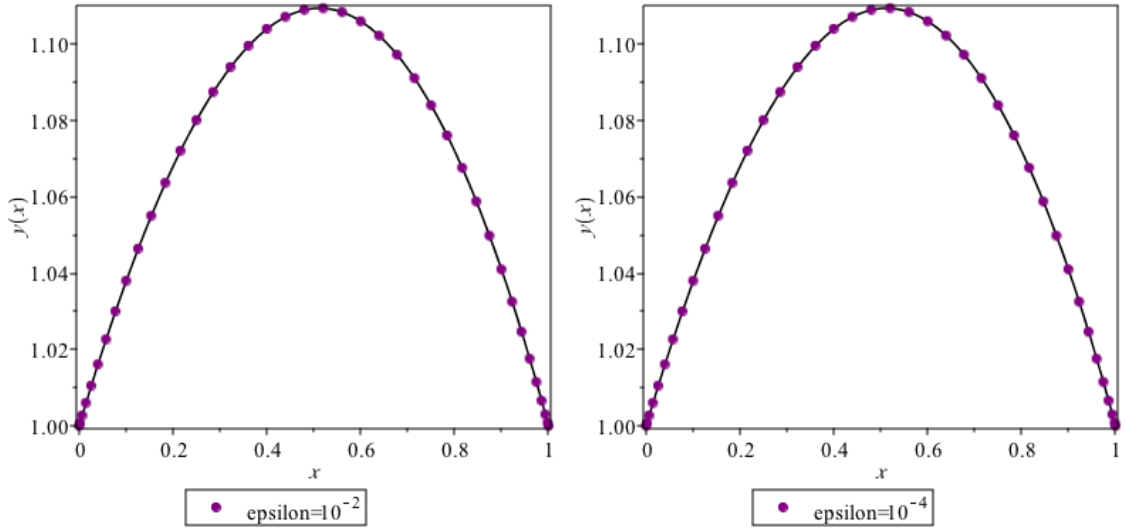


Figure 3.3 Comparison of solutions for Example 3.4.3, numerical solution (dotted line), exact solution (solid line).

Example 3.4.4 Nonlinear BVP

$$\begin{aligned} \epsilon y''' + y'' + \epsilon [(y')^2 + 2y'] + y &= 1, \quad 0 \leq x \leq \frac{\pi}{2}, \\ y(0; \epsilon) &= 0, \quad y\left(\frac{\pi}{2}; \epsilon\right) = 1 - \frac{\epsilon}{3}, \quad y'\left(\frac{\pi}{2}; \epsilon\right) = -1 + \frac{\epsilon\pi}{4}. \end{aligned} \quad (3.69)$$

The example given by Equation (3.69) is solved asymptotically in [56]. The nonlinear term $(y')^2$ is discretized as follows:

$$y_x(x_i) = \sum_{j=1}^N w_{ij} y(x_j) \quad (3.70)$$

where w_{ij} represents the first order weighting coefficients. Then,

$$y_x(x_i) y_x(x_i) = \sum_{j=1}^N w_{ij} y(x_j) \sum_{j=1}^N w_{ij} y(x_j) \quad (3.71)$$

or

$$\{y_x(x_i)\}^2 = \sum_{j=1}^N \{w_{ij}\}^2 H(x_j). \quad (3.72)$$

So the equation (3.69) can be represented as,

$$\left[\epsilon w_{ij}^{(3)} + w_{ij}^{(2)} + 2\epsilon w_{ij}^{(2)} \right] [y(x_j)] + \left[\{w_{ij}^{(2)}\}^2 H(x_j) \right] = 1. \quad (3.73)$$

where $H(x_j)$ vector is obtained by square of function values at grid points. So, the

nonlinear term $(y')^2$ is discretized by using DQ weighted sum of square of functional values at all discrete points [38]. Boundary conditions are implemented as,

$$\begin{aligned} y(0) = y(x_1) = 0, \quad y\left(\frac{\pi}{2}\right) = y(x_N) = 1 - \frac{\epsilon}{3}, \\ y'\left(\frac{\pi}{2}\right) = y'(x_N) = \sum_{j=1}^N w_{Nj}^{(1)} y(x_j) = -1 + \frac{\epsilon\pi}{4}. \end{aligned} \quad (3.74)$$

Numerical results are presented using different number of grid points and different perturbation parameter, Table 3.29, Table 3.30. We observe that results are better as $\epsilon \rightarrow 0$. Graphs of numerical solutions are given by Figure 3.4. Numerical and asymptotic results are well-matched.

Table 3.29 Absolute errors for $N = 10$

x_i	$\epsilon = 2^{-3}$	$\epsilon = 2^{-6}$	$\epsilon = 2^{-9}$
0.00000	0.0000000	0.0000000	0.0000000
0.04736	3.7335619E-02	1.2119630E-02	1.8878739E-03
0.18374	7.9491326E-02	1.5372186E-02	2.2135893E-03
0.39269	7.4803051E-02	1.0948929E-02	1.6247999E-03
0.64901	4.9647666E-02	7.2890856E-03	1.1314067E-03
0.92178	2.5020504E-02	3.5269846E-03	6.0448946E-04
1.17809	9.0508809E-03	1.3820846E-03	2.7431769E-04
1.38704	1.8843521E-03	2.4844445E-04	8.0858917E-05
1.52343	1.2271346E-04	7.2277373E-05	2.2166443E-05
1.57079	0.0000000	0.0000000	0.0000000

Table 3.30 Error Norms for Example 3.4.4

Norms	N	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$
L_2	20	2.41305E-02	2.807329E-03	3.03592E-04
	50	3.95775E-02	4.21815E-03	4.85272E-04
	100	5.62559E-02	6.12589E-03	6.35555E-04
L_∞	20	1.09241E-02	1.25137E-03	1.15486E-04
	50	1.08975E-02	1.16737E-03	1.31765E-04
	100	1.09219E-02	1.12775E-03	1.18752E-04

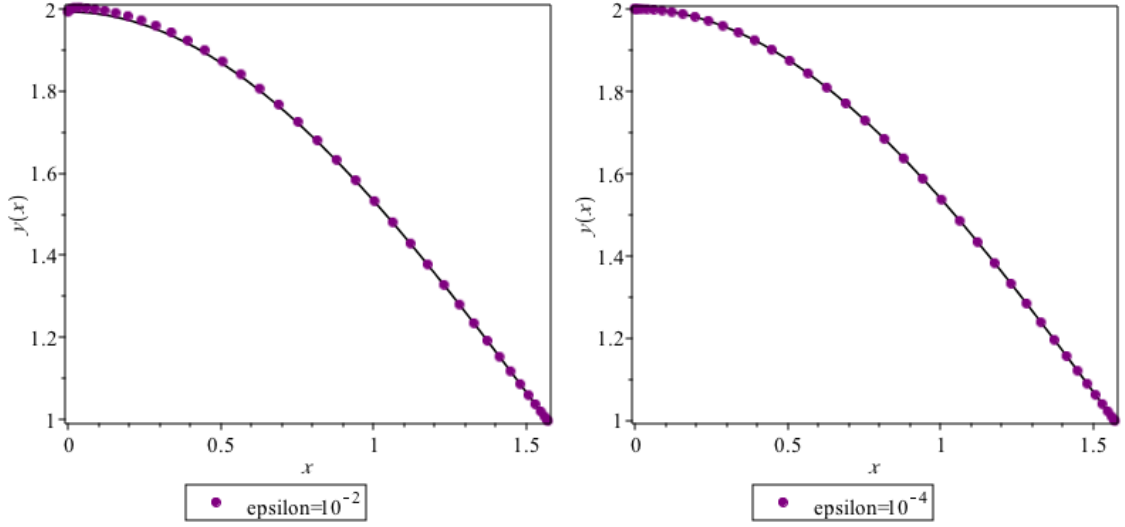


Figure 3.4 Comparison of solutions for Example 3.4.4, numerical solution (dotted line), exact solution (solid line).

Example 3.4.5 Nonlinear IVP [56]:

$$\begin{aligned} y''' + y'' + \epsilon [(y')^2 + y] &= \epsilon e^{-2x}, \quad x \geq 0, \\ y(0; \epsilon) &= 2, \quad y'(0; \epsilon) = -1 \quad y''(0; \epsilon) = 1. \end{aligned} \quad (3.75)$$

The non-linearity of the problem is analyzed using similar procedure given in Problem 3.4.4. So (3.75) is discretized as,

$$\sum_{j=1}^N w_{ij}^{(3)} y(x_j) + \sum_{j=1}^N w_{ij}^{(2)} y(x_j) + \epsilon \left[\sum_{j=1}^N \{w_{ij}^{(2)}\}^2 H(x_j) + y(x_i) \right] = \epsilon e^{-2x}. \quad (3.76)$$

Boundary conditions are discretized as,

$$\begin{aligned} y(0) &= y(x_1) = 2, \quad y'(0) = y'(x_1) = \sum_{j=1}^N w_{1j}^{(1)} y(x_j) = -1, \\ y''(0) &= y''(x_1) = \sum_{j=1}^N w_{1j}^{(2)} y(x_j) = 1. \end{aligned} \quad (3.77)$$

Numerical results are presented using different number of grid points and different perturbation parameter, Table 3.31, Table 3.32. It is observed that results are better as $\epsilon \rightarrow 0$, as well as $x \rightarrow 0$. Graphs of numerical solutions are given by Figure 3.5. In addition, results are approximately same for grid points $N = 20$ and $N = 50$ but more effective when compared to $N = 100$.

Table 3.31 Absolute errors for $N = 10$

x_i	$\epsilon = 2^{-3}$	$\epsilon = 2^{-6}$	$\epsilon = 2^{-9}$
0.00000	0.0000000	0.0000000	0.0000000
0.03015	1.1284872E-06	1.4117507E-07	1.7646933E-08
0.11697	6.3586312E-05	7.9735834E-06	9.9702782E-07
0.25000	5.8757288E-04	7.3946828E-05	9.2508359E-06
0.41317	2.4785985E-03	3.1331781E-04	3.9218769E-05
0.58682	6.6040254E-03	8.3877317E-04	1.0505471E-04
0.75000	1.2875218E-02	1.6426687E-03	2.0585904E-04
0.88302	1.9871632E-02	2.5448133E-03	3.1906734E-04
0.96984	2.5386175E-02	3.2591185E-03	4.0875552E-04
1.00000	2.7478437E-02	3.5308080E-03	4.4287960E-04

Table 3.32 Error Norms for Example 3.4.5

Norms	N	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$
L_2	20	5.08206E-03	5.09283E-04	5.09391E-05
	50	7.90977E-03	7.92645E-04	7.92813E-05
	100	1.11126E-02	1.11496E-03	1.11520E-04
L_∞	20	2.26293E-03	2.26881E-04	2.26863E-05
	50	2.26629E-03	2.26811E-04	2.26863E-05
	100	2.26293E-03	2.26811E-04	2.26863E-05

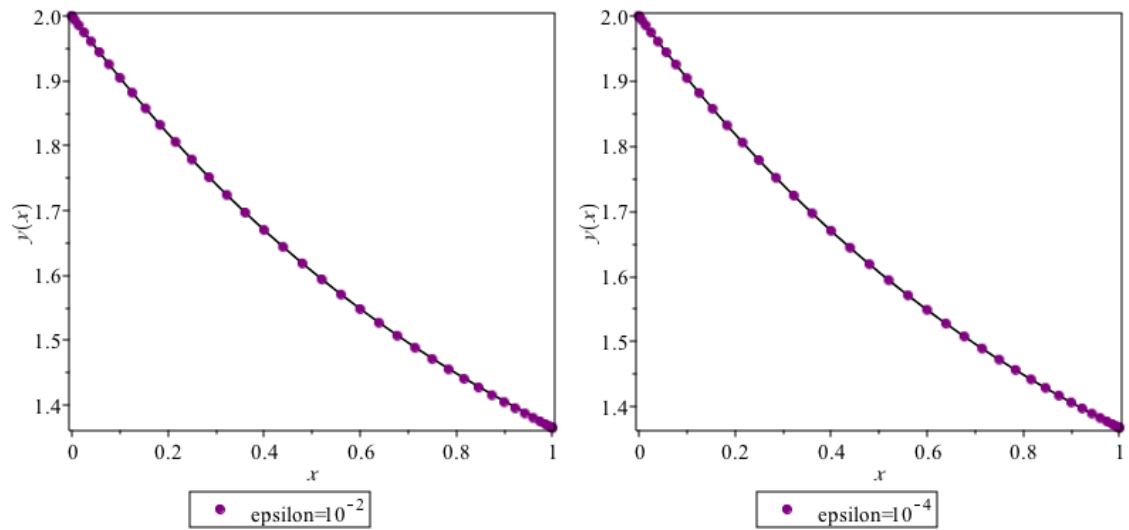


Figure 3.5 Comparison of solutions for Example 3.4.5, numerical solution (dotted line), exact solution (solid line).

APPLICATIONS ON QUADRATURE DISCRETIZATION FOR SOME PARTIAL DIFFERENTIAL EQUATIONS

This chapter deals with numerical schemes of linear and nonlinear partial differential equations. Spatial discretization of the such equations are performed by DQ method. Weighting coefficients matrix which is essential to obtain quadrature solutions is calculated mentioned in Sections 2.3.3 and 2.3.4. For time integration, fourth order Runge-Kutta method is used. Matrix stability analysis is performed for all equations based on the DQ discretized matrix. The DQ is applied to linear and nonlinear diffusion equations in Section 4.1. Section 4.2 presents DQ application of reaction-diffusion equations on nonuniform media. Numerical simulation of Kuramoto-Sivashinsky equation is given in Section 4.3. Section 4.4 deals with the study of nonlinear advection problems and inviscid Burger's equation. In Section 4.5, numerical solutions of Ginzburg-Landau equation is solved numerically.

4.1 Quadrature Solution of Linear and Nonlinear Diffusion Equations

Diffusion equations have been presented as models of many physical and chemical processes, groundwater flow, heat conduction, etc. [70–74]. The model has been extensively studied numerical or analytical [71–73, 75, 76]. Akman [44] investigated diffusion equations in two dimensions numerically by the present technique with different boundary conditions. Chen [77] studied steady-state heat conduction problems by differential quadrature element method. Guraslan and Sari [72] solved one-dimensional diffusion equations using polynomial based differential quadrature. They used Lagrange polynomials based DQ to discretize spatial derivatives and Runge-Kutta method for time integration. Guraslan [71] studied same problems numerically using compact finite difference method.

In this section, we consider linear and nonlinear diffusion equations given by the

following equations,

$$U_t = U_{xx} + g(x, t), \quad 0 < x < l, \quad t > 0, \quad (4.1)$$

$$U_t = (D(U)U_x)_x, \quad 0 < x < l, \quad t > 0, \quad (4.2)$$

subject to following initial and Dirichlet boundary conditions,

$$U(x, 0) = f(x), \quad U(0, t) = F_1, \quad U(l, t) = F_2. \quad (4.3)$$

The function $g(x, t)$ represents source term and given as linear. The function $D(U)$ denotes the diffusion coefficient of density. $D(U)$ is represented in various forms such as exponential and power functions. Here, we studied the case for $D(U) = U^n$ when $n = 2$ which exhibits slow diffusion process [76].

4.1.1 Discretization of Linear Diffusion Equation

Equation (4.1) by using nodal points x_i ,

$$U_t(x_i, t) = kU_{xx}(x_i, t) + g(x_i, t). \quad (4.4)$$

Then, for spatial discretization DQ formulations are applied as,

$$\frac{\partial U(x_i, t)}{\partial t} = k \sum_{j=1}^N w_{ij}^{(2)} U(x_j, t) + g(x_i, t), \quad (4.5)$$

or

$$\frac{\partial U(x_i, t)}{\partial t} = [kw_{ij}^{(2)}]U(x_j, t) + g(x_i, t). \quad (4.6)$$

Also, we implemented the method for boundary and initial conditions,

$$U(0, t) = U(x_1, t) = F_1, \quad U(l, t) = U(x_N, t) = F_2, \quad U(x_i, 0) = f(x_i). \quad (4.7)$$

The system can be expressed as,

$$\{T\} = [A]\{U\} + \{B\} \quad (4.8)$$

where the vectors T and B represent time derivative and boundary conditions vector respectively and coefficient matrix A consists of second order derivative. Time integration of the system is adapted by Runge-Kutta fourth order method. Stability analysis has been established depends on the eigenvalue distribution of the numerical

discretized matrix $[A]$. We consider eigenvalues of the following matrix $[A]$ by considering different number of grid points. So, the resultant matrix form becomes as follows:

$$\begin{bmatrix} \frac{dU(x_2, t)}{dt} \\ \frac{dU(x_3, t)}{dt} \\ \vdots \\ \frac{dU(x_{N-1}, t)}{dt} \end{bmatrix} = A \begin{bmatrix} U(x_2, t) \\ U(x_3, t) \\ \vdots \\ U(x_{N-1}, t) \end{bmatrix} + B \quad (4.9)$$

We will investigate problem (4.1) to validate our approach in terms of stability and comparisons with exact and previous studies. We analyzed homogeneous and inhomogeneous test problems and reported numerical solutions via tables and figures.

4.1.2 Discretization of Nonlinear Diffusion Equation

Equation (4.2) is represented as,

$$U_t = 2U(U_x)^2 + U^2 U_{xx} \quad (4.10)$$

Spatial discretization is performed using DQ formulations as,

$$\frac{\partial U(x_i, t)}{\partial t} = 2U(x_i, t) \sum_{j=1}^N (w_{ij}^{(1)} U(x_j, t))^2 + U^2(x_i, t) \sum_{j=1}^N (w_{ij}^{(2)}) U(x_j, t) \quad (4.11)$$

Also, boundary and initial conditions are adapted as in (4.7). Then the system (4.11) is solved to obtain numerical solutions. Time integration of the system is adapted by Runge-Kutta fourth order method. To establish the stability analysis, the nonlinear term $U(x_i) = \alpha_i$ is assumed to be constant. So, the resultant matrix form becomes,

$$\begin{bmatrix} \frac{dU(x_2, t)}{dt} \\ \frac{dU(x_3, t)}{dt} \\ \vdots \\ \frac{dU(x_{N-1}, t)}{dt} \end{bmatrix} = A \begin{bmatrix} U(x_2, t) \\ U(x_3, t) \\ \vdots \\ U(x_{N-1}, t) \end{bmatrix} + B \quad (4.12)$$

where coefficient matrix A is given as,

$$A = \begin{bmatrix} 2\alpha_2(w_{21}^{(1)})^2 + \alpha_2^2(w_{21}^{(2)}) & \cdots & 2\alpha_2(w_{2,N-1}^{(1)})^2 + \alpha_2^2(w_{2,N-1}^{(2)}) \\ \vdots & \ddots & \vdots \\ 2\alpha_{N-1}(w_{N-1,N-1}^{(1)})^2 + \alpha_{N-1}^2(w_{N-1,N-1}^{(2)}) & \cdots & 2\alpha_{N-1}(w_{N-1,N-1}^{(1)})^2 + \alpha_{N-1}^2(w_{N-1,N-1}^{(2)}) \end{bmatrix} \quad (4.13)$$

Now, we will investigate numerical solutions of problem (4.2) to validate our approach in terms of stability and comparisons with exact and previous studies.

