Chapter 2

Lagrange Operational Matrix Methods to Lane-Emden, Riccati's and Bessel's Equations

2.1 Introduction

The mathematical modeling of physical problems often leads to initial value problems (IVPs) in ordinary differential equations (ODEs), in which the singular IVPs have great importance in the area of mathematics, engineering, and physics. Some of the important equations in this category are Lane–Emden type equations, Riccati's equations, and Bassel's equations [63–66]. Due to the regular occurrence of singular second order nonlinear ODEs, mathematicians and physicists are attracted to find their numerical solutions. This chapter is devoted to finding numerical solutions of Lane-Emden, Riccati's and Bessel equations based on ISF and OLBF. The description of these three types of equations is as follow:

2.1.1 Lane-Emden type equation (Type-I)

The Lane–Emden type equation introduced in 1870 by Lane [64] and studied in 1907 by Emden [63] is given as

$$P_2(x)y''(x) + P_1(x)y'(x) + f(x, y) = g(x), \tag{2.1}$$

This chapter in the form of paper has been published with title "Lagrange Operational Matrix Methods to Lane-Emden, Riccati's and Bessel's Equations" in **International Journal of Applied and Computational Mathematics** Vol. 5 (2019), p.79

with initial conditions

$$y(0) = \alpha_0, \quad y'(0) = \alpha_1, \tag{2.2}$$

where $P_2(x) = 1$, $P_1(x) = \frac{\alpha}{x}$, α_0, α_1 are constants, $g(x) \in C[0,1]$ and $f(x,y)$ is a nonlinear function of x and y.
From equation (2.1),

$$y''(x) + \frac{\alpha}{x}y'(x) + f(x,y) = g(x). \tag{2.3}$$

The various phenomena, including parts of stellar structure, the thermal history of a spherical cloud of gas, isothermal gas spheres and thermionic currents are described by Lane-Emden equation. Mathematically, these phenomena are described by different values of $f(x,y)$ (see [65]). For example: one of the best known forms is $f(x,y) = y^n$.
If $f(x,y) = y^n$, $\alpha = 2$, $g(x) = 0$, $\alpha_0 = 1$ and $\alpha_1 = 0$, equation (2.3) becomes,

$$y''(x) + \frac{2}{x}y'(x) + y^n(x) = 0, \quad x > 0, \tag{2.4}$$

with initial conditions

$$y(0) = 1, y'(0) = 0. \tag{2.5}$$

For $n = 0, 1, ..., 5$, the analytical solutions of above equation exist (see [65]). Another important form is

$$f(x,y) = (y^2 - C)^{\frac{3}{2}}, \quad g(x) = 0, \quad \alpha_0 = 0, \quad \alpha_1 = 0,$$

where C is a constant. This equation is known as white-dwarf equation (see [65]). It has been shown in [65] that the analytic solution of equation (2.4) with initial conditions $y(0) = \alpha_0$, $y'(0) = 0$ exists in the neighborhood of singularity $x = 0$.

This equation models the gravitational potential of white dwarf star [65].
Again, if

$$f(x,y) = e^y, \quad \alpha = 2, \quad g(x) = 0, \quad \alpha_0 = 0, \quad \alpha_1 = 0,$$

then the resulting equation reduces to gas sphere equation [65].

Various authors handled the singularity of the equation (2.3) differently. In article [67] authors removed the singularity of equation (2.3) by converting it into Volterra integral equation with the help of method of variation of parameter. In [68, 69], Legendre and Bernstein operational matrices of differentiation have been used to solve the equation (2.3) by reducing it into system of algebraic equations. In [70, 71], the authors used the Adomian decomposition method (ADM). Also, the variational approach has been applied in [72, 73] to solve the Lane-Emden equation analytically. Moreover, optimal homotopy asymptotic [74], series approach [75], homotopy perturbation method [76] and homotopy analysis method [77], have been discussed in order to solve the Lane-Emden equation.

Some numerical techniques to solve type-I problem, include the Legendre wavelet method [38], the rational Legendre pseudo spectral scheme [78] and the hybrid functions [79]. In 2010, Parand et al. [80, 81] used the Lagrangian interpolation method and Hermite functions collocation method to solve Lane-Emden equation numerically. In 2013, Dehghan [82] proposed interpolation method and Lakestani [83] proposed a method based on B-spline. Moreover, some techniques based on operational matrices have been established in [9, 10, 84] to solve Lane-Emden equation numerically.

2.1.2 Riccati's equation (Type-II)

In Type-II, consider the Riccati's equation of the form

$$y'(x) = P(x) + Q(x)y + R(x)y^2, \qquad 0 < x < 1, \tag{2.6}$$

with initial condition

$$y(0) = \alpha. \tag{2.7}$$

The Riccati differential equation has been named after the Italian nobleman Count Jacopo Francesco Riccati (1676-1754). The theory about Riccati's equation is described by W.T. Ried in the book [66]. These equations are of the non-linear type and play an important role in many fields of applied sciences. A well-known example is one-dimensional static Schrodinger equation which is exactly related to a Riccati's equation [85]. Solitary wave solution of a nonlinear PDE can be expressed as a polynomial in two elementary functions satisfying a projective Riccati equation [85]. Besides these important applications, Riccati equation also appear in stochastic realization theory, optimal control, robust stabilization, network synthesis and financial mathematics [86, 87]. In the progress of the solution of Riccati's equation, various methods have been proposed by many authors. In [88], the unconditionally stable scheme has been developed by the authors. Adomian decomposition method has been used in [89] to solve Riccati's equation. A variational iteration technique and iterated perturbation method have been proposed in [90] and [91], respectively. Also, homotopy analysis method is proposed in [92]. A scheme has been developed in [93] to solve Riccati equation analytically. Moreover, a ultraspherical wavelets spectral method has been discussed in [9] for the solution of fractional Riccati differential equations.

2.1.3 Bessel's equation (Type-III)

The Bessel differential equation is the linear second-order ODE. Due to its applications in different areas such as heat transfer, vibrations, and fluid mechanics, it plays an important role in physics. Bessel functions are very useful in solving many problems in mathematical physics like wave propagation and signal processing [94], which are the solution of the said equation. Bessel functions of first kind and second kind are denoted by $J_n(x)$ and $Y_n(x)$, respectively. Mathematician Daniel Bernoulli credited as first to use the concept of Bessel functions which are then generalized by Friedrich Bessel. A well-known method for solving the differential equation is the power series method [95].

Bessel differential equation is a second order ODE of the form :

$$x^2y''(x) + xy'(x) + (x^2 - n^2)y(x) = 0, \;\; n \geq 0. \tag{2.8}$$

Bessel functions $J_n(x)$ of first kind of order n are the solutions of Bessel's differential equation which satisfy the following recurrence relations

$$J_{n-1}(x) - J_{n+1}(x) = J_n'(x), \;\; J_{n-1}(x) + J_{n+1}(x) = \frac{2n}{x}J_n(x),$$

where $J_n(x)$ can be defined as [96]

$$J_n(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!(m+n)!} \left(\frac{x}{2}\right)^{2m+n}. \tag{2.9}$$

In this chapter, two schemes based on Lagrange polynomials are discussed to solve linear as well as non-linear initial value problems. The description of these schemes are given below.

2.2 Numerical schemes

2.2.1 Construction of basis functions for scheme-I

Suppose P_{N+1} is the Legendre polynomial of degree $(N+1)$ and let x_k for $k = 0, 1, 2, ..., N$ denote the roots of P_{N+1}. Consider the Lagrange polynomials with $(N+1)$ distinct nodes $x_0, x_1, ..., x_N$ defined as follows

$$l_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^{N} \frac{(x - x_i)}{(x_j - x_i)}, \tag{2.10}$$

that satisfies the Kronecker property $l_i(x_j) = \delta_{ij}$. The polynomials $l_i(x)$ are called Lagrange characteristic polynomials [97].

The interpolating scaling functions (ISF) are given by [98]

$$\theta_k(x) = \begin{cases} \sqrt{\frac{2}{w_k}} l_k(2x-1), & x \in [0,1], \\ 0, & otherwise, \end{cases} \tag{2.11}$$

where $w_k,\ k = 0, 1, ..., N$ are the Gauss-Legendre quadrature weights, given by

$$w_k = \frac{1}{(n+1)P'_{n+1}(x_k)P_n(x_k)},$$

and

$$\Theta(x) = [\theta_0(x), \theta_1(x), \theta_2(x), ..., \theta_N(x)]^T \tag{2.12}$$

form an orthonormal basis in $[0, 1)$. We can expand a function $g(x) \in L^2[0, 1]$ in terms of basis function ISF as

$$g(x) = \sum_{i=0}^{\infty} d_i \theta_i(x).$$

If we truncate the series upto the level N, then

$$g(x) \approx \sum_{i=0}^{N} d_i \theta_i(x),$$

where the coefficients are given by

$$d_i = \sqrt{\frac{w_i}{2}} g(\hat{x}_i), \qquad i = 0, 1, ..., N,$$

and

$$\hat{x}_i = \frac{x_i + 1}{2}.$$

Function approximation by scheme-I

Let $\Theta(x)$ be the ISF defined by (2.11), then a function $f(x) \in L^2[0,1]$ can be approximated by using $\Theta(x)$ as

$$f(x) = \sum_{i=0}^{\infty} a_i^{\theta}\theta_i(x) \approx \sum_{i=0}^{N} a_i^{\theta}\theta_i(x) = [A_\Theta]^T\Theta(x),$$

where $[A_\Theta]^T$ is $1 \times (N+1)$ vector given by

$$[A_\Theta]^T = \begin{bmatrix} \int_0^1 f(x)\theta_0(x)dx \\ \int_0^1 f(x)\theta_1(x)dx \\ \int_0^1 f(x)\theta_2(x)dx \\ \vdots \\ \int_0^1 f(x)\theta_N(x)dx \end{bmatrix}^T_{(N+1)\times 1}. \tag{2.13}$$

Error bound for scheme-I

Lemma 2.2.1. Suppose $x_0, ..., x_N$ are $(N+1)$ distinct roots of Legendre's polynomial, where N be a non-negative integer. Then

$$F(x) - F_N(x) = \frac{F^{(N+1)}(\mu_x)}{2^{N+1}(N+1)!}\left[(2x-1) - x_0\right]\cdots\left[(2x-1) - x_N\right], \;\; \mu_x \in (x_0, x_N), \tag{2.14}$$

where $F_N(x)$ is the numerical approximation of $F(x)$ by ISF.

