FATİH UNIVERSITY

The Graduate School of Sciences and Engineering

Master of Science in Mathematics

SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS BY DIFFERENTIAL QUADRATURE METHOD

by

İbrahim YAZICI

June 2016

İbrahim YAZICI

SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS BY DIFFERENTIAL QUADRATURE METHOD

M.S. 2016

SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS BY DIFFERENTIAL QUADRATURE METHOD

by

İbrahim YAZICI

A thesis submitted to

the Graduate School of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mathematics

June 2016
Istanbul, Turkey

APPROVAL PAGE

This is to certify that I have read this thesis written by İbrahim YAZICI and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Mathematics.

Assoc. Prof. Ali ŞAHİN
Thesis Supervisor

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Mathematics.

Prof. Feyzi BAŞAR
Head of Department

Examining Committee Members

Prof. Mustafa BAYRAM

Assoc. Prof. İbrahim KARATAY

Assoc. Prof. Ali ŞAHİN

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate School of Sciences and Engineering.

Prof. Nurullah ARSLAN
Director

June 2016

SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS BY DIFFERENTIAL QUADRATURE METHOD

İbrahim YAZICI

M.S. Thesis – Mathematics
June 2016

Thesis Supervisor: Assoc. Prof. Ali ŞAHİN

ABSTRACT

In this thesis, the differential quadrature method is presented. The weighted coefficients of the first, second and higher order derivatives are obtained by polynomial based differential quadrature method along with different approaches. The method is applied to solve initial and boundary value problems of ordinary differential equations using different grid point distributions. The effect of grid point distributions on accuracy of solution are demonstrated by results. Afterwards, solutions of initial and boundary value problems of partial differential equations are examined using polynomial based differential quadrature method along with finite difference method. Eventually, the sample advection-diffusion-reaction problems and their solutions by polynomial based differential quadrature method are studied. In the calculations of algebraic systems, MATLAB programming is used and all the algorithms written in MATLAB are given in Appendices.

Keywords: Differential quadrature method, polynomial based differential quadrature method, initial and boundary value problems, ordinary differential equations, partial differential equations, advection-diffusion-reaction equations.

DİFERANSİYEL KARESELLEŞTİRME METODU İLE KISMİ TÜREVLİ DİFERANSİYEL DENKLEMLERİN ÇÖZÜMÜ

İbrahim YAZICI

Yüksek Lisans Tezi – Matematik
Haziran 2016

Tez Danışmanı: Doç. Dr. Ali ŞAHİN

ÖZ

Bu tezde diferansiyel kareselleştirme metodu sunuldu. Birinci, ikinci ve yüksek mertebeli türevlerin ağırlık katsayıları farklı yaklaşımlar kullanılarak polinom tabanlı diferansiyel kareselleştirme metodu ile elde edildi. Metod, farklı grid nokta dağılımları altında adi diferansiyel denklemlerin başlangıç ve sınır değer problemlerinin çözülmesinde kullanıldı. Grid nokta dağılımının çözümün doğruluğuna etkisi sonuçlarla gösterildi. Sonrasında, kısmi türevli diferansiyel denklemlerin başlangıç ve sınır değer problemlerinin çözümleri polinom tabanlı diferansiyel kareselleştirme metodu ve sonlu farklar metodu birlikte kullanılarak incelendi. Son olarak, adveksiyon-difüzyon-reaksiyon problemleri ve çözümlerinin diferansiyel kareselleştirme metodu ile elde edilmesi çalışıldı. Bütün cebirsel sistem hesaplamalarında MATLAB programı kullanıldı ve MATLAB'ta yazılan algoritmalar eklerde verildi.

Anahtar Kelimeler: Diferansiyel kareselleştirme metodu, polinom tabanlı diferansiyel kareselleştirme metodu, başlangıç ve sınır değer problemleri, adi diferansiyel denklemler, kısmi türevli diferansiyel denklemler, adveksiyon-difüzyon-reaksiyon denklemleri.

To my wife

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor Assoc. Prof. Ali ŞAHİN for the continuous support of my study and research, for his patience, motivation, and enthusiasm.

I would also like to express my very profound appreciation to my wife and to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. Lastly, but in no sense the least, I am thankful to all colleagues and friends who made my stay at the university a memorable and valuable experience.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF SYMBOLS AND ABBREVIATIONS

SYMBOL/ABBREVIATION

a_{ij}	The weighted coefficients of the first order derivative
$w_{ij}^{(2)}$	The weighted coefficients of the second order derivative
$w_{ij}^{(m)}$	The weighted coefficients of the (m)'th order derivative
$r_k(x)$	Basis polynomials
$K_N(x)$	N'th order shifted Legendre polynomial
$L_N(x)$	N'th order Legendre polynomial
$M(x)$	Lagrange interpolation polynomials
δ_{ij}	Kronecker operator
ADR	Advection-Diffusion-Reaction
CDQ	Cosine Expansion-based Differential Quadrature
CGL	Chebyshev-Gauss-Lobatto
DQ	Differential Quadrature
FDQ	Fourier Expansion-based Differential Quadrature
KdV	Korteweg-De Vries
ODEs	Ordinary Differential Equations
PDEs	Partial Differential Equations
PDQ	Polynomial-based Differential Quadrature
RLW	Regularized Long Wave

CHAPTER 1

INTRODUCTION

Despite the developments, problems involving nonlinearity, discontinuity, multiple scale, singularity and irregularity continue to cause difficulties in the field of computational science and engineering. The numerical solutions might be the only way to solve problems when there is not any analytical solution. Thus, there have been improved a variety of numerical solution methods. Among all these methods, the differential quadrature has distinguished because of high accuracy, obvious application and generality in various problems. As a result, differential quadrature method has seen a great increase in research interest and experienced significant development [1-23].

Most of the modeling of engineering problems can be solved by numerical methods such as finite difference, finite element and finite volume method. However, these methods necessitate the usage of a large number of grid points. In order to obtain accurate numerical solutions using small number of grid points, Bellman and his friends developed Differential Quadrature (DQ) method. In DQ method, the partial derivatives of a function at grid points with respect to a coordinate direction were expressed as a linear weighted sum of functional values of all the grid points. The crucial point of the method is computation of weighted coefficients of derivatives of any order. Bellman presented two methods to compute weighted coefficients [2,3]. In the first method, a system of algebraic equations is solved and the coordinates of the grid points can be selected optionally. The disadvantage of the method is the difficulty of solving of the system of algebraic equation when its order is large. For this reason, Bellman’s first approach could be applied when the number of grid points is less than or equal to 13. In the second approach of Bellman, a simple algebraic equation is used. The drawback of this method is that, the grid points are chosen from the roots of Legendre polynomials. Two approaches of Bellman are used in early applications of DQ. There have been made some efforts to develop DQ method by other scientists [4-23].

Mingle [4] obtained numerical solutions to transient nonlinear diffusion problems by DQ method. Civan and Sliepcevich [5] solved problems related to transport phenomena using DQ method and observed that DQ is more advantageous compared to conventional finite element and finite difference methods because of reduction in programming effort and computational time.

Quan and Chang [6,7] advanced a new approach for computation of weighted coefficients of the first and the second order derivatives by applying Lagrange interpolation polynomials as test functions. Shu and Richards [8,9] generalized the methods until then and developed formulations by analyzing high order polynomial approximation and linear vector space. In their approach, weighted coefficients of the first order derivative are determined by simple algebraic formulation and weighted coefficients of second and higher order derivatives determined by a recurrence relationship. Shu [10] also wrote a book about DQ and its applications in engineering. In this book, the DQ method was examined in details.

Shu and Xue [11] solved two-dimensional Helmholtz equation by polynomial based differential quadrature (PDQ) and Fourier based differential quadrature (FDQ). It was observed that FDQ approach is more effective than PDQ to obtain accurate solutions for Helmholtz equation. Bert and Malik [12] wrote a review article about using DQ method in computational mechanics. In this study, DQ applied to integrodifferential equation, torsion of rectangular-cross-section shaft and one-dimensional, time-dependent heat diffusion in a sphere.

Zhong [13] developed a quintic B-spline based differential quadrature method to solve fourth order differential equations. Explicit expressions of weighted coefficients for approximation of derivatives are obtained by constructing cardinal spline interpolations using the normalized quintic B-spline functions. Numerical results showed that the spline based differential quadrature is an effective and alternative to the conventional differential quadrature. Zhong and Guo [14] applied the spline based differential quadrature method to vibration analysis of rectangular plates with free corners and acquired accurate results.

In the study of Shu, Ding and Yeo [15] the idea of DQ method extended to a general case in such a way that any spatial derivative is approximated by a linear weighted sum of all the functional values in the complete physical domain. The weighted coefficients in the new approach are determined by radial basis functions. The method can be consistently well applied to linear and nonlinear problems. Numerical examples showed that radial basis functions based on differential quadrature method have a potential to become an efficient approach for solving partial differential equations.

Since conventional DQ method requires the function to be determined highly differentiable, Zong, Lam and Zhang [16] developed a multi-domain differential quadrature approach to plane elastic problems with material discontinuity. The method has the first order accuracy at the interfaces of two different materials, but it is high order accurate elsewhere. Numerical examples demonstrated effectiveness of the method.

Shu, Chen, Xue and Du [17] dealt with a study about the effect of grid point distribution on the accuracy of DQ solution for beams and plates. It was seen that the stretching of grid towards the boundary gives results that are more accurate in DQ method. Besides, it was observed that the optimal grid distribution might not be from the roots of orthogonal polynomials. This work also gives a simple and effective formulation for stretching the grid towards the boundary.

Akman [18] applied DQ method together with fourth order Runge-Kutta method to solve time dependent diffusion equation. Derivatives in space directions were discretized with DQ method while time derivative was discretized by fourth order Runge-Kutta method. It was observed that the combination of these two methods gives accurate, efficient and stable solutions.

Korkmaz [19] examined the numerical solutions of one-dimensional non-linear wave equation. He used DQ and Cosine expansion based differential quadrature (CDQ) methods to acquire solutions of Regularized Long Wave (RLW), Korteweg-De Vries (KdV), Burger, Kawahara and modified Kawahara equations. Considering high convergent results, it was stated that DQ method could be accepted as a sufficient alternative method to solve one-dimensional PDEs.

Yucel [20] applied both PDQ and FDQ methods to compute the eigenvalues of Sturm-Liouville problem. It was found that in order to obtain accurate numerical results for the first kth eigenvalues of the problem ($k = 1, 2, 3, \dots$), the minimum number of grid points N must be equal to $2k$. It was also found that when the number of grid points increased to $2k$, the accuracy of the DQ results can be improved. Additionally, effectiveness of Fourier DQ approach was observed compared to PDQ approach.

Gurarslan and Sarı [21] studied linear and nonlinear diffusion processes using a combination of PDQ method in space and third order Runge-Kutta method in time. The method is seen to be a very good choice while dealing with nonlinear diffusion problems. Temelcan [22] studied on the solution of some types of linear ODEs and PDEs by PDQ method. Kurtar [23] also analyzed PDQ and FDQ in details.

In this study, the PDQ method was examined in details and some sample problems of ODEs and PDEs were solved. In Chapter 2, the method of differential quadrature and its properties are examined in details. Some examples related to the linear homogeneous and non-homogeneous ordinary differential equations are solved and results are shown in tables by comparing close-form solution of each differential equation. In Chapter 3, similar to Chapter 2, solutions of partial differential equations are examined and solutions are compared with close-form solutions obtained by the separation of variables method. In addition, since it is used in the solution process of PDEs in time direction, the finite difference method is explained shortly. Finally, in Chapter 4 that is the main part of this study, the sample advection-diffusion-reaction problems and their solutions by PDQ method are examined. Results are shown in tables by comparing exact solutions. At the end of this study, the conclusions of the study along with the future projects are added to the text.

CHAPTER 2

DIFFERENTIAL QUADRATURE METHOD

Differential Quadrature (DQ) is a numerical solution method that has been used in the solutions of Ordinary Differential Equations (ODEs) and Partial Differential Equations (PDEs). It was introduced by R. E. Bellman and his friends in 1971. In the method, derivative of a function at a point is expressed as a linear weighted sum of the functional values of all the grid points. It was come up with the idea of integral quadrature.

Pursuant to this method, let the function $f(x)$ be continuous on the region it is defined and x_i be grid points in the region. The value of the first derivative at a point will be defined as a linear sum of the values of all grid points as follows

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

The formula (2.1) is called differential quadrature (DQ). In the formula a_{ij} corresponds to weighting coefficients. The significant point of the method is computing weighting coefficients. The values of a_{ij} will differ in the distinct points of x_i.

In the computation of weighting coefficients, test functions are used. A test function is the function that is convergent to the function of the problem $f(x)$. There are different approaches according to the selection of the test functions. The two of the essential approaches are polynomial based differential quadrature method (PDQ) and Fourier based differential quadrature method (FDQ). In this study, the PDQ will be investigated.

In the beginning, some definitions and properties that are used in computation of weighting coefficients will be introduced.

2.1 DEFINITION AND PROPERTIES OF LINAR VECTOR SPACES

The properties of Linear Vector Spaces are used to compute the weighted coefficients. Thus, some definitions and properties need to be known will be given in this subsection.

Definition 2.1.1 (Definition of a Field):

A field is a set of scalar elements that has addition "$+$" and multiplication "$\times$" operators. In the field ***F*** the operators must satisfy the conditions given below.

(1) For all the elements $a,b \in F$, there must exist $a+b \in F$ and $a \times b \in F$.

(2) Addition and multiplication operators are commutative such that

$a+b=b+a$ and $a \times b = b \times a$ for all $a,b \in F$.

(3) Addition and multiplication operators are associative such that

$(a+b)+c = a+(b+c)$ and $(a \times b) \times c = a \times (b \times c)$ for all $a,b,c \in F$.

(4) Multiplication operator is distributive in respect of addition such that

$a \times (b+c) = (a \times b) + (a \times c)$ for all $a,b,c \in F$.

(5) F includes two elements "0" and "1" such that

$a+0=a$ and $1 \times a = a$ for all $a \in F$.

(6) For every element a in F , there exists an element b such that

$a+b=0$.

(7) For every element a in F , there exists an element c such that

$a \times c = 1$.

Considering the features given above, any set of objects for which the addition and multiplication operators can be defined in, may form a field. In our study, we will encounter the most familiar ones, such as real numbers and rational functions with real coefficients. The two operators in these fields are defined in the usual way.

Definition 2.1.2 (Definition of a Linear Vector Space):

A set of elements called vectors, a field F and two operators named vector addition and scalar multiplication constitutes a linear vector space. The two operators must satisfy the following conditions.

(1) For all the elements $\alpha, \beta \in V$, there must exist $\alpha + \beta \in V$. Here $\alpha + \beta$ is the vector addition of α and β .

(2) Vector addition is commutative such that

$\alpha+\beta=\beta+\alpha$ for all $\alpha,\beta\in V$.

(3) Vector addition is associative such that

$(\alpha+\beta)+\gamma=\alpha+(\beta+\gamma)$ for all $\alpha,\beta,\gamma\in V$.

(4) V includes a vector "0" called zero vector such that

$0+\alpha=\alpha$ for all $\alpha\in V$.

(5) For every vector $\alpha\in V$ there exists a vector $\beta\in V$ such that

$\alpha+\beta=0$.

(6) For every $c\in F$ and $\alpha\in V$ there exists the vector $c\alpha\in V$ named as "the scalar product of c and α"

(7) Scalar multiplication is associative such that for any $a,b\in F$ and $\gamma\in V$,

$a(b\gamma)=(ab)\gamma$.

(8) Scalar multiplication is distributive in respect of vector addition such that for any $a\in F$ and $\beta,\gamma\in V$, $a(\beta+\gamma)=a\beta+a\gamma$.

(9) Scalar multiplication is distributive in respect of scalar addition such that for any $a,b\in F$ and $\gamma\in V$, $(a+b)\gamma=a\gamma+b\gamma$.

(10) For all $\alpha\in V$, there $1\alpha=\alpha$ where 1 is the element 1 in F .

As an example of linear vector spaces we can consider $P_n(x)$ which is the set of all polynomials of degree less than n with real number coefficients such that

$$P_n(x)=\sum_{i=0}^{n-1}c_i x^i \tag{2.2}$$

$$\sum_{i=0}^{n-1}c_i x^i+\sum_{i=0}^{n-1}d_i x^i=\sum_{i=0}^{n-1}(c_i+d_i)x^i \tag{2.3}$$

$$a\left(\sum_{i=0}^{n-1}c_i x^i\right)=\sum_{i=0}^{n-1}(ac_i)x^i \tag{2.4}$$

The vector addition defined in (2.3) and scalar multiplication defined in (2.4) satisfies the properties of operators of linear vector spaces. Thus, $P_n(x)$ constitutes a linear vector space.

Some properties that are used in PDQ can be explained as follows.

Definition 2.1.3 (Linear Independence):

Let $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ be a set of vectors in a linear vector space V and $c_1, c_2, c_3, \ldots, c_n$ be a set of element in a field F. If the equation

$$c_1\alpha_1 + c_2\alpha_2 + c_3\alpha_3 + \cdots + c_n\alpha_n = 0 \tag{2.5}$$

is satisfied only in the case $c_1 = c_2 = \cdots = c_n = 0$ then the vectors $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ are said to be linearly independent. One can see that if the vectors are linearly dependent, then at least one of the vectors can be expressed as a linear combination of the others.

Definition 2.1.4 (Dimension of a Linear Vector Space):

In a linear vector space V, the highest number of linearly independent vectors is called to be the dimension of the linear vector space.

Definition 2.1.5 (Basis of a Linear Vector Space):

Let B be a set of linearly independent vectors that has finite number of elements. If every vector in V can be expressed as a unique linear combination of elements of B then B is called as a basis of V.

In a n dimensional linear vector space, V_n any set of linearly independent vectors can be considered as a basis. In order to show this property, let α be an arbitrary vector in V_n and let $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ be any n linearly independent vectors in V_n. From the definition of the dimension of a linear vector space, it can easily be seen that the set of $(n+1)$ vectors $\alpha, \alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ is linearly dependent. So, the following equation can be written in which $c_0, c_1, c_2, c_3, \ldots, c_n$ in F, not all zero

$$c_0\alpha + c_1\alpha_1 + c_2\alpha_2 + c_3\alpha_3 + \cdots + c_n\alpha_n = 0. \tag{2.6}$$

In the Eq. (2.6), the coefficient c_0 should not be zero, $c_0 \neq 0$. If $c_0 = 0$, then the Eq. (2.5) will be obtained. It causes that $c_0 = c_1 = \cdots = c_n = 0$. This result contradicts the linear dependence of $\alpha, \alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$, thus $c_0 \neq 0$. Let $d_i = -\frac{c_i}{c_0}$, then the Eq. (2.6) becomes as follows

$$\alpha = d_1\alpha_1 + d_2\alpha_2 + d_3\alpha_3 + \cdots + d_n\alpha_n. \tag{2.7}$$

From Eq. (2.7) we can say that all vectors in V_n can be expressed as a linear combination of $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$. Let us show the uniqueness of this combination. Suppose that there is another linear combination of α such that

$$\alpha = \overline{d_1}\alpha_1 + \overline{d_2}\alpha_2 + \overline{d_3}\alpha_3 + \cdots + \overline{d_n}\alpha_n . \tag{2.8}$$

Subtracting the Eq. (2.8) from the Eq. (2.7) , we obtain

$$0 = \left(d_1 - \overline{d_1}\right)\alpha_1 + \left(d_2 - \overline{d_2}\right)\alpha_2 + \left(d_3 - \overline{d_3}\right)\alpha_3 + \cdots + \left(d_n - \overline{d_n}\right)\alpha_n .$$

Considering the linear independence of the vectors, $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$ we can say that $d_i = \overline{d_i}$. As an important consequence of this property, we can say that every vector in V_n has a unique combination of the base vectors.