Example 4.1.1: Homogeneous diffusion equation

Let us consider the problem (4.1) on $x \in [0, \pi]$ when $g(x, t) = 0$. Analytical solution of the problem is given in [78] as,

$$U(x, t) = e^{-t} \sin(x) \quad (4.14)$$

with the following conditions,

$$F_1 = F_2 = 0, \quad U(x, 0) = \sin(x). \quad (4.15)$$

Table 4.1 presents errors at different time levels for Example 4.1.1. Eigenvalues of the DQ discretized matrix are calculated for different number of grid points to see the stability of the problem. Figures 4.1-4.2 show the eigenvalue distribution for the Example 4.1.1 for various grid points. According to the graphs, eigenvalues of the problem consist of only real parts. It can be obtained that as the number of grid points increase, absolute value of the eigenvalues increase, so this requires using less number of time grids to keep the eigenvalues inside the region of stability. Maximum values of eigenvalues are given by Table 4.2.

Heat concentration graph is given by Figure 4.3 with projection onto xt - plane under the parameters $\Delta t = 0.001$ and $N = 120$. 2D graph of comparison between exact and numerical solutions is given by Figure 4.6. We can conclude that exact and present method graphs are well-matched.

Example 4.1.2: Inhomogeneous diffusion equation

Case (a): We studied the problem (4.1) when $g(x, t) = \cos x$ on $x \in [0, 1]$ under the following conditions,

$$U(0, t) = 1 - e^{-t}, \quad U(1, t) = \cos(1)(1 - e^{-t}), \quad U(x, 0) = 0. \quad (4.16)$$

Analytical solution of the problem is given in [72] as,

$$U(x, t) = \cos(x)(1 - e^{-t}). \quad (4.17)$$

Table 4.3 presents errors on $0 \leq t \leq 10$ for Example 4.1.2(a) using $\Delta t = 0.01$ and $N = 80$. Table 4.4 shows errors for present method and polynomial differential quadrature [72] for Example 4.1.2(a). Both methods give effective results. Heat concentration graph is given by Figure 4.4 with projection onto xt - plane under the parameters $\Delta t = 0.001$ and $N = 120$. 2D comparison graphs between exact and numerical solutions are given by Figure 4.7. We can conclude that exact and present method graphs are well-matched.

Case (b): We studied the problem (4.2) when $g(x, t) = \sin x$ on $x \in [0, \pi]$ under the following conditions,

$$U(0, t) = e^{-t}, \quad U(\pi, t) = -e^{-t}, \quad U(x, 0) = \cos(x). \quad (4.18)$$

Analytical solution of the problem is given in [78] as,

$$U(x, t) = (1 - e^{-t}) \sin(x) + e^{-t} \cos(x). \quad (4.19)$$

Table 4.5 presents errors on $0 \leq t \leq 10$ for Example 4.1.2(b) using $\Delta t = 0.01$ and $N = 80$. Both methods give effective results. Heat concentration graphs are given by Figure 4.5 with projection onto xt - plane under the parameters $\Delta t = 0.001$ and $N = 120$. 2D comparison graphs between exact and numerical solutions are given by Figure 4.8. We can conclude that exact and present method graphs are well-matched.

Example 4.1.3: Slow diffusion process

As a last example, we studied the nonlinear diffusion equation given in (4.2) on $x \in [0, 1]$ under the following conditions,

$$U(0, t) = \frac{a}{2\sqrt{c^2 - t}}, \quad U(1, t) = \frac{1+a}{2\sqrt{c^2 - t}}, \quad U(x, 0) = \frac{x+a}{2c}. \quad (4.20)$$

Analytical solution of the problem is [72],

$$U(x, t) = \frac{x+a}{2\sqrt{c^2 - t}} \quad t < c^2. \quad (4.21)$$

Table 4.6 shows errors for present method and polynomial differential quadrature [72] when $t = 0.1$ of Example 4.1.3. $U(x, t)$ solution graph for constant time steps is given

by Figure 4.9 using the parameters $\Delta t = 0.001$ and $N = 30$.

Table 4.1 Errors for different time levels of Example 4.1.1 for $N = 40$

t	L_2	L_∞
0.5	1.915119E-10	3.603346E-11
1	2.323157E-10	3.091319E-11
2	1.709284E-10	2.274465E-11
5	2.127506E-11	2.830974E-12
10	2.468127E-08	3.955037E-09

Table 4.2 Maximum eigenvalues for different number of grid points

N	20	40	80	120
$\Delta t \Re(\lambda_{max}) $	0.456601	1.661454	128.381152	659.694

Table 4.3 Errors for different time levels of Example 4.1.2 (a) for $N = 80$

t	L_2	L_∞
0.5	1.915119E-10	3.603346E-11
1	2.323157E-10	3.091319E-11
2	1.709284E-10	2.274465E-11
5	2.127506E-11	2.830974E-12
10	2.468127E-08	3.955037E-09

Table 4.4 Comparison of absolute errors of Example 4.1.2 (a) when $t = 0.1$

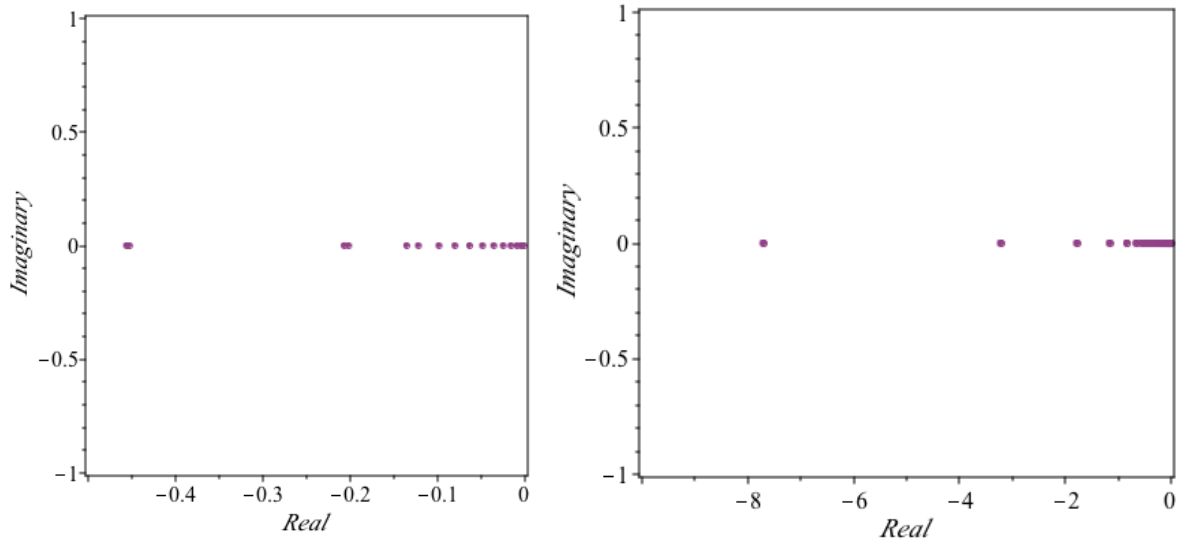
x	Present Method	PDQ [72]
x_2	1.98E-08	2.34E-13
x_5	9.82E-10	1.10E-13
x_8	1.91E-09	8.43E-14

Table 4.5 Errors for different time levels of Example 4.1.2 (b) for $N = 80$

t	L_2	L_∞
0.5	2.279323E-10	3.603346E-11
1	2.764959E-10	4.371079E-11
2	2.034343E-10	3.216060E-11
5	2.532100E-11	4.002956E-12
10	3.118000E-07	8.105380E-08

Table 4.6 Comparison of absolute errors of Example 4.1.3 when $t = 0.1$

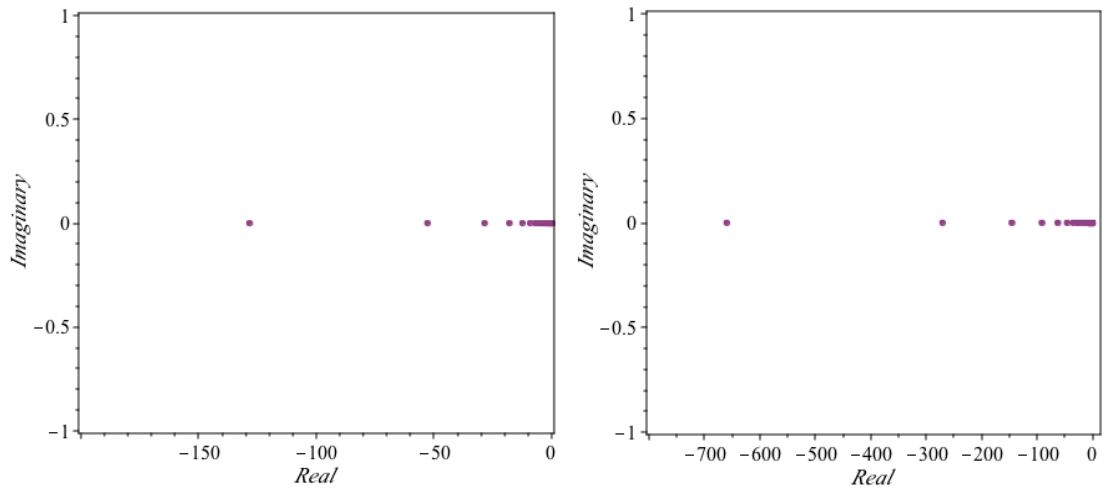
x	Present Method	PDQ [72]
x_2	3.81E-05	9.10E-15
x_5	4.25E-04	4.46E-14
x_8	9.89E-04	9.10E-15



(a)

(b)

Figure 4.1 Eigenvalue distribution when (a) $N = 20$, (b) $N = 40$



(c)

(d)

Figure 4.2 Eigenvalue distribution when (c) $N = 80$ and (d) $N = 120$

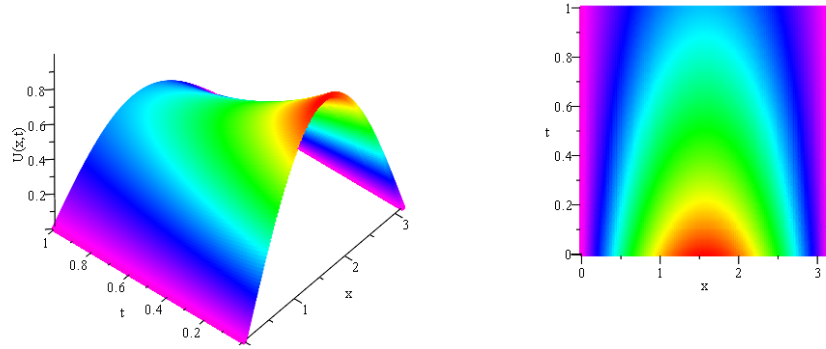


Figure 4.3 Numerical solutions and projection on $x t$ - plane of Example 4.1.1

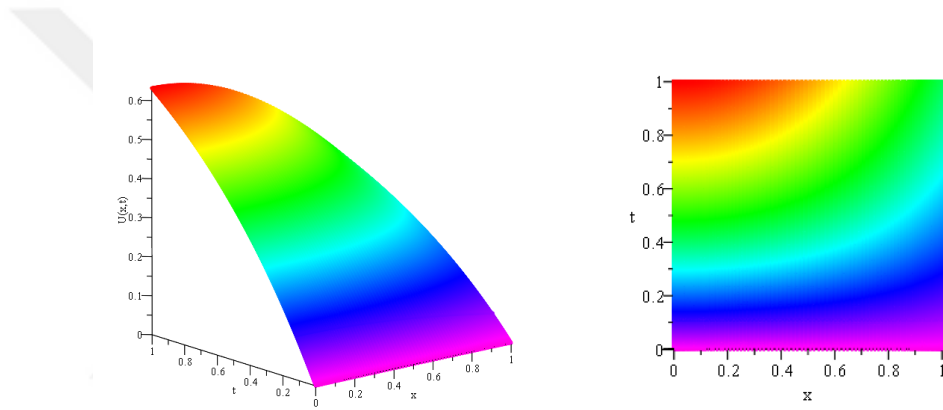


Figure 4.4 Numerical solutions and projection on $x t$ - plane of Example 4.1.2 (a)

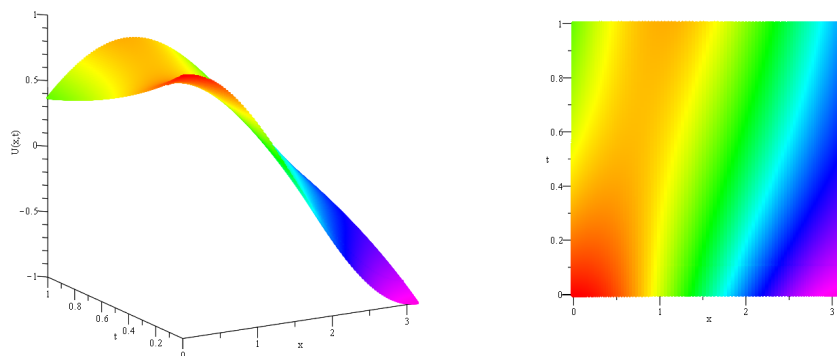


Figure 4.5 Numerical solutions and projection on $x t$ - plane of Example 4.1.2 (b)

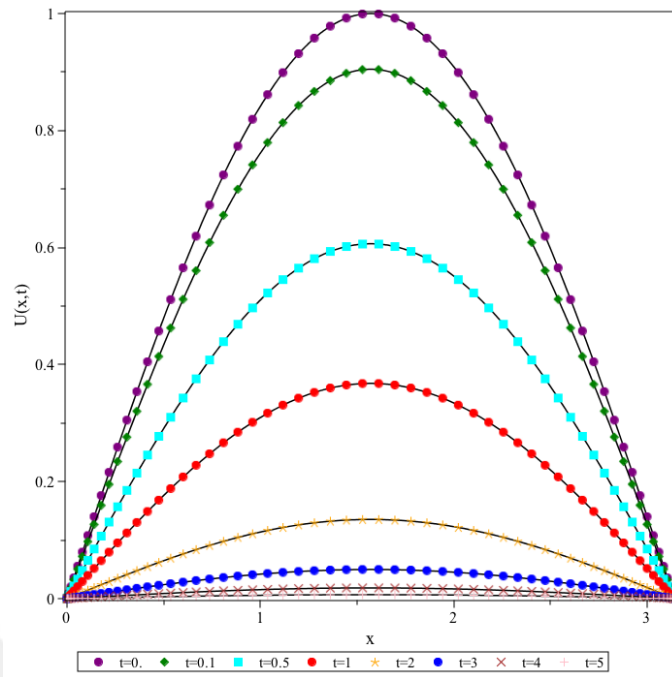


Figure 4.6 Comparison of numerical and exact solutions for constant time levels of Example 4.1.1

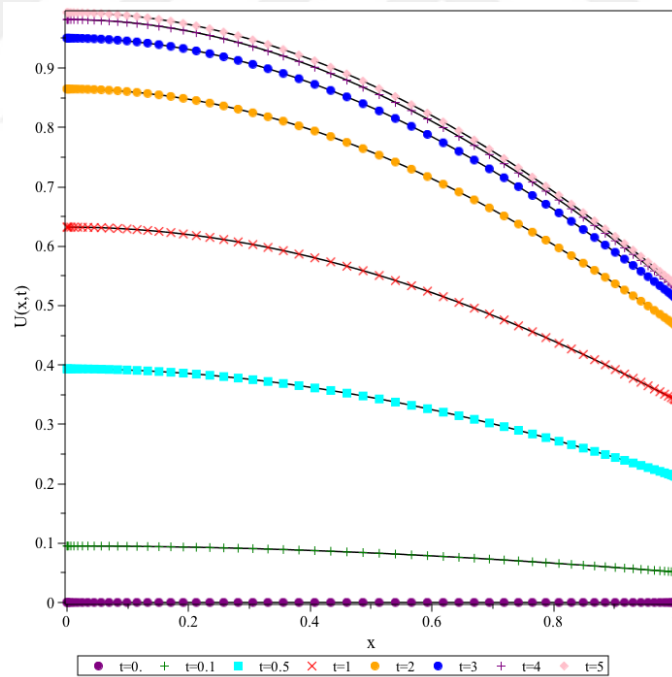


Figure 4.7 Comparison of numerical and exact solutions for constant time levels of Example 4.1.2 (a)

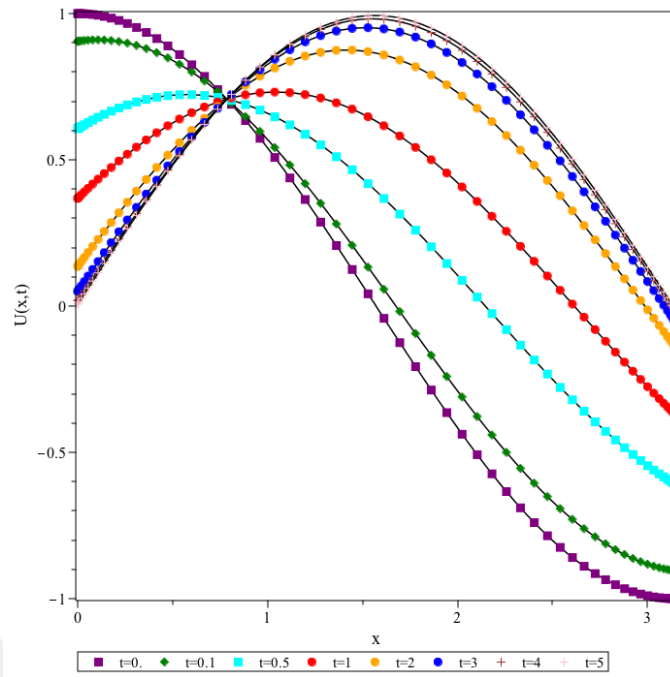


Figure 4.8 Comparison of numerical and exact solutions for constant time levels of Example 4.1.2 (b)

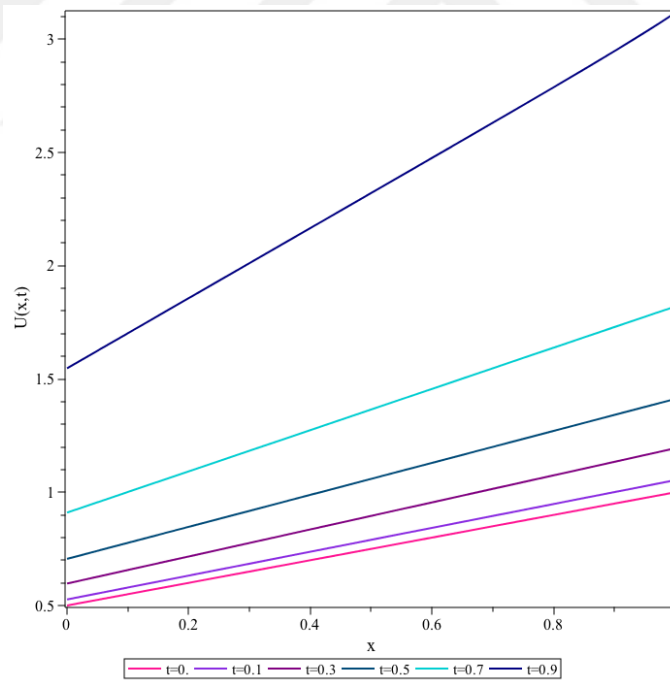


Figure 4.9 Numerical solutions for constant time levels of Example 4.1.3

4.2 Quadrature Solution of Reaction-Diffusion Equation in Non-Uniform Media

In this section, we have presented one-dimensional reaction-diffusion equation on composite media which is named by functionally graded material. The model describes the conductivity non-uniformly as a function of space. A functionally graded material is a non-uniform composite in which the volume fraction varies from one point to another. Composition profile of such materials enables to reduce singularities from transitions between two surfaces. When compared to traditional materials, FGMs improve the thermal properties of structure of the material since it presents gradual changing. FGM exists in many applications such as aerodynamics, thermal barrier cooling, high temperature problems due to its flexibility in material design [79, 80]. The solution process converts the model equation to an advection-reaction-diffusion equation.