So,

$$|F(x) - F_N(x)| \leq \frac{K_1K_2}{2^{N+1}(N+1)!} \; ,$$

where $K_1 = \max\limits_{0\leq\mu_x\leq 1}\left|F^{(N+1)}(\mu_x)\right|$ and $K_2 = \max\limits_{0\leq x\leq 1}\left|\left[(2x-1) - x_0\right]\cdots\left[(2x-1) - x_N\right]\right|.$

Proof. Consider,

$$H(x) = F(x) - F_N(x) - A\left[(2x-1) - x_0\right]\cdots\left[(2x-1) - x_N\right], \tag{2.15}$$

where

$$F_N(x) \approx \sum_{k=0}^{N} a_k^{\theta}\theta_k(x). \tag{2.16}$$

Choosing the value of A such that $H(\bar{x}) = 0$, $\bar{x} \in [x_0, x_N]$, so

$$\begin{aligned} F_N(\hat{x}_l) &= \sum_{k=0}^{N} a_k^{\theta}\theta_k(\hat{x}_l), \quad l = 0, 1, ..., N \\ &= \sum_{k=0}^{N} a_k^{\theta}\sqrt{\frac{2}{w_k}}l_k(2\hat{x}_l - 1) \\ &= \sum_{k=0}^{N} F(\hat{x}_k)l_k(x_l) \\ &= F(\hat{x}_l), \end{aligned}$$

and

$$\begin{aligned} H(\hat{x}_l) &= F(\hat{x}_l) - F_N(\hat{x}_l) - A\left[(2\hat{x}_l - 1) - x_0\right]\cdots\left[(2\hat{x}_l - 1) - x_N\right] \\ &= F(\hat{x}_l) - F(\hat{x}_l) - A(x_l - x_0)\cdots(x_l - x_N) \\ &= 0, \quad l = 0, 1, ..., N. \end{aligned}$$

Hence, $H(x)$ vanishes at $(N+2)$ points. Now by Rolle's theorem $H^{(N+1)}(x)$ will vanish for some $\mu_x \in (x_0, x_N)$. Therefore,

$$\begin{aligned} H^{(N+1)}(\mu_x) &= 0, \\ F^{(N+1)}(\mu_x) - 0 - A2^{(N+1)}(N+1)! &= 0, \\ A &= \frac{F^{(N+1)}(\mu_x)}{2^{(N+1)}(N+1)!}. \end{aligned}$$

Now generalizing for any $x \in [x_0, x_N]$, we get required result of (2.14) as

$$\begin{aligned} |F(x) - F_N(x)| &\leq \left|\frac{F^{(N+1)}(\mu_x)}{2^{(N+1)}(N+1)!}\right| \; |[(2x-1) - x_0]\cdots[(2x-1) - x_N]| \\ &= \frac{K_1K_2}{2^{(N+1)}(N+1)!}, \end{aligned}$$

where

$K_1 = \max_{0 \le \mu_x \le 1} \left| F^{(N+1)}(\mu_x) \right|$, and $K_2 = \max_{0 \le x \le 1} \left| [(2x-1) - x_0] \cdots [(2x-1) - x_N] \right|$.

2.2.2 Construction of basis functions for scheme-II

In this scheme, we have obtained the orthonormal basis functions (OLBF), $\phi_0, \phi_1, ..., \phi_N$ using Gram-Schmidt orthogonalization process on Lagrange polynomials $l_0, l_1, ..., l_N$ (given in (2.10)).

$$\phi_0(x) = \frac{l_0(x)}{\|l_0(x)\|}, \tag{2.17}$$

$$\phi_i(x) = \frac{\xi_i(x)}{\|\xi_i(x)\|}, \qquad i = 1, 2, ..., N, \tag{2.18}$$

$$\text{where } u_0(x) = l_0(x),\ \ \xi_i(x) = l_i(x) - u_i(x), \qquad i = 1, 2, ..., N, \tag{2.19}$$

$$u_i(x) = \sum_{j=0}^{i-1} \frac{\int_0^1 l_i(x) u_j(x) dx}{\int_0^1 u_j(x)^2 dx} u_j(x), \qquad i = 1, 2, ..., N,$$

then

$$\Phi(x) = [\phi_0(x), \phi_1(x), ..., \phi_N(x)]^T \tag{2.20}$$

forms an orthonormal basis in $[0, 1]$.

Function approximation by scheme-II

Let $\Phi(x)$ be the OLBF defined by (2.17)-(2.18), then any function $f(x) \in L^2[0,1]$ can be approximated by $\Phi(x)$ as

$$f(x) = \sum_{i=0}^{\infty} a_i^{\phi} \phi_i(x) \approx \sum_{i=0}^{N} a_i^{\phi} \phi_i(x) = [A_\Phi]^T \Phi(x),$$

where $[A_\Phi]^T$ is $1 \times (N+1)$ vector given by

$$[A_\Phi]^T = \begin{bmatrix} \int_0^1 f(x)\phi_0(x)dx \\ \int_0^1 f(x)\phi_1(x)dx \\ \int_0^1 f(x)\phi_2(x)dx \\ \vdots \\ \int_0^1 f(x)\phi_N(x)dx \end{bmatrix}^T_{(N+1)\times 1} . \tag{2.21}$$

Error bound for scheme-II

Lemma 2.2.2. Suppose $x_0, x_1, x_2, ..., x_N$ be $(N+1)$ distinct node points in given closed interval $[0,1]$, where $x_0 = 0 < x_1 < ... < x_N = 1$ and N is a non-negative integer. Let $F(x) \in C^{N+1}[0,1]$, then for each $x \in [0,1]$ there exist μ_1 and $\mu_2 \in (0,1)$ such that

$$F(x) - F_N(x) = R_N(x) + \bar{R}_N(x), \tag{2.22}$$

where $F_N(x)$ is numerical approximation of $F(x)$ by OLBF,

$$R_N(x) = \frac{F^{N+1}(\mu_1)}{(N+1)!}(x-x_0)(x-x_1)\cdots(x-x_N), \tag{2.23}$$

$$\bar{R}_N(x) = \frac{F^N(\mu_2)}{N!}(x-x_0)(x-x_1)\cdots(x-x_{N-1}), \tag{2.24}$$

and

$$|F(x) - F_N(x)| \leq \frac{w_1}{(N+1)!}K_1 + \frac{w_2}{N!}K_2 , \tag{2.25}$$

where $w_1 = \max\limits_{0\le x\le 1} |(x-x_0)\cdots(x-x_N)|$, $w_2 = \max\limits_{0\le x\le 1} |(x-x_0)\cdots(x-x_{N-1})|$, $K_1 = \max\limits_{0\le \mu_1\le 1} |F^{N+1}(\mu_1)|$ and $K_2 = \max\limits_{0\le \mu_2\le 1} |F^N(\mu_2)|$.

Proof. Using equations (2.10), (2.17), (2.18) and (2.19), we can write

$$\xi_0(x) = l_0(x),\ \phi_0(x) = \frac{1}{\|l_0(x)\|}l_0(x) = N_{00}l_0(x),\ where\ N_{00} = \frac{1}{\|l_0(x)\|},$$

$$
\begin{aligned}
\xi_1(x) &= l_1(x) - \frac{\int_0^1 l_1(x)u_0(x)dx}{\int_0^1 u_0^2(x)dx}u_0(x). \\
\text{Now} \quad \phi_1(x) &= \frac{\xi_1(x)}{\|\xi_1(x)\|} \\
&= N_{10}l_0(x) + N_{11}l_1(x),
\end{aligned}
$$

where $N_{10} = -\frac{\int_0^1 l_1(x)u_0(x)dx}{\|\xi_1(x)\|}N_{00}$, and $N_{11} = \frac{1}{\|\xi_1(x)\|}$.

In general, we can write

$$\phi_k(x) = \sum_{j=0}^{k} N_{kj}l_j(x), \quad k = 0, 1, ..., N, \tag{2.26}$$

where

$$N_{kj} = \begin{cases} \frac{1}{\|\xi_k(x)\|}, & \text{if } j = k, \\ -\frac{\int_0^1 l_k(x)\phi_j(x)dx}{\|\xi_k(x)\|}N_{jj}, & \text{if } j = k-1, \\ \frac{-1}{\|\xi_k(x)\|}\sum_{i=0}^{k-j}\left(\int_0^1 l_k(x)\phi_i(x)dxN_{ij}\right), & \text{if } j = 0, 1, 2, ..., k-2. \end{cases} \tag{2.27}$$

Now, approximating the function $F(x)$ by OLBF as

$$F(x) = \sum_{i=0}^{\infty} a_k^{\phi}\phi_k(x), \tag{2.28}$$

and truncated the series (2.28) at $(N+1)$ level and using equations (2.26)- (2.27), equation (2.28) can be written in terms of Lagrangian interpolating polynomials as