Definition 2.1.6 (Change of Basis):

In a linear vector space V_n of n dimensions, there can be many sets of basis. Each set of basis has as expression in terms of another set of basis. As an example let $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$ and $\beta_1, \beta_2, \beta_3, \dots, \beta_n$ be two sets of basis in V_n. From this property, it can be written that

$$\alpha_i = \sum_{j=1}^{n} c_{ij}\beta_j , \qquad i = 1, 2, \dots, n \tag{2.9}$$

$$\beta_i = \sum_{j=1}^{n} d_{ij}\alpha_j , \qquad i = 1, 2, \dots, n . \tag{2.10}$$

Definition 2.1.7 (Linear Operators):

A function satisfying the following condition is said to be a linear operator, such that

$$L\left(c_1\alpha_1 + c_2\alpha_2\right) = c_1 L\left(\alpha_1\right) + c_2 L\left(\alpha_2\right) \qquad \alpha_1, \alpha_2 \in V_n \quad \text{and} \quad c_1, c_2 \in F . \tag{2.11}$$

As an example, let us show that the Eq. (2.1) is a linear operator

$$\left.\frac{d\left(c_1\alpha_1 + c_2\alpha_2\right)}{dx}\right|_{x_i} = \sum_{j=1}^{N} a_{ij}\left[c_1 f_1\left(x_j\right) + c_2 f_2\left(x_j\right)\right],$$

$$= \sum_{j=1}^{N} a_{ij}c_1 f_1\left(x_j\right) + \sum_{j=1}^{N} a_{ij}c_2 f_2\left(x_j\right),$$

$$=c_1\sum_{j=1}^{N}a_{ij}f_1\left(x_j\right)+c_2\sum_{j=1}^{N}a_{ij}f_2\left(x_j\right).$$

Theorem 2.1 In a n dimensional linear vector space in which V_n having the set $\alpha_1,\alpha_2,\alpha_3,\ldots,\alpha_n$ as a set of basis, if all the base vectors $\alpha_1,\alpha_2,\alpha_3,\ldots,\alpha_n$ satisfy a linear operator, then any vector in the space satisfies the linear operator as well.

Proof 2.1 Let L be a linear equation in V_n such that

$$L\left(\alpha_i\right)=0,\qquad i=0,1,\ldots,n \tag{2.12}$$

and α be an arbitrary vector in V_n. Using the Eqs. (2.7) and (2.12) we have

$$L\left(\alpha\right)=L\left(\sum_{i=1}^{n}d_i\alpha_i\right)=\sum_{i=1}^{n}L\left(d_i\alpha_i\right)=\sum_{i=1}^{n}d_iL\left(\alpha_i\right)=0\,.$$

Theorem 2.2 In an n dimensional linear vector space V_n if one basis satisfy a linear operator, another basis satisfy the linear operator as well.

Proof 2.2 Let $\alpha_1,\alpha_2,\ldots,\alpha_n$ and $\beta_1,\beta_2,\ldots,\beta_n$ be two basis in V_n. Assume that $\alpha_1,\alpha_2,\ldots..,\alpha_n$ satisfy the Eq. (2.12), then using Eqs. (2.10) and (2.11), one can write that

$$L\left(\beta_i\right)=L\left(\sum_{j=1}^{n}d_{ij}\alpha_j\right)=\sum_{j=1}^{n}L\left(d_{ij}\alpha_j\right)=\sum_{j=1}^{n}d_{ij}L\left(\alpha_j\right)=0\,.$$

2.2 HIGH ORDER POLYNOMIAL APPROXIMATION

The principles of high order polynomial approximation are from the Weierstrass first theorem. According to this theorem for every function $f\left(x\right)$ which is continuous and defined in the closed interval $\left[a,b\right]$, there exists a polynomial $P_n\left(x\right)$ of degree $n=n\left(\varepsilon\right)$ such that

$$\left|f\left(x\right)-P_n\left(x\right)\right|\le\varepsilon\,.$$

Using Weierstrass first theorem $f\left(x\right)$ can be expressed as follows

$$f\left(x\right)\approx P_N\left(x\right)=\sum_{k=0}^{N-1}c_kx^k\,. \tag{2.13}$$

Hence $1, x, x^2, \ldots, x^{N-1}$ is one of the basis of V_n, $P_N(x)$ is an N dimensional linear vector space in V_n.

Assume that there are N grid points in a closed interval $[a,b]$ such that $a = x_1, x_2, x_3, \ldots, x_N = b$ and the value of $f(x)$ at x_i is $f(x_i)$. Then the equation (2.13) can be expressed as

$$c_0 + c_1 x_1 + c_2 x_1^2 + \cdots + c_{N-1} x_1^{N-1} = f(x_1)$$

$$c_0 + c_1 x_2 + c_2 x_2^2 + \cdots + c_{N-1} x_2^{N-1} = f(x_2)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$c_0 + c_1 x_N + c_2 x_N^2 + \cdots + c_{N-1} x_N^{N-1} = f(x_N)$$

The determinant of the coefficients' matrix is in Vandermonde form. Thus it is not singular and the values of $c_0, c_1, \ldots, c_{N-1}$ can be computed. After computation of $c_0, c_1, \ldots, c_{N-1}$, $P_N(x)$ will be obtained. However, when N is a large number it is not easy to compute coefficients. In order to deal with the problem different test functions have been used such as Lagrange interpolated polynomial as follows

$$P_N(x) = \sum_{i=1}^{N} f(x_i) l_i(x) \tag{2.14}$$

where

$$l_i(x) = \frac{M(x)}{M'(x_i)(x - x_i)},$$

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

$$M'(x_i) = (x_i - x_1)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_N) = \prod_{k=1, k\neq i}^{N} (x_i - x_k) \tag{2.15}$$

where $M'(x_i)$ is the derivative of $M(x)$ at $x = x_i$. The function $l_k(x), \quad k = 1, 2, \ldots, N$ is the basis of the linear vector space such that

$$l_k(x_i) = \begin{cases} 1, & k = i, \\ 0, & k \neq i. \end{cases}$$

2.3 POLYNOMIAL BASED DIFFERENTIAL QUADRATURE METHOD

Polynomial Based Differential Quadrature Method (PDQ) is one of the main approaches in DQ. In this method, properties of linear vector spaces and functional approach have been used in order to obtain a higher order approximated polynomial to the solutions of ODEs and PDEs. The principle point of PDQ is computation of the weighted coefficients. The method was firstly dealt with Bellman and his associates. In the first approach of Bellman, there exists a system of algebraic equations. In his first method, the grid points can be chosen arbitrarily, however it is not useful when the number of grid points is more than 13 because of the difficulty of solving the system of algebraic equations. In the second approach of Bellman, Legendre polynomials are used as test functions. In his second approach, the roots of Legendre polynomials are chosen as grid points. Since there are some restrictions in the methods of Bellman, some different approaches have been improved in order to overcome these difficulties.

Quan and Chang used Lagrange interpolation polynomials as test function to solve the first and the second order derivatives. Shu introduced different formulas to compute the weighted coefficients of first, second and higher order derivatives. It has been seen that by using suitable base vectors and analyzing the linear vector spaces, there can be found different ways for computation of weighted coefficients.

In this subsection, the different approaches of PDQ that have been developed to compute weighted coefficients of first, second, and higher order derivatives will be introduced.

2.3.1 Computation of Weighted Coefficients for the First Order Derivative

In a closed interval $[a,b]$, let us define a function $f(x)$ which is continuous in the domain. Bellman supposed that when the domain is divided into N grid points such that $a = x_1, x_2, x_3, ..., x_N = b$, the value of the first order derivative of $f(x)$ at x_i can be expressed as a linear weighted sum of the functional values of all the grid points as follows

$$f'(x_i) = \sum_{j=1}^{N} a_{ij} f(x_j), \qquad i = 1, 2, \ldots, N \tag{2.16}$$

where a_{ij} are called as weighted coefficients of the first order derivative. As stated before the significant point of PDQ is determination of the weighted coefficients a_{ij}.

There have been improved several approaches to determine weighted coefficients by using different base polynomials. Assume that $f(x)$ is the approached polynomial and $r_k(x)$ is the basis polynomials. Then $f(x)$ can be expressed by linear sum of the basis polynomials as follows

$$f(x)=\sum_{k=1}^{N} c_k r_k(x) \tag{2.17}$$

where c_k's are constants. The first order derivative of $f(x)$ in respect of x can be stated as follows

$$f'(x)=\sum_{k=1}^{N} c_k r_k'(x) \tag{2.18}$$

When we substitute Eqs. (2.17) and (2.18) into (2.16), the following will be obtained

$$\sum_{k=1}^{N} c_k r_k'(x_i)=\sum_{j=1}^{N} a_{ij}\left[\sum_{k=1}^{N} c_k r_k(x_j)\right], \quad i=1,2,\ldots,N,$$

$$\sum_{k=1}^{N} c_k r_k'(x_i)=\sum_{j=1}^{N}\sum_{k=1}^{N} a_{ij} c_k r_k(x_j), \quad i=1,2,\ldots,N,$$

$$\sum_{k=1}^{N} c_k r_k'(x_i)=\sum_{k=1}^{N} c_k \sum_{j=1}^{N} a_{ij} r_k(x_j), \quad i=1,2,\ldots,N.$$

By simplifying, c_k's it can be obtained as follows

$$r_k'(x_i)=\sum_{j=1}^{N} a_{ij} r_k(x_j) \tag{2.19}$$

where a_{ij} are the weighted coefficients to be computed. Because of Eq. (2.19), different approached polynomials can be obtained by using different basis polynomials.

2.3.1.1 The First Approach of Bellman

In the first approach of Bellman, the test functions are chosen as follows

$$r_k(x)=x^k, \quad k=0,1,2,\ldots,N-1$$

When N number of grid points $(x_1,x_2,x_3,\ldots,x_N)$ applied to N number of test functions, there will be obtained a system of $N\times N$ number of equations for the weighted coefficients $a_{ij}, \quad i,j=1,2,\ldots,N$.

When $r_0(x)=1$, then $\frac{dr_0(x)}{dx}=0$. Applying Eq. (2.16) the following can be obtained

$$\left.\frac{dr_0(x)}{dx}\right|_{x_i}=\sum_{j=1}^{N}a_{ij}r_0(x_j), \quad i=1,2,3,\ldots,N\ ,$$

$$0=\sum_{j=1}^{N}a_{ij}=a_{i1}+a_{i2}+a_{i3}+\cdots+a_{iN}, \quad i=1,2,3,\ldots,N\ . \tag{2.20}$$

When $r_1(x)=x$, then $\frac{dr_1(x)}{dx}=1$. Applying Eq. (2.16) the following can be obtained

$$1=\sum_{j=1}^{N}a_{ij}x_j=x_1a_{i1}+x_2a_{i2}+x_3a_{i3}+\cdots+x_Na_{iN}, \quad i=1,2,\ldots,N\ . \tag{2.21}$$

When $r_k(x)=x^k$ for $k\geq 2$ then $\frac{dr_k(x)}{dx}=kx^{k-1}$. Applying Eq. (2.16) the following can be obtained

$$kx^{k-1}=\sum_{j=1}^{N}a_{ij}x_j^k, \quad i=1,2,\ldots,N\ . \tag{2.22}$$

Combining Eqs. (2.20), (2.21) and (2.22) the following system of equations can be written

$$a_{i1}+a_{i2}+a_{i3}+\cdots+a_{iN}=0$$

$$x_1a_{i1}+x_2a_{i2}+x_3a_{i3}+\cdots+x_Na_{iN}=1$$

$$x_1^2a_{i1}+x_2^2a_{i2}+x_3^2a_{i3}+\cdots+x_N^2a_{iN}=2x_i$$

$$x_1^3a_{i1}+x_2^3a_{i2}+x_3^3a_{i3}+\cdots+x_N^3a_{iN}=3x_i^2$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$x_1^{N-1}a_{i1}+x_2^{N-1}a_{i2}+x_3^{N-1}a_{i3}+\cdots+x_N^{N-1}a_{iN}=(N-1)x_i^{N-2}$$

Let us show the system in the matrix form as

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & \cdots & x_N \\ x_1^2 & x_2^2 & x_3^2 & \cdots & \cdots & x_N^2 \\ x_1^3 & x_2^3 & x_3^3 & \cdots & \cdots & x_N^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{N-1} & x_2^{N-1} & x_3^{N-1} & \cdots & \cdots & x_N^{N-1} \end{pmatrix}\begin{pmatrix} a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ \vdots \\ a_{iN} \end{pmatrix}=\begin{pmatrix} 0 \\ 1 \\ 2x_i \\ 3x_i^2 \\ \vdots \\ (N-1)x_i^{N-2} \end{pmatrix}$$

The $N \times N$ matrix in the system is in Vandermonde form. Therefore, its inverse exists and the weighted coefficients a_{ij} can be computed. Increasing the number of grid points N causes some difficulties when finding the inverse of the Vandermonde matrix. Thus, this method is useful when N is less than or equal to 13.

2.3.1.2 The Second Approach of Bellman

In the second approach of Bellman, the test functions are chosen as follows

$$r_k(x) = \frac{K_N(x)}{(x - x_k)K'_N(x)}, \qquad k = 1, 2, \ldots, N \tag{2.23}$$

where the functions $K_N(x)$ are called as shifted Legendre polynomials. In this method grid points are chosen from the roots of $K_N(x)$. The relation between Legendre polynomials $L_N(x)$ and $K_N(x)$ is given by $K_N(x) = L_N(1 - 2x)$. The definition of the Legendre polynomials $L_N(x)$ is given as

$$L_0(x) = 1,$$

$$L_1(x) = x,$$

$$L_{k+1}(x) = \frac{2k+1}{k+1} x L_k(x) - \frac{k}{k+1} L_{k-1}(x), \qquad k = 1, 2, \ldots, N-1.$$

The first order derivative of Eq. (2.23) is

$$r_k{}'(x) = \frac{K'_N(x)(x - x_k) - K_N(x)}{(x - x_k)^2 K'_N(x)}, \qquad k = 1, 2, \ldots, N. \tag{2.24}$$

Applying Eqs. (2.23) and (2.24) in Eq. (2.16), the following can be obtained

$$\frac{K'_N(x_i)(x_i - x_k) - K_N(x_i)}{(x_i - x_k)^2 K'_N(x_k)} = \sum_{j=1}^{N} a_{ij} \frac{K_N(x_j)}{(x_j - x_k)K'_N(x_k)},$$

$$\frac{K'_N(x_i)(x_i - x_k) - K_N(x_i)}{(x_i - x_k)^2} = \sum_{j=1}^{N} a_{ij} \frac{K_N(x_j)}{(x_j - x_k)}. \tag{2.25}$$

When $k = j \neq i$ there exists a $0/0$ uncertainty on the right side. In order to remove this uncertainty, the rule of L'Hospital can be applied as follows

$$\lim_{x \to x_k} \frac{K_N(x)}{(x - x_k)} = \lim_{x \to x_k} \frac{K'_N(x)}{1} = K'_N(x_k) = K'_N(x_j). \tag{2.26}$$

After applying, the result of (2.26) in (2.23) the following can be obtained

$$\frac{{K_N}'(x_i)(x_i - x_k) - K_N(x_i)}{(x_i - x_k)^2} = \sum_{j=1}^{N} a_{ij} K_N'(x_j). \tag{2.27}$$

Hence x_i are the roots of shifted Legendre polynomials, $K_N(x_i) = 0$. Saying that $k = j$ the Eq. (2.27) can be reduced as

$$\frac{{K_N}'(x_i)}{(x_i - x_j)} = \sum_{j=1}^{N} a_{ij} K_N'(x_j). \tag{2.28}$$

Finally, for $i \neq j$ the weighted coefficients a_{ij} can be defined as

$$a_{ij} = \frac{{K_N}'(x_i)}{(x_i - x_j) K_N'(x_j)}.$$

When $k = j = i$ there exists a $0/0$ uncertainty on the left side of (2.27). In order to remove this uncertainty, the rule of L'Hospital can be applied as follows

$$\begin{aligned}
\lim_{x \to x_k} \frac{K_N'(x)(x - x_k) - K_N(x)}{(x - x_k)^2} &= \lim_{x \to x_k} \frac{K_N''(x)(x - x_k) + K_N'(x) - K_N'(x)}{2(x - x_k)}, \\
&= \lim_{x \to x_k} \frac{K_N''(x)(x - x_k) + K_N'(x) - K_N'(x)}{2(x - x_k)}, \\
&= \frac{K_N''(x_k)}{2} = \frac{K_N''(x_i)}{2}.
\end{aligned} \tag{2.29}$$

Then, by substituting (2.29) to the left side of (2.27) the following can be obtained

$$\frac{K_N''(x_i)}{2} = \sum_{i=1}^{N} a_{ij} K_N'(x_i).$$

Thus, the weighted coefficients a_{ij} for $i = j$ can be stated as follows

$$a_{ii} = \frac{K_N''(x_i)}{2K_N'(x_i)}. \tag{2.30}$$

In order to define a_{ij} for $i = j$ in a different way by using the Legendre differential equation that is stated as

$$(1 - t^2) L_N''(t) - 2tL_N'(t) + N(N+1) L_N(t) = 0, \tag{2.31}$$

let us substitute t by $1 - 2x$ in (2.31), then

$$\left(1-(1-2x)^2\right)L_N''(1-2x)-2(1-2x)L_N'(1-2x)+N(N+1)L_N(1-2x)=0$$

$$\left(4-4x^2\right)L_N''(1-2x)-2(1-2x)L_N'(1-2x)+N(N+1)L_N(1-2x)=0 \qquad (2.32)$$

will be obtained. Since $K_N'(x)=-2L_N'(1-2x)$ and $K_N''(x)=4L_N''(1-2x)$ from the determination of shifted Legendre polynomial, (2.32) can be expressed as

$$\left(x-x^2\right)K_N''(x)+(1-2x)K_N'(x)+N(N+1)K_N(x)=0 \qquad (2.33)$$

Substituting x by the roots x_i's of $K_N(x)$ in (2.33), it can be obtained the following

$$\left(x_i-x_i^2\right)K_N''(x_i)+(1-2x_i)K_N'(x_i)+N(N+1)K_N(x_i)=0$$

$$\left(x_i-x_i^2\right)K_N''(x_i)=-(1-2x_i)K_N'(x_i)$$

$$\frac{K_N''(x_i)}{K_N'(x_i)}=\frac{-(1-2x_i)}{\left(x_i-x_i^2\right)}. \qquad (2.34)$$

Finally, considering Eqs. (2.30) and (2.34) together the weighted coefficients for $i=j$ can be stated as follows

$$a_{ii}=\frac{1-2x_i}{2x_i(x_i-1)}$$

Due to restriction of choosing the grid points from the roots of the shifted Legendre polynomial in the second approach of Bellman, his first approach is more useful. Thus comparing to the second method, first method is more preferable in practical applications.

