An-Najah National University

Faculty of Graduate studies

NUMERICAL SOLUTIONS OF SYSTEM OF FREDHOLM INTEGRO-DIFFERENTIAL EQUATIONS

By

Ansam Mohammad Zaidan

Supervisor

Prof. Naji Qatanani

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

2023

NUMERICAL SOLUTIONS OF SYSTEM OF FREDHOLM INTEGRO-DIFFERENTIAL EQUATIONS

By

Ansam Mohammad Abdel-fattah Zaidan

This Thesis was Defended Successfully on 29/1/2023 and approved by

Prof. Naji Qatanani
Supervisor

Signature

Dr. Abdelhalim Ziqan
External Examiner

Signature

Dr. Mohammed Borini Yasin
Internal Examiner

Signature

Dedication

This thesis is dedicated to my husband, my parents, my grandmother, my sisters and my brothers for their support, as well as to my whole family and friends. With respect and love.

Acknowledgments

First, I want to thank Allah for giving me the strength and courage to complete this work. I am grateful to my supervisor Prof. Naji Qatanani for his continuous help and guidance. Furthermore, I would like to thank all the stuff member of Mathematics Department at An-Najah National University for their contribution during my studies.

Declaration

I, the undersigned, declare that I submitted the thesis entitled:

NUMERICAL SOLUTIONS OF SYSTEM OF FREDHOLM INTEGRO-DIFFERENTIAL EQUATIONS

I declare that 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: Ansam Mohammad Abdel-fattah Zaidan

Signature: ____________________

Date: 29/1/2023

List of Contents

List of Tables

List of Figures

List of Appendices

SOLUTION OF SYSTEM OF FREDHOLM INTEGRO-DIFFERENTIAL EQUATIONS

By

Ansam Mohammad Zaidan

Supervisor

Prof. Naji Qatanani

Abstract

Background: System of Fredholm integro-differential equations is a hot topic in applied mathematics; in particular, the field of signal processing, neural networks and the development of telecommunications.

Aims: This thesis presents some numerical methods based on matrix polynomials to solve system of Fredholm integro-differential equations. These methods are: Laguerre polynomials, Bessel functions and Legendre polynomial.

Methods: In this study, we use Matlab software to implement these methods in form of algorithms to solve some numerical examples.

Results: Numerical results show that the convergence and accuracy proprieties of these methods are in good agreement with the exact solution.

Conclusion: A comparison of numerical results illustrated in tables and figures shows that the Laguerre polynomials method provides more accurate results than the other methods and therefore more effective.

Keywords: solution of system, fredholm, integro-differential, matlab, numerical methods, integrodifferential.

Chapter One
Introduction

The system of integro-differential equations is used in the study of processes in chemistry, physics like the theory of signal processing and neural networks, biology and economy. It also presents a survey of the earlier works that deal with the boundary value and initial problems of the Fredholm and Volterra integro-differential models. Many authors discuss the solvability of different problems and the methods that can be used to find their solutions. Much research was based on finding easy and simple ways to solve these equations, but a lot of research was limited to solving one equation[34, 27, 9].

More research have been carried out to solve systems of Integral equations and system of integro-differential equations. This include: Hybrid orthogonal Bernstein[4], Rationalized Haar Functions Method[29], Adomian decomposition method[33, 7], Galerkin methods with Hybrid Functions[19], Homotopy perturbation method[5], Differential Transform method[25], Tau method[28], Modified Decomposition method[21], Bessel matrix method[38, 24], the block pulse functions method[20], Fibonacci polynomials method[23, 30], Sinc Collocation Method[16], Bernoulli Polynomials Operational Matrix[18, 15], spectral collocation matrix method[26], Exponential spline method[32] and Reconstruction of Variational Iteration method (RVIM)[14, 13].

Most of the aforementioned methods were implemented to find an approximate solution to systems of integral equations and systems of integro-differential equations and then compare the solution with some existing methods that appeared in the literature.

Zaffer Elahi et al [8], presents a framework that combines the operations of Laguerre operational matrix and Fredholm linear frameworks to solve the system of integrated variable equations (IDEs). The results of the study indicate that the Laguerre approach is more accurate than the other methods.

In applied mathematics, many problems lead to the following linear system of Fredholm integro-differential equations. A system of Fredholm integro-differential equations is a set of equations in which the unknown functions appear inside an integral sign and contains ordinary derivatives, given as

$$\sum_{k=0}^{n}\sum_{q=1}^{r}\sigma_{p,q}^{k}(x)y_{q}^{(k)}(x) = f_p(x) + \int_a^b \sum_{q=1}^{r} K_{p,q}(x,t)y_q(t)\mathrm{d}t, \qquad p = 1,2,\ldots,r, \quad 0 \le a \le x \le b, \tag{1.1}$$

subject to the initial conditions:

$$\sum_{q=0}^{n-1}(a_{p,q}^{k}y_{k}^{(q)}(a) + b_{p,q}^{k}y_{k}^{(q)}(b)) = \mu_{k,p}, \quad p = 0,1,\ldots,n-1, \quad k = 1,2,\ldots,r, \tag{1.2}$$

where $y_q(x)$ is the unknown solution and $\sigma_{p,q}^{k},\ K_{p,q}(x,t),\ f_p(x)$ are known functions defined an interval $[a,b]$, the kernel function $K_{p,q}(x,t)$ can be expanded using Maclaurin series and also $a_{p,q}^{k},\ b_{p,q}^{k},\ \mu_{k,p}$ are real constants.

In this work we present some numerical methods based on matrix polynomials to solve system of Fredholm integro-differential equations. These methods are: Laguerre polynomials, Bessel functions and Legendre polynomial. We use Matlab software to implement these methods in the form of algorithms to solve some numerical examples. A comparison between the methods will be carried out.

1.1 Thesis Organization

Chapter one contains some mathematical preliminaries and some basic concepts of system of integro-differential equations and their solvability. In chapter two, we present three numerical methods for solving systems of integro-differential equations, these include: laguerre polynomials, Bessel functions and legendre polynomials. In chapter three, the proposed methods are implemented using Matlab software to solve some numerical examples with known analytical solutions.

Chapter Two

Mathematical Preliminaries

In this chapter, we introduce some basic concepts of systems of integro-differential equations and their solvability.

2.1 Classification of systems of Integro-Differential Equations

2.1.1 System of Volterra Integro-Differential Equations

A system of Volterra integro-differential equations can be written as[1]:

$$\begin{aligned}
y_1^{(n)}(x) &= f_1(x) + \int_a^x (K_{11}(x,t)y_1(t) + K_{12}(x,t)y_2(t) + ... + K_{1j}(x,t)y_j(t))dt,\\
y_2^{(n)}(x) &= f_2(x) + \int_a^x (K_{21}(x,t)y_1(t) + K_{22}(x,t)y_2(t) + ... + K_{2j}(x,t)y_j(t))dt,\\
&\vdots\\
y_i^{(n)}(x) &= f_i(x) + \int_a^x (K_{i1}(x,t)y_1(t) + K_{i2}(x,t)y_2(t) + ... + K_{ij}(x,t)y_j(t))dt,
\end{aligned} \tag{2.1}$$

The functions $y_1(x)$, $y_2(x), \ldots,\ y_i(x)$ are to be determined, the kernels $K_{i1}(x,t)$, $K_{i2}(x,t), \ldots,\ K_{ij}(x,t)$ and the functions $f_i(x)$ are real-valued functions.

2.1.2 System of Fredholm Integro-Differential Equations

A system of Fredholm integro-differential equations can be expressed as:

$$\begin{aligned}
y_1^{(n)}(x) &= f_1(x) + \int_a^b (K_{11}(x,t)y_1(t) + K_{12}(x,t)y_2(t) + ... + K_{1j}(x,t)y_j(t))dt,\\
y_2^{(n)}(x) &= f_2(x) + \int_a^b (K_{21}(x,t)y_1(t) + K_{22}(x,t)y_2(t) + ... + K_{2j}(x,t)y_j(t))dt,\\
&\vdots\\
y_i^{(n)}(x) &= f_i(x) + \int_a^b (K_{i1}(x,t)y_1(t) + K_{i2}(x,t)y_2(t) + ... + K_{ij}(x,t)y_j(t))dt,
\end{aligned} \tag{2.2}$$

2.1.3 System of Volterra-Fredholm Integro-Differential Equations

A system of Fredholm integro-differential equations appears in the form [35]:

$$\begin{aligned}
y_1^{(n)}(x) = f_1(x) &+ \int_a^b (K_{11}(x,t)y_1(t) + K_{12}(x,t)y_2(t) + \cdots + K_{1j}(x,t)y_j(t))dt\\
&+ \int_a^t (Kv_{11}(x,t)y_1(t) + Kv_{12}(x,t)y_2(t) + \cdots + Kv_{1j}(x,t)y_j(t))dt,
\end{aligned}$$

$$y_2^{(n)}(x) = f_2(x) + \int_a^b (K_{21}(x,t)y_1(t) + K_{22}(x,t)y_2(t) + \cdots + K_{2j}(x,t)y_j(t))dt$$
$$+ \int_a^t (K_{21}(x,t)y_1(t) + K_{22}(x,t)y_2(t) + \cdots + K_{2j}(x,t)y_j(t))dt,$$

$$\vdots$$

$$y_i^{(n)}(x) = f_i(x) + \int_a^b (K_{i1}(x,t)y_1(t) + K_{i2}(x,t)y_2(t) + \cdots + K_{ij}(x,t)y_j(t))dt$$
$$+ \int_a^t (K_{i1}(x,t)y_1(t) + K_{i2}(x,t)y_2(t) + \cdots + K_{ij}(x,t)y_j(t))dt,$$

2.2 Linearity[31]

The system of integro-differential equations

$$y_i^{(n)}(x) = f_i(x) + \int_{g(x)}^{h(x)} (\sum_{j=1}^{N} K_{ij}(x,t)y_j(t))dt, 0 \leq i \leq N, \tag{2.3}$$

is linear if the functions $y_j(x)$ and all it's derivatives do not contain nonlinear functions of $y_j(x)$ like $(\sin u,\ \cos u,\ \sinh u,\ \cosh u,\ \exp u, \dots)$.

2.3 Homogeneity[31]

The system of integro-differential equations in equation (2.3) is homogeneous if $f_i(x)$ is identically zero for all i.

2.4 Matrix Relation for polynomials methods to solve the system of Fredholm Integro-Differential equations (FIDE)

Matrix methods is widely used as numerical methods to find augmented matrix of the system like [17, 10, 2, 3, 36].

The following system of linear Fredholm Integro-differential Equations is to be considered:

$$\sum_{k=0}^{n}\sum_{q=1}^{r} \sigma_{p,q}^{k}(x) y_q^{(k)}(x) = f_p(x) + \int_a^b \sum_{q=1}^{r} K_{p,q}(x,t) y_q(t)\mathrm{d}t, \tag{2.4}$$
$$p = 1, 2, \dots, r, \quad 0 \leq a \leq x \leq b,$$

subject to the initial conditions:

$$\sum_{q=0}^{n-1}(a^k_{p,q}y^{(q)}_k(a)+b^k_{p,q}y^{(q)}_k(b))=\mu_{k,p},$$
$$p=0,1,\ldots,n-1, \quad k=1,2,\ldots,r, \tag{2.5}$$

where $y_q(x)$ is the unknown and $\sigma^k_{p,q},\ K_{p,q}(x,t),\ f_p(x)$are the known functions defined in the interval $[a,b]$, the kernel function $K_{p,q}(x,t)$ can be expanded using Maclaurin series and also $a^k_{p,q},\ b^k_{p,q},\ \mu_{k,p}$ are real constants.

Taking $y_p(x)$ as the approximate solution of equation (2.4) in terms of truncated Laguerre series yields

$$y_p(x)=\sum_{n=0}^{N}\alpha_{p,n}\,L_n(x), \qquad p=1,2,\ldots,r, \quad a\le x\le b, \tag{2.6}$$

where $\alpha_{p,n}$ are the unknown Laguerre coefficients, to be determined for $n=0,1,2,\ldots,N$ and $L_n(x)$ be the Laguerre polynomials.

This can be expressed in matrix form as:

$$y_q(x)=\mathbf{L}(x)\,\mathbf{A}_q, \qquad q=1,2,\ldots,r, \tag{2.7}$$

where,

$$\mathbf{A}_q=(\alpha_{q,0},\alpha_{q,1},\ldots,\alpha_{q,N})^{\mathrm{T}}. \tag{2.8}$$

or from equation (2.8)

$$y_q(x)=\mathbf{S}(x)\psi^{\mathrm{T}}\mathbf{A}_q, \qquad q=1,2,\ldots,r. \tag{2.9}$$

Assume that the polynomials $L_n(x)$ has a matrix form given by the following matrix:

$$\psi = \begin{pmatrix} \psi_{1,1} & \psi_{1,2} & \dots & \psi_{1,N-1} & \psi_{1,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \psi_{N,1} & \psi_{N,2} & \dots & \psi_{N,N-1} & \psi_{N,N} \end{pmatrix}. \tag{2.10}$$

$$\mathbf{L}(x) = \mathbf{S}(x)\,\psi^{\mathrm{T}}, \tag{2.11}$$

where;

$$\mathbf{L}(x) = \begin{pmatrix} L_0(x) & L_1(x) & L_2(x) & \dots & L_N(x) \end{pmatrix},$$
$$\mathbf{S}(x) = \begin{pmatrix} 1 & x & x^2 & \dots & x^N \end{pmatrix},$$

The relation between the matrix $\mathbf{S}(x)$ and it's first and second derivatives are:

$$\mathbf{S}^{(2)}(x) = S^{(1)}(x)\beta, \quad \mathbf{S}^{(1)}(x) = S(x)\beta, \quad \mathbf{S}^{(0)}(x) = \mathbf{S}(x), \tag{2.12}$$

where,

$$\beta = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & N \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}. \tag{2.13}$$

It follows from equations (2.11) and (2.12) that

$$\mathbf{S}^{(p)}(x) = \mathbf{S}^{(p-1)}(x)\beta = \mathbf{S}(x)(\beta)^p, \qquad p = 0, 1, 2, \dots, n, \tag{2.14}$$

$$\mathbf{L}^{(p)}(x) = \mathbf{S}^{(p)}(x)\psi^{\mathrm{T}} = \mathbf{S}(x)(\beta)^p\psi^{\mathrm{T}}, \qquad p = 0, 1, 2, \dots, n. \tag{2.15}$$

The following recurrence relation is obtained using equations (2.14) and (2.15)

$$\begin{aligned} y_q^{(p)}(x) &= \mathbf{L}^{(p)}(x)\mathbf{A}_q \\ &= \mathbf{S}^{(p)}(x)\psi^{\mathrm{T}}\mathbf{A}_q \\ &= \mathbf{S}(x)(\beta)^p\psi^{\mathrm{T}}\mathbf{A}_q, \\ p = 0,1,2,\ldots,n,\ q &= 1,2,\ldots,r. \end{aligned} \tag{2.16}$$

Hence, the matrices $\mathbf{y}^{(p)}(x)$ can be expressed as

$$\mathbf{y}^{(p)}(x) = \widetilde{\mathbf{S}}(x)(\widetilde{\beta})^p\widetilde{\psi}\mathbf{A}, \quad p = 0,1,2,\ldots,n, \tag{2.17}$$

where,

$$\mathbf{y^{(p)}}(\mathbf{x}) = \begin{pmatrix} y_1^{(p)}(x) \\ y_2^{(p)}(x) \\ \vdots \\ y_r^{(p)}(x) \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_r \end{pmatrix},$$

$$\widetilde{\mathbf{S}}(x) = \begin{pmatrix} \mathbf{S}(x) & 0 & \ldots & 0 \\ 0 & \mathbf{S}(x) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \mathbf{S}(x) \end{pmatrix}_{r\times r},$$

$$\widetilde{\beta} = \begin{pmatrix} \beta & 0 & \ldots & 0 \\ 0 & \beta & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \beta \end{pmatrix}_{r\times r} \quad \text{and} \quad \widetilde{\psi} = \begin{pmatrix} \psi^{\mathrm{T}} & 0 & \ldots & 0 \\ 0 & \psi^{\mathrm{T}} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \psi^{\mathrm{T}} \end{pmatrix}.$$

The matrix form of the system can be defined in equation (2.4) is as follows

$$\sum_{p=0}^{n} \mathbf{P}_p(x)\mathbf{y}^{(p)}(x) = \mathbf{f}(x) + \mathbf{I}(x), \tag{2.18}$$

where,

$$\mathbf{y}^{(p)}(x) = \begin{pmatrix} y_1^{(p)}(x) \\ y_2^{(p)}(x) \\ \vdots \\ y_r^{(p)}(x) \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_r(x) \end{pmatrix}, \quad \mathbf{I} = \begin{pmatrix} I_1(x) \\ I_2(x) \\ \vdots \\ I_r(x) \end{pmatrix},$$

$$\mathbf{P}_p(x) = \begin{pmatrix} \sigma_{1,1}^p(x) & \sigma_{1,2}^p(x) & \dots & \sigma_{1,r}^p(x) \\ \sigma_{2,1}^p(x) & \sigma_{2,2}^p(x) & \dots & \sigma_{2,r}^p(x) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{r,1}^p(x) & \sigma_{r,2}^p(x) & \dots & \sigma_{r,r}^p(x) \end{pmatrix},$$

$$\mathbf{K}(x,t) = \begin{pmatrix} K_{1,1}(x,t) & K_{1,2}(x,t) & \dots & K_{1,r}(x,t) \\ K_{2,1}(x,t) & K_{2,2}(x,t) & \dots & K_{2,r}(x,t) \\ \vdots & \vdots & \ddots & \vdots \\ K_{r,1}(x,t) & K_{r,2}(x,t) & \dots & K_{r,r}(x,t) \end{pmatrix}, \mathbf{I}(x) = \int_a^b \mathbf{K}(x,t)u(t)\mathrm{d}t,$$

$$I_p(x) = \int_a^b \sum_{q=1}^{r} K_{p,q}(x,t)y_q(t). \tag{2.19}$$

Using truncated Taylor and Laguerre/Bessel series to approximate the kernel $K_{p,q}(x,t)$ yields

$$\begin{aligned} K_{p,q}(x,t) &= \sum_{m=0}^{N}\sum_{n=0}^{N} k_{pq}^t s^m t^n \\ K_{p,q}(x,t) &= \sum_{m=0}^{N}\sum_{n=0}^{N} k_{pq}^l L_m(x)L_n(t), \end{aligned} \tag{2.20}$$

where,

$$k_{pq}^t = \frac{1}{m!n!}\frac{\partial^{m+n}K(0,0)}{\partial s^m \partial t^n}, \quad m,n = 0,1,2,\dots,N, \quad p,q = 0,1,2,\dots,r. \tag{2.21}$$

Equation (2.20) can be put into the matrix form, as

$$K_{p,q}(x,t) = \mathbf{S}(x)\mathbf{K}_{pq}^t\mathbf{S}^{\mathrm{T}}(t), \quad \mathbf{K}_{pq}^t = \left(k_{pq}^t\right), \tag{2.22}$$

and

$$K_{p,q}(x,t) = \mathbf{L}(x)\mathbf{K}^l_{pq}\mathbf{L}^{\mathrm{T}}(t), \quad \mathbf{K}^l_{pq} = \left(k^l_{pq}\right). \tag{2.23}$$