[97]

$$\begin{aligned} F(x) &= \sum_{i=0}^{N} c_i l_i(x) + \frac{F^{N+1}(\mu_1)}{(N+1)!}(x-x_0)\cdots(x-x_N) \\ &+ \sum_{i=0}^{(N-1)} \bar{c}_i l_i(x) + \frac{F^{N}(\mu_2)}{(N+1)!}(x-x_0)\cdots(x-x_{N-1}), \end{aligned}$$

or

$$F(x)-F_N(x) = \frac{F^{N+1}(\mu_1)}{(N+1)!}(x-x_0)\cdots(x-x_N) + \frac{F^{N}(\mu_2)}{(N+1)!}(x-x_0)\cdots(x-x_{N-1}), \tag{2.29}$$

where

$$c_i = a_i^{\Phi} N_{ii},$$

and

$$\bar{c}_i = \sum_{j=i}^{(N-1)} a_j^{\Phi} N_{ji}.$$

Hence

$$|F(x)-F_N(x)| \leq \frac{w_1}{(N+1)!}K_1 + \frac{w_2}{N!}K_2. \tag{2.30}$$

2.3 Operational matrices

2.3.1 Operational matrices for integration

Lemma 2.3.1. Let $\Psi(x)$ be the one dimensional vector defined in equation (2.12) and (2.20), then

$$\int_0^x \Psi(t)dt \approx B_1^T \Psi(x) \text{ and } \int_0^x \int_0^t \Psi(t')dt'dt \approx B_2^T \Psi(x),$$

where B_1^T, B_2^T are $(N+1)\times(N+1)$ operational matrices and the coefficients of these matrices are determined by $B_1^T(i,j) = \int_0^1 \left(\int_0^x \psi_i(t)dt\psi_j(x)\right)dx$, $B_2^T(i,j) =$

$\int_0^1 \left(\int_0^x \int_0^t \psi_i(t')dt'dt\psi_j(x) \right) dx$, respectively.

Proof. Let $\Psi(x)$ be the basis function defined in (2.12) and (2.20), then

$$\int_0^x \Psi(t)dt = \begin{bmatrix} \int_0^x \psi_0(t)dt \\ \int_0^x \psi_1(t)dt \\ \int_0^x \psi_2(t)dt \\ \vdots \\ \int_0^x \psi_N(t)dt \end{bmatrix} = \begin{bmatrix} f_0(x) \\ f_1(x) \\ f_2(x) \\ \vdots \\ f_N(x) \end{bmatrix} \approx \begin{bmatrix} \sum_{i=0}^N f_0^i \psi_i(x) \\ \sum_{i=0}^N f_1^i \psi_i(x) \\ \sum_{i=0}^N f_2^i \psi_i(x) \\ \vdots \\ \sum_{i=0}^N f_N^i \psi_i(x) \end{bmatrix} = B_1^T \Psi(x), \quad (2.31)$$

and

$$\int_0^x \int_0^t \Psi(t')dt'dt = \begin{bmatrix} \int_0^x \int_0^t \psi_0(t')dt'dt \\ \int_0^x \int_0^t \psi_1(t')dt'dt \\ \int_0^x \int_0^t \psi_2(t')dt'dt \\ \vdots \\ \int_0^x \int_0^t \psi_N(t')dt'dt \end{bmatrix} = \begin{bmatrix} g_0(x) \\ g_1(x) \\ g_2(x) \\ \vdots \\ g_N(x) \end{bmatrix} \approx \begin{bmatrix} \sum_{i=0}^N g_0^i \psi(x) \\ \sum_{i=0}^N g_1^i \psi(x) \\ \sum_{i=0}^N g_2^i \psi(x) \\ \vdots \\ \sum_{i=0}^N g_N^i \psi(x) \end{bmatrix} = B_2^T \Psi(x), \quad (2.32)$$

where

$$B_1^T = \begin{bmatrix} f_0^0 & f_0^1 & \cdots & f_0^N \\ f_1^0 & f_1^1 & \cdots & f_1^N \\ f_2^0 & f_2^1 & \cdots & f_2^N \\ \vdots & \vdots & \vdots & \vdots \\ f_N^0 & f_N^1 & \cdots & f_N^N \end{bmatrix}_{(N+1)\times(N+1)},$$

and

$$B_2^T = \begin{bmatrix} g_0^0 & g_0^1 & \cdots & g_0^N \\ g_1^0 & g_1^1 & \cdots & g_1^N \\ g_2^0 & g_2^1 & \cdots & g_2^N \\ \vdots & \vdots & \vdots & \vdots \\ g_N^0 & g_N^1 & \cdots & g_N^N \end{bmatrix}_{(N+1)\times(N+1)},$$

and the coefficients of aforementioned matrices B_1^T and B_2^T are calculated by the following formulae

$$f_i^j = \int_0^1 f_i(x)\psi_j(x)dx, \quad i, j = 0, 1, ..., N,$$

$$g_i^j = \int_0^x g_i(x)\psi_j(x)dx, \quad i, j = 0, 1, ..., N,$$

where $f_i(x) = \int_0^x \psi_i(t)dt$ and $g_i(x) = \int_0^x \int_0^t \psi_i(t')dt'dt$.

Lemma 2.3.2. Let $\Psi(x)$ be one dimensional vector defined in equation (2.12) and (2.20), then

$$x\Psi(x) \approx B_3^T\Psi(x),$$

where B_3^T is a $(N+1)\times(N+1)$ operational matrix and coefficients of matrix are determined by $B_3^T(i,j) = \int_0^1 (x\psi_i(x)\psi_j(x))\,dx$.

Proof. Let $\Psi(x)$ be the basis function define in (2.12) and (2.20), then

$$x\Psi(x) = \begin{bmatrix} x\psi_0(x) \\ x\psi_1(x) \\ x\psi_2(x) \\ \vdots \\ x\psi_N(x) \end{bmatrix} = \begin{bmatrix} h_0(x) \\ h_1(x) \\ h_2(x) \\ \vdots \\ h_N(x) \end{bmatrix} \approx \begin{bmatrix} \sum_{i=0}^N h_0^i\psi(x) \\ \sum_{i=0}^N h_1^i\psi(x) \\ \sum_{i=0}^N h_2^i\psi(x) \\ \vdots \\ \sum_{i=0}^N h_N^i\psi(x) \end{bmatrix} = B_3^T\Psi(x), \tag{2.33}$$

where

$$B_3^T = \begin{bmatrix} h_0^0 & h_0^1 & \cdots & h_0^N \\ h_1^0 & h_1^1 & \cdots & h_1^N \\ h_2^0 & h_2^1 & \cdots & h_2^N \\ \vdots & \vdots & \vdots & \vdots \\ h_N^0 & h_N^1 & \cdots & h_N^N \end{bmatrix}_{(N+1)\times(N+1)},$$

and the coefficients of aforementioned are calculated as

$$h_i^j = \int_0^x h_i(x)\psi_j(x)dx, \quad i,j = 0,1,...,N,$$

where $h_i(x) = x\psi_i(x)$.

2.3.2 Product operational matrix

Lemma 2.3.3. Let $\Psi(x)$ be the one dimensional vector defined in equation (2.12) and (2.20) then, $\Psi(x)\Psi^T(x)C \approx \tilde{C}\Psi(x)$, where $\tilde{C}$ is the product operational matrix whose coefficients are given by $\tilde{C_{i,j}} = \sum_{i=0}^N a_i^{kl} c_j, \quad j = 0,1,...,N$ and $a_i^{kl} = \int_0^1 \psi_k(x)\psi_l(x)\psi_i(x)dx, \quad i,l,k = 0,1,...,N$(see [36]).

Proof. By the property of product of two vectors $\Psi(x)$ and $\Psi^T(x)$, we get

$$(C^T\Psi(x))^2 = C^T\Psi(x)\Psi^T(x)C = C^T\tilde{C}\Psi(x), \tag{2.34}$$

where $\tilde{C}\Psi(x) \approx \Psi(x)\Psi^T(x)C$, $C = [c_0, c_1, ..., c_N]$ is a vector and $\tilde{C}$ is product operational matrix.

Now to determine $\tilde{C}$, we have

$$\Psi(x)\Psi^T(x)C = \begin{bmatrix} \sum_{i=0}^{N} c_i\psi_i(x)\psi_0 \\ \sum_{i=0}^{N} c_i\psi_i(x)\psi_1 \\ \sum_{i=0}^{N} c_i\psi_i(x)\psi_2 \\ \vdots \\ \sum_{i=0}^{N} c_i\psi_i(x)\psi_N \end{bmatrix}. \tag{2.35}$$

Using approximation $\psi_i(x)\psi_j(x) = \sum_{k=0}^{N} a_k^{ij}\psi_k(x)$, and multiplying by $\psi_l(x)$, $l = 0, 1, ..., N$, and then integrating from 0 to 1, we can determine the coefficients as

$$\int_0^1 \psi_i(x)\psi_j(x)\psi_l(x)dx = \sum_{k=0}^{N} a_k^{ij} \int_0^1 \psi_k(x)\psi_l(x)dx, \quad i, j, l = 0, 1, ..., N. \tag{2.36}$$

Using the orthonormality condition, we get

$$\int_0^1 \psi_i(x)\psi_j(x)\psi_l(x)dx = a_l^{ij}, \quad i, j, l = 0, 1, ..., N. \tag{2.37}$$

In order to obtain the coefficients of $\Psi(x)$ as in equation (2.34), $\tilde{C}$ is obtained as

$$\tilde{C} = [\tilde{C_{i,j}}], \quad i, j, l = 0, 1, ..., N,$$

where $\tilde{C}$ is a $(N + 1) \times (N + 1)$ matrix, given by

$$\tilde{C_{i,j}} = \sum_{i=0}^{N} a_i^{kl} c_j, \quad j = 0, 1, ..., N.$$

Remark 2.3.1. We will use operational matrices in section 2.4 via scheme-I and scheme-II for the basis functions $\Theta(x)$ and $\Phi(x)$ in place of $\Psi(x)$.