2.3.1.3 The Approach of Quan and Chang

In order to improve the methods of Bellman, there have been many attempts for computation of weighted coefficients a_{ij}. One of the most practicable one was improved by Quan and Chang, in which Lagrange interpolation polynomials $M(x)$ defined in (2.15) were used as test functions like

$$r_k(x)=\frac{M(x)}{(x-x_k)M'(x_k)}. \qquad (2.35)$$

The first derivative of (2.35) is

$$r_k'(x) = \frac{M'(x)(x-x_k) - M(x)}{(x-x_k)^2 M'(x_k)} \tag{2.36}$$

and applying Eqs. (2.35) along with (2.36) in Eq. (2.16), the following can be obtained

$$\frac{M'(x_i)(x_i - x_k) - M(x_i)}{(x_i - x_k)^2 M'(x_k)} = \sum_{j=1}^{N} a_{ij} \frac{M(x_j)}{(x_j - x_k) M'(x_k)},$$

$$\frac{M'(x_i)(x_i - x_k) - M(x_i)}{(x_i - x_k)^2} = \sum_{j=1}^{N} a_{ij} \frac{M(x_j)}{(x_j - x_k)}. \tag{2.37}$$

When $k = j \neq i$ there exists a $0/0$ uncertainty on the right side. In order to remove this uncertainty, the rule of L'Hospital can be applied as

$$\lim_{x \to x_k} \frac{M(x)}{(x - x_k)} = M'(x_k) = M'(x_j). \tag{2.38}$$

Then, the Eq. (2.37) can be expressed as

$$\frac{M'(x_i)(x_i - x_k) - M(x_i)}{(x_i - x_k)^2} = \sum_{j=1}^{N} a_{ij} M'(x_j). \tag{2.39}$$

Since the values x_i are the roots of Lagrange interpolation polynomial, for $k = j$ the Eq. (2.39) can be reduced as

$$\frac{M'(x_i)}{(x_i - x_j)} = \sum_{j=1}^{N} a_{ij} M'(x_j).$$

Finally, for $i \neq j$ the weighted coefficients a_{ij} can be defined as

$$a_{ij} = \frac{M'(x_i)}{(x_i - x_j) M'(x_j)}. \tag{2.40}$$

When $k = j = i$ there exists a $0/0$ uncertainty on the left side of (2.39). In order to remove this uncertainty, the rule of L'Hospital can be applied as follows

$$\lim_{x \to x_k} \frac{M'(x)(x - x_k) - M(x)}{(x - x_k)^2} = \lim_{x \to x_k} \frac{M''(x)(x - x_k) - M'(x)}{2(x - x_k)},$$

$$= \lim_{x \to x_k} \frac{M''(x)(x - x_k) + M'(x) - M'(x)}{2(x - x_k)},$$

$$= \frac{M''(x_k)}{2} = \frac{M''(x_i)}{2}. \tag{2.41}$$

Then by substituting (2.41) to the left side of (2.39), the weighted coefficients for $i = j$ can be obtained in the form of

$$a_{ii} = \frac{M''(x_i)}{2M'(x_i)}. \tag{2.42}$$

Using the grid points in Eqs (2.40) and (2.42), the weighted coefficients can be stated as

$$a_{ij} = \frac{1}{x_j - x_i} \prod_{k=1,\, k \neq i}^{N} \frac{x_i - x_k}{x_j - x_k}, \qquad i \neq j\,, \tag{2.43}$$

$$a_{ii} = \sum_{k=1,\, k \neq i}^{N} \frac{1}{x_i - x_k}. \tag{2.44}$$

Since there is no restriction in selection of grid points, they can be chosen arbitrarily in this approach.

2.3.1.4 The General Approach of Shu

The general approach of Shu was originated by the approaches of Bellman and the approach of Quan and Chang. Since different approaches used for computation of weighted coefficients and the different approaches gave the same result, he considered to obtain an alternative way to compute the weighted coefficients by practicing on linear vector spaces and changing basis polynomials.

As stated before, the solution of an ordinary differential equation can be expressed as $N-1$ order approached polynomial such that

$$f(x) = \sum_{k=1}^{N} c_k x^{k-1}$$

where c_k's are constants. There are many basis vectors, which are also polynomials. For this reason, it can be called basis vectors as basis polynomials. Some of the important basis polynomials, which are used in the approaches shown before, can be given as

$$r_k(x) = x^{k-1}, \qquad k = 1, 2, \ldots, N\,, \tag{2.45}$$

$$r_k(x) = \frac{K_N(x)}{(x - x_k) K_N'(x)}, \qquad k = 1, 2, \ldots, N\,, \tag{2.46}$$

$$r_k(x) = \frac{M(x)}{(x - x_k)M'(x)}, \quad k = 1, 2, \ldots, N, \tag{2.47}$$

$$r_1(x) = 1,$$

$$r_k(x) = (x - x_{k-1})r_{k-1}(x), \quad k = 1, 2, \ldots, N. \tag{2.48}$$

The polynomials given in Eq. (2.45) are similar to the test functions used in the first approach of Bellman and the polynomials given in Eq. (2.46) are similar to the test functions used in the second approach of Bellman. Polynomials in Eqs. (2.46) and (2.47) are obtained from Lagrange interpolation polynomials. The difference is that in Eq. (2.46) the roots of the polynomials are chosen from the roots of shifted Legendre polynomials, however in Eq. (2.47) the roots of the polynomials can be chosen arbitrarily. Therefore, it can be said that the Eq. (2.46) is a special case of the Eq. (2.47).

According to the Theorem 2.2, one can state that all the sets of basis polynomials will give the same weighted coefficients. For this reason, by using different test functions, the same weighted coefficients will be obtained. As a result, as long as V_N contains different sets of basis polynomials, there will be different ways for computation of weighted coefficients.

In the general approach of Shu, two sets of basis polynomials will be used. Firstly, the sets of polynomials given in (2.47) will be applied as follows

Let us define $M(x)$ such that

$$M(x) = N(x, x_k)(x - x_k), \quad k = 1, 2, \ldots, N \tag{2.49}$$

where

$$N(x_i, x_j) = M'(x_i)\delta_{ij} \tag{2.50}$$

and δ_{ij} is named as Kronecker delta that is defined as

$$\delta_{ij} = \begin{cases} 0, & i \neq j, \\ 1, & i = j. \end{cases}$$

By substituting (2.49) in (2.47) the following will be obtained

$$r_k(x) = \frac{M(x)}{(x - x_k)M'(x_k)} = \frac{N(x, x_k)(x - x_k)}{(x - x_k)M'(x_k)} = \frac{N(x, x_k)}{M'(x_k)}. \tag{2.51}$$

The first order derivative of $r_k(x)$ in Eq. (2.51) is

$$r_k'(x) = \frac{N'(x, x_k)}{M'(x_k)}. \tag{2.52}$$

Using Eqs. (2.51) and (2.52) in Eq. (2.16) the following can be obtained

$$\frac{N'(x_i, x_k)}{M'(x_k)} = \sum_{j=1}^{N} a_{ij} \frac{N(x_j, x_k)}{M'(x_k)}, \qquad i, k = 1, 2, \ldots, N$$

where

$$N'(x_i, x_k) = \sum_{j=1}^{N} a_{ij} N(x_j, x_k), \qquad i, k = 1, 2, \ldots, N. \tag{2.53}$$

When $k = j \neq i$ in Eq. (2.53), using the Eq. (2.50) the weighted coefficients can be expressed as

$$a_{ij} = \frac{N'(x_i, x_j)}{N(x_i, x_j)} = \frac{N'(x_i, x_j)}{M'(x_j)\delta_{jj}} = \frac{N'(x_i, x_j)}{M'(x_j)}. \tag{2.54}$$

In Eq. (2.54), the value of $M'(x_j)$ can be computed using Eq. (2.15). For computation of $N'(x_i, x_j)$ let us use Eq. (2.49) as follows

$$M(x) = N(x, x_k)(x - x_k),$$

$$M'(x) = N'(x, x_k)(x - x_k) + N(x, x_k),$$

$$M''(x) = N''(x, x_k)(x - x_k) + 2N'(x, x_k),$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$M^{(m)}(x) = N^{(m)}(x, x_k)(x - x_k) + mN^{(m-1)}(x, x_k). \tag{2.55}$$

to obtain the recurrence formula in (2.55). Using the formula of $M'(x)$ above, the value of $N'(x_i, x_j)$ can be obtained as

$$M'(x_i) = N'(x_i, x_j)(x_i - x_j) + N(x_i, x_j),$$

$$N'(x_i, x_j) = \frac{M'(x_i) - M'(x_i)\delta_{ij}}{x_i - x_j},$$

$$N'(x_i, x_j) = \frac{M'(x_i)(1 - \delta_{ij})}{x_i - x_j},$$

$$N'\left(x_i, x_j\right) = \frac{M'\left(x_i\right)}{x_i - x_j}, \quad i \neq j\,. \tag{2.56}$$

In a similar way, using the formula of $M''(x)$ like

$$M''\left(x_i\right) = N''\left(x_i, x_i\right)\left(x_i - x_i\right) + 2N'\left(x_i, x_i\right)$$

it can be obtained as

$$N'\left(x_i, x_i\right) = \frac{M''\left(x_i\right)}{2}\,. \tag{2.57}$$

Substituting Eqs. (2.56) and (2.57) in Eq. (2.54) the weighted coefficients will be stated as follows

$$a_{ij} = \frac{N'\left(x_i, x_j\right)}{M'\left(x_j\right)} = \frac{M'\left(x_i\right)}{\left(x_i - x_j\right)M'\left(x_j\right)}, \quad i \neq j\,, \tag{2.58}$$

$$a_{ii} = \frac{N'\left(x_i, x_i\right)}{M'\left(x_i\right)} = \frac{M''\left(x_i\right)}{2M'\left(x_i\right)}\,. \tag{2.59}$$

Using Eq. (2.15) it is easy to compute the value of $M'\left(x_i\right)$ in equations above; however, the value of $M''\left(x_i\right)$ in Eq. (2.59) is difficult to compute. In order to overcome this problem the set of basis polynomials given in Eq. (2.45) will be used as the second set of basis polynomials in this approach. As stated before, if one of the sets of basis polynomials satisfies a linear operator then the other sets of basis polynomials will satisfy the linear operator in linear vector space V_N. Accordingly when $k = 1$, the weighted coefficients will satisfy Eq. (2.60) which is obtained from the basis polynomials of Eq. (2.45). Therefore, when $i = j$ the weighted coefficients a_{ii} can be computed as

$$\sum_{j=1}^{N} a_{ij} = 0, \quad i = 1, 2, \ldots, N$$

or

$$a_{ii} = -\sum_{j=1,\ j\neq i}^{N} a_{ij}, \quad i = 1, 2, \ldots, N\,. \tag{2.60}$$

In the general approach of Shu, the Eqs. (2.58) and (2.60) are used to compute the weighted coefficients a_{ij}. These formulas are obtained from two types of basis polynomials given in Eqs. (2.45) and (2.47).

2.3.2 Computation of Weighted Coefficients for the Second Order Derivative

The second order derivative of a function can be stated as a linear sum of the values of grid points as follows

$$f''(x_i) = \sum_{j=1}^{N} w_{ij}^{(2)} f(x_j), \qquad i = 1, 2, \ldots, N \tag{2.61}$$

where $w_{ij}^{(2)}$ are the weighted coefficients of the second order derivative. The Eq. (2.61) is similar to the first order derivative of $f(x)$. The difference is that the values of the weighted coefficients are different. In second order derivative, the significant point is computing weighted coefficients $w_{ij}^{(2)}$ likewise the first order derivative.

2.3.2.1 The Approach of Quan and Chang

Quan and Chang used Lagrange interpolation polynomials as test functions in order to compute weighted coefficients of second order derivatives. The following formulas are obtained in this approach

$$w_{ij}^{(2)} = \frac{2}{x_j - x_i} \left(\prod_{k=1,\, k \neq i, j}^{N} \frac{x_i - x_k}{x_j - x_k} \right) \left(\prod_{l=1,\, l \neq i, j}^{N} \frac{2}{x_i - x_l} \right), \qquad i \neq j, \tag{2.62}$$

$$w_{ii}^{(2)} = 2 \sum_{k=1,\, k \neq i}^{N-1} \left[\frac{1}{x_i - x_k} \left(\sum_{l=k+1,\, l \neq i}^{N} \frac{1}{x_i - x_l} \right) \right]. \tag{2.63}$$

2.3.2.2 The General Approach of Shu

Shu used the two sets of basis polynomials in Eqs. (2.45) and (2.47) to compute the weighted coefficients of second order derivatives. According to this approach by substituting Eq. (2.50) in Eq. (2.60), the following will be obtained

$$w_{ij}^{(2)} = \frac{N''(x_i, x_j)}{M'(x_j)}. \tag{2.64}$$

Using the recurrence formula in (2.55), it can be computed the followings

$$M''(x_i) = N''(x_i, x_j)(x_i - x_j) + 2N'(x_i, x_j),$$

$$N'(x_i, x_j) = \frac{M''(x_i) - 2N'(x_i, x_j)}{(x_i - x_j)}, \quad i \neq j, \tag{2.65}$$

$$M'''(x_i) = N'''(x_i, x_i)(x_i - x_i) + 3N''(x_i, x_i),$$

$$N''(x_i, x_i) = \frac{M'''(x_i)}{3}. \tag{2.66}$$

By substituting Eqs. (2.65) and (2.66) in Eq. (2.64), it can be written that

$$w_{ij}^{(2)} = \frac{N''(x_i, x_j)}{M'(x_j)} = \frac{M''(x_i) - 2N'(x_i, x_j)}{(x_i - x_j)M'(x_j)}, \quad i \neq j, \tag{2.67}$$

$$w_{ii}^{(2)} = \frac{N''(x_i, x_i)}{M'(x_i)} = \frac{M'''(x_i)}{3M'(x_i)}. \tag{2.68}$$

Using Eqs. (2.58) and (2.59) in Eqs. (2.67) and (2.68), for $i \neq j$ it can be obtained as

$$w_{ij}^{(2)} = \frac{M''(x_i)}{(x_i - x_j)M'(x_j)} - \frac{2N'(x_i, x_j)}{(x_i - x_j)M'(x_j)},$$

$$w_{ij}^{(2)} = \frac{2M'(x_i)M''(x_i)}{2M'(x_i)(x_i - x_j)M'(x_j)} - \frac{2}{(x_i - x_j)M'(x_j)}\frac{M'(x_i)}{(x_i - x_j)},$$

$$w_{ij}^{(2)} = 2\frac{2M''(x_i)}{2M'(x_i)}\frac{M'(x_i)}{(x_i - x_j)M'(x_j)} - \frac{2}{(x_i - x_j)}\frac{M'(x_i)}{M'(x_j)(x_i - x_j)},$$

$$w_{ij}^{(2)} = 2a_{ii}\frac{M'(x_i)}{(x_i - x_j)M'(x_j)} - \frac{2}{(x_i - x_j)}\frac{M'(x_i)}{(x_i - x_j)M'(x_j)},$$

$$w_{ij}^{(2)} = 2a_{ii}a_{ij} - \frac{2}{(x_i - x_j)}a_{ij},$$

$$w_{ij}^{(2)} = 2a_{ij}\left(a_{ii} - \frac{1}{(x_i - x_j)}\right), \quad i \neq j. \tag{2.69}$$

Using Eq. (2.69) the values of weighted coefficients can be computed for $i \neq j$. However, for $i = j$ in Eq. (2.68) it is difficult to compute the value of $M'''(x_i)$. In order to remove this difficulty the set of basis polynomials in Eq. (2.45) will be used for

$k = 1$. According to the properties of linear vector spaces, it can be stated that the weighted coefficients $w_{ij}^{(2)}$ will satisfy the following relations:

$$\sum_{j=1}^{N} w_{ij}^{(2)} = 0\,,$$

$$w_{ii}^{(2)} = -\sum_{j=1,\, j\neq i}^{N} w_{ij}^{(2)}\,. \tag{2.70}$$

As a result, in order to compute the weighted coefficients of second order derivatives with general approach of Shu, the Eqs. (2.69) and (2.70) will be applied.

2.3.3 Computation of Weighted Coefficients for Higher Order Derivatives

In this section, computing weighted coefficients for higher order derivatives will be explained using Shu's recurrence formula and matrix multiplication approach respectively.

2.3.3.1 Shu Recurrence Formula

The following two linear operators are used in order to indicate higher order derivatives for $i = 1, 2, \ldots, N$ and $m = 2, 3, \ldots, N-1$.

$$f^{(m-1)}(x_i) = \sum_{j=1}^{N} w_{ij}^{(m-1)} f\left(x_j\right), \tag{2.71}$$

$$f^{(m)}(x_i) = \sum_{j=1}^{N} w_{ij}^{(m)} f\left(x_j\right) \tag{2.72}$$

where the weighted coefficients $w_{ij}^{(m-1)}$ represents $(m-1)$'st derivative and $w_{ij}^{(m)}$ represents (m)'th derivative. Two sets of basis polynomials are used in this approach. One of these sets is the basis polynomials $r_k(x)$ which are given in Eq. (2.51). When these basis polynomials are replaced in Eq. (2.71) together with their $(m-1)$'st and (m)'th order derivatives respectively, it will be obtained the following

$$w_{ij}^{(m-1)} = \frac{N^{(m-1)}\left(x_i, x_j\right)}{M'\left(x_j\right)}, \tag{2.73}$$

$$w_{ij}^{(m)} = \frac{N^{(m)}\left(x_i, x_j\right)}{M'\left(x_j\right)}.$$
(2.74)

The Eq. (2.73) can be expressed as

$$N^{(m-1)}\left(x_i, x_j\right) = w_{ij}^{(m-1)} M'\left(x_j\right) \tag{2.75}$$

for all i and j. Using the recurrence relation in Eq. (2.55), the following is obtained

$$M^{(m)}\left(x_i\right) = N^{(m)}\left(x_i, x_i\right)\left(x_i - x_i\right) + mN^{(m-1)}\left(x_i, x_i\right),$$

$$N^{(m-1)}\left(x_i, x_i\right) = \frac{M^{(m)}\left(x_i\right)}{m}. \tag{2.76}$$

In a similar way, the following relation can be acquired

$$M^{(m)}\left(x_i\right) = N^{(m)}\left(x_i, x_j\right)\left(x_i - x_j\right) + mN^{(m-1)}\left(x_i, x_j\right),$$

$$N^{(m)}\left(x_i, x_j\right) = \frac{M^{(m)}\left(x_i\right) - mN^{(m-1)}\left(x_i, x_j\right)}{\left(x_i - x_j\right)}, \quad i \neq j. \tag{2.77}$$

Besides, for $i = j$, it is obtained as

$$M^{(m+1)}\left(x_i\right) = N^{(m+1)}\left(x_i, x_i\right)\left(x_i - x_i\right) + \left(m+1\right) N^{(m)}\left(x_i, x_i\right),$$

$$N^{(m)}\left(x_i, x_i\right) = \frac{M^{(m+1)}\left(x_i\right)}{m+1}. \tag{2.78}$$