Numerical methods for heat conduction equations in FGMs have been extensively studied. Sutradhar et al [81] have employed numerical approach using Laplace transform boundary element method for transient heat conduction problem in FGMs. Zhao et al [82] considered transient heat conduction in FGMs using method of fundamental solutions. Yu et al [83] have investigated heat conduction in FGMs by implementing boundary element and differential transform methods. Tian and Jiang [84] studied heat conduction in FGMs considering exponential heat source with hybrid numerical method based on weighted residuals.

Karagoz [85] investigated several reaction-diffusion equations in FGMs using an analytical method. Yigit et al [86] studied the vibration of functionally graded beams. Various studies on functionally graded beams based on DQM have been studied. Rajasekaran [87, 88] have presented vibration analysis of functionally graded beams using differential transform and differential quadrature methods. Vosoughi [89] studied thermal buckling analysis in FGMs numerically using DQM. Jiao et al [90] considered bending of functionally graded beams using differential quadrature.

The governed reaction-diffusion equation in FGMs is given in [85],

$$\rho(x)\mu(x)\frac{\partial U(x,t)}{\partial t} = \frac{\partial}{\partial x} \left[k(x)\frac{\partial U(x,t)}{\partial x} \right] + p(x,t)U(x,t) \quad (4.22)$$

where U is the concentration, p is the reaction parameter and $\rho(x)\mu(x)$ is the heat capacity and $k(x)$ represents conductivity with subject to proper initial and boundary conditions,

$$U(x,0) = F(x)$$

$$U(0, t) = f_0(t), \quad U_x(0, t) = f_1(t) \quad (4.23)$$

Now, due to the structure of the FGMs, heat capacity and conductivity parameters are assumed to be exponential functions given as,

$$\begin{aligned} \rho(x)\mu(x) &= \rho_0\mu_0 e^{\alpha x} \\ k(x) &= k_0 e^{\alpha x}, \quad \rho_0, \mu_0, k_0 \neq 0, \quad 0 \leq \alpha \leq 3. \end{aligned} \quad (4.24)$$

Using (4.24) into (4.22),

$$\rho_0\mu_0 e^{\alpha x} \frac{\partial U(x, t)}{\partial t} = k_0 \alpha e^{\alpha x} \frac{\partial U(x, t)}{\partial x} + k_0 e^{\alpha x} \frac{\partial^2 U(x, t)}{\partial x^2} + p(x, t)U(x, t) \quad (4.25)$$

or equivalently rewritten as,

$$\frac{\partial U(x, t)}{\partial t} = D \left\{ \frac{\partial^2 U(x, t)}{\partial x^2} + \alpha \frac{\partial U(x, t)}{\partial x} + q(x, t)U(x, t) \right\} \quad (4.26)$$

where

$$D = \frac{k_0}{\rho_0\mu_0}, \quad q(x, t) = \frac{e^{-\alpha x}}{k_0} p(x, t). \quad (4.27)$$

Equation (4.26) can be expressed as

$$U_t(x, t) = D \{ U_{xx}(x, t) + \alpha U_x(x, t) + q(x, t)U(x, t) \} \quad (4.28)$$

Exact solution of the problem is given in [85]. Quadrature discretization of (4.28) gives,

$$U_t(x_i, t) = D \{ U_{xx}(x_i, t) + \alpha U_x(x_i, t) + q(x_i, t)U(x_i, t) \} \quad (4.29)$$

Then, spatial derivatives are replaced by DQ formulations as,

$$\frac{\partial U(x_i, t)}{\partial t} = D \left\{ \sum_{j=1}^N w_{ij}^{(2)} U(x_j, t) + \alpha \sum_{j=1}^N w_{ij}^{(1)} U(x_j, t) + q(x_i, t)U(x_i, t) \right\} \quad (4.30)$$

DQM is applied to the boundary conditions as,

$$U(0, t) = U(x_1, t) = f_0, \quad U_x(0, t) = \sum_{j=1}^N w_{1j}^{(1)} U(x_j, t) = f_1 \quad (4.31)$$

Equation (4.30) can be rewritten as,

$$w_{11}^{(1)} U(x_1, t) = f_1 - \sum_{j=2}^N w_{1j}^{(1)} U(x_j, t) \quad (4.32)$$

or

$$U(x_1, t) = \frac{f_1 - \sum_{j=2}^N w_{1j}^{(1)} U(x_j, t)}{w_{11}^{(1)}} \quad (4.33)$$

So, discretized system given by (4.30) becomes,

$$\frac{\partial U(x_i, t)}{\partial t} = D \left\{ \sum_{j=2}^N w_{ij}^{(2)} U(x_j, t) + \alpha \sum_{j=2}^N w_{ij}^{(1)} U(x_j, t) + q(x_i, t) U(x_i, t) + B \right\} \quad (4.34)$$

where

$$B = U(x_1, t) = w_{11}^{(1)} f_0 - f_1 + \sum_{j=2}^N w_{1j}^{(1)} U(x_j, t) \quad (4.35)$$

The system can be expressed as,

$$\{T\} = [A]\{U\} + \{B\} \quad (4.36)$$

where the vectors T and B represent time derivative and vector containing boundary conditions respectively. For time integration of the system, we use Runge-Kutta fourth order method. Stability analysis depends on the eigenvalue distribution of the numerical discretized matrix [A]. Boundary conditions and non-homogeneous part of the problem have no effect on stability. We will consider eigenvalues of the following matrix [A] by considering different number of grid points. So, the resultant matrix form becomes as,

$$\begin{bmatrix} \frac{dU(x_2, t)}{dt} \\ \frac{dU(x_3, t)}{dt} \\ \vdots \\ \frac{dU(x_N, t)}{dt} \end{bmatrix} = A \begin{bmatrix} U(x_2, t) \\ U(x_3, t) \\ \vdots \\ U(x_N, t) \end{bmatrix} + B \quad (4.37)$$

Now, we will investigate numerical solutions some test problems to validate our approach.

Example 4.2.1 Advection-reaction-diffusion equation is considered when $D = 1$ and $q(x, t) = -1$ with following initial and boundary conditions

$$U(x, 0) = e^{-x} + x, \quad (4.38)$$

$$U(0, t) = \alpha t e^{-t} + e^{-\alpha t}, \quad U_x(0, t) = e^{-t} - e^{\alpha t}$$

Exact solution of the problem is given in [85] using Adomian decomposition

method. We investigate numerical solutions of the problem for different levels of non-homogeneity parameter together with different time levels. In Table 4.7 we compare results with homogeneous problem as the non-homogeneity parameter is sufficiently small like $\alpha \rightarrow 0$. Errors on $0 \leq t \leq 3$ are presented by Table 4.8. Graphs of numerical solution are given by Figures 4.12-4.14 using parameters $\Delta t = 0.001$, $N = 80$, $0 \leq t \leq 1$, for different non-homogeneity levels such as $\alpha = 0.001$, $\alpha = 1$ and $\alpha = 3$. In Figure 4.15, graphs of solutions are depicted for fixed time levels. In all cases, results are well matched with exact solutions of the problem. We can conclude that, as non-homogeneity parameter is away from zero, solutions are less stable.

The approximations express accurate solutions when using $N = 120$, increasing grid points also causes costs in terms of process time, so we did not consider large grid numbers since Runge-Kutta method requires matrix vector multiplication for each time step.

To check the stability of the presented method we used eigenvalues of grid points when $N = 20$, $N = 40$, $N = 80$, $N = 120$ for $\Delta t = 0.001$. Graphs of real and imaginary parts of the eigenvalues are given by Figures 4.10-4.11. We can conclude that, as grid points increase, eigenvalues also increase, so time steps should be decreased while grid points increase. Detailed values of eigenvalues are given in Table 4.9.

Table 4.7 Maximum absolute errors for non-homogeneity parameters when $t = 1$ and $N = 30$

$\alpha = 0.001$	$\alpha = 1$	$\alpha = 3$
1.036825E-10	2.907920E-09	1.572733E-06

Table 4.8 Error Norms for different time levels when $\Delta t = 0.01$, $\alpha = 1$ and $N = 30$

t	L_2	L_∞
0.1	1.567752E-10	2.971670E-11
0.5	1.496585E-10	8.940508E-11
1	5.165439E-10	9.892987E-11
2	1.433503E-06	4.949908E-07
3	5.531280E-04	1.794557E-04

Table 4.9 Maximum eigenvalues of the discretized system for different number of grid points

N	20	30	40	80	120
$\Delta t Re(\lambda_{max}) $	0.036855	0.371330	1.661454	21.769829	227.817311
$\Delta t Im(\lambda_{max}) $	0.039955	0.319993	1.308291	17.085405	157.275180

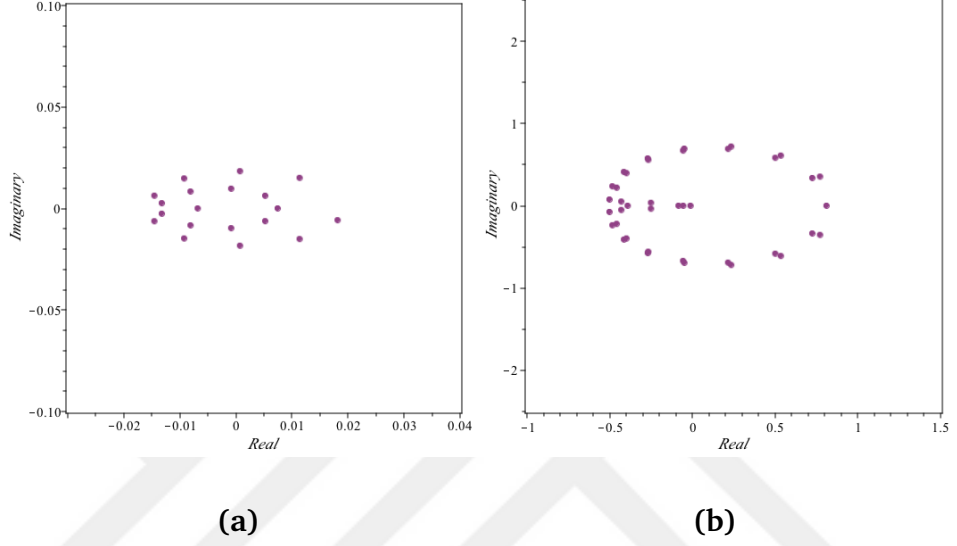


Figure 4.10 Eigenvalue distribution when (a) $N = 20$, (b) $N = 40$

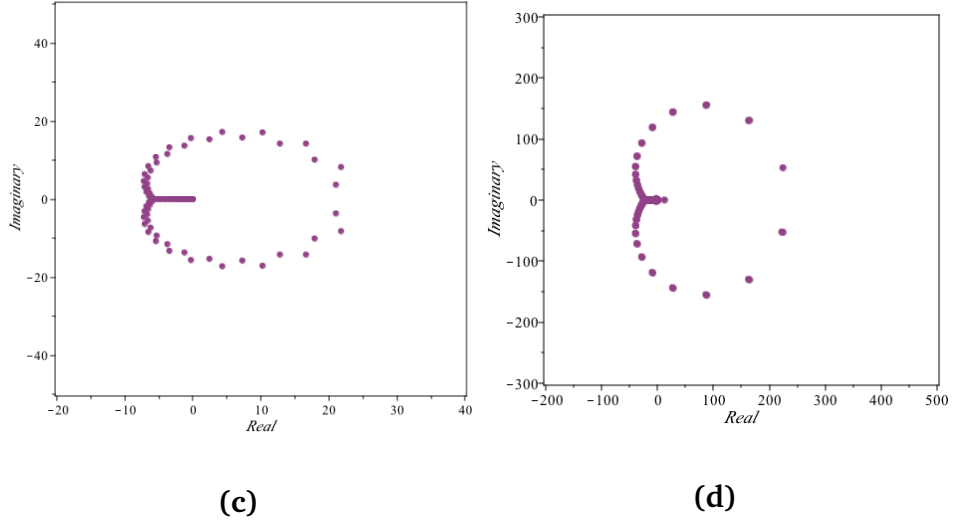


Figure 4.11 Eigenvalue distribution when (c) $N = 80$, (d) $N = 120$

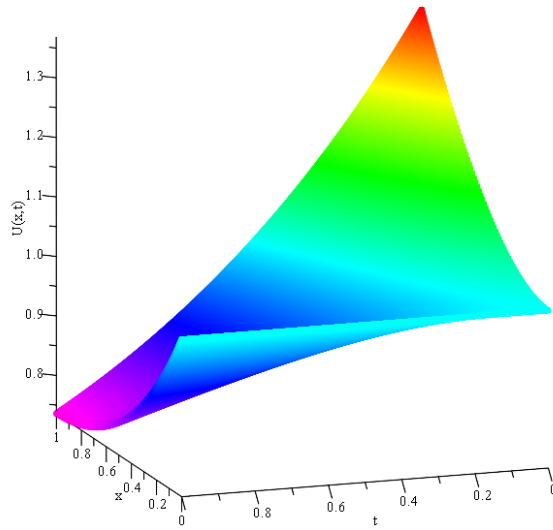


Figure 4.12 Numerical solution of Example 4.2.1 for $\alpha = 0.001$

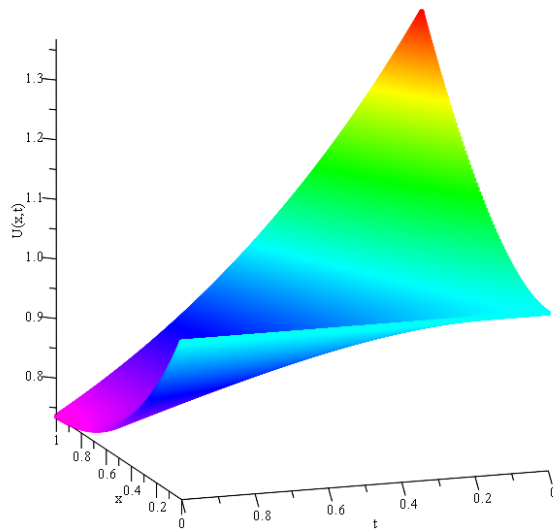


Figure 4.13 Numerical solution of Example 4.2.1 for $\alpha = 1$

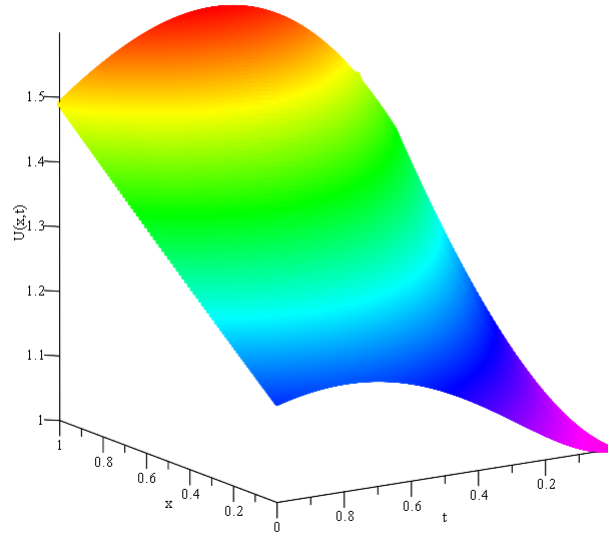


Figure 4.14 Numerical solution of Example 4.2.1 for $\alpha = 3$

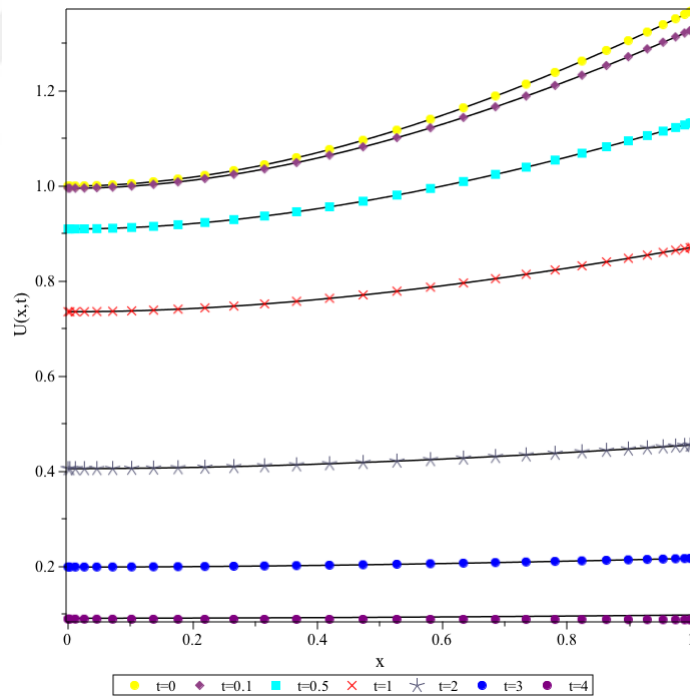


Figure 4.15 Comparison between numerical and exact solutions for different time levels of Example 4.2.1 for $\Delta t = 0.01$ and $N = 30$

Example 4.2.2 We consider following advection-reaction-diffusion equation,

$$U_t(x, t) = D \{U_{xx}(x, t) - \alpha U_x(x, t) + q(x, t)U(x, t)\} \quad (4.39)$$

when $D = 1$ and reaction parameter $q(x, t) = 2t$ for following two cases:

Case (a) $\alpha = 1$ with subject to following initial and boundary conditions:

$$\begin{aligned} U(x, 0) &= e^{-x}, \\ U(0, t) &= e^{t^2} \quad U_x(0, t) = -e^{t^2}. \end{aligned} \quad (4.40)$$

Exact solution of the problem (4.39) is given as $U(x, t) = e^{-x+t^2}$ in [85].

Case (b) $\alpha = 0.001$ with subject to following initial and boundary conditions:

$$\begin{aligned} U(x, 0) &= e^x, \\ U(0, t) &= e^{t+t^2} \quad U_x(0, t) = e^{t+t^2}. \end{aligned} \quad (4.41)$$

Exact solution of the problem (4.39) is given as $U(x, t) = e^{x+t+t^2}$ when $\alpha = 0$ in [91] and an approximate exact solution is given in [85] when non-homogeneity parameter is sufficiently small such as $\alpha = 0.001$.

We investigate numerical solutions of the problems for different levels of non-homogeneity parameter together with different time levels. On $0 \leq t \leq 1$ errors are presented by Table 4.10 and 4.11. In Figures 4.18-4.19, graphs of solutions are depicted for fixed time levels. In all cases, results are well matched with exact solutions. Graphs of numerical solutions are given by Figures 4.20-4.21 using parameters $\Delta t = 0.001$, $N = 80$, $0 \leq t \leq 1$.

Stability of the problem is analyzed using eigenvalues when grid points are $N = 20$, $N = 40$, $N = 80$, $N = 120$ for $\Delta t = 0.001$. Regions of eigenvalues are presented in Figures 4.16-4.17. We conclude that, as grid points increase, absolute value of eigenvalues also increase, so time steps should be decreased.