2.4 Applications of proposed schemes

2.4.1 For Lane-Emden equation

Equation (2.3) can be rewritten as

$$xy''(x) + \alpha y'(x) + xf(x, y) = xg(x), \tag{2.38}$$

where $y(x)$ is unknown and to determine this unknown, first we approximate $y''(x)$ as

$$y''(x) \approx \sum_{i=0}^{N} a_i \psi_i(x) = A^T \Psi(x), \tag{2.39}$$

where $A^T = [a_0, a_1, a_2, ..., a_N]$ and $\Psi(x) = [\psi_0, \psi_1, \psi_2, ..., \psi_N]^T$ are $1 \times (N + 1)$ and $(N + 1) \times 1$ vectors, respectively.

Integrating equation (2.39) from 0 to x, we get

$$y'(x) - y'(0) = A^T \int_0^x \Psi(t)dt.$$

Using initial condition $y'(0) = \alpha_1$, we get

$$y'(x) = \alpha_1 + A^T \int_0^x \Psi(t)dt. \tag{2.40}$$

Again integrating equation (2.40) from 0 to x, we get

$$y(x) - y(0) = \alpha_1 + A^T \int_0^x \int_0^t \Psi(t')dt'dt.$$

By making use of initial condition $y(0) = \alpha_0$, we obtain

$$y(x) = \alpha_0 + \alpha_1 x + A^T \int_0^x \int_0^t \Psi(t')dt'dt, \tag{2.41}$$

where the $\alpha_0 + \alpha_1 x$ and α_1 are approximated as

$$\alpha_0 + \alpha_1 x = F(x) \approx \sum_{i=0}^{N} b_i \psi_i(x) = B^T \Psi(x), \tag{2.42}$$

$$\alpha_1 \approx \sum_{i=0}^{N} \bar{b} \psi_i(x) = \bar{B}^T \Psi(x), \tag{2.43}$$

approximating the right hand side of equation 2.38

$$xg(x) \approx \sum_{i=0}^{N} c_i \psi_i(x) = C^T \Psi(x). \tag{2.44}$$

Using equations (2.43) and (2.31), the equation (2.40) becomes

$$y'(x) = (\bar{B}^T + A^T B_1^T)\Psi(x). \tag{2.45}$$

Again, using equation (2.42) and (2.32), equation (2.41) becomes

$$y(x) = (B^T + A^T B_2^T)\Psi(x). \tag{2.46}$$

The approximation of function $f(x, y)$ using (2.46) is obtained as

$$xf(x, y) = \bar{f}(x, y) = \bar{f}(x, (B^T + A^T B_2^T)\Psi(x)). \tag{2.47}$$

Substituting the matrix relations (2.42)-(2.47) into equation (2.38), we get the matrix equation as

$$xA^T \psi(x) + \alpha(\bar{B}^T + A^T B_1^T)\psi(x) + xf(x, (B^T + A^T B_2^T)\psi(x)) = C^T \Psi(x), \tag{2.48}$$

which can be written as

$$A^T x\psi(x) + \alpha(\bar{B}^T + A^T B_1^T)\psi(x) + xf(x, (B^T + A^T B_2^T)\psi(x)) = C^T \Psi(x). \tag{2.49}$$

After using (2.33), it becomes

$$A^T B_3^T \psi(x) + \alpha(B^T + A^T B_2^T)\psi(x) + \bar{f}(x, (B^T + A^T B_2^T)\psi(x)) = C^T \Psi(x). \quad (2.50)$$

This matrix equation leads to linear or non-linear system of algebraic equations according to the function $f(x, y)$. Then the numerical solution of equation (2.3) is

$$y(x) = B^T \Psi(x) + A^T B_2^T \Psi(x),$$

or,

$$y(x) = \left(B^T + A^T B_2^T\right) \Psi(x). \quad (2.51)$$

Remark 2.4.1. Here two basis $\Theta(x)$ (ISF) and $\Phi(x)$ (OLBF) are taken so

(i) Numerical solution with respect to $\Theta(x)$ is

$$y(x) = \left(B_\Theta^T + A_\Theta^T [B_2]_\Theta^T\right) \Theta(x).$$

(ii) Numerical solution with respect to $\Phi(x)$ is

$$y(x) = \left(B_\Phi^T + A_\Phi^T [B_2]_\Phi^T\right) \Phi(x).$$

(iii) The matrices with subscripts Θ and Φ can be calculated using ISF and OLBF, respectively.

2.4.2 *For Riccati's equation*

The general form of Riccati's equation is written as

$$y'(x) = p(x) + q(x)y(x) + r(x)y^2(x), \qquad 0 < x < 1, \quad (2.52)$$

with initial condition as

$$y(0) = \alpha. \tag{2.53}$$

Approximating the derivative term $y'(x)$ as

$$y'(x) = \sum_{i=0}^{N} a_i \psi_i(x) = A^T \Psi(x). \tag{2.54}$$

Integrating the equation (2.54) from 0 to x,

$$y(x) - y(0) = A^T \int_0^x \Psi(t)dt, \tag{2.55}$$

$$y(x) = \alpha + A^T \int_0^x \Psi(t)dt. \tag{2.56}$$

Using equation (2.31), we get

$$A^T \int_0^x \Psi(t)dt = A^T B_1^T \Psi(x), \tag{2.57}$$

where B_1^T is a $(N+1) \times (N+1)$ matrix calculated in equation (2.31) in section 2.3. Now from equation (2.52),

$$A^T \Psi(x) = p(x) + q(x) \left(\alpha + A^T B_1^T \Psi(x)\right) + r(x) \left(\alpha + A^T B_1^T \Psi(x)\right)^2, \tag{2.58}$$

$$\begin{aligned} A^T \Psi(x) = p(x) + \alpha q(x) + q(x) A^T B_1^T \Psi(x) + \alpha^2 r(x) + r(x) \left(A^T B_1^T \Psi(x)\right)^2 \\ +2r(x)\alpha A^T B_1^T \Psi(x), \end{aligned} \tag{2.59}$$

or

$$A^T\Psi(x) = p(x) + \alpha q(x) + A^T B_1^T q(x)\Psi(x) + \alpha^2 r(x) + r(x)\left(A^T B_1^T \Psi(x)\right)^2 \\ + 2\alpha A^T B_1^T r(x)\Psi(x). \tag{2.60}$$

By the property of product of two vectors

$$\left(A^T B_1^T \Psi(x)\right)^2 = \left(\hat{A}^T \Psi(x)\right)^2 = \hat{A}^T \Psi(x)\Psi^T(x)\hat{A} = \hat{A}^T P^* \Psi(x),$$

where $\hat{A} = A^T B_1^T$ is an $1 \times (N+1)$ vector and P^* is $(N+1)\times(N+1)$ product operational matrix.

Let $\hat{A} = [\hat{a_0}, \hat{a_1}, \hat{a_2}, ..., \hat{a_N}]$ be a vector, then we have

$$\Psi(x)\Psi^T(x)\hat{A} = \begin{bmatrix} \sum_{i=0}^{N} \hat{a}_i \psi_i(x)\psi_0 \\ \sum_{i=0}^{N} \hat{a}_i \psi_i(x)\psi_1 \\ \sum_{i=0}^{N} \hat{a}_i \psi_i(x)\psi_2 \\ \vdots \\ \sum_{i=0}^{N} \hat{a}_i \psi_i(x)\psi_N \end{bmatrix} = P^*\Psi(x). \tag{2.61}$$

The following approximations of known functions will also be used in (2.60) as

$$p(x) + \alpha q(x) \approx \sum_{i=0}^{N} d_i \psi_i(x) = D^T \Psi(x), \tag{2.62}$$

$$\alpha^2 r(x) \approx \sum_{i=0}^{N} e_i \psi_i(x) = E^T \Psi(x), \tag{2.63}$$

$$q(x)\Psi(x)dx = \begin{bmatrix} q(x)\psi_0(x) \\ q(x)\psi_1(x) \\ \vdots \\ q(x)\psi_N(x) \end{bmatrix} \approx \hat{A_2}^T\Psi(x), \tag{2.64}$$

$$r(x)\Psi(x)dx = \begin{bmatrix} r(x)\psi_0(x)dx \\ r(x)\psi_1(x)dx \\ \vdots \\ r(x)\psi_N(x)dx \end{bmatrix} \approx \hat{A_3}^T\Psi(x). \tag{2.65}$$

Using equations (2.61)-(2.65) in equation (2.60), we get

$$A^T\Psi(x) = D^T\Psi(x) + A^T B_1^T \hat{A_2}^T\Psi(x) + E^T\Psi(x) + r(x)\hat{A}P^*\Psi(x) + 2\alpha A^T B_1^T \hat{A_3}^T\Psi(x),$$

or

$$A^T\Psi(x) = D^T\Psi(x) + A^T B_1^T \hat{A_2}^T\Psi(x) + E^T\Psi(x) + \hat{A}P^*\left(r(x)\Psi(x)\right) + 2\alpha A^T B_1^T \hat{A_3}^T\Psi(x). \tag{2.66}$$

Again using equation (2.65), we get the following system of algebraic equations

$$A^T\Psi(x) = D^T\Psi(x) + A^T B_1^T \hat{A_2}^T\Psi(x) + E^T\Psi(x) + \hat{A}P^*\hat{A}_3\Psi(x) + 2\alpha A^T B_1^T \hat{A_3}^T\Psi(x), \tag{2.67}$$

or

$$A^T\Psi(x) - D^T\Psi(x) - A^T B_1^T \hat{A_2}^T\Psi(x) - E^T\Psi(x) - \hat{A}P^*\hat{A}_3\Psi(x) - 2\alpha A^T B_1^T \hat{A_3}^T\Psi(x) = 0. \tag{2.68}$$

The equation (2.68) is a system of non-linear algebraic equations in $(N+1)$ variables. After solving it, we get the value of unknown A^T.

Hence the numerical solution of Riccati's equation is

$$y(x) = M_{\alpha}^{T}\Psi(x) + A^{T}B_{1}^{T}\Psi(x), \tag{2.69}$$

where M_{α}^{T} is the $1 \times (N+1)$ vector obtained as

$$\alpha \approx \sum_{i=0}^{N} m_i \psi(x) = M_{\alpha}^{T}\Psi(x).$$

Remark 2.4.2.

