An-Najah National University

Faculty of Graduate Studies

Numerical Methods for Solving Third Order Two-Point Boundary Value Problems

By

Saja Jamal Abu Shanab

Supervisor

Dr. Samir Matar

This Thesis is Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Computational Mathematics, Faculty of Graduate Studies, An -Najah National University, Nablus, Palestine.

2017

Numerical Methods for Solving Third Order Two-Point Boundary Value Problems

By

Saja Jamal Abu Shanab

This thesis was defended successfully on 13/7/2017 and approved by:

Defense Committee Members	**Signature**
- **Dr. Samir Matar/ Supervisor**	……..
- **Dr. Abdulhaleem Ziqan/ External Examiner**	…….....
- **Dr. Mohammad Assad/ Internal Examiner**	……..

Dedication

I dedicate this thesis to my mother, father, my brothers and my family. Without their support, love, patience and understanding, nothing will be complete.

Thanks all.

Acknowledgment

First and before all, thanks and praises to Allah for blessing me much more I deserve.

Thanks to my supervisor Dr. Samir Matar for his understanding and support, I really appreciate his advices and assistance. Finally, much of love and thanks for my mother, my father and my family for their support and love.

الإقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل عنوان :

Numerical Methods for Solving Third Order Two-Point Boundary Value Problems

أقر بأن ما اشتملت عليه هذه الرسالة إنما هي نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وأن هذه الرسالة ككل، أو أي جزء منها لم يقدم من قبل لنيل أية درجة علمية أو بحث علمي أو بحثي لدى أية مؤسسة تعليمية أو بحثية أخرى.

Declaration

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's Name: **اسم الطالب:**

Signature: **التوقيع:**

Date: **التاريخ:**

Table of contents

List of Figures

List of Tables

Numerical Methods for solving Third-Order Two Point Boundary Value Problems

By
Saja Jamal Abu Shanab
Supervisor
Dr. Samir Matar

Abstract

Third order two point boundary value problems have clearly emerged in many branches of science, for example technology, engineering, physics and many others. So based on the importance of third order two point boundary value problems, new efficient and more accurate numerical methods were discussed, studied and analyzed.

These numerical methods are Shooting Method, Finite Difference Method, Quartic B-Spline Method, Pade Approximation and Rational Chebyshev Approximation Method. Each method has been studied and implemented with examples and A MATLAB code was written for each method to obtain very accurate results. The Numerical results will be compared to determine the best method which is the fastest and most accurate.

Chapter One

Introduction

Chapter One

1.1. Introduction

Boundary value problem in general arises in different fields of science, engineering, technology, control, optimization theory, draining and coating flows and various dynamic systems. So there are different problems concerning two-point boundary value problems which have drawn the attention of researchers throughout the world. And also boundary value problems (BVPs) for higher order ordinary differential equations are frequently encountered in several applications [15].

One of these problems where the boundary conditions are specified at two points, these points are called endpoints. In science research, therefore faster and accurate numerical methods for solving two-point boundary value problems are very important. In general it's difficult to find analytic solutions for such problems, so solutions have to be obtained by numerical methods.

The aim of our work is to study and develop some new numerical methods for solving third order two-point boundary value problems with higher accuracy and writing complete algorithms and computer programs for each method that will be developed in our work.

After taking some examples on third order two-point boundary value problems, the results that will be obtained from each numerical method that have been developed in our work will be compared to determine the best method with the highest accuracy.

1.2. Nth Order Two-Point Boundary Value Problems

A boundary value problem is a differential equation with a set of constraints called boundary conditions. In these problems the boundary conditions are specified at two points.

The following equation represented the general formula for linear nth order two-point boundary value problem [26], [31]:

$$y^{(n)}(x) = p_n y^{(n-1)}(x) + \ \ldots + p_2 y'(x) + p_1 y(x) + p_0, a \le x \le b$$

where p_i are functions of x for $i = 0,1, \ldots, n$, with specific boundary conditions.

And the following equation expressed the general formula for nonlinear nth order two-point boundary value problems with specific boundary conditions

$$y^{(n)} = f\left(x, y, y', \ldots, y^{(n-1)}\right), a \le x \le b$$

The specific boundary conditions can be expressed in several ways, one of them is in vector-matrix notation as

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \ldots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} y(a) \\ y'(a) \\ \vdots \\ y^{(n)}(a) \end{bmatrix} + \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ b_{21} & \ldots & b_{2n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{bmatrix} \begin{bmatrix} y(b) \\ y'(b) \\ \vdots \\ y^{(n)}(b) \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}$$

Where a_{ij}, b_{ij} and r_i are given and defined in the problem for $i = 1,2, \ldots, n$ and $j = 1,2, \ldots, n$. The solution for the previous nth order two-point boundary value problem is the function $y(x)$ that satisfies the given boundary conditions.

1.3. Third Order Two-Point Boundary Value Problems

A type of nth order two-point boundary value problem where the order $n = 3$. The following equation represents the general formula for linear third

order two-point boundary value problem with some boundary conditions

$$y^{(3)}(x) = p(x)y''(x) + q(x)y'(x) + r(x)y(x) + s(x), a \le x \le b$$

The following equation expressed the general formula for nonlinear third order boundary value problem with some boundary conditions

$$y^{(3)} = f(x, y, y', y''), a \le x \le b$$

The boundary conditions can be expressed in vector-matrix notation as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} y(a) \\ y'(a) \\ y''(a) \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} y(b) \\ y'(b) \\ y''(b) \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

Where a_{ij} and b_{ij} are constants, for $i = 1,2,3$ and $j = 1,2,3$. r_1, r_2 and r_3 are the given boundary conditions.

For example if the given boundary conditions are as the following

$$y(a) = \alpha, y'(b) = \beta,\ y''(a) = \gamma$$

All constants will put to zeros except a_{11}, a_{33} and b_{22} which all must equal ones, where $r_1 = \alpha, r_2 = \beta$ and $r_3 = \gamma$ as the following

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y(a) \\ y'(a) \\ y''(a) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y(b) \\ y'(b) \\ y''(b) \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

Previous works

As third order two-point boundary value problems arise in different areas of applied mathematics, technology, physics and engineering, many works have been published on studying and developing numerical methods for solving third order two-point boundary value problems. F.A. Abd El-Salam, A.A. El-Sabbagh and Z.A. Zaki [1], Ghazala Akram and Imran Talib [6], Talaat S. El-Danaf [14] and S. ul Islam, I. A. Trmizi and M. A. khan [17] have used quartic non-polynomial spline technique for solving

third order two-point boundary value problem where the numerical solution of a third order two-point boundary value problem is represented by quartic non-polynomial spline function $Q_i(x)$, where $Q_i(x) = S_i$ be the approximated solution to the third order two-point boundary value problem. Any quartic non-polynomial spline has the following form

$$Q_i(x) = a_i \cos k(x - x_i) + b_i \sin k(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i) + e_i, i = 0, 1, \dots, N \qquad (1.4.1)$$

They have used equation (1.4.1) and its first and third derivatives to build a system of $N + 1$ linear equations with $N + 1$ unknown variables. N. N. Abd Hamid, A. Abd Majid and A. I. Md. Ismail [2] have used Quartic B-spline interpolation method for solving linear two-point boundary value problem and Y. Gupta and P. k. Srivastava [16] have used cubic B-spline for solving fourth order two-point boundary value problem, this give a help for solving the same problem but of order three. E.A. Al-Said and M.A. Noor [4], G.B. Loghmani and M. Ahmadinia [20] and X. Zhang [29] have developed a method using cubic B-spline to construct an approximation solution for third order linear and nonlinear boundary value problems.

A. S. Abdullah, Z. A. Majid, and N. Senu [3] have solved third order nonlinear two-point boundary value problem using fifth order block method where the advantage of this method is to solve boundary value problem without reduce it to a system of first order ordinary differential equations. The fifth order two-point block method where the interval $[a, b]$ is divided into a series of blocks with each block containing two points, this

method will solve the nonlinear third order boundary value problem by shooting technique using constant step size.

1.4. Applications There are many engineering and science applications on third order two-point boundary value problem that attract attention of scientists and researchers.

1.4.1. Hliemenz Magnetic Flow

One of the applications on third order two-point boundary value problems is Hliemenz Magnetic Flow [21] where Hliemenz flow means that the velocity of the incoming fluid is perpendicular to a plane surface. In addition, if the fluid is electrically conducting, then the flow called Hliemenz Magnetic Flow. The solution of this problem is important since it's one of the few exact solutions of Navier-Stokes equations in magneto-hydrodynamics.

The Navier-Stokes equations for such flows are

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \qquad (1.5.1)$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = a^2 x + \nu\frac{\partial^2 u}{\partial y^2} + \rho\sigma B^2(ax - u) \qquad (1.5.2)$$

With the following boundary conditions

$$y = 0: u = 0, v = 0$$

$$y = \infty: u = ax$$

Where

u : The x component of the velocity

υ : The y component of the velocity

ν : The viscosity

ρ : The density

a : Constant characteristic of the incoming flow

σ : The electrical conductivity

B : The magnetic induction

And the first two boundary conditions are based on the physical observations that there is neither slip nor mass transfer on the surface, whereas the boundary condition at infinity means that the velocity of the fluid approaches a linear relation with x.

Hliemenz separated the variables by assuming that

$$u = ax\frac{\partial F}{\partial \eta}$$

And

$$v = -\sqrt{av}F(\eta)$$

Where

$$\eta = \sqrt{a/v}\, y$$

It can be shown that equation (1.5.1) will be satisfied identically and equation (1.5.2) becomes as

$$\frac{d^3F}{d\eta^3} + F\frac{d^2F}{d\eta^2} + 1 - (\frac{dF}{d\eta})^2 + M\left(1 - \frac{dF}{d\eta}\right) = 0 \qquad (1.5.3)$$

With boundary conditions

$$\eta = 0: F = 0, \qquad \frac{dF}{d\eta} = 0$$

$$\eta = \infty: \frac{dF}{d\eta} = 1$$

Where the dimensionless constant M is defined by

$$M = \frac{\sigma B^2}{a\rho}$$

1.5.2. Flow in a Channel

The second application is the problem of fluid injection through one side of a long vertical channel [7]. The Navier-stokes and the heat transfer equations can be reduced to the following system

$$f''' - R[(f')^2 - ff''] + RA = 0 \quad (1.5.4)$$

$$\text{With } f(0) = f'(0) = 0, f(1) = 1, f'(1) = 0 \quad (1.5.5)$$

$$h'' + Rfh' + 1 = 0 \quad (1.5.6)$$

$$\text{With } h(0) = h(1) = 0 \quad (1.5.7)$$

$$\theta'' + Pf\theta' = 0 \quad (1.5.8)$$

$$\text{With } \theta(0) = 0, \theta(1) = 1 \quad (1.5.9)$$

Where

f, h : Two potential functions.

θ : Temperature distribution function.

A : Undetermined constant.

R : Reynolds number.

P : Peclet number.

The original problem is effectively broken into three subproblems. Thus the equation (1.5.4) with given boundary conditions (1.5.5) represented a nonlinear third-order ordinary differential equation for f, where equation (1.5.5) represented four boundary conditions. To get the standard form for equation (1.5.4), there is one way that is to differentiate it, obtaining that

$$f'''' = R[f'f'' - ff'''] \qquad (1.5.10)$$

So the problem (1.5.10) and (1.5.5) is in standard form and no longer explicitly involves A. Then the equations (1.5.6), (1.5.7) and (1.5.8) and (1.5.9) are two separated, linear second order problems in standard form. The difficulty in solving the nonlinear problem (1.5.10) numerically depends on Reynolds number R. For moderate values of R, say $R = 10$, the problem is easy, but it get tougher as R increases and for $R = 10{,}000$ there is a fast change in some solution values near $x = 0$. This is called a boundary layer.

Chapter Two
Shooting Method

Chapter Two

Shooting Method

2.1. Shooting Method for Second Order Linear Two-Point Boundary Value Problems

In numerical analysis, the shooting method is an iterative method that has been used for solving boundary value problem by reducing it to an initial value problem. The general idea of the shooting method based on converting the boundary value problem into a system of initial value problems with specified initial value conditions. The unknown initial conditions are guessed to solve the initial value problems. The accuracy of the guessed missing initial condition is then checked by comparing the calculated value of the dependent variable at the terminal point with its given value there. If a difference exists, another value of the missing initial condition must be guessed and the process is repeated. This process is continued until the agreement between the calculated and the given condition at the terminal point is within the specified degree of accuracy.

The following second order linear two-point boundary value problem

$$y'' = p(x)y' + q(x)y + r(x)\ , a \le x \le b, y(a) = \alpha, y(b) = \beta \quad (2.1.1)$$

According to Burden [12], the equation (2.1.1) will be converted into two second order linear initial value problems

$$u'' = p(x)u' + q(x)u + r(x)\ , a \le x \le b, u(a) = \alpha, u'(a) = 0 \quad (2.1.2)$$

And

$$v'' = p(x)v' + q(x)v, a \le x \le b, v(a) = 0, v'(a) = 1 \qquad (2.1.3)$$

The equation (2.1.2) can be written in vector notation as we let $u_1 = u$ and $u_2 = u'$, so the first initial value problem becomes as

$$u_2' = p(x)u_2 + q(x)u_1 + r(x), a \le x \le b$$

$$u_1(a) = \alpha, u_2(a) = 0$$

And this can be written in vector-matrix notation as

$$\overrightarrow{u'} = \begin{bmatrix} 0 & 1 \\ q(x) & p(x) \end{bmatrix} \vec{u} + \begin{bmatrix} 0 \\ r(x) \end{bmatrix}, \vec{u}(a) = \begin{bmatrix} \alpha \\ 0 \end{bmatrix} \quad (2.1.4)$$

Where $\vec{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ and $\overrightarrow{u'} = \begin{bmatrix} u_1' \\ u_2' \end{bmatrix}$

In the same way, the equation (2.1.3) can be written in vector notation, let $v_1 = v$ and $v_2 = v'$, so the second initial value problem becomes as

$$v_2' = p(x)v_2 + q(x)v_1, a \le x \le b$$

$$v_1(a) = 0, v_2(a) = 1$$

And this can be written in vector-matrix notation as

$$\overrightarrow{v'} = \begin{bmatrix} 0 & 1 \\ q(x) & p(x) \end{bmatrix} \vec{v}, \vec{v}(a) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (2.1.5)$$

Where $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ and $\overrightarrow{v'} = \begin{bmatrix} v_1' \\ v_2' \end{bmatrix}$

Let $u(x)$ is the solution for equation (2.1.2) and $v(x)$ is the solution for equation (2.1.3) then we define $z(x)$

$$\vec{z} = \vec{u} + \theta\vec{v}$$

where θ is a constant number. So

$$\vec{z}(x) = \begin{bmatrix} u_1(x) + \theta v_1(x) \\ u_2(x) + \theta v_2(x) \end{bmatrix} \quad (2.1.6)$$

Using equation (2.1.2) and (2.1.3), it is easy to see that

$$(u + \theta v)'' = p(x)(u + \theta v)' + q(x)(u + \theta v) + r(x)$$

$$(u + \theta v)(a) = \alpha, (u + \theta v)'(a) = \theta$$

$z(x)$ satisfies

$$z'' = p(x)z' + q(x)z + r(x), z(a) = \alpha, z'(a) = \theta \qquad (2.1.7)$$

The only difference between (2.1.1) and (2.1.7) is that in (2.1.7) we know the value of z' at $x = a$, but do not know whether $z(b) = \beta$. If we can choose θ in such a way that $z(b)$ equals β, this will mean that we have solved the boundary value problem (2.1.1).

To determine the value of θ, we first solve the two initial value problems (2.1.2) and (2.1.3) and find the values of $u(b)$ and $v(b)$, then we choose the value θ_0 by requiring that $z(b) = \beta$, so

$$z(b) = u(b) + \theta_0 v(b) = \beta$$

And

$$\theta_0 = \frac{\beta - u(b)}{v(b)}$$

Burden [12] assumed that

$$z(x) = u(x) + \frac{\beta - u(b)}{v(b)} v(x) \qquad (2.1.8)$$

Let equation (2.1.8) is a solution for second order linear boundary value problem shown in (2.1.1).

To check that, we assume the first and second derivative for equation (2.1.8)

$$z'(x) = u'(x) + \frac{\beta - u(b)}{v(b)} v'(x) \qquad (2.1.9)$$

$$z''(x) = u''(x) + \frac{\beta - u(b)}{v(b)} v''(x) \qquad (2.1.10)$$

Substitute equations (2.1.2) and (2.1.3) in (2.1.10)

$$z''(x) = p(x)u' + q(x)u + r(x) + \frac{\beta - u(b)}{v(b)} [p(x)v' + q(x)v]$$

$$z''(x) = p(x)\left[u' + \frac{\beta - u(b)}{v(b)} v'\right] + q(x)\left[u + \frac{\beta - u(b)}{v(b)} v\right] + r(x)$$

$$z''(x) = p(x)z' + q(x)z + r(x)$$

Moreover,

$$z(a) = u(a) + \frac{\beta - u(b)}{v(b)} v(a), z(a) = \alpha + \frac{\beta - u(b)}{v(b)} . 0 = \alpha$$

The same

$$z(b) = u(b) + \frac{\beta - u(b)}{v(b)} v(b), z(b) = u(b) + \beta - u(b) = \beta$$

So equation (2.1.9) is a solution for the second order linear boundary value problem (2.1.1) with its given boundary conditions (2.1.2). The nonlinear shooting method is the same as shooting method except it's difficult to express the solution of nonlinear boundary value problem as a combination of solutions of initial value problems, so we depends on the solutions of a sequence of initial value problems involving a parameter t. The following equation expresses the general form for nonlinear second order boundary value problem [12]

$$y'' = f(x, y, y'), a \leq x \leq b$$

With the following boundary conditions

$$y(a) = \alpha, y(b) = \beta$$

The purpose of this method is to find $t = t_k$ to ensure that

$$\lim_{k \to \infty} y(b, t_k) = y(b) = \beta$$

The general form of the solution of sequence of initial value problems is

$$y'' = f(x, y, y'), a \leq x \leq b, with\ y(a) = \alpha, y'(a) = t \qquad (2.1.11)$$

If we rewrite the equation (2.1.11) that the solution depends on x and t

$$y''(x, t) = f\big(x, y(x, t), y'(x, t)\big), a \leq x \leq b, with\ y(a, t) = \alpha, y'(a, t) = t \qquad (2.1.12)$$

We take the first derivative with respect to t of equation (2.1.12) and rewrite it by substituting $\frac{\partial y}{\partial t}(x,t) = z(x,t)$, we get that

$$z''(x,t) = \frac{\partial f}{\partial y}(x,y,y')z(x,t) + \frac{\partial f}{\partial y'}(x,y,y')z'(x,t), a \le x \le b$$

$$with\ z(a,t) = 0, z'(a,t) = 1 \qquad (2.1.13)$$

We use the Newton's method to generate the sequence t_k , for both initial value problems (2.1.11) and (2.1.13) as

$$t_k = t_{k-1} - \frac{y(b,t_{k-1}) - \beta}{z(b,t_{k-1})}$$

None of these initial value problems is solved exactly, but the solutions are approximated using Runge-Kutta method of order four to approximate both solutions required by Newton's method.

2.2. Shooting Method for Third Order Linear Two-Point Boundary Value Problems

Let

$$y''' = p(x)y'' + q(x)y' + r(x)y + s(x)\ , a \le x \le b \qquad (2.2.1)$$

$$y(a) = \alpha\ , y'(a) = \beta\ , y(b) = \gamma \qquad (2.2.2)$$

An equation (2.2.1) and (2.2.2) is a third order linear two-point boundary value problem with the specific boundary conditions. In shooting method the third order boundary value problem will turn into two initial value problems, where we replace the boundary conditions with specific initial conditions for each initial value problem:

$$u''' = p(x)u'' + q(x)u' + r(x)u + s(x)\ , a \le x \le b \qquad (2.2.3)$$

$$u(a) = \alpha, u'(a) = 0, u''(a) = 0 \qquad (2.2.4)$$

$$v''' = p(x)v'' + q(x)v' + r(x)v \ , a \le x \le b \qquad (2.2.5)$$

$$v(a) = 0, v'(a) = 1, v''(a) = 0 \qquad (2.2.6)$$

Equation (2.2.3) and boundary conditions (2.2.4) can be written in vector-matrix notation as we let

$$u_1 = u \ , \ u_2 = u' \ , u_3 = u'', \ u_3' = u'''$$

So the first initial value problem becomes as:

$$u_3' = p(x)u_3 + q(x)u_2 + r(x)u_1 + s(x) \ , a \le x \le b \qquad (2.2.7)$$

$$u_1(a) = \alpha, u_2(a) = 0, u_3(a) = 0 \qquad (2.2.8)$$

And this can be written as:

$$\overrightarrow{u'} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ r(x) & q(x) & p(x) \end{bmatrix} \vec{u} + \begin{bmatrix} 0 \\ 0 \\ s(x) \end{bmatrix} , \vec{u}(a) = \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix}$$

And also equation (2.2.5) and boundary conditions (2.2.6) can be written in vector-matrix notation as

$$v_1 = v \ , v_2 = v' \ , v_3 = v'' \ , v_3' = v'''$$

So the second initial value problem becomes as:

$$v_3' = p(x)v_3 + q(x)v_2 + r(x)v_1 \ , a \le x \le b \qquad (2.2.9)$$

$$v_1(a) = 0, v_2(a) = 1, v_3(a) = 0 \qquad (2.2.10)$$

And this can be written as:

$$\overrightarrow{v'} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ r(x) & q(x) & p(x) \end{bmatrix} \vec{v} \ , \vec{v}(a) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

We will use the Runge-Kutta method of order 4 (RK4) to find the solution of the two initial value problems $\vec{u}(x)$ and $\vec{v}(x)$.Then we construct $\vec{z}$ where:

$$\vec{z} = \vec{u} + \theta\vec{v}$$

$\vec{z}$ can be written in vector notation as:

$$\vec{z}(x) = \begin{bmatrix} u_1(x) + \theta v_1(x) \\ u_2(x) + \theta v_2(x) \\ u_3(x) + \theta v_3(x) \end{bmatrix} \tag{2.2.11}$$

where θ is a constant number.

Let $u(x)$ and $v(x)$ denote the solutions to the third order linear initial value problem (2.2.3) and (2.2.5) respectively. Define that

$$z(x) = u(x) + \frac{\gamma - u(b)}{v(b)} v(x), v(b) \neq 0 \tag{2.2.12}$$

Then $z(x)$ is the solution to the third order linear boundary value problem (2.2.1). To see this, first note that

$$z'(x) = u'(x) + \frac{\gamma - u(b)}{v(b)} v'(x)$$

$$z''(x) = u''(x) + \frac{\gamma - u(b)}{v(b)} v''(x)$$

And

$$z'''(x) = u'''(x) + \frac{\gamma - u(b)}{v(b)} v'''(x) \tag{2.2.13}$$

Substituting equation (2.2.3) and (2.2.5) in equation (2.2.13), we get that

$$z'''(x) = [p(x)u'' + q(x)u' + r(x)u + s(x)] + \frac{\gamma - u(b)}{v(b)} [p(x)v''$$

$$+ q(x)v' + r(x)v]$$

$$z'''(x) = p(x)\left[u'' + \frac{\gamma - u(b)}{v(b)} v''\right] + q(x)\left[u' + \frac{\gamma - u(b)}{v(b)} v'\right]$$

$$+ r(x)\left[u + \frac{\gamma - u(b)}{v(b)} v\right] + s(x)$$

$$z'''(x) = p(x)z'' + q(x)z' + r(x)z + r(x)$$

Moreover,

$$z(a) = u(a) + \frac{\gamma - u(b)}{v(b)} v(a), z(a) = \alpha + \frac{\gamma - u(b)}{v(b)} . 0 = \alpha$$

The same

$$z(b) = u(b) + \frac{\gamma - u(b)}{v(b)} v(b), z(b) = u(b) + \gamma - u(b) = \gamma$$

So equation (2.1.12) is a solution for the third order linear boundary value problem (2.2.1) and boundary conditions (2.2.2).

Chapter Three
Finite Difference Method

Chapter Three

Finite Difference Method

3.1. Finite Difference Method for Second Order Linear Two-Point Boundary Value Problems

The finite difference approximation for derivatives are one of the simplest and of the oldest methods to solve differential equations where the main idea of finite difference method is replacing the derivatives to appropriate finite differences and the differential equation is reduced to a system of algebraic equations. Taylor series are used to approximate derivatives as the following [13]:

For some η_i in (x_{i-1}, x_{i+1})

$$y'(x_i) \cong \frac{y(x_{i+1}) - y(x_{i-1})}{2h} - \frac{h^2}{6} y'''(\eta_i) \qquad (3.1.1)$$

For some ξ_i in (x_{i-1}, x_{i+1})

$$y''(x_i) \cong \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} - \frac{h^2}{12} y^{(4)}(\xi_i) \qquad (3.1.2)$$

And for some μ_i in (x_{i-2}, x_{i+2})

$$y'''(x_i) \cong \frac{y(x_{i+2}) - 2y(x_{i+1}) + 2y(x_{i-1}) - y(x_{i-2})}{2h^3} - \frac{h^2}{4} y^{(5)}(\mu_i) \qquad (3.1.3)$$

For the following second order linear two-point boundary value problem with given boundary conditions, let

$$y'' = p(x)y' + q(x)y + r(x), a \le x \le b \qquad (3.1.4)$$

$$y(a) = \alpha, y'(a) = \beta \qquad (3.1.5)$$

We substitute equations (3.1.1) and (3.1.2) in (3.1.4), so we get that

$$\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} = p(x_i)\frac{y(x_{i+1}) - y(x_{i-1})}{2h} + q(x_i)y(x_i) + r(x_i) - \frac{h^2}{12}\left[2p(x_i)y'''(\eta_i) - y^{(4)}(\xi_i)\right]$$

A Finite Difference Method with truncation error of order $O(h^2)$ results by using this equation together with the boundary conditions to define the system of linear equations [12]

$$w_0 = \alpha, w_{N+1} = \beta$$

And

$$\frac{-w_{i+1} + 2w_i - w_{i-1}}{h^2} + p\left(\frac{w_{i+1} - w_{i-1}}{2h}\right) + q(x_i)w_i = -r(x_i) \quad (3.1.6)$$

Equation (3.1.6) will be rewritten as

$$-\left(1 + \frac{h}{2}p(x_i)\right)w_{i-1} + \left(2 + h^2q(x_i)\right)w_i - \left(1 - \frac{h}{2}p(x_i)\right)w_{i+1} = -h^2r(x_i)$$

For $i = 1, 2, \dots, N$, the resulting system is expressed in $N \times N$ matrix form

$$Aw = b \quad (3.1.7)$$

Where

$$A = \begin{bmatrix} 2 + h^2q(x_1) & -1 + \frac{h}{2}p(x_1) & 0 & \cdots & & \dots & 0 \\ -1 - \frac{h}{2}p(x_2) & 2 + h^2q(x_2) & -1 + \frac{h}{2}p(x_2) & 0 & & & \vdots \\ 0 & \ddots & & \ddots & & & 0 \\ \vdots & & \cdots & 0 & -1 - \frac{h}{2}p(x_{N-1}) & 2 + h^2q(x_{N-1}) & -1 + \frac{h}{2}p(x_{N-1}) \\ 0 & \cdots & \dots & & 0 & -1 - \frac{h}{2}p(x_N) & 2 + h^2q(x_N) \end{bmatrix}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix} \text{ and } b = \begin{bmatrix} -h^2 r(x_1) + \left(1 + \frac{h}{2} p(x_1)\right) w_0 \\ -h^2 r(x_2) \\ \vdots \\ -h^2 r(x_{N-1}) \\ -h^2 r(x_N) + \left(1 - \frac{h}{2} p(x_N)\right) w_{N+1} \end{bmatrix}$$

The linear system (3.1.7) will be solved using any iterative method to find the approximated solution w_i.