Table 4.10 Error Norms for constant time levels when $\Delta t = 0.001$, and $N = 40$ for Example 4.2.2 (a)

t	L_2	L_∞
0.1	2.174044E-04	5.451645E-05
0.25	5.612583E-04	1.523352E-04
0.5	2.397659E-03	6.893236E-04
0.75	5.804446E-03	1.487297E-03
1	9.594511E-03	2.349227E-03

Table 4.11 Error Norms for constant time levels when $\Delta t = 0.001$, and $N = 40$ for Example 4.2.2 (b)

t	L_2	L_∞
0.1	1.809551E-14	4.180954E-15
0.25	1.159057E-13	2.677794E-14
0.5	1.088085E-12	2.514013E-13
0.75	7.367476E-12	1.702249E-12
1	4.385473E-11	1.013260E-11

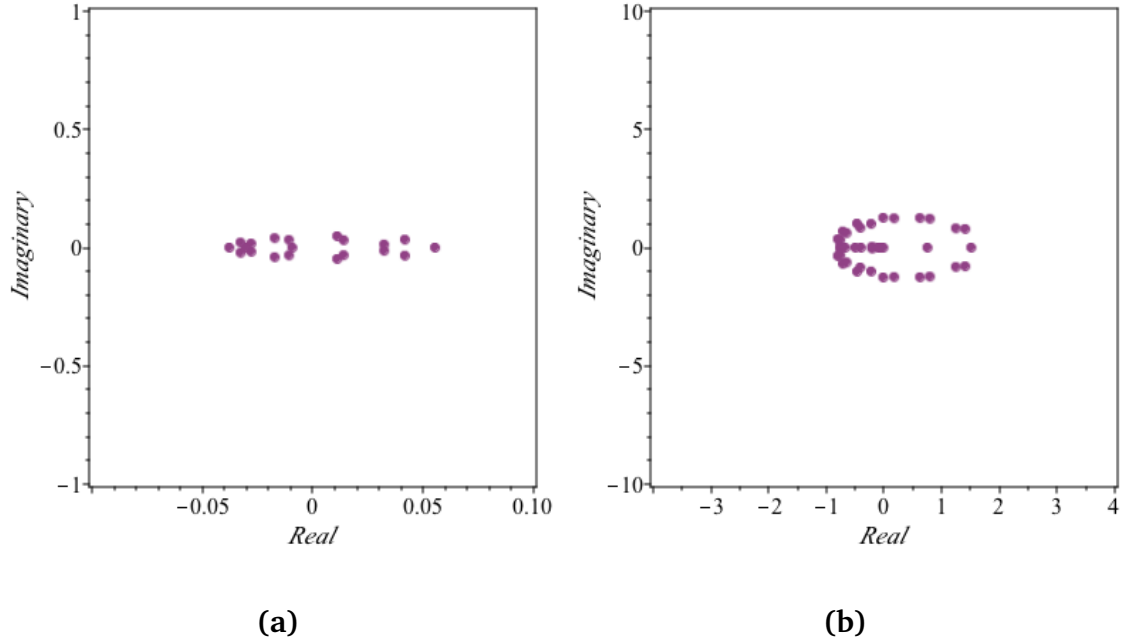


Figure 4.16 Eigenvalue distribution when (a) $N = 20$, (b) $N = 40$

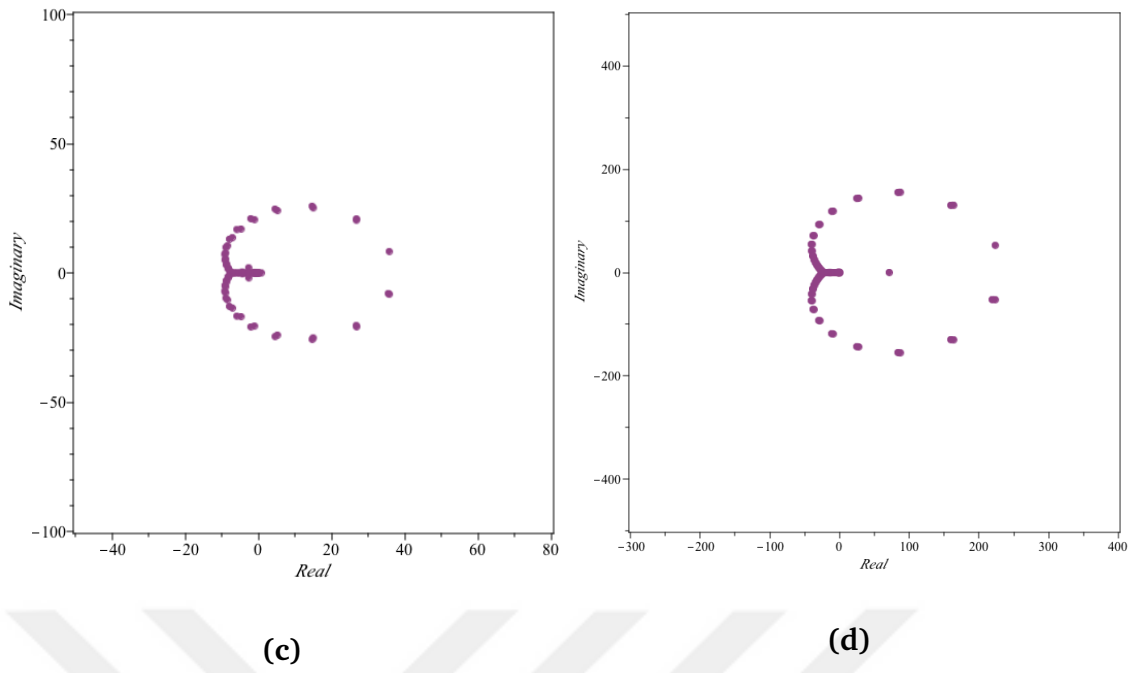


Figure 4.17 Eigenvalue distribution when (c) $N = 80$, (d) $N = 120$

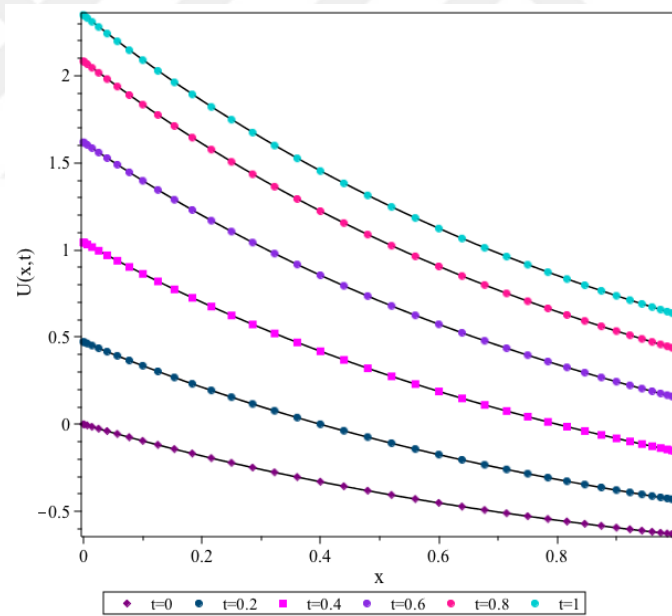


Figure 4.18 Numerical solutions for different time levels of Example 4.2.2 (a) for $\Delta t = 0.001$ and $N = 40$

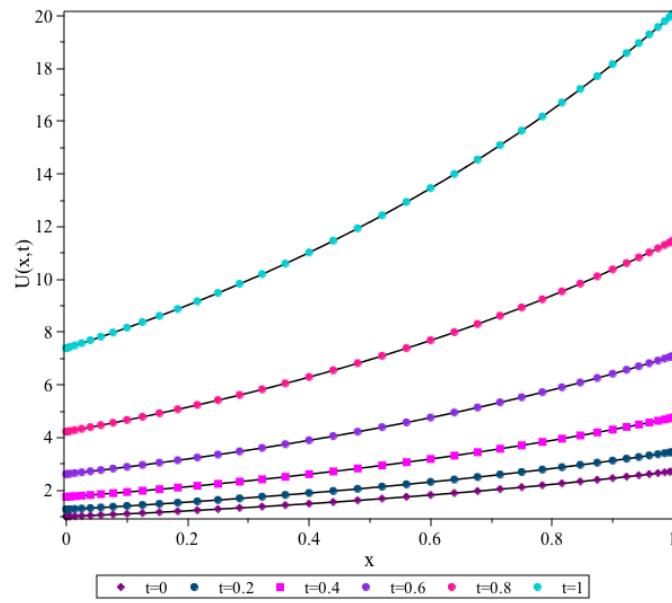


Figure 4.19 Numerical solutions for different time levels of Example 4.2.2 (b) for $\Delta t = 0.001$ and $N = 40$

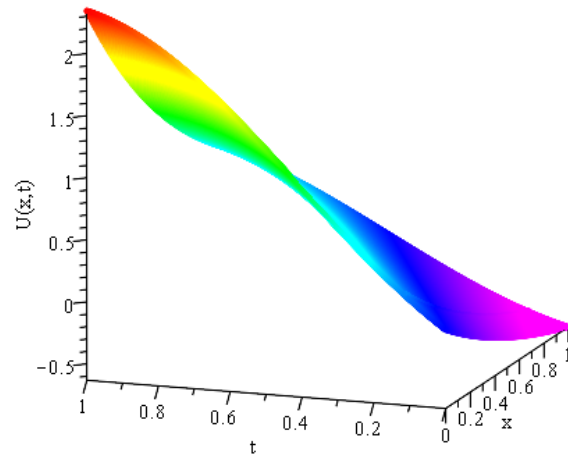


Figure 4.20 Graph of numerical solutions for Example 4.2.2 (a)

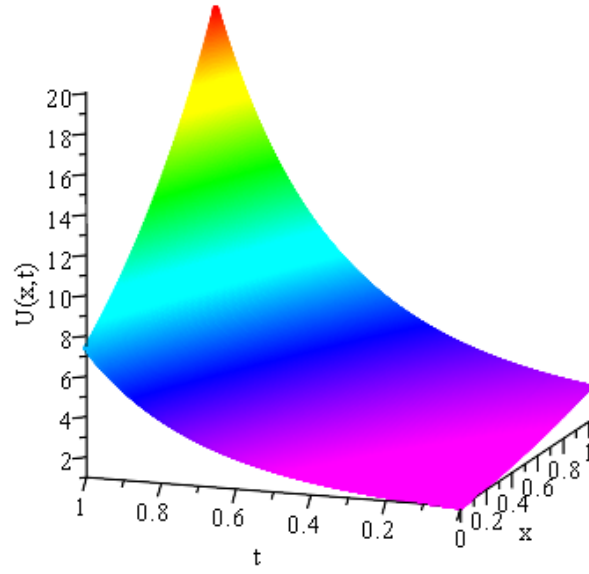


Figure 4.21 Graph of numerical solutions for Example 4.2.2 (b)

4.3 Quadrature Solution of Kuramoto-Sivashinsky Equation

The Kuramoto-Sivashinsky (KS) equation is presented as a model of phase turbulence in reaction-diffusion systems [92, 93], plasma instabilities and flame front propagation [94]. The model equation is a widely studied topic both analytically and numerically. A few of the methods considered here consist of Chebyshev spectral scheme [95], quintic B-splines [96], exponential cubic B-splines [97]. Solutions of the KS equation are analyzed by many methods such as the finite difference methods [98], discontinuous Galerkin method [99], numeric meshless method for space derivatives using radial basis function [100], He's variational iteration method [101]. Rademacher and Wattenberg studied on viscous shocks for the model equation [102]. Cerpa et al presented control results of the equation [103, 104]. Lai and Ma [105] solved the model numerically using Lattice Boltzmann technique. Mittal and Dahiya [106] studied the same problem using B-spline based differential quadrature. Lately, Hepson [107], studied model equation by a numerical algorithm based on trigonometric cubic B-splines. Yigit and Bayram [108] studied the equation using DQM with Euler time discretization.

The Kuramoto-Sivashinsky (KS) equation is a nonlinear partial differential equation given by,

$$\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} + \alpha \frac{\partial^2 U}{\partial x^2} + \nu \frac{\partial^4 U}{\partial x^4} = 0, \quad x \in [x_0, x_N], \quad t \in (0, T), \quad (4.42)$$

along with the initial and boundary conditions

$$\begin{aligned}
U(x, 0) &= U_0, \\
U(x_0, t) &= g_0, \quad U(x_N, t) = g_1, \\
U_x(x_0, t) &= 0, \quad U_x(x_N, t) = 0, \\
U_{xx}(x_0, t) &= 0, \quad U_{xx}(x_N, t) = 0,
\end{aligned} \tag{4.43}$$

where α represents growth of the linear stability and ν denotes surface tension. Note that for $\nu = 0$, the term surface tension is removed, then the model equation becomes Burgers' equation [96].

The KS equation is rewritten as

$$U_t = -(UU_x + \alpha U_{xx} + \nu U_{xxxx}), \tag{4.44}$$

$$U_t(x_i, t) = -(U(x_i, t)U_x(x_i, t) + \alpha U_{xx}(x_i, t) + \nu U_{xxxx}(x_i, t)). \tag{4.45}$$

Then, spatial derivatives are replaced by the Differential Quadrature equality,

$$\frac{\partial U(x_i, t)}{\partial t} = -\left(U(x_i, t) \sum_{j=1}^N w_{ij}^{(1)} U(x_j, t) + \alpha \sum_{j=1}^N w_{ij}^{(2)} U(x_j, t) + \nu \sum_{j=1}^N w_{ij}^{(4)} U(x_j, t) \right), \tag{4.46}$$

Applying boundary conditions to (4.46) gives,

$$\begin{aligned}
\frac{\partial U(x_i, t)}{\partial t} &= -\left\{ U(x_i, t) \sum_{j=2}^{N-1} w_{ij}^{(1)} U(x_j, t) + \alpha \sum_{j=2}^{N-1} w_{ij}^{(2)} U(x_j, t) + \right. \\
&\quad \left. \nu \sum_{j=2}^{N-1} w_{ij}^{(4)} U(x_j, t) \right\} + B(U),
\end{aligned} \tag{4.47}$$

where

$$\begin{aligned}
B(U) &= -\{ U(x_i, t) [w_{i1}^{(1)} g_0(t) + w_{iN}^{(1)} g_1(t)] + \alpha [w_{i1}^{(2)} g_0(t) + w_{iN}^{(2)} g_1(t)] \\
&\quad + \nu [w_{i1}^{(4)} g_0(t) + w_{iN}^{(4)} g_1(t)] \}.
\end{aligned} \tag{4.48}$$

The system can be expressed in the form of a matrix equation as,

$$\{T\} = [A]\{U\} + \{B\} \tag{4.49}$$

where the vectors T and B represents time derivative and boundary conditions

vector respectively and coefficient matrix A consists of first, second and fourth order derivatives. Time integration of the system is adapted by Runge-Kutta fourth order method. Stability analysis has been established depends on the eigenvalue distribution of the numerical discretized matrix $[A]$ with the assumption of $U(x_i) = \kappa_i$, (κ_i constant). We consider eigenvalues of the following matrix $[A]$ by considering different number of grid points. So, the resultant matrix form becomes,

$$\begin{bmatrix} \frac{dU(x_2, t)}{dt} \\ \frac{dU(x_3, t)}{dt} \\ \vdots \\ \frac{dU(x_{N-1}, t)}{dt} \end{bmatrix} = A \begin{bmatrix} U(x_2, t) \\ U(x_3, t) \\ \vdots \\ U(x_{N-1}, t) \end{bmatrix} + B \quad (4.50)$$

We studied four test examples and reported numerical solutions via tables and figures.

Example 4.3.1

Case (a) KS equation (4.13) is studied when $\alpha = 1$ and $\nu = 1$ and analytical solution is given by [96],

$$u(x, t) = b + \frac{15}{19} \sqrt{\frac{11}{19}} [-9 \tanh(k(x - bt - x_0)) + 11 \tanh^3(k(x - bt - x_0))]. \quad (4.51)$$

The initial condition is evaluated using analytical solution. Domain of the problem is considered as $[-30, 30]$. Comparisons between the exact and numerical solutions are given by Table 4.12 for constant time levels. We use the parameters $b = 5$, $k = \frac{1}{2} \sqrt{\frac{11}{19}}$, $x_0 = -12$. To check the stability of the presented method we used eigenvalues of grid points when $N = 20$, $N = 40$, $N = 80$, $N = 120$ for $\Delta t = 0.001$. Graphs of real and imaginary parts of the eigenvalues are given by Figures 4.22-4.23. We conclude that, as grid points increase, eigenvalues also increase, so time steps should be decreased as grid points increase.

Case (b) The problem is solved for $\alpha = -1$ and $\nu = 1$. Exact solution of the problem is given by [96],

$$u(x, t) = b + \frac{15}{19} \sqrt{\frac{1}{19}} [-3 \tanh(k(x - bt - x_0)) + 11 \tanh^3(k(x - bt - x_0))]. \quad (4.52)$$

The initial and boundary conditions are evaluated using analytical solution. Domain of the problem is considered as $[-50, 50]$. We use the parameters $b = 5$, $k = \frac{1}{2\sqrt{19}}$, $x_0 =$

–25. Errors for different time levels are presented by Table 4.13 using $\Delta t = 0.001$. The solution models the shock wave propagation with speed b and initial position x_0 . Numerical and exact solutions graphs are given by Figure 4.24.

Example 4.3.2

The KS equation is considered when $\alpha = 1$ and different values of parameter ν with following initial and boundary conditions [96],

$$U(x, 0) = -\sin(\pi x), -1 \leq x \leq 1 \quad (4.53)$$

$$U(x_0, t) = U(x_N, t) = 0 \quad (4.54)$$

Number of space grid is taken as $N = 80$ and $\Delta t = 0.001$. Graph of the numerical solutions are given by Figures 4.25-4.26 at constant time levels choosing $\nu = 0.4/\pi^2$ and $\nu = 0.8/\pi^2$.

Example 4.3.3

Case (a): The KS equation is solved under the parameters $\alpha = 1$ and $\nu = 1$ which shows turbulent flow character over a finite space domain [96], with initial condition in the form of a Gaussian curve as,

$$U(x, 0) = e^{-x^2}, \quad (4.55)$$

and boundary conditions

$$U(x_0, t) = U(x_N, t) = 0. \quad (4.56)$$

Domain is considered as $[x_0, x_N] = [-30, 30]$ and $\Delta t = 0.005$ with $N = 100$ nodal points. The solutions of KS equation is given by Figures 4.27-4.28 when $t = 5$ and $t = 10$ respectively. The chaotic nature of the KS equation is simulated by Figure 4.29.