(i) Numerical solution with respect to $\Theta(x)$ is

$$y(x) = [M_{\alpha}^{T}]_{\Theta}\Theta(x) + A_{\Theta}^{T}[B_{1}^{T}]_{\Theta}\Theta(x).$$

(ii) Numerical solution with respect to $\Phi(x)$ is

$$y(x) = [M_{\alpha}^{T}]_{\Phi}\Phi(x) + A_{\Phi}^{T}[B_{1}^{T}]_{\Phi}\Phi(x).$$

(iii) The matrices with subscripts Θ and Φ can be calculated using ISF and OLBF, respectively.

2.4.3 For Bessel's equation

Consider the following Bessel equation of order zero as

$$xy''(x) + y'(x) + xy(x) = 0, \tag{2.70}$$

with initial conditions

$$y(0) = 1, y'(0) = 0. \tag{2.71}$$

In order to use the initial conditions, we approximate $y''(x)$ as

$$y''(x) = A^{T}\Psi(x). \tag{2.72}$$

integrating the equation from 0 to x, we get

$$y'(x) - y'(0) = A^T \int_0^x \Psi(t)dt.$$

Using initial condition, we get

$$y'(x) = A^T \int_0^x \Psi(t)dt. \tag{2.73}$$

Again integrating from 0 to x, and using initial condition, we obtain

$$y(x) = 1 + A^T \int_0^x \int_0^t \Psi(t')dt'dt. \tag{2.74}$$

Substituting the values of $y''(x)$, $y'(x)$ and $y(x)$ in equation (2.70), we get,

$$xA^T\Psi(x) + A^T \int_0^x \Psi(t)dt + x\left(1 + A^T \int_0^x \int_0^t \Psi(t')dt'dt\right) = 0,$$

which can be written as

$$xA^T\Psi(x) + A^T \int_0^x \Psi(t)dt + x + xA^T \int_0^x \int_0^t \Psi(t')dt'dt = 0. \tag{2.75}$$

Now,

$$x \approx \sum_{i=0}^{N} a_i\psi_i(x) = Q\Psi(x), \tag{2.76}$$

$$x\int_0^x \int_0^t \Psi(t')dt'dt \approx \begin{bmatrix} x\int_0^x \int_0^t \psi_0(t')dt'dt \\ x\int_0^x \int_0^t \psi_1(t')dt'dt \\ \vdots \\ x\int_0^x \int_0^t \psi_N(t')dt'dt \end{bmatrix} = \hat{A_6}^T\Psi(x), \tag{2.77}$$

where ${\hat{A_6}}^T$ is an $(N+1) \times (N+1)$ matrix and Q is an $1 \times (N+1)$ vector. Using the equations (2.76) and (2.77) in equation (2.75), we get

$$A^T {\hat{B_3}}^T \Psi(x) + A^T B_1^T \Psi(x) + Q\Psi(x) + A^T {\hat{A_6}}^T \Psi(x) = 0, \tag{2.78}$$

$$A^T \left({\hat{B_3}}^T + B_1^T + {\hat{A_6}}^T \right) = -Q. \tag{2.79}$$

This equation is a system of $(N+1)$ linear equations with $(N+1)$ unknowns, which can be rewritten as

$$A^T = -Q \left({\hat{B_3}}^T + B_1^T + {\hat{A_6}}^T \right)^{-1}. \tag{2.80}$$

Hence, the numerical solution of the Bessel's equation is

$$y(x) = \left(M_1 + A^T {B_2}^T \right) \Psi(x),$$

where ${B_2}^T$ is calculated in equation (2.32) of section 2.3 and M_1 is a matrix obtained after approximating 1 using basis function.

Remark 2.4.3.

(i) Numerical solution with respect to $\Theta(x)$ is

$$y(x) = \left([M_1]_\Theta + A_\Theta^T [B_2]_\Theta^T \right) \Theta(x).$$

(ii) Numerical solution with respect to $\Phi(x)$ is

$$y(x) = \left([M_1]_\Phi + A_\Phi^T [B_2]_\Phi^T \right) \Phi(x).$$

(iii) The matrices with subscripts Θ and Φ are calculated using ISF and OLBF, respectively.

2.5 Numerical stability analysis

A numerical scheme is said to be stable if it continuously depends on initial data, i.e., if we add some noise (ϵ) in the initial data then there is a slight change in the final solution. To show the numerical stability, we have added some random noise $\epsilon_1 = 10^{-2}$, $\epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$ in the examples.

2.5.1 Numerical solution with noise

For Lane-Emden equation

Let ϵ be the random noises added to the initial data $y'(x) = \alpha_1$, $y(x) = \alpha_0$, $y'^{\epsilon}(x) = \alpha_1^{\epsilon}$ and $y^{\epsilon}(x) = \alpha_0^{\epsilon}$ are the initial data after adding noise.
Now from equation (2.41), we obtain

$$y(x) = \alpha_0^{\epsilon} + \alpha_1^{\epsilon}x + [A^T]_{\epsilon}\int_0^x \int_0^t \Psi(t')dt'dt.$$

Hence, the numerical solution with noise is

$$y(x) = (B_{\epsilon}^T + [A^T]_{\epsilon}B_2^T)\Psi(x), \tag{2.81}$$

where

$$\alpha_0^{\epsilon} + \alpha_1^{\epsilon}x \approx \sum_{i=0}^{N} b_i^{\epsilon}\psi_i(x) = B_{\epsilon}^T\Psi(x). \tag{2.82}$$

For Riccati's equation

Let $y_{\epsilon}(0) = \alpha_{\epsilon}$ be the initial data after addition of noise in initial condition (2.53). Hence, the numerical solution with noise is

$$y(x) = \left(M_{\alpha}^{\epsilon} + [A^T]_{\epsilon}B_1^T\right)\Psi(x), \tag{2.83}$$

where,

$$\alpha_{\epsilon} \approx M_{\alpha}^{\epsilon}\Psi(x). \tag{2.84}$$

For Bessel's equation

Similarly, after adding noise in initial data, equation (2.71) becomes $y(0) = 1+\epsilon,\ y'(0) = \epsilon$. Hence, from equation (2.73), we get

$$y'(x) = \epsilon + [A^T]_\epsilon \int_0^x \Psi(t)dt. \tag{2.85}$$

From equation (2.74),

$$y(x) = (1+\epsilon) + \epsilon x + [A^T]_\epsilon \int_0^x \int_0^t \Psi(t')dt'dt. \tag{2.86}$$

Hence,

$$y(x) = \left(M_1^\epsilon + [A^T]_\epsilon {B_2}^T\right)\Psi(x), \tag{2.87}$$

where

$$1 + \epsilon + \epsilon x \approx M_1^\epsilon \Psi(x). \tag{2.88}$$

2.6 Numerical examples

2.6.1 Examples for Type-I

Example 2.6.1. (Isothermal gas sphere equation [65])

$$\begin{cases} y''(x) + \frac{2}{x}y'(x) + e^{y(x)} = 0, \\ \qquad \text{y(0)=0,} \qquad \text{y'(0)=0.} \end{cases}$$

In this example the following approximation for $e^{y(x)}$ is used

$$e^{y(x)} \approx 1 + A^T\Psi(x) + \left(A^T\Psi(x)\right)^2,$$

where

$$y \approx A^T\Psi(x).$$

The unknown coefficient vectors A^T for example 2.6.1 using scheme-I and scheme-II are given in Table 2.1 and Table 2.2, respectively for $N = 2, 3$ and 4. The values of unknown coefficient with noise for $N = 4$ by scheme-I and scheme-II are given in Table 2.3 and Table 2.4, respectively. Moreover, the values of vector B_{ϵ}^{T} described in equation (2.82) by scheme-I and scheme-II are given in Table 2.5 and Table 2.6, respectively.

Table 2.1: Unknown coefficient vector for $N = 2, 3$ and 4 using scheme-I.

N	A_{Θ}^{T}
2	$[-0.206147, -0.175058, -0.138888]$
3	$[-0.184243, -0.138807, -0.166449, -0.107495]$
4	$[-0.164960, -0.160484, -0.114652, -0.136695, -0.087615]$

Table 2.2: Unknown coefficient vector for $N = 2, 3$ and 4 using scheme-II.

N	A_{Φ}^{T}
2	$[-0.156150, -0.247680, -0.082116]$
3	$[-0.148274, -0.159395, -0.203088, -0.061940]$
4	$[-0.115753, -0.172319, -0.128175, -0.174629, -0.049567]$

Table 2.3: $[A^T]_{\epsilon}$ using scheme-I for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	$[A_{\Theta}^{T}]_{\epsilon}$
ϵ_1	$[-0.169177, -0.152346, -0.155574, -0.136845, -0.088545]$
ϵ_2	$[-0.165380, -0.159670, -0.118743, -0.136709, -0.087705]$
ϵ_3	$[-0.165001, -0.160403, -0.115060, -0.136696, -0.087622]$

Table 2.4: $[A^T]_{\epsilon}$ using scheme-II for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	$[A_{\Phi}^{T}]_{\epsilon}$
ϵ_1	$[-0.156140, -0.162791, -0.131042, -0.177882, -0.053904]$
ϵ_2	$[-0.119790, -0.171365, -0.128460, -0.174953, -0.050000]$
ϵ_3	$[-0.116156, -0.172223, -0.128203, -0.174661, -0.049610]$

Table 2.5: B_{ϵ}^{T} using scheme-I for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	B_{ϵ}^{T}
ϵ_1	$[0.008, 0.006020, 0.003603, 0.008655, 0.006722]$
ϵ_2	$[0.0008, 0.000602, 0.0003603, 0.000865, 0.000672]$
ϵ_3	$[0.00008, 0.000060, 0.000036, 0.000086, 0.000067]$

Table 2.6: B_ϵ^T using scheme-II for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	B_ϵ^T
ϵ_1	$[0.003427, 0.007127, 0.005366, 0.011224, 0.004000]$
ϵ_2	$[0.000342, 0.000712, 0.000536, 0.001122, 0.000400]$
ϵ_3	$[0.000034, 0.000071, 0.000053, 0.000112, 0.000040]$

The values of absolute error at some points of domain by scheme-I and scheme-II with noises $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$ for $N = 4$ are given in Table 2.7.