By substituting Eq. (2.76) in Eq. (2.77), it is written as

$$\begin{aligned} N^{(m)}\left(x_i, x_j\right) &= \frac{M^{(m)}\left(x_i\right) - mN^{(m-1)}\left(x_i, x_j\right)}{\left(x_i - x_j\right)}, \\ &= \frac{mN^{(m-1)}\left(x_i, x_i\right) - mN^{(m-1)}\left(x_i, x_j\right)}{\left(x_i - x_j\right)}, \\ &= \frac{m\left[N^{(m-1)}\left(x_i, x_i\right) - N^{(m-1)}\left(x_i, x_j\right)\right]}{\left(x_i - x_j\right)}, \quad i \neq j \end{aligned} \tag{2.79}$$

and applying Eq. (2.75), it can be expressed as

$$N^{(m)}\left(x_i, x_j\right) = \frac{m\left[w_{ii}^{(m-1)} M'\left(x_i\right) - w_{ij}^{(m-1)} M'\left(x_j\right)\right]}{\left(x_i - x_j\right)}, \quad i \neq j \tag{2.80}$$

when Eqs. (2.58) and (2.80) are used in Eq. (2.74), the following will be obtained

$$w_{ij}^{(m)} = \frac{N^{(m)}\left(x_i, x_j\right)}{M'\left(x_j\right)} = \frac{m\left[w_{ii}^{(m-1)}M'\left(x_i\right) - w_{ij}^{(m-1)}M'\left(x_j\right)\right]}{\left(x_i - x_j\right)M'\left(x_j\right)},$$

$$= m\left[\frac{w_{ii}^{(m-1)}M'\left(x_i\right)}{\left(x_i - x_j\right)M'\left(x_j\right)} - \frac{w_{ij}^{(m-1)}M'\left(x_j\right)}{\left(x_i - x_j\right)M'\left(x_j\right)}\right],$$

$$w_{ij}^{(m)} = m\left(a_{ij}w_{ii}^{(m-1)} - \frac{w_{ij}^{(m-1)}}{x_i - x_j}\right), \quad i, j = 1, 2, \ldots, N, \quad m = 2, 3, \ldots, N-1. \tag{2.81}$$

By using Eq. (2.78) in Eq. (2.74) it will be acquired as follows

$$w_{ii}^{(m)} = \frac{N^{(m)}\left(x_i, x_i\right)}{M'\left(x_i\right)} = \frac{M^{(m+1)}\left(x_i\right)}{\left(m+1\right)M'\left(x_i\right)}, \quad i, j = 1, 2, \ldots, N, \quad m = 2, 3, \ldots, N-1. \tag{2.82}$$

Similar to the previous approaches when $i \neq j$, the weighted coefficients $w_{ij}^{(m)}$ are easy to compute. However, for the weighted coefficients $w_{ii}^{(m)}$ it is not easy to compute the values because of including $\left(m+1\right)$st derivative of the function $M\left(x\right)$ in Eq. (2.82). In order to solve this problem, the properties of linear vector space will be used. The system of equations $w_{ij}^{(m)}$, which are obtained from the basis polynomials in Eq. (2.47), can be found using the basis polynomials in Eq. (2.45) as well. When the basis polynomials (2.45) are used, for $k = 1$, $w_{ij}^{(m)}$ will satisfy the following relation:

$$\sum_{j=1}^{N} w_{ij}^{(m)} = 0,$$

$$w_{ii}^{(m)} = -\sum_{j=1,\ j\neq i}^{N} w_{ij}^{(m)}. \tag{2.83}$$

According to the Shu recurrence formula, it can be stated that the Eqs. (2.81) and (2.83) are two formulas to compute weighted coefficients for higher order derivatives.

2.3.3.2 Approach of Matrix Multiplication

The differential operator can be stated as

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) \tag{2.84}$$

when DQ formula applied twice to Eq. (2.84), it will be obtained as follows

$$f''(x_i)=\sum_{k=1}^{N}a_{ik}f'(x_k)=\sum_{k=1}^{N}a_{ik}\sum_{j=1}^{N}a_{kj}f\left(x_j\right),$$

$$=\sum_{j=1}^{N}\left[\sum_{k=1}^{N}a_{ik}a_{kj}\right]f\left(x_j\right),\qquad i=1,2,\ldots,N\,. \tag{2.85}$$

Checking against Eq. (2.85) and second order DQ formula

$$f''(x_i)=\sum_{j=1}^{N}w_{ij}^{(2)}f\left(x_j\right),\qquad i=1,2,\ldots,N$$

it can be acquired that

$$w_{ij}^{(2)}=\sum_{k=1}^{N}a_{ik}a_{kj} \tag{2.86}$$

where the matrices are defined as

$$\left[A^{(1)}\right]=\begin{pmatrix}a_{11}&a_{12}&\cdots&a_{1N}\\a_{21}&a_{22}&\cdots&a_{2N}\\\vdots&\vdots&\ddots&\vdots\\a_{N1}&a_{N2}&\cdots&a_{NN}\end{pmatrix},\qquad\left[A^{(2)}\right]=\begin{pmatrix}w_{11}^{(2)}&w_{12}^{(2)}&\cdots&w_{1N}^{(2)}\\w_{21}^{(2)}&w_{22}^{(2)}&\cdots&w_{2N}^{(2)}\\\vdots&\vdots&\ddots&\vdots\\w_{N1}^{(2)}&w_{N2}^{(2)}&\cdots&w_{NN}^{(2)}\end{pmatrix}.$$

By using above matrices, the Eq. (2.86) can be expressed as

$$\left[A^{(2)}\right]=\left[A^{(1)}\right]\left[A^{(1)}\right].$$

Eventually, it is observed that the weighted coefficients of the second order derivative can be computed by multiplying the matrices of the weighted coefficients of the first order derivative. In order to generalize the formula, let us write the differential derivative of (m)'th order derivative as

$$\frac{\partial^{(m)}f}{\partial x^{(m)}}=\frac{\partial}{\partial x}\left(\frac{\partial^{(m-1)}f}{\partial x^{(m-1)}}\right)=\frac{\partial^{(m-1)}}{\partial x^{(m-1)}}\left(\frac{\partial f}{\partial x}\right). \tag{2.87}$$

Assume that the weighted coefficients of $(m-1)$'st and (m)'th order derivatives are $\left[A^{(m-1)}\right]$ and $\left[A^{(m)}\right]$, respectively. By applying the DQ formula to Eq. (2.87), the following relation will be obtained for $m=2,3,\ldots,N-1$ such that

$$\left[A^{(m)}\right]=\left[A^{(1)}\right]\left[A^{(m-1)}\right]=\left[A^{(m-1)}\right]\left[A^{(1)}\right]. \tag{2.88}$$

2.4 SOLUTIONS OF ORDINARY DIFFERENTIAL EQUATIONS WITH POLYNOMIAL BASED DIFFERENTIAL QUADRATURE METHOD

In this section, first, second and higher order linear ordinary differential equations will be solved using PDQ. At each grid point, absolute error will be computed as well as computation of exact solutions and approximate solutions. Shu's approaches will be used to find the weighted coefficients. In calculations of algebraic systems, MATLAB is used and all the MATLAB programs are in Appendix A.

2.4.1 Types of Grid Distributions

Two types of grid distributions named as uniform grid distribution and Chebyshev-Gauss-Lobatto grid distribution are used in the applications of this study. According to the structure of the problem, they can be advantageous or disadvantageous.

2.4.1.1 Uniform Grid Distribution

In this type of distribution, the coordinates are chosen with constant interval such that

$$x_i = \frac{i-1}{N_x - 1}, \quad i = 1, 2, \ldots, N_x, \tag{2.89}$$

$$y_j = \frac{j-1}{N_y - 1}, \quad j = 1, 2, \ldots, N_y$$

where N_x and N_y are number of grid points in x and y coordinate directions. In uniform grid distribution, the coordinates of grid points are easy to compute and apply.

2.4.1.2 Chebyshev-Gauss-Lobatto Grid Distribution

In this grid distribution, the points are chosen more frequently towards the points of boundary conditions. The coordinates of grid points can be computed as follows

$$x_i = \frac{1 - \cos\left(\frac{i-1}{N_x - 1}\pi\right)}{2}, \quad i = 1, 2, \ldots, N_x, \tag{2.90}$$

$$y_j = \frac{1-\cos\left(\dfrac{j-1}{N_y-1}\pi\right)}{2}, \quad i = 1, 2, \ldots, N_y$$

where N_x and N_y are number of grid points in x and y coordinate directions. Compared to the uniform grid distribution, Chebyshev-Gauss-Lobatto grid distribution gives more convergent and reliable solutions.

2.4.2 Applications

In this subsection, the first, the second and higher order initial value problems and boundary value problems will be solved using PDQ method. The formulas obtained in Shu's approaches will be used to compute the weighted coefficients of first, second and higher order derivatives.

As a beginning, let us compute the weighted coefficients a_{ij} using the Eqs. (2.58) and (2.60). The uniform grid distribution will be applied as follows

$$x_i = \frac{i-1}{N-1}, \quad i = 1, 2, \ldots, N .$$

The weighted coefficients for $N = 5$ and $N = 6$ are obtained as

$$a_{ij} = \begin{pmatrix} -\frac{25}{3} & 16 & -12 & \frac{16}{3} & -1 \\ -1 & -\frac{10}{3} & 6 & -2 & \frac{1}{3} \\ \frac{1}{3} & -\frac{8}{3} & 0 & \frac{8}{3} & -\frac{1}{3} \\ -\frac{1}{3} & 2 & -6 & \frac{10}{3} & 1 \\ 1 & -\frac{16}{3} & 12 & -16 & \frac{25}{3} \end{pmatrix},$$

$$a_{ij} = \begin{pmatrix} -\frac{137}{12} & 25 & -25 & \frac{50}{3} & -\frac{25}{4} & 1 \\ -1 & -\frac{65}{12} & 10 & -5 & \frac{5}{3} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{5}{2} & -\frac{5}{3} & 5 & -\frac{5}{4} & \frac{1}{6} \\ -\frac{1}{6} & \frac{5}{4} & -5 & \frac{5}{3} & \frac{5}{2} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{5}{3} & 5 & -10 & \frac{65}{12} & 1 \\ -1 & \frac{25}{4} & -\frac{50}{3} & 25 & -25 & \frac{137}{12} \end{pmatrix},$$

respectively. If CGL grid point distribution is applied such that

$$x_i = \frac{1-\cos\left(\frac{i-1}{N-1}\pi\right)}{2}, \quad i = 1, 2, \ldots, N,$$

the weighted coefficients can be computed for $N = 5$ and $N = 6$, respectively, as

$$a_{ij} = \begin{pmatrix} -11 & \frac{1393}{102} & -4 & \frac{2786}{1189} & -1 \\ -\frac{1393}{408} & -\frac{1393}{985} & \frac{3363}{1189} & -\frac{1393}{985} & \frac{577}{985} \\ 1 & -\frac{3363}{1189} & 0 & \frac{3363}{1189} & -1 \\ -\frac{577}{985} & \frac{1393}{985} & -\frac{3363}{1189} & -\frac{1393}{985} & \frac{1393}{408} \\ 1 & -\frac{2786}{1189} & 4 & -\frac{1393}{102} & 11 \end{pmatrix},$$

$$a_{ij} = \begin{pmatrix} -17 & \frac{6765}{323} & -\frac{1974}{341} & \frac{987}{323} & -\frac{754}{341} & 1 \\ -\frac{1597}{305} & \frac{2255}{963} & 4 & -\frac{2889}{1615} & \frac{1597}{1292} & -\frac{377}{682} \\ \frac{987}{682} & -4 & \frac{329}{963} & \frac{987}{305} & -\frac{2889}{1615} & \frac{987}{1292} \\ -\frac{987}{1292} & \frac{2889}{1615} & -\frac{987}{305} & -\frac{329}{963} & 4 & -\frac{987}{682} \\ \frac{377}{682} & -\frac{1597}{1292} & \frac{2889}{1615} & -4 & -\frac{2255}{963} & \frac{1597}{305} \\ -1 & \frac{754}{341} & -\frac{987}{323} & \frac{1974}{341} & -\frac{6765}{323} & 17 \end{pmatrix}.$$

Example 2.1

In a closed interval $0 \le x \le 1$, the initial value problem is given as

$$y' - 4y = x, \qquad y(0) = 1$$

a) Solve the problem using uniform grid distribution for $N = 8$ and $N = 11$.

b) Solve the problem using CGL grid distribution for $N = 8$ and $N = 11$.

Solution 2.1

When the grid points x_i, $i = 1, 2, \ldots, N$ are applied to solve the initial value problem, it can be written the following

$$y'(x_i) - 4y(x_i) = x_i$$

if the DQ is replaced for $y'(x_i)$, it will be obtained as

$$\sum_{j=1}^{N} a_{ij} y(x_j) - 4y(x_i) = x_i \,.$$

Let us denote $y(x_j) = y_j$ and $y(x_i) = y_i$. Then the above equation can be stated as

$$\sum_{j=1}^{N} a_{ij} y_j - 4y_i = x_i \,.$$

Using the equation above, the following system of equations can be acquired

$$\left\{ a_{11} y(x_1) + a_{12} y(x_2) + a_{13} y(x_3) + \cdots + a_{1N} y(x_N) \right\} - 4y(x_1) = x_1 \,,$$

$$\left\{ a_{21} y(x_1) + a_{22} y(x_2) + a_{23} y(x_3) + \cdots + a_{2N} y(x_N) \right\} - 4y(x_2) = x_2 \,,$$

$$\vdots$$

$$\left\{ a_{N1} y(x_1) + a_{N2} y(x_2) + a_{N3} y(x_3) + \cdots + a_{NN} y(x_N) \right\} - 4y(x_N) = x_N \,.$$

Let us express the system of equations in the form of matrix and vector as follows

$$\left[a_{ij} - 4I \right]_{N \times N} \left[y(x_j) \right]_{N \times 1} = \left[x_i \right]_{N \times 1},$$

when the weighted coefficient matrix is written as $\left[A_{ij} \right]_{N \times N} = \left[a_{ij} - 4I \right]_{N \times N}$, the system will be in the following form:

$$\left[A_{ij} \right]_{N \times N} \left[y(x_j) \right]_{N \times 1} = \left[x_i \right]_{N \times 1}.$$

Using the initial condition $y(x_1) = 1$, the number of unknowns in the system of equations decreases to $N - 1$. One of the equations in the system can be removed. Generally, for initial condition, the first row of the matrix is removed.

a) Using uniform grid distribution for $N=8$, the system of equations is given as

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{18} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{28} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{38} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{81} & A_{82} & A_{83} & \cdots & A_{88} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_8 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_8 \end{pmatrix}$$

when the first row of the matrix above is replaced by the initial condition, it will be acquired the following

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ A_{21} & A_{22} & A_{23} & \cdots & A_{28} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{38} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{81} & A_{82} & A_{83} & \cdots & A_{88} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_8 \end{pmatrix} = \begin{pmatrix} 1 \\ x_2 \\ x_3 \\ \vdots \\ x_8 \end{pmatrix}.$$

The approximated values of y_i's can be obtained by solving the above system of equations. On the other hand, using analytical methods the solution of the initial value problem can be found as

$$y(x) = -\frac{1}{16}(4x+1) + \frac{17}{16}e^{4x}.$$

Now, the approximated solutions obtained by PDQ and the exact solutions, which are denoted by y_A and y_E, respectively, will be shown in Table 2.1 as well as absolute error at each grid point.

Table 2.1 The exact values, approximated values and absolute errors for uniform grid distribution when $N=8$

x_i	y_E	y_A	$\lvert y_E - y_A \rvert$
0.0000	1.00000000	1.00000000	0
0.1429	1.78325535	1.84770940	0.06445405
0.2857	3.19776836	3.30971434	0.11194598
0.4286	5.73010926	5.92888862	0.19877936
0.5714	10.24189413	10.59359952	0.35170539
0.7143	18.25886839	18.88195405	0.62308567
0.8571	32.48281433	33.58565189	1.10283756
1.0000	57.69803441	59.65292067	1.95488626

When the number of grid points $N=11$, the following results will be acquired at each grid point.

Table 2.2 The exact values, approximated values and absolute errors for uniform grid distribution when $N = 11$

x_i	y_E	y_A	$\lvert y_E - y_A \rvert$
0.0000	1.00000000	1.00000000	0
0.1000	1.49756374	1.49739846	0.00016528
0.2000	2.25213724	2.25189442	0.00024282
0.3000	3.39012423	3.38976138	0.00036285
0.4000	5.10009695	5.09955584	0.00054111
0.5000	7.66337211	7.66256476	0.00080734
0.6000	11.49962490	11.49842057	0.00120433
0.7000	17.23493719	17.23314044	0.00179675
0.8000	25.80331333	25.80063308	0.00268026
0.9000	38.59812409	38.59412505	0.00399904
1.0000	57.69803441	57.69207199	0.00596242

It can be realized from Tables 2.1 and 2.2 that increasing the number of grid points results obtaining more convergent approximated solutions.

b) Using CGL point distribution, the following values will be acquired for $N = 8$ and $N = 11$, respectively.

Table 2.3 The exact values, approximated values and absolute errors for CGL grid distribution when $N = 8$

x_i	y_E	y_A	$\lvert y_E - y_A \rvert$
0.0000	1.00000000	1.00000000	0
0.0495	1.22034930	1.22661880	0.00626951
0.1883	2.14655283	2.15069593	0.00414310
0.3887	4.87113392	4.88654926	0.01541534
0.6113	12.03640676	12.06946649	0.03305971
0.8117	27.05415919	27.13028296	0.07612377
0.9505	47.28700942	47.41878974	0.13178032
1.0000	57.69803441	57.85876993	0.16073552

Table 2.4 The exact values, approximated values and absolute errors for CGL grid distribution when $N=11$

x_i	y_E	y_A	$\lvert y_E - y_A \rvert$
0.0000	1.00000000	1.00000000	0
0.0245	1.10314757	1.10314257	0.00000500
0.0955	1.47036205	1.47036063	0.00000142
0.2061	2.30908868	2.30908136	0.00000732
0.3455	4.08277139	4.08276322	0.00000817
0.5000	7.66337211	7.66335317	0.00001894
0.6545	14.33941500	14.33938261	0.00003238
0.7939	25.17578060	25.17572233	0.00005827
0.9045	39.30462135	39.30453151	0.00008984
0.9755	52.29475067	52.29463109	0.00011965
1.0000	57.69803441	57.69790249	0.00013192

The Tables 2.3 and 2.4 show that increasing number of grid points decreases absolute errors in CGL grid distribution.

Example 2.2

In a closed interval $0 \le x \le 1$, the boundary value problem is given as

$$y'' + (x+1)y' - 2y = \left(1 - x^2\right)e^{-x}, \qquad y(0) = -1, \quad y(1) = 0$$

find the solution of the problem by PDQ method.