Case (b): The KS equation is considered when $\alpha = 1$ and different ν parameters on the interval $[0, 4\pi]$ with following initial condition [97],

$$U(x, 0) = \cos\left(\frac{x}{2}\right) \sin\left(\frac{x}{2}\right) \quad (4.57)$$

and boundary conditions

$$U_{xx}(0, t) = U_{xx}(4\pi, t) = 0. \quad (4.58)$$

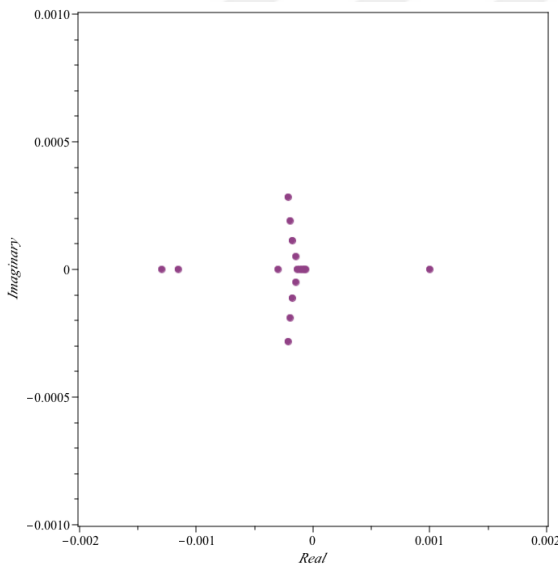
Number of space grid is taken as $N = 120$ and $\Delta t = 0.0025$. Chaotic behavior is presented by Figure 4.30. In both cases simulations provide the expected behaviors.

Table 4.12 Comparison of Global Relative Errors of Example 4.3.1 (a) for Different Methods

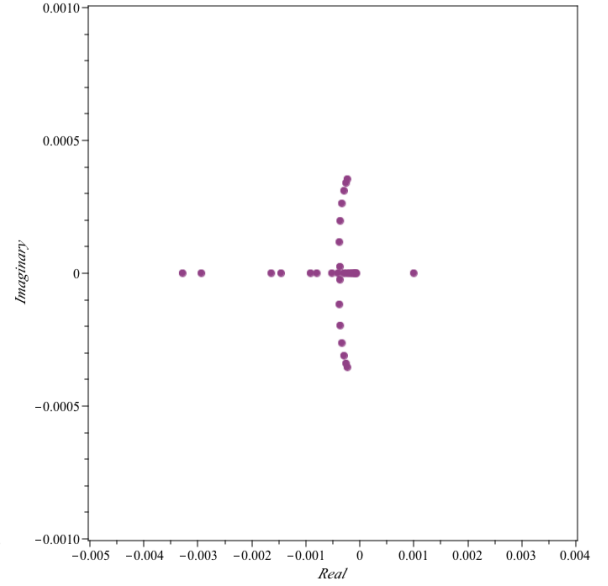
t	Present Method	[97]	[96]	[105]
1	1.292422E-03	8.74634E-04	3.81725E-04	6.7923E-04
2	5.454200E-02	1.30146E-03	5.51142E-03	1.1503E-03

Table 4.13 Errors for different time levels of Example 4.3.1 (b) for $N = 30$

t	L_2	L_∞
0.1	1.049152E-09	2.992243E-10
0.25	6.885439E-09	1.963763E-09
0.5	7.195535E-07	2.051961E-08
0.75	5.936704E-07	1.693200E-07
1	4.551143E-06	1.297370E-06



(a)



(b)

Figure 4.22 Eigenvalue distribution when (a) $N = 20$, (b) $N = 40$

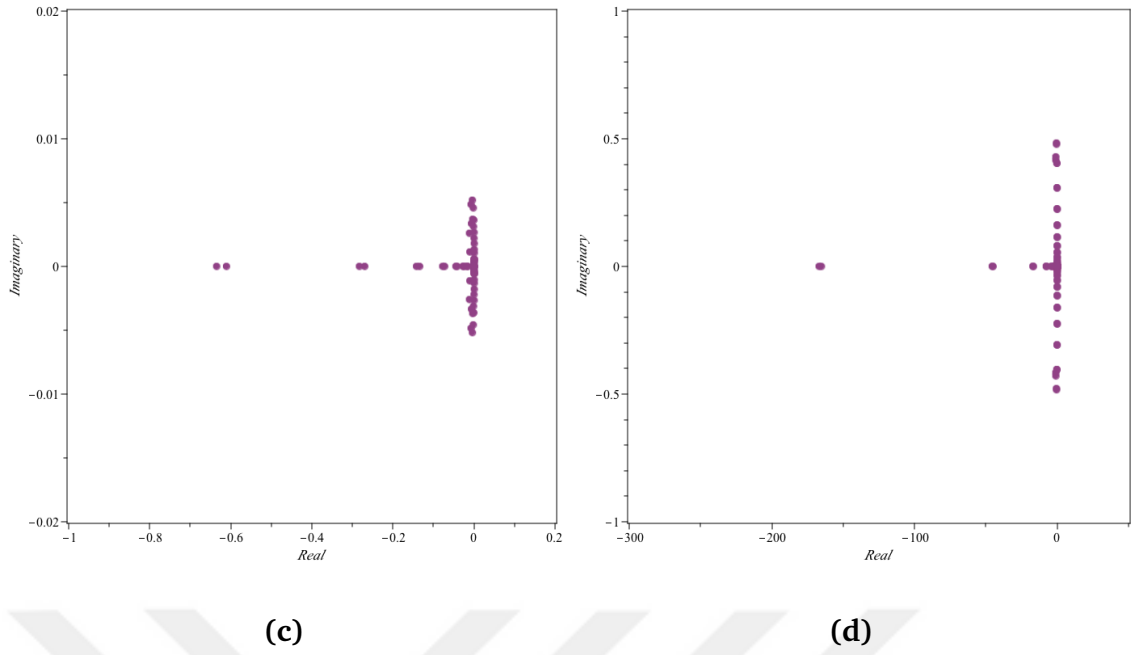


Figure 4.23 Eigenvalue distribution when (c) $N = 80$, (d) $N = 120$

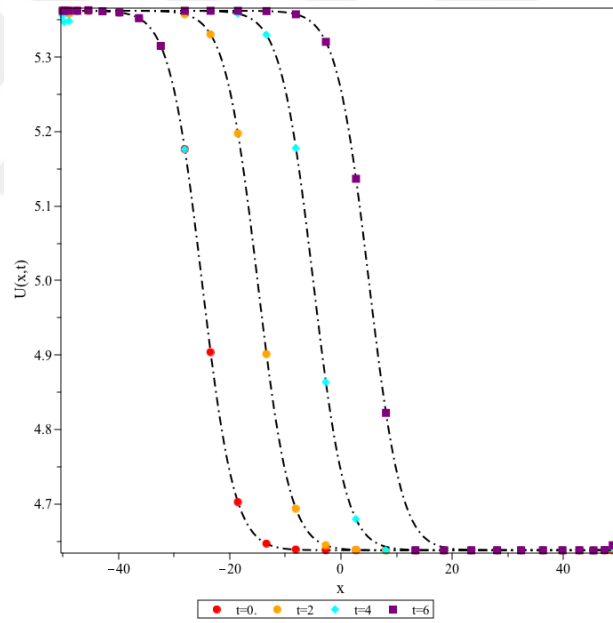


Figure 4.24 Solutions of Example 4.3.1 (b) for different values of constant time

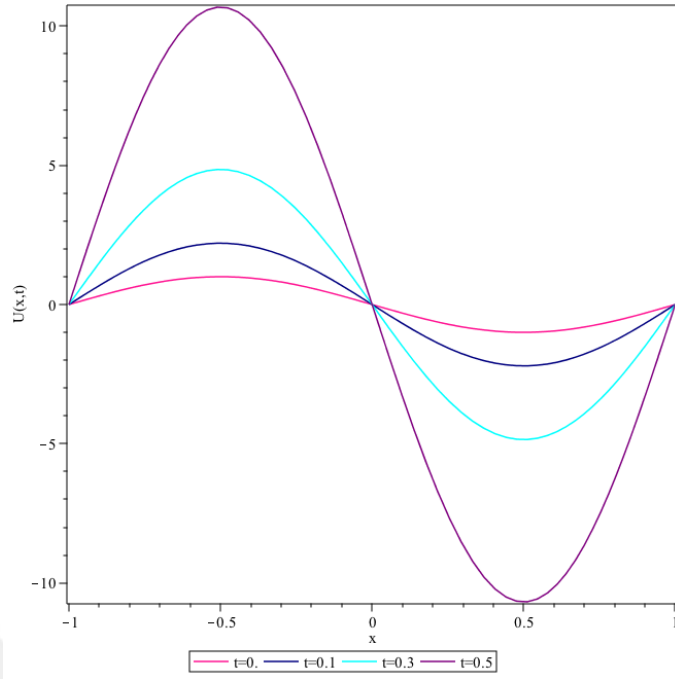


Figure 4.25 Numerical solutions for different time levels of Example 4.3.2 when $\nu = 0.4/\pi^2$

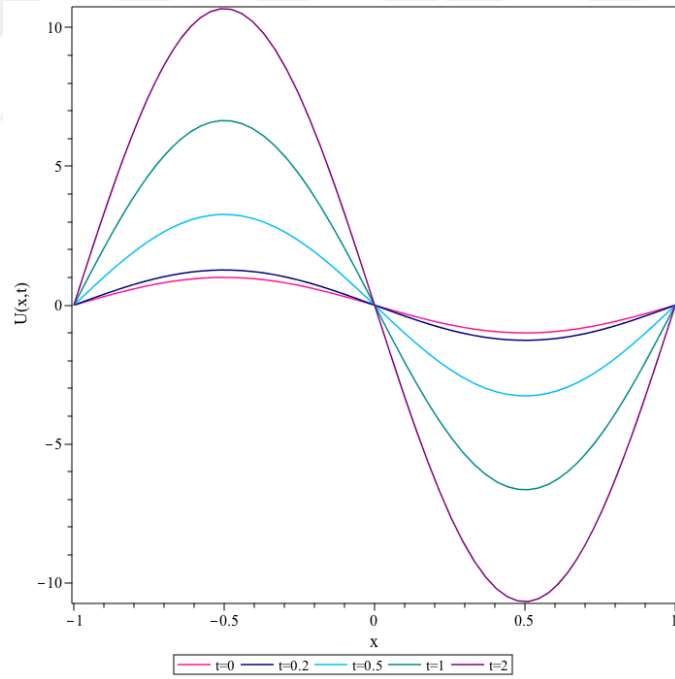


Figure 4.26 Numerical solutions for different time levels of Example 4.3.2 when $\nu = 0.8/\pi^2$

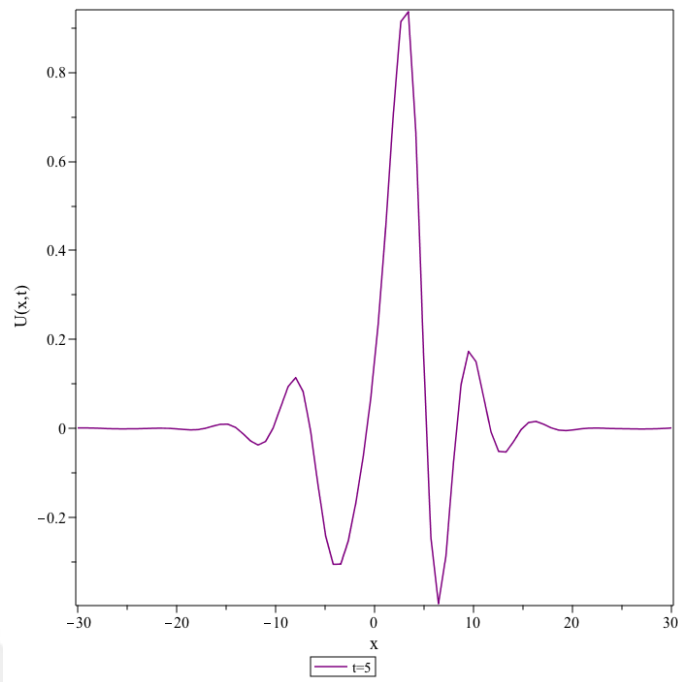


Figure 4.27 The chaotic simulation of the KS equation with Gaussian initial condition at $t = 5$

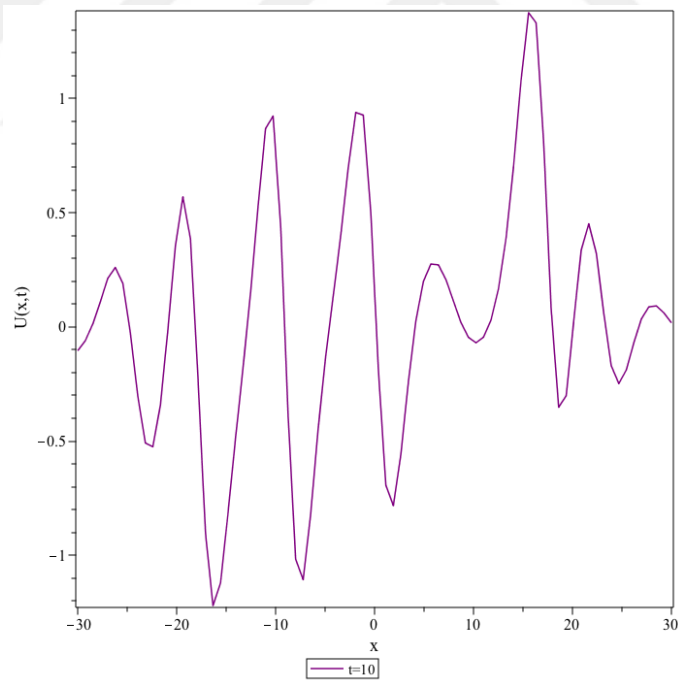


Figure 4.28 The chaotic simulation of the KS equation with Gaussian initial condition at $t = 10$

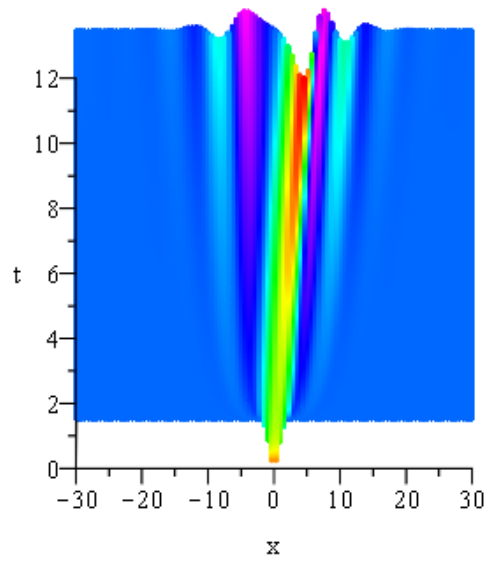


Figure 4.29 The chaotic solution of the KS equation for Example 4.3.3 (a)

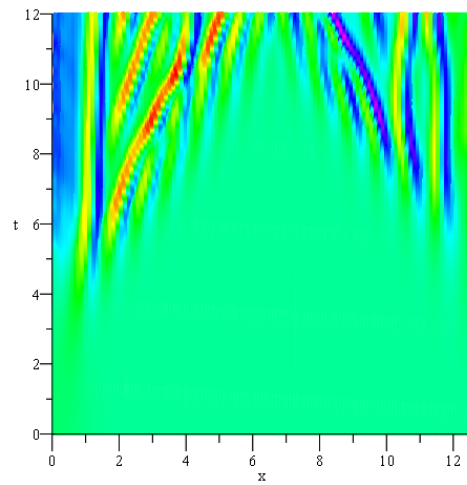


Figure 4.30 The chaotic solution of the KS equation for Example 4.3.3 (b)

4.4 Quadrature Solution of Nonlinear Advection Equation

In this section, we have presented first-order, quasilinear, partial differential equations which describe conservation laws. First we deal with the one dimensional conservative form given in [109],

$$U_t + \frac{1}{2}(U^2)_x = f(x, t), \quad U(x, 0) = g(x). \quad (4.59)$$

The source term f generally describes growth term in biological problems or reaction term in chemical processes. Equation (4.59) can also be represented as,

$$U_t + \frac{1}{2}2UU_x = f(x, t), \quad (4.60)$$

$$U_t + q(U)U_x = f(x, t), \quad (4.61)$$

where $q(U) = U$ resembles the wave speed. If $f(x, t) = 0$, then (4.61) represents the kinematic wave equation and (4.61) is also called as inviscid Burgers' equation which is defined under the absence of diffusion and viscosity. The inviscid Burger's equation expresses wave propagation which arises in traffic flow on highways as well as various chemical and physical processes [109].

We consider DQ discretization of following equation,

$$U_t = -UU_x + f(x, t) \quad (4.62)$$

Using all grid points, (4.62) is rewritten as,

$$U_t(x_i, t) = -U(x_i, t)U_x(x_i, t) + f(x_i, t). \quad (4.63)$$

The spatial derivatives are replaced by the Differential Quadrature equality,

$$\frac{\partial U(x_i, t)}{\partial t} = -U(x_i, t) \sum_{j=1}^N w_{ij}^{(1)} U(x_j, t) + f(x_i, t). \quad (4.64)$$

The system can be expressed as,

$$\{T\} = [A]\{U\} + \{B\} \quad (4.65)$$

where the vector T represents time derivative and vector B consists of inhomogeneous part of the problem. Coefficient matrix A includes first order derivative. (4.64) should be applied at all grid points including boundaries. Time integration of the system is

adapted by Runge-Kutta fourth order method. Stability analysis has been established depends on the eigenvalue distribution of the discretized matrix $[A]$. We consider eigenvalues of the matrix $[A]$ by noting different number of grid points. So, the resultant matrix takes the form,

$$\begin{bmatrix} \frac{dU(x_1, t)}{dt} \\ \frac{dU(x_2, t)}{dt} \\ \vdots \\ \frac{dU(x_N, t)}{dt} \end{bmatrix} = A \begin{bmatrix} U(x_1, t) \\ U(x_2, t) \\ \vdots \\ U(x_N, t) \end{bmatrix} + B \quad (4.66)$$

We will investigate problem to validate our approach in terms of stability, convergence and comparison with exact solutions studied in previous studies. We analyzed homogeneous and inhomogeneous test problems and reported numerical solutions via tables and figures.

Example 4.4.1 Consider the numerical solution for the inhomogeneous nonlinear advection equation [78], namely,

$$U_t + \frac{1}{2}(U^2)_x = e^x + t^2 e^{2x}, \quad 0 < x < 1, \quad t > 0, \quad (4.67)$$

with the initial condition $U(x, 0) = 0$. Exact solution of the problem is,

$$U(x, t) = te^x. \quad (4.68)$$

Domain of the problem is considered as $[0, 1]$. Errors for different number of partitions are given by Table 4.14 using $\Delta t = 0.01$ at fixed time $t = 0.5$. Comparisons between the exact and numerical solutions are given by Table 4.16 for various time levels. Eigenvalues of the discretized system are depicted in Figures 4.31-4.32 choosing grid points as $N = 20, N = 40, N = 80, N = 120$ with $\Delta t = 0.001$. The given system is consistent in terms of stability. 2D comparison graphs between the exact and the numerical solutions are given by Figure 4.33. We can observe that the numerical solutions of the method are in good agreement with the exact solutions for case where the exact solution is known. Also, a 3D numerical solutions graph is given by Figures 4.35 using parameters $\Delta t = 0.001, N = 80, 0 \leq t \leq 1$.

Example 4.4.2 Consider the approximated solution for inhomogeneous nonlinear advection equation [78]:

$$U_t + \frac{1}{2}(U^2)_x = x, \quad 0 < x < 1, \quad t > 0, \quad (4.69)$$

with the initial condition $U(x, 0) = 2$ for which the exact solution has the form

$$U(x, t) = 2\text{sech } t + x \tanh t. \quad (4.70)$$

Spatial domain of the problem is considered $[0,1]$. Errors for different number of partitions are given by Table 4.15 using $\Delta t = 0.01$ at fixed time $t = 0.5$. Comparisons between the exact and the numerical solutions are presented in Table 4.17 for various time levels. 2D comparison graphs between the exact and the numerical solutions are given by Figure 4.34. We can conclude that exact and present method graphs are well-matched. Also, a 3D numerical solutions graph is given by Figure 4.36 using parameters $\Delta t = 0.001$, $N = 80$, $0 \leq t \leq 1$. We can observe that the numerical solutions of the method are in good agreement with the exact solutions for case where the exact solution is known.