Substituting equations (2.16) and (2.23) in equation (2.19) to get the matrix form of the integral part yields

$$\begin{aligned} I_p(x) &= \int_a^b \sum_{q=0}^{r} \mathbf{L}(x)\mathbf{K}^l_{pq}\mathbf{L}^{\mathrm{T}}(t)\mathbf{S}(t)\psi^{\mathrm{T}}\mathbf{A}_q \mathrm{d}t \\ &= \sum_{q=0}^{r} \int_a^b \mathbf{L}(x)\mathbf{K}^l_{pq}\mathbf{L}^(T)(t)\mathbf{S}(t)\psi^{\mathrm{T}}\mathbf{A}_q \mathrm{d}t \\ &= \sum_{q=0}^{r} \mathbf{L}(x)\mathbf{K}^l_{pq}\mathbf{Q}\mathbf{A}_q, \end{aligned}$$

where,

$$\begin{aligned} \mathbf{Q} &= \int_a^b \mathbf{L}^{\mathrm{T}}(t)\mathbf{S}(t)\psi^{\mathrm{T}}\mathrm{d}t \\ &= \int_a^b \psi\mathbf{S}^{\mathrm{T}}(t)\mathbf{S}(t)\psi^{\mathrm{T}}\mathrm{d}t = \psi\mathbf{M}\psi^{\mathrm{T}}, \end{aligned}$$

$$\mathbf{M} = \int_a^b \mathbf{S}^{\mathrm{T}}(t)\mathbf{S}(t)\mathrm{d}t = (m_{ij}), \tag{2.24}$$

where $m_{ij} = \frac{b^{i+j+1} - a^{i+j+1}}{i+j+1}, \qquad i,j = 0,1,2,\ldots,N.$

$$I_p(x) = \sum_{q=0}^{r} \mathbf{S}(x)\psi^{\mathrm{T}}\mathbf{K}^l_{pq}\mathbf{Q}\mathbf{A}_q. \tag{2.25}$$

In order to determine the system of equations in matrix form, we substitute the collocation points into equation (2.18) to equation (2.27) defined by

$$x_t = a + \frac{b-a}{N}t, \quad t = 0,1,2,\ldots,N, \tag{2.26}$$

leads to

$$\sum_{p=0}^{n} \mathbf{P}_p(x_t)\mathbf{y}^{(p)}(x_t) = \mathbf{f}(x_t) + \mathbf{I}(x_t) \tag{2.27}$$

or

$$\sum_{p=0}^{n} \mathbf{P}_p \mathbf{Y}^{(p)} = \mathbf{F} + \mathbf{I}, \tag{2.28}$$

where

$$\mathbf{P}_p = \begin{pmatrix} \mathbf{P}_p(x_0) & 0 & \dots & 0 \\ 0 & \mathbf{P}_p(x_1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{P}_p(x_N) \end{pmatrix},$$
$$\mathbf{Y}^{(p)} = \begin{pmatrix} \mathbf{y}^{(p)}(x_0) \\ \mathbf{y}^{(p)}(x_1) \\ \vdots \\ \mathbf{y}^{(p)}(x_N) \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \mathbf{f}(x_0) \\ \mathbf{f}(x_1) \\ \vdots \\ \mathbf{f}(x_N) \end{pmatrix} \text{ and } \quad \mathbf{I} = \begin{pmatrix} \mathbf{I}(x_0) \\ \mathbf{I}(x_1) \\ \vdots \\ \mathbf{I}(x_N) \end{pmatrix}. \tag{2.29}$$

Using the collocation points in equation (2.17) yields

$$\begin{gathered} \mathbf{y}^{(p)}(x_t) = \widetilde{\mathbf{S}}(x_t)(\widetilde{\beta})^p \widetilde{\psi} \mathbf{A}, \quad t = 0, 1, 2, \dots, N, \\ p = 0, 1, 2, \dots, r, \end{gathered} \tag{2.30}$$

which can be written as

$$\mathbf{y}^{(p)} = \mathbf{S}(\widetilde{\beta})^p \widetilde{\psi} \mathbf{A}, \tag{2.31}$$

where

$$\mathbf{S} = \begin{pmatrix} \widetilde{\mathbf{S}}(x_0) \\ \widetilde{\mathbf{S}}(x_1) \\ \vdots \\ \widetilde{\mathbf{S}}(x_N) \end{pmatrix}, \quad \widetilde{\mathbf{S}}(x_t) = \begin{pmatrix} \mathbf{S}(x_t) & 0 & \dots & 0 \\ 0 & \mathbf{S}(x_t) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{S}(x_t) \end{pmatrix}_{r \times r} \tag{2.32}$$

where;

$$t = 0, 1, 2, \dots, N,$$
$$p = 0, 1, 2, \dots, r.$$

Substituting the collocation points defined in equation (2.25) leads to

$$I_p(x_t) = \sum_{q=0}^{r} \mathbf{S}(x_t)\psi^{\mathrm{T}}\mathbf{K}^l_{pq}\mathbf{Q}\mathbf{A}_q, \quad t = 0, 1, 2, \dots, N, \qquad p = 0, 1, 2, \dots, r. \tag{2.33}$$

Similarly, substituting the collocation points into the matrix $\mathbf{I}(x)$ of equation (2.18) and using equation (2.33) yields

$$\mathbf{I}(x_t) = \widetilde{\mathbf{S}}(x_t)\widetilde{\psi}\mathbf{K}_l\widetilde{\mathbf{Q}}\mathbf{A}, \qquad t = 0, 1, 2, \dots, N, \qquad p = 0, 1, 2, \dots, r, \tag{2.34}$$

where

$$\mathbf{I}(x_t) = \begin{pmatrix} I_1(x_t) \\ I_2(x_t) \\ \vdots \\ I_r(x_t) \end{pmatrix}, \quad \widetilde{\mathbf{S}}(x_t) = \begin{pmatrix} \mathbf{S}(x_t) & 0 & \dots & 0 \\ 0 & \mathbf{S}(x_t) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{S}(x_t) \end{pmatrix}_{r\times r},$$

$$\widetilde{\psi} = \begin{pmatrix} \psi^{\mathrm{T}} & 0 & \dots & 0 \\ 0 & \psi^{\mathrm{T}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \psi^{\mathrm{T}} \end{pmatrix}, \mathbf{K}_l = \begin{pmatrix} \mathbf{K}^l_{11} & \mathbf{K}^l_{12} & \dots & \mathbf{K}^l_{1r} \\ \mathbf{K}^l_{21} & \mathbf{K}^l_{22} & \dots & \mathbf{K}^l_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}^l_{r1} & \mathbf{K}^l_{r2} & \dots & \mathbf{K}^l_{rr} \end{pmatrix},$$

$$\widetilde{\mathbf{Q}} = \begin{pmatrix} \mathbf{Q} & 0 & \dots & 0 \\ 0 & \mathbf{Q} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{Q} \end{pmatrix} \quad \text{and} \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_r \end{pmatrix}$$

Thus, substituting equation (2.34) in $\mathbf{I}$ into equation (2.28) gives

$$\mathbf{I} = \begin{pmatrix} \mathbf{I}(x_0) \\ \mathbf{I}(x_1) \\ \vdots \\ \mathbf{I}(x_N) \end{pmatrix} = \mathbf{S}\widetilde{\psi}\mathbf{K}_l\widetilde{\mathbf{Q}}\mathbf{A}. \tag{2.35}$$

By substituting equations (2.30) and (2.35) into the fundamental matrix equation (2.18) of the system (2.4) gives:

$$\left(\sum_{p=0}^{n} \mathbf{P}_p \mathbf{S}(\widetilde{\beta})^p \widetilde{\psi} - \mathbf{S}\widetilde{\psi}\mathbf{K}_l \widetilde{\mathbf{Q}}\right) \mathbf{A} = \mathbf{F}. \tag{2.36}$$

$r(N+1) \times r(N+1)$ and $r(N+1) \times 1$ are the dimensions of the respective matrices $\mathbf{P}_p$, $\mathbf{S}$, $(\widetilde{\beta})^p$, $\widetilde{\psi}$, $\mathbf{K}_l$, $\widetilde{\mathbf{Q}}$ and $\mathbf{A}, \mathbf{F}$.

Moreover, matrix equation (2.36) is written as

$$\mathbf{W}\mathbf{A} = \mathbf{F}, \quad \text{or} \quad (\mathbf{W}; \mathbf{F}), \tag{2.37}$$

where

$$\mathbf{W} = (w_{m,n}) = \sum_{p=0}^{n} \mathbf{P}_p \mathbf{S}(\widetilde{\beta})^p \widetilde{\psi} - \mathbf{S}\widetilde{\psi}\mathbf{K}_l \widetilde{\mathbf{Q}}, \;\; m, n = 1, 2, \ldots, r(N+1). \tag{2.38}$$

The conditions defined in equation (2.5) can be expressed in the following matrix form:

$$\begin{aligned}
&\sum_{q=0}^{n-1} \left[a^1_{p,q} y_1^{(q)}(a) + b^1_{p,q} y_1^{(q)}(b)\right] = \mu_{1,p}, \\
&\sum_{q=0}^{n-1} \left[a^2_{p,q} y_2^{(q)}(a) + b^2_{p,q} y_2^{(q)}(b)\right] = \mu_{2,p}, \\
&\qquad\qquad\qquad\qquad \vdots \\
&\sum_{q=0}^{n-1} \left[a^r_{p,q} y_r^{(q)}(a) + b^r_{p,q} y_r^{(q)}(b)\right] = \mu_{r,p}
\end{aligned} \tag{2.39}$$

or

$$\sum_{q=0}^{n-1}\left[a_q^1 y_1^{(q)}(a)+b_q^1 y_1^{(q)}(b)\right]=\mu_1, \tag{2.40}$$

$$\sum_{q=0}^{n-1}\left[a_q^2 y_2^{(q)}(a)+b_q^2 y_2^{(q)}(b)\right]=\mu_2, \tag{2.41}$$

$$\vdots \tag{2.42}$$

$$\sum_{q=0}^{n-1}\left[a_q^r y_r^{(q)}(a)+b_q^r y_r^{(q)}(b)\right]=\mu_r, \tag{2.43}$$

briefly;

$$\sum_{q=0}^{n-1}\left[a_q y^{(q)}(a)+b_q y^{(q)}(b)\right]=\mu, \tag{2.44}$$

$$\mu_p=\begin{pmatrix}\mu_{p,0}\\ \mu_{p,1}\\ \vdots\\ \mu_{p,n-1}\end{pmatrix}_{n\times 1},\quad a_q^p=\begin{pmatrix}a_{0,q}^p\\ a_{1,q}^p\\ \vdots\\ a_{n-1,q}^p\end{pmatrix}_{n\times 1},\quad b_q^p=\begin{pmatrix}b_{0,q}^p\\ b_{1,q}^p\\ \vdots\\ b_{n-1,q}^p\end{pmatrix}_{n\times 1},$$

$$p=1,2,\dots,r.$$

where

$$a_q=\begin{pmatrix}a_q^1 & 0 & \dots & 0\\ 0 & a_q^2 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & a_q^r\end{pmatrix}_{r\times r},\, b_q=\begin{pmatrix}b_q^1 & 0 & \dots & 0\\ 0 & b_q^2 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & b_q^r\end{pmatrix}_{r\times r},$$

$$\mu=\begin{pmatrix}\mu_1\\ \mu_2\\ \vdots\\ \mu_r\end{pmatrix}_{r\times 1}.$$

Substitute the values of $y^{(q)}(a)$ and $y^{(q)}(b)$ from equation (2.17) into equation (2.39) yields:

$$\sum_{q=0}^{n-1}\left[a_q\widetilde{\mathbf{S}}(a)+b_q\widetilde{\mathbf{S}}(b)\right](\widetilde{\beta})^q\widetilde{\psi}\mathbf{A}=\mu. \tag{2.45}$$

or briefly

$$\mathbf{VA}=\mu \text{ or } (\mathbf{V};\mu), \tag{2.46}$$

where

$$\mathbf{V}=\sum_{q=0}^{n-1}\left[a_q\widetilde{\mathbf{S}}(a)+b_q\widetilde{\mathbf{S}}(b)\right](\widetilde{\beta})^q\widetilde{\psi} \tag{2.47}$$

The matrix form of the conditions of equation (2.46) defined in equation (2.4). By replacing the augmented matrix in equation (2.46) with the augmented matrix in equation (2.37) yields:

$$\tilde{\mathbf{W}}\mathbf{A}=\tilde{\mathbf{F}}. \tag{2.48}$$

The augmented matrix of system (2.48) can be obtained by replacing the n-rows of the matrix $\mathbf{W}$ as follows:

$$(\tilde{\mathbf{W}};\tilde{\mathbf{F}})=\begin{pmatrix}
w_{1,1} & w_{1,2} & \cdots & w_{1,r(N+1)} & | & f_1(x_0)\\
\vdots & \vdots & \ddots & \vdots & | & \vdots\\
w_{r,1} & w_{r,2} & \cdots & w_{r,r(N+1)} & | & f_r(x_0)\\
\vdots & \vdots & \ddots & \vdots & | & \vdots\\
w_{r(N-n+1),1} & w_{r(N-n+1),2} & \cdots & w_{r(N-n+1),r(N+1)} & | & f_r(x_{N-n})\\
v_{1,1} & v_{1,2} & \cdots & v_{1,r(N+1)} & | & \mu_{1,0}\\
\vdots & \vdots & \ddots & \vdots & | & \vdots\\
v_{r,1} & v_{r,2} & \vdots & v_{r,r(N+1)} & | & \mu_{1,n-1}\\
\vdots & \vdots & \ddots & \vdots & | & \vdots\\
v_{nr,1} & v_{nr,2} & \cdots & v_{nr,r(N+1)} & | & \mu_{r,n-1}
\end{pmatrix}. \tag{2.49}$$

If $|\tilde{W}|\neq 0$, the solution must be found; else, the solution does not exist.

2.5 Error analysis

Equation (2.6) is the approximate solution of the system defined in equation (2.4). So, when substituting $y_q(x)$ by $y_{p,N}(x)$, $p = 1, 2, \dots$,r in equation (2.4), the absolute error for $x = x_i \in [a,\, b]$, $i = 0, 1, 2, \dots$ can be expressed as:

$$E_p(x_i) = \left| \sum_{k=0}^{n} \sum_{q=1}^{r} \sigma_{p,q}^{k}(x_i) y_{p,N}^{(k)}(x_i) - f_p(x_i) - \int_a^b \sum_{q=1}^{r} K_{p,q}(x_i, t) y_{p,N}(t) \mathrm{d}t \right| \cong 0,$$

$$p = 1, 2, \dots, r$$

$E_p(x_i) \leq 10^{-k_i}$, where k_i is a non negative integer.
For max $10^{-k_i} \neq 10^{-k}$, the error can be evaluate by the function below:

$$E_{p,N}(x) = \sum_{k=0}^{n} \sum_{q=1}^{r} \sigma_{p,q}^{k}(x) y_q^{(k)}(x) - f_p(x) - \int_a^b \sum_{q=1}^{r} K_{p,q}(x, t) y_q(t) \mathrm{d}t \cong 0 \qquad (2.50)$$

$p = 1, 2, \dots, r$
The error will be decreasing, if $E_{p,N}(x) \to 0$, for enough large N.

Chapter Three
Numerical Techniques For Solving Systems of Fredholm Integro-Differential Equations(FIDEs)

3.1 Laguerre polynomials method

The main objective is to show how to use the Laguerre matrix method for solving the system of linear FIDEs.

Taking $y_p(x)$ as the approximate solution of system (2.4) in terms of truncated Laguerre series yields

$$y_p(x) = \sum_{n=0}^{N} \alpha_{p,n} L_n(x), \quad p = 1, 2, \ldots, r, \quad a \leq x \leq b, \tag{3.1}$$

where $\alpha_{p,n}$ are the unknown Laguerre coefficients to be determined for $n = 0, 1, 2, \ldots, N$ and $L_n(x)$ is the Laguerre polynomials defined by the relation defined in equation (3.1). These Laguerre polynomials $L_n(x)$ has a matrix form given by:

$$\mathbf{L}(x) = \mathbf{S}(x)\,\psi^{\mathrm{T}}, \tag{3.2}$$

where;

$\mathbf{L}(x) = \begin{pmatrix} L_0(x) & L_1(x) & L_2(x) & \ldots & L_N(x) \end{pmatrix},$

$\mathbf{S}(x) = \begin{pmatrix} 1 & x & x^2 & \ldots & x^N \end{pmatrix},$

$$y_q(x) = \mathbf{S}(x)\psi^{\mathrm{T}}\mathbf{A}_q, \quad q = 1, 2, \ldots, r. \tag{3.3}$$

The explicit form of these functions is[11, 22, 8]:

$$L_n(x) = \sum_{k=0}^{n} \frac{(-1)^k}{k!} \begin{pmatrix} n \\ n-k \end{pmatrix} x^k, \quad 0 \leq x \leq b < \infty. \tag{3.4}$$

These polynomials satisfy the following recurrence relation:

$L_0(x) = 1, \;\; L_1(x) = 1 - x,$

$$L_{n+1}(x) = \frac{1}{n+1}[(2n+1-x)L_n - (n)L_{n-1}(x)], \; n = 1, 2, \ldots \tag{3.5}$$

This can be expressed in matrix form as: The polynomials is defined in equation (3.4). This can be expressed in matrix form as

$$y_q(x) = \mathbf{L}(x)\,\mathbf{A}_q, \quad q = 1, 2, \ldots, r, \tag{3.6}$$

where

$$\mathbf{A}_q = (\alpha_{q,0}, \alpha_{q,1}, \ldots, \alpha_{q,N})^{\mathrm{T}}.$$

$$\psi = \begin{pmatrix} \frac{(-1)^0}{0!}\binom{0}{0} & 0 & 0 & \cdots & 0 \\ \frac{(-1)^0}{0!}\binom{1}{0} & \frac{(-1)^1}{1!}\binom{1}{1} & 0 & \cdots & 0 \\ \frac{(-1)^0}{0!}\binom{2}{0} & \frac{(-1)^1}{1!}\binom{2}{1} & \frac{(-1)^2}{2!}\binom{2}{2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{(-1)^0}{0!}\binom{N}{0} & \frac{(-1)^1}{1!}\binom{N}{1} & \frac{(-1)^2}{2!}\binom{N}{2} & \cdots & \frac{(-1)^N}{N!}\binom{N}{N} \end{pmatrix}. \tag{3.7}$$

The Laguerre polynomials have the following properties:

- $L_q(x).L_p(x) = \int_0^\infty e^{-x} L_p(x) L_q(x) = 0, \;\; where\; p \neq q$
- $||L_q(x)|| = 1, \;\; q = 0, 1, 2, \ldots$

3.2 Bessel functions method

The Bessel matrix method is used for solving the system of linear Fredholm IDEs[38]. Taking $y_p(x)$ as the approximate solution of system (2.4) in terms of truncated Bessel series yields

$$y_p(x) = \sum_{n=0}^{N} \alpha_{p,n}\, \Lambda_n(x), \quad p = 1, 2, \ldots, r, \quad a \le x \le b, \tag{3.8}$$

where $\alpha_{p,n}$ are the unknown Bessel coefficients.

$n = 0, 1, 2, \ldots, N$, and $\Lambda_n(x)$ be the Bessel functions.