3.2. Finite Difference Method for Third Order Linear Two-Point Boundary Value Problems

Consider the following third order linear two-point boundary value problem with the given boundary conditions:

$$y''' = p(x)y'' + q(x)y' + r(x)y + s(x), a \le x \le b \qquad (3.2.1)$$

$$y(a) = \alpha, y'(a) = \beta, y(b) = \gamma \qquad (3.2.2)$$

Now we will rewrite the equation (3.2.1) in finite difference form with the grid point. Let N be the number of subintervals, so $x_{i+1} = x_i + h$, where $i = 0,1, \dots, N$ and $x_0 = a$

$$\frac{y_{i+2} - 2y_{i+1} + 2y_{i-1} - y_{i-2}}{2h^3} - \frac{h^2}{4} y^{(5)}(\mu_i)$$
$$= p(x_i)\left[\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - \frac{h^2}{12} y^{(4)}(\xi_i)\right]$$
$$+ q(x_i)\left[\frac{y_{i+1} - y_{i-1}}{2h} - \frac{h^2}{6} y'''(\eta_i)\right] + r(x_i)y_i + s(x_i)$$

A Finite Difference Method with truncation error of order $O(h^2)$ results by using this equation together with the boundary conditions to define the system of linear equations

$$w(a) = \alpha, w'(a) = \beta, w(b) = \gamma$$

And by multiplying the equation with $2h^3$, we get that

$$w_{i+2} - 2w_{i+1} + 2w_{i-1} - w_{i-2}$$
$$= 2h[w_{i+1} - 2w_i + w_{i-1}]p(x_i) + h^2[w_{i+1} - w_{i-1}]q(x_i)$$
$$+ 2h^3 r(x_i)w_i + 2h^3 s(x_i)$$
$$w_{i+2} - 2w_{i+1} + 2w_{i-1} - w_{i-2} - 2h[w_{i+1} - 2w_i + w_{i-1}]p(x_i)$$
$$- h^2[w_{i+1} - w_{i-1}]q(x_i) - 2h^3 r(x_i)w_i = 2h^3 s(x_i)$$

So for $i = 0,1, \dots, N$

$$w_{i+2} - [2 + 2hp(x_i) + h^2 q(x_i)]w_{i+1} + [4hp(x_i) - 2h^3 r(x_i)]w_i +$$
$$[2 - 2hp(x_i) + h^2 q(x_i)]w_{i-1} - w_{i-2} = 2h^3 s(x_i) \qquad (3.2.3)$$

And the boundary conditions become as:

$w(a) = \alpha$ changed to $w_0 = \alpha$

$w'(a) = \beta$ to $\frac{w_1 - w_{-1}}{2h} = \beta$ since $i = 0$

$w(b) = \gamma$ to $w_N = \gamma$

For $i = 0,1, \dots, N$, we will get $N + 1$ algebraic equations for the solution of $N + 1$ unknown variables.

If we substitute $i = 0$ into equation (3.2.3), we get that

$$w_2 - [2 + 2hp(x_0) + h^2 q(x_0)]w_1 + [4hp(x_0) - 2h^3 r(x_0)]w_0$$
$$+ [2 - 2hp(x_0) + h^2 q(x_0)]w_{-1} - w_{-2} = 2h^3 s(x_0)$$

We note that there are unwanted terms as w_{-1} and w_{-2} , and the same problem happens when $i = N$

$$w_{N+2} - [2 + 2hp(x_N) + h^2 q(x_N)]w_{N+1} + [4hp(x_N) - 2h^3 r(x_N)]w_N$$
$$+ [2 - 2hp(x_N) + h^2 q(x_N)]w_{N-1} - w_{N-2} = 2h^3 s(x_N)$$
$$for\ i = 0,1, \dots, N$$

There are unwanted terms w_{N+1} and w_{N+2} .So we have $N + 1$ equations with $N + 1$ unknown variables with addition to w_{-1}, w_{-2}, w_{N+1} and w_{N+2} .

To get rid of these terms, we will use the following assumption:

Let

$$\frac{dw}{dx} = z \tag{3.2.4}$$

We substitute (3.2.4) in (3.2.1) and (3.2.2), so we get

$$z'' = p(x)z' + q(x)z + r(x)w + s(x) \tag{3.2.5}$$

$$w(a) = \alpha, z(a) = \beta, w(b) = \gamma \tag{3.2.6}$$

By replacing (3.2.5) using finite difference form, we get that:

$$\frac{z_{i+1} - 2z_i + z_{i-1}}{h^2} = p(x_i)\left[\frac{z_{i+1} - z_{i-1}}{2h}\right] + q(x_i)z_i + r(x_i)w_i + s(x_i)$$

$$z_{i+1} - 2z_i + z_{i-1} = \frac{h}{2}[z_{i+1} - z_{i-1}]p(x_i) + h^2q(x_i)z_i + h^2r(x_i)w_i + h^2s(x_i)$$

$$[1 - \frac{h}{2}p(x_i)]z_{i+1} - [2 + h^2q(x_i)]z_i + \left[1 + \frac{h}{2}p(x_i)\right]z_{i-1} - h^2r(x_i)w_i = h^2s(x_i)$$

Let

$$a_i = 1 - \frac{h}{2}p(x_i)$$

$$b_i = -2 - h^2q(x_i)$$

$$c_i = 1 + \frac{h}{2}p(x_i)$$

$$d_i = -h^2r(x_i)$$

$$e_i = h^2s(x_i)$$

So the equation (3.2.5) becomes as

$$a_iz_{i+1} + b_iz_i + c_iz_{i-1} + d_iw_i = e_i, for\ i = 0, 1, \dots, N \tag{3.2.7}$$

Equation (3.2.4) can also be written using finite difference approximation around the point $x_{i-\frac{1}{2}}$:

$$\frac{w_i - w_{i-1}}{h} = \frac{z_i + z_{i-1}}{2}$$

$$w_i - w_{i-1} - \frac{h}{2}z_i - \frac{h}{2}z_{i-1} = 0 \qquad (3.2.8)$$

And the boundary conditions can be written as:

$$w_0 = \alpha, z_0 = \beta, w_N = \gamma \qquad (3.2.9)$$

If we substitute $i = 0$ in equations (3.2.7) and (3.2.8), we get that

$$w_0 - w_{-1} - \frac{h}{2}z_0 - \frac{h}{2}z_{-1} = 0$$

$$\alpha - w_{-1} - \frac{h}{2}\beta - \frac{h}{2}z_{-1} = 0$$

$$w_{-1} + \frac{h}{2}z_{-1} = \alpha - \frac{h}{2}\beta$$

And

$$a_0 z_1 + b_0 z_0 + c_0 z_{-1} + d_0 w_0 = e_0$$

$$a_0 z_1 + b_0 \beta + c_0 z_{-1} + d_0 \alpha = e_0$$

$$a_0 z_1 + c_0 z_{-1} = e_0 - b_0 \beta - d_0 \alpha$$

For $i = 1$

$$w_1 - w_0 - \frac{h}{2}z_1 - \frac{h}{2}z_0 = 0$$

$$w_1 - \alpha - \frac{h}{2}z_1 - \frac{h}{2}\beta = 0$$

$$w_1 - \frac{h}{2}z_1 = \alpha + \frac{h}{2}\beta$$

And

$$a_1 z_2 + b_1 z_1 + c_1 z_\circ + d_1 w_1 = e_1$$

$$a_1 z_2 + b_1 z_1 + c_1 \beta + d_1 w_1 = e_1$$

$$a_1 z_2 + b_1 z_1 + d_1 w_1 = e_1 - c_1 \beta$$

For $2 \leq i \leq N - 1$, equations (3.2.7) and (3.2.8) can be used without any change. For $i = N$, equation (3.2.8) becomes as

$$w_N - w_{N-1} - \frac{h}{2}z_N - \frac{h}{2}z_{N-1} = 0$$

$$\gamma - w_{N-1} - \frac{h}{2}z_N - \frac{h}{2}z_{N-1} = 0$$
$$w_{N-1} + \frac{h}{2}z_N + \frac{h}{2}z_{N-1} = \gamma$$

And equation (3.2.7) becomes as

$$a_N z_{N+1} + b_N z_N + c_N z_{N-1} + d_N w_N = e_N$$
$$a_N z_{N+1} + b_N z_N + c_N z_{N-1} + d_N \gamma = e_N$$
$$a_N z_{N+1} + b_N z_N + c_N z_{N-1} = e_N - d_N \gamma$$

All boundary conditions have been taking into consideration. The above steps eliminate w_0, w_N and z_0 as variables. Here, we have $2N + 2$ equations for the solution of $2N + 2$ unknown variables: w_i where $i = -1, 1, \dots, N-1$ and z_i where $i = -1, 1, \dots, N, N+1$

In vector-matrix form, this can be written as:

$$Am = k \qquad (3.2.10)$$

Where

$$m = \begin{bmatrix} \begin{bmatrix} w_{-1} \\ z_{-1} \end{bmatrix} \\ \begin{bmatrix} w_1 \\ z_1 \end{bmatrix} \\ \begin{bmatrix} w_2 \\ z_2 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} w_{N-1} \\ z_{N-1} \end{bmatrix} \\ \begin{bmatrix} z_N \\ z_{N+1} \end{bmatrix} \end{bmatrix} \; and \; k = \begin{bmatrix} \begin{bmatrix} r_\circ \\ s_\circ \end{bmatrix} \\ \begin{bmatrix} r_1 \\ s_1 \end{bmatrix} \\ \begin{bmatrix} r_2 \\ s_2 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} r_{N-1} \\ s_{N-1} \end{bmatrix} \\ \begin{bmatrix} r_N \\ s_N \end{bmatrix} \end{bmatrix}$$

As $r_0 = \alpha - \frac{h}{2}\beta, r_1 = \alpha + \frac{h}{2}\beta, r_N = \gamma$, but $r_i = 0$ for all $i = 2,3, \dots, N-1$

And $s_0 = e_0 - b_0\beta - d_0\alpha, s_1 = e_1 - c_1\beta,\ s_N = e_N - d_N\gamma$, but $s_i = e_i$ for all $i = 2,3, \dots, N-1$

$$A = \begin{bmatrix} \begin{bmatrix} 1 & \frac{h}{2} \\ 0 & c_0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & a_0 \end{bmatrix} & \cdots & & & \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 1 & \frac{-h}{2} \\ d_1 & b_1 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & a_1 \end{bmatrix} & & & \\ & \vdots & & \ddots & & \\ & & & & \begin{bmatrix} -1 & \frac{-h}{2} \\ 0 & c_{N-1} \end{bmatrix} & \begin{bmatrix} 1 & \frac{-h}{2} \\ d_{N-1} & b_{N-1} \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ a_{N-1} & 0 \end{bmatrix} \\ & \vdots & & \cdots & & \begin{bmatrix} 1 & \frac{h}{2} \\ 0 & c_N \end{bmatrix} & \begin{bmatrix} \frac{h}{2} & 0 \\ b_N & a_N \end{bmatrix} \end{bmatrix}$$

The linear system (3.2.10) will be solved using LU Decomposition or any iterative method like Jacobi or Gauss Seidel to find the approximated solution.

Chapter Four

Pade Approximation Method

Chapter Four

Pade Approximation Method

4.1. Introduction

There are a sufficient number of polynomials to approximate any continuous function on a closed interval to within an arbitrary tolerance, where the advantage is that derivatives and integrals are easily determined and polynomials are easily evaluated at arbitrary values. But polynomial approximations have a disadvantage in tendency for oscillation. This often causes error bounds in polynomial approximation to significantly exceed the average approximation error, because error bounds are determined by the maximum approximation error [12]. We now consider methods that spread the approximation error more evenly over the approximation interval. These techniques involve rational functions. Let r(x) is a rational functions of degree N, and

$$r(x) \cong \frac{p(x)}{q(x)} = \frac{p_0 + p_1 x + \cdots + p_n x^n}{q_0 + q_1 x + \cdots + q_m x^m}$$

Where $p(x)$ and $q(x)$ are polynomials whose degrees sum to N, and $r(x)$ is the approximation function for $f(x)$ on a closed interval I. For the interval I containing zero it requires to have $q_0 \neq 0$ in order to make $r(x)$ is defined at zero. We can assume that $q_0 = 1$, for if this is not the case we simply replace $p(x)$ by $p(x)/q_0$ and $q(x)$ by $q(x)/q_0$. Every polynomial is considered as a rational function if we set $q(x) = 1$.

Pade Approximation technique is the extension of Taylor polynomial approximation to rational functions [12]. If we consider that

$$f(x) - r(x) = f(x) - \frac{p(x)}{q(x)}$$

$$f(x) - r(x) = \frac{f(x)q(x) - p(x)}{q(x)}$$

Let the Maclaurin series for $f(x) = \sum_{=0}^{\infty} a_i x^i$, then

$$f(x) - r(x) = \frac{\sum_{i=0}^{\infty} a_i x^i \sum_{i=0}^{m} q_i x^i - \sum_{i=0}^{n} p_i x^i}{\sum_{i=0}^{m} q_i x^i}$$

We need to choose the values of $q_1, q_2, \dots, q_m$ and $p_0, p_1, \dots, p_n$ to obtain that $f^{(k)}(0) - r^{(k)}(0) = 0$ for $k = 0, 1, \dots, N$.

$$(a_0 + a_1 x + a_2 x^2 + \cdots)(1 + q_1 x + \cdots + q_m x^m) - (p_0 + p_1 x + \cdots + p_n x^n)$$

For example the Pade approximation to the function $f(x) = e^x$ of degree 4 where $n = 2$ and $m = 2$ can be calculated as the Maclaurin series for $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$, so we have that

$$\left(1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \cdots\right)(1 + q_1 x + q_2 x^2) - (p_0 + p_1 x + p_2 x^2)$$

By expanding and collecting terms so that the coefficients of x^k for $k = 0, 1, \cdots, 4$ are zeros, then we have

$$x^0 : 1 = p_0$$

$$x^1 : 1 + q_1 = p_1$$

$$x^2 : \frac{1}{2} + q_1 + q_2 = p_2$$

$$x^3 : \frac{1}{6} + \frac{1}{2} q_1 + q_2 = 0$$

$$x^4 : \frac{1}{24} + \frac{1}{6} q_1 + \frac{1}{2} q_2 = 0$$

We can solve the previous system using Maple and get that

$$p_1 = \frac{1}{2}, p_2 = \frac{1}{12}, q_1 = \frac{-1}{2}, q_2 = \frac{1}{12}$$

So the Pade approximation for e^x of degree 4 with $n = 2$ and $m = 2$ is

$$e^x = \frac{1 + \frac{1}{2}x + \frac{1}{12}x^2}{1 - \frac{1}{2}x + \frac{1}{12}x^2}$$

Pade approximant is a rational approximation where Pade approximant to the function $f(z)$ on $[a, b]$ is the quotient of two polynomials $P_k(z)$ and $Q_\mu(z)$ of degree k and μ, respectively. So we denote this quotient

$$R_{k,\mu}(z) = \frac{P_k(z)}{Q_\mu(z)} + O(z^{k+\mu}) \qquad (4.1.1)$$

If $k = \mu$, $R_{k,\mu}(z)$ will be called as a diagonal Pade approximant [19], [30]. Pade approximant $R_{k,\mu}(z)$ will be calculated for $f(z) = e^z$ as the following [11], [24], [28]:

$$P_k(z) = \sum_{j=0}^{k} \frac{(\mu + k - j)!\, k!}{(\mu + k)!\, j!\, (k - j)!} z^j \qquad (4.1.2)$$

And

$$Q_\mu(z) = \sum_{j=0}^{\mu} \frac{(\mu + k - j)!\, \mu!}{(\mu + k)!\, j!\, (\mu - j)!} (-z)^j \qquad (4.1.3)$$

Then the Pade Approximation (k, μ) for

$$e^z \cong \frac{P_k(z)}{Q_\mu(z)} \qquad (4.1.4)$$

4.2. Pade Approximation (2,2) For Solving Third Order Two-Point Boundary Value Problems

For Pade Approximation (2,2) where $k = 2$ and $\mu = 2$, using equation (4.1.2) and (4.1.3) we get that

$$P_2(z) = 1 + \frac{z}{2} + \frac{z^2}{12}$$

$$Q_2(z) = 1 - \frac{z}{2} + \frac{z^2}{12}$$

So

$$e^z = \frac{1 + \frac{z}{2} + \frac{z^2}{12}}{1 - \frac{z}{2} + \frac{z^2}{12}}$$

Using Taylor series, we get that

$$u(x+h) \cong u(x) + hDu(x) + \frac{h^2}{2!}D^2u(x)$$

So

$$u(x+h) = \left[1 + hD + \frac{h^2}{2!}D^2\right]u(x)$$

And

$$u(x+h) = e^{hD}u(x)$$

where D is the differential operator and h is the step size distance. Now using the relation $u(x+h) = e^{hD}u(x)$, e^{hD} can be replaced by Pade Approximation with operator D, we get that

$$u(x+h) = \frac{1 + \frac{hD}{2} + \frac{(hD)^2}{12}}{1 - \frac{hD}{2} + \frac{(hD)^2}{12}}u(x)$$

When u is a vector, this becomes

$$\left[I - \frac{hD}{2} + \frac{(hD)^2}{12}\right]u(x+h) = \left[I + \frac{hD}{2} + \frac{(hD)^2}{12}\right]u(x) \qquad (4.2.1)$$

Now, consider the general form for a third order two point boundary value problems with the following given conditions:

$$y''' + p(x)y'' + q(x)y' + r(x)y = s(x)\,, a \le x \le b \qquad (4.2.2)$$

$$y(a) = \alpha\,, y'(a) = \beta\,, y(b) = \gamma \qquad (4.2.3)$$

System of equations (4.2.2) and (4.2.3) can be converted into first order vector-matrix system as following:

Let $u_0 = y,\ u_1 = y',\ u_2 = y'',\ u_2' = y'''$ and

$$u_2' = -p(x)u_2 - q(x)u_1 - r(x)u_0 + s(x)$$

So $\vec{u} = \begin{bmatrix} u_2 \\ u_1 \\ u_0 \end{bmatrix}$ and $\vec{u}' = \begin{bmatrix} u_2' \\ u_1' \\ u_0' \end{bmatrix}$

$$\vec{u}' = \begin{bmatrix} -p(x) & -q(x) & -r(x) \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_2 \\ u_1 \\ u_0 \end{bmatrix} + \begin{bmatrix} s(x) \\ 0 \\ 0 \end{bmatrix}$$

$$D\vec{u} = Q\vec{u} + \vec{P} \tag{4.2.4}$$

Boundary conditions in equation (4.2.3) can also be converted as:

$$u_0(a) = \alpha, u_1(a) = \beta, u_0(b) = \gamma$$

Using equation (4.2.4) and its second derivative, where

$$D^2\vec{u} = Q'\vec{u} + Q^2\vec{u} + Q\vec{P} + \vec{P}' \tag{4.2.5}$$

And substituting them in equation (4.2.1) and applying them on the discrete point x_i, where $x_i = a + ih,\ i = 0,1, \dots, N$ and $h = (b-a)/N$ with N subintervals.

So we get that

$$A_{i+1}u_{i+1} + B_i u_i = E_{i+1} + F_i, for\ i = 0, 1, \dots N-1 \tag{4.2.6}$$

Where

$$A_{i+1} = \begin{bmatrix} a_{i+1,1,1} & a_{i+1,1,2} & a_{i+1,1,3} \\ a_{i+1,2,1} & a_{i+1,2,2} & a_{i+1,2,3} \\ a_{i+1,3,1} & a_{i+1,3,2} & a_{i+1,3,3} \end{bmatrix}$$

$$B_i = \begin{bmatrix} b_{i,1,1} & b_{i,2,1} & b_{i,1,3} \\ b_{i,2,1} & b_{i,2,2} & b_{i,2,3} \\ b_{i,3,1} & b_{i,3,2} & b_{i,3,3} \end{bmatrix}$$

And

$$E_{i+1} = \begin{bmatrix} e_{i+1.1} \\ e_{i+1,2} \\ e_{i+1,3} \end{bmatrix} \ and\ F_k = \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ f_{i,3} \end{bmatrix}$$

Such that

$$a_{i+1,1,1} = 1 + \frac{h}{2}p_{i+1} + \frac{h^2}{12}(-p'_{i+1} + p^2_{i+1} - q_{i+1})$$

$$a_{i+1,1,2} = \frac{h}{2}q_{i+1} + \frac{h^2}{12}(-q'_{i+1} + p_{i+1}q_{i+1} - r_{i+1})$$

$$a_{i+1,1,3} = \frac{h}{2}r_{i+1} + \frac{h^2}{12}(-r'_{i+1} + p_{i+1}r_{i+1})$$

$$a_{i+1,2,1} = -\frac{h}{2} - \frac{h^2}{12}p_{i+1}$$

$$a_{i+1,2,2} = 1 - \frac{h^2}{12}q_{i+1}$$

$$a_{i+1,2,3} = -\frac{h^2}{12}r_{i+1}$$

$$a_{i+1,3,1} = \frac{h^2}{12}$$

$$a_{i+1,3,2} = -\frac{h}{2}$$

$$a_{i+1,3,3} = 1$$

And

$$b_{i,1,1} = -1 + \frac{h}{2}p_i - \frac{h^2}{12}(-p'_i + p^2_i - q_i)$$

$$b_{i,1,2} = \frac{h}{2}q_i - \frac{h^2}{12}(-q'_i + p_i q_i - r_i)$$

$$b_{i,1,3} = \frac{h}{2}r_i - \frac{h^2}{12}(-r'_i + p_i r_i)$$

$$b_{i,2,1} = -\frac{h}{2} + \frac{h^2}{12}p_i$$

$$b_{i,2,2} = -1 + \frac{h^2}{12}q_i$$

$$b_{i,2,3} = \frac{h^2}{12}r_i$$

$$b_{i,3,1} = -\frac{h^2}{12}$$

$$b_{i,3,2} = -\frac{h}{2}$$

$$b_{i,3,3} = -1$$

$$e_{i+1,1} = \frac{h}{2}s_{i+1} - \frac{h^2}{12}(-p_{i+1}s_{i+1} + s'_{i+1})$$

$$e_{i+1.2} = -\frac{h^2}{12}s_{i+1}$$

$$e_{i+1,3} = 0$$

$$f_{i,1} = \frac{h}{2}s_i + \frac{h^2}{12}(-p_i s_i + s_i')$$

$$f_{i.2} = \frac{h^2}{12}s_i$$

$$f_{i,3} = 0$$

Here the functions $p_{i+1}, q_{i+1}, r_{i+1}, s_{i+1}$ and their first derivatives are applied at point x_{i+1} and also p_i, q_i, r_i and their first derivatives are applied at point x_i. Note that boundary conditions are converted to $u_{0,0} = \alpha, u_{1,0} = \beta$ and $u_{0,N} = \gamma$.

The result is a block matrix system of $3N$ equations with $3N$ unknown variables, where

$$AU = C \tag{4.2.7}$$

Such that

$$A = \begin{bmatrix} A_1 & & & & & & & B_0 \\ B_1 & A_2 & & & & & & \\ & B_2 & A_3 & & & & & \\ & & \ddots & \ddots & & & & \\ & & & \ddots & \ddots & & & \\ & & & & \ddots & \ddots & & \\ & & & & & B_{N-2} & A_{N-1} & \\ & & & & & & B_{N-1} & A_N \end{bmatrix}$$

Where $B_0 = \begin{bmatrix} 0 & 0 & b_{0,1,1} \\ 0 & 0 & b_{0,2,1} \\ 0 & 0 & b_{0,3,1} \end{bmatrix}$ and $A_N = \begin{bmatrix} a_{N,1,1} & a_{N,1,2} & 0 \\ a_{N,2,1} & a_{N,2,2} & 0 \\ a_{N,3,1} & a_{N,3,2} & 0 \end{bmatrix}$

And U and C are defined as

$$U = \begin{bmatrix} \begin{bmatrix} u_{2,1} \\ u_{1,1} \\ u_{0,1} \end{bmatrix} \\ \begin{bmatrix} u_{2,2} \\ u_{1,2} \\ u_{0,2} \end{bmatrix} \\ \vdots \\ \vdots \\ \begin{bmatrix} u_{2,N-1} \\ u_{1,N-1} \\ u_{0,N-1} \end{bmatrix} \\ \begin{bmatrix} u_{2,N} \\ u_{1,N} \\ u_{2,0} \end{bmatrix} \end{bmatrix} \text{ and } C = \begin{bmatrix} \begin{bmatrix} \mathrm{k}_1 \\ \mathrm{k}_2 \\ \mathrm{k}_3 \end{bmatrix} \\ \begin{bmatrix} c_{2,1} \\ c_{2,2} \\ c_{2,3} \end{bmatrix} \\ \vdots \\ \vdots \\ \begin{bmatrix} c_{N-1,1} \\ c_{N-1,2} \\ c_{N-1,3} \end{bmatrix} \\ \begin{bmatrix} \mathrm{k}_4 \\ \mathrm{k}_5 \\ \mathrm{k}_6 \end{bmatrix} \end{bmatrix}$$

Where

$$\mathrm{k}_1 = c_{1,1} - b_{0,1,2}u_{1,0} - b_{0,1,3}u_{0,0}$$

$$\mathrm{k}_2 = c_{1,2} - b_{0,2,2}u_{1,0} - b_{0,2,3}u_{0,0}$$

$$\mathrm{k}_3 = c_{1,3} - b_{0,3,2}u_{1,0} - b_{0,3,3}u_{0,0}$$

$$\mathrm{k}_4 = c_{N,1} - a_{N,1,3}u_{0,N}$$

$$\mathrm{k}_5 = c_{N,2} - a_{N,2,3}u_{0,N}$$

$$\mathrm{k}_6 = c_{N,3} - a_{N,3,3}u_{0,N}$$

And for $i = 1, 2, \dots, N$

$$c_{i,1} = \frac{h}{2}(s_i + s_{i+1}) + \frac{h^2}{12}(-p_i s_i + s_i' + p_{i+1}s_{i+1} - s_{i+1}')$$

$$c_{i,2} = \frac{h^2}{12}(s_i - s_{i+1})$$

$$c_{i,3} = 0$$

The previous linear system equation (4.2.7) can be solved using LU Decomposition to get the approximated solution u_i for $i = 0,1, \dots, N$.

4.3. Pade Approximation (3,3) For Solving Third Order Two-Point Boundary Value Problems

The Pade Approximation Method (3,3) is the same as (2,2) but we set $\mu = 3$ and $k = 3$, so

$$P_3(z) = 1 + \frac{z}{2} + \frac{z^2}{10} + \frac{z^3}{120}$$
$$Q_3(z) = 1 - \frac{z}{2} + \frac{z^2}{10} - \frac{z^3}{120}$$

Then

$$e^z = \frac{1 + \frac{z}{2} + \frac{z^2}{10} + \frac{z^3}{120}}{1 - \frac{z}{2} + \frac{z^2}{10} - \frac{z^3}{120}}$$

Using the relation that $u(x+h) = e^{hD}u(x)$

$$u(x+h) = \frac{1 + \frac{hD}{2} + \frac{(hD)^2}{10} + \frac{(hD)^3}{120}}{1 - \frac{hD}{2} + \frac{(hD)^2}{10} - \frac{(hD)^3}{120}} u(x)$$

So

$$\left[1 - \frac{hD}{2} + \frac{(hD)^2}{10} - \frac{(hD)^3}{120}\right] u(x+h) = \left[1 + \frac{hD}{2} + \frac{(hD)^2}{10} + \frac{(hD)^3}{120}\right] u(x) \quad (4.3.1)$$

The same as Pade Approximation (3,3), we need the second and the third derivative of equation (4.2.4)

$$D\vec{u} = Q\vec{u} + \vec{P} \quad (4.3.2)$$

$$D^2\vec{u} = Q'\vec{u} + Q^2\vec{u} + Q\vec{P} + \vec{P}' \quad (4.3.3)$$

$$D^3\vec{u} = [Q^3 + Q'' + 2Q'Q + QQ']\vec{u} + Q^2\vec{P} + 2Q'\vec{P} + Q\vec{P}' + \vec{P}'' \quad (4.3.4)$$

By substituting the equations (4.3.2), (4.3.3) and (4.3.4) in equation (4.3.1) and applying the resulted equation into the discrete point x_i, where $i = 0, 1, \dots, N$ and $h = (b-a)/N$.