Example 4.4.3 Consider the quadrature solution for the inviscid Burgers' equation which represents undamped wave [110]:

$$U_t + \frac{1}{2}(U^2)_x = 0, \quad 0 < x < 1, \quad t > 0. \quad (4.71)$$

Case (a): With the spatially periodic initial condition of the form

$$U(x, 0) = \sin(2\pi x) \quad (4.72)$$

which is posed on a bounded spatial domain $[0,1]$. We run the algorithm using $N = 80$ and $\Delta t = 0.001$. Numerical solutions graph is given by Figure 4.37 for various time levels. Graph obtained by differential quadrature shows similar behavior with literature [110] and provides the expected behaviors.

Case (b): With the initial condition in the form of a Gaussian curve as [111],

$$U(x, 0) = e^{(-2(x-1)^2)}. \quad (4.73)$$

Spatial domain of the problem is considered as $[-1,3]$. We run the algorithm using $N = 100$ and $\Delta t = 0.001$. 2D numerical solutions graphs are given by Figures 4.38. In both cases higher values of t results in the oscillations on the boundaries.

Table 4.14 Numerical solutions of Example 4.4.1 at $t = 0.5$

N	L_2	L_∞
20	2.040395E-04	9.607361E-05
40	3.446770E-04	9.701222E-05
80	4.756709E-04	9.723473E-05

Table 4.15 Numerical solutions of Example 4.4.2 at $t = 0.5$

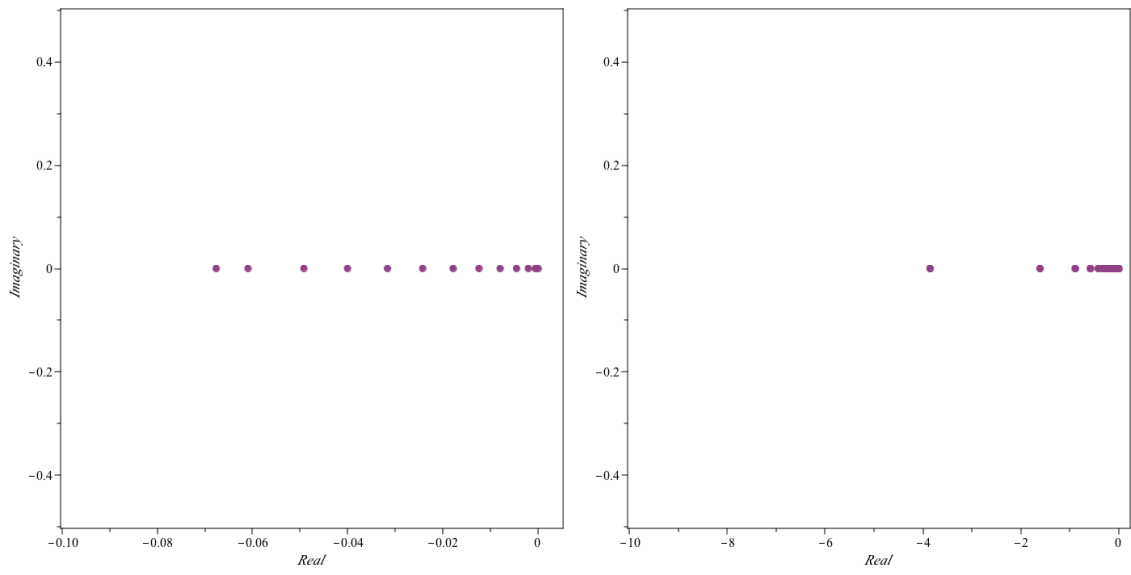
N	L_2	L_∞
20	1.286815E-04	3.322958E-05
40	1.819590E-04	3.327868E-05
80	2.573072E-04	3.329026E-05

Table 4.16 Error Norms for different time levels when $\Delta t = 0.001$ for Example 4.4.1

t	L_2	L_∞
0.1	3.761004E-06	3.415257E-05
0.25	4.309549E-05	2.380287E-04
0.5	1.763390E-04	9.195425E-04
0.75	3.040028E-04	1.528923E-04
1	5.530058E-04	1.924994E-04

Table 4.17 Error Norms for different time levels when $\Delta t = 0.001$, for Example 4.4.2

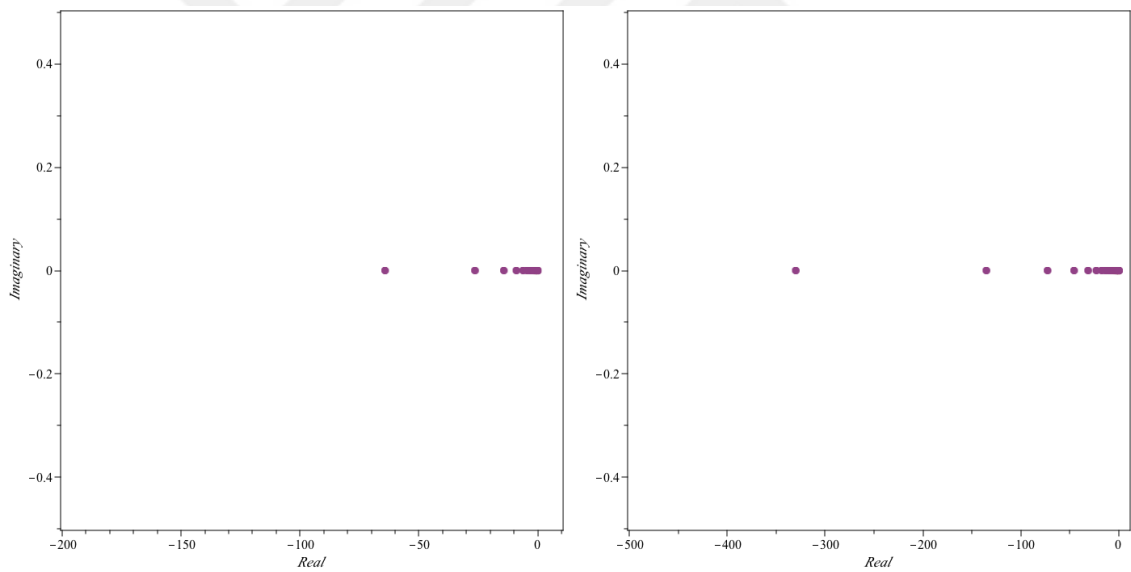
t	L_2	L_∞
0.1	3.406445E-05	7.861966E-06
0.25	8.736547E-05	1.968006E-05
0.5	1.286861E-04	3.322958E-05
0.75	1.233299E-04	3.460554E-05
1	8.185081E-05	2.605578E-05



(a)

(b)

Figure 4.31 Eigenvalue distribution when (a) $N = 20$, (b) $N = 40$



(c)

(d)

Figure 4.32 Eigenvalue distribution when (c) $N = 80$, (d) $N = 120$

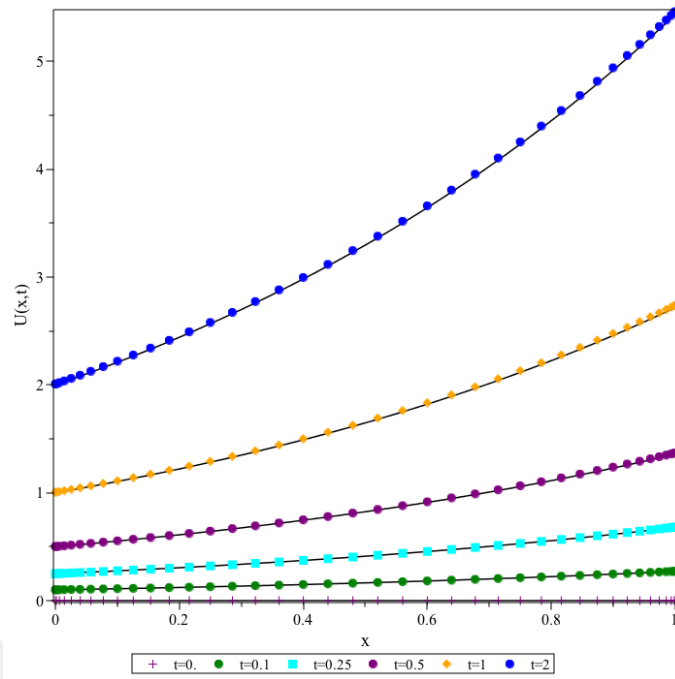


Figure 4.33 Comparison between numerical and exact solutions of Example 4.4.1

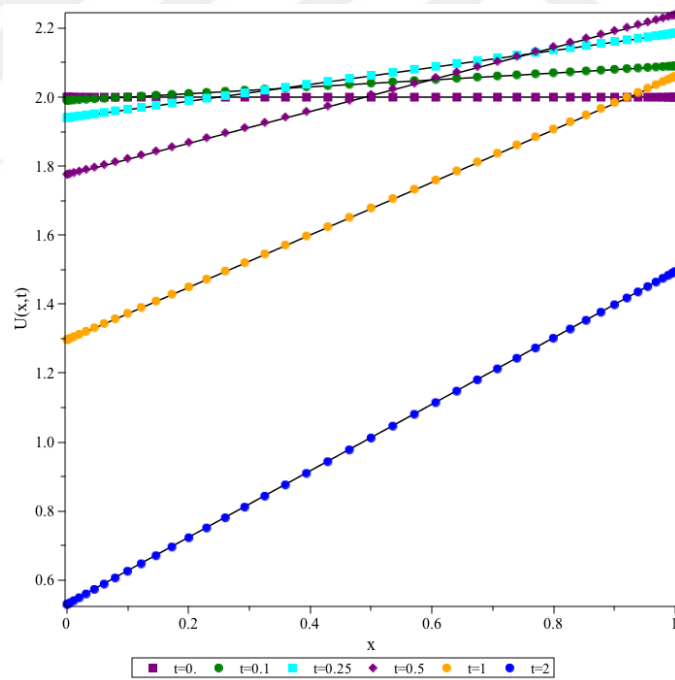


Figure 4.34 Comparison between numerical and exact solutions of Example 4.4.2

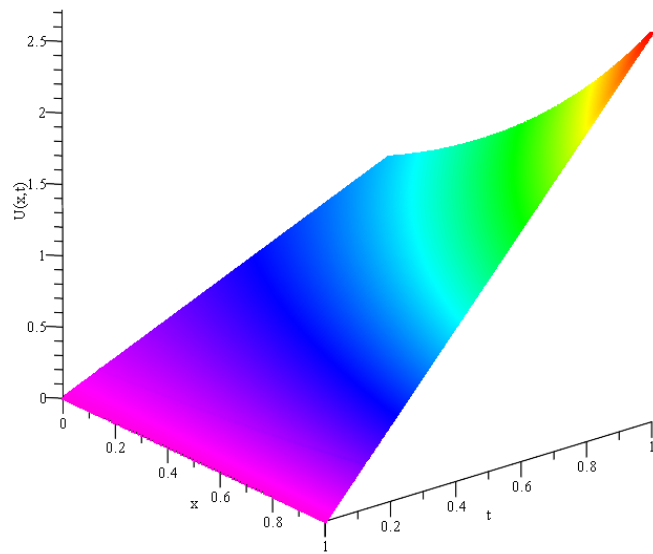


Figure 4.35 Numerical solutions of Example 4.4.1

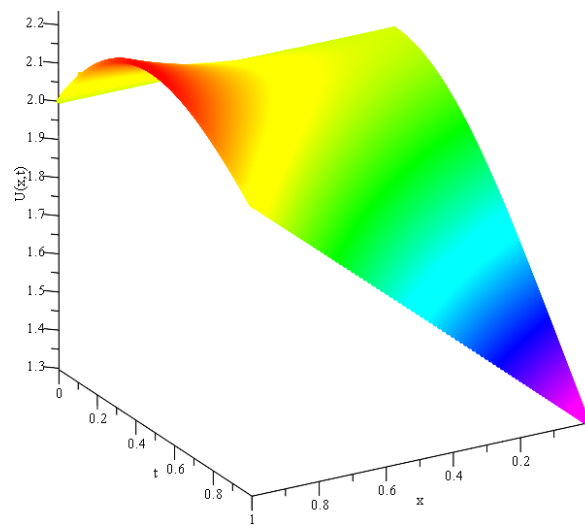


Figure 4.36 Numerical solutions of Example 4.4.2

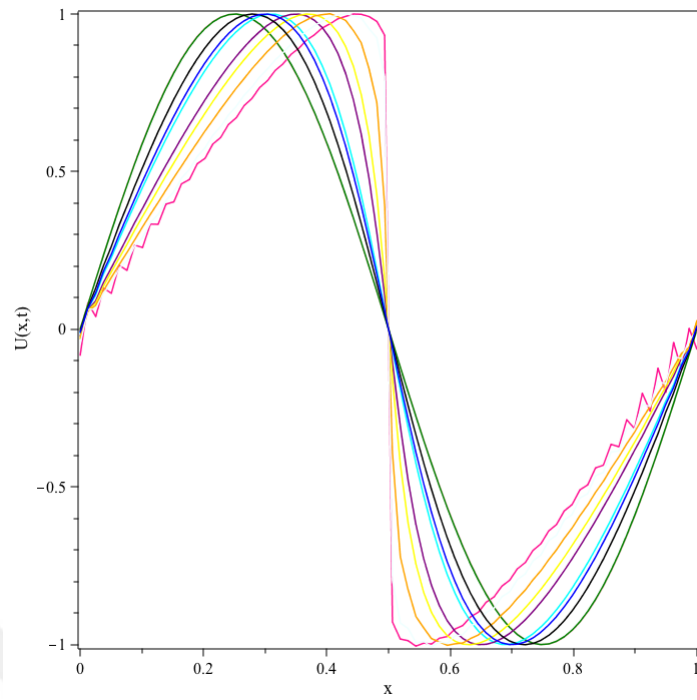


Figure 4.37 Numerical simulation of Example 4.4.3 (a) at various time steps with sinusoid initial condition

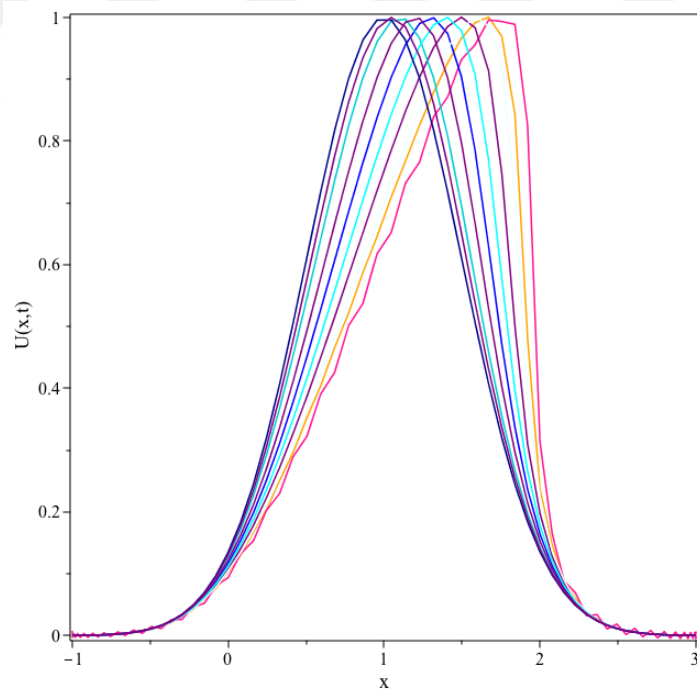


Figure 4.38 Numerical simulation of Example 4.4.3 (b) at various time steps with Gaussian initial condition

4.5 Quadrature Solution of Ginzburg Landau Equation

In this section, we present the numerical solutions of Ginzburg-Landau equation which arises in a wide variety of physical systems [112]. The complex form Ginzburg-Landau equation is given as,

$$A_t = A + (1 + i\alpha)A_{xx} - (1 + i\beta)|A|^2A \quad (4.74)$$

where $A(x, t)$ represents the complex amplitude and real parameters α and β characterize linear and nonlinear dispersion [112]. Complex Ginzburg-Landau equation presents unstable waves and models as superconductivity and phase transitions [109]. As limit of $\alpha, \beta \rightarrow \infty$, (4.74) is reduced to nonlinear Schrödinger equation [113].

When the imaginary coefficients are given as $\alpha = \beta = 0$, the real Ginzburg Landau equation is obtained as,

$$A_t = A + A_{xx} - |A|^2A. \quad (4.75)$$

Various numerical and analytical approaches for the solutions of Ginzburg-Landau equation have been presented such as Chebyshev wavelet collocation [114], finite element method [115], decomposition method [116], homotopy perturbation method [117].

Here, we will investigate quadrature solutions of real Ginzburg Landau equation. Equation (4.75) can be rewritten using all grid points,

$$U_t(x_i, t) = U(x_i, t) + U_{xx}(x_i, t) - |U(x_i, t)|^2 U(x_i, t) \quad (4.76)$$

The spatial derivatives are replaced by the Differential Quadrature equality,

$$\frac{\partial U(x_i, t)}{\partial t} = U(x_i, t) + \sum_{j=1}^N w_{ij}^{(2)} U(x_j, t) - |U(x_i, t)|^2 U(x_i, t) \quad (4.77)$$

The system can be expressed as,

$$\{T\} = \{U\} + [A]\{U\} - |\{U\}|^2 \{U\} + \{B\} \quad (4.78)$$

where the vector T represents time derivative and vector B consists of inhomogeneous part of the problem. Coefficient matrix A consists of second order derivative. Equation (4.77) should be applied to all grid points including boundaries. Time integration of the system is adapted by Runge-Kutta fourth order method.

Example 4.5 Consider the approximated solution for inhomogeneous real Ginzburg-Landau equation, which has the form [116],

$$U_t - U + |U|^2 U - U_{xx} = x(1 - t - x^2 t^3), \quad (4.79)$$

with the following boundary and initial conditions,

$$U(0, t) = 0, \quad U(1, t) = t, \quad (4.80)$$

$$U(x, 0) = 0. \quad (4.81)$$

Exact solution of the problem is

$$U(x, t) = xt. \quad (4.82)$$

Spatial domain of the problem is considered as $[0,1]$. Comparisons between the different methods are given by Table 4.18 for various time levels. Eigenvalues of the discretized system are depicted in Figures 4.39-4.40 choosing grid points as $N = 20$, $N = 40$, $N = 80$, $N = 120$ for $\Delta t = 0.001$. As the space splits get close to time steps absolute value of the eigenvalues increase. So, to keep the eigenvalues within the region of stability, we should use less time steps as space nodes N increase. 2D numerical and exact solutions comparison graph is given by Figure 4.43. Exact solution graph is given by Figure 4.41 Also, a 3D numerical solution graph is given by Figure 4.42 using parameters $\Delta t = 0.001$, $N = 80$, $0 \leq t \leq 1$. We can conclude that exact and present method graphs are well-matched.