These Bessel functions $\Lambda_n(x)$ has a matrix form given by equation (3.2):

$$\mathbf{\Lambda}(x) = \mathbf{S}(x)\, \psi^{\mathrm{T}}, \tag{3.9}$$

where;

$$\mathbf{\Lambda}(x) = \begin{pmatrix} \Lambda_0(x) & \Lambda_1(x) & \Lambda_2(x) & \ldots & \Lambda_N(x) \end{pmatrix},$$

$$\mathbf{S}(x) = \begin{pmatrix} 1 & x & x^2 & \ldots & x^N \end{pmatrix},$$

or from equation (3.12)

$$y_q(x) = \mathbf{S}(x)\psi^{\mathrm{T}}\mathbf{A}_q, \quad q = 1, 2, \ldots, r. \tag{3.10}$$

The explicit from of these functions is in the from

$$\Lambda_n(x) = \sum_{k=0}^{[\frac{N-n}{2}]} \frac{(-1)^k}{k!(k+n)!} \left(\frac{x}{2}\right)^{2k+n}, \qquad \begin{matrix} 0 \le x \le \infty \\ n = 1, 2, 3 \ldots. \end{matrix} \tag{3.11}$$

The polynomials is defined in equation (3.8). This can be expressed in matrix form as:

$$y_q(x) = \mathbf{\Lambda}(x)\, \mathbf{A}_q, \quad q = 1, 2, \ldots, r, \tag{3.12}$$

where,

$$\mathbf{A}_q = (\alpha_{q,0}, \alpha_{q,1}, \ldots, \alpha_{q,N})^{\mathrm{T}}.$$

When N is odd;

$$\psi = \begin{pmatrix} \frac{1}{0!0!2^0} & 0 & \frac{-1}{1!1!2^2} & \cdots & \frac{(-1)^{\frac{N-1}{2}}}{(\frac{N-1}{2})!(\frac{N-1}{2})!2^{N-1}} & 0 \\ 0 & \frac{1}{0!1!2^1} & 0 & \cdots & 0 & \frac{(-1)^{\frac{N-1}{2}}}{(\frac{N-1}{2})!(\frac{N-1}{2})!2^{N}} \\ 0 & 0 & \frac{1}{0!2!2^2} & \cdots & \frac{(-1)^{\frac{N-1}{2}}}{(\frac{N-1}{2})!(\frac{N-1}{2})!2^{N-1}} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & \frac{(1)}{(N-1)(N-1)!2^{N-1}} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{(1)}{0!N!2^N} \\ & & & & & \end{pmatrix}_{(N+1)\times(N+1)} \tag{3.13}$$

When N is even;

$$\psi = \begin{pmatrix} \frac{1}{0!0!2^0} & 0 & \frac{-1}{1!1!2^2} & \cdots & 0 & \frac{(-1)^{\frac{N-1}{2}}}{(\frac{N-1}{2})!(\frac{N-1}{2})!2^{N}} \\ 0 & \frac{1}{0!1!2^1} & 0 & \cdots & 0 & \frac{(-1)^{\frac{N-1}{2}}}{(\frac{N-1}{2})!(\frac{N-1}{2})!2^{N}} \\ 0 & 0 & \frac{1}{0!2!2^2} & \cdots & 0 & \frac{(-1)^{\frac{N-1}{2}}}{(\frac{N-1}{2})!(\frac{N-1}{2})!2^{N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & \frac{(1)}{(N-1)(N-1)!2^{N-1}} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{(1)}{0!N!2^N} \\ & & & & & \end{pmatrix}_{(N+1)\times(N+1)} \tag{3.14}$$

3.3 Legendre polynomials method[37, 6, 12]

Going back to system (2.4), we implement the matrix Legendre polynomials for solving system of FIDE.

Since $y_p(x)$ is the approximate solution of system (2.4), then the truncated Legendre series yields

$$y_p(x) = \sum_{n=0}^{N} \alpha_{p,n}\, G_n(x), \quad p = 1, 2, \ldots, r, \quad a \leq x \leq b, \tag{3.15}$$

where $\alpha_{p,n}$ are the unknown Legendre coefficients.

$n = 0, 1, 2, \ldots, N$, and $G_n(x)$ be the Legendre polynomial.

These Legendre polynomials $G_n(x)$ has a matrix form given by:

$$\mathbf{G}(x) = \mathbf{S}(x)\,\psi^{\mathrm{T}}, \tag{3.16}$$

where;

$$\mathbf{G}(x) = \begin{pmatrix} G_0(x) & G_1(x) & G_2(x) & \dots & G_n(x) \end{pmatrix},$$

$$\mathbf{S}(x) = \begin{pmatrix} 1 & x & x^2 & \dots & x^N \end{pmatrix},$$

or from equation (3.19)

$$y_q(x) = \mathbf{S}(x)\psi^{\mathrm{T}}\mathbf{A}_q, \quad q = 1, 2, \dots, r. \tag{3.17}$$

The explicit from of these polynomials is in the from:

$$G_n(x) = \frac{1}{2^n}\sum_{k=0}^{[n/2]}(-1)^k \begin{pmatrix} n \\ k \end{pmatrix} \begin{pmatrix} 2n-2k \\ n \end{pmatrix} x^{n-2k}, n = 0, 1, \dots, \tag{3.18}$$

where;

$$\left[\frac{n}{2}\right] = \left\{ \begin{array}{l} \frac{n}{2}, \dots \; n\ even \\ \frac{n-1}{2}, \dots \; n\ odd \end{array} \right\}$$

The polynomials is defined in equation (3.18). This can be expressed in matrix form as

$$y_q(x) = \mathbf{G}(x)\,\mathbf{A}_q, \quad q = 1, 2, \dots, r, \tag{3.19}$$

where,

$\mathbf{A}_q = (\alpha_{q,0}, \alpha_{q,1}, \dots, \alpha_{q,N})^{\mathrm{T}}$.

When N is odd;

$$\psi = \begin{pmatrix}
\frac{(-1)^0}{2^0}\binom{0}{0}\binom{0}{0} & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & \frac{(-1)^0}{2^1}\binom{1}{0}\binom{2}{1} & 0 & 0 & \dots & 0 & 0 \\
\frac{(-1)^1}{2^1}\binom{2}{1}\binom{2}{2} & 0 & \frac{(-1)^0}{2^2}\binom{2}{0}\binom{4}{2} & 0 & \dots & 0 & 0 \\
0 & \frac{(-1)^1}{2^3}\binom{3}{1}\binom{4}{3} & 0 & \frac{(-1)^0}{2^3}\binom{3}{0}\binom{6}{3} & \dots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \\
0 & \frac{(-1)^{N-2/2}}{2^{N-1}}\binom{N-1}{\frac{N-1}{2}}\binom{N}{N-1} & 0 & \frac{(-1)^{N-4/2}}{2^{N-1}}\binom{N-1}{\frac{N-4}{2}}\binom{N+2}{N-1} & \dots & \frac{(-1)^0}{2^{N-1}}\binom{N-1}{0}\binom{2N-2}{N-1} & 0 \\
\frac{(-1)^{N/2}}{2^{N-1}}\binom{N}{\frac{N}{2}}\binom{N}{N} & 0 & \frac{(-1)^{N-2/2}}{2^{N}}\binom{N}{\frac{N-2}{2}}\binom{N+2}{N} & 0 & \dots & 0 & \frac{(-1)^0}{2^N}\binom{N}{0}\binom{2N}{N}
\end{pmatrix} \tag{3.20}$$

When N is even;

$$\psi = \left(\begin{array}{ccc}
\frac{(-1)^0}{2^0}\begin{pmatrix}0\\0\end{pmatrix}\begin{pmatrix}0\\0\end{pmatrix} & 0 & 0 \\
0 & \frac{(-1)^0}{2^1}\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}2\\1\end{pmatrix} & 0 \\
\frac{(-1)^1}{2^1}\begin{pmatrix}2\\1\end{pmatrix}\begin{pmatrix}2\\2\end{pmatrix} & 0 & \frac{(-1)^0}{2^2}\begin{pmatrix}2\\0\end{pmatrix}\begin{pmatrix}4\\2\end{pmatrix} \\
0 & \frac{(-1)^1}{2^3}\begin{pmatrix}3\\1\end{pmatrix}\begin{pmatrix}4\\3\end{pmatrix} & 0 \\
\vdots & \vdots & \vdots \\
0 & \frac{(-1)^{N-1/2}}{2^N}\begin{pmatrix}N\\ \frac{N-1}{2}\end{pmatrix}\begin{pmatrix}N+1\\N\end{pmatrix} & 0
\end{array}\right.$$

$$\left.\begin{array}{cccc}
0 & \dots & 0 & 0 \\
0 & \dots & 0 & 0 \\
0 & \dots & 0 & 0 \\
\frac{(-1)^0}{2^3}\begin{pmatrix}3\\0\end{pmatrix}\begin{pmatrix}6\\3\end{pmatrix} & \dots & 0 & 0 \\
\ddots & \vdots & \vdots & \\
0 & \dots & \frac{(-1)^0}{2^{N-1}}\begin{pmatrix}N-1\\0\end{pmatrix}\begin{pmatrix}2N-2\\N-1\end{pmatrix} & 0 \\
0 & \dots & 0 & \frac{(-1)^0}{2^N}\begin{pmatrix}N\\0\end{pmatrix}\begin{pmatrix}2N\\N\end{pmatrix}
\end{array}\right) \qquad (3.21)$$

Chapter Four

Numerical Examples & Results

4.1 Examples

In this chapter, we will solve some numerical examples of a system of Fredholm integro-differential equations using the Laguerre polynomials method, Bessel functions method and legendre polynomials method.

Example 4.1

Consider the following system of Fredholm integro-differential equation:

$$\begin{aligned} y_1^{(2)}(x) &= \frac{8}{9} + \int_0^1 \left\{ \frac{1}{3} y_1(t) + \frac{1}{4} y_2(t) \right\} \mathrm{d}t \\ y_2^{(2)}(x) &= 6x - \frac{x^2}{18} + \int_0^1 \left\{ \frac{x^2}{6} y_1(t) - \frac{x^2}{3} y_2(t) \right\} \mathrm{d}t, \qquad 0 \le x \le 1, \end{aligned} \tag{4.1}$$

subject to the initial conditions:
$y_1(0) = 0, \quad y_1^{(1)}(0) = \frac{1}{3},$
$y_2(0) = 0, \quad y_2^{(1)}(0) = -\frac{1}{2}.$
System (4.1) has the exact solution:
$y_1(x) = \frac{x^2}{2} + \frac{x}{3}$ and $y_2(x) = x^3 - \frac{x}{2}$.

Solution of system (4.1) using Laguerre polynomials method:

Applying the method of Laguerre polynomials on this example using Matlab in Appendix A, the results of the approximate solution is:
$y_1(t) = 0.5\,x^2 + 0.33333333333333333333559208785934\,x$
$y_2(t) = x^3 - 0.5\,x.$

Table (1) contains both the exact and the numerical solutions of system (4.1).

Table 1
Summary of approximate results of solving Example.1 using Laguerre polynomials.

Solution of $y_1(x)$			
x	y_{exact}	y_{app}	Absolute error$\|y_{exact}$-$y_{app}\|$
0	0	0	0
0.1	3.8333×10^{-02}	3.8333×10^{-02}	2.2588×10^{-22}
0.2	8.6667×10^{-02}	8.6667×10^{-02}	4.5175×10^{-22}
0.3	1.4500×10^{-01}	1.4500×10^{-01}	6.7763×10^{-22}
0.4	2.1333×10^{-01}	2.1333×10^{-01}	9.0350×10^{-22}
0.5	2.9167×10^{-01}	2.9167×10^{-01}	1.1294×10^{-21}
0.6	3.8000×10^{-01}	3.8000×10^{-01}	1.3553×10^{-21}
0.7	4.7833×10^{-01}	4.7833×10^{-01}	1.5811×10^{-21}
0.8	5.8667×10^{-01}	5.8667×10^{-01}	1.8070×10^{-21}
0.9	7.0500×10^{-01}	7.0500×10^{-01}	2.0329×10^{-21}
1.0	8.3333×10^{-01}	8.3333×10^{-01}	2.2588×10^{-21}
Solution of $y_2(x)$			
x	y_{exact}	y_{app}	Absolute error$\|y_{exact}$-$y_{app}\|$
0	0	0	0
0.1	-4.9000×10^{-02}	-4.9000×10^{-02}	0
0.2	-9.2000×10^{-02}	-9.2000×10^{-02}	0
0.3	-1.2300×10^{-01}	-1.2300×10^{-01}	0
0.4	-1.3600×10^{-01}	-1.3600×10^{-01}	0
0.5	-1.2500×10^{-01}	-1.2500×10^{-01}	0
0.6	-8.4000×10^{-02}	-8.4000×10^{-02}	0
0.7	-7.0000×10^{-02}	-7.0000×10^{-02}	0
0.8	1.1200×10^{-01}	1.1200×10^{-01}	0
0.9	2.7900×10^{-01}	2.7900×10^{-01}	0
1.0	5.0000×10^{-01}	5.0000×10^{-01}	0

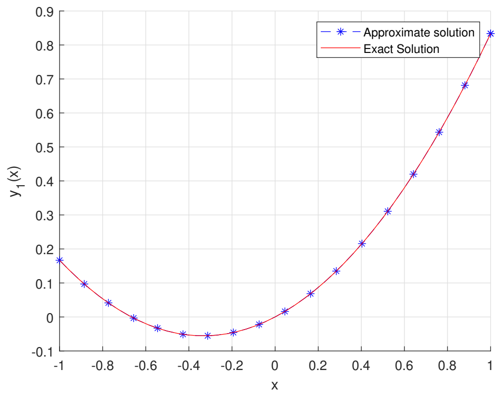
The maximum error corresponding to y_1 and y_2 is $E_1 \simeq 2.2588 \times 10^{-21}$ and $E_2 = 0$ respectively for $x \in [0, 1]$

Figure 1a compares the exact solution $y_1(x)$ and the approximate solution with N=3. Figure 1b shows the error in the approximate solution of $y_1(x)$.

Figure 1c compares the exact solution $y_2(x)$ and the approximate solution with N=3, Figure 1d shows the error in the approximate solution of $y_2(x)$.

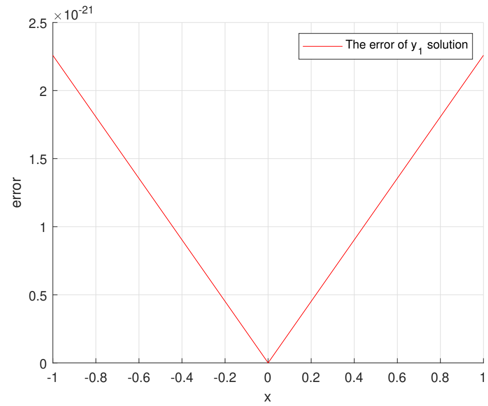
Figure 1

The solution of system (4.1) using Laguerre polynomials.



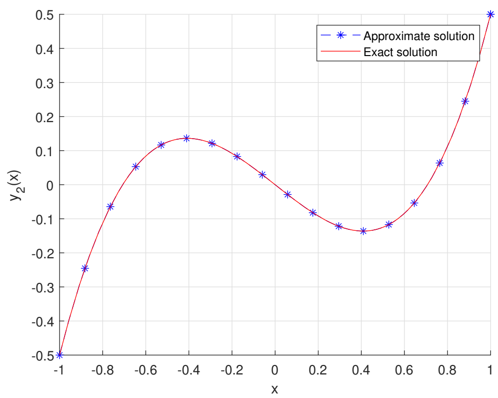
(a)

The approximate and the exact solution of $y_1(x)$



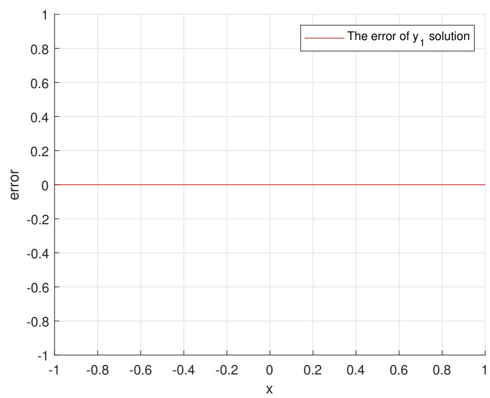
(b)

The absolute error of approximate solution of $y_1(x)$



(c)

The approximate and the exact solution of $y_2(x)$



(d)

The absolute error of approximate solution of $y_2(x)$

Solution of system (4.1) using Bessel functions method:

To solve system (4.1) by the Bessel functions method we use algorithm in Appendix B and MATLAB software to get:

$$y_1(x) = 0.4999999999999999993698074872428\, x^2 + 0.333333333333333333335592087859 34\, x$$

$$y_2(x) = 0.999999999999999999997663711077905\, x^3 - 0.5\, x$$

compared to the exact solution:
$y_1(x) = \frac{x^2}{2} + \frac{x}{3}$ and $y_2(x) = x^3 - \frac{x}{2}$.
Table (2) contains both the exact and the numerical solutions of system (4.2).

Table 2
Summary of approximate results of solving Example.1 using Bessel functions.

		Solution of $y_1(x)$	
x	y_{exact}	y_{app}	Absolute error $\lvert y_{exact} - y_{app}\rvert$
0	0	0	0
0.1	0.038333333333333	0.038333333333333	$6.276698876157836\times10^{-21}$
0.2	0.086666666666667	0.086666666666667	$2.510674035925717\times10^{-20}$
0.3	0.145000000000000	0.145000000000000	$5.649012444929801\times10^{-20}$
0.4	0.213333333333333	0.213333333333333	$1.004268511462803\times10^{-19}$
0.5	0.291666666666667	0.291666666666667	$1.569169204502042\times10^{-19}$
0.6	0.380000000000000	0.380000000000000	$2.259603323610696\times10^{-19}$
0.7	0.478333333333333	0.478333333333333	$3.075570868788764\times10^{-19}$
0.8	0.586666666666667	0.586666666666667	$4.017071840036247\times10^{-19}$
0.9	0.705000000000000	0.705000000000000	$5.084106237353145\times10^{-19}$
1	0.833333333333333	0.833333333333333	$6.276674060739459\times10^{-19}$
		Solution of $y_2(x)$	
x	y_{exact}	y_{app}	Absolute error $\lvert y_{exact} - y_{app}\rvert$
0	0	0	0
0.1	-0.049000000000000	-0.049000000000000	$2.336288922094771\times10^{-23}$
0.2	-0.092000000000000	-0.092000000000000	$1.869031137675817\times10^{-22}$
0.3	-0.123000000000000	-0.123000000000000	$6.307980089655882\times10^{-22}$
0.4	-0.136000000000000	-0.136000000000000	$1.495224910140654\times10^{-21}$
0.5	-0.125000000000000	-0.125000000000000	$2.920361152618464\times10^{-21}$
0.6	-0.084000000000000	-0.084000000000000	$5.046384071724706\times10^{-21}$
0.7	-0.007000000000000	-0.007000000000000	$8.013471002785065\times10^{-21}$
0.8	0.112000000000000	0.112000000000000	$1.196179928112523\times10^{-20}$
0.9	0.279000000000000	0.279000000000000	$1.703154624207088\times10^{-20}$
1	0.500000000000000	0.500000000000000	$2.336288922094771\times10^{-20}$

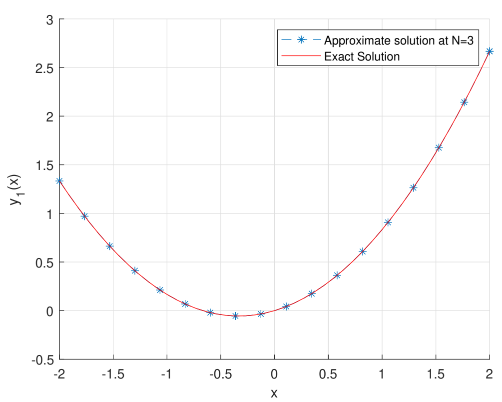
The maximum error corresponding to y_1 and y_2 is $E_1 \simeq 6.276674060739459 \times 10^{-19}$ and $E_2 = 2.336288922094771 \times 10^{-20}$ respectively for $x \in [0, 1]$

Figure 2a compares the exact solution $y_1(x)$ and the approximate solution with N=3. Figure 2b shows the error in the approximate solution of $y_1(x)$.