Solution 2.2

First, let us apply the grid points x_i 's to the given equation for $i = 1, 2, ..., N$

$$y''(x_i) + (x_i + 1)y'(x_i) - 2y(x_i) = \left(1 - x_i^{\,2}\right)e^{-x_i}$$

when DQ methods of the first and the second order derivatives are applied for $y'(x_i)$ and $y''(x_i)$, respectively, it will be obtained as

$$\sum_{j=1}^{N} w_{ij}^{(2)} y(x_j) + (x_i + 1)\sum_{j=1}^{N} a_{ij} y(x_j) - 2y(x_i) = \left(1 - x_i^{\,2}\right)e^{-x_i}$$

the above equation can be written as follows

$$\left\{w_{11}^{(2)} y_1 + \cdots + w_{1N}^{(2)} y_N\right\} + (x_1 + 1)\left\{a_{11} y_1 + \cdots + a_{1N} y_N\right\} - 2y_1 = \left(1 - x_1^{\,2}\right)e^{-x_1},$$

$$\left\{w_{21}^{(2)} y_1 + \cdots + w_{2N}^{(2)} y_N\right\} + (x_2 + 1)\left\{a_{21} y_1 + \cdots + a_{2N} y_N\right\} - 2y_2 = \left(1 - x_2^{\,2}\right)e^{-x_2},$$

$$\vdots$$

$$\left\{w_{N1}^{(2)}y_1+\cdots+w_{NN}^{(2)}y_N\right\}+\left(x_N+1\right)\left\{a_{N1}y_1+\cdots+a_{NN}y_N\right\}-2y_N=\left(1-x_N{}^2\right)e^{-x_N}.$$

Expressing the above equation in matrix and vector form, it will be represented as

$$\left[w_{ij}^{(2)}+\left(x_i+1\right)a_{ij}-2I\right]_{N\times N}\left[y\left(x_j\right)\right]_{N\times 1}=\left[\left(1-x_i{}^2\right)e^{-x_i}\right]_{N\times 1}.$$

If the weighted coefficient matrix is written as $\left[A_{ij}\right]_{N\times N}=\left[w_{ij}^{(2)}+\left(x_i+1\right)a_{ij}-2I\right]_{N\times N}$ the system will be in the form of

$$\left[A_{ij}\right]_{N\times N}\left[y\left(x_j\right)\right]_{N\times 1}=\left[\left(1-x_i{}^2\right)e^{-x_i}\right]_{N\times 1}.$$

When the grid points are applied to the boundary conditions it can be written as

$$y(0)=y\left(x_1\right)=-1,$$

$$y(1)=y\left(x_N\right)=0.$$

By substituting, the boundary conditions in the first row and the last row of the system of equations above, it will be stated that

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{(N-1)1} & A_{(N-1)2} & A_{(N-1)3} & \cdots & A_{(N-1)N} \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{pmatrix}=\begin{pmatrix} -1 \\ \left(1-x_2{}^2\right)e^{-x_2} \\ \vdots \\ \left(1-x_{(N-1)}^2\right)e^{-x_{(N-1)}} \\ 0 \end{pmatrix}.$$

The solution of the above system gives the approximated values of y_i 's by PDQ method. Using analytical methods, the solution of the given boundary value problem can be found as

$$y=\left(x-1\right)e^{-x}.$$

Applying the uniform grid distribution for $N=10$, it will be obtained the following results.

Table 2.5 The exact values, approximated values and absolute errors for uniform grid distribution when $N = 10$

x_i	y_E	y_A	$\lvert y_E - y_A \rvert$
0.0000	-1.00000000	-1.00000000	0
0.1111	-0.79541273	-0.79541273	1.06×10^{-10}
0.2222	-0.62279576	-0.62279576	9.53×10^{-11}
0.3333	-0.47768754	-0.47768754	9.21×10^{-11}
0.4444	-0.35621133	-0.35621133	9.08×10^{-11}
0.5556	-0.25500152	-0.25500152	9.19×10^{-11}
0.6667	-0.17113904	-0.17113904	9.50×10^{-11}
0.7778	-0.10209463	-0.10209463	9.90×10^{-11}
0.8889	-0.04567914	-0.04567914	1.09×10^{-11}
1.0000	0.00000000	0.00000000	0

Using the CGL grid distribution for $N = 10$, the following results can be acquired.

Table 2.6 The exact values, approximated values and absolute errors for CGL grid distribution when $N = 10$

x_i	y_E	y_A	$\lvert y_E - y_A \rvert$
0.0000	-1.00000000	-1.00000000	0
0.0302	-0.94103838	-0.94103838	1.62×10^{-12}
0.1170	-0.78554095	-0.78554095	2.98×10^{-14}
0.2500	-0.58410059	-0.58410059	1.49×10^{-12}
0.4132	-0.38821107	-0.38821107	1.53×10^{-14}
0.5868	-0.22976323	-0.22976323	1.31×10^{-12}
0.7500	-0.11809164	-0.11809164	5.58×10^{-13}
0.8830	-0.04837397	-0.04837397	6.06×10^{-13}
0.9698	-0.01143251	-0.01143251	1.62×10^{-12}
1.0000	0.00000000	0.00000000	0

Example 2.3

In a closed interval $0 \le x \le 1$, find the solution of the initial value problem

$$y'' - 5y' + 6y = 0, \quad y(0) = 0, \quad y(0) = -1$$

by PDQ method.

Solution 2.3

If the grid points x_i's are applied to the initial value problem for $i = 1, 2, \ldots, N$, it can be written the following

$$y''(x_i) - 5y'(x_i) + 6y(x_i) = 0$$

when DQ methods of first and second order derivatives are applied in the above equations for $y'(x_i)$ and $y''(x_i)$, respectively, it will be obtained as

$$\sum_{j=1}^{N} w_{ij}^{(2)} y(x_j) - 5\sum_{j=1}^{N} a_{ij} y(x_j) + 6y(x_i) = 0 .$$

It can be expressed the above equation as follows

$$\left\{w_{11}^{(2)} y_1 + w_{12}^{(2)} y_2 + \cdots + w_{1N}^{(2)} y_N\right\} - 5\left\{a_{11}y_1 + a_{12}y_2 + \cdots + a_{1N}y_N\right\} + 6y_1 = 0 ,$$

$$\left\{w_{21}^{(2)} y_1 + w_{22}^{(2)} y_2 + \cdots + w_{2N}^{(2)} y_N\right\} - 5\left\{a_{21}y_1 + a_{22}y_2 + \cdots + a_{2N}y_N\right\} + 6y_2 = 0 ,$$

$$\vdots$$

$$\left\{w_{N1}^{(2)} y_1 + w_{N2}^{(2)} y_2 + \cdots + w_{NN}^{(2)} y_N\right\} - 5\left\{a_{N1}y_1 + a_{N2}y_2 + \cdots + a_{NN}y_N\right\} + 6y_N = 0 .$$

If the above system of equations is written in the form of matrix and vector, it will be acquired as

$$\left[w_{ij}^{(2)} - 5a_{ij} + 6I\right]_{N\times N} \left[y(x_j)\right]_{N\times 1} = [0]_{N\times 1} .$$

When the weighted coefficient matrix is written as $\left[A_{ij}\right]_{N\times N} = \left[w_{ij}^{(2)} - 5a_{ij} + 6I\right]_{N\times N}$ the system of equations can be expressed in the form of

$$\left[A_{ij}\right]_{N\times N} \left[y(x_j)\right]_{N\times 1} = [0]_{N\times 1} .$$

Applying the grid points to the initial conditions, it can be written as

$$y(0) = y(x_1) = 0 ,$$

$$y'(0) = y'(x_1) = \sum_{j=1}^{N} a_{1j} y(x_j) = -1 .$$

By substituting the initial conditions in the first and the last rows of the system of equations above, it will be stated the following

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{(N-1)1} & A_{(N-1)2} & A_{(N-1)3} & \cdots & A_{(N-1)N} \\ a_{11} & a_{12} & a_{13} & \cdots & a_{1N} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix}.$$

The approximated values of y_i's by PDQ method can be obtained from the system of equations above. Analytical methods give the solution of the initial value problem as

$$y = e^{2x} - e^{3x}.$$

Applying the uniform grid and CGL grid distribution, respectively, the following solutions will be acquired for $N = 9$.

Table 2.7 The exact values, approximated values and absolute errors for uniform grid distribution when $N = 9$

x_i	y_E	y_A	$\lvert y_E - y_A \rvert$
0.0000	0.00000000	0.00000000	0
0.1250	-0.17096600	-0.17119339	0.00022739
0.2500	-0.46827875	-0.46898981	0.00071107
0.3750	-0.96321683	-0.96473547	0.00151864
0.5000	-1.76340724	-1.76623917	0.00283193
0.6250	-3.03047616	-3.03539548	0.00491932
0.7500	-5.00604677	-5.01422992	0.00818316
0.8750	-8.04997151	-8.06319678	0.01322526
1.0000	-12.69648082	-12.71737111	0.02089029

Table 2.8 The exact values, approximated values and absolute errors for CGL grid distribution when $N = 9$

x_i	y_E	y_A	$\lvert y_E - y_A \rvert$
0.0000	0.00000000	0.00000000	0
0.0381	-0.04186210	-0.04186679	0.00000468
0.1464	-0.21138284	-0.21140215	0.00001931
0.3087	-0.67038148	-0.67043797	0.00005649
0.5000	-1.76340724	-1.76355905	0.00015181
0.6913	-3.97120366	-3.97153905	0.00033539
0.8536	-7.43137190	-7.43200161	0.00062971
0.9619	-11.07076998	-11.07170497	0.00093499
1.0000	-12.69648082	-12.69755390	0.00107308

Example 2.4

In a closed interval $0 \le x \le 1$, solve the initial value problem

$$y^{(4)} - y = 0, \qquad y(0) = \frac{7}{2}, \quad y'(0) = -4, \quad y''(0) = \frac{5}{2}, \quad y'''(0) = -2$$

by PDQ method.

Solution 2.4

Applying the grid points x_i's to the initial value problem for $i = 1, 2, \ldots, N$, the differential equation can be written as in the form of

$$y^{(4)}(x_i) - y(x_i) = 0$$

when DQ method of fourth order derivative is applied in the above equation for $y^{(4)}(x_i)$, it will be obtained as

$$\sum_{j=1}^{N} w_{ij}^{(4)} y(x_j) - y(x_i) = 0\,.$$

It can be stated the above equation as follows

$$\left\{ w_{11}^{(4)} y_1 + w_{12}^{(4)} y_2 + \cdots + w_{1N}^{(4)} y_N \right\} - y_1 = 0\,,$$

$$\left\{ w_{21}^{(4)} y_1 + w_{22}^{(4)} y_2 + \cdots + w_{2N}^{(4)} y_N \right\} - y_2 = 0\,,$$

$$\vdots$$

$$\left\{ w_{N1}^{(4)} y_1 + w_{N2}^{(4)} y_2 + \cdots + w_{NN}^{(4)} y_N \right\} - y_N = 0\,.$$

If we express the above system of equations in the form of matrix and vector, it will be acquired as

$$\left[w_{ij}^{(4)} - I \right]_{N\times N} \left[y(x_j) \right]_{N\times 1} = [0]_{N\times 1}\,.$$

When the weighted coefficient matrix is shown as $\left[A_{ij} \right]_{N\times N} = \left[w_{ij}^{(4)} - I \right]_{N\times N}$, the system can be written in the following form

$$\left[A_{ij} \right]_{N\times N} \left[y(x_j) \right]_{N\times 1} = [0]_{N\times 1}\,.$$

Applying the grid points to the initial conditions, it can be stated as follows

$$y(0)=y(x_1)=\frac{7}{2},$$

$$y'(0)=y'(x_1)=\sum_{j=1}^{N}a_{1j}y(x_j)=-4,$$

$$y''(0)=y''(x_1)=\sum_{j=1}^{N}w_{1j}^{(2)}y(x_j)=\frac{5}{2},$$

$$y'''(0)=y'''(x_1)=\sum_{j=1}^{N}w_{1j}^{(3)}y(x_j)=-2.$$

By replacing the initial conditions, $y(0)$ and $y'(0)$ in the first two rows, and the initial conditions, $y''(0)$ and $y'''(0)$ in the last two rows, of the system of equations above, it can be written the following system of equations to be solved:

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_{11} & a_{12} & a_{13} & \cdots & a_{1N} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{(N-2)1} & A_{(N-2)2} & A_{(N-2)3} & \cdots & A_{(N-2)N} \\ w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} & \cdots & w_{1N}^{(2)} \\ w_{11}^{(3)} & w_{12}^{(3)} & w_{13}^{(3)} & \cdots & w_{1N}^{(3)} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-2} \\ y_{N-1} \\ y_N \end{pmatrix} = \begin{pmatrix} 7/2 \\ -4 \\ 0 \\ \vdots \\ 0 \\ 5/2 \\ -2 \end{pmatrix}.$$

The solution of the system above gives approximated values of y_i 's by PDQ method. By using analytical methods, the solution of the initial value problem will be found as

$$y=3e^{-x}+\frac{1}{2}\cos x-\sin x.$$

Applying the uniform and CGL grid distributions, respectively, the following solutions will be obtained for $N=13$.

Table 2.9 The exact values, approximated values and absolute errors for uniform grid distribution when $N = 13$

x_i	y_E	y_A	$\lvert y_E - y_A \rvert$
0.0000	3.50000000	3.50000000	0
0.0833	3.17516122	3.17516122	4.21×10^{-13}
0.1667	2.86662066	2.86662066	4.22×10^{-12}
0.2500	2.57345460	2.57345460	1.58×10^{-11}
0.3333	2.29487771	2.29487771	3.96×10^{-11}
0.4167	2.03022886	2.03022886	8.01×10^{-11}
0.5000	1.77895772	1.77895772	1.42×10^{-10}
0.5833	1.54061197	1.54061197	2.30×10^{-10}
0.6667	1.31482518	1.31482519	3.48×10^{-10}
0.7500	1.10130533	1.10130533	5.00×10^{-10}
0.8333	0.89982389	0.89982390	6.93×10^{-10}
0.9167	0.71020556	0.71020556	9.29×10^{-10}
1.0000	0.53231849	0.53231850	1.21×10^{-9}

Table 2.10 The exact values, approximated values and absolute errors for CGL grid distribution when $N = 13$

x_i	y_E	y_A	$\lvert y_E - y_A \rvert$
0.0000	3.50000000	3.50000000	0
0.0170	3.43221284	3.43221284	2.92×10^{-13}
0.0670	3.23756262	3.23756262	3.17×10^{-13}
0.1464	2.94003976	2.94003976	3.38×10^{-13}
0.2500	2.57345460	2.57345460	3.26×10^{-13}
0.3706	2.17487071	2.17487071	2.59×10^{-13}
0.5000	1.77895772	1.77895772	1.27×10^{-13}
0.6294	1.41423922	1.41423922	5.88×10^{-14}
0.7500	1.10130533	1.10130533	2.72×10^{-13}
0.8536	0.85273043	0.85273043	4.80×10^{-13}
0.9330	0.67439120	0.67439120	6.50×10^{-13}
0.9830	0.56773777	0.56773777	7.59×10^{-13}
1.0000	0.53231849	0.53231849	7.97×10^{-13}

CHAPTER 3

SOLUTION OF PARTIAL DIFFERENTIAL EQUATION BY DIFFERENTIAL QUADRATURE METHOD

The numerical solutions might be the only way to solve a partial differential equation due to unavailability of obtaining the analytical solution. Various numerical solution methods have been improved to solve PDEs. One of the most commonly used one among these methods is finite difference method. This method has been applied in solutions of ODEs and PDEs by reason of easiness in its usage.

3.1 FINITE DIFFERENCE METHOD

In this section, it will be introduced the finite difference method which will be applied together with polynomial based differential quadrature method (PDQ) in the solutions of PDEs as a subsection of this study [19].

Using the Taylor series expansion, the value of a function $f(x)$ at the point $(x+\Delta x)$ can be expressed as

$$f(x_i+\Delta x)=f(x_i)+\frac{f'(x_i)}{1!}\Delta x+\frac{f''(x_i)}{2!}(\Delta x)^2+\cdots+\frac{f^{(n)}(x_i)}{n!}(\Delta x)^n+\cdots.$$

The expression above can be written as follows

$$f(x_i+\Delta x)=f(x_i)+\frac{f'(x_i)}{1!}\Delta x+\frac{f''(x_i)}{2!}(\Delta x)^2+\cdots+\frac{f^{(n)}(x_i)}{n!}(\Delta x)^n+R_n(x) \quad (3.1)$$

where $R_n(x)$ is a remainder term, symbolizing the difference between the Taylor polynomial of degree n and the original function $f(x)$. When the expression (3.1) is truncated after the first-degree derivative, it will be obtained the following

$$f(x+\Delta x) = f(x) + \frac{f'(x)}{1!}\Delta x + R_1(x),$$

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} - \frac{R_1(x)}{\Delta x}. \tag{3.2}$$

Assuming that $R_1(x)$ is sufficiently small then the first order approximation of the function $f(x)$ is given by

$$f'(x) \approx \frac{f(x+\Delta x) - f(x)}{\Delta x}. \tag{3.3}$$

The expression (3.3) is called as forward difference approximation. When the equation (3.1) is stated as

$$f(x_i - \Delta x) = f(x_i) - \frac{f'(x_i)}{1!}\Delta x + \frac{f''(x_i)}{2!}(\Delta x)^2 + \cdots - \frac{f^{(n)}(x_i)}{n!}(\Delta x)^n + R_n(x)$$

the first order derivative of $f(x)$ will be acquired as follows

$$f(x-\Delta x) = f(x) - \frac{f'(x)}{1!}\Delta x + R_1(x), \tag{3.4}$$

$$f'(x) \approx \frac{f(x) - f(x-\Delta x)}{\Delta x} \tag{3.5}$$

where the expression (3.5) is called as backward difference approximation. When the Eqs. (3.2) and (3.4) are added side by side and rearranged, it will be obtained as

$$f'(x) \approx \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} \tag{3.6}$$

which is called as central difference approximation. The order of the local truncation error in forward and backward difference approximations can be stated as

$$E_n = \pm\frac{\Delta x}{2} f''(\xi) = O(\Delta x), \quad x_n \le \xi \le x_n + \Delta x, \quad x_n - \Delta x \le \xi \le x_n.$$

However, the order of the local truncation error in central difference approximation can be expressed as

$$E_n = -\frac{\Delta x}{6} f'''(\xi) = O(\Delta x)^2, \quad x_{n-1} \le \xi \le x_{n+1} + \Delta x.$$

There exists first order local truncation error in forward and backward difference approximations whereas the central difference approximation exists the second order local truncation error.

3.2 SOLVING PDEs BY DIFFERENTIAL QUADRATURE METHOD

In this section, the first and the second order partial differential equations under initial and boundary value conditions will be solved using polynomial differential quadrature method. The value of the solution obtained by PDQs will be checked at each grid point using the exact solution of the problem and the absolute error is shown to compare the results with other solution methods. In the computation, the weighted coefficients are used similar to Shu’s approaches. The MATLAB programming is used in the calculation of algebraic systems and all the algorithms written in MATLAB are given in Appendix B.