Table 2.7: Absolute error for example 2.6.1, with noise $\epsilon_1 = 10^{-2}$, $\epsilon_2 = 10^{-3}$, $\epsilon_3 = 10^{-4}$ and $N = 4$.

	Scheme-I			Scheme-II		
x	$\epsilon_1 = 10^{-2}$	$\epsilon_2 = 10^{-3}$	$\epsilon_3 = 10^{-4}$	$\epsilon_1 = 10^{-2}$	$\epsilon_2 = 10^{-3}$	$\epsilon_3 = 10^{-5}$
0.1	1.0364e-02	1.0318e-03	9.8641e-05	1.0364e-02	1.0318e-03	9.8641e-05
0.2	1.0113e-02	1.0133e-03	1.0329e-04	1.0113e-02	1.0133e-03	1.0329e-04
0.3	9.9125e-03	9.9161e-04	9.9479e-05	9.9125e-03	9.9161e-04	9.9479e-05
0.4	9.7526e-03	9.7376e-04	9.5791e-05	9.7526e-03	9.7376e-04	9.5791e-05
0.5	9.6168e-03	9.6078e-04	9.5035e-05	9.6168e-03	9.6078e-04	9.5035e-05
0.6	9.4817e-03	9.4947e-04	9.6030e-05	9.4817e-03	9.4947e-04	9.6030e-05
0.7	9.3194e-03	9.3488e-04	9.6117e-05	9.3194e-03	9.3488e-04	9.6117e-05
0.8	9.1015e-03	9.1358e-04	9.4383e-05	9.1015e-03	9.1358e-04	9.4383e-05
0.9	8.8023e-03	8.8759e-04	9.5604e-05	8.8023e-03	8.8759e-04	9.5604e-05
1.0	8.4039e-03	8.6903e-04	1.1490e-04	8.4039e-03	8.6903e-04	1.1490e-04

The comparison of numerical scheme-I and scheme-II with numerical schemes given in [71],[81], [68] is presented in Table 2.8 and comparison of absolute error at some points of the domain $[0, 1]$ is given in Table 2.9.

Table 2.8: Comparison of numerical solution with existing schemes.

x	Present Schemes, N=8		By [81]	By [71]	By [68]
	Scheme-I	Scheme -II			
0.0	-2.043595404e-10	-5.194067568e-08	0.0000000000	0.0000000000	-9.248064e-18
0.1	-0.0016658339	-0.00166586155	-0.0016664188	-0.0016658339	-0.0016658333
0.2	-0.0066533671	-0.00665336928	-0.0066539713	-0.0066533671	-0.0066533333
0.5	-0.0411539978	-0.041154078892	-0.0411545150	-0.0411539568	-.04145833333
1.0	-0.1588370919	-0.15883515641	-0.1588281737	-0.1588273527	-0.1583333333

Table 2.9: Comparison of absolute error with existing schemes discussed in [81] and [68].

x	Present Schemes, N=8 Scheme-I	Scheme-II	By [81], N=30	By [68], N=10
0.0	2.04e-10	5.19e-08	0.00e-00	9.24e-18
0.1	1.88e-11	5.49e-09	5.85e-07	5.28e-10
0.2	2.48e-11	1.73e-08	6.04e-07	3.37e-08
0.5	8.70e-08	1.00e-08	5.58e-07	8.12e-06
1.0	2.53e-05	2.72e-05	8.20e-07	4.93e-04

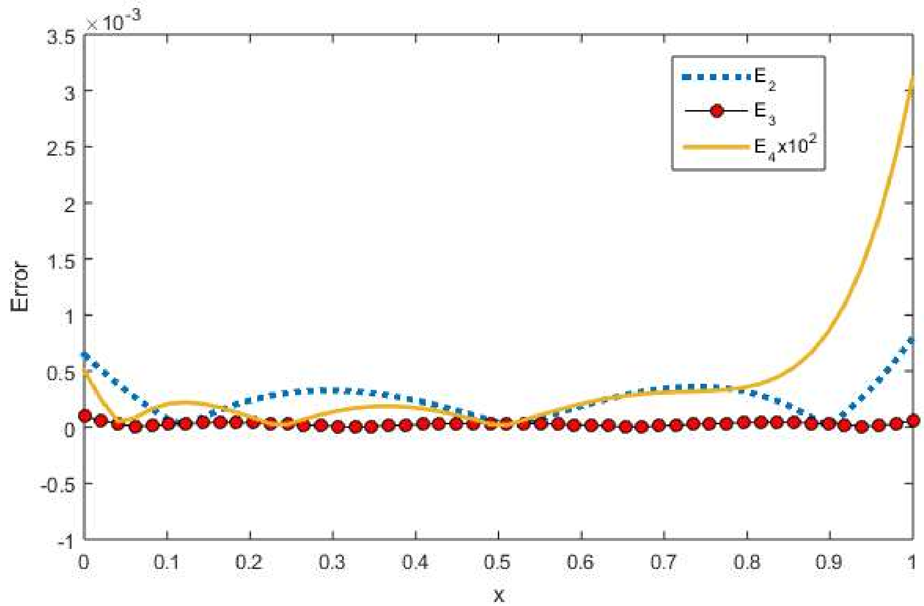

Figure 2.1: *Plots of absolute error of example 2.6.1 using scheme-I for* $N = 2$, $N = 3$ *and* $N = 4$ *without noise.*

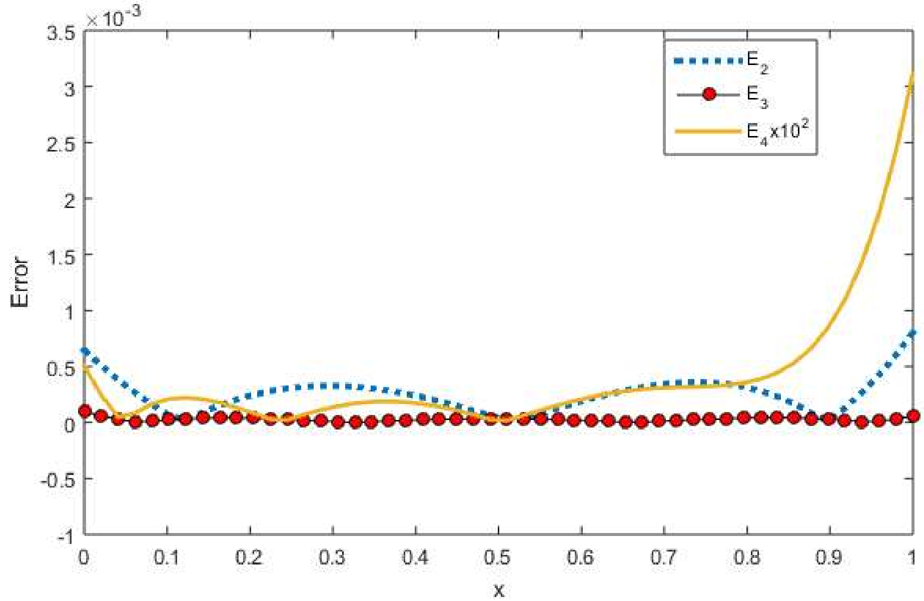

Figure 2.2: *Plots of absolute error of example 2.6.1 using scheme-II for $N = 2,\ N = 3$ and $N = 4$ without noise.*

2.6.2 Examples for Type-II

Example 2.6.2. Consider the example of Type-II problem [99]

$$\begin{cases} y'(x) = 1 + x^2 - y^2(x), \\ \qquad \text{y(0)=1}, \end{cases}$$

with exact solution as $y(x) = x + \frac{e^{-x^2}}{1+\int_0^x e^{-t^2}dt}$.

The unknown coefficient vectors A^T for example 2.6.2 using scheme-I and scheme-II are given in Table 2.10 and Table 2.11, respectively for $N = 2,\ 3$ and 4. The values of unknown coefficient with noise for $N = 4$ are given in Table 2.12 and Table 2.13.

Table 2.10: Unknown coefficient vector for $N = 2,\ 3$ and 4 using scheme-I.

N	A_Θ^T
2	$[0.120934, 0.006800, 0.240086]$
3	$[0.050363, 0.001966, 0.169371, 0.202146]$
4	$[0.097613, 0.022449, 0.000740, 0.179771, 0.172461]$

Table 2.11: Unknown coefficient vector for $N = 2,\ 3$ and 4 using scheme-II.

N	A_Φ^T
2	$[-0.023509, 0.193693, 0.185045]$
3	$[0.017932, 0.008801, 0.232163, 0.133380]$
4	$[-0.009227, 0.054144, 0.034478, 0.237602, 0.106926]$

Value of vector M_α^T using scheme-I for $N = 4$ is

$M_\alpha^T = [0.533333, 0.489197, 0.344185, 0.489197, 0.344185]$.

Value of vector M_α^T using scheme-II for $N = 4$ is

$M_\alpha^T = [0.342732, 0.539440, 0.393929, 0.629569, 0.200000]$.