Figure 2c compares the exact solution $y_2(x)$ and the approximate solution with N=3. Figure 2d shows the error in the approximate solution of $y_2(x)$.

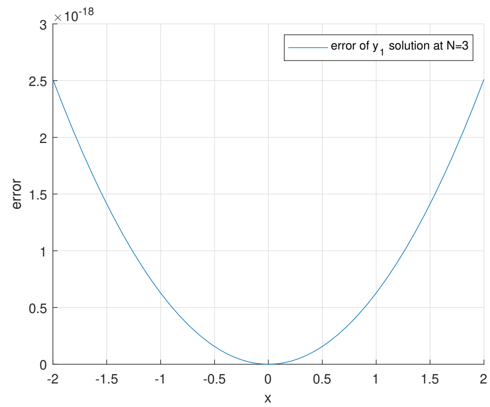
Figure 2

The solution of system (4.1) using Bessel functions method.



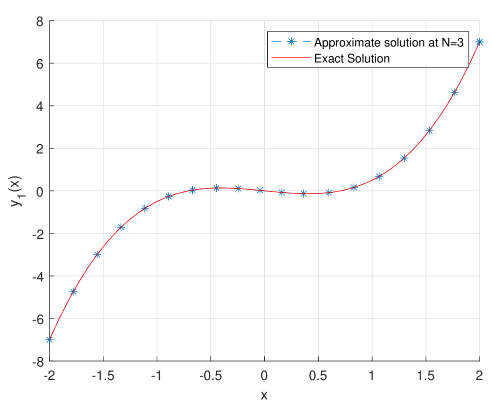
(a)

The approximate and the exact solution of $y_1(x)$



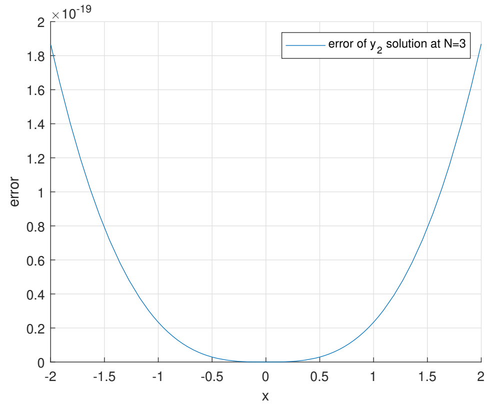
(b)

Absolute Error in the approximate $y_1(t)$ *solution*



(c)

The approximate and the exact solution of $y_2(x)$



(d)

Absolute Error in the approximate $y_2(t)$ *solution*

Solution of system (4.1) using Legendre polynomials method:

To solve system (4.1) by the Legendre polynomials method we use algorithm in Appendix C and MATLAB software to get:

For N=3 is:

$y_1(x) = 0.00000000000000041079865310790170349144676928 14\, x^3$

$+ 0.49999999999992110138772004619 4953\, x^2$

$+ 0.33333333333333333333333305760646\, x$ and

$y_2(x) = 1.0000000000000117770463439506405\, x^3$

$- 0.000000000000001137448959233696035243104400348\, x^2 - 0.5\, x$

Table (3) contains both the exact and the numerical solutions of system (4.1).

Table 3
Summary of approximate results of solving Example.1 using Legendre polynomials.

	Solution of $y_1(x)$		
x	y_{exact}	y_{app}	Absolute error $\|y_{exact} - y_{app}\|$
0	0	0	0
0.1	0.038333333333333	0.038333333333325	$7.885753241491874\times10^{-15}$
0.2	0.086666666666667	0.086666666666635	$3.152658101978804\times10^{-14}$
0.3	0.145000000000000	0.144999999999929	$7.089783541570200\times10^{-14}$
0.4	0.213333333333333	0.213333333333207	$1.259748685100473\times10^{-13}$
0.5	0.291666666666667	0.291666666666470	$1.967330323836375\times10^{-13}$
0.6	0.380000000000000	0.379999999999717	$2.831476791172861\times10^{-13}$
0.7	0.478333333333333	0.478333333332948	$3.851941607918065\times10^{-13}$
0.8	0.586666666666667	0.586666666666164	$5.028478294880125\times10^{-13}$
0.9	0.705000000000000	0.704999999999364	$6.360840372867174\times10^{-13}$
1.0	0.833333333333333	0.833333333332548	$7.848781362687347\times10^{-13}$
	Solution of $y_2(x)$		
x	y_{exact}	y_{app}	Absolute error $\|y_{exact} - y_{app}\|$
0	0	0	0
0.1	-0.049000000000000	-0.049000000000000	$1.019678495794190\times10^{-16}$
0.2	-0.092000000000000	-0.092000000000000	$3.607632129418733\times10^{-16}$
0.3	-0.123000000000000	-0.123000000000001	$7.057238120236591\times10^{-16}$
0.4	-0.136000000000000	-0.136000000000001	$1.066187368761073\times10^{-15}$
0.5	-0.125000000000000	-0.125000000000001	$1.371491605090410\times10^{-15}$
0.6	-0.084000000000000	-0.084000000000002	$1.550974242947967\times10^{-15}$
0.7	-0.007000000000000	-0.007000000000002	$1.533973004270041\times10^{-15}$
0.8	0.112000000000000	0.111999999999999	$1.249825610992927\times10^{-15}$
0.9	0.279000000000000	0.278999999999999	$6.278697850529210\times10^{-16}$
1.0	0.500000000000000	0.500000000000000	$4.025567516136801\times10^{-16}$

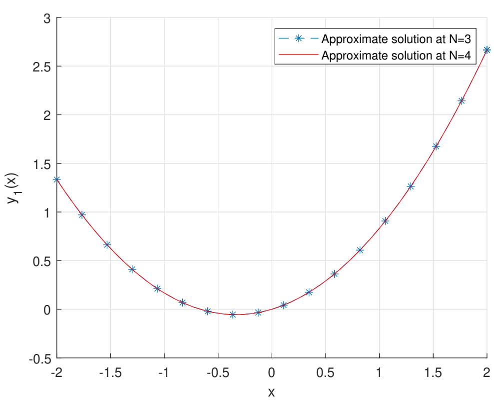
The maximum error corresponding to y_1 and y_2 is $E_1 \simeq 7.848781362687347 \times 10^{-13}$ and $E_2 = 1.550974242947967 \times 10^{-15}$ respectively for $x \in [0, 1]$

Figure 3a compares the exact solution$y_1(x)$ and the approximate solution with N=3. Figure 3b shows the error in the approximate solution of $y_1(x)$.

Figure 3c compares the exact solution $y_2(x)$ and the approximate solution with N=3. Figure 3d shows the error in the approximate solution of $y_2(x)$.

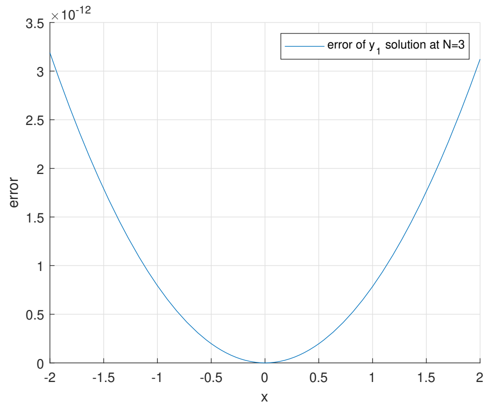
Figure 3

The solution of system (4.1) using Legendre polynomials method.



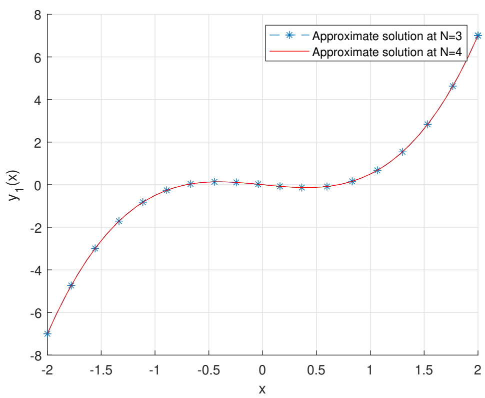
(a)

The approximate and the exact solution of $y_1(x)$



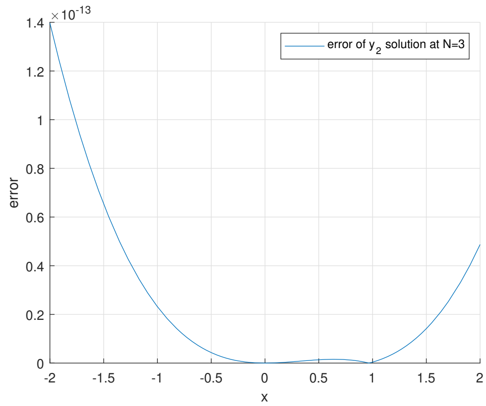
(b)

The absolute error of approximate solution of $y_1(t)$



(c)

The approximate and the exact solution of $y_2(x)$



(d)

The absolute error of approximate solution of $y_2(t)$

Example 4.4

Consider the following system of Fredholm integro-differential equations:

$$y_1^{(2)}(x) - xy_2^{(1)}(x) - y_1(x) = (x-2)\sin(x) + \int_0^1 x\cos(t)y_1(t)\mathrm{d}t - \int_0^1 x\sin(t)y_2(t)\mathrm{d}t$$

$$-2xy_1^{(1)}(x)+y_2^{(2)}(x)+y_2(x) = -2x\cos(x)+\int_0^1 \sin(x)\cos(t)y_1(t)\mathrm{d}t-\int_0^1 \sin(x)\sin(t)y_2(t)dt, \tag{4.2}$$

subject to the initial conditions:

$y_1(0)=0, \quad y_1^{(1)}(0)=1, \quad y_2(0)=1, \quad y_2^{(1)}(0)=0.$

System (4.2) has the exact solution:

$y_1(x)=\sin(x)$ and $y_2(x)=\cos(x)$.

Solution of system (4.2) using Laguerre polynomials method:

Applying the method of Laguerre polynomials on this example using Matlab in Appendix: A, the results are:

when N=3;

$y_1(x) = x - 0.16380480174553502424540825377353\, x^3$

$y_2(x) = 0.026737218335614209524386542795801\, x^3 - 0.5\, x^2 + 1.0$

when N=4;

$y_1(x) = 0.010274735667402386412002307297317\, x^4$

$-\, 0.16992984622837251434891887857948\, x^3 + x$

$y_2(x) = 0.040305681942432110993881693924989\, x^4$

$+\, 0.00067096735562544843422617424648271\, x^3 - 0.5\, x^2 + 1.0$

when N=10;

$y_1(x) = -0.0000001069210430280062317349172451089\, 6\, x^{10}$

$+\, 0.000002955329086047608560438811302410 8\, x^9$

$-\, 0.000000211231320636604118462817458795 89\, x^8$

$-\, 0.00019827416866797536290028492072016\, x^7$

$-\, 0.000000057881749905671456222507011263688\, x^6$

$$+ 0.0083333484824510613128597098743164\, x^5$$

$$- 0.000000002343834213026268001281676899235\,4\, x^4$$

$$- 0.1666666667234521396658692132114\,5\, x^3 + x$$

$$y_2(x) = -0.0000002531436701337305896737885070596\,8\, x^{10}$$

$$- 0.0000000561281717224531007993845841650\,11\, x^9$$

$$+ 0.0000248684064589457221588055553631\,51\, x^8$$

$$- 0.000000046618941449735141043909969751\,585\, x^7$$

$$- 0.0013888686994811417494183453081279\, x^6$$

$$- 0.000000005464419713872182884066134250\,5632\, x^5$$

$$+ 0.041666667513867466847292972496192\, x^4$$

$$- 0.0000000002914596913155317187138493295\,6062\, x^3 - 0.5\, x^2 + 1.0$$

Table (4) contains both the exact and the numerical solutions of system (4.2).

Table 4

Summary of approximate results of solving Example.2 using Laguerre polynomials at N=3, N=10

			Solution of $y_1(x)$		
x	y_{exact}	y_{app} at N=3	Absolute error $\lvert y_{exact} - y_{app}\rvert$	y_{app} at N=10	Absolute error $\lvert y_{exact} - y_{app}\rvert$
0	0	0	0	0	0
0.1	0.099833416646828	0.099836195198254	0.000002778551426	0.099833416646643	$1.856296500973948\times10^{-13}$
0.2	0.198669330795061	0.198689561586036	0.000020230790975	0.198669330793324	$1.736946325588512\times10^{-12}$
0.3	0.295520206661340	0.295577270352871	0.000057063691531	0.295520206655217	$6.122469903536685\times10^{-12}$
0.4	0.389418342308651	0.389516492688286	0.000098150379635	0.389418342293753	$1.489713313597507\times10^{-11}$
0.5	0.479425538604203	0.479524399781808	0.000098861177605	0.479425538574400	$2.980292734531234\times10^{-11}$
0.6	0.564642473395035	0.564618162822964	0.000024310572071	0.564642473342192	$5.284309614497971\times10^{-11}$
0.7	0.644217687237691	0.643814953001281	0.000402734236410	0.644217687151317	$8.637452555189104\times10^{-11}$
0.8	0.717356090899523	0.716131941506286	0.001224149393237	0.717356090766258	$1.332647501963417\times10^{-10}$
0.9	0.783326909627483	0.780586299527505	0.002740610099978	0.783326909431077	$1.964062030520073\times10^{-10}$
1	0.841470984807897	0.836195198254465	0.005275786553432	0.841470984541469	$2.664272960679261\times10^{-10}$
			Solution of $y_2(x)$		
x	y_{exact}	y_{app}	Absolute error $\lvert y_{exact} - y_{app}\rvert$	Absolute error	$\lvert y_{exact} - y_{app}\rvert$
0	1	1	0	1	0
0.1	0.995004165278026	0.995026737218336	0.000022571940310	0.995004165277781	$2.452419905675136\times10^{-13}$
0.2	0.980066577841242	0.980213897746685	0.000147319905443	0.980066577839357	$1.884763260675314\times10^{-12}$
0.3	0.955336489125606	0.955721904895062	0.000385415769456	0.955336489119254	$6.352540068825196\times10^{-12}$
0.4	0.921060994002885	0.921711181973479	0.000650187970594	0.921060993987673	$1.521154640808983\times10^{-11}$
0.5	0.877582561890373	0.878342152291952	0.000759590401579	0.877582561860158	$3.021467033082371\times10^{-11}$
0.6	0.825335614909678	0.825775239160493	0.000439624250814	0.825335614856282	$5.339642865727340\times10^{-11}$
0.7	0.764842187284488	0.764170865889116	0.000671321395373	0.764842187197296	$8.719216890589634\times10^{-11}$
0.8	0.696706709347165	0.693689455787834	0.003017253559331	0.696706709212591	$1.345740550885357\times10^{-10}$
0.9	0.621609968270664	0.614491432166663	0.007118536104002	0.621609968071075	$1.995896879042254\times10^{-10}$
1	0.540302305868140	0.526737218335614	0.013565087532526	0.540302305574183	$2.939571576874493\times10^{-10}$

The maximum error at N=3 corresponding to y_1 and y_2 is $E_1 \simeq 0.005275786553432$ and $E_2 = 0.013565087532526$ respectively for $x \in [0, 1]$.

The maximum error at N=4 corresponding to y_1 and y_2 is $E_1 \simeq 0.001126095368867$ and $E_2 = 6.743434299178420 \times 10^{-4}$ respectively for $x \in [0, 1]$.

The maximum error at N=10 corresponding to y_1 and y_2 is $E_1 \simeq 0.013565087532526$ and $E_2 = 2.939571576874493 \times 10^{-10}$ respectively for $x \in [0, 1]$

Figure 4a compares the exact solution $y_1(x)$ and the approximate solution with N=3,4,10.

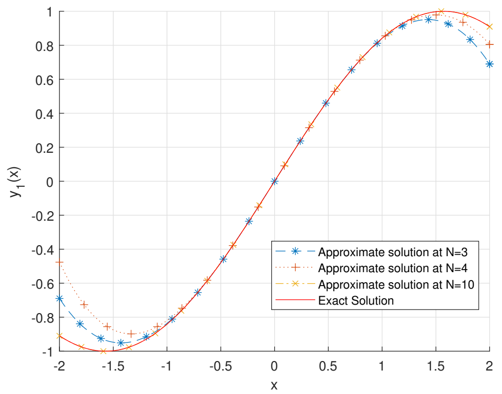
Figure 4b shows the error in the approximate solution of $y_1(x)$.

Figure 4c compares the exact solution $y_2(x)$ and the approximate solution with N=3,4,10.

Figure 4d shows the error in the approximate solution of $y_2(x)$.

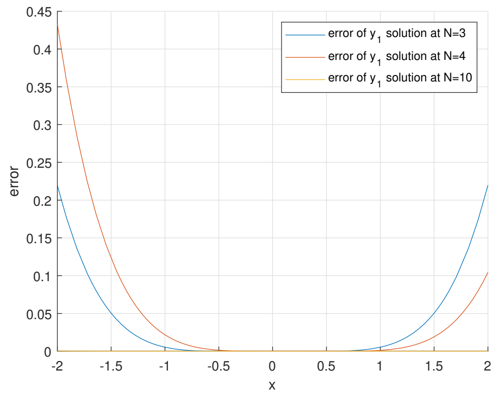
Figure 4

The solution of system (4.2) using Laguerre polynomials method.