Example 3.1 Homogeneous Parabolic (Heat) Equation

Homogeneous parabolic (heat) equation can be defined as

$$u_t(x,t) = u_{xx}(x,t), \quad 0 < x < 1, \quad t > 0$$

along with the boundary and initial value conditions, respectively,

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

$$u(x,0) = -\frac{1}{2}\sin 3\pi x + \frac{3}{2}\sin \pi x, \quad 0 < x < 1.$$

For the solution process let us write down the homogeneous heat equation at each grid points x_i, $i = 1, 2, \ldots\ldots, N$, like

$$U_t(x_i,t) = U_{xx}(x_i,t).$$

Now, let us apply the forward difference approximation given in (3.3) for the first order derivative of the function with respect to t. On the other hand, let us use the differential quadrature formula for the second order derivative of the function with respect to x. Finally, it will be obtained a recurrence relation at each grid point as

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

$$U(x_i,t+\Delta t) = U(x_i,t) + \Delta t \sum_{j=1}^{N} w_{ij}^{(2)} U(x_j,t).$$

Using the equation above, the following system of equations can be acquired

$$U(x_1,t+\Delta t)=U(x_1,t)+\Delta t\left\{w_{11}^{(2)}U(x_1,t)+w_{12}^{(2)}U(x_2,t)+\cdots+w_{1N}^{(2)}U(x_N,t)\right\}$$

$$U(x_2,t+\Delta t)=U(x_2,t)+\Delta t\left\{w_{21}^{(2)}U(x_1,t)+w_{22}^{(2)}U(x_2,t)+\cdots+w_{2N}^{(2)}U(x_N,t)\right\}$$

$$\vdots$$

$$U(x_N,t+\Delta t)=U(x_N,t)+\Delta t\left\{w_{N1}^{(2)}U(x_1,t)+w_{N2}^{(2)}U(x_2,t)+\cdots+w_{NN}^{(2)}U(x_N,t)\right\}.$$

Let us express the system of equations in matrix and vector forms as follows

$$\left[U(x_i,t+\Delta t)\right]_{N\times 1}=\left[I+\Delta t\, w_{ij}^{(2)}\right]_{N\times N}\left[U(x_j,t)\right]_{N\times 1}.$$

When the weighted coefficient matrix is stated as $\left[A_{ij}\right]_{N\times N}=\left[I+\Delta t\, w_{ij}^{(2)}\right]_{N\times N}$, then the system will be represented as:

$$\left[U(x_i,t+\Delta t)\right]_{N\times 1}=\left[A_{ij}\right]_{N\times N}\left[U(x_j,t)\right]_{N\times 1}. \tag{3.7}$$

In Equation (3.7), using the initial value for time like $t=0$, it can be obtained the value of the function at the next time step $t+\Delta t$. Let us show the solution process:
The value of the unknown function $u(x,t)$ at the first and the last grid points are known as boundary values like

$$u(0,t)=U(x_1,t)=0, \qquad u(1,t)=U(x_N,t)=0$$

and for $t=0$, the value of the unknown function at each middle points are given as initial conditions like

$$u(x,0)=U(x_i,0)=-\frac{1}{2}\sin 3\pi x_i+\frac{3}{2}\sin \pi x_i .$$

Let us write the system of equations for the next time step $t+\Delta t$ using Equation (3.7) along with the boundary and initial values. If it is substituted the boundary conditions of $U(0,t)$ and $U(1,t)$ in the first and last rows of $\left[A_{ij}\right]$, respectively, the Eq. (3.7) can be shown as in the matrix form as

$$\begin{pmatrix} U(x_1,0+\Delta t) \\ U(x_2,0+\Delta t) \\ \vdots \\ U(x_{N-1},0+\Delta t) \\ U(x_N,0+\Delta t) \end{pmatrix}=\begin{pmatrix} 1 & 0 & \cdots & 0 \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{(N-1)1} & A_{(N-1)2} & \cdots & A_{(N-1)N} \\ 0 & 0 & \cdots & 1 \end{pmatrix}\begin{pmatrix} U(x_1,0) \\ U(x_2,0) \\ \vdots \\ U(x_{N-1},0) \\ U(x_N,0) \end{pmatrix}.$$

Since the value of the unknown function at $t=0$ for each middle grid point is known then the system of the equations above turns into the following system to be solved:

$$\begin{pmatrix} U(x_1,0+\Delta t) \\ U(x_2,0+\Delta t) \\ \vdots \\ U(x_{N-1},0+\Delta t) \\ U(x_N,0+\Delta t) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{(N-1)1} & A_{(N-1)2} & \cdots & A_{(N-1)N} \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} 0 \\ -\frac{1}{2}\sin 3\pi x_2 + \frac{3}{2}\sin \pi x_2 \\ \vdots \\ -\frac{1}{2}\sin 3\pi x_{N-1} + \frac{3}{2}\sin \pi x_{N-1} \\ 0 \end{pmatrix}.$$

The solution of the system above gives the approximated values of $U(x_i,0+\Delta t)$ by polynomial differential quadrature along with the finite difference method. By using analytical methods, so called separation of variables, the solution of the given heat equation will be obtained as

$$u(x,t) = \frac{3}{2}\sin(\pi x)e^{-\pi^2 t} - \frac{1}{2}\sin(3\pi x)e^{-9\pi^2 t}.$$

Applying the uniform grid distribution for $N=10$ given in Chapter 2 for different values of time steps like $\Delta t = 0.1$, $\Delta t = 0.01$ and $\Delta t = 0.001$, respectively, the following results can be obtained. In these tables, u_E and U_A represent the exact and the approximate values of the unknown function $u(x,t)$, respectively.

Table 3.1 The exact values, approximated values and absolute errors for uniform grid distribution when $\Delta t = 0.1$ and $N = 10$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0	0	0
0.1111	0.19115029	3.26966056	3.07851027
0.2222	0.35929788	3.45092721	3.09162933
0.3333	0.48416168	0.01067521	0.47348648
0.4444	0.55062845	-3.39264583	3.94327428
0.5556	0.55062845	-3.39264583	3.94327428
0.6667	0.48416168	0.01067521	0.47348648
0.7778	0.35929788	3.45092721	3.09162933
0.8889	0.19115029	3.26966056	3.07851027
1.0000	0	0	0

Table 3.2 The exact values, approximated values and absolute errors for uniform grid distribution when $\Delta t = 0.01$ and $N = 10$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0	0	0
0.1111	0.28668659	0.39898182	0.11229523
0.2222	0.69543772	0.82314456	0.12770684
0.3333	1.17695198	1.17020182	0.00675016
0.4444	1.51650846	1.37993732	0.13657114
0.5556	1.51650846	1.37993732	0.13657114
0.6667	1.17695198	1.17020182	0.00675016
0.7778	0.69543772	0.82314456	0.12770684
0.8889	0.28668659	0.39898182	0.11229523
1.0000	0	0	0

Table 3.3 The exact values, approximated values and absolute errors for uniform grid distribution when $\Delta t = 0.001$ and $N = 10$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0	0	0
0.1111	0.11178320	0.11191394	0.00013074
0.2222	0.55850361	0.56036630	0.00186268
0.3333	1.28628017	1.28615448	0.00012570
0.4444	1.85891234	1.85719563	0.00171673
0.5556	1.85891236	1.85719563	0.00171673
0.6667	1.28628017	1.28615448	0.00012570
0.7778	0.55850362	0.56036630	0.00186268
0.8889	0.11178320	0.11191394	0.00013074
1.0000	0	0	0

Example 3.2 Non-homogeneous Parabolic (Heat) Equation

As an example of non-homogeneous parabolic (heat) equation can be defined as

$$u_t(x,t) = u_{xx}(x,t) + 2\sin x, \quad 0 < x < \pi, \quad t > 0$$

along with the boundary and initial value conditions, respectively,

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

$$u(x,0) = 3\sin x, \quad 0 < x < \pi.$$

For the solution process let us write down the homogeneous heat equation at each grid points x_i, $i = 1, 2, \ldots, N$, like

$$U_t(x_i, t) = U_{xx}(x_i, t) + 2\sin x_i .$$

Now, let us apply the forward difference approximation given in (3.3) for the first order derivative of the function with respect to t. On the other hand, let us use the differential quadrature formula for the second order derivative of the function with respect to x. Finally, it will be obtained a recurrence relation at each grid point as

$$\frac{U(x_i, t+\Delta t) - U(x_i, t)}{\Delta t} = \sum_{j=1}^{N} w_{ij}^{(2)} U(x_j, t) + 2\sin x_i ,$$

$$U(x_i, t+\Delta t) = U(x_i, t) + \Delta t \sum_{j=1}^{N} w_{ij}^{(2)} U(x_j, t) + 2\Delta t \sin x_i .$$

Using the equation above, the following system of equations can be written

$$\begin{aligned}
U(x_1, t+\Delta t) &= U(x_1, t) \\
&\quad + \Delta t \left\{ w_{11}^{(2)} U(x_1, t) + w_{12}^{(2)} U(x_2, t) + \cdots + w_{1N}^{(2)} U(x_N, t) \right\} + 2\Delta t \sin x_1 \\
U(x_2, t+\Delta t) &= U(x_2, t) \\
&\quad + \Delta t \left\{ w_{21}^{(2)} U(x_1, t) + w_{22}^{(2)} U(x_2, t) + \cdots + w_{2N}^{(2)} U(x_N, t) \right\} + 2\Delta t \sin x_2 \\
&\vdots \\
U(x_N, t+\Delta t) &= U(x_N, t) \\
&\quad + \Delta t \left\{ w_{N1}^{(2)} U(x_1, t) + w_{N2}^{(2)} U(x_2, t) + \cdots + w_{NN}^{(2)} U(x_N, t) \right\} + 2\Delta t \sin x_N \; .
\end{aligned}$$

Expressing the system of equations in the form of matrix and vector it will be acquired the following

$$\left[U(x_i, t+\Delta t)\right]_{N\times 1} = \left[I + \Delta t\, w_{ij}^{(2)}\right]_{N\times N} \left[U(x_j, t)\right]_{N\times 1} + 2\Delta t \sin x_j$$

When the weighted coefficient matrix is donated as $\left[A_{ij}\right]_{N\times N} = \left[I + \Delta t\, w_{ij}^{(2)}\right]_{N\times N}$, the system will be in the following form.

$$\left[U(x_i, t+\Delta t)\right]_{N\times 1} = \left[A_{ij}\right]_{N\times N} \left[U(x_j, t)\right]_{N\times 1} + \left[B_j\right]_{N\times 1} \tag{3.8}$$

where

$$B=\begin{pmatrix} 2\Delta t\sin x_1 \\ 2\Delta t\sin x_2 \\ \vdots \\ 2\Delta t\sin x_{N-1} \\ 2\Delta t\sin x_N \end{pmatrix}=\begin{pmatrix} 0 \\ 2\Delta t\sin x_2 \\ \vdots \\ 2\Delta t\sin x_{N-1} \\ 0 \end{pmatrix}$$

In Equation (3.8), using the initial value for time like $t=0$, it can be obtained the value of the function at the next time step $t+\Delta t$. Let us show the solution process:

The value of the unknown function $u(x,t)$ at the first and the last grid points are known as boundary values like

$$u(0,t)=U(x_1,t)=0, \qquad u(1,t)=U(x_N,t)=0$$

and for $t=0$, the value of the unknown function at each middle points are given as initial conditions like

$$u(x,0)=U(x_i,0)=3\sin x_i.$$

Let us write the system of equations for the next time step $t+\Delta t$ using Equation (3.8) along with the boundary and initial values. If it is substituted the boundary conditions of $U(0,t)$ and $U(1,t)$ in the first and last rows of $\left[A_{ij}\right]$, respectively, the Eq. (3.8) can be shown as in the matrix form as

$$\begin{pmatrix} U(x_1,0+\Delta t) \\ U(x_2,0+\Delta t) \\ \vdots \\ U(x_{N-1},0+\Delta t) \\ U(x_N,0+\Delta t) \end{pmatrix}=\begin{pmatrix} 1 & 0 & \cdots & 0 \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{(N-1)1} & A_{(N-1)2} & \cdots & A_{(N-1)N} \\ 0 & 0 & \cdots & 1 \end{pmatrix}\begin{pmatrix} U(x_1,0) \\ U(x_2,0) \\ \vdots \\ U(x_{N-1},0) \\ U(x_N,0) \end{pmatrix}+\begin{pmatrix} 0 \\ 2\Delta t\sin x_2 \\ \vdots \\ 2\Delta t\sin x_{N-1} \\ 0 \end{pmatrix}.$$

Since the value of the unknown function at $t=0$ for each middle grid point is known then the system of the equations above turns into the following system to be solved:

$$\begin{pmatrix} U(x_1,0+\Delta t) \\ U(x_2,0+\Delta t) \\ \vdots \\ U(x_{N-1},0+\Delta t) \\ U(x_N,0+\Delta t) \end{pmatrix}=\begin{pmatrix} 1 & 0 & \cdots & 0 \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{(N-1)1} & A_{(N-1)2} & \cdots & A_{(N-1)N} \\ 0 & 0 & \cdots & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 3\sin x_2 \\ \vdots \\ 3\sin x_{N-1} \\ 0 \end{pmatrix}+\begin{pmatrix} 0 \\ 2\Delta t\sin x_2 \\ \vdots \\ 2\Delta t\sin x_{N-1} \\ 0 \end{pmatrix}.$$

The solution of the system above gives the approximated values of $U(x_i,0+\Delta t)$ by polynomial differential quadrature along with the finite difference method. By using analytical methods, so called separation of variables, the solution of the given heat equation will be obtained as

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

Applying the uniform grid distribution for $N = 11$ given in Chapter 2 for different values of time steps like $\Delta t = 0.03$ and $\Delta t = 0.003$, respectively, the following results can be obtained. In these tables, u_E and U_A represent the exact and the approximate values of the unknown function $u(x,t)$, respectively.

Table 3.4 The exact values, approximated values and absolute errors for uniform grid distribution when $\Delta t = 0.03$ and $N = 11$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0	0	0
0.3142	0.91791815	0.91778048	1.38×10^{-4}
0.6283	1.74598408	1.74572220	2.62×10^{-4}
0.9425	2.40314092	2.40278047	3.60×10^{-4}
1.2566	2.82506159	2.82463785	4.24×10^{-4}
1.5708	2.97044553	2.97000000	4.46×10^{-4}
1.8850	2.82506158	2.82463785	4.24×10^{-4}
2.1991	2.40314092	2.40278047	3.60×10^{-4}
2.5133	1.74598408	1.74572220	2.62×10^{-4}
2.8274	0.91791815	0.91778048	1.38×10^{-4}
3.1416	0	0	0

Table 3.5 The exact values, approximated values and absolute errors for uniform grid distribution when $\Delta t = 0.003$ and $N = 11$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0	0	0
0.3142	0.92612532	0.92612393	1.39×10^{-6}
0.6283	1.76159504	1.76159240	2.64×10^{-6}
0.9425	2.42462757	2.42462393	3.64×10^{-6}
1.2566	2.85032065	2.85031638	4.28×10^{-6}
1.5708	2.99700450	2.99700000	4.50×10^{-6}
1.8850	2.85032065	2.85031638	4.28×10^{-6}
2.1991	2.42462757	2.42462393	3.64×10^{-6}
2.5133	1.76159504	1.76159240	2.64×10^{-6}
2.8274	0.92612532	0.92612393	1.39×10^{-6}
3.1416	0	0	0

CHAPTER 4

SOLUTION OF ADVECTION DIFFUSION EQUATIONS BY DIFFERENTIAL QUADRATURE METHOD

Advection-Diffusion-Reaction (ADR) equation is one of the most commonly used equations in mathematical physics. In this chapter, it will be introduced ADR equation and, subsequently, its numerical solution will be obtained by PDQ method.

4.1 ADVECTION-DIFFUSION-REACTION EQUATIONS

Many events in real life can be modeled using ADR equation. Therefore, solution of ADR equation has been studied by many scientists. Since both advection and diffusion terms exist in the ADR equation, it occurs frequently in transferring mass, heat and energy in engineering and chemistry. In addition, ADR equation has been used to model physical and chemical applications such as heat transfer in draining film, the spread of pollutants in rivers and streams, the absorption of chemical into beds, thermal pollution in river systems and some economic and financial forecasting [24].

Let us consider a concentration $u(x,t)$ of a certain chemical species with the space variable x and time t. Taking $h>0$ as a small number and considering the average concentration $\bar{u}(x,t)$ in a cell $\Omega(x)=\left[x-\frac{1}{2}h, x+\frac{1}{2}h\right]$, it can be written the following

$$\bar{u}(x,t)=\frac{1}{h}\int_{x-h/2}^{x+h/2} u(x',t)\,dx' = u(x,t)+\frac{1}{24}h^2 u_{xx}(x,t)+\cdots.$$

If the species is carried along by a flowing medium with velocity $\nu(x,t)$, then according to the mass conservation law the change of $\bar{u}(x,t)$ per unit of time is the net balance of inflow and outflow over the cell boundaries such that

$$\frac{\partial}{\partial t}\overline{u}(x,t)=\frac{1}{h}\left[\nu\left(x-\frac{1}{2}h,t\right)u\left(x-\frac{1}{2}h,t\right)-\nu\left(x+\frac{1}{2}h,t\right)u\left(x+\frac{1}{2}h,t\right)\right]$$

where $\nu\left(x\pm\frac{1}{2}h,t\right)u\left(x\pm\frac{1}{2}h,t\right)$ are the fluxes over the left and right cell boundaries. If we let $h\to 0$, the concentration satisfies the following

$$\frac{\partial}{\partial t}u(x,t)+\frac{\partial}{\partial x}\big(\nu(x,t)u(x,t)\big)=0\,. \tag{4.1}$$

The Eq. (4.1) is called as advection equation. Similarly, when it is considered the effect of diffusion, the change of $\overline{u}(x,t)$ is caused by gradients in the solution and the fluxes across the cell boundaries are $-D\left(x\pm\frac{1}{2}h,t\right)u_x\left(x\pm\frac{1}{2}h,t\right)$ where $D(x,t)$ is named as the diffusion coefficient. The diffusion equation can be stated as follows

$$\frac{\partial}{\partial t}u(x,t)=\frac{\partial}{\partial x}\left(D(x,t)\frac{\partial}{\partial x}u(x,t)\right). \tag{4.2}$$

Combining the effects in Equations (4.1) and (4.2), the overall change in concentration can be expressed as

$$\frac{\partial}{\partial t}u(x,t)+\frac{\partial}{\partial x}\big(\nu(x,t)u(x,t)\big)=\frac{\partial}{\partial x}\left(D(x,t)\frac{\partial}{\partial x}u(x,t)\right). \tag{4.3}$$

The Eq. (4.3) is called as Advection-Diffusion-Reaction equation [25].

It is not easy to obtain analytical solution when the initial and boundary conditions are complex and the advection term is dominant. Thus, there have been developed a variety of numerical methods to solve ADR equations [24].

4.2 SOLVING PDEs BY DIFFERENTIAL QUADRATURE METHOD

In this section, the numerical solution of advection equation and ADR equation will be obtained by PDQ method, respectively. At each grid point, the exact solution that is obtained analytically and the approximated solution will be computed and the comparison of these solutions will be shown on the tables. Shu's approaches will be applied to find the weighted coefficients. In calculations of algebraic systems MATLAB is used and all the MATLAB programs are given in Appendix C.

Example 4.1 Homogeneous Advection Equation with Constant Velocity

The advection equation with constant velocity is given as

$u_t(x,t) - \nu u_x(x,t) = 0$, $\quad 0 < x < 1$, $\quad t > 0$,

$u(x,0) = x^2$

where ν represents the constant velocity.

At the beginning let us obtain close-form solution of the given advection equation by the method of characteristics [26].

Considering coefficients of the advection equation the following system of equations can be set

$$\frac{dx}{-\nu} = \frac{dt}{1}\ ,\ du = 0 \tag{4.4}$$

Solving the first equation of (4.4) the following will be written

$$x + \nu t = c$$

where c is a constant. The second equation of (4.4) gives the solution below

$$u(x,t) = c_1$$

This means that

$$u(x,t) = c_1 = g(c) = g(x + \nu t) \tag{4.5}$$

where $g(x+\nu t)$ is an arbitrary function. Substituting the given initial condition into (4.5) gives the following

$$g(x) = x^2$$

Accordingly, it can be acquired as

$$g(x+\nu t) = (x+\nu t)^2$$

Thus the close-form solution is obtained as follows

$$u(x,t) = (x+\nu t)^2 \tag{4.6}$$

Now let us continue with obtaining numerical solution of the given advection equation by DQ method.