Table 2.12: $[A^T]_\epsilon$ using scheme-I for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	$[A_\Theta^T]_\epsilon$
ϵ_1	$[0.093600, 0.016254, -0.005553, 0.177553, 0.171363]$
ϵ_2	$[0.097211, 0.021830, 0.000112, 0.179548, 0.172351]$
ϵ_3	$[0.097573, 0.022387, 0.000676, 0.179748, 0.172450]$

Table 2.13: $[A^T]_\epsilon$ using scheme-II for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	$[A_\Phi^T]_\epsilon$
ϵ_1	$[-0.015713, 0.048019, 0.030959, 0.234790, 0.106342]$
ϵ_2	$[-0.009873, 0.053532, 0.034125, 0.237320, 0.106867]$
ϵ_3	$[-0.009233, 0.054137, 0.034474, 0.237599, 0.106925]$

The values of $[M_\alpha^T]_\epsilon$ described in equation (2.84) with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$ using scheme-I and scheme-II are given in Table 2.14 and Table 2.15, respectively.

Table 2.14: $[M_\alpha^T]_\epsilon$ using scheme-I for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	$[M_\alpha^T]_\epsilon$
ϵ_1	$[0.538666, 0.494089, 0.347627, 0.494089, 0.347627]$
ϵ_2	$[0.533866, 0.489686, 0.344529, 0.489686, 0.344529]$
ϵ_3	$[0.533386, 0.489246, 0.344219, 0.489246, 0.344219]$

Table 2.15: $[M_\alpha^T]_\epsilon$ using scheme-II for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	$[M_\alpha^T]_\epsilon$
ϵ_1	$[0.346160, 0.544834, 0.397868, 0.635865, 0.202000]$
ϵ_2	$[0.343075, 0.539980, 0.394323, 0.630199, 0.200200]$
ϵ_3	$[0.342767, 0.539494, 0.393968, 0.629632, 0.200020]$

The values of absolute error for example 2.6.2 at some points of domain using scheme-I and scheme-II with noises $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$ for $N = 4$ are given in Table 2.16.

Table 2.16: Absolute error for example 2.6.2, with noise $\epsilon_1 = 10^{-2}$, $\epsilon_2 = 10^{-3}$, $\epsilon_3 = 10^{-4}$ and $N = 4$.

	Scheme-I			Scheme-II		
x	$\epsilon_1 = 10^{-2}$	$\epsilon_2 = 10^{-3}$	$\epsilon_3 = 10^{-4}$	$\epsilon_1 = 10^{-2}$	$\epsilon_2 = 10^{-3}$	$\epsilon_3 = 10^{-4}$
0.1	1.0041e-02	1.0454e-03	1.4584e-04	1.0041e-02	1.0454e-03	1.4588e-04
0.2	7.9826e-03	7.8210e-04	6.1358e-05	7.9826e-03	7.8210e-04	7.9276e-05
0.3	6.4022e-03	6.4505e-04	6.8311e-05	6.4022e-03	6.4505e-04	1.0063e-04
0.4	5.1259e-03	5.2730e-04	6.6334e-05	5.1259e-03	5.2730e-04	1.1023e-04
0.5	4.0786e-03	4.1128e-04	4.3500e-05	4.0784e-03	4.1128e-04	9.6711e-05
0.6	3.2273e-03	3.1225e-04	1.9790e-05	3.2273e-03	3.1225e-04	8.0534e-05
0.7	2.5498e-03	2.4542e-04	1.4160e-05	2.5498e-03	2.4542e-04	8.1025e-05
0.8	2.0153e-03	2.0879e-04	2.7434e-05	2.0153e-03	2.0879e-04	9.9291e-05
0.9	1.5766e-03	1.7370e-04	3.2831e-05	1.5766e-03	1.7370e-04	1.0874e-04
1.0	1.1596e-03	7.5142e-05	3.3737e-05	1.1596e-03	7.5142e-05	4.5370e-05

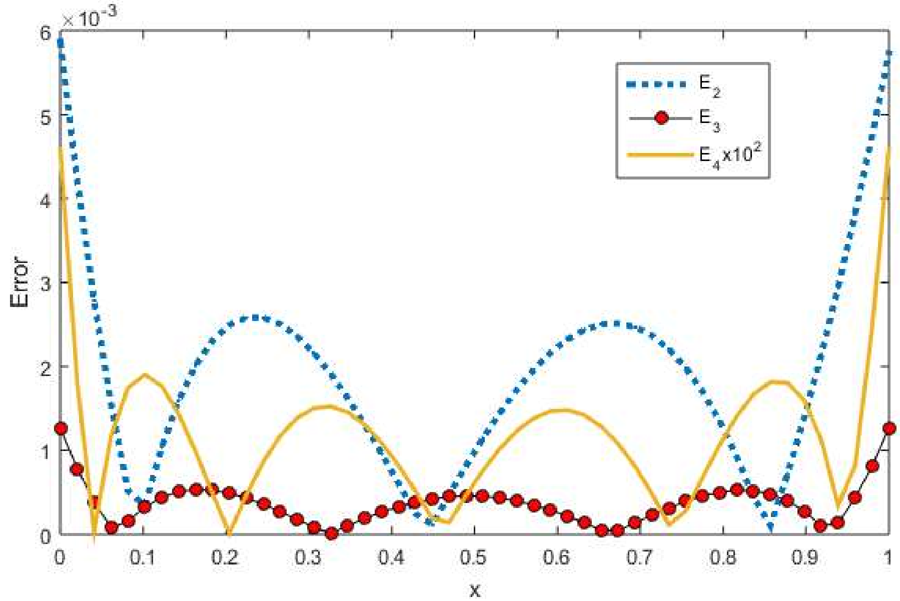

Figure 2.3: *Plots of the absolute error for example 2.6.2 using scheme-I for* $N = 2, N = 3$ *and* $N = 4$ *without noise.*

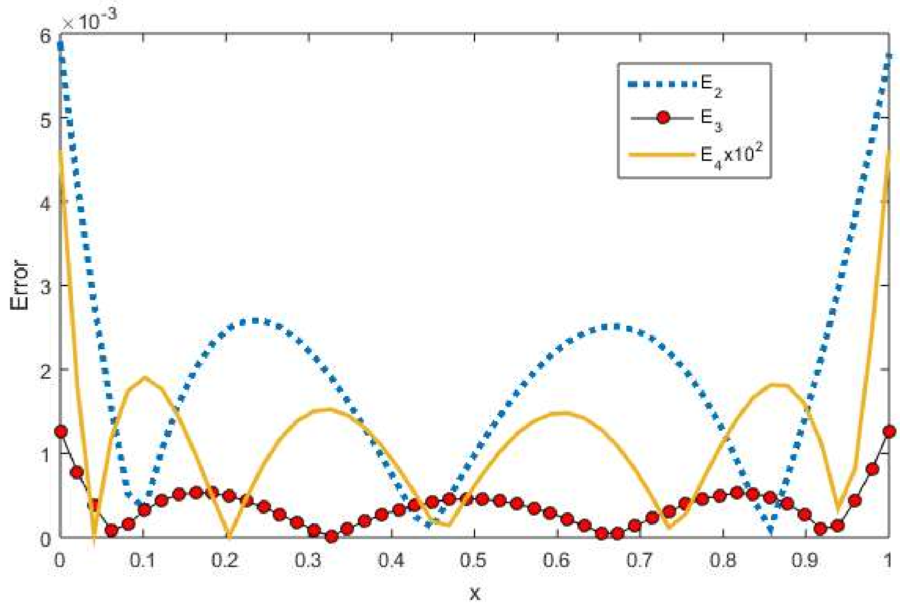

Figure 2.4: *Plots of the absolute error for example 2.6.2 using scheme-I for* $N = 2, N = 3$ *and* $N = 4$ *without noise.*

2.6.3 Examples for Type-III

Example 2.6.3. Consider the Bessel equation of order zero ([54]) as

$$\begin{cases} xy''(x) + y'(x) + xy = 0, \\ \text{y(0)=1}, \quad \text{y'(0)=0}. \end{cases}$$

The unknown coefficient vectors A^T for example 2.6.3 using scheme-I and scheme-II are given in Table 2.17 and Table 2.18, respectively for $N = 2, 3$ and 4. The values of unknown coefficient $[A^T]_\epsilon$ with noise for $N = 4$ are given in Table 2.19 and Table 2.20, respectively. Moreover, the values of vector M_1^ϵ described in equation (2.88) by scheme-I and scheme-II are given in Table 2.21 and Table 2.22, respectively.

Table 2.17: Unknown coefficient vector for $N = 2, 3$ and 4 using scheme-I.

N	A_Θ^T
2	$[-0.302428, -0.262420, -0.189960]$
3	$[-0.273963, -0.208128, -0.238920, -0.144784]$
4	$[-0.242097, -0.239733 -0.171950, -0.192516, -0.117072]$

The value of vector M_1 using scheme-I for $N = 4$ is

$M_1 = [0.344185, 0.489197, 0.5333333, 0.489197, 0.344185]$.

Table 2.18: Unknown coefficient vector for $N = 2, 3$ and 4 using scheme-II.

N	A_Φ^T
2	$[-0.2363929, -0.358970, -0.108034]$
3	$[-0.221051, -0.240010, -0.288698, -0.081261]$
4	$[-0.174437, -0.254943, -0.191864, -0.244852, -0.065030]$

Similarly, the value of vector M_1 using scheme-II for $N = 4$ is

$M_1 = [0.342732, 0.539440, 0.393929, 0.629569, 0.200000]$.

Table 2.19: $[A^T]_\epsilon$ using scheme-I for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	$[A_\Theta^T]_\epsilon$
ϵ_1	$[-0.245265, -0.239551, -0.205780, -0.194274, -0.118354]$
ϵ_2	$[-0.242413, -0.2397148, -0.175333, -0.192691, -0.117200]$
ϵ_3	$[-0.242128, -0.239731, -0.172288, -0.192533, -0.117084]$

Table 2.20: $[A^T]_\epsilon$ using scheme-II for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	$[A^T_\Phi]_\epsilon$
ϵ_1	$[-0.208088, -0.253679, -0.1938954, -0.248602, -0.067698]$
ϵ_2	$[-0.177802, -0.254816, -0.192067, -0.245227, -0.065296]$
ϵ_3	$[-0.174773, -0.254930, -0.191884, -0.244889, -0.065056]$

Table 2.21: M_1^ϵ using scheme-I for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	M_1^ϵ
ϵ_1	$[0.5413333, 0.495218, 0.347788, 0.497852, 0.350907]$
ϵ_2	$[0.5341333, 0.489799, 0.344545, 0.490063, 0.344857]$
ϵ_3	$[0.533413, 0.489257, 0.344221, 0.489284, 0.344252]$

Table 2.22: M_1^ϵ using scheme-II for $N = 4$ with noise $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$.