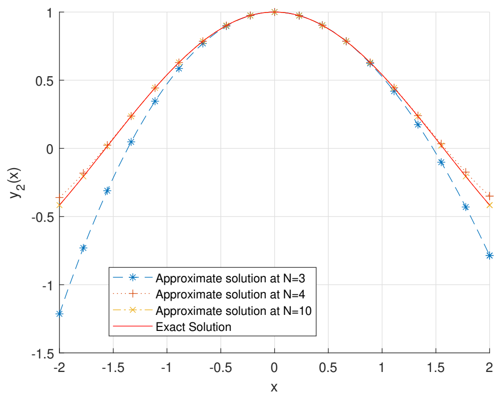
(a)

The approximate and the exact solution of $y_1(x)$



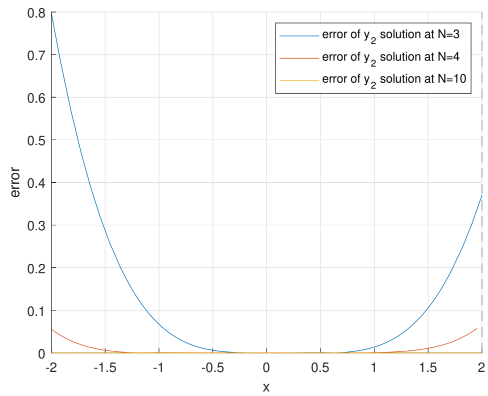
(b)

Absolute error in the approximate solution of $y_1(x)$



(c)

The approximate and the exact solution of $y_2(x)$



(d)

Absolute error in the approximate solution of $y_2(x)$

Solution of system (4.2) using Bessel functions method:

Applying the method of Laguerre polynomials on this example using Matlab software in Appendix: B, the results are:

when N=3;

$y_1(x) = x - 0.16380480174553502427590143987468\, x^3$

$y_2(x) = 0.026737218335614209498552037904545\, x^3 - 0.5\, x^2 + 1.0$

when N=4;

$$\begin{aligned} y_1(x) = {} & 0.010274735667402390567334188290101\, x^4 \\ & - 0.16992984622837250077945106356558\, x^3 + x \end{aligned}$$

$$\begin{aligned} y_2(x) = {} & 0.040305681942432115340854779234059\, x^4 \\ & + 0.00067096735562546190907776207410862\, x^3 - 0.5\, x^2 + 1.0 \end{aligned}$$

When N=10;

$$\begin{aligned} y_1(x) = {} & -0.0000001069210430074063606937308931588 5\, x^{10} \\ & + 0.0000029553290861367048928521889297931\, x^9 \\ & - 0.0000002112313208701014760237392442096 4\, x^8 \\ & - 0.00019827416866787761974178602509 59\, x^7 \\ & - 0.0000000578817500906114164568990934059 44\, x^6 \\ & + 0.00833334848245102001216847646507 26\, x^5 \\ & - 0.0000000023438342146312042381041154822564\, x^4 \\ & - 0.16666666672345215229004825908954\, x^3 + x \end{aligned}$$

$$
\begin{aligned}
y_2(x) = -0.0000002531436701373027708680834675353 7\, x^{10} \\
- 0.00000005612817180892882232608894405090 1\, x^9 \\
+ 0.000024868406458400532747423722114354\, x^8 \\
- 0.000000046618941402204326885926633702935\, x^7 \\
- 0.0013888686994812025349378600000763\, x^6 \\
- 0.0000000054644197140184691213862705559531\, x^5 \\
+ 0.041666667513867462994140095436379\, x^4 \\
- 0.0000000002914597024268139257763306726224 7\, x^3 - 0.5\, x^2 + 1.0
\end{aligned}
$$

Table (5) contains both the exact and the numerical solutions of system (4.2).

Table 5

Summary of approximate results of solving Example.2 using Bessel functions at N =3, N=10

			Solution of $y_1(x)$		
x	y_{exact}	y_{app} at N=3	Absolute error $\|y_{exact} - y_{app}\|$	y_{app} at N=10	Absolute error $\|y_{exact} - y_{app}\|$
0	0	0	0	0	0
0.1	0.099833416646828	0.099836195198254	0.000002778551426	0.099833416646643	$1.856296634724838\times10^{-13}$
0.2	0.198669330795061	0.198689561586036	0.000020230790975	0.198669330793324	$1.736946453501135\times10^{-12}$
0.3	0.295520206661340	0.295577270352871	0.000057063691531	0.295520206655217	$6.122470484639423\times10^{-12}$
0.4	0.389418342308651	0.389516492688286	0.000098150379635	0.389418342293753	$1.489713513280836\times10^{-11}$
0.5	0.479425538604203	0.479524399781808	0.000098861177605	0.479425538574400	$2.980293315832400\times10^{-11}$
0.6	0.564642473395035	0.564618162822964	0.000024310572071	0.564642473342192	$5.284311108313346\times10^{-11}$
0.7	0.644217687237691	0.643814953001281	0.000402734236410	0.644217687151316	$8.637456020057305\times10^{-11}$
0.8	0.717356090899523	0.716131941506286	0.001224149393237	0.717356090766258	$1.332648238375431\times10^{-10}$
0.9	0.783326909627483	0.780586299527505	0.002740610099978	0.783326909431077	$1.964063480425859\times10^{-10}$
1	0.841470984807897	0.836195198254465	0.005275786553432	0.841470984541469	$2.664275625956884\times10^{-10}$
			Solution of $y_2(x)$		
x	y_{exact}	y_{app} at N=3	Absolute error at N=3 $\|y_{exact} - y_{app}\|$	y_{app} at N=10	Absolute error at N=10 $\|y_{exact} - y_{app}\|$
0	1	1	0	1	0
0.1	0.995004165278026	0.995026737218336	0.000022571940310	0.995004165277781	$2.452420021271451\times10^{-13}$
0.2	0.980066577841242	0.980213897746685	0.000147319905443	0.980066577839357	$1.884763360499633\times10^{-12}$
0.3	0.955336489125606	0.955721904895062	0.000385415769456	0.955336489119254	$6.352540471806556\times10^{-12}$
0.4	0.921060994002885	0.921711181973479	0.000650187970594	0.921060993987673	$1.521154777079257\times10^{-11}$
0.5	0.877582561890373	0.878342152291952	0.000759590401579	0.877582561860158	$3.021467484559923\times10^{-11}$
0.6	0.825335614909678	0.825775239160493	0.000439624250814	0.825335614856282	$5.339644312366901\times10^{-11}$
0.7	0.764842187284488	0.764170865889116	0.000671321395373	0.764842187197296	$8.719221192338034\times10^{-11}$
0.8	0.696706709347165	0.693689455787834	0.003017253559331	0.696706709212591	$1.345741718280637\times10^{-10}$
0.9	0.621609968270664	0.614491432166663	0.007118536104002	0.621609968071074	$1.995899776230342\times10^{-10}$
1	0.540302305868140	0.526737218335614	0.013565087532526	0.540302305574182	$2.939578212901901\times10^{-10}$

The maximum error at N=3 corresponding to y_1 and y_2 is $E_1 \simeq 0.005275786553432$ and $E_2 = 0.013565087532526$ respectively for $x \in [0, 1]$.

The maximum error at N=4 corresponding to y_1 and y_2 is $E_1 \simeq 0.001126095368867$ and $E_2 = 6.743434299178599 \times 10^{-4}$ respectively for $x \in [0, 1]$.

The maximum error at N=10 corresponding to y_1 and y_2 is $E_1 \simeq 2.664275625956884 \times 10^{-10}$ and $E_2 = 2.939578212901901 \times 10^{-10}$ respectively for $x \in [0, 1]$.

These approximate solutions are illustrated as follows;

Figure 5a compares the exact solution $y_1(x)$ and the approximate solution with N=3,4,10.

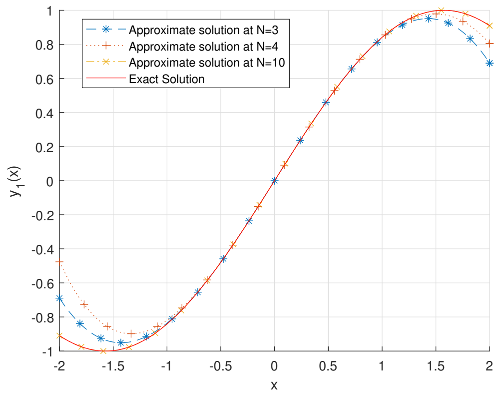
Figure 5b shows the error in the approximate solution of $y_1(x)$.

Figure 5c compares the exact solution $y_2(x)$ and the approximate solution with N=3,4,10.

Figure 5d shows the error in the approximate solution of $y_2(x)$.

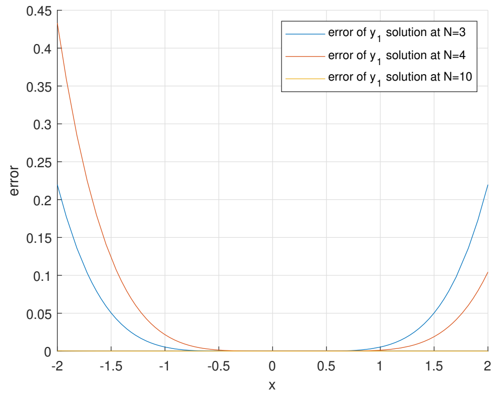
Figure 5

The solution of system (4.2) using Bessel functions method.



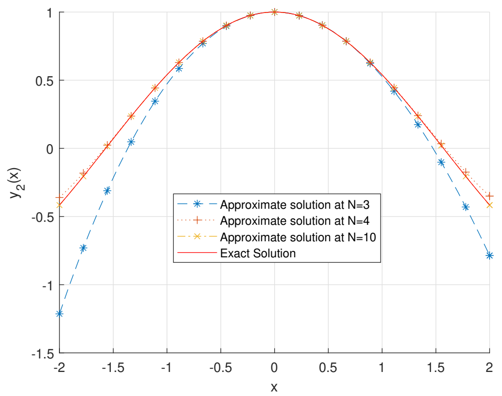
(a)

The approximate and the exact solution of $y_1(x)$



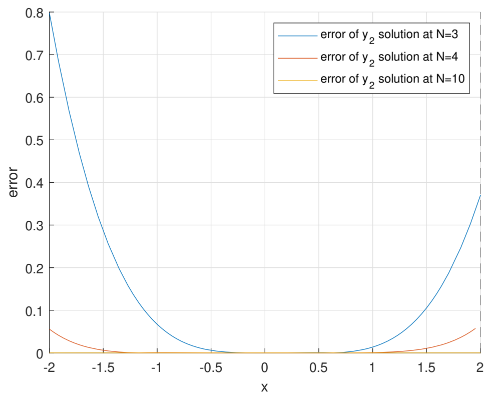
(b)

The Absolute error in the approximate solution of $y_1(x)$



(c)

The approximate and the exact solution of $y_2(x)$



(d)

The Absolute error in the approximate solution of $y_2(x)$

Solution of system (4.2) using Legendre polynomials method:

Applying the method of Legendre polynomials on this example using Matlab software in Appendix C, the results are: when N=3;

$$\begin{aligned} y_1 = &-0.16380480174553507585004353229452\,x^3 \\ &- 0.00000000000000000023886329112571269561726694519166\,x^2 \\ &+ 1.0\,x + 0.000000000000000000013552527156068805425093160010874 \end{aligned}$$

$$\begin{aligned} y_2 = &\ 0.026737218335614157855376760283481\,x^3 \\ &- 0.50000000000000000024394548880924\,x^2 + 1.0 \end{aligned}$$

When N=4;

$$\begin{aligned} y_1(x) = &\ 0.010274735667402140702779143638552\,x^4 \\ &- 0.16992984622837332158825827087278\,x^3 \\ &- 0.00000000000000000024616895029578103604173122676002\,x^2 + 1.0\,x \\ &+ 0.000000000000000000013552527156068805425093160010874 \end{aligned}$$

$$\begin{aligned} y_2(x) = &\ 0.040305681942431853546687705452944\,x^4 \\ &+ 0.00067096735562464660671263085890692\,x^3 \\ &- 0.50000000000000000027105054312137\,x^2 \\ &- 0.000000000000000000033881317890172013562732900027186\,x + 1.0 \end{aligned}$$

When N=10;

$$
\begin{aligned}
y_1(x) = {} & -0.0000001069038079980344379924832023430 4\, x^{10} \\
& + 0.000002955314279012769341034318688971 1\, x^{9} \\
& - 0.000000211184748004115867767068109071 03\, x^{8} \\
& - 0.000198274050781271639310283091552 19\, x^{7} \\
& - 0.000000057867474824248628755223399059571\, x^{6} \\
& + 0.0083333495324761449795556805765148\, x^{5} \\
& - 0.0000000023432247739794501742029092592867\, x^{4} \\
& - 0.16666666145404249096815445588 26\, x^{3} \\
& - 0.00000000000000000067351031657704823675054086471516\, x^{2} \\
& + 0.99999999999999999970997591886012 8\, x \\
& - 0.000000000000000000125085590485778802571914525287 87
\end{aligned}
$$

$$
\begin{aligned}
y_2(x) = {} & -0.0000002531189355587400267647963695543 2\, x^{10} \\
& - 0.000000056157128749225896141323854266723\, x^{9} \\
& + 0.0000248684704162178374741743168813 4\, x^{8} \\
& - 0.000000046431381970032848449977136281 39\, x^{7} \\
& - 0.0013888686800960780539548486357728\, x^{6} \\
& - 0.0000000044157648785877192659884060612491\, x^{5} \\
& + 0.041666667514691865863786556303805\, x^{4} \\
& + 0.0000000049779340533128017353317715416872\, x^{3} \\
& - 0.49999999999999999880602235755033 8\, x^{2} \\
& + 0.00000000000000000050775101834726090057722772846439\, x \\
& + 1.000000000000000000216840434497 1
\end{aligned}
$$

Table (6) contains both the exact and the numerical solutions of system (4.2).

Table 6

Summary of approximate results of solving Example.2 using Bessel functions at N =3, N=10

			Solution of $y_1(x)$		
x	y_{exact}	y_{app} at N=3	Absolute error at N=3 $\lvert y_{exact} - y_{app}\rvert$	y_{app} at N=10	Absolute error at N=10 $\lvert y_{exact} - y_{app}\rvert$
0	0	0	0	-0	$1.250855904857788\times10^{-18}$
0.1	0.099833416646828	0.099836195198254	0.000002778551426	0.099833416651923	$5.094366101505239\times10^{-12}$
0.2	0.198669330795061	0.198689561586036	0.000020230790975	0.198669330835819	$4.075784785755639\times10^{-11}$
0.3	0.295520206661340	0.295577270352871	0.000057063691531	0.295520206800087	$1.387471396049444\times10^{-10}$
0.4	0.389418342308651	0.389516492688286	0.000098150379635	0.389418342642044	$3.333930031863094\times10^{-10}$
0.5	0.479425538604203	0.479524399781808	0.000098861177605	0.479425539267242	$6.630385216219262\times10^{-10}$
0.6	0.564642473395035	0.564618162822964	0.000024310572071	0.564642474566817	$1.171781642019386\times10^{-9}$
0.7	0.644217687237691	0.643814953001281	0.000402734236410	0.644217689149310	$1.911619094772933\times10^{-9}$
0.8	0.717356090899523	0.716131941506286	0.001224149393237	0.717356093844659	$2.945136410124856\times10^{-9}$
0.9	0.783326909627483	0.780586299527505	0.002740610099978	0.783326913977198	$4.349714720490889\times10^{-9}$
1	0.841470984807897	0.836195198254465	0.005275786553432	0.841470991042676	$6.234779277224345\times10^{-9}$

			Solution of $y_2(x)$		
x	y_{exact}	y_{app} at N=3	Absolute error at N=3 $\lvert y_{exact} - y_{app}\rvert$	y_{app} at N=10	Absolute error at N=10 $\lvert y_{exact} - y_{app}\rvert$
0	1	1	0	1	$2.168404344971009\times10^{-18}$
0.1	0.995004165278026	0.995026737218336	0.000022571940310	0.995004165283061	$5.034762291969259\times10^{-12}$
0.2	0.980066577841242	0.980213897746685	0.000147319905443	0.980066577881853	$4.061107178642230\times10^{-11}$
0.3	0.955336489125606	0.955721904895062	0.000385415769456	0.955336489264141	$1.385349279844542\times10^{-10}$
0.4	0.921060994002885	0.921711181973479	0.000650187970594	0.921060994336098	$3.332126058936699\times10^{-10}$
0.5	0.877582561890373	0.878342152291952	0.000759590401579	0.877582562553640	$6.632671752960817\times10^{-10}$
0.6	0.825335614909678	0.825775239160493	0.000439624250814	0.825335616083208	$1.173529743361902\times10^{-9}$
0.7	0.764842187284488	0.764170865889116	0.000671321395373	0.764842189202088	$1.917599395969983\times10^{-9}$
0.8	0.696706709347165	0.693689455787834	0.003017253559331	0.696706712308397	$2.961231756345293\times10^{-9}$
0.9	0.621609968270664	0.614491432166663	0.007118536104002	0.621609972657172	$4.386507961602291\times10^{-9}$
1	0.540302305868140	0.526737218335614	0.013565087532526	0.540302312159735	$6.291595204158371\times10^{-9}$

The maximum error at N=3 corresponding to y_1 and y_2 is $E_1 \simeq 0.005275786553432$ and $E_2 = 0.013565087532526$ respectively for $x \in [0, 1]$.

The maximum error at N=4 corresponding to y_1 and y_2 is $E_1 \simeq 0.001126095368868$ and $E_2 = 6.743434299167827 \times 10^{-4}$ respectively for $x \in [0, 1]$.

The maximum error at N=10 corresponding to y_1 and y_2 is $E_1 \simeq 6.234779277224345 \times 10^{-9}$ and $E_2 = 6.291595204158371 \times 10^{-9}$ respectively for $x \in [0, 1]$.

These approximate solutions are illustrated as follows;

Figure 6a compares the exact solution $y_1(x)$ and the approximate solution with N=3,4,10.

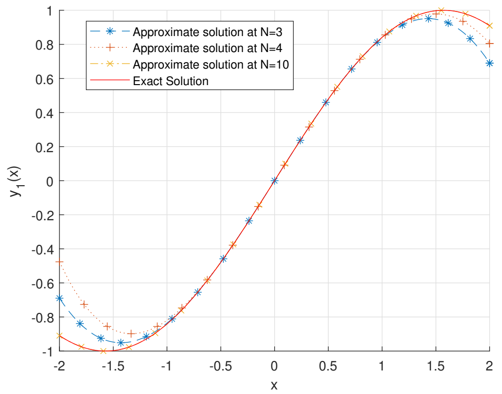
Figure 6b shows the error in the approximate solution of $y_1(x)$.

Figure 6c compares the exact solution $y_2(x)$ and the approximate solution with N=3,4,10.

Figure 6d shows the error in the approximate solution of $y_2(x)$.

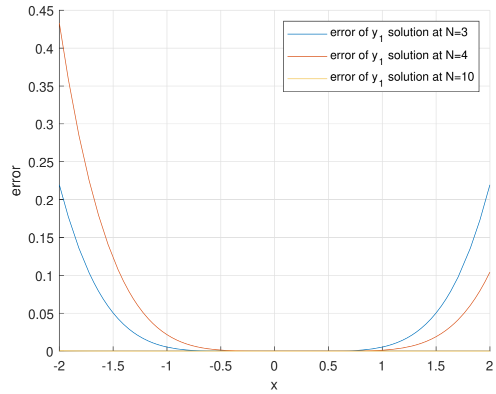
Figure 6

The solution of system (4.2) using Legendre polynomials method.



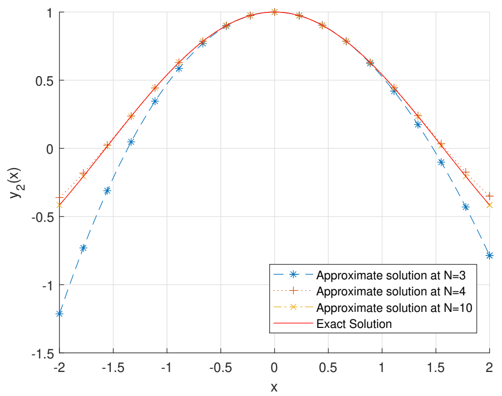
(a)

The approximate and the exact solution of $y_1(x)$.