So we get that

$$A_{i+1}u_{i+1} + B_i u_i = E_{i+1} + F_i\ ,\ for\ i = 0, 1, \dots N-1 \qquad (4.3.5)$$

Where

$$A_{i+1} = \begin{bmatrix} a_{i+1,1,1} & a_{i+1,1,2} & a_{i+1,1,3} \\ a_{i+1,2,1} & a_{i+1,2,2} & a_{i+1,2,3} \\ a_{i+1,3,1} & a_{i+1,3,2} & a_{i+1,3,3} \end{bmatrix}$$

$$B_i = \begin{bmatrix} b_{i,1,1} & b_{i,2,1} & b_{i,1,3} \\ b_{i,2,1} & b_{i,2,2} & b_{i,2,3} \\ b_{i,3,1} & b_{i,3,2} & b_{i,3,3} \end{bmatrix}$$

And

$$E_{i+1} = \begin{bmatrix} e_{i+1.1} \\ e_{i+1,2} \\ e_{i+1,3} \end{bmatrix}\ and\ F_i = \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ f_{i,3} \end{bmatrix}$$

Such that

$$a_{i+1,1,1} = 1 + \frac{h}{2}p_{i+1} + \frac{h^2}{10}\left(-p'_{i+1} + p^2_{i+1} - q_{i+1}\right) - \frac{h^3}{120}(-p^3_{i+1} + 2p_{i+1}q_{i+1} - r_{i+1} - p''_{i+1} + 3p'_{i+1}p_{i+1} - 2q'_{i+1})$$

$$a_{i+1,1,2} = \frac{h}{2}q_{i+1} + \frac{h^2}{10}(-q'_{i+1} + p_{i+1}q_{i+1} - r_{i+1}) - \frac{h^3}{120}(-p^2_{i+1}q_{i+1} + p_{i+1}r_{i+1} + q^2_{i+1} - q''_{i+1} + 2p'_{i+1}q_{i+1} - 2r'_{i+1} + p_{i+1}q'_{i+1})$$

$$a_{i+1,1,3} = \frac{h}{2}r_{i+1} + \frac{h^2}{10}(-r'_{i+1} + p_{i+1}r_{i+1}) - \frac{h^3}{120}(-p^2_{i+1}r_{i+1} + q_{i+1}r_{i+1} - r''_{i+1} + 2p'_{i+1}r_{i+1} + p_{i+1}r'_{i+1})$$

$$a_{i+1,2,1} = -\frac{h}{2} - \frac{h^2}{10}p_{i+1} - \frac{h^3}{120}({p_{i+1}}^2 - q_{i+1} - p'_{i+1})$$

$$a_{i+1,2,2} = 1 - \frac{h^2}{10}q_{i+1} - \frac{h^3}{120}(p_{i+1}q_{i+1} - r_{i+1} - q'_{i+1})$$

$$a_{i+1,2,3} = -\frac{h^2}{10}r_{i+1} - \frac{h^3}{120}(p_{i+1}r_{i+1} - r'_{i+1})$$

$$a_{i+1,3,1} = \frac{h^2}{10} + \frac{h^3}{120}p_{i+1}$$

$$a_{i+1,3,2} = -\frac{h}{2} + \frac{h^3}{120}q_{i+1}$$

$$a_{i+1,3,3} = 1 + \frac{h^3}{120}r_{i+1}$$

And

$$b_{i,1,1} = -1 + \frac{h}{2}p_i - \frac{h^2}{10}\left(-p_i' + p_i^2 - q_i\right) - \frac{h^3}{120}(-p_i^3 + 2p_iq_i - r_i - p_i'' + 3p_i'p_i - 2q_i')$$

$$b_{i,1,2} = \frac{h}{2}q_i - \frac{h^2}{10}(-q_i' + p_iq_i - r_i) - \frac{h^3}{120}(-p_i^2q_i + p_ir_i + q_i^2 - q_i'' + 2p_i'q_i - 2r_i' + p_iq_i')$$

$$b_{i,1,3} = \frac{h}{2}r_i - \frac{h^2}{10}(-r_i' + p_ir_i) - \frac{h^3}{120}(-p_i^2r_i + q_ir_i - r_i'' + 2p_i'r_i + p_ir_i')$$

$$b_{i,2,1} = -\frac{h}{2} + \frac{h^2}{10}p_i - \frac{h^3}{120}({p_i}^2 - q_i - p_i')$$

$$b_{i,2,2} = -1 + \frac{h^2}{10}q_i - \frac{h^3}{120}(p_iq_i - r_i - q_i')$$

$$b_{i,2,3} = \frac{h^2}{10}r_i - \frac{h^3}{120}(p_ir_i - r_i')$$

$$b_{i,3,1} = -\frac{h^2}{10} + \frac{h^3}{120}p_i$$

$$b_{i,3,2} = -\frac{h}{2} + \frac{h^3}{120}q_i$$

$$b_{i,3,3} = -1 + \frac{h^3}{120}r_i$$

$$e_{i+1,1} = \frac{h}{2}s_{i+1} - \frac{h^2}{10}(s_{i+1}' - p_{i+1}s_{i+1}) + \frac{h^3}{120}(-p_{i+1}s_{i+1}' + \left(p_{i+1}^2 - q_{i+1} - 2p_{i+1}'\right)s_{i+1} + s_{i+1}'')$$

$$e_{i+1.2} = -\frac{h^2}{10}s_{i+1} + \frac{h^3}{120}(s_{i+1}' - p_{i+1}s_{i+1})$$

$$e_{i+1,3} = \frac{h^3}{120}s_{i+1}$$

$$f_{i,1} = \frac{h}{2}s_i + \frac{h^2}{10}(s_i' - p_is_i) + \frac{h^3}{120}(-p_is_i' + \left(p_i^2 - q_i - 2p_i'\right)s_i + s_i'')$$

$$f_{i.2} = \frac{h^2}{10}s_i + \frac{h^3}{120}(s_i' - p_is_i)$$

$$f_{i,3} = \frac{h^3}{120}s_i$$

Here the functions $p_{i+1}, q_{i+1}, r_{i+1}, s_{i+1}$ and their first derivatives are applied at point x_{i+1} and also p_i, q_i, r_i and their first and second derivatives

are applied at point x_i. Note that boundary conditions are transferred to $u_{0,0} = \alpha, u_{1,0} = \beta$ and $u_{0,N} = \gamma$.

The result is a block matrix system of $3N$ equations with $3N$ unknown variables, where

$$AU = C \tag{4.3.6}$$

Such that

$$A = \begin{bmatrix} A_1 & & & & & & & B_0 \\ B_1 & A_2 & & & & & & \\ & B_2 & A_3 & & & & & \\ & & \ddots & \ddots & & & & \\ & & & \ddots & \ddots & & & \\ & & & & \ddots & \ddots & & \\ & & & & & B_{N-2} & A_{N-1} & \\ & & & & & & B_{N-1} & A_N \end{bmatrix}$$

Where $B_0 = \begin{bmatrix} 0 & 0 & b_{0,1,1} \\ 0 & 0 & b_{0,2,1} \\ 0 & 0 & b_{0,3,1} \end{bmatrix}$ and $A_N = \begin{bmatrix} a_{N,1,1} & a_{N,1,2} & 0 \\ a_{N,2,1} & a_{N,2,2} & 0 \\ a_{N,3,1} & a_{N,3,2} & 0 \end{bmatrix}$

And U and C defined as

$$\mathrm{U} = \begin{bmatrix} \begin{bmatrix} u_{2,1} \\ u_{1,1} \\ u_{0,1} \end{bmatrix} \\ \begin{bmatrix} u_{2,2} \\ u_{1,2} \\ u_{0,2} \end{bmatrix} \\ \vdots \\ \vdots \\ \begin{bmatrix} u_{2,N-1} \\ u_{1,N-1} \\ u_{0,N-1} \end{bmatrix} \\ \begin{bmatrix} u_{2,N} \\ u_{1,N} \\ u_{2,0} \end{bmatrix} \end{bmatrix} \text{ and } C = \begin{bmatrix} \begin{bmatrix} \mathrm{k}_1 \\ \mathrm{k}_2 \\ \mathrm{k}_3 \end{bmatrix} \\ \begin{bmatrix} c_{2,1} \\ c_{2,2} \\ c_{2,3} \end{bmatrix} \\ \vdots \\ \vdots \\ \begin{bmatrix} c_{N-1,1} \\ c_{N-1,2} \\ c_{N-1,3} \end{bmatrix} \\ \begin{bmatrix} \mathrm{k}_4 \\ \mathrm{k}_5 \\ \mathrm{k}_6 \end{bmatrix} \end{bmatrix}$$

Where

$$\mathrm{k}_1 = c_{1,1} - b_{0,1,2}u_{1,0} - b_{0,1,3}u_{0,0}$$

$$\mathrm{k}_2 = c_{1,2} - b_{0,2,2}u_{1,0} - b_{0,2,3}u_{0,0}$$

$$\mathrm{k}_3 = c_{1,3} - b_{0,3,2}u_{1,0} - b_{0,3,3}u_{0,0}$$

$$\mathrm{k}_4 = c_{N,1} - a_{N,1,3}u_{0,N}$$

$$\mathrm{k}_5 = c_{N,2} - a_{N,2,3}u_{0,N}$$

$$\mathrm{k}_6 = c_{N,3} - a_{N,3,3}u_{0,N}$$

And for $i = 1, 2, \dots, N$

$$c_{i,1} = \frac{h}{2}(s_i + s_{i+1}) + \frac{h^2}{10}(s_i' - p_i s_i - s_{i+1}' + p_{i+1}s_{i+1}) + \frac{h^3}{120}(-p_i s_i' + \left(p_i^2 - q_i - 2p_i'\right)s_i + s_i'' - p_{i+1}s_{i+1}'\left(p_{i+1}^2 - q_{i+1} - 2p_{i+1}'\right)s_{i+1} + s_{i+1}'')$$

$$c_{i,2} = \frac{h^2}{10}(s_i - s_{i+1}) + \frac{h^3}{120}(s_i' - p_i s_i + s_{i+1}' - p_{i+1}s_{i+1})$$

$$c_{i,3} = \frac{h^3}{120}(s_i + s_{i+1})$$

The previous linear system equation (4.3.6) can be solved using LU Decomposition to get the approximated solution u_i for $i = 0,1, \dots, N$.

Chapter Five
Rational Chebyshev Approximation Method

Chapter Five

Rational Chebyshev Approximation Method

5.1. Introduction

Chebyshev polynomial approximation is a method to find good polynomial approximation to a given function $f(x)$ in a given interval $a \leq x \leq b$. The purpose of this method is to find a good approximation for $f(x)$ that is rational function. The reason for doing so is that, for some functions and some intervals, the optimal rational function approximation is able to achieve substantially higher accuracy than the optimal polynomial approximation with the same number of coefficients. This must be weighed against the fact that finding a rational function approximation is not as straightforward as finding a polynomial approximation. Let the desired rational function $R(x)$ have numerator of degree n and denominator of degree m and $T_k(x)$ is the kth-degree Chebyshev polynomial [23], [27], Then we have

$$R_N(x) = \frac{\sum_{k=0}^{n} p_k T_k\,(x)}{\sum_{k=0}^{m} q_k T_k\,(x)}, a \leq x \leq b \qquad (5.1.1)$$

The unknown quantities that we need to find are $p_0, p_1, p_2, \dots, p_n$ and $q_1, q_2, \dots, q_m$, that is, $n + m + 1$ quantities in all, where $N = n + m$ and $q_0 = 1$. Writing $f(x)$ in a series involving Chebyshev polynomial [12] as

$$f(x) = \sum_{k=0}^{\infty} a_k T_k(x) \qquad (5.1.2)$$

Let $r(x)$ denotes the deviation of $R_N(x)$ from $f(x)$, and let r denotes its maximum absolute value as

$$r(x) = f(x) - R_N(x)$$

$$r(x) = \sum_{k=0}^{\infty} a_k T_k(x) - \frac{\sum_{k=0}^{n} p_k T_k(x)}{\sum_{k=0}^{m} q_k T_k(x)}$$

Or

$$r(x) = \frac{\sum_{k=0}^{\infty} a_k T_k(x) \sum_{k=0}^{m} q_k T_k(x) - \sum_{k=0}^{n} p_k T_k(x)}{\sum_{k=0}^{m} q_k T_k(x)} \qquad (5.1.3)$$

$$r = \max_{a \le x \le b} |r(x)|$$

The best solution is to find the values of $p's$ and $q's$ that minimize r or the coefficients $q_1, q_2, \ldots, q_m$ and $p_0, p_1, \ldots, p_n$ are chosen so that the numerator on the right-hand side of equation (5.1.3) has zero coefficients for $T_k(x)$ when $k = 0, 1, \ldots, N$. This implies that the series

$$(a_0 T_0(x) + a_1 T_1(x) + \cdots)\big(T_0(x) + q_1 T_1(x) + \cdots + q_m T_m(x)\big)$$
$$- \big(p_0 T_0(x) + p_1 T_1(x) + \cdots + p_n T_n(x)\big)$$

Has no terms of degree less than or equal to N.

5.6. Rational Chebyshev Approximation Method for Solving Third Order Two-Point Boundary Value Problems

Consider the following equation represented the general form for a third order linear two-point boundary value problem with the following given boundary conditions:

$$y''' + p(x)y'' + q(x)y' + r(x)y = s(x)\,, a \le x \le b \qquad (5.2.1)$$

$$y(a) = \alpha\,, y'(a) = \beta\,, y(b) = \gamma \qquad (5.2.2)$$

System of equations (5.2.1) and (5.2.2) can be converted into first order vector matrix system as following:

Let $u_0 = y, u_1 = y', u_2 = y'', u_2' = y'''$ and

$$u_2' = -p(x)u_2 - q(x)u_1 - r(x)u_0 + s(x)$$

So $\vec{u} = \begin{bmatrix} u_2 \\ u_1 \\ u_0 \end{bmatrix}$ and $\vec{u}' = \begin{bmatrix} u_2' \\ u_1' \\ u_0' \end{bmatrix}$

$$\vec{u}' = \begin{bmatrix} -p(x) & -q(x) & -r(x) \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_2 \\ u_1 \\ u_0 \end{bmatrix} + \begin{bmatrix} s(x) \\ 0 \\ 0 \end{bmatrix}$$

$$D\vec{u} = Q\vec{u} + \vec{P} \tag{5.2.3}$$

Boundary conditions in equation (5.2.2) can also be converted as:

$$u_0(a) = \alpha, u_1(a) = \beta, u_0(b) = \gamma$$

Now let $R(x)$ is the Chebyshev rational approximation for $f(x) = e^x$of degree 6 with $n = m = 3$, then

$$e^x = \frac{a_0 + a_1 x + a_2 x^2 + a_3 x^3}{b_0 + b_1 x + b_2 x^2 + b_3 x^3}$$

The constants a_i and b_i for $i = 0, 1, 2, 3$ will be calculated by Maple using the following command after loading the *orthopoly* and *numapprox* packages

$r := convert(chebyshev(e^x, x), ratpoly, 3,3)$

So using the relation $u(x + h) = e^{hD}u(x)$ and let $u(x + h) = u_{i+1}$

and $u(x) = u_i$, so we get that

$$u_{i+1} = \frac{a_0 + a_1 hD + a_2 (hD)^2 + a_3 (hD)^3}{b_0 + b_1 hD + b_2 (hD)^2 + b_3 (hD)^3} u_i$$

$$[b_0 + b_1 hD + b_2 h^2 D^2 + b_3 h^3 D^3]u_{i+1}$$

$$= [a_0 + a_1 hD + a_2 h^2 D^2 + a_3 h^3 D^3]u_i \tag{5.2.4}$$

By substituting equation (5.2.3) and its second and third derivatives in equation (5.2.4) and applying them on the discrete point x_i, where $x_i = a + ih, i = 0,1, \dots, N$ and $h = (b-a)/N$.

So we get that

$$A_{i+1}u_{i+1} + B_i u_i = E_{i+1} + F_i \,, for\ i = 0, 1, \dots N-1 \qquad (5.2.5)$$

Where

$$A_{i+1} = \begin{bmatrix} a_{i+1,1,1} & a_{i+1,1,2} & a_{i+1,1,3} \\ a_{i+1,2,1} & a_{i+1,2,2} & a_{i+1,2,3} \\ a_{i+1,3,1} & a_{i+1,3,2} & a_{i+1,3,3} \end{bmatrix}$$

$$B_i = \begin{bmatrix} b_{i,1,1} & b_{i,2,1} & b_{i,1,3} \\ b_{i,2,1} & b_{i,2,2} & b_{i,2,3} \\ b_{i,3,1} & b_{i,3,2} & b_{i,3,3} \end{bmatrix}$$

And

$$E_{i+1} = \begin{bmatrix} e_{i+1.1} \\ e_{i+1,2} \\ e_{i+1,3} \end{bmatrix} \ and\ F_i = \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ f_{i,3} \end{bmatrix}$$

Such that

$$a_{i+1,1,1} = b_0 - b_1 h p_{i+1} + b_2 h^2\left(p_{i+1}^2 - q_{i+1} - p'_{i+1}\right) + b_3 h^3(-p_{i+1}^3 + 2p_{i+1}q_{i+1} - r_{i+1} - p''_{i+1} + 3p'_{i+1}p_{i+1} - 2q'_{i+1})$$

$$a_{i+1,1,2} = -b_1 h q_{i+1} + b_2 h^2(p_{i+1}q_{i+1} - r_{i+1} - q'_{i+1}) + b_3 h^3\left(-p_{i+1}^2 q_{i+1} + p_{i+1}r_{i+1} + q_{i+1}^2 - q''_{i+1} + 2p'_{i+1}q_{i+1} - 2r'_{i+1} + p_{i+1}q'_{i+1}\right)$$

$$a_{i+1,1,3} = -b_1 h r_{i+1} + b_2 h^2(p_{i+1}r_{i+1} - r'_{i+1}) + b_3 h^3\left(-p_{i+1}^2 r_{i+1} + q_{i+1}r_{i+1} - r''_{i+1} + 2p'_{i+1}r_{i+1} + p_{i+1}r'_{i+1}\right)$$

$$a_{i+1,2,1} = b_1 h - b_2 h^2 p_{i+1} + b_3 h^3\left(p_{i+1}^2 - q_{i+1} - p'_{i+1}\right)$$

$$a_{i+1,2,2} = b_0 - b_2 h^2 q_{i+1} + b_3 h^3(-q'_{i+1} + p_{i+1}q_{i+1} - r_{i+1})$$

$$a_{i+1,2,3} = -b_2 h^2 r_{i+1} + b_3 h^3(-r'_{i+1} + p_{i+1}r_{i+1})$$

$$a_{i+1,3,1} = b_2 h^2 - b_3 h^3 p_{i+1}$$

$$a_{i+1,3,2} = b_1 h - b_3 h^3 q_{i+1}$$

$$a_{i+1,3,3} = b_0 - b_3 h^3 r_{i+1}$$

And

$$b_{i,1,1} = -a_0 + a_1 h p_i - a_2 h^2 \left(p_i^2 - q_i - p_i'\right) - a_3 h^3 (-p_i^3 + 2p_i q_i - r_i - p_i'' + 3p_i' p_i - 2q_i')$$

$$b_{i,1,2} = a_1 h q_i - a_2 h^2 (p_i q_i - r_i - q_i') - a_3 h^3 \left(-p_i^2 q_i + p_i r_i + q_i^2 - q_i'' + 2p_i' q_i - 2r_i' + p_i q_i'\right)$$

$$b_{i,1,3} = a_1 h r_i - a_2 h^2 (p_i r_i - r_i') - a_3 h^3 \left(-p_i^2 r_i + q_i r_i - r_i'' + 2p_i' r_i + p_i r_i'\right)$$

$$b_{i,2,1} = -a_1 h + a_2 h^2 p_i - a_3 h^3 \left(p_i^2 - q_i - p_i'\right)$$

$$b_{i,2,2} = -a_0 + a_2 h^2 q_i - a_3 h^3 (-q_i' + p_i q_i - r_i)$$

$$b_{i,2,3} = a_2 h^2 r_i - a_3 h^3 (-r_i' + p_i r_i)$$

$$b_{i,3,1} = -a_2 h^2 + a_3 h^3 p_i$$

$$b_{i,3,2} = -a_1 h + a_3 h^3 q_i$$

$$b_{i,3,3} = -a_0 + a_3 h^3 r_i$$

$$e_{i+1,1} = -b_1 h s_{i+1} - b_2 h^2 (-p_{i+1} s_{i+1} + s_{i+1}') - b_3 h^3 \left(\left(p_{i+1}^2 - q_{i+1} - 2p_{i+1}'\right) s_{i+1} - p_{i+1} s_{i+1}' + s_{i+1}''\right)$$

$$e_{i+1,2} = -b_2 h^2 s_{i+1} - b_3 h^3 (-p_{i+1} s_{i+1} + s_{i+1}')$$

$$e_{i+1,3} = -b_3 h^3 s_{i+1}$$

$$f_{i,1} = a_1 h s_i + a_2 h^2 (-p_i s_i + s_i') + a_3 h^3 \left(\left(p_i^2 - q_i - 2p_i'\right) s_i - p_i s_i' + s_i''\right)$$

$$f_{i,2} = a_2 h^2 s_i + a_3 h^3 (-p_i s_i + s_i')$$

$$f_{i,3} = a_3 h^3 s_i$$

Here the functions $p_{i+1}, q_{i+1}, r_{i+1}, s_{i+1}$ and their first derivatives are applied at point x_{i+1} and also p_i, q_i, r_i and their first and second derivatives

are applied at point x_i. Note that boundary conditions are transferred to $u_{0,0} = \alpha, u_{1,0} = \beta$ and $u_{0,N} = \gamma$.

The result is a block matrix system of $3N$ equations with $3N$ unknown variables, where

$$AU = C \tag{5.2.6}$$

Such that

$$A = \begin{bmatrix} A_1 & & & & & & & B_0 \\ B_1 & A_2 & & & & & & \\ & B_2 & A_3 & \ddots & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & B_{N-2} & A_{N-1} & & \\ & & & & & B_{N-1} & A_N & \end{bmatrix}$$

Where $B_0 = \begin{bmatrix} 0 & 0 & b_{0,1,1} \\ 0 & 0 & b_{0,2,1} \\ 0 & 0 & b_{0,3,1} \end{bmatrix}$ and $A_N = \begin{bmatrix} a_{N,1,1} & a_{N,1,2} & 0 \\ a_{N,2,1} & a_{N,2,2} & 0 \\ a_{N,3,1} & a_{N,3,2} & 0 \end{bmatrix}$

And U and C defined as

$$U = \begin{bmatrix} \begin{bmatrix} u_{2,1} \\ u_{1,1} \\ u_{0,1} \end{bmatrix} \\ \begin{bmatrix} u_{2,2} \\ u_{1,2} \\ u_{0,2} \end{bmatrix} \\ \vdots \\ \vdots \\ \begin{bmatrix} u_{2,N-1} \\ u_{1,N-1} \\ u_{0,N-1} \end{bmatrix} \\ \begin{bmatrix} u_{2,N} \\ u_{1,N} \\ u_{2,0} \end{bmatrix} \end{bmatrix} \text{ and } C = \begin{bmatrix} \begin{bmatrix} \mathrm{k}_1 \\ \mathrm{k}_2 \\ \mathrm{k}_3 \end{bmatrix} \\ \begin{bmatrix} c_{2,1} \\ c_{2,2} \\ c_{2,3} \end{bmatrix} \\ \vdots \\ \vdots \\ \begin{bmatrix} c_{N-1,1} \\ c_{N-1,2} \\ c_{N-1,3} \end{bmatrix} \\ \begin{bmatrix} \mathrm{k}_4 \\ \mathrm{k}_5 \\ \mathrm{k}_6 \end{bmatrix} \end{bmatrix}$$

Where

$$\mathrm{k}_1 = c_{1,1} - b_{0,1,2}u_{1,0} - b_{0,1,3}u_{0,0}$$

$$\mathrm{k}_2 = c_{1,2} - b_{0,2,2}u_{1,0} - b_{0,2,3}u_{0,0}$$

$$\mathrm{k}_3 = c_{1,3} - b_{0,3,2}u_{1,0} - b_{0,3,3}u_{0,0}$$

$$\mathrm{k}_4 = c_{N,1} - a_{N,1,3}u_{0,N}$$

$$\mathrm{k}_5 = c_{N,2} - a_{N,2,3}u_{0,N}$$

$$\mathrm{k}_6 = c_{N,3} - a_{N,3,3}u_{0,N}$$

And for $i = 1, 2, \dots, N$

$$c_{i,1} = h[-b_1 s_{i+1} + a_1 s_i] + h^2[-b_2(-p_{i+1}s_{i+1} + s'_{i+1}) + a_2(-p_i s_i + s'_i)] + h^3\left[-b_3\left(\left(p_{i+1}^2 - q_{i+1} - 2p'_{i+1}\right)s_{i+1} - p_{i+1}s'_{i+1} + s''_{i+1}\right) + a_3\left(\left(p_i^2 - q_i - 2p'_i\right)s_i - p_i s'_i + s''_i\right)\right]$$

$$c_{i,2} = h^2[-b_2 s_{i+1} + a_2 s_i] + h^3[-b_3(-p_{i+1}s_{i+1} + s'_{i+1}) + a_3(-p_i s_i + s'_i)]$$

$$c_{i,3} = h^3[-b_3 s_{i+1} + a_3 s_i]$$

The previous linear system shown in equation (5.2.6) can be solved using LU Decomposition to get the approximated solution u_i for $i = 0,1, \dots, N$.

Chapter Six
Quartic B-Spline Method

Chapter Six

Quartic B-Spline Method

6.1. Quartic B-Spline for Solving Third Order Two-Point Boundary Value Problems

There are many different approaches for calculating spline curves. Defining the explanation of basic cubic spline interpolation will set the basis for other more specific types. The problem that all designers faced was how to construct a curve passed through a given $n+1$ points which are $x_0, x_1, \dots, x_n$. B-splines have interesting points called global control; this means that if one of the control points moved, this will affect the entire curve.

The most basic method that which uses cubic spline is B-splines, where B stands for basis [18].

B-splines have a property called local control that means if a position of one data point have been moved this only affects a portion of the curve. This saves the other portions of the curve that may already be acceptable. This makes B-splines useful in geometric design and modeling [9], [25].

A spline function of degree k has a polynomial of degree less or equal to k on each interval $[x_i, x_{i+1}]$, where the spline function has a continuous $(k-1)$ derivative on interval $[a, b]$. There are many types of B-spline, linear, quadratic, cubic and quartic splines. In this chapter we will use Quartic B-Spline for solving third order linear two-point boundary value problem.