Noise	M_1^ϵ
ϵ_1	$[0.346160, 0.546567, 0.399295, 0.640794, 0.204000]$
ϵ_2	$[0.343075, 0.540153, 0.394465, 0.630692, 0.200400]$
ϵ_3	$[0.342767, 0.539511, 0.393982, 0.629681, 0.200040]$

The values of absolute error for example 2.6.3 at some points of domain using scheme-I and scheme-II with noises $\epsilon_1 = 10^{-2}, \epsilon_2 = 10^{-3}$ and $\epsilon_3 = 10^{-4}$ for $N = 4$ are given in Table 2.23.

Table 2.23: Absolute error for example 2.6.3, with noise $\epsilon_1 = 10^{-2}$, $\epsilon_2 = 10^{-3}$, $\epsilon_3 = 10^{-4}$ and $N = 4$.

	Scheme-I			Scheme-II		
x	$\epsilon_1 = 10^{-2}$	$\epsilon_2 = 10^{-3}$	$\epsilon_3 = 10^{-4}$	$\epsilon_1 = 10^{-2}$	$\epsilon_2 = 10^{-3}$	$\epsilon_3 = 10^{-5}$
0.1	1.0238e-02	1.0198e-03	9.7912e-05	1.0238e-02	1.0198e-03	9.7912e-05
0.2	1.0329e-02	1.0347e-03	1.0520e-04	1.0329e-02	1.0347e-03	1.0520e-04
0.3	1.0239e-02	1.0242e-03	1.0272e-04	1.0239e-02	1.0246e-03	1.0272e-04
0.4	1.0044e-02	1.0030e-03	9.8904e-05	1.0044e-02	1.0030e-03	9.8904e-05
0.5	9.7971e-02	9.7876e-04	9.6920e-05	9.7971e-03	9.7876e-04	9.6920e-05
0.6	9.5213e-03	9.5292e-04	9.6086e-05	9.5213e-03	9.5292e-04	9.6086e-05
0.7	9.2167e-03	9.2323e-04	9.3888e-05	9.2167e-03	9.2323e-04	9.3888e-05
0.8	8.8616e-03	8.8605e-04	8.8525e-05	8.8616e-03	8.8605e-04	8.8525e-05
0.9	8.4146e-03	8.3947e-04	8.1957e-05	8.4146e-03	8.3947e-04	8.1957e-05
1.0	7.8206e-03	7.8681e-04	8.3430e-05	7.8206e-03	7.8681e-04	8.3430e-05

In Table 2.24, we have compared the results obtained by scheme-I and scheme-II with the numerical results obtained in the article [54] and with $J_0(x)$.

Here $J_0(x)$ is taken as

$$J_0(x) \approx 1 - \left(\frac{x}{2}\right)^2 + \frac{1}{4}\left(\frac{x}{2}\right)^4 - \frac{1}{36}\left(\frac{x}{2}\right)^6 + \frac{1}{576}\left(\frac{x}{2}\right)^8.$$

Table 2.24: Comparison of numerical solutions.

x	Present Schemes, N=4 Scheme-I	Scheme-II	By [54]	Solution of $J_0(x)$
0.0	0.999995	1.000000	1.000000	1.000000
0.1	0.997503	0.997501	0.997502	0.997501
0.2	0.990025	0.990024	0.990024	0.990024
0.3	0.977625	0.977626	0.977625	0.977626
0.4	0.960396	0.960398	0.960396	0.960398
0.5	0.938469	0.938469	0.938468	0.938469
0.6	0.912006	0.912004	0.912004	0.912004
0.7	0.881202	0.881200	0.881200	0.881200
0.8	0.846286	0.846287	0.846285	0.846287
0.9	0.807521	0.807523	0.807524	0.807523
1.0	0.765203	0.765197	0.765197	0.765197

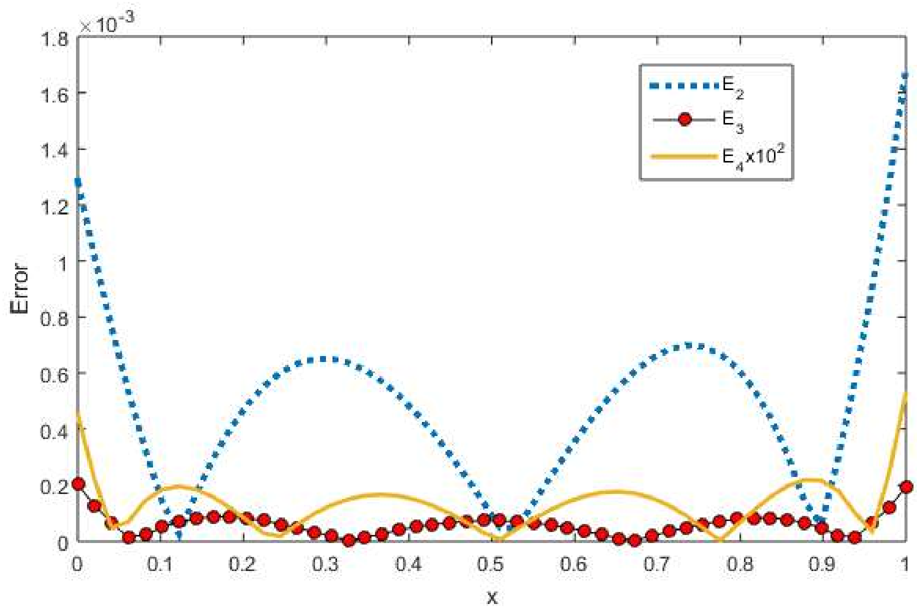

Figure 2.5: *Plots of the absolute error of example 2.6.3 using scheme-I for* $N = 2, N = 3$ *and* $N = 4$ *without noise.*

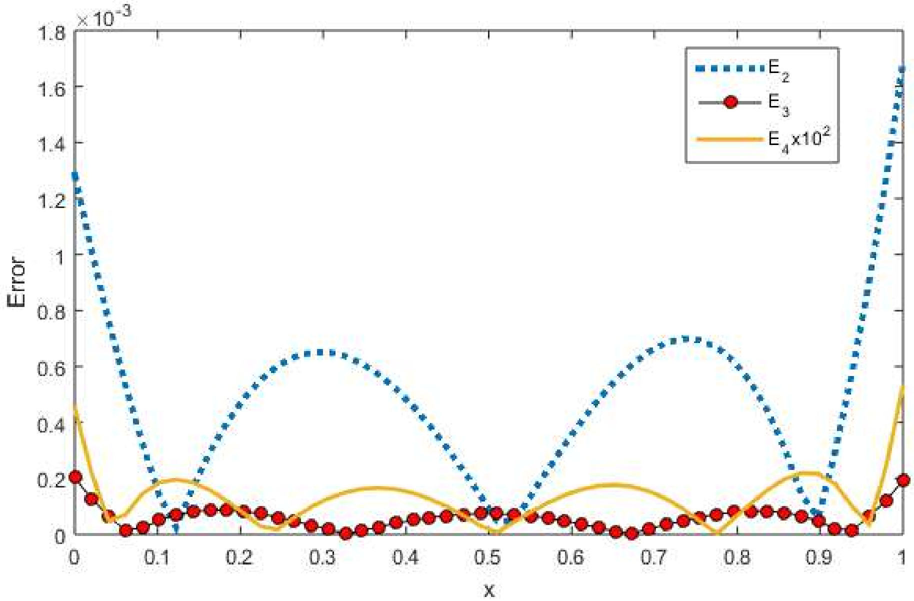

Figure 2.6: *Plots of the absolute error of example 2.6.3 using scheme-II for* $N = 2, N = 3$ *and* $N = 4$ *without noise.*

2.7 Conclusion

This chapter deals with two numerical schemes namely, scheme-I and scheme-II, which are based on ISF and OLBF approximations, respectively to find the numerical solutions of proposed problems (see section 2.1 type-I, type-II and type-III). New operational matrices based on ISF and OLBF are constructed for integration and product (see Lemma 2.3.1- 2.3.3). The implementation of schemes leads to the system of linear and non-linear algebraic equations and to solve this system of equations, the *fsolve* from the toolbox of *Matlab* is used. Moreover, the convergence of the function approximation has been proven by providing an upper bound (see Lemma 2.2.1 and 2.2.2). A drive is taken to establish the numerical stability of the proposed schemes at some noises in section 2.5. Furthermore, some literature test problems have been provided to illustrate the efficiency and accuracy of scheme-I and scheme-II. Hence it can be concluded through figures (Figures 2.1, 2.3, 2.5 for scheme-I and 2.2, 2.4, 2.6 for scheme-II) and tables (Tables 2.7, 2.8, 2.9, 2.16, 2.23 and 2.24) that the numerical schemes used in this chapter produce better accuracy for small number of basis functions (for $N = 2, 3$ and 4). However in scheme-I, the error increases as N increase since it is the polynomial interpolation. The advantage of the scheme-II over scheme-I is that one can construct the orthonormal polynomial without finding the roots of the Legendre polynomial. On the other hand, the floating-point arithmetic is used in each operation. Therefore, the computational error may increases as N is increased. Finally, it is concluded that the errors associated with the scheme-I and scheme-II are smaller as compared to the existing schemes ([81], [100] and [54]). Also, these schemes can be applied to any class of linear and non-linear initial value problems.

****