Table 4.18 Comparison of methods of Example 4.5 at various time levels

t	x_i	DQM	[114]
$t = 0.083$	0.083	3.736412E-07	6.817922E-12
	0.250	9.840062E-07	9.038495E-12
	0.417	1.778463E-06	4.323513E-12
	0.583	2.548683E-06	1.892028E-12
	0.750	3.270093E-06	9.219082E-12
	0.917	3.903754E-06	9.219082E-13
$t = 0.250$	0.083	1.226255E-06	3.161798E-11
	0.250	3.469128E-06	3.681351E-11
	0.417	5.836739E-06	1.459357E-11
	0.583	8.364498E-06	3.071293E-12
	0.750	1.073205E-05	3.224615E-12
	0.917	1.281159E-05	1.535493E-12
$t = 0.417$	0.083	2.002439E-06	1.003995E-10
	0.250	6.467399E-06	1.138260E-10
	0.417	9.336086E-06	4.364175E-11
	0.583	1.449377E-05	8.796990E-12
	0.750	1.940182E-05	9.392875E-12
	0.917	2.327454E-05	4.502509E-12
$t = 0.583$	0.083	3.062774E-06	1.840555E-10
	0.250	9.892020E-06	1.919165E-10
	0.417	1.582664E-05	6.379077E-11
	0.583	2.216845E-05	5.228983E-12
	0.750	2.967528E-05	4.305118E-12
	0.917	3.559884E-05	4.502509E-12
$t = 0.750$	0.083	4.822075E-06	4.337897E-10
	0.250	1.364181E-05	4.650852E-10
	0.417	2.295197E-05	1.650103E-10
	0.583	3.289172E-05	2.804240E-11
	0.750	4.226125E-05	2.957167E-11
	0.917	4.970079E-05	1.410593E-11
$t = 0.917$	0.083	5.811366E-06	1.078974E-09
	0.250	1.876927E-05	1.172872E-09
	0.417	3.002965E-05	4.266622E-10
	0.583	4.206250E-05	8.190992E-11
	0.750	5.630564E-05	8.685707E-11
	0.917	6.754455E-05	4.152266E-11

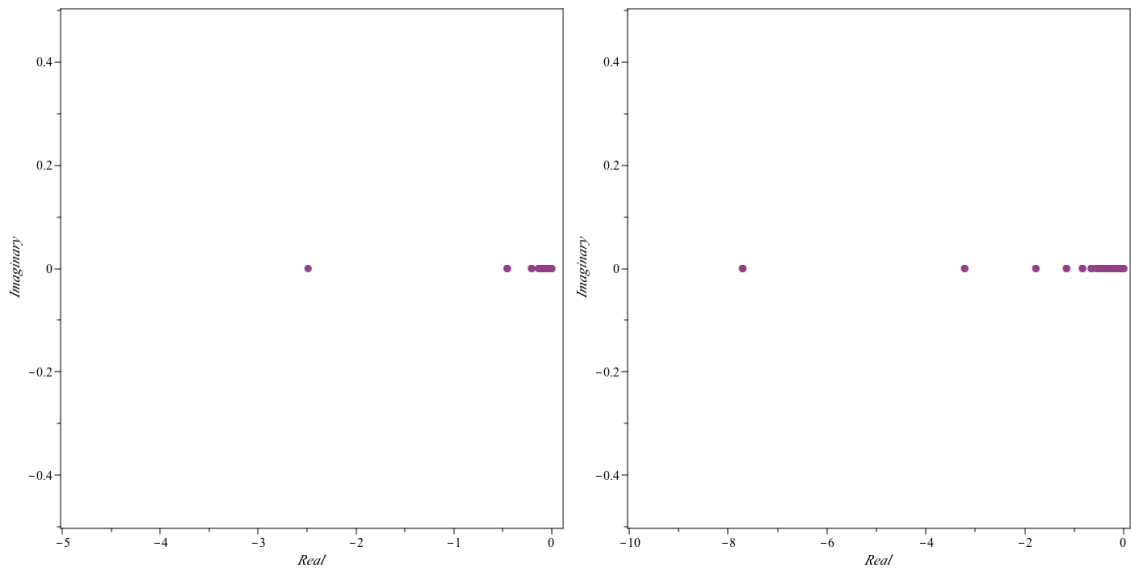


Figure 4.39 Eigenvalue distribution for (a) $N = 20$, (b) $N = 40$

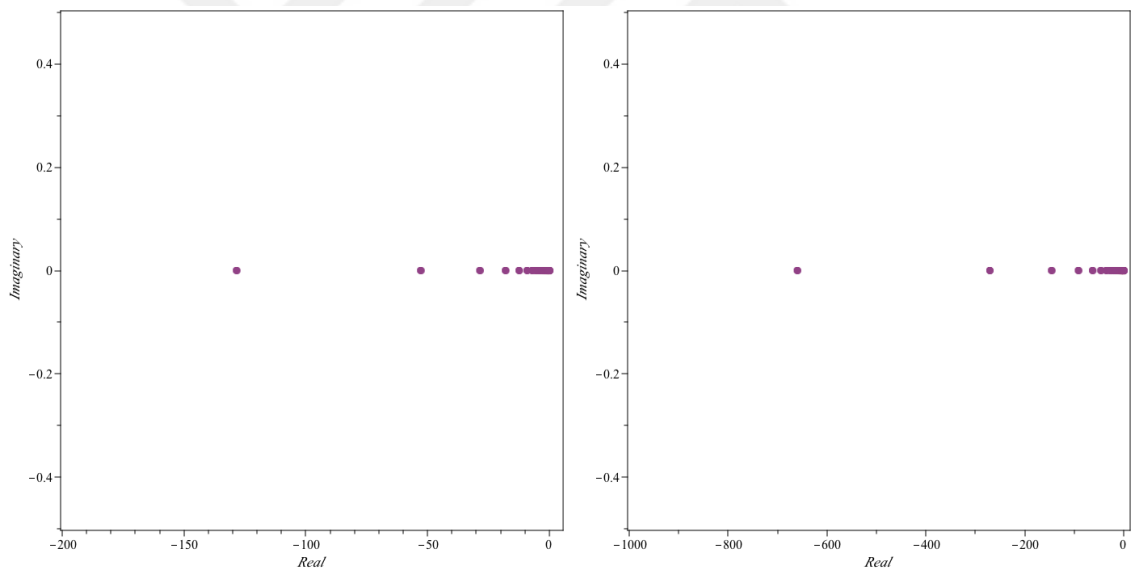


Figure 4.40 Eigenvalue distribution for (c) $N = 80$, (d) $N = 120$

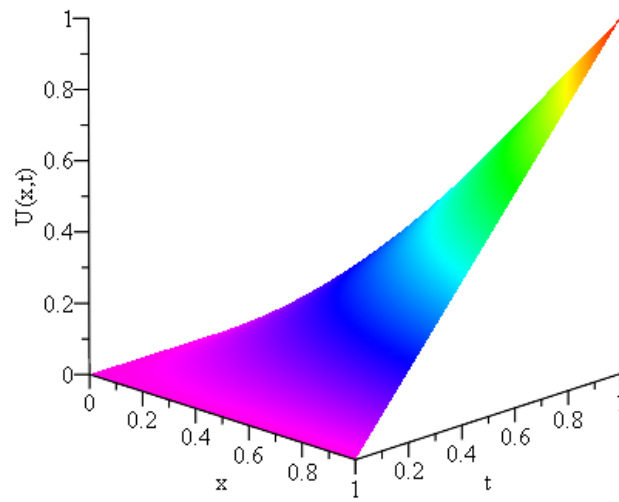


Figure 4.41 Exact solution of Ginzburg Landau Equation

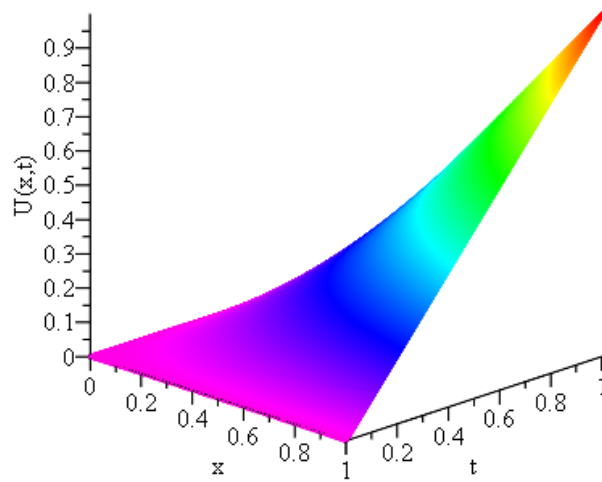


Figure 4.42 Numerical solution of Ginzburg Landau Equation

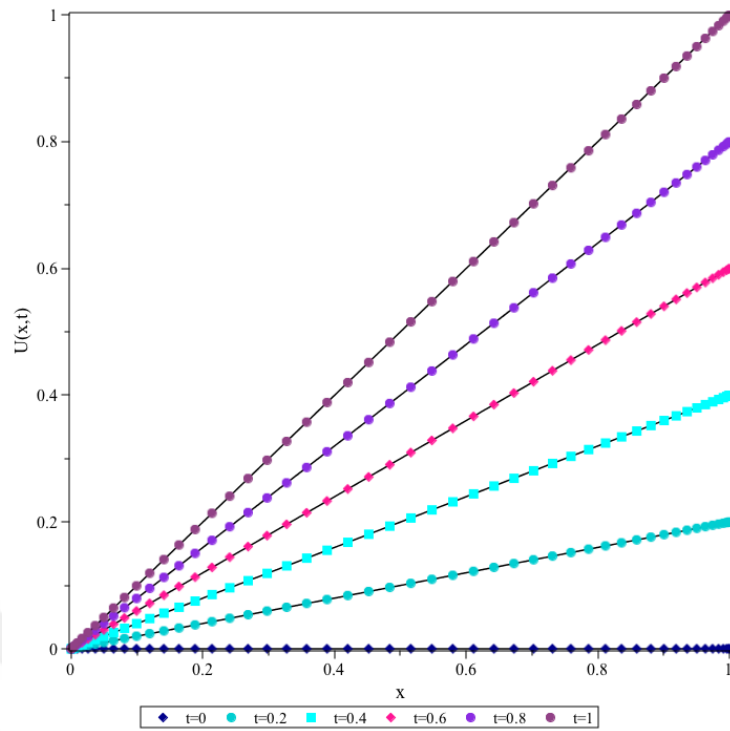


Figure 4.43 Numerical solutions of Ginzburg Landau equation for different values of constant time

Conclusion and Recommendations

This chapter serves to present the conclusive finding of the work accomplished in the current thesis, namely the applications of differential quadrature. We also present some directions for the extension of the current method.

Numerical approach is established for linear and nonlinear models. Most of the solutions were compared with well-known existing studies. The methods works better in linear problems compared to nonlinear models. Based on the results, we conclude that differential quadrature method gives effective solutions to higher order singularly perturbed problems especially when using refined grid point spacing.

In Chapter 4, we have given the main part of this thesis. One dimensional time dependent problems were analyzed in terms of stability analysis which is established by eigenvalues of the discretized matrix. Stability regions for such problems satisfy the stability criteria mentioned in Chapter 2. All numerical results were presented in tables and graphs. Solution scheme of time dependent partial differential equation is constructed with differential quadrature and fourth order Runge Kutta method. In the beginning of this chapter, we set up the scheme for linear and nonlinear diffusion equations. Solutions of diffusion processes were investigated. Quadrature results and literature are in good agreement. We studied reaction-diffusion equation in graded materials. Such problems have not been solved with this technique before. Present approach also offers to serve an alternative methodology to solve equations given in composite media. We studied the problem with different non-homogeneity parameters and we observed that for sufficiently small α values, numerical results show similar behaviors with analytical approach. Chebyshev based differential quadrature method is applied to Kuramoto-Sivashinsky (KS) equation. Periodic and chaotic behaviors of the nonlinear model were simulated. Comparison of analytical and numerical solutions were tabulated in terms of global relative error. Solutions were also compared with literature and observed similar accuracy. We solved the nonlinear advection problems numerically. Sinusoid and Gaussian profile of the inviscid Burgers' equation were simulated. Graphs of such solutions proves the

expected behaviors. But, for higher time levels, oscillations start to evolve towards the boundaries. Ginzburg-Landau equation was solved which is a nonlinear complex equation. We investigate numerical solution of the real part of the equation only. Graphs of analytical solutions and quadrature solutions were depicted to see the effectiveness of the numerical method. Differential quadrature results found in the thesis are in good agreement for problems that could be verified from the literature.

The current method can be applied to fractional order differential equations. Extended quadrature methods such as complex differential quadrature can be studied for multi-dimensional models. Theoretical extensions can also be studied.



References

- [1] J. N. Reddy, *An Introduction to Nonlinear Finite Element Analysis: with applications to heat transfer, fluid mechanics, and solid mechanics*. OUP Oxford, 2014.
- [2] X. Wang, *Differential quadrature and differential quadrature based element methods: theory and applications*. Butterworth-Heinemann, 2015.
- [3] C. W. Bert and M. Malik, “Differential quadrature method in computational mechanics: A review,” *Applied mechanics reviews*, vol. 49, no. 1, pp. 1–28, 1996.
- [4] R. Bellman, B. Kashef, and J Casti, “Differential quadrature: A technique for the rapid solution of nonlinear partial differential equations,” *Journal of computational physics*, vol. 10, no. 1, pp. 40–52, 1972.
- [5] R Bellman, B. Kashef, E. Lee, and R Vasudevan, *Computers and mathematics with applications*, 1976.
- [6] J. Quan and C. Chang, “New insights in solving distributed system equations by the quadrature method—i. analysis,” *Computers & Chemical Engineering*, vol. 13, no. 7, pp. 779–788, 1989.
- [7] J. Quan and C.-T. Chang, “New insights in solving distributed system equations by the quadrature method—ii. numerical experiments,” *Computers & Chemical Engineering*, vol. 13, no. 9, pp. 1017–1024, 1989.
- [8] C. Shu, *Differential quadrature and its application in engineering*. Springer Science & Business Media, 2012.
- [9] A. Korkmaz and I. Dag, “Shock wave simulations using sinc differential quadrature method,” *Engineering Computations*, vol. 28, no. 6, pp. 654–674, 2011.
- [10] J. Cheng, B. Wang, and S.-Y. Du, “A theoretical analysis of piezoelectric/composite laminate with larger-amplitude deflection effect, part ii: Hermite differential quadrature method and application,” *International Journal of Solids and Structures*, vol. 42, no. 24-25, pp. 6181–6201, 2005.
- [11] Y. Wu and C Shu, “Development of rbf-dq method for derivative approximation and its application to simulate natural convection in concentric annuli,” *Computational Mechanics*, vol. 29, no. 6, pp. 477–485, 2002.
- [12] R. Mittal and R. Rohila, “A study of one dimensional nonlinear diffusion equations by bernstein polynomial based differential quadrature method,” *Journal of Mathematical Chemistry*, vol. 55, no. 2, pp. 673–695, 2017.
- [13] D. C. O’Mahoney, “A differential quadrature solution of the two-dimensional inverse heat conduction problem,” *International communications in heat and mass transfer*, vol. 30, no. 8, pp. 1061–1070, 2003.

- [14] I. Dag, A. Korkmaz, and B. Saka, "Cosine expansion-based differential quadrature algorithm for numerical solution of the rlw equation," *Numerical Methods for Partial Differential Equations: An International Journal*, vol. 26, no. 3, pp. 544–560, 2010.
- [15] A. Korkmaz and I. Dag, "A differential quadrature algorithm for simulations of nonlinear schrödinger equation," *Computers & Mathematics with Applications*, vol. 56, no. 9, pp. 2222–2234, 2008.
- [16] A. Korkmaz, I. Dag, "A differential quadrature algorithm for nonlinear schrödinger equation," *Nonlinear Dynamics*, vol. 56, no. 1-2, pp. 69–83, 2009.
- [17] A. Korkmaz and I. Dag, "A differential quadrature algorithm for nonlinear schrodinger equation," *Nonlinear Dynamics*, vol. 56, no. 1-2, pp. 69–83, 2009.
- [18] A. Korkmaz, "Numerical algorithms for solutions of korteweg–de vries equation," *Numerical methods for partial differential equations*, vol. 26, no. 6, pp. 1504–1521, 2010.
- [19] A. Korkmaz and I. Dag, "Polynomial based differential quadrature method for numerical solution of nonlinear burgers' equation," *Journal of the Franklin Institute*, vol. 348, no. 10, pp. 2863–2875, 2011.
- [20] H. Zhong, "Spline-based differential quadrature for fourth order differential equations and its application to kirchhoff plates," *Applied Mathematical Modelling*, vol. 28, no. 4, pp. 353–366, 2004.
- [21] G. Arora and B. K. Singh, "Numerical solution of burgers' equation with modified cubic b-spline differential quadrature method," *Applied Mathematics and Computation*, vol. 224, pp. 166–177, 2013.
- [22] A. Korkmaz, "Numerical solutions of some one dimensional partial differential equations using b-spline differential quadrature method," *Doctoral Dissertation, Eskisehir Osmangazi University, Eskisehir*, 2010.
- [23] A. Korkmaz and I. Dag, "Cubic b-spline differential quadrature methods for the advection-diffusion equation," *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 22, no. 8, pp. 1021–1036, 2012.
- [24] A. Korkmaz, I. Dag, "Cubic b-spline differential quadrature methods and stability for burgers' equation," *Engineering Computations*, vol. 30, no. 3, pp. 320–344, 2013.
- [25] A. Korkmaz and I. Dag, "Numerical simulations of boundary-forced rlw equation with cubic b-spline-based differential quadrature methods," *Arabian Journal for Science and Engineering*, vol. 38, no. 5, pp. 1151–1160, 2013.
- [26] A. Korkmaz, A. M. Aksoy, and I. Dag, "Quartic b-spline differential quadrature method," *Int. J. Nonlinear Sci*, vol. 11, no. 4, pp. 403–411, 2011.
- [27] A. Korkmaz and I. Dag, "Quartic and quintic b-spline methods for advection–diffusion equation," *Applied Mathematics and Computation*, vol. 274, pp. 208–219, 2016.
- [28] A. Başhan, S. B. G. Karakoc, and T. Geyikli, "Approximation of the kdvb equation by the quintic b-spline differential quadrature method," *Kuwait Journal of Science*, vol. 42, no. 2, 2015.