For the following third order boundary value problem with the given boundary conditions:

$$y''' = p(x)y'' + q(x)y' + r(x)y + s(x)\,, a \le x \le b \qquad (6.1.1)$$

$$y(a) = \alpha\,, y'(a) = \beta\,, y(b) = \gamma \qquad (6.1.2)$$

The main basic of B-splines is that if we let $y(x)$ be the continuous solution for equation (6.1.1) with boundary conditions given in (6.1.2). Divide the range into n subintervals with step size $h = \frac{b-a}{n}$, by inserting knots at the points $x_0, x_1, \dots, x_n$ where $a = x_0 < x_1 < \cdots < x_n = b$, then $S(x)$ is a quartic spline interpolating function for $y(x)$ if [12]:

I. $S(x)$ is a polynomial of degree four in each interval $[x_i, x_{i+1}]$.

II. $S(x_i) = y(x_i), S'(x_i) = y'(x_i), S''(x_i) = y''(x_i)$ and $S'''(x_i) = y'''(x_i)$.

III. $S(x)$, $S'(x)$, $S''(x)$ and $S'''(x)$ are continuous on the interval $[a, b]$.

IV. The fourth derivatives will be discontinuous on the interval $[a, b]$.

Then $S(x)$ can be written as:

$$S(x) = \sum_{i=-4}^{n-1} c_i \emptyset^5{}_i(x) \qquad (6.1.3)$$

The quartic B-splines are defined on $n + 1$ nodes and the basis function for $S(x)$ is defining by $\emptyset_i(x)$ for $i = -4, -3, \dots, n-2, n-1$ and c_i are the unknown real coefficients.

Therefore, for a given function $y(x)$ where $y(x)$ is the solution for equation (6.1.1) and (6.1.2), there exists a unique spline $S(x)$ that satisfying the interpolation boundary conditions:

$$y(a) = \alpha\,, y'(a) = \beta\,, y(b) = \gamma$$

Quartic B-spline basis could be obtained by calculating the basis up to order five. The first degree B-spline when $k = 1$ will be written as [5], [8], [10], [22]:

$$\emptyset_i^1 = \begin{Bmatrix} 1\ , x \in [x_i, x_{i+1}] \\ 0\ , otherwise \end{Bmatrix} \tag{6.1.4}$$

In general $\emptyset_i^k(x)$ can be defined as:

$$\emptyset_i^k(x) = \frac{x - x_i}{x_{i+k-1} - x_i}\emptyset_i^{k-1} + \frac{x_{i+k} - x}{x_{i+k} - x_{i+1}}\emptyset_{i+1}^{k-1} \tag{6.1.5}$$

For $k = 2$

$$\emptyset_i^2(x) = \frac{x - x_i}{x_{i+1} - x_i}\emptyset_i^1 + \frac{x_{i+2} - x}{x_{i+2} - x_{i+1}}\emptyset_{i+1}^1$$

$$\emptyset_i^2(x) = \frac{1}{h}[(x - x_i)\emptyset_i^1 + (x_{i+2} - x)\emptyset_{i+1}^1]$$

For $k = 3$, we get that

$$\emptyset_i^3(x) = \frac{x - x_i}{x_{i+2} - x_i}\emptyset_i^2 + \frac{x_{i+3} - x}{x_{i+3} - x_{i+1}}\emptyset_{i+1}^2$$

$$\emptyset_i^3(x) = \frac{1}{2h}[(x - x_i)\emptyset_i^2 + (x_{i+3} - x)\emptyset_{i+1}^2]$$

$$\emptyset_i^3(x) = \frac{1}{2h^2}[(x - x_i)^2\emptyset_i^1 + [(x - x_i)(x_{i+2} - x) + (x - x_{i+1})(x_{i+3} - x)]\emptyset_{i+1}^1 + (x_{i+3} - x)^2\emptyset_{i+2}^1]$$

And for $k = 4$, we get that

$$\emptyset_i^4(x) = \frac{x - x_i}{x_{i+3} - x_i}\emptyset_i^3 + \frac{x_{i+4} - x}{x_{i+4} - x_{i+1}}\emptyset_{i+1}^3$$

$$\emptyset_i^4(x) = \frac{1}{3h}[(x - x_i)\emptyset_i^3 + (x_{i+4} - x)\emptyset_{i+1}^3]$$

$$\emptyset_i^4(x) = \frac{1}{6h^3}[(x - x_i)^3\emptyset_i^1 + [(x - x_i)^2(x_{i+2} - x) + (x - x_i)(x_{i+3} - x)(x - x_{i+1}) + (x_{i+4} - x)(x - x_{i+1})^2]\emptyset_{i+1}^1 + [(x - x_i)(x_{i+3} - x)^2 + (x_{i+4} - x)(x - x_{i+1})(x_{i+3} - x) + (x_{i+4} - x)^2(x - x_{i+2})]\emptyset_{i+2}^1 + (x_{i+4} - x)^3\emptyset_{i+3}^1]$$

Finally for $k = 5$, $\emptyset_i^5(x)$ will be defined as:

$$\emptyset_i^5(x) = \frac{x - x_i}{x_{i+4} - x_i}\emptyset_i^4 + \frac{x_{i+5} - x}{x_{i+5} - x_{i+1}}\emptyset_{i+1}^4$$

$$\emptyset_i^5(x) = \frac{1}{4h}[(x - x_i)\emptyset_i^4 + (x_{i+5} - x)\emptyset_{i+1}^4]$$

And also $\emptyset_i^5(x)$ for the interval $[x_i, x_{i+5}]$ can be written as *:*

$$\begin{aligned}\emptyset_i^5(x) = \frac{1}{24h^4}[&(x - x_i)^4\emptyset_i^1 + [(x - x_i)^3(x_{i+2} - x)\\ &+ (x - x_i)^2(x_{i+3} - x)(x\\ &- x_{i+1}) + (x - x_i)(x_{i+4} - x)(x - x_{i+1})^2\\ &+ (x_{i+5} - x)(x - x_{i+1})^3]\emptyset_{i+1}^1 + [(x - x_i)^2(x_{i+3} - x)^2\\ &+ (x - x_i)(x_{i+4} - x)(x - x_{i+1})(x_{i+3} - x)\\ &+ (x - x_i)(x_{i+4} - x)^2(x - x_{i+2})\\ &+ (x_{i+5} - x)(x - x_{i+1})^2(x_{i+3} - x)\\ &+ (x_{i+5} - x)(x - x_{i+1})(x_{i+4} - x)(x - x_{i+2})\\ &+ (x_{i+5} - x)^2(x - x_{i+2})^2]\ \emptyset_{i+2}^1\\ &+ [(x - x_i)(x_{i+4} - x)^3 + (x_{i+5} - x)(x - x_{i+1})(x_{i+4} - x)^2\\ &+ (x_{i+5} - x)^2(x - x_{i+2})(x_{i+4} - x) + (x_{i+5} - x)^3(x\\ &- x_{i+3})]\emptyset_{i+3}^1 + (x_{i+5} - x)^4\emptyset_{i+4}^1]\end{aligned}$$

Following equation (6.1.4), $\emptyset_{i+1}^1$, $\emptyset_{i+2}^1$, $\emptyset_{i+3}^1$ and $\emptyset_{i+4}^1$ can be written as:

$$\emptyset_{i+1}^1 = \begin{Bmatrix}1\ , x \in [x_{i+1}, x_{i+2}]\\ 0\ , otherwise\end{Bmatrix}$$

$$\emptyset_{i+2}^1 = \begin{Bmatrix}1\ , x \in [x_{i+2}, x_{i+3}]\\ 0\ , otherwise\end{Bmatrix}$$

$$\emptyset_{i+3}^1 = \begin{Bmatrix}1\ , x \in [x_{i+3}, x_{i+4}]\\ 0\ , otherwise\end{Bmatrix}$$

and

$$\emptyset_{i+4}^1 = \begin{Bmatrix}1\ , x \in [x_{i+4}, x_{i+5}]\\ 0\ , otherwise\end{Bmatrix}$$

For $x \in [x_i, x_{i+1}]$, where $\emptyset_i^1 = 1$ and the other terms equal zeros

$$\emptyset_i^5(x) = \frac{1}{24h^4}(x - x_i)^4$$

For $x \in [x_{i+1}, x_{i+2}]$, where $\emptyset_{i+1}^1 = 1$ and the other terms equal zeros

$$\emptyset_i^5(x) = \frac{1}{24h^4}[(x - x_i)^3(x_{i+2} - x) + (x - x_i)^2(x_{i+3} - x)(x - x_{i+1}) + (x - x_i)(x_{i+4} - x)(x - x_{i+1})^2 + (x_{i+5} - x)(x - x_{i+1})^3]$$

For $x \in [x_{i+2}, x_{i+3}]$, where $\emptyset_{i+2}^1 = 1$ and the other terms equal zeros

$$\begin{aligned}\emptyset_i^5(x) = \frac{1}{24h^4}[&(x - x_i)^2(x_{i+3} - x)^2 + (x - x_i)(x_{i+4} - x)(x - x_{i+1})(x_{i+3} - x) \\ &+ (x - x_i)(x_{i+4} - x)^2(x - x_{i+2}) \\ &+ (x_{i+5} - x)(x - x_{i+1})^2(x_{i+3} - x) \\ &+ (x_{i+5} - x)(x - x_{i+1})(x_{i+4} - x)(x - x_{i+2}) \\ &+ (x_{i+5} - x)^2(x - x_{i+2})^2]\end{aligned}$$

For $x \in [x_{i+3}, x_{i+4}]$, where $\emptyset_{i+3}^1 = 1$ and the other terms equal zeros

$$\emptyset_i^5(x) = \frac{1}{24h^4}[(x - x_i)(x_{i+4} - x)^3 + (x_{i+5} - x)(x - x_{i+1})(x_{i+4} - x)^2 + (x_{i+5} - x)^2(x - x_{i+2})(x_{i+4} - x) + (x_{i+5} - x)^3(x - x_{i+3})]$$

For $x \in [x_{i+4}, x_{i+5}]$, where $\emptyset_{i+4}^1 = 1$ and the other terms equal zeros

$$\emptyset_i^5(x) = \frac{1}{24h^4}(x_{i+5} - x)^4$$

Finally $\emptyset_i^5(x)$ can be written as piecewise function:

$$\emptyset_i^5(x) = \frac{1}{24h^4}\begin{cases} (x - x_i)^4, x \in [x_i, x_{i+1}] \\ (x - x_i)^3(x_{i+2} - x) + (x - x_i)^2(x_{i+3} - x)(x - x_{i+1}) + \\ (x - x_i)(x_{i+4} - x)(x - x_{i+1})^2 + (x_{i+5} - x)(x - x_{i+1})^3, x \in [x_{i+1}, x_{i+2}] \\ (x - x_i)^2(x_{i+3} - x)^2 + (x - x_i)(x_{i+4} - x)(x - x_{i+1})(x_{i+3} - x) + \\ (x - x_i)(x_{i+4} - x)^2(x - x_{i+2}) + (x_{i+5} - x)(x - x_{i+1})^2(x_{i+3} - x) + \\ (x_{i+5} - x)(x - x_{i+1})(x_{i+4} - x)(x - x_{i+2}) + \\ (x_{i+5} - x)^2(x - x_{i+2})^2, x \in [x_{i+2}, x_{i+3}] \\ (x - x_i)(x_{i+4} - x)^3 + (x_{i+5} - x)(x - x_{i+1})(x_{i+4} - x)^2 + \\ (x_{i+5} - x)^2(x - x_{i+2})(x_{i+4} - x) + (x_{i+5} - x)^3(x - x_{i+3}), x \in [x_{i+3}, x_{i+4}] \\ (x_{i+5} - x)^4, x \in [x_{i+4}, x_{i+5}] \end{cases} \quad (6.1.6)$$

Note that $\emptyset_i^5(x)$ is continuous on the first, second and third derivatives, as the following table:

Table 6.1: the first, second and the third derivatives for the basis functions $\emptyset_i^5(x)$

	$\emptyset_i(x)$	$\emptyset_i'(x)$	$\emptyset_i''(x)$	$\emptyset_i'''(x)$
x_i	0	0	0	0
x_{i+1}	$1/24$	$1/6\,h$	$1/2h^2$	$1/h^3$
x_{i+2}	$11/24$	$1/2h$	$-1/2h^2$	$-3/h^3$
x_{i+3}	$11/24$	$-1/2\,h$	$-1/2h^2$	$3/h^3$
x_{i+4}	$1/24$	$-1/6\,h$	$1/2h^2$	$-1/h^3$
x_{i+5}	0	0	0	0

So, the solution of the equations (6.1.1) and (6.1.2) will be approximated as

$$S(x_i) = y(x_i) \approx \sum_{i=-4}^{n-1} c_i \emptyset^5{}_i(x_i), \qquad 0 \le i \le n$$

For the point x_i , there are four nonzero quartic B-spline basis $\emptyset_{i-4}^5(x_i), \emptyset_{i-3}^5(x_i), \emptyset_{i-2}^5(x_i)$ and $\emptyset_{i-1}^5(x_i)$, so the approximation solution at the point x_i can be written as

$$S(x_i) = c_{i-4}\emptyset^5{}_{i-4}(x_i) + c_{i-3}\emptyset^5{}_{i-3}(x_i) + c_{i-2}\emptyset^5{}_{i-2}(x_i) + c_{i-1}\emptyset^5{}_{i-1}(x_i) \approx y(x_i) \qquad (6.1.7)$$

And also

$$S'(x_i) = c_{i-4}\emptyset^5{}_{i-4}{}'(x_i) + c_{i-3}\emptyset^5{}_{i-3}{}'(x_i) + c_{i-2}\emptyset^5{}_{i-2}{}'(x_i) + c_{i-1}\emptyset^5{}_{i-1}{}'(x_i) \approx y'(x_i) \qquad (6.1.8)$$

$$S''(x_i) = c_{i-4}\emptyset^5{}_{i-4}{}''(x_i) + c_{i-3}\emptyset^5{}_{i-3}{}''(x_i) + c_{i-2}\emptyset^5{}_{i-2}{}''(x_i) + c_{i-1}\emptyset^5{}_{i-1}{}''(x_i) \approx y''(x_i) \qquad (6.1.9)$$

And

$$S'''(x_i) = c_{i-4}\emptyset^5{}_{i-4}'''(x_i) + c_{i-3}\emptyset^5{}_{i-3}'''(x_i) + c_{i-2}\emptyset^5{}_{i-2}'''(x_i) + c_{i-1}\emptyset^5{}_{i-1}'''(x_i) \approx y'''(x_i) \quad (6.1.10)$$

By substituting the relation (6.1.6) in equations (6.1.7-6.1.10), we obtain that:

$$S(x_i) = \frac{1}{24}(c_{i-4} + 11c_{i-3} + 11c_{i-2} + c_{i-1}) \quad (6.1.11)$$

$$S'(x_i) = \frac{1}{6h}(-c_{i-4} - 3c_{i-3} + 3c_{i-2} + c_{i-1}) \quad (6.1.12)$$

$$S''(x_i) = \frac{1}{2h^2}(c_{i-4} - c_{i-3} - c_{i-2} + c_{i-1}) \quad (6.1.13)$$

$$S'''(x_i) = \frac{1}{h^3}(-c_{i-4} + 3c_{i-3} - 3c_{i-2} + c_{i-1}) \quad (6.1.14)$$

Since $S(x_i) \approx y(x_i)$, $S'(x_i) \approx y'(x_i)$, $S''(x_i) \approx y''(x_i)$ and $S'''(x_i) \approx y'''(x_i)$, we obtain that

$$S'''(x_i) = p(x_i)S''(x_i) + q(x_i)S'(x_i) + r(x_i)S(x_i) + s(x_i) \quad (6.1.15)$$

And the boundary conditions in equation (6.1.2) will be shown as:

$$S(x_0) = \alpha, S'(x_0) = \beta, S(x_n) = \gamma \quad (6.1.16)$$

By substituting the equations (6.1.11-6.1.14) in equation (6.1.15), we obtain that

$\frac{1}{h^3}(-c_{i-4} + 3c_{i-3} - 3c_{i-2} + c_{i-1}) = p(x_i)[\frac{1}{2h^2}(c_{i-4} - c_{i-3} - c_{i-2} + c_{i-1})] + q(x_i)[\frac{1}{6h}(-c_{i-4} - 3c_{i-3} + 3c_{i-2} + c_{i-1})] + r(x_i)[\frac{1}{24}(c_{i-4} + 11c_{i-3} + 11c_{i-2} + c_{i-1})] + s(x_i)$

$$-c_{i-4} + 3c_{i-3} - 3c_{i-2} + c_{i-1} = p_i\left[\frac{h}{2}(c_{i-4} - c_{i-3} - c_{i-2} + c_{i-1})\right] + q_i\left[\frac{h^2}{6}(-c_{i-4} - 3c_{i-3} + 3c_{i-2} + c_{i-1})\right] + r_i\left[\frac{h^3}{24}(c_{i-4} + 11c_{i-3} + 11c_{i-2} + c_{i-1})\right] + h^3 s_i$$

$$\left[-1-\frac{h}{2}p_i+\frac{h^2}{6}q_i-\frac{h^3}{24}r_i\right]c_{i-4}+\left[3+\frac{h}{2}p_i+\frac{h^2}{2}q_i-\frac{11h^3}{24}r_i\right]c_{i-3}$$
$$+\left[-3+\frac{h}{2}p_i-\frac{h^2}{2}q_i-\frac{11h^3}{24}r_i\right]c_{i-2}$$
$$+\left[1-\frac{h}{2}p_i-\frac{h^2}{6}q_i-\frac{h^3}{24}r_i\right]c_{i-1}=h^3s_i$$

Let

$$a_i=-1-\frac{h}{2}p_i+\frac{h^2}{6}q_i-\frac{h^3}{24}r_i$$
$$b_i=3+\frac{h}{2}p_i+\frac{h^2}{2}q_i-\frac{11h^3}{24}r_i$$
$$d_i=-3+\frac{h}{2}p_i-\frac{h^2}{2}q_i-\frac{11h^3}{24}r_i$$
$$e_i=1-\frac{h}{2}p_i-\frac{h^2}{6}q_i-\frac{h^3}{24}r_i$$
$$k_i=h^3s_i$$

And

$$a_ic_{i-4}+b_ic_{i-3}+d_ic_{i-2}+e_ic_{i-1}=k_i, i=0,1,\dots,n \qquad (6.1.17)$$

Also substituting (6.1.11) and (6.1.12) in boundary conditions (6.1.16) we get that

$S(a)=\alpha$, when $i=0$

$$S(x_0)=\frac{1}{24}(c_{-4}+11c_{-3}+11c_{-2}+c_{-1})$$
$$\frac{1}{24}(c_{-4}+11c_{-3}+11c_{-2}+c_{-1})=\alpha \qquad (6.1.18)$$

$S'(a)=\beta$, when $i=0$

$$S'(x_0)=\frac{1}{6h}(-c_{-4}-3c_{-3}+3c_{-2}+c_{-1})$$
$$\frac{1}{6h}(-c_{-4}-3c_{-3}+3c_{-2}+c_{-1})=\beta \qquad (6.1.19)$$

$S(b)=\gamma$, when $i=n$

$$S(x_n)=\frac{1}{24}(c_{n-4}+11c_{n-3}+11c_{n-2}+c_{n-1})$$

$$\frac{1}{24}(c_{n-4} + 11c_{n-3} + 11c_{n-2} + c_{n-1}) = \gamma \qquad (6.1.20)$$

The equations (6.1.17), (6.1.18), (6.1.19) and (6.1.20) form $n + 4$ linear equations with $n + 4$ unknown coefficients $c_i, i = -4, -3, \dots, n-2, n-1$

Let F is a vector matrix of size $n + 4$, where $F_0 = \alpha, F_1 = \beta, F_{n+4} = \gamma$ and

$$F_{i+3} = k_i, i = 0,1, \dots, n$$

In vector matrix form, $Ac = F$, where

$$A = \begin{bmatrix} \frac{1}{24} & \frac{11}{24} & \frac{11}{24} & \frac{1}{24} & 0 & \cdots & & \cdots & & \cdots & 0 \\ \frac{-1}{6h} & \frac{-1}{2h} & \frac{1}{2h} & \frac{1}{6h} & 0 & \ddots & & & & & \vdots \\ a_0 & b_0 & d_0 & e_0 & 0 & \ddots & & & & & \\ 0 & a_1 & b_1 & d_1 & e_1 & \ddots & & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & \ddots & & & \\ & & & & & & & \ddots & & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & & & & & & & \ddots & \vdots \\ & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & & & 0 & a_n & b_n & d_n & e_n \\ 0 & \cdots & & \cdots & & \cdots & 0 & \frac{1}{24} & \frac{11}{24} & \frac{11}{24} & \frac{1}{24} \end{bmatrix}, F = \begin{bmatrix} \alpha \\ \beta \\ k_0 \\ k_1 \\ \vdots \\ \vdots \\ \\ \vdots \\ \vdots \\ k_n \\ \gamma \end{bmatrix}$$

And

$$c = [c_{-4}, c_{-3}, c_{-2}, \dots \dots, c_{n-3}, c_{n-2}, c_{n-1}]^T$$

This forms a linear system of $n + 4$ linear equations with $n + 4$ unknown coefficients; it can be solved using direct method such as LU Decomposition or any method of iterative methods such as Gauss-Seidel.

Chapter Seven
Numerical Comparison and Examples

Chapter Seven

Numerical Comparison and Examples

7.1 Linear Shooting Method Algorithm

The following algorithm implements the Linear Shooting Method for solving linear third order two-point boundary value problem using the Matlab software.

Algorithm 7.1

To approximate the solution of the following third order boundary value problem:

$$y''' = p(x)y'' + q(x)y' + r(x)y + s(x)\,, a \le x \le b$$

with $y(a) = \alpha, y'(a) = \beta, y(b) = \gamma$

Inputs:

Endpoints a and b, boundary conditions $\alpha, \beta \; and \; \gamma$ and number of subintervals N.

Outputs:

$x_i\,, W1_i$: Approximated value, $Exact_i \; and \; Error_i$

for each $i = 1\,,2\,, \ldots\,, N+1$

Step 1:

Set $h = \frac{b-a}{N}$

$u_{1,0} = \alpha\,, u_{2,0} = 0\,, u_{3,0} = 0$

$v_{1,0} = 0\,, v_{2,0} = 1\,, v_{3,0} = 0$

Step 2:

For $i = 0, 1, \ldots, N-1$, do step 3 and 4 (The Runge-Kutta method is used in step 3 and 4)

Step 3:

Set $x = a + ih$

Step 4:

$$k_{1,1} = hu_{2,i}$$

$$k_{1,2} = hu_{3,i}$$

$$k_{1,3} = h[p(x)u_{3,i} + q(x)u_{2,i} + r(x)u_{1,i} + s(x)]$$

$$k_{2,1} = h[u_{2,i} + \frac{1}{2}k_{1,2}]$$

$$k_{2,2} = h[u_{3,i} + \frac{1}{2}k_{1,3}]$$

$$k_{2,3} = h[p(x + \frac{h}{2})(u_{3,i} + \frac{1}{2}k_{1,3}) + q(x + \frac{h}{2})(u_{2,i} + \frac{1}{2}k_{1,2}) + r(x + \frac{h}{2})(u_{1,i} + \frac{1}{2}k_{1,1}) + s(x + \frac{h}{2})]$$

$$k_{3,1} = h[u_{2,i} + \frac{1}{2}k_{2,2}]$$

$$k_{3,2} = h[u_{3,i} + \frac{1}{2}k_{2,3}]$$

$$k_{3,3} = h[p(x + \frac{h}{2})(u_{3,i} + \frac{1}{2}k_{2,3}) + q(x + \frac{h}{2})(u_{2,i} + \frac{1}{2}k_{2,2}) + r(x + \frac{h}{2})(u_{1,i} + \frac{1}{2}k_{2,1}) + s(x + \frac{h}{2})]$$

$$k_{4,1} = h[u_{2,i} + \frac{1}{2}k_{3,2}]$$

$$k_{4,2} = h[u_{3,i} + \frac{1}{2}k_{3,3}]$$

$$k_{4,3} = h[p(x + h)(u_{3,i} + k_{3,3}) + q(x + h)(u_{2,i} + k_{3,2}) + r(x + h)(u_{1,i} + k_{3,1}) + s(x + h)]$$

$$u_{1,i+1} = u_{1,i} + \frac{1}{6}[k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}]$$

$$u_{2,i+1} = u_{2,i} + \frac{1}{6}[k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}]$$

$$u_{3,i+1} = u_{3,i} + \frac{1}{6}[k_{1,3} + 2k_{2,3} + 2k_{3,3} + k_{4,3}]$$

$$j_{1,1} = hv_{2,i}$$

$$j_{1,2} = hv_{3,i}$$

$$j_{1,3} = h[p(x)v_{3,i} + q(x)v_{2,i} + r(x)v_{1,i}]$$

$$j_{2,1} = h[v_{2,i} + \frac{1}{2}j_{1,2}]$$

$$j_{2,2} = h[v_{3,i} + \frac{1}{2}j_{1,3}]$$

$$j_{2,3} = h[p(x + \frac{h}{2})(v_{3,i} + \frac{1}{2}j_{1,3}) + q(x + \frac{h}{2})(v_{2,i} + \frac{1}{2}j_{1,2}) + r(x + \frac{h}{2})(v_{1,i} + \frac{1}{2}j_{1,1})]$$

$$j_{3,1} = h[v_{2,i} + \frac{1}{2}j_{2,2}]$$

$$j_{3,2} = h[v_{3,i} + \frac{1}{2}j_{2,3}]$$

$$j_{3,3} = h[p(x + \frac{h}{2})(v_{3,i} + \frac{1}{2}j_{2,3}) + q(x + \frac{h}{2})(v_{2,i} + \frac{1}{2}j_{2,2}) + r(x + \frac{h}{2})(v_{1,i} + \frac{1}{2}j_{2,1})]$$

$$j_{4,1} = h[v_{2,i} + \frac{1}{2}j_{3,2}]$$

$$j_{4,2} = h[v_{3,i} + \frac{1}{2}j_{3,3}]$$

$$j_{4,3} = h[p(x + h)(v_{3,i} + j_{3,3}) + q(x + h)(v_{2,i} + j_{3,2}) + r(x + h)(v_{1,i} + j_{3,1})]$$

$$v_{1,i+1} = v_{1,i} + \frac{1}{6}[j_{1,1} + 2j_{2,1} + 2j_{3,1} + j_{4,1}]$$

$$v_{2,i+1} = v_{2,i} + \frac{1}{6}[j_{1,2} + 2j_{2,2} + 2j_{3,2} + j_{4,2}]$$

$$v_{3,i+1} = v_{3,i} + \frac{1}{6}[j_{1,3} + 2j_{2,3} + 2j_{3,3} + j_{4,3}]$$

Step 5:

Set $w_{1,0} = \alpha$

$$w_{2,0} = \frac{\gamma - u_{1,N}}{v_{1,N}}$$

Step 6:

For $i = 0, 1, \ldots, N$

Set $W1_i = u_{1,i} + w_{2,0} v_{1,i}$

$W2_i = u_{2,i} + w_{2,0} v_{2,i}$

$W3_i = u_{2,i} + w_{2,0} v_{2,i}$

Step 8:

$For\ i = 0, 1, \ldots, N$

Set

$x_i = a + i * h$

$err_i = abs(Exact_i - W1_i)$

$Output(\ x_i\ , Exact_i\ , W1_i\ , err_i\ )$

Stop (The process is complete)

7.2. Finite Difference Method Algorithm

Algorithm 7.2

To approximate the solution of the following third order boundary value problem:

$y''' = p(x)y'' + q(x)y' + r(x)y + s(x)\ , a \leq x \leq b,$

$$y(a) = \alpha, y'(a) = \beta, y(b) = \gamma$$

Inputs:

Endpoints a and b – boundary conditions α, β and γ – number of subintervals *N*.