When the grid points x_i, $i = 1, 2, \ldots, N$, are applied to solve the advection equation, the unknown function $u(x,t)$ at each grid point can be represented as $U(x_i,t)$ and the equation becomes

$$U_t(x_i,t)=\nu U_x(x_i,t).$$

If the forward difference approximation given in Chapter 3 is applied for the derivative of the function with respect to t and the first order DQ is applied for the derivative of the function with respect to x, it will be acquired the following

$$\frac{U(x_i,t+\Delta t)-U(x_i,t)}{\Delta t}=\nu\sum_{j=1}^{N}a_{ij}\,U(x_j,t)$$

$$U(x_i,t+\Delta t)=U(x_i,t)+\Delta t\cdot\nu\sum_{j=1}^{N}a_{ij}\,U(x_j,t).$$

If it is written in explicitly, the system of equations can be given as

$$U(x_1,t+\Delta t)=U(x_1,t)+\Delta t\cdot\nu\left\{a_{11}U(x_1,t)+a_{12}U(x_2,t)+\cdots+a_{1N}U(x_N,t)\right\},$$

$$U(x_2,t+\Delta t)=U(x_2,t)+\Delta t\cdot\nu\left\{a_{21}U(x_1,t)+a_{22}U(x_2,t)+\cdots+a_{2N}U(x_N,t)\right\},$$

$$\vdots$$

$$U(x_N,t+\Delta t)=U(x_N,t)+\Delta t\cdot\nu\left\{a_{N1}U(x_1,t)+a_{N2}U(x_2,t)+\cdots+a_{NN}U(x_N,t)\right\}.$$

Let us express the system of equations in the form of matrix and vector as

$$\left[U(x_i,t+\Delta t)\right]_{N\times 1}=\left[I+\Delta t\cdot\nu a_{ij}\right]_{N\times N}\left[U(x_j,t)\right]_{N\times 1}$$

and, when the weighted coefficient matrix is stated as $\left[A_{ij}\right]_{N\times N}=\left[I+\Delta t\cdot\nu a_{ij}\right]_{N\times N}$, then the system will be obtained like

$$\left[U(x_i,t+\Delta t)\right]_{N\times 1}=\left[A_{ij}\right]_{N\times N}\left[U(x_j,t)\right]_{N\times 1}. \tag{4.7}$$

Finally, the solution of the system gives us the value of the unknown function at each grid points in the next time step. Let us show how to apply Eq. (4.7) to the problem:
For $t=0$, let us find the values of the unknown function $U(x_i,0+\Delta t)$ at the next time step using the initial conditions those are derived from the close-form solution (4.6)

$$u(0,t)=U(x_1,t)=(\nu\cdot\Delta t)^2 \quad\text{and}\quad u(1,t)=U(x_N,t)=(1+\nu\cdot\Delta t)^2,$$

$$U(x_i,0)=(x_i)^2$$

After the substitution of boundary conditions $U(x_N,t)=(1+\nu\Delta t)^2$ and $U(x_1,t)=(\nu\Delta t)^2$ in the last and the first rows of $\left[A_{ij}\right]$, respectively, the new system becomes

$$\begin{pmatrix} U(x_1,0+\Delta t) \\ U(x_2,0+\Delta t) \\ \vdots \\ U(x_{N-1},0+\Delta t) \\ U(x_N,0+\Delta t) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{(N-1)1} & A_{(N-1)2} & \cdots & A_{(N-1)N} \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} (\nu\cdot\Delta t)^2 \\ U(x_2,0) \\ \vdots \\ U(x_{N-1},0) \\ (1+\nu\cdot\Delta t)^2 \end{pmatrix}.$$

Finally, by substituting the initial condition $U(x_i,0)=(x_i)^2$ into the system above, the new system of equations can be shown as follows:

$$\begin{pmatrix} U(x_1,0+\Delta t) \\ U(x_2,0+\Delta t) \\ \vdots \\ U(x_{N-1},0+\Delta t) \\ U(x_N,0+\Delta t) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{(N-1)1} & A_{(N-1)2} & \cdots & A_{(N-1)N} \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} (\nu\cdot\Delta t)^2 \\ (x_1)^2 \\ \vdots \\ (x_{N-1})^2 \\ (1+\nu\cdot\Delta t)^2 \end{pmatrix}.$$

The solution of the system above gives the approximated values of unknown function $U(x_i,0+\Delta t)$ at the next time step by PDQ method.

When the constant velocity $\nu=0.5$, applying the uniform grid distribution for $N=12$ given in Chapter 2 for different values of time steps like $\Delta t=0.1$ and $\Delta t=0.01$, respectively, the following results can be obtained. In these tables, u_E and U_A represent the exact and the approximate values of the unknown function $u(x,t)$, respectively.

Table 4.1 The exact values, approximated values and absolute errors for uniform grid distribution when $\nu=0.5$, $\Delta t=0.1$ and $N=12$

x_i	u_E	U_A	$\lvert u_E - U_A\rvert$
0	0.10012500	0.10012500	0
0.0909	0.28461598	0.27751479	0.00710119
0.1818	0.47609420	0.47539243	0.00070176
0.2727	0.67906752	0.67667848	0.00238905
0.3636	0.89804386	0.89529464	0.00274921
0.4545	1.13753109	1.13392204	0.00360905
0.5455	1.40203710	1.39790239	0.00413470
0.6364	1.69606978	1.69104087	0.00502891
0.7273	2.02413702	2.01888544	0.00525158
0.8182	2.39074671	2.38308597	0.00766074
0.9090	2.80040674	2.80684544	0.00643869
1.0000	3.25762500	3.25762500	0

Table 4.2 The exact values, approximated values and absolute errors for uniform grid distribution when $\nu = 0.5$, $\Delta t = 0.01$ and $N = 12$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0.01000013	0.01000013	0
0.0909	0.19270041	0.19263093	6.95×10^{-5}
0.1818	0.38015651	0.38015054	5.98×10^{-6}
0.2727	0.57687633	0.57685359	2.27×10^{-5}
0.3636	0.78736774	0.78734136	2.64×10^{-5}
0.4545	1.01613865	1.01610370	3.50×10^{-5}
0.5455	1.26769693	1.26765669	4.02×10^{-5}
0.6364	1.54655048	1.54650135	4.91×10^{-5}
0.7273	1.85720719	1.85715572	5.15×10^{-5}
0.8182	2.20417494	2.20409980	7.51×10^{-5}
0.9090	2.59196162	2.59202369	6.21×10^{-5}
1.0000	3.02507512	3.02507512	0

If the constant velocity $\nu = 1.5$, the exact and the approximate values of the unknown function $u(x,t)$ will be acquired as follows.

Table 4.3 The exact values, approximated values and absolute errors for uniform grid distribution when $\nu = 1.5$, $\Delta t = 0.1$ and $N = 12$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0.30337500	0.30337500	0
0.0909	0.49579987	0.42846913	0.06733074
0.1818	0.70017064	0.69180979	0.00836085
0.2727	0.92099521	0.89717378	0.02382143
0.3636	1.16278146	1.13582656	0.02695490
0.4545	1.43003728	1.39527515	0.03476213
0.5455	1.72727057	1.68784431	0.03942626
0.6364	2.05898920	2.01141686	0.04757234
0.7273	2.42970107	2.38035063	0.04935044
0.8182	2.84391407	2.77198742	0.07192665
0.9090	3.30613608	3.36907932	0.06294324
1.0000	3.82087500	3.82087500	0

Table 4.4 The exact values, approximated values and absolute errors for uniform grid distribution when $\nu = 1.5$, $\Delta t = 0.01$ and $N = 12$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0.03000338	0.03000338	0
0.0909	0.21300614	0.21237783	6.28×10^{-4}
0.1818	0.40126059	0.40120472	5.59×10^{-5}
0.2727	0.59927462	0.59906769	2.07×10^{-4}
0.3636	0.81155612	0.81131644	2.38×10^{-4}
0.4545	1.04261297	1.04229613	3.17×10^{-4}
0.5455	1.29695307	1.29658871	3.64×10^{-4}
0.6364	1.57908431	1.57863979	4.45×10^{-4}
0.7273	1.89351457	1.89304926	4.65×10^{-4}
0.8182	2.24475174	2.24407257	6.79×10^{-4}
0.9090	2.63730371	2.63786688	5.63×10^{-4}
1.0000	3.07567838	3.07567838	0

Example 4.2 Solution of the Advection-Diffusion-Reaction Equation

$$u_t(x,t) + \nu u_x(x,t) = D u_{xx}(x,t), \quad 0 < x < 1, \quad t > 0,$$

$$u(0,t) = f_0(t), \quad u(1,t) = f_1(t), \quad t \geq 0,$$

$$u(x,0) = u_0(x)$$

where ν and D represent the constant velocity and the constant diffusion coefficient, respectively.

When the grid points x_i, $i = 1, 2, \ldots, N$, are applied to solve the given advection-diffusion-reaction equation, the unknown function $u(x,t)$ at each grid point can be represented as $U(x_i,t)$ and the equation becomes

$$U_t(x_i,t) = -\nu U_x(x_i,t) + D U_{xx}(x_i,t).$$

If the forward difference approximation given in Chapter 3 is applied for the derivative of the function with respect to t and the differential quadrature formula is applied for the first and the second order derivative with respect to x, it will be obtained the system of equations to be solve for the solution of the advection-diffusion-reaction equation like

$$\frac{U\left(x_i,t+\Delta t\right)-U\left(x_i,t\right)}{\Delta t}=-\nu\sum_{j=1}^{N}a_{ij}\,U\left(x_j,t\right)+D\sum_{j=1}^{N}w_{ij}^{(2)}\,U\left(x_j,t\right),$$

$$U\left(x_i,t+\Delta t\right)=U\left(x_i,t\right)+\Delta t\cdot(-\nu)\sum_{j=1}^{N}a_{ij}\,U\left(x_j,t\right)+\Delta t\cdot D\sum_{j=1}^{N}w_{ij}^{(2)}\,U\left(x_j,t\right).$$

where $w_{ij}^{(2)}$ represents the element of the matrix of the second order derivative for differential quadrature. Let us write the system explicitly like

$$\begin{aligned}
U\left(x_1,t+\Delta t\right)&=U\left(x_1,t\right)\\
&\quad+\Delta t\left\{\left[(-\nu)a_{11}+Dw_{11}^{(2)}\right]U\left(x_1,t\right)+\cdots+\left[(-\nu)a_{1N}+Dw_{1N}^{(2)}\right]U\left(x_N,t\right)\right\},\\
U\left(x_2,t+\Delta t\right)&=U\left(x_2,t\right)\\
&\quad+\Delta t\left\{\left[(-\nu)a_{21}+Dw_{21}^{(2)}\right]U\left(x_1,t\right)+\cdots+\left[(-\nu)a_{2N}+Dw_{2N}^{(2)}\right]U\left(x_N,t\right)\right\},\\
&\quad\vdots\\
U\left(x_N,t+\Delta t\right)&=U\left(x_N,t\right)\\
&\quad+\Delta t\left\{\left[(-\nu)a_{N1}+Dw_{N1}^{(2)}\right]U\left(x_1,t\right)+\cdots+\left[(-\nu)a_{NN}+Dw_{NN}^{(2)}\right]U\left(x_N,t\right)\right\}.
\end{aligned}$$

Expressing the system of equations in the form of matrix and vector, it will be obtained the following

$$\left[U\left(x_i,t+\Delta t\right)\right]_{N\times 1}=\left[I+\Delta t\left(-\nu a_{ij}+Dw_{ij}^{(2)}\right)\right]_{N\times N}\left[U\left(x_j,t\right)\right]_{N\times 1}.$$

If the weighted coefficient matrix is stated as $\left[A_{ij}\right]_{N\times N}=\left[I+\Delta t\left(-\nu a_{ij}+Dw_{ij}^{(2)}\right)\right]_{N\times N}$, the system will be summarized as in the form of

$$\left[U\left(x_i,t+\Delta t\right)\right]_{N\times 1}=\left[A_{ij}\right]_{N\times N}\left[U\left(x_j,t\right)\right]_{N\times 1}. \tag{4.8}$$

Finally, the solution of the system gives us the value of the unknown function at each grid points in the next time step. Let us show how to apply Eq. (4.8) to the problem:

For $t=0$, let us find the values of the unknown function $U\left(x_i,0+\Delta t\right)$ at the next time step using the initial condition as follows:

$$u(0,t)=U\left(x_1,t\right)=f_0\left(\Delta t\right)\quad\text{and}\quad u(1,t)=U\left(x_N,t\right)=f_1\left(\Delta t\right),$$

$$U\left(x_i,0\right)=u_0\left(x_i\right)$$

After the substitution of boundary conditions $U(x_1,t)=f_0(\Delta t)$ and $U(x_N,t)=f_1(\Delta t)$ in the first and the last rows of $\left[A_{ij}\right]$, respectively, the new system becomes

$$\begin{pmatrix} U(x_1,0+\Delta t) \\ U(x_2,0+\Delta t) \\ \vdots \\ U(x_{N-1},0+\Delta t) \\ U(x_N,0+\Delta t) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{(N-1)1} & A_{(N-1)2} & \cdots & A_{(N-1)N} \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} f_0(\Delta t) \\ U(x_2,0) \\ \vdots \\ U(x_{N-1},0) \\ f_1(\Delta t) \end{pmatrix}.$$

Finally, by substituting the initial condition $U(x_i,0)=u_0(x_i)$ into the system above, the new system of equations can be shown as follows:

$$\begin{pmatrix} U(x_1,0+\Delta t) \\ U(x_2,0+\Delta t) \\ \vdots \\ U(x_{N-1},0+\Delta t) \\ U(x_N,0+\Delta t) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{(N-1)1} & A_{(N-1)2} & \cdots & A_{(N-1)N} \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} f_0(\Delta t) \\ u_0(x_2) \\ \vdots \\ u_0(x_{N-1}) \\ f_1(\Delta t) \end{pmatrix}.$$

The solution of the system above gives the approximated values of unknown function $U(x_i,0+\Delta t)$ at the next time step by PDQ method.

Firstly, let us examine the following case:

When the initial and boundary condition functions are given as

$$f_0(t)=e^{-0.09t}$$

$$f_1(t)=e^{1-0.09t}$$

$$u_0(x)=e^{9x}$$

and also the constant velocity and the constant diffusion coefficient are taken as

$\nu=0.1$ and $D=0.01$

Then, the close-form solution will be acquired as follows [27].

$$u(x,t)=e^{9x-0.09t}$$

Applying the uniform grid distribution for $N=11$ given in Chapter 2 for different values of time steps like $\Delta t=0.1$ and $\Delta t=0.01$, respectively, the following results can be obtained. In these tables, u_E and U_A represent the exact and the approximate values of the unknown function $u(x,t)$, respectively.

Table 4.5 The exact values, approximated values and absolute errors for uniform grid distribution when $\nu = 0.1$, $D = 0.01$, $\Delta t = 0.1$ and $N = 11$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0.99104040	0.99104040	0
0.1000	2.43756600	2.67471313	0.23714713
0.2000	5.99544491	5.95163156	0.04381336
0.3000	14.7464150	14.7591690	0.01275408
0.4000	36.2703281	36.2632083	0.00711978
0.5000	89.2106119	89.2092736	0.00133828
0.6000	219.422699	219.415974	0.00672489
0.7000	539.692752	539.649118	0.04363426
0.8000	1327.42997	1327.55684	0.12687128
0.9000	3264.95089	3261.57138	3.37950554
1.0000	8030.48336	8030.48336	0

Table 4.6 The exact values, approximated values and absolute errors for uniform grid distribution when $\nu = 0.1$, $D = 0.01$, $\Delta t = 0.01$ and $N = 11$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0.99910040	0.99910040	0
0.1000	2.45739046	2.44897875	8.41×10^{-3}
0.2000	6.04420523	6.04562141	1.42×10^{-3}
0.3000	14.8663460	14.8659397	4.06×10^{-3}
0.4000	36.5653109	36.5654573	1.46×10^{-4}
0.5000	89.9361523	89.9360346	1.18×10^{-4}
0.6000	221.207240	221.207190	4.96×10^{-5}
0.7000	544.082016	544.081782	2.34×10^{-4}
0.8000	1338.22582	1338.22570	1.22×10^{-4}
0.9000	3291.50439	3291.48262	2.18×10^{-2}
1.0000	8095.79443	8095.79443	0

Secondly, the following case will be analyzed:

If the initial and boundary condition functions are given as

$$f_0(t) = \sqrt{\frac{20}{t+20}} \exp\left[-\frac{2(2t+5)^2}{5(t+20)}\right]$$

$$f_1(t) = \sqrt{\frac{20}{t+20}} \exp\left[-\frac{(4t+5)^2}{10(t+20)}\right]$$

$$u_0(x) = \exp\left[-\frac{(x-2)^2}{8}\right]$$

and also the constant velocity and the constant diffusion coefficient are taken as

$$\nu = 0.8 \quad \text{and} \quad D = 0.1$$

Then, the close-form solution will be acquired as follows [28].

$$u(x,t) = \sqrt{\frac{20}{t+20}}\exp\left[-\frac{(x-2-0.8t)^2}{0.4(t+20)}\right]$$

Applying the uniform grid distribution for $N = 11$ given in Chapter 2 for different values of time steps like $\Delta t = 0.1$ and $\Delta t = 0.01$, respectively, the following results can be obtained. In these tables, u_E and U_A represent the exact and the approximate values of the unknown function $u(x,t)$, respectively.

Table 4.7 The exact values, approximated values and absolute errors for uniform grid distribution when $\nu = 0.8$, $D = 0.1$, $\Delta t = 0.1$ and $N = 11$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0.58239683	0.58239683	0
0.1000	0.61256164	0.59781816	1.47×10^{-2}
0.2000	0.64268809	0.64362132	9.33×10^{-4}
0.3000	0.67262092	0.67247752	1.43×10^{-4}
0.4000	0.70219892	0.70229840	9.95×10^{-5}
0.5000	0.73125629	0.73135208	9.58×10^{-5}
0.6000	0.75962409	0.75980868	1.85×10^{-4}
0.7000	0.78713190	0.78729061	1.59×10^{-4}
0.8000	0.81360941	0.81437114	7.62×10^{-4}
0.9000	0.83888819	0.83036167	8.53×10^{-3}
1.0000	0.86280342	0.86280342	0

Table 4.8 The exact values, approximated values and absolute errors for uniform grid distribution when $\nu = 0.8$, $D = 0.1$, $\Delta t = 0.01$ and $N = 11$

x_i	u_E	U_A	$\lvert u_E - U_A \rvert$
0	0.60410572	0.60410572	0
0.1000	0.63439688	0.63424863	1.48×10^{-4}
0.2000	0.66454430	0.66455382	9.52×10^{-6}
0.3000	0.69438710	0.69438576	1.34×10^{-6}
0.4000	0.72375929	0.72376041	1.12×10^{-6}
0.5000	0.75249127	0.75249235	1.08×10^{-6}
0.6000	0.78041137	0.78041335	1.97×10^{-6}
0.7000	0.80734753	0.80734926	1.73×10^{-6}
0.8000	0.83312901	0.83313665	7.65×10^{-6}
0.9000	0.85758820	0.85750480	8.34×10^{-5}
1.0000	0.88056242	0.88056242	0

CHAPTER 5

CONCLUSION

In this thesis, polynomial based differential quadrature method presented in details. The method was applied to solve not only ordinary but also partial differential equations from any order using different grid point distributions and high accurate results were obtained. Subsequently, PDQ method was used together with the finite difference method to acquire numerical solutions of partial differential equations. It was seen that both methods in company give results convergent to analytical solution. Afterwards, numerical solutions of advection-diffusion-reaction equations by PDQ were examined. In the solution process, finite difference method was used in time direction while PDQ was applied in position direction. When the results were analyzed by comparing the close-form solutions, it was seen that the method gives accurate results. Besides, it was observed that reducing the time interval increases convergence to the close-form solutions.