- [29] S. B. G. Karakoç, A. Başhan, and T. Geyikli, "Two different methods for numerical solution of the modified burgers' equation," *The Scientific World Journal*, vol. 2014, 2014.
- [30] A. Başhan, S. B. G. Karakoç, and T. Geyikli, "B-spline differential quadrature method for the modified burgers' equation," *Çankaya Üniversitesi Bilim ve Mühendislik Dergisi*, vol. 12, no. 1, 2015.
- [31] A. Başhan, Y. Uçar, N. M. Yağmurlu, and A. Esen, "A new perspective for quintic b-spline based crank-nicolson-differential quadrature method algorithm for numerical solutions of the nonlinear schrödinger equation," *The European Physical Journal Plus*, vol. 133, no. 1, p. 12, 2018.
- [32] A. Başhan, N. M. Yagmurlu, Y. Ucar, and A. Esen, "An effective approach to numerical soliton solutions for the schrödinger equation via modified cubic b-spline differential quadrature method," *Chaos, Solitons & Fractals*, vol. 100, pp. 45–56, 2017.
- [33] A. Başhan, "An effective application of differential quadrature method based on modified cubic b-splines to numerical solutions of the kdv equation," *Turkish Journal of Mathematics*, vol. 42, no. 1, pp. 373–394, 2018.
- [34] R. Mittal and S. Dahiya, "Numerical simulation on hyperbolic diffusion equations using modified cubic b-spline differential quadrature methods," *Computers & Mathematics with Applications*, vol. 70, no. 5, pp. 737–749, 2015.
- [35] R. Mittal, S. Dahiya, "Numerical solutions of differential equations using modified b-spline differential quadrature method," in *Mathematical Analysis and its Applications*, Springer, 2015, pp. 509–523.
- [36] R. Mittal and S. Dahiya, "A study of quintic b-spline based differential quadrature method for a class of semi-linear fisher-kolmogorov equations," *Alexandria Engineering Journal*, vol. 55, no. 3, pp. 2893–2899, 2016.
- [37] R. Mittal and S. Dahiya, "Numerical simulation of three-dimensional telegraphic equation using cubic b-spline differential quadrature method," *Applied Mathematics and Computation*, vol. 313, pp. 442–452, 2017.
- [38] W. Chen, "Differential quadrature method and its applications in engineering-applying special matrix product to nonlinear computations," *Ph. D. dissertation, Shanghai Jiao Tong University*, 1996.
- [39] C. W. Bert, W. Xinwei, and A. G. Striz, "Differential quadrature for static and free vibration analyses of anisotropic plates," *International Journal of Solids and Structures*, vol. 30, no. 13, pp. 1737–1744, 1993.
- [40] C. W. Bert, S. K. Jang, and A. G. Striz, "Two new approximate methods for analyzing free vibration of structural components," *AIAA journal*, vol. 26, no. 5, pp. 612–618, 1988.
- [41] S. Eftekhari, "Numerical simulation of sloshing motion in a rectangular tank using differential quadrature method," *KSCE Journal of Civil Engineering*, vol. 22, no. 11, pp. 4657–4667, 2018.
- [42] H. B. Khaniki, S. H. Hashemi, and H. B. Khaniki, "Comparison between using generalized differential quadrature method and analytical solution in analyzing vibration behavior of nonuniform nanobeam systems," *Characterization and Application of Nanomaterials*, vol. 1, no. 2, 2018.

- [43] R. Mittal and R. Jiware, "A higher order numerical scheme for some nonlinear differential equations: Models in biology," *International Journal for Computational Methods in Engineering Science and Mechanics*, vol. 12, no. 3, pp. 134–140, 2011.
- [44] M. Akman, "Differential quadrature method for time-dependent diffusion equation," PhD thesis, Citeseer, 2003.
- [45] G. Meral and M. Tezer-Sezgin, "Differential quadrature solution of nonlinear reaction–diffusion equation with relaxation-type time integration," *International Journal of Computer Mathematics*, vol. 86, no. 3, pp. 451–463, 2009.
- [46] G. Meral and M. Tezer-Sezgin, "The differential quadrature solution of nonlinear reaction–diffusion and wave equations using several time-integration schemes," *International journal for numerical methods in biomedical engineering*, vol. 27, no. 4, pp. 485–497, 2011.
- [47] S. Tomasiello, "Stability and accuracy of the iterative differential quadrature method," *International journal for numerical methods in engineering*, vol. 58, no. 9, pp. 1277–1296, 2003.
- [48] S. Tomasiello, "Numerical solutions of the burgers–huxley equation by the idq method," *International Journal of Computer Mathematics*, vol. 87, no. 1, pp. 129–140, 2010.
- [49] S. Tomasiello, "Numerical stability of dq solutions of wave problems," *Numerical Algorithms*, vol. 57, no. 3, pp. 289–312, 2011.
- [50] S. Tomasiello, "A new least-squares dq-based method for the buckling of structures with elastic supports," *Applied Mathematical Modelling*, vol. 39, no. 9, pp. 2809–2814, 2015.
- [51] T. Fung, "Solving initial value problems by differential quadrature method—part 1: First-order equations," *International Journal for Numerical Methods in Engineering*, vol. 50, no. 6, pp. 1411–1427, 2001.
- [52] T. Fung, "Solving initial value problems by differential quadrature method—part 2: Second-and higher-order equations," *International Journal for Numerical Methods in Engineering*, vol. 50, no. 6, pp. 1429–1454, 2001.
- [53] C Shu, Q Yao, and K. Yeo, "Block-marching in time with dq discretization: An efficient method for time-dependent problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 41-42, pp. 4587–4597, 2002.
- [54] J. Liu and X. Wang, "An assessment of the differential quadrature time integration scheme for nonlinear dynamic equations," *Journal of Sound and Vibration*, vol. 314, no. 1-2, pp. 246–253, 2008.
- [55] R. Peyret, *Spectral methods for incompressible viscous flow*. Springer Science & Business Media, 2013, vol. 148.
- [56] R. S. Johnson, *Singular perturbation theory: Mathematical and analytical techniques with applications to engineering*. Springer Science & Business Media, 2006.
- [57] C. Bender and S. Orszag, "Advanced mathematical methods for scientists and engineers, mcgraw-hill book company inc., new york, 1978,"

- [58] R. E. O'malley, *Singular perturbation methods for ordinary differential equations*. Springer, 1991, vol. 89.
- [59] M. Kumar and S. Tiwari, "An initial-value technique to solve third-order reaction–diffusion singularly perturbed boundary-value problems," *International Journal of Computer Mathematics*, vol. 89, no. 17, pp. 2345–2352, 2012.
- [60] F. Howes, "The asymptotic solution of a class of third-order boundary value problems arising in the theory of thin film flows," *SIAM Journal on Applied Mathematics*, vol. 43, no. 5, pp. 993–1004, 1983.
- [61] Q. Yao and Y. Feng, "The existence of solution for a third-order two-point boundary value problem," *Applied Mathematics Letters*, vol. 15, no. 2, pp. 227–232, 2002.
- [62] S Valarmathi and N Ramanujam, "Boundary value technique for finding numerical solution to boundary value problems for third order singularly perturbed ordinary differential equations," *International journal of computer mathematics*, vol. 79, no. 6, pp. 747–763, 2002.
- [63] K Phaneendra, Y. Reddy, and G. Soujanya, "Asymptotic-numerical method for third-order singular perturbation problems," *International Journal of Applied Science and Engineering*, vol. 10, no. 3, pp. 241–248, 2012.
- [64] A. R. Babu and N Ramanujam, "An asymptotic finite element method for singularly perturbed third and fourth order ordinary differential equations with discontinuous source term," *Applied Mathematics and Computation*, vol. 191, no. 2, pp. 372–380, 2007.
- [65] M. Cui and F. Geng, "A computational method for solving third-order singularly perturbed boundary-value problems," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 896–903, 2008.
- [66] V Shanthi and N Ramanujam, "A boundary value technique for boundary value problems for singularly perturbed fourth-order ordinary differential equations," *Computers & Mathematics with Applications*, vol. 47, no. 10-11, pp. 1673–1688, 2004.
- [67] K. Viswanadham, P. M. Krishna, and R. S. Koneru, "Numerical solutions of fourth order boundary value problems by galerkin method with quintic b-splines," *International Journal of Nonlinear Science*, vol. 10, no. 2, pp. 222–230, 2010.
- [68] A Ali, M. A. Iqbal, and S. Mohyud-Din, "Hermite wavelets method for boundary value problems," *International Journal of Modern Applied Physics*, vol. 3, no. 1, pp. 38–47, 2013.
- [69] G. Yiğit and M. Bayram, "Chebyshev differential quadrature for numerical solutions of third-and fourth-order singular perturbation problems," *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, pp. 1–8,
- [70] E. A. Saied, "The non-classical solution of the inhomogeneous non-linear diffusion equation," *Applied Mathematics and Computation*, vol. 98, no. 2-3, pp. 103–108, 1999.

- [71] G. Gürarlan, "Numerical modelling of linear and nonlinear diffusion equations by compact finite difference method," *Applied Mathematics and Computation*, vol. 216, no. 8, pp. 2472–2478, 2010.
- [72] G. Gürarlan and M. Sari, "Numerical solutions of linear and nonlinear diffusion equations by a differential quadrature method (dqm)," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 27, no. 1, pp. 69–77, 2011.
- [73] Q. Changzheng, "Exact solutions to nonlinear diffusion equations obtained by a generalized conditional symmetry method," *IMA Journal of Applied Mathematics*, vol. 62, no. 3, pp. 283–302, 1999.
- [74] J. Crank *et al.*, *The mathematics of diffusion*. Oxford university press, 1979.
- [75] A.-M. Wazwaz, "The variational iteration method: A powerful scheme for handling linear and nonlinear diffusion equations," *Computers & Mathematics with Applications*, vol. 54, no. 7-8, pp. 933–939, 2007.
- [76] A.M. Wazwaz, "Exact solutions to nonlinear diffusion equations obtained by the decomposition method," *Applied Mathematics and Computation*, vol. 123, no. 1, pp. 109–122, 2001.
- [77] C.-N. Chen, "The development of irregular elements for differential quadrature element method steady-state heat conduction analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 170, no. 1-2, pp. 1–14, 1999.
- [78] A.-M. Wazwaz, *Partial differential equations and solitary waves theory*. Springer Science & Business Media, 2010.
- [79] Y. Miyamoto, W. Kaysser, B. Rabin, A. Kawasaki, and R. G. Ford, *Functionally graded materials: design, processing and applications*. Springer Science & Business Media, 2013, vol. 5.
- [80] L. Marin and D. Lesnic, "The method of fundamental solutions for nonlinear functionally graded materials," *International journal of solids and structures*, vol. 44, no. 21, pp. 6878–6890, 2007.
- [81] A. Sutradhar, G. H. Paulino, and L. Gray, "Transient heat conduction in homogeneous and non-homogeneous materials by the laplace transform galerkin boundary element method," *Engineering Analysis with Boundary Elements*, vol. 26, no. 2, pp. 119–132, 2002.
- [82] N. Zhao, L. L. Cao, and H. Guo, "Transient heat conduction in functionally graded materials by lt-mfs," in *Advanced Materials Research*, Trans Tech Publ, vol. 189, 2011, pp. 1664–1669.
- [83] B. Yu, H.-L. Zhou, J. Yan, and Z. Meng, "A differential transformation boundary element method for solving transient heat conduction problems in functionally graded materials," *Numerical Heat Transfer, Part A: Applications*, vol. 70, no. 3, pp. 293–309, 2016.
- [84] J. Tian and K Jiang, "Heat conduction investigation of functionally graded material plates under the exponential heat source load," *Numerical Heat Transfer, Part A: Applications*, vol. 72, no. 2, pp. 141–152, 2017.

- [85] Y. Karagöz, “Adomian ayrışım metodu ile ısı transferi modellemesi,” *Master’s Thesis, Yıldız Technical University, Istanbul*, 2012.
- [86] G. Yiğit, A. Şahin, and M. Bayram, “Modeling of vibration for functionally graded beams,” *Open Mathematics*, vol. 14, no. 1, pp. 661–672, 2016.
- [87] S. Rajasekaran, “Free vibration of centrifugally stiffened axially functionally graded tapered timoshenko beams using differential transformation and quadrature methods,” *Applied Mathematical Modelling*, vol. 37, no. 6, pp. 4440–4463, 2013.
- [88] S. Rajasekaran, “Differential transformation and differential quadrature methods for centrifugally stiffened axially functionally graded tapered beams,” *International Journal of Mechanical Sciences*, vol. 74, pp. 15–31, 2013.
- [89] A. Vosoughi, “Thermal postbuckling analysis of functionally graded beams,” *Journal of Thermal Stresses*, vol. 37, no. 4, pp. 532–544, 2014.
- [90] P. Jiao, Y. Wang, G. Xu, Q. Yuan, W. You, *et al.*, “Linear bending of functionally graded beams by differential quadrature method,” in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, vol. 170, 2018, p. 022 160.
- [91] D. Lesnic, “The decomposition method for cauchy reaction–diffusion problems,” *Applied Mathematics Letters*, vol. 20, no. 4, pp. 412–418, 2007.
- [92] Y. Kuramoto and T. Tsuzuki, “Persistent propagation of concentration waves in dissipative media far from thermal equilibrium,” *Progress of theoretical physics*, vol. 55, no. 2, pp. 356–369, 1976.
- [93] Y. Kuramoto and T. Tsuzuki, “On the formation of dissipative structures in reaction-diffusion systems: Reductive perturbation approach,” *Progress of Theoretical Physics*, vol. 54, no. 3, pp. 687–699, 1975.
- [94] G. I. Sivashinsky, “Instabilities, pattern formation, and turbulence in flames,” *Annual Review of Fluid Mechanics*, vol. 15, no. 1, pp. 179–199, 1983.
- [95] A. Khater and R. Temsah, “Numerical solutions of the generalized kuramoto–sivashinsky equation by chebyshev spectral collocation methods,” *Computers & Mathematics with Applications*, vol. 56, no. 6, pp. 1465–1472, 2008.
- [96] R. Mittal and G. Arora, “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, 2010.
- [97] O. Ersoy and I. Dag, “The exponential cubic b-spline collocation method for the kuramoto–sivashinsky equation,” *Filomat*, vol. 30, no. 3, pp. 853–861, 2016.
- [98] G. D. Akrivis, “Finite difference discretization of the kuramoto-sivashinsky equation,” *Numerische Mathematik*, vol. 63, no. 1, pp. 1–11, 1992.
- [99] Y. Xu and C.-W. Shu, “Local discontinuous galerkin methods for the kuramoto–sivashinsky equations and the ito-type coupled kdv equations,” *Computer methods in applied mechanics and engineering*, vol. 195, no. 25-28, pp. 3430–3447, 2006.
- [100] M. Uddin, S. Haq, *et al.*, “A mesh-free numerical method for solution of the family of kuramoto–sivashinsky equations,” *Applied Mathematics and Computation*, vol. 212, no. 2, pp. 458–469, 2009.

- [101] M. G. Porshokouhi and B. Ghanbari, "Application of he's variational iteration method for solution of the family of kuramoto–sivashinsky equations," *Journal of King Saud University-Science*, vol. 23, no. 4, pp. 407–411, 2011.
- [102] J. D. Rademacher and R. W. Wittenberg, "Viscous shocks in the destabilized kuramoto-sivashinsky equation," *Journal of computational and nonlinear dynamics*, vol. 1, no. 4, pp. 336–347, 2006.
- [103] E. Cerpa, P. Guzmán, and A. Mercado, "On the control of the linear kuramoto-sivashinsky equation," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 23, no. 1, pp. 165–194, 2017.
- [104] N. Carreno and E. Cerpa, "Local controllability of the stabilized kuramoto–sivashinsky system by a single control acting on the heat equation," *Journal de Mathématiques Pures et Appliquées*, vol. 106, no. 4, pp. 670–694, 2016.
- [105] H. Lai and C. Ma, "Lattice boltzmann method for the generalized kuramoto–sivashinsky equation," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 8, pp. 1405–1412, 2009.
- [106] R. Mittal and S. Dahiya, "A quintic b-spline based differential quadrature method for numerical solution of kuramoto-sivashinsky equation," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 18, no. 2, pp. 103–114, 2017.
- [107] O. E. Hepson, "Generation of the trigonometric cubic b-spline collocation solutions for the kuramoto-sivashinsky (ks) equation," in *AIP Conference Proceedings*, AIP Publishing, vol. 1978, 2018, p. 470 099.
- [108] G. Yigit and M. Bayram, "Polynomial based differential quadrature for numerical solutions of kuramoto-sivashinsky equation," *Thermal Science*, vol. 23, no. 1, S129–S137, 2019.
- [109] L. Debnath, *Nonlinear partial differential equations for scientists and engineers*. Springer Science & Business Media, 2011.
- [110] B. R. Kumar and M. Mehra, "A wavelet-taylor galerkin method for parabolic and hyperbolic partial differential equations," *International Journal of Computational Methods*, vol. 2, no. 01, pp. 75–97, 2005.
- [111] M. Landajuela, "Burgers equation," *BCAM Internship: Basque center for applied mathematics*, 2011.
- [112] I. S. Aranson and L. Kramer, "The world of the complex ginzburg-landau equation," *Reviews of Modern Physics*, vol. 74, no. 1, p. 99, 2002.
- [113] M. Gabbay, E. Ott, and P. N. Guzdar, "Reconnection of vortex filaments in the complex ginzburg-landau equation," *Physical Review E*, vol. 58, no. 2, p. 2576, 1998.
- [114] A. Secer and Y. Bakir, "Chebyshev wavelet collocation method for ginzburg-landau equation," *Thermal Science*, vol. 23, no. 1, S57–S65, 2019.
- [115] M. Li, C. Huang, and N. Wang, "Galerkin finite element method for the nonlinear fractional ginzburg–landau equation," *Applied Numerical Mathematics*, vol. 118, pp. 131–149, 2017.

- [116] M. Inc and N. Bildik, “Non-perturbative solution of the ginzburg-landau equation,” *Mathematical and Computational Applications*, vol. 5, no. 2, pp. 113–117, 2000.
- [117] J Biazar, M Partovi, and Z Ayati, “Approximating solutions for ginzburg–landau equation by hpm and adm,” *Appl Appl Math*, pp. 1672–1681, 2010.



Publications From the Thesis

Contact Information: gulsemay.yigit@altinbas.edu.tr

Papers

1. G. Yiğit and M. Bayram, “Chebyshev Differential Quadrature for Numerical Solutions of Third-and Fourth-Order Singular Perturbation Problems”, *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, pp. 1-8, 2019.
2. G. Yiğit and M. Bayram, “Polynomial Based Differential Quadrature for Numerical Solutions of Kuramoto-Sivashinsky Equation”, *Thermal Science*, vol. 23 no.1, S129-S137, 2019.

Conference Papers

1. G. Yiğit and M. Bayram, “Numerical Simulations of Kuramoto-Sivashinsky Equation by Chebyshev Based Differential Quadrature”, VI. Kadın Matematikçiler Derneği Çalıştayı, Konya, TURKEY, 26-28 April 2019, pp.36.
2. G. Yiğit and M. Bayram, “Chebyshev Differential Quadrature for Quasilinear Hyperbolic Equations”, 7th International Conference on Applied Analysis and Mathematical Modeling (ICAAMM 2018), ISTANBUL, TURKEY, 20-24 June 2018, pp.82.
3. G. Yiğit and M. Bayram, “Polynomial Based Differential Quadrature for Numerical of Kuramoto-Sivashinsky Equation”, International Conference on Applied Analysis and Mathematical Modeling (SciCADE 2017), University of Bath, 11-15 September 2017, Bath, United Kingdom, 2017, pp.185.
4. G. Yiğit and M. Bayram, “Chebyshev Differential Quadrature for Advection-Diffusion Equation”, International Conference on Applied Analysis and Mathematical Modeling (ICAAMM 2017), ISTANBUL, TURKEY, 03-07 July 2017, pp.193.
5. Yiğit, G., and Bayram, M. “Numerical Solutions for Higher Order Singular Perturbation Problems by Polynomial Based Differential Quadrature”, 3rd International Conference on Pure and Applied Sciences (ICPAS 2017), Dubai, UAE 02-05 February 2017, pp.25.

6. Yiğit, G., “Numerical Solutions for Nonlinear Advection Problems by Chebyshev Differential Quadrature”, International Conference on Mathematics and Engineering (ICOME 2017), ISTANBUL, TURKEY, 10-12 May 2017, pp.187.