Outputs:

x_i , W_i: Approximated value,Exact_i and Error_i for each $i = 1, 2, \ldots, N + 1$

Step 1:

$h = \frac{b - a}{N}$

Step 2:

For $i = 1, 2, \ldots, N + 1$ do step 3 and 4

Step 3:

Set $x = a + (i - 1) * h$;

Step 4:

Set $a_i = 1 - \frac{h}{2} p(x_i)$

$b_i = -2 - h^2 q(x_i)$

$c_i = 1 + \frac{h}{2} p(x_i)$

$d_i = -h^2 r(x_i)$

$e_i = h^2 s(x_i)$

Step 5:

Define the five diagonals $A1, B1, C1, D1, k1$ for the matrix A

Step 6:

Set $A1_2 = c_1,\ A1_{2N+1} = \frac{h}{2},\ A1_{2N+2} = a_{N+1}$

$B1_1 = \frac{h}{2}, B1_{2N} = a_N$

$C1_{2N} = \frac{h}{2}, C1_{2N+1} = b_{N+1}$

$k1_1 = 0, k1_2 = 0, k1_{2N-1} = 1$

Step 7:

For $i = 2, 3, \ldots, N$ set

$A1_{2i} = b_i$

Step 8:

For $i = 1, 3, 5, \ldots, 2N - 3$ set

$B1_{i+2} = \frac{-h}{2}$

Step 9:

For $i = 2, 4, 6, \ldots, 2N - 4$ set

$C1_{i+2} = \frac{-h}{2}$

Step 10:

For $i = 1, 2, \ldots, N - 1$ set

$C1_{2(i-1)+3} = d_{i+1}$

$D1_{2(i-1)+2} = a_i$

$k1_{2i+2} = c_{i+2}$

Step 11:

Define$f$, set

$f_1 = \alpha - \frac{h}{2}\beta$

$f_2 = e_1 - b_1\beta - d_1\alpha$

$f_3 = \alpha + \frac{h}{2}\beta$

$f_4 = e_2 - c_1\beta$

$f_{2N+1} = \gamma$

$f_{2N+2} = e_{N+1} - d_{N+1}\gamma$

Step 12:

For $i = 5, 7, \ldots, 2N - 1$ set

$f_i = 0$

Step 13:

For $i = 6, 8, \ldots, 2N$ set

$f_i = e_i$

Step 14:

Find $y = inv(A) * f$

Step 15:

Set $W_1 = \alpha, W_{N+1} = \gamma$

For $i = 2, 3, \ldots, N$ set

$W_i = y_{2i-1}$

Step 16:

$For\ i = 1, 2, \ldots, N + 1$ Set

$x_i = a + (i - 1) * h$

$err_i = abs(Exact_i - W_i)$

$Output(\ x_i\ , Exact_i\ , W_i\ , err_i\ )$

Step 17:

Stop (The process is complete)

7.3. Pade Approximation (2,2) Method Algorithm

We will test Pade Approximation (2,2) and (3,3) for solving linear third order two-point boundary value problem.

The following algorithm implements the Pade Approximation Method (2,2) for solving linear third order two-point boundary value problem using Matlab software.

Algorithm 7.3

To approximate the solution of the following third order boundary value problem:

$$y''' + p(x)y'' + q(x)y' + r(x)y = s(x)\ , a \le x \le b,$$

$$y(a) = \alpha, y'(a) = \beta, y(b) = \gamma$$

Inputs:

Endpoints a and b – boundary conditions α, β and γ – number of subintervals N.

Outputs:

$x_i\ , W_i$: Approximated value,Exact_i and Error_i for each $i = 1\ ,2\ , \dots\ , N$

Step 1:

$$h = \frac{b-a}{N}$$

Step 2:

For $i = 1, 2, \dots, N + 1$ do step 3 and 4

Step 3:

Set

$x = a + (i - 1) * h$

Step 4:

Set

$$a_{i,1,1} = 1 + \frac{h}{2} p_i + \frac{h^2}{12}(-p_i' + p_i^2 - q_i)$$
$$a_{i,1,2} = \frac{h}{2} q_i + \frac{h^2}{12}(-q_i' + p_i q_i - r_i)$$
$$a_{i,1,3} = \frac{h}{2} r_i + \frac{h^2}{12}(-r_i' + p_i r_i)$$
$$a_{i,2,1} = -\frac{h}{2} - \frac{h^2}{12} p_i$$
$$a_{i,2,2} = 1 - \frac{h^2}{12} q_i$$
$$a_{i,2,3} = -\frac{h^2}{12} r_i$$
$$a_{i,3,1} = \frac{h^2}{12}$$
$$a_{i,3,2} = -\frac{h}{2}$$
$$a_{i,3,3} = 1$$
$$b_{i,1,1} = -1 + \frac{h}{2} p_i - \frac{h^2}{12}(-p_i' + p_i^2 - q_i)$$
$$b_{i,1,2} = \frac{h}{2} q_i - \frac{h^2}{12}(-q_i' + p_i q_i - r_i)$$
$$b_{i,1,3} = \frac{h}{2} r_i - \frac{h^2}{12}(-r_i' + p_i r_i)$$
$$b_{i,2,1} = -\frac{h}{2} + \frac{h^2}{12} p_i$$
$$b_{i,2,2} = -1 + \frac{h^2}{12} q_i$$

$b_{i,2,3} = \frac{h^2}{12} r_i$

$b_{i,3,1} = -\frac{h^2}{12}$

$b_{i,3,2} = -\frac{h}{2}$

$b_{i,3,3} = -1$

Step 5:

For $i = 1, 2, \dots, N$ do step 6

Step 6:

Set

$c_{i,1} = \frac{h}{2}(s_i + s_{i+1}) + \frac{h^2}{12}(-p_i s_i + s_i' + p_{i+1} s_{i+1} - s_{i+1}')$

$c_{i,2} = \frac{h^2}{12}(s_i - s_{i+1})$

$c_{i,3} = 0$

Step 7:

Define the five diagonals a, b, c, d, e, f, k, l for the matrix A and the vector column C.

Step 8:

For $i = 1, 2, \dots, N$ set

$a_{(3i-2)} = a_{i+1,1,1}$

$a_{(3i-1)} = a_{i+1,2,2}$

$a_{3i} = a_{i+1,3,3}$

$b_{(3i-2)} = a_{i+1,1,2}$

$b_{(3i-1)} = a_{i+1,2,3}$

$c_{(3i-2)} = a_{i+1,2,1}$

$c_{(3i-1)} = a_{i+1,3,2}$

$d_{(3i-2)} = a_{i+1,1,3}$

$e_{(3i-2)} = a_{i+1,3,1}$

Step 9:

For $i = 1, 2, \dots, N-1$ set

$c_{3i} = b_{i+1,1,3}$

$e_{(3i-1)} = b_{i+1,1,2}$

$e_{3i} = b_{i+1,2,3}$

$f_{(3i-2)} = b_{i+1,1,1}$

$f_{(3i-1)} = b_{i+1,2,2}$

$f_{3i} = b_{i+1,3,3}$

$k_{(3i-2)} = b_{i+1,2,1}$

$k_{(3i-1)} = b_{i+1,3,2}$

$l_{(3i-2)} = b_{i+1,3,1}$

Step 10:

For $i = 1, 2, \dots, N-2$ set

$C_{3i+1} = c_{i+1,1}$

$C_{3i+2} = c_{i+1,2}$

$C_{3i+3} = c_{i+1,3}$

Step 11:

Set

$A_{1,3N} = b_{1,1,1}$

$A_{2,3N} = b_{1,2,1}$

$A_{3,3N} = b_{1,3,1}$

$A_{3N-2,3N} = 0$

$A_{3N-1,3N} = 0$

$A_{3N,3N} = 0$

$C_1 = c_{1,1} - b_{1,1,2}\beta - b_{1,1,3}\alpha$

$C_2 = c_{1,2} - b_{1,2,2}\beta - b_{1,2,3}\alpha$

$C_3 = c_{1,3} - b_{1,3,2}\beta - b_{1,3,3}\alpha$

$C_{3N-2} = c_{N,1} - a_{N,1,3}\gamma$

$C_{3N-1} = c_{N,2} - a_{N,2,3}\gamma$

$C_{3N} = c_{N,3} - a_{N,3,3}\gamma$

Step 12:

$[l, u] = lu(A)$

$t = inv(l) * C$

$y = inv(u) * t$

Step 13:

Set

$W_1 = \alpha$

$W_{N+1} = \gamma$

Step 14:

For $i = 1, 2, \dots, N-1$ set

$W_{i+1} = y_{3i}$

Step 15:

$For\ i = 1\ ,2\ , \dots , N + 1$ Set

$x_i = a + (i - 1) * h$

$err_i = abs(Exact_i - W_i)$

$Output(\ x_i\ , Exact_i\ , W_i\ , err_i\ )$

STOP. (*The procedure was successful.*)

7.4. Pade Approximation (3,3) Method Algorithm

The following algorithm implements the Pade Approximation (3,3) Method for solving linear third order two-point boundary value problem using Matlab software.

Algorithm 7.4

To approximate the solution of the following third order boundary value problem:

$y''' + p(x)y'' + q(x)y' + r(x)y = s(x)\ , a \le x \le b,$

$$y(a) = \ \alpha, y'(a) = \beta, y(b) = \gamma$$

Inputs:

Endpoints a and b – boundary conditions α, β and γ – number of subintervals N.

Outputs:

$x_i\ , W_i$: Approximated value, $Exact_i\ and\ Error_i$ for each $i = 1\ ,2\ , \dots .\ , N + 1$

Step 1:

$h = \frac{b-a}{N}$

Step 2:

For $i = 1, 2, \ldots, N+1$ do step 3 and 4

Step 3:

Set

$x = a + (i-1) * h$

Step 4:

Set

$$a_{i,1,1} = 1 + \frac{h}{2}p_i + \frac{h^2}{10}\left(-p_i' + p_i^2 - q_i\right) - \frac{h^3}{120}(-p_i^3 + 2p_iq_i - r_i - p_i'' + 3p_i'p_i - 2q_i')$$

$$a_{i,1,2} = \frac{h}{2}q_i + \frac{h^2}{10}(-q_i' + p_iq_i - r_i) - \frac{h^3}{120}(-p_i^2q_i + p_ir_i + q_i^2 - q_i'' + 2p_i'q_i - 2r_i' + p_iq_i')$$

$$a_{i,1,3} = \frac{h}{2}r_i + \frac{h^2}{10}(-r_i' + p_ir_i) - \frac{h^3}{120}(-p_i^2r_i + q_ir_i - r_i'' + 2p_i'r_i + p_ir_i')$$

$$a_{i,2,1} = -\frac{h}{2} - \frac{h^2}{10}p_i - \frac{h^3}{120}({p_i}^2 - q_i - p_i')$$

$$a_{i,2,2} = 1 - \frac{h^2}{10}q_i - \frac{h^3}{120}(p_iq_i - r_i - q_i')$$

$$a_{i,2,3} = -\frac{h^2}{10}r_i - \frac{h^3}{120}(p_ir_i - r_i')$$

$$a_{i,3,1} = \frac{h^2}{10} + \frac{h^3}{120}p_i$$

$$a_{i,3,2} = -\frac{h}{2} + \frac{h^3}{120}q_i$$

$$a_{i,3,3} = 1 + \frac{h^3}{120}r_i$$

$$b_{i,1,1} = -1 + \frac{h}{2}p_i - \frac{h^2}{10}\left(-p_i' + p_i^2 - q_i\right) - \frac{h^3}{120}(-p_i^3 + 2p_iq_i - r_i - p_i'' + 3p_i'p_i - 2q_i')$$

$$b_{i,1,2} = \frac{h}{2}q_i - \frac{h^2}{10}(-q_i' + p_iq_i - r_i) - \frac{h^3}{120}(-p_i^2q_i + p_ir_i + q_i^2 - q_i'' + 2p_i'q_i - 2r_i' + p_iq_i')$$

$$b_{i,1,3} = \frac{h}{2}r_i - \frac{h^2}{10}(-r_i' + p_ir_i) - \frac{h^3}{120}(-p_i^2r_i + q_ir_i - r_i'' + 2p_i'r_i + p_ir_i')$$

$$b_{i,2,1} = -\frac{h}{2} + \frac{h^2}{10}p_i - \frac{h^3}{120}({p_i}^2 - q_i - p_i')$$

$$b_{i,2,2} = -1 + \frac{h^2}{10}q_i - \frac{h^3}{120}(p_iq_i - r_i - q_i')$$

$$b_{i,2,3} = \frac{h^2}{10}r_i - \frac{h^3}{120}(p_ir_i - r_i')$$

$$b_{i,3,1} = -\frac{h^2}{10} + \frac{h^3}{120}p_i$$

$$b_{i,3,2} = -\frac{h}{2} + \frac{h^3}{120}q_i$$

$$b_{i,3,3} = -1 + \frac{h^3}{120}r_i$$

Step 5:

For $i = 1, 2, \dots, N$ do step 6

Step 6:

Set

$$c_{i,1} = \frac{h}{2}(s_i + s_{i+1}) + \frac{h^2}{10}(s_i' - p_is_i - s_{i+1}' + p_{i+1}s_{i+1}) + \frac{h^3}{120}(-p_is_i' + \left(p_i^2 - q_i - 2p_i'\right)s_i + s_i'' - p_{i+1}s_{i+1}' + \left(p_{i+1}^2 - q_{i+1} - 2p_{i+1}'\right)s_{i+1} + s_{i+1}'')$$

$$c_{i,2} = \frac{h^2}{10}(s_i - s_{i+1}) + \frac{h^3}{120}(s_i' - p_is_i + s_{i+1}' - p_{i+1}s_{i+1})$$

$$c_{i,3} = \frac{h^3}{120}(s_i + s_{i+1})$$

Step 7:

Define the five diagonals a, b, c, d, e, f, k, l for the matrix A and the vector column C.

Step 8:

For $i = 1, 2, \dots, N$ set

$a_{(3i-2)} = a_{i+1,1,1}$

$a_{(3i-1)} = a_{i+1,2,2}$

$a_{3i} = a_{i+1,3,3}$

$b_{(3i-2)} = a_{i+1,1,2}$

$b_{(3i-1)} = a_{i+1,2,3}$

$c_{(3i-2)} = a_{i+1,2,1}$

$c_{(3i-1)} = a_{i+1,3,2}$

$d_{(3i-2)} = a_{i+1,1,3}$

$e_{(3i-2)} = a_{i+1,3,1}$

Step 9:

For $i = 1, 2, \dots, N-1$ set

$c_{3i} = b_{i+1,1,3}$

$e_{(3i-1)} = b_{i+1,1,2}$

$e_{3i} = b_{i+1,2,3}$

$f_{(3i-2)} = b_{i+1,1,1}$

$f_{(3i-1)} = b_{i+1,2,2}$

$f_{3i} = b_{i+1,3,3}$

$k_{(3i-2)} = b_{i+1,2,1}$

$k_{(3i-1)} = b_{i+1,3,2}$

$l_{(3i-2)} = b_{i+1,3,1}$

Step 10:

For $i = 1, 2, \dots, N-2$ set

$C_{3i+1} = c_{i+1,1}$

$C_{3i+2} = c_{i+1,2}$

$C_{3i+3} = c_{i+1,3}$

Step 11:

Set

$A_{1,3N} = b_{1,1,1}$

$A_{2,3N} = b_{1,2,1}$

$A_{3,3N} = b_{1,3,1}$

$A_{3N-2,3N} = 0$

$A_{3N-1,3N} = 0$

$A_{3N,3N} = 0$

$C_1 = c_{1,1} - b_{1,1,2}\beta - b_{1,1,3}\alpha$

$C_2 = c_{1,2} - b_{1,2,2}\beta - b_{1,2,3}\alpha$

$C_3 = c_{1,3} - b_{1,3,2}\beta - b_{1,3,3}\alpha$

$C_{3N-2} = c_{N,1} - a_{N,1,3}\gamma$

$C_{3N-1} = c_{N,2} - a_{N,2,3}\gamma$

$C_{3N} = c_{N,3} - a_{N,3,3}\gamma$

Step 12:

$[l, u] = lu(A)$

$t = inv(l) * C$

$y = inv(u) * t$

Step 13:

Set

$W_1 = \alpha$

$W_{N+1} = \gamma$

Step 14:

For $i = 1, 2, \dots, N-1$ set

$W_{i+1} = y_{3i}$

Step 15:

$For\ i = 1, 2, \dots, N+1$ Set

$x_i = a + (i-1) * h$

$err_i = abs(Exact_i - W_i)$

$Output(\ x_i\ , Exact_i\ , W_i\ , err_i\ )$

STOP. (*The procedure was successful.*)

7.5. Rational Chebyshev Approximation Method Algorithm

The following algorithm implements the Rational Chebyshev Approximation Method for solving linear third order two-point boundary value problem using the Matlab software.

Algorithm 7.5

To approximate the solution of the following third order boundary value problem:

$y''' + p(x)y'' + q(x)y' + r(x)y = s(x)\ , a \le x \le b,$

$$y(a) = \alpha, y'(a) = \beta, y(b) = \gamma$$

Inputs:

Endpoints a and b – boundary conditions α,β and γ – number of subintervals N, a_i and b_i for $i = 0, 1, 2, 3$.

Outputs:

x_i , W_i: Approximated value,Exact_i and Error_i for each $i = 1, 2, \ldots, N + 1$

Step 1:

$$h = \frac{b - a}{N}$$

Step 2:

For $i = 1, 2, \ldots, N + 1$ do step 3 and 4

Step 3:

Set

$$x = a + (i - 1) * h$$

Step 4:

Set

$$a_{i,1,1} = b_0 - b_1 h p_i + b_2 h^2 \left(p_i^2 - q_i - p_i'\right) + b_3 h^3 (-p_i^3 + 2p_i q_i - r_i - p_i'' + 3p_i' p_i - 2q_i')$$

$$a_{i,1,2} = -b_1 h q_i + b_2 h^2 (p_i q_i - r_i - q_i') + b_3 h^3 \left(-p_i^2 q_i + p_i r_i + q_i^2 - q_i'' + 2p_i' q_i - 2r_i' + p_i q_i'\right)$$

$$a_{i,1,3} = -b_1 h r_i + b_2 h^2 (p_i r_i - r_i') + b_3 h^3 i$$

$$a_{i,2,1} = b_1 h - b_2 h^2 p_i + b_3 h^3 \left(p_i^2 - q_i - p_i'\right)$$

$$a_{i,2,2} = b_0 - b_2h^2q_i + b_3h^3(-q_i' + p_iq_i - r_i)$$

$$a_{i,2,3} = -b_2h^2r_i + b_3h^3(-r_i' + p_ir_i)$$

$$a_{i,3,1} = b_2h^2 - b_3h^3p_i$$

$$a_{i,3,2} = b_1h - b_3h^3q_i$$

$$a_{i,3,3} = b_0 - b_3h^3r_i$$

$$b_{i,1,1} = -a_0 + a_1hp_i - a_2h^2\left(p_i^2 - q_i - p_i'\right) - a_3h^3(-p_i^3 + 2p_iq_i - r_i - p_i'' + 3p_i'p_i - 2q_i')$$

$$b_{i,1,2} = a_1hq_i - a_2h^2(p_iq_i - r_i - q_i') - a_3h^3\left(-p_i^2q_i + p_ir_i + q_i^2 - q_i'' + 2p_i'q_i - 2r_i' + p_iq_i'\right)$$

$$b_{i,1,3} = a_1hr_i - a_2h^2(p_ir_i - r_i') - a_3h^3\left(-p_i^2r_i + q_ir_i - r_i'' + 2p_i'r_i + p_ir_i'\right)$$

$$b_{i,2,1} = -a_1h + a_2h^2p_i - a_3h^3\left(p_i^2 - q_i - p_i'\right)$$

$$b_{i,2,2} = -a_0 + a_2h^2q_i - a_3h^3(-q_i' + p_iq_i - r_i)$$

$$b_{i,2,3} = a_2h^2r_i - a_3h^3(-r_i' + p_ir_i)$$

$$b_{i,3,1} = -a_2h^2 + a_3h^3p_i$$

$$b_{i,3,2} = -a_1h + a_3h^3q_i$$

$$b_{i,3,3} = -a_0 + a_3h^3r_i$$

Step 5:

For $i = 1, 2, \dots, N$ do step 6

Step 6:

Set

$$c_{i,1} = h[-b_1s_{i+1} + a_1s_i] + h^2[-b_2(-p_{i+1}s_{i+1} + s_{i+1}') + a_2(-p_is_i + s_i')] + h^3\left[-b_3\left(\left(p_{i+1}^2 - q_{i+1} - 2p_{i+1}'\right)s_{i+1} - p_{i+1}s_{i+1}' + s_{i+1}''\right) + a_3\left(\left(p_i^2 - q_i - 2p_i'\right)s_i - p_is_i' + s_i''\right)\right]$$

$$c_{i,2} = h^2[-b_2 s_{i+1} + a_2 s_i]$$
$$+ h^3[-b_3(-p_{i+1}s_{i+1} + s'_{i+1}) + a_3(-p_i s_i + s'_i)]$$
$$c_{i,3} = h^3[-b_3 s_{i+1} + a_3 s_i]$$

Step 7:

Define the five diagonals a, b, c, d, e, f, k, l for the matrix A and the vector column C.

Step 8:

For $i = 1, 2, \dots, N$ set

$a_{(3i-2)} = a_{i+1,1,1}$

$a_{(3i-1)} = a_{i+1,2,2}$

$a_{3i} = a_{i+1,3,3}$

$b_{(3i-2)} = a_{i+1,1,2}$

$b_{(3i-1)} = a_{i+1,2,3}$

$c_{(3i-2)} = a_{i+1,2,1}$

$c_{(3i-1)} = a_{i+1,3,2}$

$d_{(3i-2)} = a_{i+1,1,3}$

$e_{(3i-2)} = a_{i+1,3,1}$

Step 9:

For $i = 1, 2, \dots, N-1$ set

$c_{3i} = b_{i+1,1,3}$

$e_{(3i-1)} = b_{i+1,1,2}$

$e_{3i} = b_{i+1,2,3}$

$f_{(3i-2)} = b_{i+1,1,1}$

$f_{(3i-1)} = b_{i+1,2,2}$

$f_{3i} = b_{i+1,3,3}$

$k_{(3i-2)} = b_{i+1,2,1}$

$k_{(3i-1)} = b_{i+1,3,2}$

$l_{(3i-2)} = b_{i+1,3,1}$

Step 10:

For $i = 1, 2, \dots, N-2$ set

$C_{3i+1} = c_{i+1,1}$

$C_{3i+2} = c_{i+1,2}$

$C_{3i+3} = c_{i+1,3}$

Step 11:

Set

$A_{1,3N} = b_{1,1,1}$

$A_{2,3N} = b_{1,2,1}$

$A_{3,3N} = b_{1,3,1}$

$A_{3N-2,3N} = 0$

$A_{3N-1,3N} = 0$

$A_{3N,3N} = 0$

$C_1 = c_{1,1} - b_{1,1,2}\beta - b_{1,1,3}\alpha$

$C_2 = c_{1,2} - b_{1,2,2}\beta - b_{1,2,3}\alpha$

$C_3 = c_{1,3} - b_{1,3,2}\beta - b_{1,3,3}\alpha$

$C_{3N-2} = c_{N,1} - a_{N,1,3}\gamma$

$C_{3N-1} = c_{N,2} - a_{N,2,3}\gamma$

$C_{3N} = c_{N,3} - a_{N,3,3}\gamma$

Step 12:

$[l, u] = lu(A)$

$t = inv(l) * C$

$y = inv(u) * t$

Step 13:

Set

$W_1 = \alpha$

$W_{N+1} = \gamma$

Step 14:

For $i = 1, 2, \dots, N-1$ set

$W_{i+1} = y_{3i}$

Step 15:

$For\ i = 1, 2, \dots, N+1$ Set

$x_i = a + (i-1) * h$

$err_i = abs(Exact_i - W_i)$

$Output(\ x_i\ , Exact_i\ , W_i\ , err_i\ )$

STOP. (The procedure was successful.)

Quartic B-Spline Method Algorithm

The following algorithm implements the Quartic B-Spline Method for solving linear third order two-point boundary value problem using the Matlab software.

Algorithm 7.6

To approximate the solution of the following third order boundary value problem:

$y''' = p(x)y'' + q(x)y' + r(x)y + s(x)\,, a \le x \le b,$

$$y(a) = \alpha, y'(a) = \beta, y(b) = \gamma$$

Inputs:

Endpoints a and b – boundary conditions α, β and γ – number of subintervals N.

Outputs:

$x_i\,, W_i$: The approximated value at the point x_i.

x_i, Wd_i: $Approximated\ first\ derivative$ at x_i,$Exact_i$ and $Error_i$ for each $i = 1\,,2\,, \ldots\,, N+1$

Step 1:

Set

$h = \frac{b-a}{N}$

Step 2:

For $i = 1, 2, \ldots, N+1$ do step 3 and 4

Step 3:

Set

$x = a + (i - 1) * h$

Step 4:

Set

$a_i = -1 - \frac{h}{2} p_i + \frac{h^2}{6} q_i - \frac{h^3}{24} r_i$

$b_i = 3 + \frac{h}{2} p_i + \frac{h^2}{2} q_i - \frac{11h^3}{24} r_i$

$d_i = -3 + \frac{h}{2} p_i - \frac{h^2}{2} q_i - \frac{11h^3}{24} r_i$

$e_i = 1 - \frac{h}{2} p_i - \frac{h^2}{6} q_i - \frac{h^3}{24} r_i$

$k_i = h^3 s_i$

Step 5:

Define the matrix M of size $(N + 4) * (N + 4)$ and the vector matrix F of size $N + 4$.

Step 6:

Set

$A_{1,1} = \frac{1}{24}, A_{1,2} = \frac{11}{24}, A_{1,3} = \frac{11}{24}, A_{1,4} = \frac{1}{24}$

$A_{2,1} = \frac{-1}{6h}, A_{2,2} = \frac{-1}{2h}, A_{2,3} = \frac{1}{2h}, A_{2,4} = \frac{1}{6h}$

$A_{n+4,n+1} = \frac{1}{24}, A_{n+4,n+2} = \frac{11}{24}, A_{2+4,n+3} = \frac{11}{24}, A_{n+4,n+4} = \frac{1}{24}$

$F_1 = \alpha, F_2 = \beta, F_{n+4} = \gamma$

Step 7:

For $i = 1, 2, \dots, N + 1$ set

$F_{i+2} = k_i$

Step 8:

For $i = 1, 2, \ldots, N + 3$ set

For $i = 1, 2, \ldots, N + 3$, then check

If $i = j$ and $i \geq 3$ set

$$A_{i,j} = d_{i-2}$$

Else if $i - j = 2$ set

$$A_{i,j} = a_{i-2}$$

Else if $i - j = 1$ or $(i \neq 2 \,\&\, j \neq 1)$ set

$$A_{i,j} = b_{i-2}$$

Else if $i - j = -1$ and $i \geq 3$ set

$$A_{i,j} = e_{i-2}$$

End if clause

End for loop

End for loop

Step 9:

Ser $A_{n+3,n+4} = e_{N+1}$

Step 10:

Initialize Gauss Seidel method, define $D, L\ and\ U$ matrices all of size $(N + 4) \times (N + 4)$.