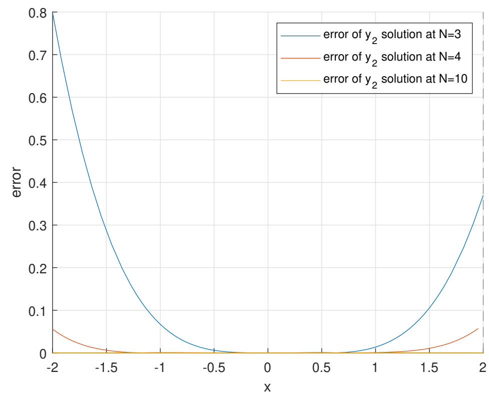
(b)

The absolute error in $y_1(x)$ approximate solution



(c)

The approximate and the exact solution of $y_2(x)$.



(d)

The absolute error in $y_2(x)$ approximate solution

Example 4.3:

Consider the following system of Fredholm integro-differential equations:

$$\begin{aligned} y_1^{(2)}(x) + y_2^{(1)}(x) &= 3x^2 + \frac{3x}{10} + 8 - \int_0^1 2xt\{y_1(t) - 3y_2(t)\}dt \\ y_1^{(1)}(x) + y_2^{(2)}(x) &= 21x + \frac{4}{5} - \int_0^1 3(2x+t^2)\{y_1(t) - 2y_2(t)\}\mathrm{d}t \end{aligned} \tag{4.3}$$

subject to the initial conditions:

$y_1(0) + y_1^{(1)}(0) = 1, y_1(1) + y_1^{(1)}(1) = 10, y_2(0) + y_2^{(1)}(0) = 1, y_2(1) + y_2^{(1)}(1) = 7.$

System (4.3) has the exact solution:

$y_1(x) = 3x^2 + 1$ and $y_2(x) = x^3 + 2x - 1.$

Solution of system (4.3) using Laguerre polynomials method:

The procedure yields the approximate solutions of the system of equations for N=3 is: $y_1(x) = 3x^2 + 1$ and $y_2(x) = x^3 + 2x - 1$ which are exactly the same as the exact one. Table (7) contains both the exact and the numerical solutions of system (4.3)

Table 7
Summary of approximate results of solving Example.3 using Laguerre polynomials.

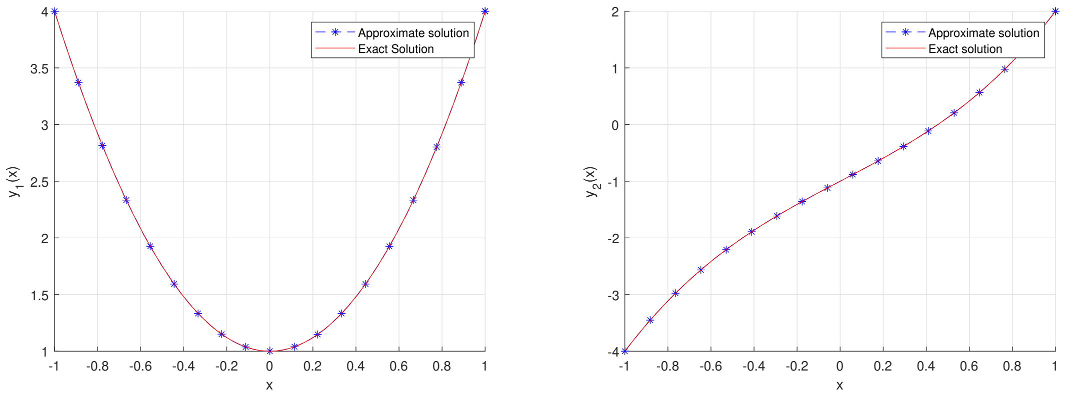
	Solution of $y_1(x)$		Solution of $y_2(x)$	
x	y_{exact}	y_{app}	y_{exact}	y_{app}
0.0	1.000000	1.000000	-1.000000	-1.000000
0.1	1.030000	1.030000	-0.799000	-0.799000
0.2	1.120000	1.120000	-0.592000	-0.592000
0.3	1.270000	1.270000	-0.373000	-0.373000
0.4	1.480000	1.480000	-0.136000	-0.136000
0.5	1.750000	1.750000	0.125000	0.125000
0.6	2.080000	2.080000	0.416000	0.416000
0.7	2.470000	2.470000	0.743000	0.743000
0.8	2.920000	2.920000	1.112000	1.112000
0.9	3.430000	3.430000	1.529000	1.529000
1.0	4.000000	4.000000	2.000000	2.000000

These approximate solutions are illustrated as follows;

Figure 7a compares the exact solution $y_1(x)$ and the approximate solution with N=3.

Figure 7b compares the exact solution $y_2(x)$ and the approximate solution with N=3.

Figure 7
The solution of system (4.3) using Laguerre polynomials method.



(a) *approximate and exact solution of $y_1(x)$ by Laguerre polynomials method*

(b) *approximate and exact solution of $y_2(x)$ by Laguerre polynomials method*

Solution of system (4.3) using Bessel functions method:

To solve system (4.3) by the Bessel functions method we use algorithm in Appendix B and MATLAB software, the procedure yields the approximate solutions of the system of equations for N=3 as follows:

$$y_1(x) = -0.000000000000000091819911542270250672721196341747\,x^3 + 3.0\,x^2 + 1.0$$
$$y_2(x) = 0.99999999999999920305719311452997\,x^3$$
$$-\,0.00000000000000068088281447065726366872309255885\,x^2 + 2.0\,x - 1.0$$

compared to the exact solution:

$y_1(x) = 3x^2 + 1$ and $y_2(x) = x^3 + 2x - 1$.

Table (8) contains both the exact and the numerical solutions of system (4.3).

Table 8
Summary of approximate results of solving Example.3 using Bessel functions.

	Solution of $y_1(x)$				Solution of $y_2(x)$	
x	y_{exact}	y_{app}	Absolute error $\lvert y_{exact} - y_{app}\rvert$	y_{exact}	y_{app}	Absolute error $\lvert y_{exact} - y_{app}\rvert$
0	0	0	0	-1.0000	-1.0000	0
0.1	1.0300	1.0300	4.4710×10^{-19}	-7.9900×10^{-01}	-7.9900×10^{-01}	3.9368×10^{-17}
0.2	1.1200	1.1200	2.9111×10^{-18}	-5.9200×10^{-01}	-5.9200×10^{-01}	1.7255×10^{-16}
0.3	1.2700	1.2700	9.0759×10^{-18}	-3.7300×10^{-01}	-3.7300×10^{-01}	4.2218×10^{-16}
0.4	1.4800	1.4800	2.0626×10^{-17}	-1.3600×10^{-01}	-1.3600×10^{-01}	8.1088×10^{-16}
0.5	1.7500	1.7500	3.9244×10^{-17}	1.2500×10^{-01}	1.2500×10^{-01}	1.3613×10^{-15}
0.6	2.0800	2.0800	6.6616×10^{-17}	4.1600×10^{-01}	4.1600×10^{-01}	2.0960×10^{-15}
0.7	2.4700	2.4700	1.0442×10^{-16}	7.4300×10^{-01}	7.4300×10^{-01}	3.0376×10^{-15}
0.8	2.9200	2.9200	1.5435×10^{-16}	1.1120	1.1120	4.2088×10^{-15}
0.9	3.4300	3.4300	2.1809×10^{-16}	1.5290	1.5290	5.6321×10^{-15}
1.0	4.0000	4.0000	2.9731×10^{-16}	2.0000	2.0000	7.3303×10^{-15}

The maximum error at corresponding to y_1 and y_2 is $E_1 \simeq 2.9731 \times 10^{-16}$ and $E_2 = 7.3303 \times 10^{-15}$ respectively for $x \in [0, 1]$.

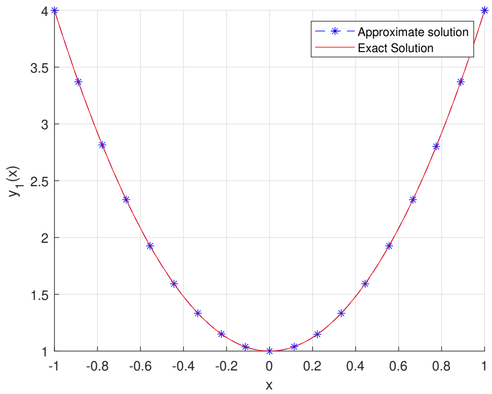
These approximate solutions are illustrated as follows;

Figure 8a compares the exact solution $y_1(x)$ and the approximate solution with N=3. Figure 8b shows the error in the approximate solution of $y_1(x)$.

Figure 8c compares the exact solution $y_2(x)$ and the approximate solution with N=3. Figure 8d shows the error in the approximate solution of $y_2(x)$.

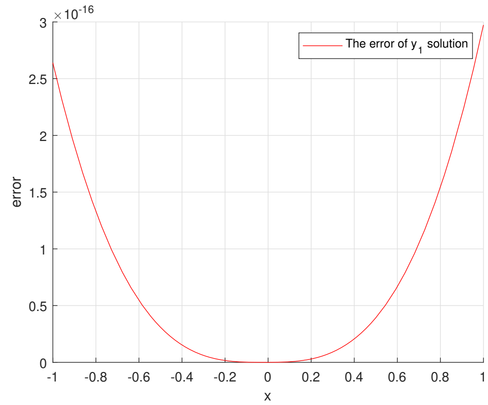
Figure 8

The solution of system (4.3) using Bessel functions method.



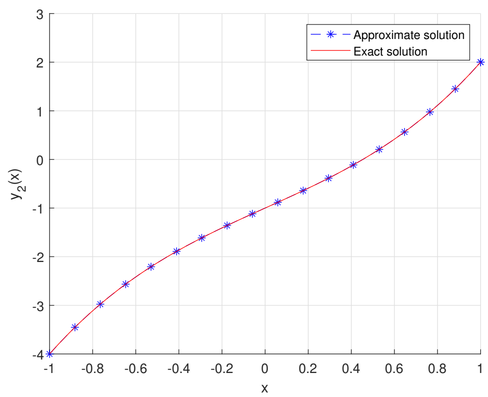
(a)

The approximate and the exact solution of $y_1(x)$.



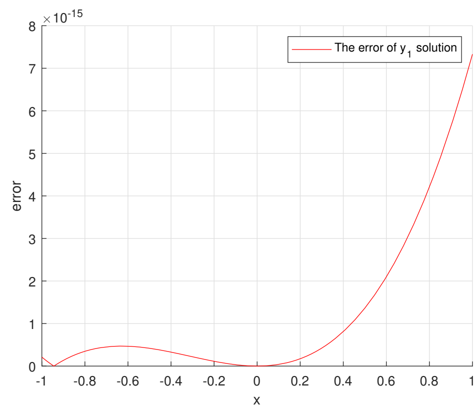
(b)

Absolute Error in $y_1(t)$ approximate solution.



(c)

The approximate and the exact solution of $y_2(x)$.



(d)

Absolute Error in $y_2(t)$ approximate solution.

Solution of system (4.3) using Legendre polynomials method:

To solve system (4.3) by the Bessel Legendre method we use algorithm in Appendix C and MATLAB software to get: Applying the algorithm in appendix C, the procedure yields the approximate solutions of the system of equations for N=3 is:

$$y_1 = -0.00000000000000028067063850031810375695099516938\, x^3 \\ + 2.99999999999999999833574966523475\, x^2 + 1.0$$

$$y_2 = 1.00000000000000037705298952528388\, x^3 \\ + 0.00000000000000035597686661810752061875287175083\, x^2 + 2.0\, x - 1.0$$

compared to the exact solution $y_1(x) = 3x^2 + 1$ and $y_2(x) = x^3 + 2x - 1$.

Table (9) contains both the exact and the numerical solutions of system (4.3).

Table 9
Summary of approximate results of solving Example.3 using Legendre polynomials.

	Solution of $y_1(x)$			Solution of $y_2(x)$		
x	y_{exact}	y_{app}	Absolute error $\lvert y_{exact} - y_{app}\rvert$	y_{exact}	y_{app}	Absolute error $\lvert y_{exact} - y_{app}\rvert$
0	0	0	0	-1.0000	-1.0000	0
0.1	1.0300	1.0300	4.4710$\times 10^{-19}$	-7.9900$\times 10^{-01}$	-7.9900$\times 10^{-01}$	3.9368$\times 10^{-17}$
0.2	1.1200	1.1200	2.9111$\times 10^{-18}$	-5.9200$\times 10^{-01}$	-5.9200$\times 10^{-01}$	1.7255$\times 10^{-16}$
0.3	1.2700	1.2700	9.0759$\times 10^{-18}$	-3.7300$\times 10^{-01}$	-3.7300$\times 10^{-01}$	4.2218$\times 10^{-16}$
0.4	1.4800	1.4800	2.0626$\times 10^{-17}$	-1.3600$\times 10^{-01}$	-1.3600$\times 10^{-01}$	8.1088$\times 10^{-16}$
0.5	1.7500	1.7500	3.9244$\times 10^{-17}$	1.2500$\times 10^{-01}$	1.2500$\times 10^{-01}$	1.3613$\times 10^{-15}$
0.6	2.0800	2.0800	6.6616$\times 10^{-17}$	4.1600$\times 10^{-01}$	4.1600$\times 10^{-01}$	2.0960$\times 10^{-15}$
0.7	2.4700	2.4700	1.0442$\times 10^{-16}$	7.4300$\times 10^{-01}$	7.4300$\times 10^{-01}$	3.0376$\times 10^{-15}$
0.8	2.9200	2.9200	1.5435$\times 10^{-16}$	1.1120	1.1120	4.2088$\times 10^{-15}$
0.9	3.4300	3.4300	2.1809$\times 10^{-16}$	1.5290	1.5290	5.6321$\times 10^{-15}$
1.0	4.0000	4.0000	2.9731$\times 10^{-16}$	2.0000	2.0000	7.3303$\times 10^{-15}$

The maximum error at corresponding to y_1 and y_2 is $E_1 \simeq 2.9731 \times 10^{-16}$ and $E_2 = 7.3303 \times 10^{-15}$ respectively for $x \in [0, 1]$.

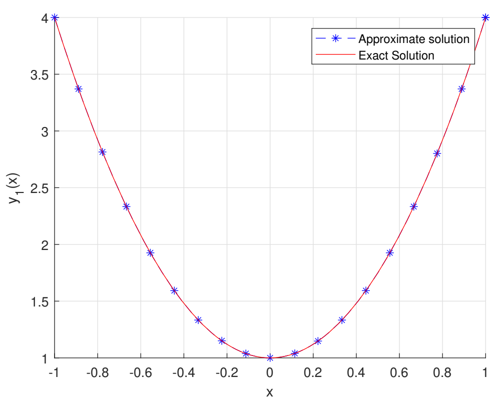
These approximate solutions are illustrated as follows;

Figure 9a compares the exact solution $y_1(x)$ and the approximate solution with N=3. Figure 9b shows the error in the approximate solution of $y_1(x)$.

Figure 9c compares the exact solution $y_2(x)$ and the approximate solution with N=3. Figure 9d shows the error in the approximate solution of $y_2(x)$.

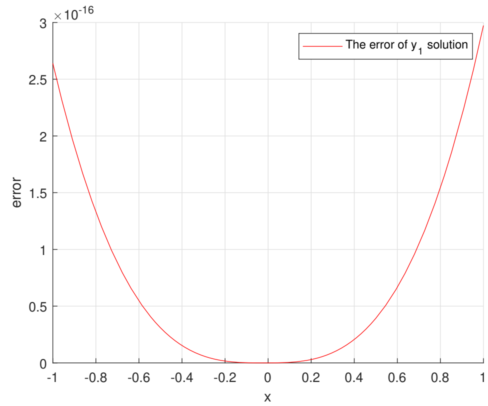
Figure 9

The solution of system (4.3) using Legendre polynomials method.



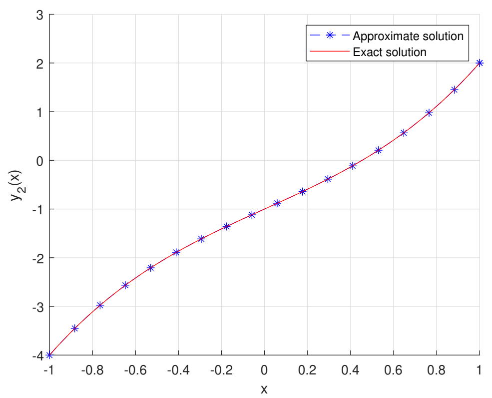
(a)

The approximate and the exact solution of $y_1(x)$ using Legendre polynomials method



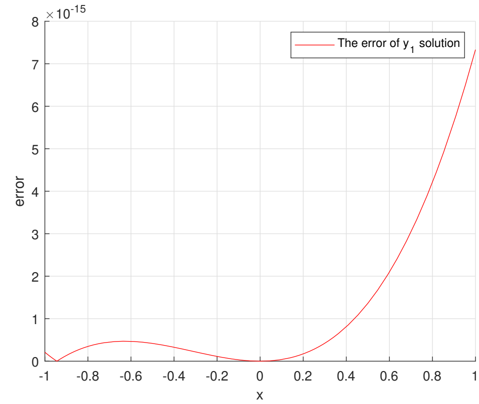
(b)

The absolute error of approximate solution of $y_1(t)$ using Legendre polynomials



(c)

The approximate and the exact solution of $y_2(x)$ using Legendre polynomials method



(d)

The absolute error of approximate solution of $y_2(t)$ using Legendre polynomials

4.2 Conclusion

In this thesis, we have solved systems of Fredholm integro-differential equations using several numerical methods. These include the Laguerre polynomials method, Bessel functions method and legendre polynomials method. These numerical methods were implemented in the form of MATLAB code to solve some numerical examples.
From the numerical results shown in tables and illustrated in figures, we see clearly that the Laguerre polynomials method is one of the most effective numerical method for solving system of Fredholm integro-differential equations.

References

[1] Al-Faour, O. M., and Saeed, R. K. Solution of a system of linear volterra integral and integro-differential equations by spectral method. *Al-Nahrain University Journal for Science 6*, 2 (2006), 30–46.

[2] Bahşı, M. M., Kurt Bahşı, A., Çevik, M., and Sezer, M. Improved jacobi matrix method for the numerical solution of fredholm integro-differential-difference equations. *Mathematical Sciences 10*, 3 (2016), 83–93.

[3] Balci, M. A., and Sezer, M. Hybrid euler–taylor matrix method for solving of generalized linear fredholm integro-differential difference equations. *Applied Mathematics and Computation 273* (2016), 33–41.

[4] Behiry, S. Solution of nonlinear fredholm integro-differential equations using a hybrid of block pulse functions and normalized bernstein polynomials. *J. Comput. Appl. Math. 260* (2014), 258–265.

[5] Biazar, J., Ghazvini, H., and Eslami, M. He's homotopy perturbation method for systems of integro-differential equations. *Chaos, Solitons & Fractals 39*, 3 (2009), 1253–1258.

[6] Bildik, N., Konuralp, A., and Yalçinbaş, S. Comparison of legendre polynomial approximation and variational iteration method for the solutions of general linear fredholm integro-differential equations. *Computers and Mathematics with Applications 59*, 6 (2010), 1909–1917.

[7] Das, A., Hazarika, B., Srivastava, H., Rabbani, M., and Arab, R. Solvability of infinite systems of nonlinear integral equations in two variables by using semi-analytic method. *Filomat 33*, 16 (2019), 5375–5386.

[8] Elahi, Z., Akram, G., and Siddiqi, S. S. Laguerre approach for solving system of linear fredholm integro-differential equations. *Mathematical Sciences 12*, 3 (2018), 185–195.