In the applications of this study, the highest number of grid points used is 13. By taking into consideration requiring small number of grid points and simplicity in applications, compared to the conventional methods, it can be stated that the differential quadrature method is a good choice to solve linear ODEs and PDEs. As a future project of this work, the differential quadrature method can be used to solve non-linear ODEs and PDEs. Additionally, the solutions of PDEs can be obtained by applying the differential quadrature method in all the directions (differentiations with respect to $t,\ x,\ y,\ z,\ldots$ etc.) of the equations.

REFERENCES

[1] Zhang, Y. and Zong, Z., *Advanced Differential Quadrature Methods*, CRC Press, USA, 2009.

[2] Bellman, R.E. and Casti, J., "Differential Quadrature and Long-Term Integration", *Journal of Mathematical Analysis and Applications,* Vol. 34(2), pp. 235-238, 1971.

[3] Bellman, R.E., Kashef, B.G. and Casti J., "Differential Quadrature: A Technique for the Rapid Solution of Nonlinear Partial Differential Equations", *Journal of Computational Physics*, Vol. 10(1), pp. 40-52, 1972.

[4] Mingle, J. O., "The Method of Differential Quadrature for Transient Nonlinear Diffusion", *Journal of Mathematical Analysis and Applications*, Vol. 60(3), pp. 559-569, 1977.

[5] Civan, F. and Sliepcevich, C.M., "Application of Differential Quadrature to Transport Processes", *Journal of Mathematical Analysis and Applications*, Vol. 93(1), pp. 206-221, 1983.

[6] Quan, J. R. and Chang, C.T., "New Insights in Solving Distributed System Equations by the Quadrature Method—I. Analysis", *Computers & Chemical Engineering*, Vol. 13(7), pp. 779-788, 1989.

[7] Quan, J. R. and Chang, C.T., "New Insights in Solving Distributed System Equations by the Quadrature Method—II. Numerical Experiments", *Computers & Chemical Engineering*, Vol. 13(9), pp. 1017-1024, 1989.

[8] Shu, C. and Richards, B. E., "High Resolution of Natural Convection in a Square Cavity by Generalized Differential Quadrature", *Advances in Numeric Methods in Engineering: Theory and Applications*, Vol. 2, pp. 978-985, 1990.

[9] Shu, C., *Generalized Differential-Integral Quadrature and Application to the Simulation of Incompressible Viscous Flows Including Parallel Computation*, Ph.D. Thesis, University of Glasgow, 1991.

[10] Shu, C., *Differential Quadrature and its Application in Engineering*, London, 2000.

[11] Shu, C. and Xue, H., "Solution of Helmholtz Equation by Differential Quadrature Method", *Computer Methods in Applied Mechanics and Engineering*, Vol. 175(1), pp. 203-212, 1999.

[12] Bert, C. W. and Malik, M., "Differential Quadrature Method in Computational Mechanics: A Review", *Applied Mechanics Reviews*, Vol. 49(1), pp. 1-28, 1996.

[13] Zhong, H., "Spline-Based Differential Quadrature for Fourth Order Differential Equations and its Application to Kirchhoff Plates", *Applied Mathematical Modelling*, Vol. 28(4), pp. 353-366, 2004.

[14] Zhong, H. and Guo, Q., "Vibration Analysis of Rectangular Plates with Free Corners Using Spline-Based Differential Quadrature", *Shock and* Vibration, Vol. 11(2), pp. 119-128, 2004.

[15] Shu, C., Ding, H. and Yeo, K.S., "Solution of Partial Differential Equations by a Global Radial Basis Function-Based Differential Quadrature Method", *Engineering Analysis with Boundary Elements*, Vol. 28(10), pp. 1217-1226, 2004.

[16] Zong, Z., Lam, K.Y. and Zhang, Y.Y., "A Multi-Domain Differential Quadrature Approach to Plane Elastic Problems with Material Discontinuity", *Mathematical and Computer Modelling*, Vol. 41(4), pp. 539-553, 2005.

[17] Shu, C., Chen, W., Xue, H. and Du, H., "Numerical Study of Grid Distribution Effect on Accuracy of DQ Analysis of Beams and Plates by Error Estimation of Derivative Approximation", *International Journal for Numerical Methods in Engineering*, Vol. 51(2), pp. 159-179, 2001.

[18] Makbule Akman, *Differential Quadrature method for Time-Dependent Diffusion Equation*, M.S. Thesis, Middle East Technical University, 2003.

[19] Alper Korkmaz, *Numerical Solutions of Some Nonlinear Partial Differential Equations by Differential Quadrature Method*, M.S. Thesis, Osmangazi University, 2006.

[20] Yücel, U., "Approximations of Sturm–Liouville Eigenvalues Using Differential Quadrature (DQ) Method", *Journal of Computational and Applied Mathematics*, Vol. 192(2), pp. 310-319, 2006.

[21] Gürarslan, G. and Sari, M., "Numerical Solutions of Linear and Nonlinear Diffusion Equations by a Differential Quadrature Method (DQM)", *International Journal for Numerical Methods in Biomedical Engineering*, Vol. 27(1), pp. 69-77, 2011.

[22] Gizem Temelcan, *Solution of Differential Equations Using Differential Quadrature Method*, M.S. Thesis, Yildiz Technical University, 2013.

[23] Selda Kurtar, *Differential Quadrature Method and Its Applications*, M.S. Thesis, Fatih University, 2014.

[24] Irk, D., Dağ, İ. and Tombul, M., "Extended Cubic B-spline Solution of the Advection-Diffusion Equation", *KSCE Journal of Civil Engineering*, Vol. 19(4), pp. 929-934, 2015.

[25] Jost, Jürgen, “Partial Differential Equations,” Series of Graduate Texts in Mathematics 214, 3rd Edition, Springer 2013.

[26] Wazwaz, Abdul-Majid, “Partial Differential Equations and Solitary Waves Theory,” Series of Nonlinear Physical Sciences, Springer 2009.

[27] Gopaul, A., Cheeneebash, J. and Baurhoo, K., "A Comparison of Recent Methods for solving a model 1D Convection Diffusion Equation", *World Academy of Science, Engineering and Technology*, Vol. 81, pp. 59-63, 2011.

[28] Dehghan, M., "On the Numerical Solution of the One-Dimensional Convection-Diffusion Equation", *Mathematical Problems in Engineering*, Vol. 2005(1), pp. 61-74, 2005.

APPENDIX A

MATLAB CODES OF THE EXAMPLES OF CHAPTER 2

Variables in MatLab Codes

N : The number of grid points

X : The vector formed by the values of grid points in x-direction

Ord : The highest order of derivative of the equation given in the question

W(: , : , 1) : The weighted coefficients' matrix of the first order derivative

W(: , : , 2) : The weighted coefficients' matrix of the second order derivative

W(: , : , n) : The weighted coefficients' matrix of the n'th order derivative

C : The right-hand-side function of the equation given in the question

YExact: The vector formed by the exact solutions at the grid points

YApp : The vector formed by the approximated solutions at the grid points

AErr : Difference between exact and approximated solutions at the grid points

Example 2.1:

```
N=8;
Ord=1;
i=1:N;
X=(i-1)/(N-1);
a=zeros(N,N);
for m=1:N
    for n=1:N
        if m==n
            continue
        else
```

```
            q=X(m)-X;
            q(m)=1;
            p=prod(q);
            s=X(n)-X;
            s(n)=1;
            r=(X(m)-X(n))*prod(s);
            a(m,n)=p/r;
        end
    end
end
for k=1:N
    a(k,k)=-sum(a(k,:));
end
W(:,:,1)=a;
A=(W(:,:,1)-4*eye(N));
A(1,:)=zeros(1,N);
A(1)=1;
X=X';
C=X;
C(1)=1;
X
YExact=((-1/16)*(4*X+1))+(17/16)*exp(4*X)
YApp=A\C
AErr=abs(YExact-YApp)
```

Example 2.2:

```
N=10;
Ord=2;
i=1:N;
X=(i-1)/(N-1);
a=zeros(N,N);
for m=1:N
    for n=1:N
```

```
        if m==n
            continue
        else
            q=X(m)-X;
            q(m)=1;
            p=prod(q);
            s=X(n)-X;
            s(n)=1;
            r=(X(m)-X(n))*prod(s);
            a(m,n)=p/r;
        end
    end
end
for k=1:N
    a(k,k)=-sum(a(k,:));
end
W(:,:,1)=a;
for m=2:Ord
    W(:,:,m)=zeros(N,N);
    for i=1:N
        for j=1:N
            if i==j
                continue
            else
                W(i,j,m)=m*(a(i,j)*W(i,i,(m-1))-W(i,j,(m-1))/(X(i)-X(j)));
            end
        end
    end
    for k=1:N
        W(k,k,m)=-sum(W(k,:,m));
    end
end
A=W(:,:,1);
for i=1:N
```

```
    A(i,:)=(X(i)+1)*A(i,:);
end
A=A+W(:,:,2)-2*eye(N);
A(1,:)=zeros(1,N);
A(N,:)=zeros(1,N);
A(1,1)=1;
A(N,N)=1;
X=X';
C=(1-X.^2).*exp(-X);
C(1)=-1;
C(N)=0;
X
YExact=(X-1).*exp(-X)
YApp=A\C
AErr=abs(YExact-YApp)
```

Example 2.3:

```
N=9;
Ord=2;
i=1:N;
X=(i-1)/(N-1);
a=zeros(N,N);
for m=1:N
    for n=1:N
        if m==n
            continue
        else
            q=X(m)-X;
            q(m)=1;
            p=prod(q);
            s=X(n)-X;
            s(n)=1;
            r=(X(m)-X(n))*prod(s);
```

```
            a(m,n)=p/r;
        end
    end
end
for k=1:N
    a(k,k)=-sum(a(k,:));
end
W(:,:,1)=a;
for m=2:Ord
    W(:,:,m)=zeros(N,N);
    for i=1:N
        for j=1:N
            if i==j
                continue
            else
                W(i,j,m)=m*(a(i,j)*W(i,i,(m-1))-W(i,j,(m-1))/(X(i)-X(j)));
            end
        end
    end
    for k=1:N
        W(k,k,m)=-sum(W(k,:,m));
    end
end
A=W(:,:,2)-5*a+6*eye(N);
A(1,:)=zeros(1,N);
A(1,1)=1;
A(N,:)=a(1,:);
C=zeros(N,1);
C(N)=-1;
X=X';
X
YExact=exp(2*X)-exp(3*X)
YApp=A\C
AErr=abs(YExact-YApp)
```

Example 2.4:

```
N=13;
Ord=4;
i=1:N;
X=(i-1)/(N-1);
a=zeros(N,N);
for m=1:N
    for n=1:N
        if m==n
            continue
        else
            q=X(m)-X;
            q(m)=1;
            p=prod(q);
            s=X(n)-X;
            s(n)=1;
            r=(X(m)-X(n))*prod(s);
            a(m,n)=p/r;
        end
    end
end
for k=1:N
    a(k,k)=-sum(a(k,:));
end
W(:,:,1)=a;
for m=2:Ord
    W(:,:,m)=zeros(N,N);
    for i=1:N
        for j=1:N
            if i==j
                continue
            else
                W(i,j,m)=m*(a(i,j)*W(i,i,(m-1))-W(i,j,(m-1))/(X(i)-X(j)));
```

```
            end
         end
      end
      for k=1:N
         W(k,k,m)=-sum(W(k,:,m));
      end
end
A=W(:,:,Ord)-eye(N);
A(1,:)=zeros(1,N);
A(1,1)=1;
A(2,:)=a(1,:);
A((N-1),:)=W(1,:,2);
A(N,:)=W(1,:,3);
C=zeros(N,1);
C(1)=7/2;
C(2)=-4;
C(N-1)=5/2;
C(N)=-2;
X=X';
X
YExact=3*exp(-X)+(1/2)*cos(X)-sin(X)
YApp=A\C
AErr=abs(YExact-YApp)
```

APPENDIX B

MATLAB CODES OF THE EXAMPLES OF CHAPTER 3

Variables in MatLab Codes

N : The number of grid points

X : The vector formed by the values of grid points in x-direction

Ord : The highest order of derivative of the equation given in the question

DT : Time step

W(: , : , 1) : The weighted coefficients' matrix of the first order derivative

W(: , : , 2) : The weighted coefficients' matrix of the second order derivative

W(: , : , n) : The weighted coefficients' matrix of the n'th order derivative

B : The right-hand-side function of the equation given in the question

U0 : Initial value condition function

UExact: The vector formed by the exact solutions at the grid points

UApp : The vector formed by the approximated solutions at the grid points

UErr : Difference between exact and approximated solutions at the grid points

Example 3.1:

```
N=10;
Ord=2;
DT=0.1;
i=1:N;
X=(i-1)/(N-1);
a=zeros(N,N);
for m=1:N
    for n=1:N
```

```
        if m==n
            continue
        else
            q=X(m)-X;
            q(m)=1;
            p=prod(q);
            s=X(n)-X;
            s(n)=1;
            r=(X(m)-X(n))*prod(s);
            a(m,n)=p/r;
        end
    end
end
for k=1:N
    a(k,k)=-sum(a(k,:));
end
W(:,:,1)=a;
for m=2:Ord
    W(:,:,m)=zeros(N,N);
    for i=1:N
        for j=1:N
            if i==j
                continue
            else
                W(i,j,m)=m*(a(i,j)*W(i,i,(m-1))-W(i,j,(m-1))/(X(i)-X(j)));
            end
        end
    end
    for k=1:N
        W(k,k,m)=-sum(W(k,:,m));
    end
end
A=eye(N)+DT*W(:,:,2);
A(1,:)=zeros(1,N);
```

```
A(1,1)=1;
A(N,:)=zeros(1,N);
A(N,N)=1;
X=X';
U0=(-1/2)*sin(3*pi*X)+(3/2)*sin(pi*X);
U0(1)=0;
U0(N)=0;
X
UExact=(3/2)*sin(pi*X)*exp(-(pi^2)*DT)-(1/2)*sin(3*pi*X)*exp(-9*(pi^2)*DT)
UApp=A*U0
UErr=abs(UExact-UApp)
```

Example 3.2:

```
N=11;
Ord=2;
DT=0.03;
i=1:N;
X=pi*(i-1)/(N-1);
a=zeros(N,N);
for m=1:N
    for n=1:N
        if m==n
            continue
        else
            q=X(m)-X;
            q(m)=1;
            p=prod(q);
            s=X(n)-X;
            s(n)=1;
            r=(X(m)-X(n))*prod(s);
            a(m,n)=p/r;
        end
    end
```

```
end
for k=1:N
    a(k,k)=-sum(a(k,:));
end
W(:,:,1)=a;
for m=2:Ord
    W(:,:,m)=zeros(N,N);
    for i=1:N
        for j=1:N
            if i==j
                continue
            else
                W(i,j,m)=m*(a(i,j)*W(i,i,(m-1))-W(i,j,(m-1))/(X(i)-X(j)));
            end
        end
    end
    for k=1:N
        W(k,k,m)=-sum(W(k,:,m));
    end
end
A=eye(N)+DT*W(:,:,2);
A(1,:)=zeros(1,N);
A(1,1)=1;
A(N,:)=zeros(1,N);
A(N,N)=1;
X=X';
U0=3*sin(X);
U0(1)=0;
U0(N)=0;
B=2*DT*sin(X);
B(1)=0;
B(N)=0;
X
UExact=exp(-DT)*sin(X)+2*sin(X)
```

```
UApp=A*U0+B
UErr=abs(UExact-UApp)
```

APPENDIX C

MATLAB CODES OF THE EXAMPLES OF CHAPTER 4

Variables in MatLab Codes

N : The number of grid points

X : The vector formed by the values of grid points in x-direction

Ord : The highest order of derivative of the equation given in the question

DT : Time step

v : Constant velocity

D : Constant diffusion coefficient

W(: , : , 1) : The weighted coefficients' matrix of the first order derivative

W(: , : , 2) : The weighted coefficients' matrix of the second order derivative

W(: , : , n) : The weighted coefficients' matrix of the n'th order derivative

U0 : Initial value condition function

UExact: The vector formed by the exact solutions at the grid points

UApp : The vector formed by the approximated solutions at the grid points

UErr : Difference between exact and approximated solutions at the grid points

Example 4.1:

```
N=12;
Ord=2;
DT=0.1;
v=0.5;
i=1:N;
X=(i-1)/(N-1);
a=zeros(N,N);
```

```
for m=1:N
   for n=1:N
      if m==n
         continue
      else
         q=X(m)-X;
         q(m)=1;
         p=prod(q);
         s=X(n)-X;
         s(n)=1;
         r=(X(m)-X(n))*prod(s);
         a(m,n)=p/r;
      end
   end
end
for k=1:N
   a(k,k)=-sum(a(k,:));
end
W(:,:,1)=a;
for m=2:Ord
   W(:,:,m)=zeros(N,N);
   for i=1:N
      for j=1:N
         if i==j
            continue
         else
            W(i,j,m)=m*(a(i,j)*W(i,i,(m-1))-W(i,j,(m-1))/(X(i)-X(j)));
         end
      end
   end
   for k=1:N
      W(k,k,m)=-sum(W(k,:,m));
   end
end
```

```
A=eye(N)+DT*v*W(:,:,1);
A(1,:)=zeros(1,N);
A(1,1)=1;
A(N,:)=zeros(1,N);
A(N,N)=1;
X=X';
U0=X.^3+2*X;
U0(1)=(v*DT)^3+2*v*DT;
U0(N)=(1+v*DT)^3+2*(1+v*DT);
X
UExact=(X+v*DT).^3+2*(X+v*DT)
UApp=A*U0
UErr=abs(UExact-UApp)
```

Example 4.2:

```
N=11;
Ord=2;
DT=0.1;
v=0.8;
D=0.1;
i=1:N;
X=(i-1)/(N-1);
a=zeros(N,N);
for m=1:N
   for n=1:N
      if m==n
         continue
      else
         q=X(m)-X;
         q(m)=1;
         p=prod(q);
         s=X(n)-X;
         s(n)=1;
```

```
        r=(X(m)-X(n))*prod(s);
        a(m,n)=p/r;
      end
    end
end
for k=1:N
    a(k,k)=-sum(a(k,:));
end
W(:,:,1)=a;
for m=2:Ord
    W(:,:,m)=zeros(N,N);
    for i=1:N
      for j=1:N
        if i==j
          continue
        else
          W(i,j,m)=m*(a(i,j)*W(i,i,(m-1))-W(i,j,(m-1))/(X(i)-X(j)));
        end
      end
    end
    for k=1:N
      W(k,k,m)=-sum(W(k,:,m));
    end
end
A=eye(N)+DT*(-v*W(:,:,1)+D*W(:,:,2));
A(1,:)=zeros(1,N);
A(1,1)=1;
A(N,:)=zeros(1,N);
A(N,N)=1;
X=X';
U0=exp(-((X-2).^2)/8);
U0(1)=sqrt(20/(DT+20))*exp(-(2*((2*DT+5)^2))/(5*(DT+20)));
U0(N)=sqrt(20/(DT+20))*exp(-((4*DT+5)^2)/(10*(DT+20)));
X
```

```
UExact=sqrt(20/(DT+20))*exp(-((X-2-0.8*DT).^2)/(0.4*(DT+20)))
UApp=A*U0
UErr=abs(UExact-UApp)
```