Step 11:

For $i = 1, 2, \ldots, N + 4$ set

for $j = 1, 2, \ldots, N + 4$ check

If $i = j$ set

$$D_{i,j} = A_{i,j}$$

Else if $i > j$ set

$$L_{i,j} = -1 * A_{i,j}$$

Else if $i > j$ set

$$U_{i,j} = -1 * A_{i,j}$$

End if clause

End for loop

End for loop

Step 12:

Set

$T_{gs} = inv(D - L) * U$

$C_{gs} = inv(D - L) * F$

Step 13:

For $k = 1, 2, \dots, 20$ set

$c = T_{gs} * c + C_{gs}$

Step 14:

$For\ i = 1, 2, \dots, N + 1$ Set

$x_i = a + (i - 1) * h$

$$W_i = \frac{1}{24}(c_i + 11c_{i+1} + 11c_{i+2} + c_{i+3})Wd_i$$

$$= \frac{1}{6h}(-c_i - 3c_{i+1} + 3c_{i+2} + c_{i+2})$$

$err_i = abs(Exact_i - W_i)$

$Output(\ x_i\ , Exact_i\ , W_i\ , err_i\ )$

Step 15:

STOP. (*The procedure was successful.*)

7.6. Numerical Examples and Results

To Test the efficiency and effectiveness of the numerical methods that have been developed and studied in previous chapters, we will test the following examples:

7.7.1 Example 1

Consider the following third order linear boundary value problem:

$y'''(x) - 2x^2y'' + 3xy' + 5x^2y = e^{2x}(3x^3 - x^2 - 5x - 4)$, on the interval $0 \leq x \leq 1$

With the following boundary conditions: $y(0) = 1$, $y'(0) = 1$, $y(1) = 0$.

The exact solution is $y(x) = e^{2x}(1 - x)$ [3], the following tables represent the results that have been obtained after solving example 1 using the previous methods.

Linear Shooting Method for solving example 1

Using Linear Shooting method in algorithm 7.1 for solving example 1, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 1: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	1.0	1.0	0.0
0.1	1.099262482344153	1.099265828452194	$3.3 * 10^{-6}$
0.2	1.193459758113016	1.193466184474031	$6.4 * 10^{-6}$
0.3	1.275483160273356	1.275492310188409	$9.1 * 10^{-6}$
0.4	1.335324557095481	1.335335953325634	$1.1 * 10^{-5}$
0.5	1.359140914229523	1.359153919449132	$1.3 * 10^{-5}$
0.6	1.328046769094619	1.328060530165123	$1.3 * 10^{-5}$
0.7	1.216559990053402	1.216573356253661	$1.3 * 10^{-5}$
0.8	0.990606484879023	0.990617884322122	$1.1 * 10^{-5}$
0.9	0.604964746441295	0.604971994973641	$7.2 * 10^{-6}$
1.0	0.0	0.0	0.0

Maximum Error = $1.3 * 10^{-5}$

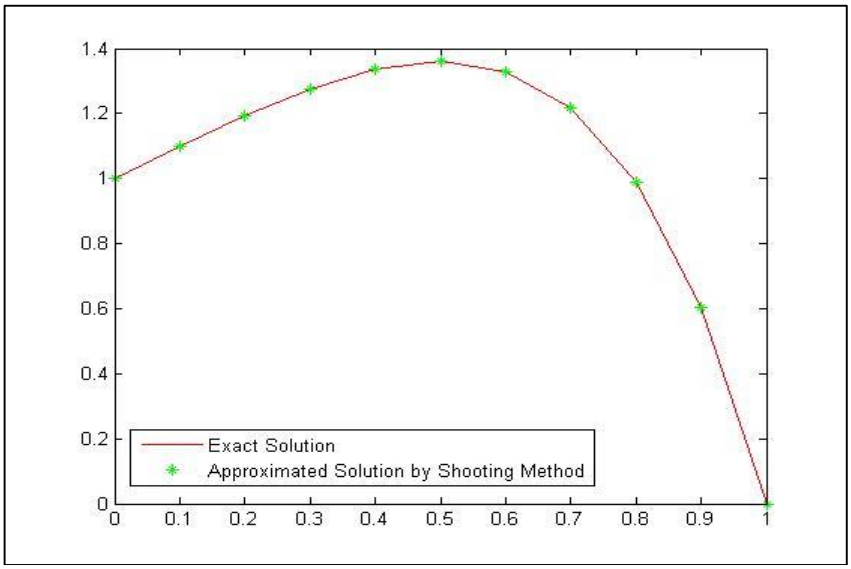

Figure7. 1: The exact and the approximated solutions for example 1 using Linear Shooting Method

Finite Difference Method for solving example 1

Using Finite Difference method in algorithm 7.2 for solving example 1, the following table represents the numerical and the exact results for $N = 10$ and for $N = 20$:

Table 7. 2: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$Err_i, h = 0.1$	$Err_i, h = 0.05$
0.0	1.0	0.0	0.0
0.1	1.099262482344153	$2.5 * 10^{-4}$	$6.4 * 10^{-5}$
0.2	1.193459758113016	$3.5 * 10^{-4}$	$9.0 * 10^{-5}$
0.3	1.275483160273356	$3.1 * 10^{-4}$	$8.0 * 10^{-5}$
0.4	1.335324557095481	$1.3 * 10^{-4}$	$3.5 * 10^{-5}$
0.5	1.359140914229523	$1.5 * 10^{-4}$	$3.8 * 10^{-5}$
0.6	1.328046769094619	$6.3 * 10^{-4}$	$1.3 * 10^{-4}$
0.7	1.216559990053402	$9.0 * 10^{-4}$	$2.2 * 10^{-4}$
0.8	0.990606484879023	$1.1 * 10^{-3}$	$2.8 * 10^{-4}$
0.9	0.604964746441295	$9.6 * 10^{-4}$	$2.4 * 10^{-4}$
1.0	0.0	0.0	0.0

The Maximum Error for $N = 10$ and $N = 20$ equal $1.3 * 10^{-3}$, $2.8 * 10^{-4}$ respectively.

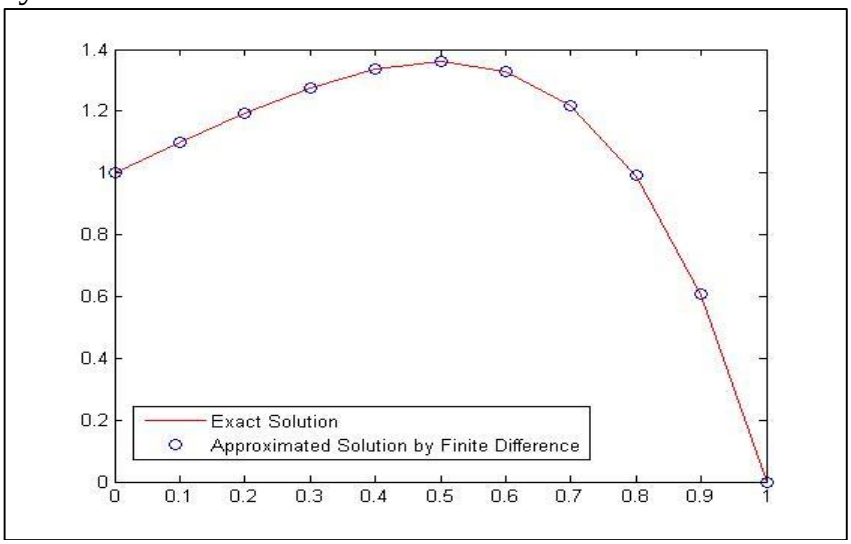

Figure7. 2: The exact and the approximated solutions for example 1 using Finite Difference

Pade Approximation (2,2) Method for solving example 1

Using Pade Approximation (2,2) in algorithm 7.3 for solving example 1, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 3: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	1.0	1.0	0.0
0.1	1.099262482344153	1.099262637328588	$1.5 * 10^{-7}$
0.2	1.193459758113016	1.193459146746167	$6.1 * 10^{-7}$
0.3	1.275483160273356	1.275481055696274	$2.1 * 10^{-6}$
0.4	1.335324557095481	1.335320481926064	$4.07 * 10^{-6}$
0.5	1.359140914229523	1.359134711390886	$6.2 * 10^{-6}$
0.6	1.328046769094619	1.328038692822045	$8.07 * 10^{-6}$
0.7	1.216559990053402	1.216550822459394	$9.16 * 10^{-6}$
0.8	0.990606484879023	0.990597684447372	$8.8 * 10^{-6}$
0.9	0.604964746441295	0.604958635681406	$6.1 * 10^{-6}$
1.0	0.0	0.0	0.0

Maximum error= $9.16 * 10^{-6}$

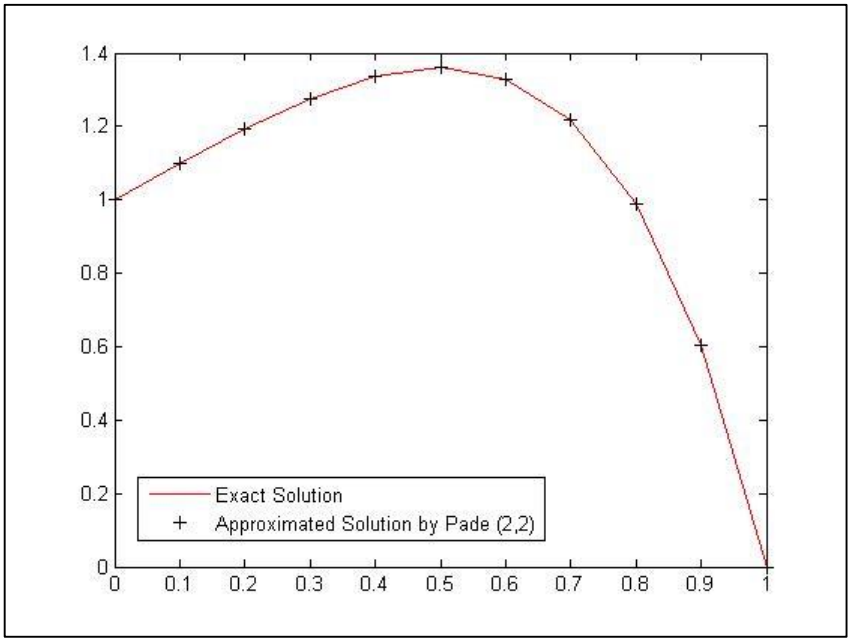

Figure7. 3: The exact and the approximated solutions for example 1 using Pade (2,2)

Pade Approximation (3,3) Method for solving example 1

Using Pade Approximation (3,3) method in algorithm 7.4 for solving example 1, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 4: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	1.0	1.0	0.0
0.1	1.099262482344153	1.099262482230163	$1.1 * 10^{-10}$
0.2	1.193459758113016	1.193459758250832	$1.3 * 10^{-10}$
0.3	1.275483160273356	1.275483160952334	$6.7 * 10^{-10}$
0.4	1.335324557095481	1.335324558507911	$1.4 * 10^{-9}$
0.5	1.359140914229523	1.359140916444255	$2.2 * 10^{-9}$
0.6	1.328046769094619	1.328046772023375	$2.9 * 10^{-9}$
0.7	1.216559990053402	1.216559993407880	$3.3 * 10^{-9}$
0.8	0.990606484879023	0.990606488116544	$3.2 * 10^{-9}$
0.9	0.604964746441295	0.604964748696508	$2.2 * 10^{-9}$
1.0	0.0	0.0	0.0

Maximum error= $3.3 * 10^{-9}$

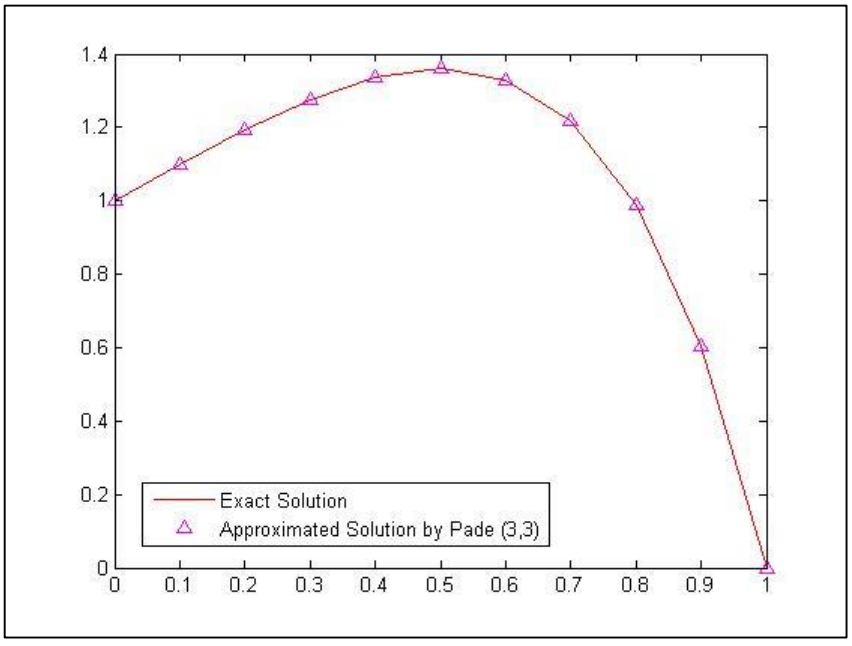

Figure7. 4: The exact and the approximated solutions for example 1 using Pade Approximation (3,3) Method

Rational Chebyshev Approximation Method for solving example 1

Using Rational Chebyshev method in algorithm 7.5 for solving example 1, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 5: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	1.0	1.0	0.0
0.1	1.099262482344153	1.099262399475198	$8.2 * 10^{-8}$
0.2	1.193459758113016	1.193459542032773	$2.1 * 10^{-7}$
0.3	1.275483160273356	1.275482772625478	$3.8 * 10^{-7}$
0.4	1.335324557095481	1.335323976618905	$5.8 * 10^{-7}$
0.5	1.359140914229523	1.359140143653942	$7.7 * 10^{-7}$
0.6	1.328046769094619	1.328045844367949	$9.2 * 10^{-7}$
0.7	1.216559990053402	1.216558992590456	$9.9 * 10^{-7}$
0.8	0.990606484879023	0.990605557684333	$9.2 * 10^{-7}$
0.9	0.604964746441295	0.604964115140274	$6.3 * 10^{-7}$
1.0	0.0	0.0	0.0

Maximum error= $9.9 * 10^{-7}$

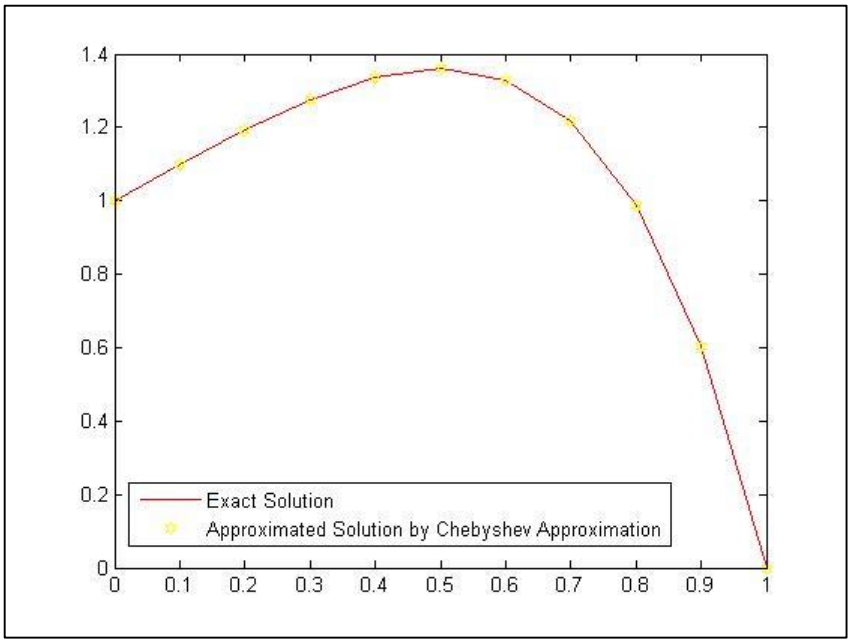

Figure7. 5: The exact and the approximated solutions for example 1 using Chebyshev Approximation Method

Quartic B-Spline Method for solving example 1

After using Quartic B-Spline method in algorithm 7.6 for solving example 1, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 6: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	1.0	1.0	0.0
0.1	1.099262482344153	1.099418824326585	$1.5 * 10^{-4}$
0.2	1.193459758113016	1.194050846836718	$5.9 * 10^{-4}$
0.3	1.275483160273356	1.276726270264298	$1.2 * 10^{-3}$
0.4	1.335324557095481	1.337355898391887	$2 * 10^{-3}$
0.5	1.359140914229523	1.361988520776878	$2.8 * 10^{-3}$
0.6	1.328046769094619	1.331593386424681	$3.5 * 10^{-3}$
0.7	1.216559990053402	1.220491960777112	$3.9 * 10^{-3}$
0.8	0.990606484879023	0.994342933228970	$3.7 * 10^{-3}$
0.9	0.604964746441295	0.607558821690503	$2.5 * 10^{-3}$
1.0	0.0	0.0	0.0

Maximum error= $3.9 * 10^{-3}$

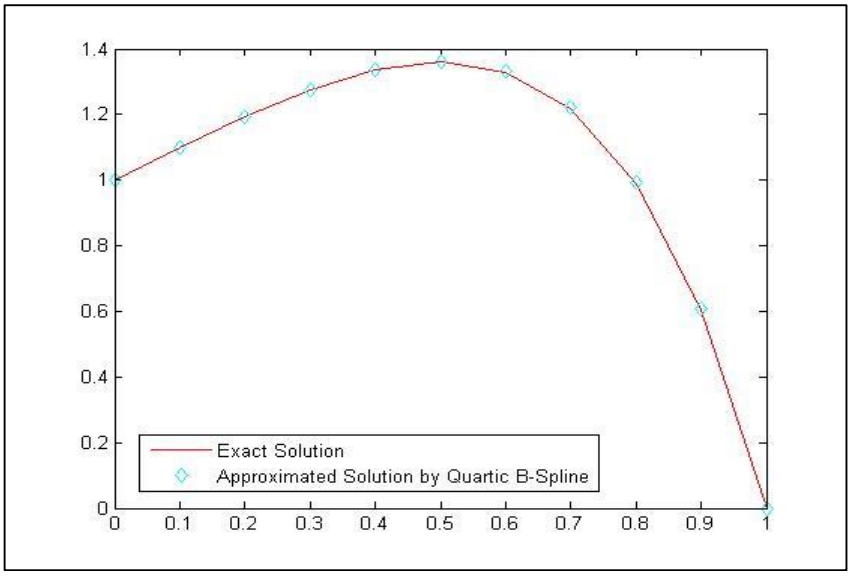

Figure7. 6: The exact and the approximated solutions for example 1 using Quartic B-Spline Method

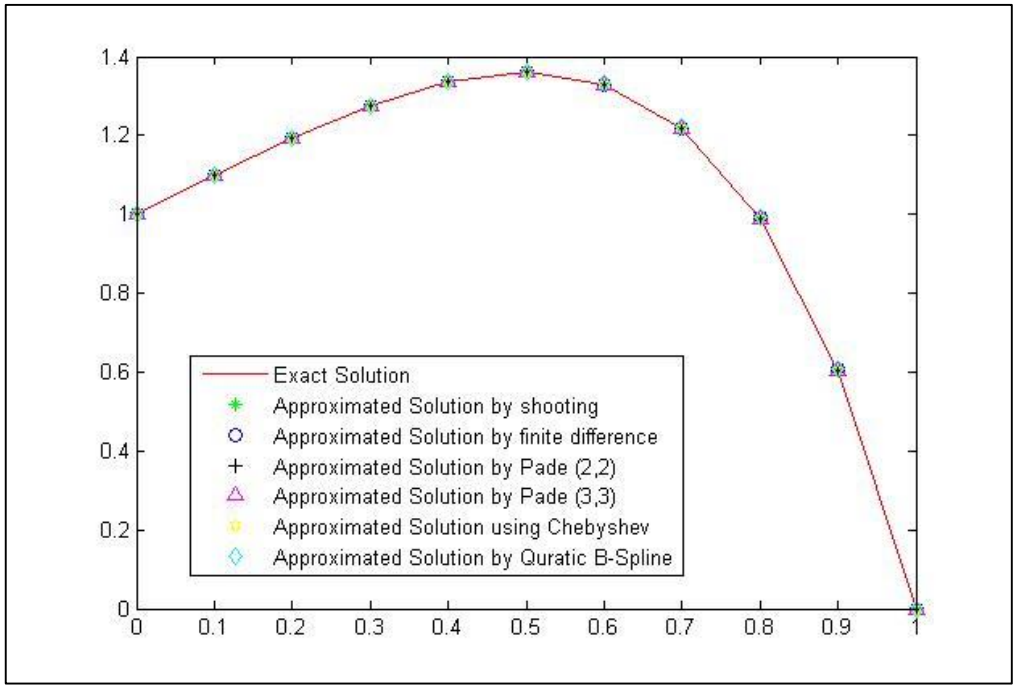

Figure7. 7: The exact and the approximated solutions for example 1 using the numerical methods

7.7.2 Example 2

Consider the following third order linear boundary value problem:

$$y'''(x) - xy(x) = e^x(x^3 - 2x^2 - 5x - 3), x \in [0,1]$$

With the following boundary conditions: $y(0) = 0\ , y'(0) = 1\ , y'(1) = -e$, where the exact solution is $y(x) = x(1-x)e^x$ [1].

Linear Shooting Method for solving example 2

Consider

$$y''' = p(x)y'' + q(x)y' + r(x)y + s(x)\ , a \le x \le b \qquad (7.7.1)$$

$$y(a) = \alpha\ , y'(a) = \beta\ , y'(b) = \gamma \qquad (7.7.2)$$

The previous linear third order boundary value problem will be turn into two initial value problem, as

$$u''' = p(x)u'' + q(x)u' + r(x)u + s(x)\ , a \leq\leq b \qquad (7.7.3)$$

$$u(a) = \alpha, u'(a) = 0, u''(a) = 0 \qquad (7.7.4)$$

$$v''' = p(x)v'' + q(x)v' + r(x)v \ , a \le x \le b \tag{7.7.5}$$

$$v(a) = 0, v'(a) = 1, v''(a) = 0 \tag{7.7.6}$$

Let $z(x)$ is the solution for equation (7.7.1) with its boundary conditions in (7.7.2), and $u(x)$ and $v(x)$ are the solution for the two initial value problem (7.7.3) and (7.7.5) with their given boundary conditions, and define that

$$z(x) = u(x) + \theta v(x) \tag{7.7.7}$$

$$z(x) = u(x) + \frac{\gamma - u'(b)}{v'(b)} v(x) \tag{7.7.8}$$

Then $z(x)$ is the solution to the third order linear boundary value problem (7.7.1). To see this, first note that

$$z'(x) = u'(x) + \frac{\gamma - u'(b)}{v'(b)} v'(x)$$

$$z''(x) = u''(x) + \frac{\gamma - u'(b)}{v'(b)} v''(x)$$

And

$$z'''(x) = u'''(x) + \frac{\gamma - u'(b)}{v'(b)} v'''(x) \tag{7.7.9}$$

Substituting equation (7.7.3) and (7.7.5) in equation (7.7.9), we get that

$$z'''(x) = [p(x)u'' + q(x)u' + r(x)u + s(x)] + \frac{\gamma - u'(b)}{v'(b)} [p(x)v'' + q(x)v' + r(x)v]$$

$$z'''(x) = p(x)\left[u'' + \frac{\gamma - u'(b)}{v'(b)} v''\right] + q(x)\left[u' + \frac{\gamma - u'(b)}{v'(b)} v'\right] + r(x)\left[u + \frac{\gamma - u'(b)}{v'(b)} v\right] + s(x)$$

$$z'''(x) = p(x)z'' + q(x)z' + r(x)z + s(x)$$

Moreover,

$$z(a) = u(a) + \frac{\gamma - u'(b)}{v'(b)} v(a), z(a) = \alpha + \frac{\gamma - u'(b)}{v'(b)} . 0 = \alpha$$

The same

$$z'(b) = u'(b) + \frac{\gamma - u'(b)}{v'(b)} v'(b), z'(b) = u'(b) + \gamma - u'(b) = \gamma$$

So equation (7.7.8) is a solution for the third order linear boundary value problem (7.7.1) and boundary conditions (7.7.2).