[9] GELMI, C. A., AND JORQUERA, H. Idsolver: A general purpose solver for nth-order integro-differential equations. *Computer Physics Communications 185*, 1 (2014), 392–397.

[10] GÜLSU, M., AND SEZER, M. Taylor collocation method for solution of systems of high-order linear fredholm–volterra integro-differential equations. *International Journal of Computer Mathematics 83*, 4 (2006), 429–448.

[11] GÜRBÜZ, B., SEZER, M., AND GÜLER, C. Laguerre collocation method for solving fredholm integro-differential equations with functional arguments. *Journal of Applied Mathematics 2014* (2014).

[12] GÜLSU, M., SEZER, M., AND TANAY, B. A matrix method for solving high-order linear difference equations with mixed argument using hybrid legendre and taylor polynomials. *Journal of the Franklin Institute 343*, 6 (2006), 647–659.

[13] HAMAYDI, J., AND QATANANI, N. Computational methods for solving linear fuzzy volterra integral equation. *Journal of Applied Mathematics 2017* (2017).

[14] HESAMEDDINI, E., AND RAHIMI, A. A new numerical scheme for solving systems of integro-differential equations. *Computational Methods for Differential Equations 1*, 2 (2013), 108–119.

[15] HESAMEDDINI, E., AND RIAHI, M. Bernoulli galerkin matrix method and its convergence analysis for solving system of volterra–fredholm integro-differential equations. *Iranian Journal of Science and Technology, Transactions A: Science 43*, 3 (2019), 1203–1214.

[16] ISSA, A., QATANANI, N., AND DARAGHMEH, A. Approximation techniques for solving linear systems of volterra integro-differential equations. *Journal of Applied Mathematics 2020* (2020).

[17] KURT, N., AND CEVIK, M. Polynomial solution of the single degree of freedom system by taylor matrix method. *Mechanics Research Communications 35*, 8 (2008), 530–536.

[18] LOH, J. R., AND PHANG, C. Numerical solution of fredholm fractional integro-differential equation with right-sided caputo's derivative using bernoulli polynomials operational matrix of fractional derivative. *Mediterranean Journal of Mathematics 16*, 2 (2019), 1–25.

[19] MALEKNEJAD, K., AND KAJANI, M. T. Solving linear integro-differential equation system by galerkin methods with hybrid functions. *Applied Mathematics and Computation 159*, 3 (2004), 603–612.

[20] MALEKNEJAD, K., SAFDARI, H., AND NOURI, M. Numerical solution of an integral equations system of the first kind by using an operational matrix with block pulse functions. *International Journal of Systems Science 42*, 1 (2011), 195–199.

[21] MATINFAR, M., NASSERI, S., GHANBARI, M., AND ABDOLLAHI, H. Numerical solution of linear fuzzy fredholm integro-differential equations by adomian decomposition method. *International Journal of Applied Mathematics 22*, 5 (2009), 677–688.

[22] MATINFAR, M., AND RIAHIFAR, A. A matrix method for system of integro-differential equations by using generalized laguerre polynomials. *differential equations ...* (2016), 14.

[23] MIRZAEE, F., AND HOSEINI, S. F. Solving systems of linear fredholm integro-differential equations with fibonacci polynomials. *Ain Shams Engineering Journal 5*, 1 (2014), 271–283.

[24] MOLLAOĞLU, T., AND SEZER, M. A numerical approach with residual error estimation for eolution of high-order linear differential-difference equations by using gegenbauer polynomials. *Celal Bayar University Journal of Science 13*, 1 (2017), 39–49.

[25] MOOSAVI NOORI, S. R., AND TAGHIZADEH, N. Modified differential transform method for solving linear and nonlinear pantograph type of differential and volterra integro-differential equations with proportional delays. *Advances in Difference Equations 2020*, 1 (2020), 1–25.

[26] Öztürk, Y., and Demir, A. İ. A spectral collocation matrix method for solving linear fredholm integro-differential–difference equations. *Computational and Applied Mathematics 40*, 6 (2021), 1–17.

[27] Palestini, A. Mathematical modeling with differential equations in physics, chemistry, biology, and economics. *MDPI, Basel* (2022).

[28] Pour-Mahmoud, J., Rahimi-Ardabili, M. Y., and Shahmorad, S. Numerical solution of the system of fredholm integro-differential equations by the tau method. *Applied Mathematics and Computation 168*, 1 (2005), 465–478.

[29] Ramadan, M. A., and Ali, M. R. Numerical solution of volterra-fredholm integral equations using hybrid orthonormal bernstein and block-pulse functions. *Asian Research Journal of Mathematics* (2017), 1–14.

[30] Seifi, A., Lotfi, T., and Allahviranloo, T. A new efficient method using fibonacci polynomials for solving of first-order fuzzy fredholm–volterra integro-differential equations. *Soft Computing 23*, 19 (2019), 9777–9791.

[31] Sekar, R. C. G., and Murugesan, K. System of linear second order volterra integro-differential equations using single term walsh series technique. *Applied Mathematics and Computation 273* (2016), 484–492.

[32] Tahernezhad, T., and Jalilian, R. Exponential spline for the numerical solutions of linear fredholm integro-differential equations. *Advances in Difference Equations 2020*, 1 (2020), 1–15.

[33] Vahidi, A., Babolian, E., Cordshooli, G. A., and Azimzadeh, Z. Numerical solution of fredholm integro-differential equation by adomian's decomposition method. *International Journal of Mathematical Analysis 3*, 33-36 (2009), 1769–1773.

[34] Wazwaz, A.-M. *Fredholm Integro-Differential Equations*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 213–235.

[35] Wazwaz, A.-M. Volterra-fredholm integro-differential equations. In *Linear and Nonlinear Integral Equations: Methods and Applications*. Springer, 2011, pp. 285–309.

[36] YALÇINBAŞ, S., SEZER, M., AND SORKUN, H. H. Legendre polynomial solutions of high-order linear fredholm integro-differential equations. *Applied Mathematics and Computation 210*, 2 (2009), 334–349.

[37] YALÇINBAŞ, S., SEZER, M., AND SORKUN, H. H. Legendre polynomial solutions of high-order linear fredholm integro-differential equations. *Applied Mathematics and Computation 210*, 2 (2009), 334–349.

[38] YÜZBAŞI, Ş., ŞAHIN, N., AND SEZER, M. Numerical solutions of systems of linear fredholm integro-differential equations with bessel polynomial bases. *Computers and Mathematics with Applications 61*, 10 (2011), 3079–3096.

Appendices

Appendix A

Solving Example 1 by Laguerre polynomials method

```
clc
clear;Co=3;Vk=2;
Psi=zeros(Co+1,Co+1);
for ic=1:Co+1
for Jc=ic:2:Co+1
Psi(ic,Jc)=(((-1)^Jc)/factorial(Jc))*com(ic,Jc);
end
end
format rational
Psi
```

Psi = -1 0 0 0 0 1/2 0 0 0 0 -1/6 0 0 0 0 1/24

```
format rational
a=0;b=1;
ic=0:Co;
col=a+(b-a)*ic/Co;
Z=zeros(Co+1,Co+1);
for ic=1:Co
Z(ic+1,ic)=ic;
end
Z
```

Z = 0 0 0 0 1 0 0 0 0 2 0 0 0 0 3 0

```
syms xs
for ic=1:(Co+1);
Lxs(ic)=xs^(ic-1);
end
Lxs
```

Lxs =

$$\begin{pmatrix} 1 & xs & xs^2 & xs^3 \end{pmatrix}$$

```
LLxs=kron(eye(Vk),Lxs);
D=kron(eye(Vk),Psi')
```

D = -1 0 0 0 0 0 0 0 0 1/2 0 0 0 0 0 0 0 0 -1/6 0 0 0 0 0 0 0 0 1/24 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 1/2 0 0 0 0 0 0 0 0 -1/6 0 0 0 0 0 0 0 0 1/24

```
Z=kron(eye(Vk),Z');
p0xs=[0 0;0 0];p1xs=[0 0;0 0];p2xs=[1 0;0 1];
```

```
p3xs=[0 0;0 0];p4xs=[0 0;0 0];
gxs=[8/9;6*xs-(1/18)*xs^2]
P0=kron(eye(Co+1),p0xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
P0(ic,:)=subs(P0(ic,:),col(Jc));
end
end

P1=kron(eye(Co+1),p1xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
P1(ic,:)=subs(P1(ic,:),col(Jc));
end
end

P2=kron(eye(Co+1),p2xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
P2(ic,:)=subs(P2(ic,:),col(Jc));
end
end

P3=kron(eye(Co+1),p3xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
P3(ic,:)=subs(P3(ic,:),col(Jc));
end
end

P4=kron(eye(Co+1),p4xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
P4(ic,:)=subs(P4(ic,:),col(Jc));
end
end

for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Ga(ic,1)=subs(gxs(ic-(Jc-1)*Vk,1),col(Jc));
end
end
Ga
```

$$\begin{pmatrix} \frac{8}{9} \\ 0 \\ \frac{8}{9} \\ \frac{323}{162} \\ \frac{8}{9} \\ \frac{322}{81} \\ \frac{8}{9} \\ \frac{107}{18} \end{pmatrix}$$

```
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
X(ic,:)=subs(LLxs(ic-(Jc-1)*Vk,:),col(Jc));
end
end
X
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & \frac{1}{3} & \frac{1}{9} & \frac{1}{27} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{3} & \frac{1}{9} & \frac{1}{27} \\ 1 & \frac{2}{3} & \frac{4}{9} & \frac{8}{27} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{2}{3} & \frac{4}{9} & \frac{8}{27} \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

```
syms xs t
f11=1/3;
for row=0:Co
for column=0:Co
k11(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f11,xs,row),t,column),{xs,t},{0,0});
end
end
k11;
Kl11=(Psi^-1)'*k11*(Psi^-1);
f12=1/4;
for row=0:Co
for column=0:Co
k12(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f12,xs,row),t,column),{xs,t},{0,0});
end
end
k12;
Kl12=(Psi^-1)'*k12*(Psi^-1);
f21=xs^2/6;
for row=0:Co
for column=0:Co
k21(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f21,xs,row),t,column),{xs,t},{0,0});
end
end
```

```
k21;
Kl21=(Psi^-1)'*k21*(Psi^-1);
f22=-xs^2/3;
for row=0:Co
for column=0:Co
k22(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f22,xs,row),t,column),{xs,t},{0,0});
end
end
k22;
Kl22=(Psi^-1)'*k22*(Psi^-1);
Klf=[Kl11 Kl12;Kl21 Kl22];
M=zeros(column+1,column+1);
for ic=1:(column+1)
for Jc=1:(column+1)
M(ic,Jc)=(b^(ic+Jc-1)-a^(ic+Jc-1))/(ic+Jc-1);
end
end
M
```

M = 1 1/2 1/3 1/4 1/2 1/3 1/4 1/5 1/3 1/4 1/5 1/6 1/4 1/5 1/6 1/7

```
Q=Psi*M*Psi'
```

Q = 1 -1/4 1/18 -1/96 -1/4 1/12 -1/48 1/240 1/18 -1/48 1/180 -1/864 -1/96 1/240 -1/864 1/4032

```
Qp=kron(eye(Vk),Q);
I=X*D*Klf*Qp;
W=P0*X*D+P1*X*Z*D+P2*X*Z^2*D+P3*X*Z^3*D+P4*X*Z^4*D-I;
t0=0;yt0=0;
t1=0;yt1=1/3;
t2=0;yt2=0;
t3=0;yt3=-1/2;
L0=yt0
```

```
L1=yt1
```

```
L2=yt2
```

```
L3=yt3;
U0=subs(LLxs(1,:),t0)*D
U1=subs(LLxs(1,:),t1)*Z*D
U2=subs(LLxs(2,:),t2)*D
U3=subs(LLxs(2,:),t3)*Z*D
W(Vk*(Co+1)-3,:)=[U0];
W(Vk*(Co+1)-2,:)=[U1];
W(Vk*(Co+1)-1,:)=[U2];
W(Vk*(Co+1),:)=[U3];
WW=W
```

$$\begin{pmatrix} \frac{1}{3} & -\frac{1}{12} & -\frac{17}{54} & -\frac{1}{288} & \frac{1}{4} & -\frac{1}{16} & \frac{1}{72} & -\frac{1}{384} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & 0 \\ \frac{1}{3} & -\frac{1}{12} & -\frac{17}{54} & \frac{23}{288} & \frac{1}{4} & -\frac{1}{16} & \frac{1}{72} & -\frac{1}{384} \\ \frac{1}{54} & -\frac{1}{216} & \frac{1}{972} & -\frac{1}{5184} & -\frac{1}{27} & \frac{1}{108} & -\frac{163}{486} & \frac{217}{2592} \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \end{pmatrix}$$

```
Ga(Vk*(Co+1)-3 ,:)=[L0];
Ga(Vk*(Co+1)-2,:)=[L1];
Ga(Vk*(Co+1)-1 ,:)=[L2];
Ga(Vk*(Co+1),:)=[L3];
GaGa=Ga
```

```
A=WW\GaGa;
syms xs
y=vpa(LLxs*D*A,12);
y_1=y(1)
```

$$0.5\,xs^2 + 0.33333333333333333333559208785934\,xs$$

```
y_2=y(2)
```

$$xs^3 - 0.5\,xs$$

```
y1=@(xs) (1/2)*xs^2+(1/3)*xs;%the exsact solution
y2=@(xs) xs^3-xs/2; %the exsact solutionb
error_y1=abs(y1-y_1);
error_y2=abs(y2-y_2);
c=0;
approxsimatesolution1=zeros(1,10);
approxsimatesolution2=zeros(1,10);
Error_y1=zeros(1,10);
Error_y2=zeros(1,10);
c=0;
for ic=0:0.1:1
    c=c+1;
    cou(c)=ic;
approxsimatesolution1(c)=subs(y_1,ic);
exsactsolution1(c)=y1(ic);
%error_y1(c)=abs(approxsimatesolution1(c)-exsactsolution1(c));
approxsimatesolution2(c)=subs(y_2,ic);
exsactsolution2(c)=y2(ic);
Error_y1(c)=subs(error_y1,ic);
Error_y2(c)=subs(error_y2,ic);
%error_y2(c)=abs(approxsimatesolution2(c)-exsactsolution2(c));
end
figure
hold on
fplot(y_1,[-1 1],'--*b')
fplot(y1,[-1 1],'r')
```

```
xslabel ('xs');
ylabel ('y_1(xs)');
legend('Approxsimate solution','Exsact Solution');
grid on
hold off
figure
hold on
fplot(error_y1,[-1 1],'r')
xslabel ('xs');
ylabel ('error');
legend('The error of y_1 solution');
grid on
hold off
xslabel ('xs');
ylabel ('y_2(xs)');
legend('Approxsimate solution','Exsact solution');
grid on
hold off
figure
hold on
fplot(error_y2,[-1 1],'r')
xslabel ('xs');
ylabel ('error');
legend('The error of y_1 solution');
grid on
hold off
Names={ 't';'Approxsimate y_1(t)'; 'Exsact y1(t)';...
'Error y_1';'Approxsimate y_2(t)'; 'Exsact y_2(t)';'Error y_2'}
```

Names = 7xs1 cell 't' 'Approxsimate y$_1(t)''Exsacty1(t)''Errory'_1'Approxsimatey_2(t)''Exsacty_2(t)''Errory'_2$

```
format shortE
Result=[ cou; exsactsolution1;approxsimatesolution1;...
Error_y1;exsactsolution2;approxsimatesolution2;Error_y2].';
%Result=Result.';

result=table(Result)
```

```
clc
clear;Co=3;Vk=2;
Psi=zeros(Co+1,Co+1);
for ic=1:Co+1
for Jc=ic:2:Co+1
Psi(ic,Jc)=(((-1)^Jc)/factorial(Jc))*com(ic,Jc);
end
end
format rational
Psi
```

Psi = -1 0 0 0 0 1/2 0 0 0 0 -1/6 0 0 0 0 1/24

```
format rational
a=0;b=1;
ic=0:Co;
col=a+(b-a)*ic/Co;
Z=zeros(Co+1,Co+1);
for ic=1:Co
Z(ic+1,ic)=ic;
end
Z
```

Z = 0 0 0 0 1 0 0 0 0 2 0 0 0 0 3 0

```
syms xs
for ic=1:(Co+1);
Lxs(ic)=xs^(ic-1);
end
Lxs
```

Lxs = $\begin{pmatrix} 1 & xs & xs^2 & xs^3 \end{pmatrix}$

```
LLxs=kron(eye(Vk),Lxs);
D=kron(eye(Vk),Psi');
LLxs
```

LLxs = $\begin{pmatrix} 1 & xs & xs^2 & xs^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & xs & xs^2 & xs^3 \end{pmatrix}$

```
Z=kron(eye(Vk),Z');
p0xs=[0 0;0 0];p1xs=[0 0;0 0];p2xs=[1 0;0 1];
p3xs=[0 0;0 0];p4xs=[0 0;0 0];
gxs=[8/9;6*xs-(1/18)*xs^2]
```

gxs = $\begin{pmatrix} \frac{8}{9} \\ 6\,xs - \frac{xs^2}{18} \end{pmatrix}$

```
Pm0=kron(eye(Co+1),p0xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm0(ic,:)=subs(Pm0(ic,:),col(Jc));
end
end
Pm1=kron(eye(Co+1),p1xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm1(ic,:)=subs(Pm1(ic,:),col(Jc));
end
end
Pm2=kron(eye(Co+1),p2xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm2(ic,:)=subs(Pm2(ic,:),col(Jc));
```

```
end
end
Pm3=kron(eye(Co+1),p3xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm3(ic,:)=subs(Pm3(ic,:),col(Jc));
end
end
Pm4=kron(eye(Co+1),p4xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm4(ic,:)=subs(Pm4(ic,:),col(Jc));
end
end
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Ga(ic,1)=subs(gxs(ic-(Jc-1)*Vk,1),col(Jc));
end
end
Ga
```

$$Ga = \begin{pmatrix} \frac{8}{9} \\ 0 \\ \frac{8}{9} \\ \frac{323}{162} \\ \frac{8}{9} \\ \frac{322}{81} \\ \frac{8}{9} \\ \frac{107}{18} \end{pmatrix}$$

```
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
X(ic,:)=subs(LLxs(ic-(Jc-1)*Vk,:),col(Jc));
end
end
X
```

$$X = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & \frac{1}{3} & \frac{1}{9} & \frac{1}{27} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{3} & \frac{1}{9} & \frac{1}{27} \\ 1 & \frac{2}{3} & \frac{4}{9} & \frac{8}{27} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{2}{3} & \frac{4}{9} & \frac{8}{27} \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

```
syms xs t
f11=1/3;
for row=0:Co
for column=0:Co
k11(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f11,xs,row),t,column),{xs,t},{0,0});
```

```
end
end
k11;
Kl11=(Psi^-1)'*k11*(Psi^-1);
f12=1/4;
for row=0:Co
for column=0:Co
k12(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f12,xs,row),t,column),{xs,t},{0,0});
end
end
k12;
Kl12=(Psi^-1)'*k12*(Psi^-1);
f21=xs^2/6;
for row=0:Co
for column=0:Co
k21(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f21,xs,row),t,column),{xs,t},{0,0});
end
end
k21;
Kl21=(Psi^-1)'*k21*(Psi^-1);
f22=-xs^2/3;
for row=0:Co
for column=0:Co
k22(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f22,xs,row),t,column),{xs,t},{0,0});
end
end
k22;
Kl22=(Psi^-1)'*k22*(Psi^-1);
Klf=[Kl11 Kl12;Kl21 Kl22];
M=zeros(column+1,column+1);
for ic=1:(column+1)
for Jc=1:(column+1)
M(ic,Jc)=(b^(ic+Jc-1)-a^(ic+Jc-1))/(ic+Jc-1);
end
end
M
```

M = 1 1/2 1/3 1/4 1/2 1/3 1/4 1/5 1/3 1/4 1/5 1/6 1/4 1/5 1/6 1/7