After using Linear Shooting method in algorithm 7.1 for solving example 2, the following table gives the numerical and the exact results for $N = 10$:

Table 7. 7: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	0.0	0.0	0.0
0.1	0.099465382626808	0.099464021880689	$1.3 * 10^{-7}$
0.2	0.195424441305627	0.195421705670486	$2.7 * 10^{-7}$
0.3	0.283470349590961	0.283466232021989	$4.1 * 10^{-7}$
0.4	0.358037927433905	0.358032432697738	$5.4 * 10^{-6}$
0.5	0.412180317675032	0.412173468394630	$6.8 * 10^{-6}$
0.6	0.437308512093722	0.437300356493690	$8.1 * 10^{-6}$
0.7	0.422888068568800	0.422878690368993	$9.3 * 10^{-6}$
0.8	0.356086548558795	0.356076079559584	$1.0 * 10^{-5}$
0.9	0.221364280004125	0.221352916022185	$1.1 * 10^{-5}$
1.0	0.0	-0.000011979046890	$1.19 * 10^{-5}$

Maximum error= $1.19 * 10^{-5}$

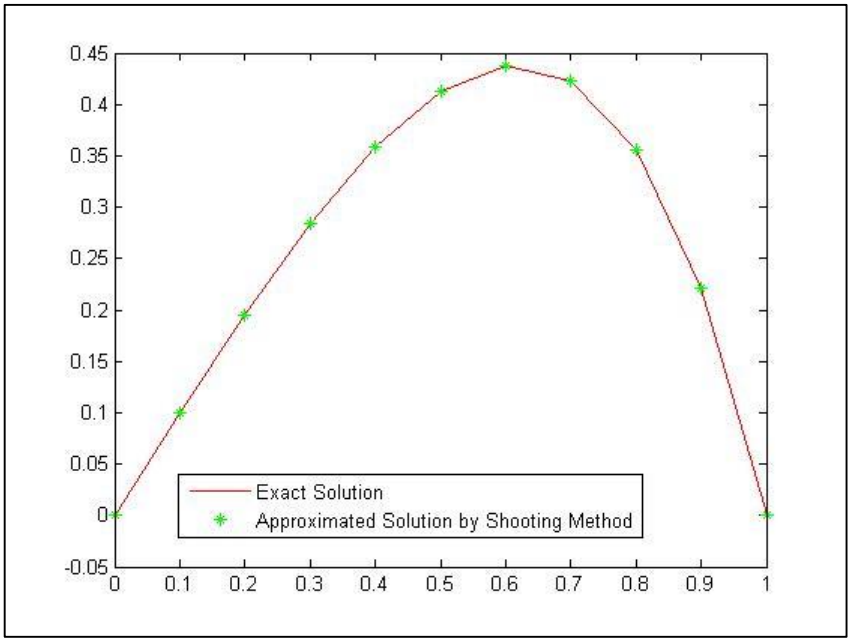

Figure7. 8: The exact and the approximated solutions for example 2 using Linear Shooting Method

Finite Difference Method for solving example 2

Before using the algorithm 7.2 we must modified it according to the given boundary conditions in example 2, so we get that

$$w_0 = \alpha, z_0 = \beta, z_N = \gamma$$

$$Am = k$$

where

$$A=\begin{bmatrix} \begin{bmatrix} 1 & \frac{h}{2} \\ 0 & c_0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & a_0 \end{bmatrix} & \cdots & & & \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 1 & \frac{-h}{2} \\ d_1 & b_1 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & a_1 \end{bmatrix} & & & \\ \vdots & & \ddots & & \vdots & \\ & & & \begin{bmatrix} -1 & \frac{-h}{2} \\ 0 & c_{N-1} \end{bmatrix} & \begin{bmatrix} 1 & \frac{-h}{2} \\ d_{N-1} & b_{N-1} \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ & & \cdots & & \begin{bmatrix} -1 & \frac{-h}{2} \\ 0 & c_N \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ d_N & a_N \end{bmatrix} \end{bmatrix}$$

and $m = \begin{bmatrix} \begin{bmatrix} y_{-1} \\ z_{-1} \end{bmatrix} \\ \begin{bmatrix} y_1 \\ z_1 \end{bmatrix} \\ \begin{bmatrix} y_2 \\ z_2 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} y_{N-1} \\ z_{N-1} \end{bmatrix} \\ \begin{bmatrix} y_N \\ z_{N+1} \end{bmatrix} \end{bmatrix}$ $and\ k = \begin{bmatrix} \begin{bmatrix} r_\circ \\ s_\circ \end{bmatrix} \\ \begin{bmatrix} r_1 \\ s_1 \end{bmatrix} \\ \begin{bmatrix} r_2 \\ s_2 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} r_{N-1} \\ s_{N-1} \end{bmatrix} \\ \begin{bmatrix} r_N \\ s_N \end{bmatrix} \end{bmatrix}$

As $r_\circ = \alpha - \frac{h}{2}\beta, r_1 = \alpha + \frac{h}{2}\beta, r_N = \frac{h}{2}\gamma$, but $r_i = 0$ for all $i = 2,3, \dots, N - 1$

And $s_\circ = e_\circ - b_\circ\beta - d_\circ\alpha, s_1 = e_1 - c_1\beta,\ s_{N-1} = e_{N-1} - a_{N-1}\gamma\ , s_N = e_N - b_N\gamma,$ but $s_i = e_i$ for all $i = 2,3, \dots, N - 2$

After using Finite Difference method in algorithm 7.2 for solving example 2, the following table represents the numerical and the exact results for $N = 10$ and $N = 20$:

Table 7. 8: the exact and approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$Err_i, h = 0.1$	$Err_i, h = 0.05$
0.0	0.0	0.0	0.0
0.1	0.099465382626808	$3.3 * 10^{-4}$	$8.3 * 10^{-5}$
0.2	0.195424441305627	$8.4 * 10^{-4}$	$2.1 * 10^{-4}$
0.3	0.283470349590961	$1.5 * 10^{-3}$	$3.8 * 10^{-4}$
0.4	0.358037927433905	$2.3 * 10^{-3}$	$5.9 * 10^{-4}$
0.5	0.412180317675032	$3.3 * 10^{-3}$	$8.5 * 10^{-4}$
0.6	0.437308512093722	$4.5 * 10^{-3}$	$1.1 * 10^{-3}$
0.7	0.422888068568800	$5.9 * 10^{-3}$	$1.4 * 10^{-3}$
0.8	0.356086548558795	$7.5 * 10^{-3}$	$1.8 * 10^{-3}$
0.9	0.221364280004125	$9.2 * 10^{-3}$	$2.2 * 10^{-3}$
1.0	0.0	0.0112	$2.8 * 10^{-3}$

Maximum error for $N = 10$ and $N = 20$ equal 0.0112 and $2.8 * 10^{-3}$ respectively.

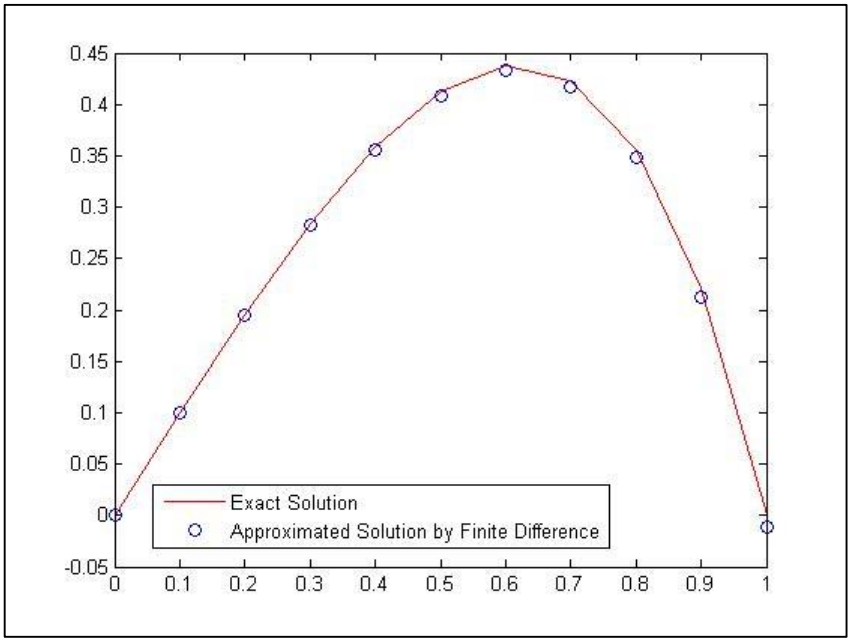

Figure7. 9: The exact and the approximated solutions for example 2 using Finite Difference Method

Pade Approximation (2,2) Method for solving example 2

For the given boundary conditions in example 2, the algorithm 7.3 will be updated as

$$B_0 = \begin{bmatrix} 0 & b_{0,1,1} & 0 \\ 0 & b_{0,2,1} & 0 \\ 0 & b_{0,3,1} & 0 \end{bmatrix} \; and \; A_{N+1} = \begin{bmatrix} a_{N,1,1} & 0 & a_{N,1,3} \\ a_{N,2,1} & 0 & a_{N,2,3} \\ a_{N,3,1} & 0 & a_{N,3,3} \end{bmatrix}$$

And

$$\mathrm{k}_4 = c_{N,1} - a_{N,1,2}u_{1,N}$$

$$\mathrm{k}_5 = c_{N,2} - a_{N,2,2}u_{1,N}$$

$$\mathrm{k}_6 = c_{N,3} - a_{N,3,2}u_{1,N}$$

After using Pade Approximation (2,2) method in algorithm 7.3 for solving example 2, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 9: the exact and approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	0.0	0.0	0.0
0.1	0.099465382626808	0.099465624667358	$2.4 * 10^{-7}$
0.2	0.195424441305627	0.195425003027445	$5.6 * 10^{-7}$
0.3	0.283470349590961	0.283471326653904	$9.7 * 10^{-7}$
0.4	0.358037927433905	0.358039436258280	$1.5 * 10^{-6}$
0.5	0.412180317675032	0.412182498655699	$2.1 * 10^{-6}$
0.6	0.437308512093722	0.437311533377447	$3.0 * 10^{-6}$
0.7	0.422888068568800	0.422892130502515	$4.0 * 10^{-6}$
0.8	0.356086548558795	0.356091888936601	$5.3 * 10^{-6}$
0.9	0.221364280004125	0.221371180264582	$6.9 * 10^{-6}$
1.0	0.0	0.000008415014490	$8.4 * 10^{-6}$

Maximum error $= 8.4 * 10^{-6}$

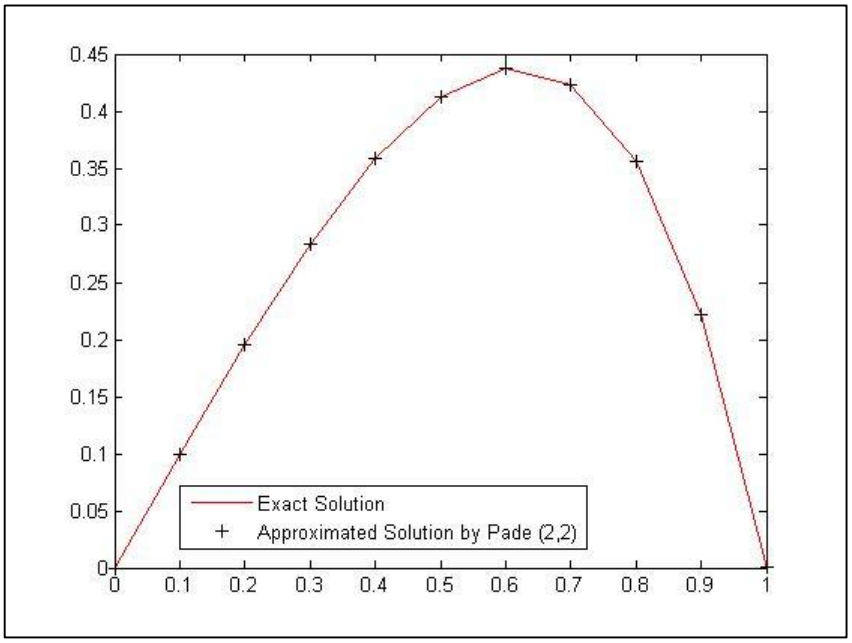

Figure7. 10: The exact and the approximated solutions for example 2 using Pade Approximation (2,2) Method

Pade Approximation (3,3) Method for solving example 2

For the given boundary conditions in example 2, the algorithm 7.4 will be updated as

$$B_0 = \begin{bmatrix} 0 & b_{0,1,1} & 0 \\ 0 & b_{0,2,1} & 0 \\ 0 & b_{0,3,1} & 0 \end{bmatrix} \; and \; A_{N+1} = \begin{bmatrix} a_{N,1,1} & 0 & a_{N,1,3} \\ a_{N,2,1} & 0 & a_{N,2,3} \\ a_{N,3,1} & 0 & a_{N,3,3} \end{bmatrix}$$

And

$$\mathrm{k}_4 = c_{N,1} - a_{N,1,2}u_{1,N}$$

$$\mathrm{k}_5 = c_{N,2} - a_{N,2,2}u_{1,N}$$

$$\mathrm{k}_6 = c_{N,3} - a_{N,3,2}u_{1,N}$$

After using Pade Approximation (3,3) method in algorithm 7.4 for solving example 2, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 10: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	0.0	0.0	0.0
0.1	0.099465382626808	0.099465437841428	$5.5 * 10^{-8}$
0.2	0.195424441305627	0.195424662234985	$2.2 * 10^{-7}$
0.3	0.283470349590961	0.283470846757753	$5.2 * 10^{-7}$
0.4	0.358037927433905	0.358038811454920	$4.9 * 10^{-7}$
0.5	0.412180317675032	0.412181699415514	$1.3 * 10^{-6}$
0.6	0.437308512093722	0.437310502937840	$1.9 * 10^{-6}$
0.7	0.422888068568800	0.422890780839658	$2.7 * 10^{-6}$
0.8	0.356086548558795	0.356090096121791	$3.5 * 10^{-6}$
0.9	0.221364280004125	0.221368779088166	$4.4 * 10^{-6}$
1.0	0.0	0.000005275790723	$5.15 * 10^{-6}$

Maximum error= $5.15 * 10^{-6}$

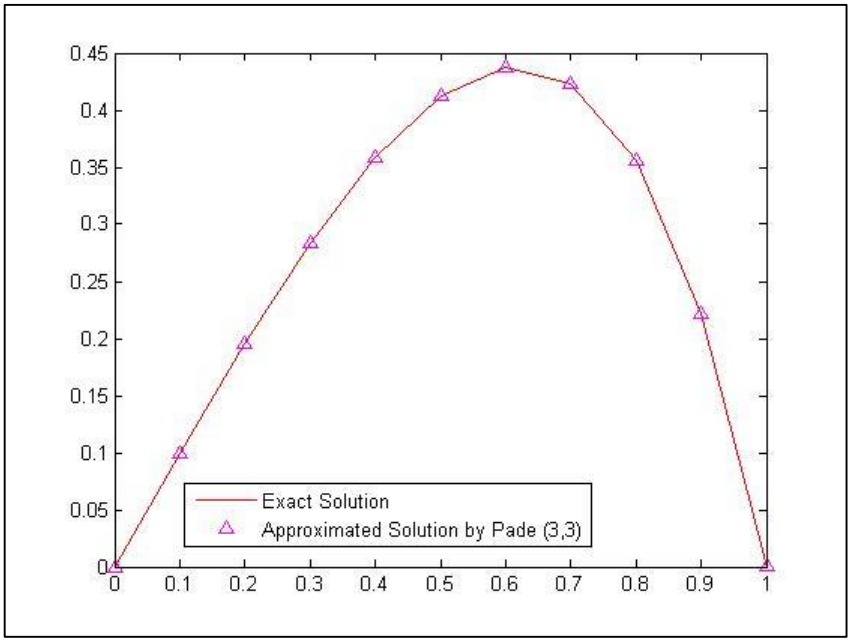

Figure7. 11: The exact and the approximated solutions for example 2 using Pade Approximation (3,3) Method

Rational Chebyshev Approximation Method for solving example 2

For the given boundary conditions in example 2, the algorithm 7.5 will be updated as

$$B_0 = \begin{bmatrix} 0 & b_{0,1,1} & 0 \\ 0 & b_{0,2,1} & 0 \\ 0 & b_{0,3,1} & 0 \end{bmatrix} \; and \; A_{N+1} = \begin{bmatrix} a_{N,1,1} & 0 & a_{N,1,3} \\ a_{N,2,1} & 0 & a_{N,2,3} \\ a_{N,3,1} & 0 & a_{N,3,3} \end{bmatrix}$$

And

$$\mathrm{k}_4 = c_{N,1} - a_{N,1,2}u_{1,N}$$

$$\mathrm{k}_5 = c_{N,2} - a_{N,2,2}u_{1,N}$$

$$\mathrm{k}_6 = c_{N,3} - a_{N,3,2}u_{1,N}$$

After using Rational Chebyshev Approximation method in algorithm 7.5 for solving example 2, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 11: the exact and approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	0.0	0.0	0.0
0.1	0.099465382626808	0.099465290939128	$9.1 * 10^{-8}$
0.2	0.195424441305627	0.195424326064729	$1.1 * 10^{-7}$
0.3	0.283470349590961	0.283470290095864	$5.9 * 10^{-8}$
0.4	0.358037927433905	0.358038016475043	$8.9 * 10^{-8}$
0.5	0.412180317675032	0.412180664286605	$3.4 * 10^{-7}$
0.6	0.437308512093722	0.437309244823395	$7.3 * 10^{-7}$
0.7	0.422888068568800	0.422889339372624	$1.2 * 10^{-6}$
0.8	0.356086548558795	0.356088537448285	$1.9 * 10^{-6}$
0.9	0.221364280004125	0.221367200595080	$2.9 * 10^{-6}$
1.0	0.0	0.000003814283018	$3.8 * 10^{-6}$

Maximum error= $3.8 * 10^{-6}$

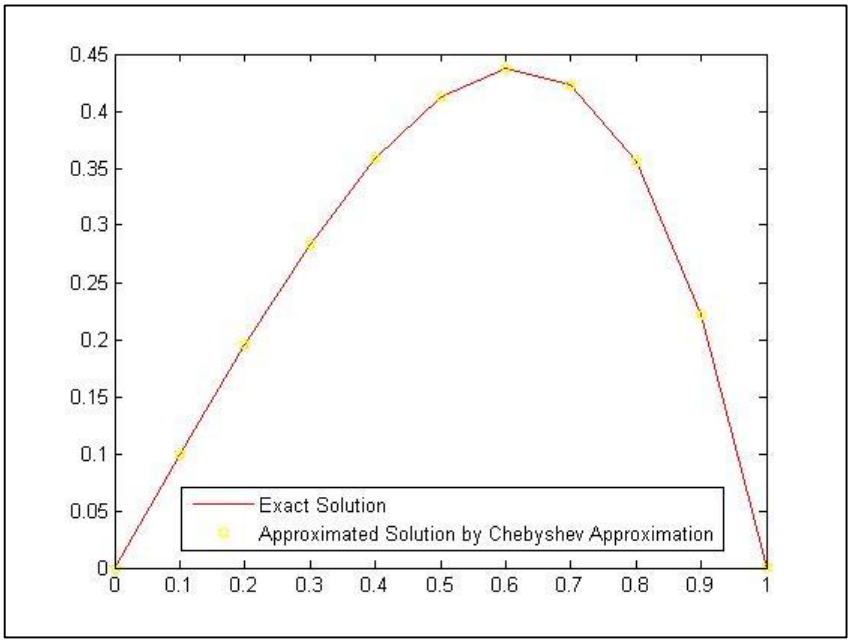

Figure7. 12: The exact and the approximated solutions for example 2 using Rational Chebyshev Approximation Method

Quartic B-Spline Method for solving example 2

For the given boundary conditions in example 2, we get that

$y(a) = \alpha$ will be approximated as $S(x_0) = \alpha$, so for $i = 0$

$$\frac{1}{24}(c_{-4} + 11c_{-3} + 11c_{-2} + c_{-1}) = \alpha \qquad (7.7.10)$$

$y'(a) = \beta$ will be approximated as $S'(x_0) = \beta$, so for $i = 0$

$$\frac{1}{6h}(-c_{-4} - 3c_{-3} + 3c_{-2} + c_{-1}) = \beta \qquad (7.7.11)$$

$y'(b) = \gamma$ will be approximated as $S'(x_n) = \gamma$, so for $i = n$, we get that

$$\frac{1}{6h}(-c_{n-4} - 3c_{n-3} + 3c_{n-2} + c_{n-1}) = \gamma \qquad (7.7.12)$$

Then,

$$a_i c_{i-4} + b_i c_{i-3} + d_i c_{i-2} + e_i c_{i-1} = k_i, i = 0,1, \dots, n \qquad (7.7.13)$$

Equations (7.7.10), (7.7.11), (7.7.12) and (7.7.13) form a linear system with $n + 4$ unknowns c_i for $i = -4, -3, \dots, n-2, n-1$

$$Ac = F \qquad (7.7.14)$$

Where F is a vector matrix of size $n + 4$, where $F_0 = \alpha, F_1 = \beta, F_{n+4} = \gamma$ and

$$F_{i+3} = k_i, i = 0,1, \dots, n$$

$$A = \begin{bmatrix} \frac{1}{24} & \frac{11}{24} & \frac{11}{24} & \frac{1}{24} & 0 & \cdots & & \cdots & & \cdots & 0 \\ \frac{-1}{6h} & \frac{-1}{2h} & \frac{1}{2h} & \frac{1}{6h} & 0 & \ddots & & & & & \vdots \\ a_0 & b_0 & d_0 & e_0 & 0 & \ddots & & & & & \\ 0 & a_1 & b_1 & d_1 & e_1 & \ddots & & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & \ddots & & & \\ & & & & & & & \ddots & & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & & & & & & & \ddots & \vdots \\ & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & & & 0 & a_n & b_n & d_n & e_n \\ 0 & \cdots & & \cdots & & \cdots & 0 & \frac{-1}{6h} & \frac{-1}{2h} & \frac{1}{2h} & \frac{1}{6h} \end{bmatrix}, F = \begin{bmatrix} \alpha \\ \beta \\ k_0 \\ k_1 \\ \vdots \\ \vdots \\ \\ \vdots \\ \vdots \\ k_n \\ \gamma \end{bmatrix}$$

This forms a linear system on $N + 4$ linear equations with $N + 4$ unknown coefficients.

After using Quartic B-Spline method in algorithm 7.6 for solving example

7.2, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 12: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	0.0	0.0	0.0
0.1	0.099465382626808	0.099518676503010	$5.3 * 10^{-5}$
0.2	0.195424441305627	0.195627794661920	$2.0 * 10^{-4}$
0.3	0.283470349590961	0.283904646003533	$4.3 * 10^{-4}$
0.4	0.358037927433905	0.358765551436395	$7.2 * 10^{-4}$
0.5	0.412180317675032	0.413242188079152	$1.0 * 10^{-3}$
0.6	0.437308512093722	0.438720718504239	$1.4 * 10^{-3}$
0.7	0.422888068568800	0.424638045977124	$1.7 * 10^{-3}$
0.8	0.356086548558795	0.358128703960093	$2.0 * 10^{-3}$
0.9	0.221364280004125	0.223614960619726	$2.2 * 10^{-3}$
1.0	0.0	0.002331664349832	$2.3 * 10^{-3}$

Maximum error= $2.3 * 10^{-3}$

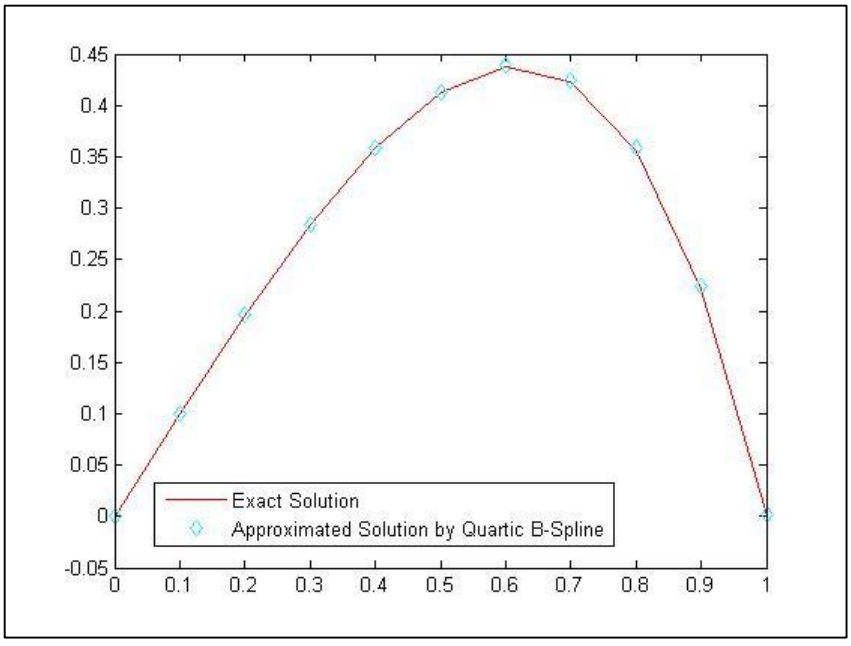

Figure7. 13: The exact and the approximated solutions for example 2 using Quartic B-Spline Method

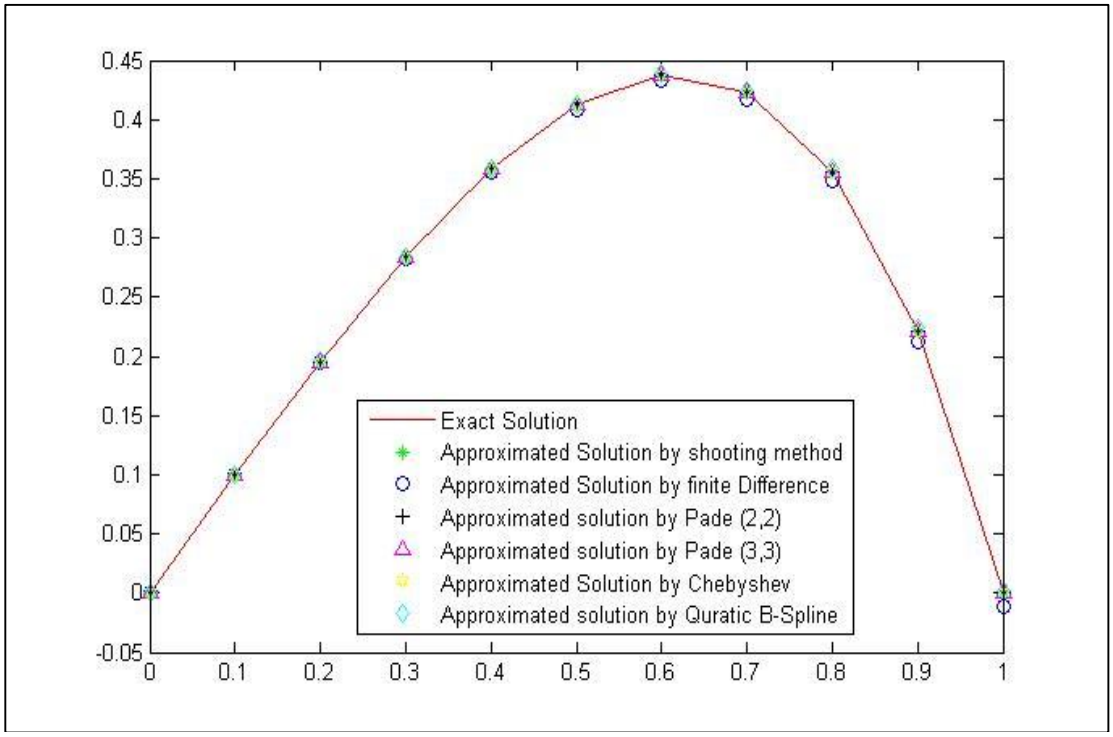

Figure7. 14: The exact and the approximated solutions for example 2 using all numerical methods

7.7.3 Example 3

Consider the following third order linear boundary value problem:

$$y'''(x) + y(x) = (x-4)\sin x + (1-x)\cos x\,, x \in [0,1]$$

With the following boundary conditions: $y(0) = 0\,, y'(0) = -1\,, y'(1) = \sin 1,$ where the exact solution is $y(x) = x(x-1)\sin x$ [3].

Linear Shooting Method for solving example 3

After using Linear shooting method in algorithm 7.1 for solving example 3, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 13: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	0.0	0.0	0.0
0.1	-0.089850074982145	0.090522437138974	0.180372512121119
0.2	-0.158935464636049	0.181703523734594	0.340638988370643
0.3	-0.206864144662938	0.273674764096053	0.480538908758991
0.4	-0.233651005385190	0.366010665011299	0.599661670396489
0.5	-0.239712769302102	0.457754390210411	0.697467159512513
0.6	-0.225856989358014	0.547449809476706	0.773306798834720
0.7	-0.193265306171307	0.633179883141782	0.826445189313089
0.8	-0.143471218179905	0.712611225649064	0.856082443828968
0.9	-0.078332690962748	0.783044597578950	0.861377288541698
1.0	0.0	0.841470984807897	0.841470984807897

Maximum error= 0.86137

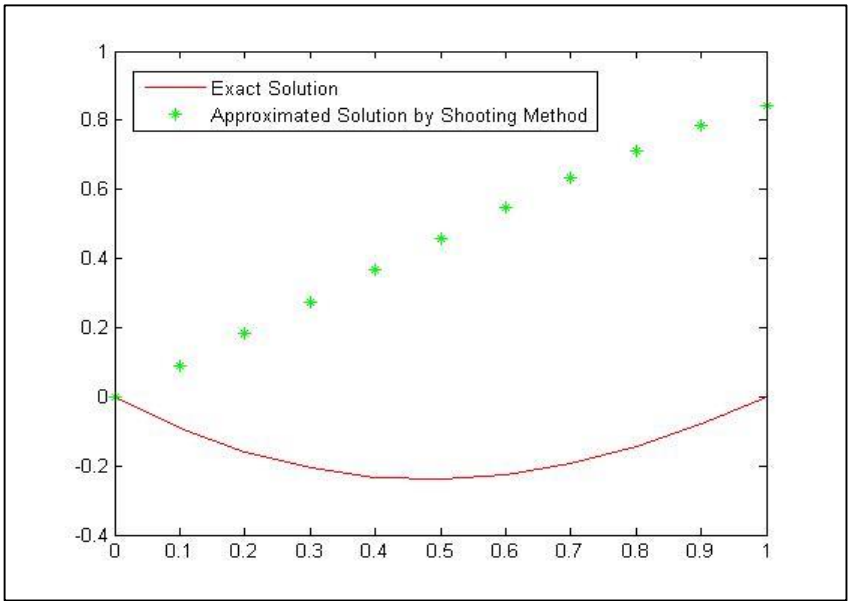

Figure7. 15: The exact and the approximated solutions for example 3 using Linear Shooting Method

Finite Difference Method for solving example 3

After using Finite difference method in algorithm 7.2 for solving example 3, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 14: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	0.0	0.0	0.0
0.1	-0.089850074982145	-0.089781284582543	$6.8 * 10^{-5}$
0.2	-0.158935464636049	-0.158827170975968	$1.08 * 10^{-4}$
0.3	-0.206864144662938	-0.206745723337955	$1.18 * 10^{-4}$
0.4	-0.233651005385190	-0.233551944929851	$9.9 * 10^{-5}$
0.5	-0.239712769302102	-0.239662666246607	$5 * 10^{-5}$
0.6	-0.225856989358014	-0.225885513752673	$2.8 * 10^{-5}$
0.7	-0.193265306171307	-0.193402137491925	$1.3 * 10^{-4}$
0.8	-0.143471218179905	-0.143745951576379	$2.74 * 10^{-4}$
0.9	-0.078332690962748	-0.078774713891856	$4.42 * 10^{-4}$
1.0	0.0	-0.000638339391634	$6.38 * 10^{-4}$

Maximum error= $6.38 * 10^{-4}$

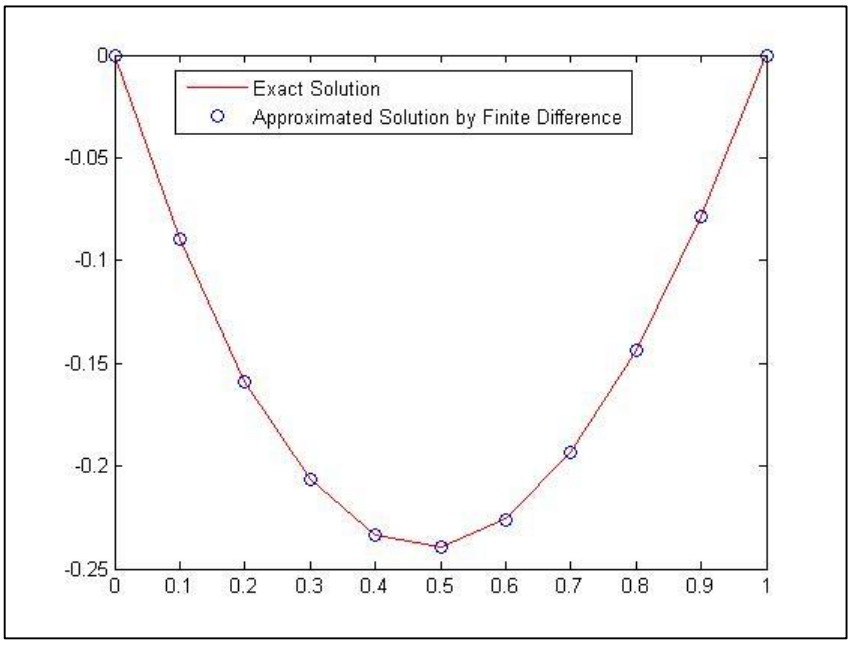

Figure7. 16: The exact and the approximated solutions for example 3 using Finite Difference Method

Pade Approximation (2,2) Method for solving example 3

After using Pade Approximation (2,2) in algorithm 7.3 for solving example 3, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 15: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	0.0	0.0	0.0
0.1	-0.089850074982145	-0.089850066525210	$8.45 * 10^{-9}$
0.2	-0.158935464636049	-0.158935458683938	$5.95 * 10^{-9}$
0.3	-0.206864144662938	-0.206864152098574	$7.43 * 10^{-9}$
0.4	-0.233651005385190	-0.233651036672231	$3.12 * 10^{-8}$
0.5	-0.239712769302102	-0.239712834140077	$6.48 * 10^{-8}$
0.6	-0.225856989358014	-0.225857096337676	$1.06 * 10^{-7}$
0.7	-0.193265306171307	-0.193265462435046	$1.56 * 10^{-7}$
0.8	-0.143471218179905	-0.143471429092252	$2.1 * 10^{-7}$
0.9	-0.078332690962748	-0.078332959796112	$2.68 * 10^{-7}$
1.0	0.0	-0.000000327640807	$3.27 * 10^{-7}$

Maximum error= $3.27 * 10^{-7}$

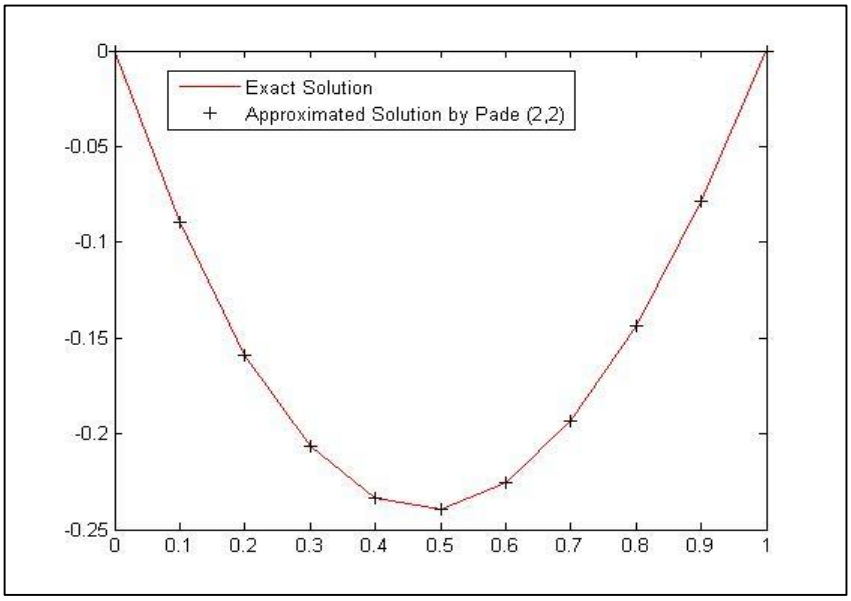

Figure7. 17: The exact and the approximated solutions for example 3 using Pade Approximation (2,2) Method

Pade Approximation (3,3) Method for solving example 3

After using Pade Approximation (3,3) in algorithm 7.4 for solving example 3, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 16: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	0.0	0.0	0.0
0.1	-0.089850074982145	-0.089850074981677	$3.69 * 10^{-13}$
0.2	-0.158935464636049	-0.158935464636163	$1.14 * 10^{-13}$
0.3	-0.206864144662938	-0.206864144664671	$1.73 * 10^{-12}$
0.4	-0.233651005385190	-0.233651005389533	$4.34 * 10^{-12}$
0.5	-0.239712769302102	-0.239712769309969	$7.86 * 10^{-12}$
0.6	-0.225856989358014	-0.225856989370212	$1.21 * 10^{-11}$
0.7	-0.193265306171307	-0.193265306188504	$1.71 * 10^{-11}$
0.8	-0.143471218179905	-0.143471218202601	$2.26 * 10^{-11}$
0.9	-0.078332690962748	-0.078332690991246	$2.84 * 10^{-11}$
1.0	0.0	-0.000000000034380	$3.43 * 10^{-11}$

Maximum error= $3.43 * 10^{-11}$

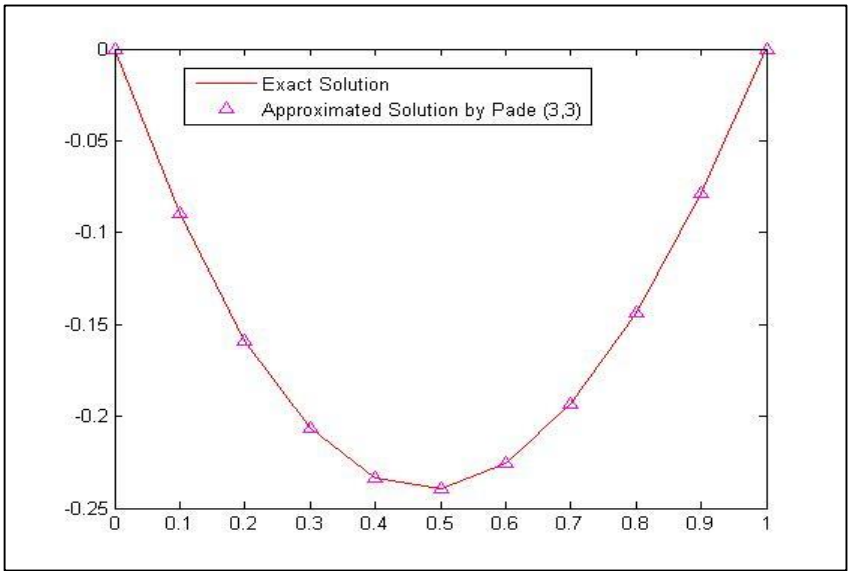

Figure7. 18: The exact and the approximated solutions for example 3 using Pade Approximation (3,3) Method

Rational Chebyshev Approximation Method for solving example 3

After using Rational Chebyshev in algorithm 7.5 for solving example 3, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 17: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	0.0	0.0	0.0
0.1	-0.089850074982145	-0.089850024655716	$5.03 * 10^{-8}$
0.2	-0.158935464636049	-0.158935416397238	$4.82 * 10^{-8}$
0.3	-0.206864144662938	-0.206864145816339	$1.15 * 10^{-9}$
0.4	-0.233651005385190	-0.233651096722768	$9.13 * 10^{-8}$
0.5	-0.239712769302102	-0.239712983787528	$2.14 * 10^{-7}$
0.6	-0.225856989358014	-0.225857350922651	$3.61 * 10^{-7}$
0.7	-0.193265306171307	-0.193265828645045	$5.22 * 10^{-7}$
0.8	-0.143471218179905	-0.143471904378819	$6.68 * 10^{-7}$
0.9	-0.078332690962748	-0.078333531954009	$8.4 * 10^{-7}$
1.0	0.0	-0.000000974562619	$9.74 * 10^{-7}$

Maximum error= $9.74 * 10^{-7}$

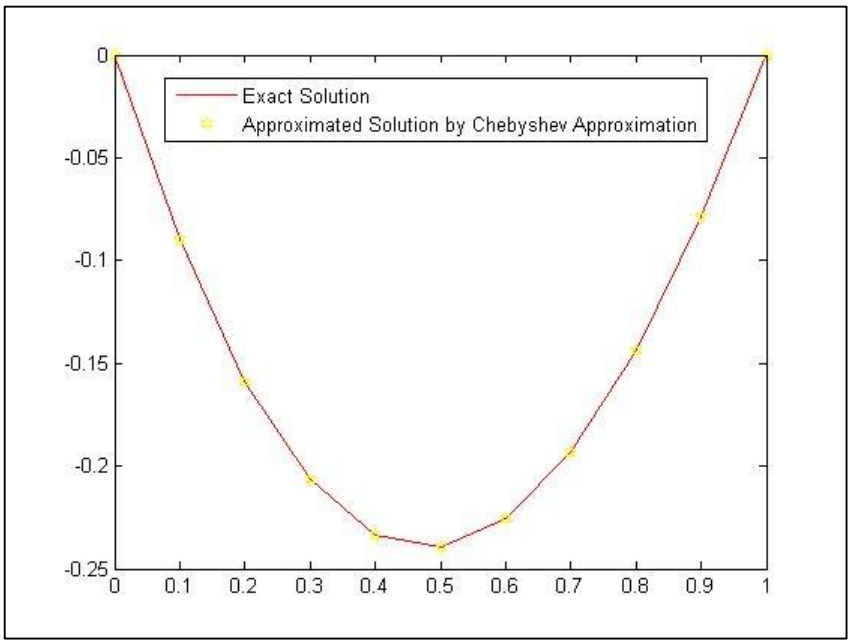

Figure7. 19: The exact and the approximated solutions for example 3 using Rational Chebyshev Approximation Method

Quartic B-Spline Method for solving example 3

After using Quartic B-Spline in algorithm 7.6 for solving example 3, the following table represents the numerical and the exact results for $N = 10$:

Table 7. 18: the exact and the approximated solutions for x_i where $i = 0, 1, \dots, 10$

x_i	$Exact_i$	$W1_i$	Err_i
0.0	0.0	0.0	0.0
0.1	-0.089850074982145	-0.089852249326635	$2.17 * 10^{-6}$
0.2	-0.158935464636049	-0.158944492680312	$9.02 * 10^{-6}$
0.3	-0.206864144662938	-0.206884777759470	$2.06 * 10^{-5}$
0.4	-0.233651005385190	-0.233687555723246	$3.65 * 10^{-5}$
0.5	-0.239712769302102	-0.239768602603939	$5.58 * 10^{-5}$
0.6	-0.225856989358014	-0.225934028642328	$7.7 * 10^{-5}$
0.7	-0.193265306171307	-0.193363554090052	$9.82 * 10^{-5}$
0.8	-0.143471218179905	-0.143588305637233	$1.17 * 10^{-4}$
0.9	-0.078332690962748	-0.078463459830076	$1.3 * 10^{-4}$
1.0	0.0	-0.000136127745933	$1.36 * 10^{-4}$

Maximum error= $1.36 * 10^{-4}$

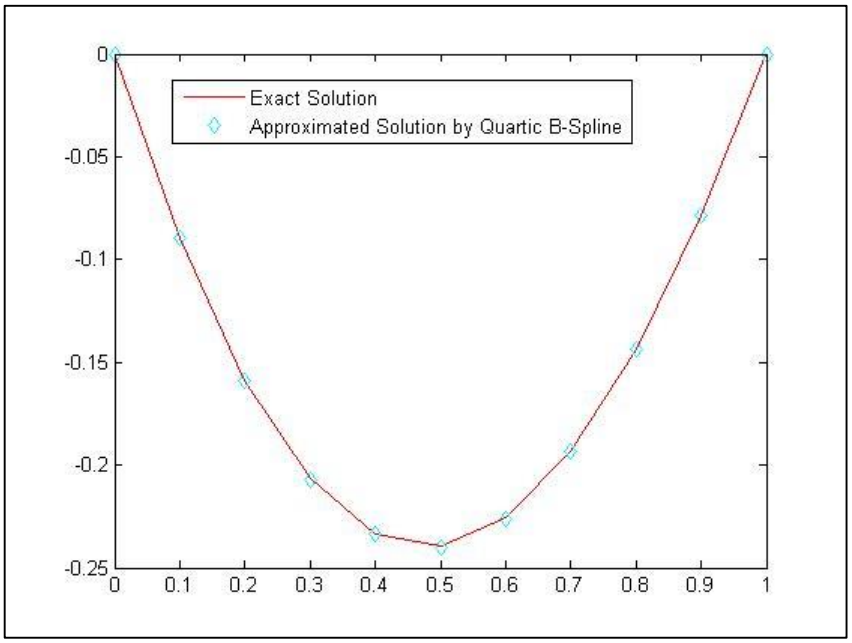

Figure7. 20: The exact and the approximated solutions for example 3 using Quartic B-Spline Method

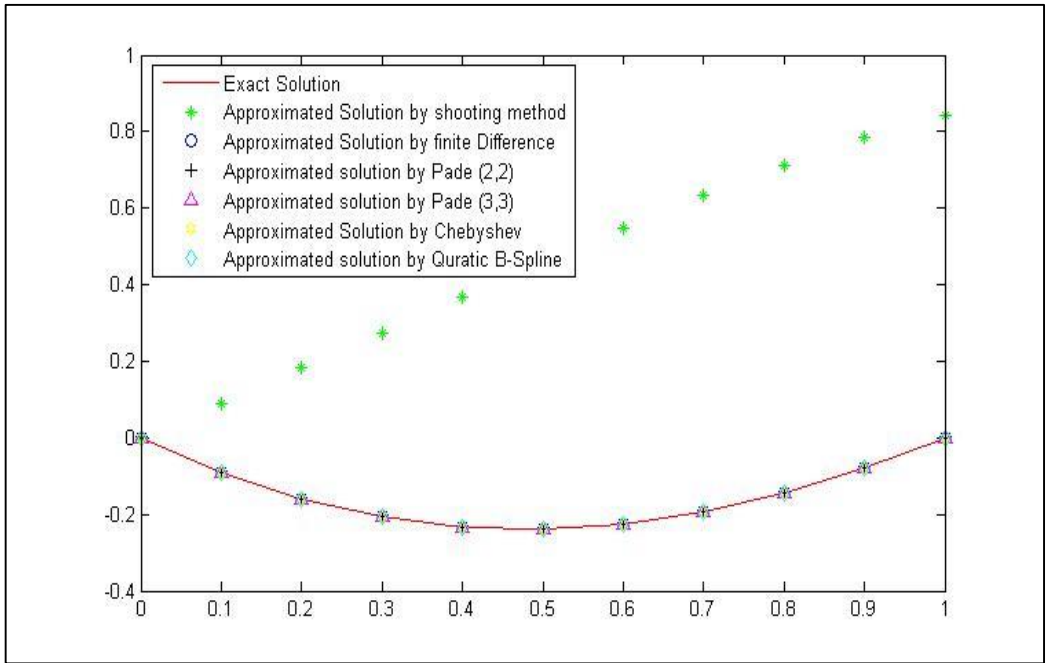

Figure7. 21: The exact and the approximated solutions for example 3 using all numerical methods

Conclusion

The numerical results that have been obtained from testing the numerical methods that have been studied and developed through this work show the following conclusions:

For example 1, we have applied the following numerical methods: Linear Shooting Method, Finite Difference Method, Pade Approximation (2,2) Method, Pade Approximation (3,3) Method, Rational Chebyshev Approximation Method and Quartic B-Spline Method and obtained the following results:

Numerical Methods	Maximum Error	CPU Time
Linear Shooting Method	$1.3 * 10^{-5}$	0.133600 seconds
Finite Difference Method	$1.0 * 10^{-3}$	0.109797 seconds
Pade Approximation (2,2) Method	$9.16 * 10^{-6}$	0.159932 seconds
Pade Approximation (3,3) Method	$3.3 * 10^{-9}$	0.256540 seconds
Rational Chebyshev Approximation Method	$9.9 * 10^{-7}$	0.243652 seconds
Quartic B-Spline Method	$3.9 * 10^{-3}$	0.112895 seconds

From the above table we can see that Pade Approximation (3,3) Method is the most efficient method for solving example 1.

We have used the numerical methods that have been developed in our work for solving example 2 and get the following results:

Numerical Methods	Maximum Error	CPU Time
Linear Shooting Method	$1.19 * 10^{-5}$	0.102304 seconds
Finite Difference Method	$1.12 * 10^{-2}$	0.329263 seconds
Pade Approximation (2,2) Method	$8.4 * 10^{-6}$	0.133821 seconds
Pade Approximation (3,3) Method	$5.15 * 10^{-6}$	0.126747 seconds
Rational Chebyshev Approximation Method	$3.8 * 10^{-6}$	0.254738 seconds
Quartic B-Spline Method	$2.3 * 10^{-3}$	0.124506 seconds

From the above table we can see that Rational Chebyshev Approximation Method is the most efficient method for solving example 2.

We have used the numerical methods that have been developed in our work for solving example 3 and get the following results:

Numerical Methods	Maximum Error	CPU Time
Linear Shooting Method	0.86137	0.148834 seconds
Finite Difference Method	$6.38 * 10^{-4}$	0.129586 seconds
Pade Approximation (2,2) Method	$3.27 * 10^{-7}$	0.103488 seconds
Pade Approximation (3,3) Method	$3.43 * 10^{-11}$	0.105732 seconds
Rational Chebyshev Approximation Method	$9.74 * 10^{-7}$	0.098478 seconds
Quartic B-Spline Method	$1.36 * 10^{-4}$	0.168454 seconds

From the above table we can see that Pade Approximation (3,3) Method is the most efficient method for solving example 3.

References

[1] F.A. Abd El-Salam, A.A. El-Sabbagh and Z.A. ZAki, ***The Numerical Solution of Linear Third Order Boundary Value Problems using Nonpolynomial Spline Technique***, Department of Engineering Mathematics and Physics, Faculty of Engineering, Benha University, Cairo, Egypt, **Journal of American Science**, 2010.

[2] Nur Nadiah Abd Hamid, Ahmad Abd. Majid and Ahmad Izani Md. Ismail, ***Quartic B-spline Interpolation Method for Linear two-point Boundary Value Problem***, School of Mathematical Sciences, University Sains Malaysia, Malaysia, **World Applied Sciences Journal** 17, 2012.

[3] A. S. Abdullah, Z. A. Majid, and N. Senu, **Solving third order boundary value problem with fifth order block method**, Mathematical Methods in Engineering and Economics. 2013.

[4] E.A. Al-Said, M.A. Noor, **Cubic splines method for a system of third-order boundary value problems**, Elsevier, 2003.

[5] Max K. Agoston, **Computer Graphics and Geometric Modeling,** Springer, 1986, (404-409).

[6] Ghazala Akram and Imran Talib, ***Quartic Non-polynomial Spline Solution of a Third Order Singularly Perturbed Boundary Value Problem***, Department of Mathematics, University of the Punjab, Pakistan, Research **Journal of Applied Sciences**, Engineering and Technology, 2014.

[7] Uri M. Ascher, Robert M. M. Mattheij, Robert D. Russell, **Numerical Solution of Boundary Value Problems for Ordinary Differential Equations,** Society for Industrial and Applied Mathematics, 1987, (36-37).

[8] Jana Proch´azkov´ and David Procházka, **Implementation of NURBS Curve Derivatives in Engineering Practice**, Brno University of Technology, Mendel University in Brno, Czech Republic, 2007.

[9] Richard H. Bartels, John C. Beatty and Brian A. Barsky, **An Introduction to the Use of Splines in Computer Graphics**, Department of Computer Science, University of Waterloo, Waterloo, Canada, Morgan Kaufmann, 1995.

[10] Carl R de Boor, **A Practical Guide to Spline**, University of Wisconsin–Madison, New york, Springer, 1978.

[11] C. Brezinski, **History of Continued Fractions and Pad´e Approximants**, Springer-Verlag, Berlin, 1991, (79-81).

[12] Richard L. Burden, J. Douglas Faires, **Numerical Analysis (9th edition)**, Brooks Cole, 2010.

[13] David Eberly, **Derivative Approximation by Finite Differences,** Geometric Tools, LLC, 2015, (1-8).

[14] Talaat S. El-Danaf, ***Quartic Nonpolynomial Spline Solutions for Third Order Two-Point Boundary Value Problem***, **International Journal of Mathematical,** Computational, Physical, Electrical and Computer Engineering, World Academy of Science, Engineering and Technology, 2008.

[15] Laure Gouba, ***The importance of Mathematics in everyday life***, 6 Melrose Road, Muizenberg 7945, South Africa, **African Institute for Mathematical Sciences**, 2008.

[16] Y. Gupta and P. K. Srivastava, **A Computational Method for Solving Two Point Boundary Value Problems of Order Four**, Department of Mathematics, Jaypee Institute of Information Technolog, India, al, Int. J. Comp. Tech. Appl, 2011.

[17] Siraj-ul-Islam, Ikram **A. Tirmizi and Muhammad Azam Khan, Quartic non-polynomial spline approach to the solution of a system of third-order boundary –value problems**, University of Engineering, Pakistan, Elsevier, 2007.

[18] Eric Lengyel, **Mathematics for 3D Game Programming and Computer Graphics (3rd edition)**, Cengage Learning PTR, 2012, (335-33).

[19] Josef Kallrath, **On Rational Function Techniques and Pad´e Approximants**, Ludwigshafen, Germany, BASF – AG, 2012.

[20] G. B. Loghmani and M. Ahmadinia, ***Numerical Solution of Third-order Boundary Value Problems***, The Islamic Republic of Iran, Shiraz University, **Iranian Journal of Science & Technology** , 2006.

[21] T.Y.Na, **Computational Methods in Engineering Boundary Value Problems**, University of Michigan-Dearborn, Dearborn, Michigan, Academic Press, 1980, (119-121).

[22] Les Piegl and Wayne Tiller, **The NURBS Book (2nd edition)**, Department of Computer Science and Engineering, University of South Florida, Springer, 2013, (50-58).

[23] William H. P., Saul A. T., William T. V. and Brian P. F., **Numerical Recipes in C: The Art of Scientific Computing (2nd Edition)**, Cambridge University Press, 1992, (204-206).

[24] E. B. Saff, **Introduction to Pade Approximants**, Vanderbilt University, Center for Constructive Approximation, 1986, (14-16).

[25] J. Rashidinia, Sh. Shari, ***B-Spline Method for Two-Point Boundary* Value Problems**, Department of Mathematics and statistics, Central Tehran Branch, Islamic Azad University, Tehran, Iran, **International Journal of Mathematical Modelling & Computations**, 2015.

[26] William F. Trench, **Elementary Differential Equations with Boundary Value Problems**, Department of Mathematics, Trinity University, USA, Brooks Cole, 2001.

[27] Lloyd N. Trefethen, **Rational Chebyshev Approximation on the Unit Disk**, Computer Science Department, Stanford University, Stanford, California 94305, USA, Springer-Verlag , 1981.

[28] E.H. Twizell and S.A. Matar, ***Numerical methods for computing the eigenvalues of linear fourth-order boundary-value problems***, Department of Mathematics and Statistics, Brunel University, United Kingdom, **Journal of Computational and Applied Mathematics** , 1991.

[29] Xiaolei Zhang, ***Modified Cubic B-Spline Solution of Singular Two-point Boundary Value Problems,*** Department of Mathematics, China, **Journal of Information & Computational Science**, 2014.

[30] J. Zinn-Justin, **Springer Tracts in Modern Physics (Course on Pade Approximants)**, Springer, 1970, (249)

[31] D. G. ZILL, **A first Course in Differential Equations with Modeling Applications (10th ed.)**, Loyola Marymount University, Brooks Cole, 2012.

جامعة النجاح الوطنية
كلية الدراسات العليا

طرق عددية لحل مسائل القيم الحدية ذات البعدين من الدرجة الثالثة

إعداد
سجى جمال ابراهيم أبو شنب

إشراف
د. سمير مطر

قدمت هذه الأطروحة استكمالا لمتطلبات الحصول على درجة الماجستير في الرياضيات المحوسبة بكلية الدراسات العليا في جامعة النجاح الوطنية، نابلس – فلسطين.
2017

طرق عددية لحل مسائل القيم الحدية ذات البعدين من الدرجة الثالثة

إعداد

سجى جمال ابراهيم أبو شنب

إشراف

د. سمير مطر

الملخص

نظرا لأهمية المسائل التفاضلية الحدية ذات البعدين ومداها الواسع في مجال الدراسات العلمية والهندسية، نشرت الكثير من الدراسات والبحوث وطورت العديد من الطرق العددية لحلها ولكن هذه الدراسات قامت بالتركيز على المسائل التفاضلية الحدية ذات البعدين من الدرجة الثانية، لذلك نحن نركز في هذه الأطروحة على حل المسائل الحدية ذات البعدين من الدرجة الثالثة.

لقد قمنا في هذه الأطروحة بدراسة بعض الطرق العددية كطريقة التصويب وطريقة تقريب المشتقات عن طريق الفررق المحدودة وتقريب Pade من الدرجة (2,2) ومن الدرجة (3,3) وطريقة Chebyshev للتقريب باستخدام الأعداد النسبية، بالإضافة إلى طريقة تقريب المنحنيات B−Spline من الدرجة الرابعة.

بعض الأمثلة قد تم حلها لاختبار مدى فعالية الطرق العددية التي تم دراستها وتطويرها في هذه الأطروحة لتحديد الطريقة الأفضل فعالية، وقد تم إجراء مقارنات بين النتائج التحليلية والنتائج التقريبية. وقد أظهرت بعض الطرق دقة عالية في قرب النتائج التحليلية من النتائج العددية.