```
Q=Psi*M*Psi'
```

Q = 1 -1/4 1/18 -1/96 -1/4 1/12 -1/48 1/240 1/18 -1/48 1/180 -1/864 -1/96 1/240 -1/864 1/4032

```
Qp=kron(eye(Vk),Q);
I=X*D*Klf*Qp;
W=Pm0*X*D+Pm1*X*Z*D+Pm2*X*Z^2*D+Pm3*X*Z^3*D+Pm4*X*Z^4*D-I;
t0=0;yt0=0;
t1=0;yt1=1/3;
```

```
t2=0;yt2=0;
t3=0;yt3=-1/2;
L0=yt0;
```

```
L1=yt1;
L2=yt2;
L3=yt3;
U0=subs(LLxs(1,:),t0)*D;
U1=subs(LLxs(1,:),t1)*Z*D;
U2=subs(LLxs(2,:),t2)*D;
U3=subs(LLxs(2,:),t3)*Z*D ;
W(Vk*(Co+1)-3,:)=[U0];
W(Vk*(Co+1)-2,:)=[U1];
W(Vk*(Co+1)-1,:)=[U2];
W(Vk*(Co+1),:)=[U3];
WW=W
```

$$WW = \begin{pmatrix} \frac{1}{3} & -\frac{1}{12} & -\frac{17}{54} & -\frac{1}{288} & \frac{1}{4} & -\frac{1}{16} & \frac{1}{72} & -\frac{1}{384} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & 0 \\ \frac{1}{3} & -\frac{1}{12} & -\frac{17}{54} & \frac{23}{288} & \frac{1}{4} & -\frac{1}{16} & \frac{1}{72} & -\frac{1}{384} \\ \frac{1}{54} & -\frac{1}{216} & \frac{1}{972} & -\frac{1}{5184} & -\frac{1}{27} & \frac{1}{108} & -\frac{163}{486} & \frac{217}{2592} \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \end{pmatrix}$$

```
Ga(Vk*(Co+1)-3 ,:)=[L0];
Ga(Vk*(Co+1)-2,:)=[L1];
Ga(Vk*(Co+1)-1 ,:)=[L2];
Ga(Vk*(Co+1),:)=[L3];
GaGa=Ga
```

$$GaGa = \begin{pmatrix} \frac{8}{9} \\ 0 \\ \frac{8}{9} \\ \frac{323}{162} \\ 0 \\ \frac{1}{3} \\ 0 \\ -\frac{1}{2} \end{pmatrix}$$

```
A=WW\GaGa;
syms xs
y=vpa(LLxs*D*A,12);
y_1=y(1);
y_2=y(2);
```

Appendix B

Solving Exsample 1 by Bessel polynomials method

```
clc
clear;Co=3;Vk=2;
Psi=zeros(Co+1,Co+1);
kk=0;
row=0;
p=0;
sg=0;
for ic=1:Co+1
    p=ic-1;
kk=ic-1;
for Jc=ic:2:Co+1
Psi(ic,Jc)=((-1)^(sg))/(factorial(row)*factorial(p)*2^(kk));
row=row+1;
p=p+1;
kk=kk+2;
sg=sg+1;
end
end
Psi
a=0;b=1;
ic=0:Co;
col=a+(b-a)*ic/Co;
Z=zeros(Co+1,Co+1);
for ic=1:Co
Z(ic+1,ic)=ic;
end
Z
syms xs
for ic=1:(Co+1);
Lxs(ic)=xs^(ic-1);
end
Lxs
LLxs=kron(eye(Vk),Lxs);
D=kron(eye(Vk),Psi')
LLxs

Z=kron(eye(Vk),Z');
p0xs=[0 0;0 0];p1xs=[0 0;0 0];p2xs=[1 0;0 1];
p3xs=[0 0;0 0];p4xs=[0 0;0 0];

gxs=[8/9;6*xs-(1/18)*xs^2]
Pm0=kron(eye(Co+1),p0xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
```

```
Pm0(ic,:)=subs(Pm0(ic,:),col(Jc));
end
end

Pm1=kron(eye(Co+1),p1xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm1(ic,:)=subs(Pm1(ic,:),col(Jc));
end
end

Pm2=kron(eye(Co+1),p2xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm2(ic,:)=subs(Pm2(ic,:),col(Jc));
end
end

Pm3=kron(eye(Co+1),p3xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm3(ic,:)=subs(Pm3(ic,:),col(Jc));
end
end

Pm4=kron(eye(Co+1),p4xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm4(ic,:)=subs(Pm4(ic,:),col(Jc));
end
end

for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Ga(ic,1)=subs(gxs(ic-(Jc-1)*Vk,1),col(Jc));
end
end
Ga

for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
X(ic,:)=subs(LLxs(ic-(Jc-1)*Vk,:),col(Jc));
end
end
X
syms xs t
f11=1/3;
for row=0:Co
for column=0:Co
k11(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f11,xs,row),t,column),{xs,t},{0,0});
```

```
end
end
k11;
Kl11=(Psi^-1)'*k11*(Psi^-1);
f12=1/4;
for row=0:Co
for column=0:Co
k12(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f12,xs,row),t,column),{xs,t},{0,0});
end
end
k12;
Kl12=(Psi^-1)'*k12*(Psi^-1);
f21=xs^2/6;
for row=0:Co
for column=0:Co
k21(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f21,xs,row),t,column),{xs,t},{0,0});
end
end
k21;
Kl21=(Psi^-1)'*k21*(Psi^-1);
f22=-xs^2/3;
for row=0:Co
for column=0:Co
k22(row+1,column+1)=(1/(factorial(row)*factorial(column)))*...
subs(diff(diff(f22,xs,row),t,column),{xs,t},{0,0});
end
end
k22;
Kl22=(Psi^-1)'*k22*(Psi^-1);
Klf=[Kl11 Kl12;Kl21 Kl22];
M=zeros(column+1,column+1);
for ic=1:(column+1)
for Jc=1:(column+1)
M(ic,Jc)=(b^(ic+Jc-1)-a^(ic+Jc-1))/(ic+Jc-1);
end
end
M
Q=Psi*M*Psi'
Qp=kron(eye(Vk),Q);
I=X*D*Klf*Qp;
W=Pm0*X*D+Pm1*X*Z*D+Pm2*X*Z^2*D+Pm3*X*Z^3*D+Pm4*X*Z^4*D-I;
t0=0;yt0=0;
t1=0;yt1=1/3;
t2=0;yt2=0;
t3=0;yt3=-1/2;
L0=yt0
L1=yt1
L2=yt2
L3=yt3;
```

```
U0=subs(LLxs(1,:),t0)*D
U1=subs(LLxs(1,:),t1)*Z*D
U2=subs(LLxs(2,:),t2)*D
U3=subs(LLxs(2,:),t3)*Z*D
W(Vk*(Co+1)-3,:)=[U0];
W(Vk*(Co+1)-2,:)=[U1];
W(Vk*(Co+1)-1,:)=[U2];
W(Vk*(Co+1),:)=[U3];
WW=W
Ga(Vk*(Co+1)-3 ,:)=[L0];
Ga(Vk*(Co+1)-2,:)=[L1];
Ga(Vk*(Co+1)-1 ,:)=[L2];
Ga(Vk*(Co+1),:)=[L3];
GaGa=Ga
A=WW\GaGa;
syms xs
y=vpa(LLxs*D*A,12);
y_1=y(1)
y_2=y(2)
y1=@(xs) (1/2)*xs^2+(1/3)*xs;%the exsact solution
y2=@(xs) xs^3-xs/2; %the exsact solutionb
error_y1=abs(y1-y_1);
error_y2=abs(y2-y_2);
c=0;
approxsimatesolution1=zeros(1,10);
approxsimatesolution2=zeros(1,10);
Error_y1=zeros(1,10);
Error_y2=zeros(1,10);
c=0;
for ic=0:0.1:1
    c=c+1;
    cou(c)=ic;
approxsimatesolution1(c)=subs(y_1,ic);
exsactsolution1(c)=y1(ic);
%error_y1(c)=abs(approxsimatesolution1(c)-exsactsolution1(c));
approxsimatesolution2(c)=subs(y_2,ic);
exsactsolution2(c)=y2(ic);
Error_y1(c)=subs(error_y1,ic);
Error_y2(c)=subs(error_y2,ic);
%error_y2(c)=abs(approxsimatesolution2(c)-exsactsolution2(c));

end


figure
hold on
fplot(y_1,[-1 1],'--*b')
fplot(y1,[-1 1],'r')
xslabel ('xs');
ylabel ('y_1(xs)');
legend('Approxsimate solution','Exsact Solution');
```

```
grid on
hold off
figure
hold on
fplot(error_y1,[-1 1],'r')
xslabel ('xs');
ylabel ('error');
legend('The error of y_1 solution');
grid on
hold off

figure
hold on
fplot(y_2,[-1 1],'--*b')
fplot(y2,[-1 1],'r')
xslabel ('xs');
ylabel ('y_2(xs)');
legend('Approxsimate solution','Exsact solution');
grid on
hold off

figure
hold on
fplot(error_y2,[-1 1],'r')
xslabel ('xs');
ylabel ('error');
legend('The error of y_1 solution');
grid on
hold off
Names={ 't';'Approximate y_1(t)'; 'Exsact y1(t)';...
'Error y_1';'Approximate y_2(t)'; 'Exsact y_2(t)';'Error y_2'}
format shortE
Result=[ cou; exsactsolution1;approxsimatesolution1;...
Error_y1;exsactsolution2;approxsimatesolution2;Error_y2].'
%Result=Result.';

result=table(Result)
```

Appendix C

Solving Example 1 by Legendre polynomials method

```
clc %Legendre method
clear;Co=3;
Psi=zeros(Co+1,Co+1);
syms Vk
column=0;
for ic=1:Co+1
for Jc=1:1:Co+1
    if mod(column,2)==0
Psi(ic,Jc)= (1/2^column)*symsum(com(column,Vk)*...
com(2*column-2*Vk,column),Vk,0,column/2);
    else
Psi(ic,Jc)= (1/2^column)*symsum(com(column,Vk)*...
com(2*column-2*Vk,column),Vk,0,(column-1)/2);
    end
column=column+1;
end
end
Psi
Vk=2;
a=0;b=1;
ic=0:Co;
col=a+(b-a)*ic/Co;
Z=zeros(Co+1,Co+1);
for ic=1:Co
Z(ic+1,ic)=ic;
end
Z
syms xs
for ic=1:(Co+1);
Lxs(ic)=xs^(ic-1);
end
Lxs
LLxs=kron(eye(Vk),Lxs);
D=kron(eye(Vk),Psi')
LLxs

Z=kron(eye(Vk),Z');
p0xs=[0 0;0 0];p1xs=[0 0;0 0];p2xs=[1 0;0 1];
p3xs=[0 0;0 0];p4xs=[0 0;0 0];

gxs=[8/9;6*xs-(1/18)*xs^2]
Pm0=kron(eye(Co+1),p0xs);
for Jc=1:(Co+1);
```

```
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm0(ic,:)=subs(Pm0(ic,:),col(Jc));
end
end

Pm1=kron(eye(Co+1),p1xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm1(ic,:)=subs(Pm1(ic,:),col(Jc));
end
end

Pm2=kron(eye(Co+1),p2xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm2(ic,:)=subs(Pm2(ic,:),col(Jc));
end
end

Pm3=kron(eye(Co+1),p3xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm3(ic,:)=subs(Pm3(ic,:),col(Jc));
end
end

Pm4=kron(eye(Co+1),p4xs);
for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Pm4(ic,:)=subs(Pm4(ic,:),col(Jc));
end
end

for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
Ga(ic,1)=subs(gxs(ic-(Jc-1)*Vk,1),col(Jc));
end
end
Ga

for Jc=1:(Co+1);
for ic=Jc*Vk-(Vk-1):Jc*Vk;
X(ic,:)=subs(LLxs(ic-(Jc-1)*Vk,:),col(Jc));
end
end
X
syms xs t
f11=1/3;
for row=0:Co
for column=0:Co
k11(row+1,column+1)=(1/(factorial(row)*factorial(column)))...
```

```
*subs(diff(diff(f11,xs,row),t,column),{xs,t},{0,0});
end
end
k11;
Kl11=(Psi^-1)'*k11*(Psi^-1);
f12=1/4;
for row=0:Co
for column=0:Co
k12(row+1,column+1)=(1/(factorial(row)*factorial(column)))...
*subs(diff(diff(f12,xs,row),t,column),{xs,t},{0,0});
end
end
k12;
Kl12=(Psi^-1)'*k12*(Psi^-1);
f21=xs^2/6;
for row=0:Co
for column=0:Co
k21(row+1,column+1)=(1/(factorial(row)*factorial(column)))...
*subs(diff(diff(f21,xs,row),t,column),{xs,t},{0,0});
end
end
k21;
Kl21=(Psi^-1)'*k21*(Psi^-1);
f22=-xs^2/3;
for row=0:Co
for column=0:Co
k22(row+1,column+1)=(1/(factorial(row)*factorial(column)))...
*subs(diff(diff(f22,xs,row),t,column),{xs,t},{0,0});
end
end
k22;
Kl22=(Psi^-1)'*k22*(Psi^-1);
Klf=[Kl11 Kl12;Kl21 Kl22];
M=zeros(column+1,column+1);
for ic=1:(column+1)
for Jc=1:(column+1)
M(ic,Jc)=(b^(ic+Jc-1)-a^(ic+Jc-1))/(ic+Jc-1);
end
end
M
Q=Psi*M*Psi'
Qp=kron(eye(Vk),Q);
I=X*D*Klf*Qp;
W=Pm0*X*D+Pm1*X*Z*D+Pm2*X*Z^2*D+Pm3*X*Z^3*D+Pm4*X*Z^4*D-I;
t0=0;yt0=0;
t1=0;yt1=1/3;
t2=0;yt2=0;
t3=0;yt3=-1/2;
L0=yt0
L1=yt1
L2=yt2
```

```
L3=yt3;
U0=subs(LLxs(1,:),t0)*D
U1=subs(LLxs(1,:),t1)*Z*D
U2=subs(LLxs(2,:),t2)*D
U3=subs(LLxs(2,:),t3)*Z*D
W(Vk*(Co+1)-3,:)=[U0];
W(Vk*(Co+1)-2,:)=[U1];
W(Vk*(Co+1)-1,:)=[U2];
W(Vk*(Co+1),:)=[U3];
WW=W
Ga(Vk*(Co+1)-3 ,:)=[L0];
Ga(Vk*(Co+1)-2,:)=[L1];
Ga(Vk*(Co+1)-1 ,:)=[L2];
Ga(Vk*(Co+1),:)=[L3];
GaGa=Ga
A=WW\GaGa;
syms xs
y=vpa(LLxs*D*A,12);
y_1=y(1)
y_2=y(2)
y1=@(xs) (1/2)*xs^2+(1/3)*xs;%the exsact solution
y2=@(xs) xs^3-xs/2; %the exsact solutionb
error_y1=abs(y1-y_1);
error_y2=abs(y2-y_2);
c=0;
approxsimatesolution1=zeros(1,10);
approxsimatesolution2=zeros(1,10);
Error_y1=zeros(1,10);
Error_y2=zeros(1,10);
c=0;
for ic=0:0.1:1
    c=c+1;
    cou(c)=ic;
approxsimatesolution1(c)=subs(y_1,ic);
exsactsolution1(c)=y1(ic);
%error_y1(c)=abs(approxsimatesolution1(c)-exsactsolution1(c));
approxsimatesolution2(c)=subs(y_2,ic);
exsactsolution2(c)=y2(ic);
Error_y1(c)=subs(error_y1,ic);
Error_y2(c)=subs(error_y2,ic);
%error_y2(c)=abs(approxsimatesolution2(c)-exsactsolution2(c));

end


figure
hold on
fplot(y_1,[-1 1],'--*b')
fplot(y1,[-1 1],'r')
xslabel ('xs');
ylabel ('y_1(xs)');
```

```
legend('Approxsimate solution','Exsact Solution');
grid on
hold off
figure
hold on
fplot(error_y1,[-1 1],'r')
xslabel ('xs');
ylabel ('error');
legend('The error of y_1 solution');
grid on
hold off

figure
hold on
fplot(y_2,[-1 1],'--*b')
fplot(y2,[-1 1],'r')
xslabel ('xs');
ylabel ('y_2(xs)');
legend('Approxsimate solution','Exsact solution');
grid on
hold off

figure
hold on
fplot(error_y2,[-1 1],'r')
xslabel ('xs');
ylabel ('error');
legend('The error of y_1 solution');
grid on
hold off
Names={ 't';'Approxsimate y_1(t)'; 'Exsact y1(t)';'Error y_1'
    ;...
'Approxsimate y_2(t)'; 'Exsact y_2(t)';'Error y_2'}
format shortE
Result=[ cou; exsactsolution1;approxsimatesolution1;...
Error_y1;exsactsolution2;approxsimatesolution2;Error_y2].'
%Result=Result.';

result=table(Result)
```

جـامعـــة النجاح الوطنية

كليـــة الدراســـات العليـــا

الحلول العددية لنظام معادلات فريدهولم التكاملية التفاضلية

إعداد

أنسام محمد عبد الفتاح زيدان

إشراف

أ.د. ناجي قطناني

قدمت هذه الرسالة استكمالا لمتطلبات الحصول على درجة الماجستير في الرياضيات، من كلية الدراسات العليا، في جامعة النجاح الوطنية، نابلس– فلسطين.

2023

الحلول العددية لنظام معادلات فريدهولم التكاملية التفاضلية

اعداد

أنسام محمد عبد الفتاح زيدان

اشراف

أ.د. ناجي قطناني

الملخص

الخلفية: نظام المعادلات التكاملية التفاضلية فريدهولم موضوع مهم في الرياضيات التطبيقية. على وجه الخصوص، مجال معالجة الإشارات والشبكات العصبية وتطوير الاتصالات.

الهدف: تقدم هذه الأطروحة بعض الطرق العددية التي تعتمد على كثيرات حدود المصفوفة لحل نظام معادلات فريدهولم التكاملية التفاضلية. هذه الطرق هي: كثيرات حدود لاجير، اقتران بيسيل وكثيرات حدود ليجيندر.

المنهجية: تم استخدام برنامج Matlab لتنفيذ هذه الطرق في شكل خوارزميات لحل بعض الأمثلة العددية.

النتائج: تظهر النتائج العددية أن خصائص التقارب والدقة لهذه الطرق تتفق جيدًا مع الحل الدقيق.
الخلاصة: توضح مقارنة النتائج العددية الموضحة في الجداول والأشكال أن طريقة لاجير متعددة الحدود توفر نتائج أكثر دقة من الطرق الأخرى وبالتالي فهي أكثر فعالية.

كلمات مفتاحية: حل النظام، الطرق العددية، المعادلات التفاضلية التكاملية، فريدهولم.