วิธีปริพันธ์อันตะโดยใช้พหุนามเชบีเชฟสำหรับแก้สมการเชิงอนุพันธ์ย่อยเชิงเส้นที่ขึ้นกับเวลาและสมการเชิงอนุพันธ์อันดับเศษส่วนเชิงเส้น

นายอานนท์ แสงศิริทองไชย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2561

FINITE INTEGRATION METHOD USING CHEBYSHEV POLYNOMIALS FOR SOLVING TIME-DEPENDENT LINEAR PARTIAL DIFFERENTIAL EQUATIONS AND LINEAR FRACTIONAL ORDER DIFFERENTIAL EQUATIONS

Mr. Arnont Saengsiritongchai

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Applied Mathematics and Computational Science
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2018

Thesis Title	FINITE INTEGRATION METHOD USING CHEBYSHEV POLYNOMIALS FOR SOLVING TIME-DEPENDENT LINEAR PARTIAL DIFFERENTIAL EQUATIONS AND LINEAR FRACTIONAL ORDER DIFFERENTIAL EQUATIONS
By	Mr. Arnont Saengsiritongchai
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Assistant Professor Ratinan Boonklurb, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

. . . . Dean of the Faculty of Science

(Professor Polkit Sangvanich, Ph.D.)

THESIS COMMITTEE

. . . . Chairman

(Associate Professor Anusorn Chonwerayuth, Ph.D.)

. . . . Thesis Advisor

(Assistant Professor Ratinan Boonklurb, Ph.D.)

. . . . Examiner

(Assistant Professor Petarpa Boonserm, Ph.D.)

. . . . External Examiner

(Assistant Professor Tawikan Treeyaprasert, Ph.D.)

อานนท์ แสงศิริทองไชย : วิธีปริพันธ์อันตะโดยใช้พหุนามเชบีเชฟสำหรับแก้สมการเชิงอนุพันธ์ย่อยเชิงเส้นที่ขึ้นกับเวลาและสมการเชิงอนุพันธ์อันดับเศษส่วนเชิงเส้น. (FINITE INTEGRATION METHOD USING CHEBYSHEV POLYNOMIALS FOR SOLVING TIME-DEPENDENT LINEAR PARTIAL DIFFERENTIAL EQUATIONS AND LINEAR FRACTIONAL ORDER DIFFERENTIAL EQUATIONS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผู้ช่วยศาสตราจารย์ ดร. รตินันท์ บุญเคลือบ, 111 หน้า.

วิทยานิพนธ์ฉบับนี้ เราสร้างขั้นตอนวิธีเชิงตัวเลขที่อาศัยวิธีการหาปริพันธ์อันตะโดยใช้พหุนามเชบีเชฟเพื่อหาผลเฉลยเชิงตัวเลขของสมการเชิงอนุพันธ์ย่อยเชิงเส้นที่ขึ้นกับเวลาและสมการเชิงอนุพันธ์อันดับเศษส่วนเชิงเส้น ผลลัพธ์สำหรับสมการเชิงอนุพันธ์ย่อยที่ขึ้นกับเวลาปรากฏว่าขั้นตอนวิธีที่สร้างขึ้นให้ค่าประมาณที่ดีกว่ามากเมื่อเปรียบเทียบกับวิธีปริพันธ์อันตะแบบดั้งเดิม หลายตัวอย่างแสดงให้เห็นว่าขั้นตอนวิธีสำหรับสมการเชิงอนุพันธ์อันดับเศษส่วนเชิงเส้นให้ค่าประมาณของผลเฉลยที่ดี

ภาควิชา	คณิตศาสตร์และ	ลายมือชื่อนิสิต ....
	วิทยาการคอมพิวเตอร์	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....
สาขาวิชา	คณิตศาสตร์ประยุกต์	
	และวิทยาการคณนา	
ปีการศึกษา	2561	

5972100123 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORDS : CHEBYSHEV POLYNOMIAL / FINITE INTEGRATION METHOD / TIME-DEPENDENT / FRACTIONAL ORDER DIFFERENTIAL EQUATIONS

ARNONT SAENGSIRITONGCHAI : FINITE INTEGRATION METHOD USING CHEBYSHEV POLYNOMIALS FOR SOLVING TIME-DEPENDENT LINEAR PARTIAL DIFFERENTIAL EQUATIONS AND LINEAR FRACTIONAL ORDER DIFFERENTIAL EQUATIONS. ADVISOR : ASSISTANT PROFESSOR RATINAN BOONKLURB, Ph.D., 111 pp.

In this thesis, we devise numerical algorithms base on the finite integration method (FIM) using Chebyshev polynomial to find numerical solutions of time-dependent linear partial differential equations and linear fractional order differential equations. The results show that for time-dependent linear partial differential equations, our algorithm gives a lot better accuracy than the traditional FIMs. Several examples illustrate that our algorithm for linear fractional order differential equations gives good approximate solution as well.

Department	: Mathematics and Computer Science	Student's Signature	....
		Advisor's Signature	....
Field of Study	: Applied Mathematics and Computational Science		
Academic Year	: 2018		

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor Assistant Professor Ratinan Boonklurb, Ph.D. who taught me with his kindness, patience and all advice on my research. It would not have been complete without his suggestion. In addition, I would like to thanks Mr. Ampol Duangpan for helping me and teaching me fundamental knowledge for this thesis.

I would like to express my appreciation to all of my thesis committee members, Associate Professor Anusorn Chonwerayuth, Ph.D., Assistant Professor Petarpa Boonserm, Ph.D. and Assistant Professor Tawikan Treeyaprasert, Ph.D., for their advice and comments on this thesis

I would like to thank my family and all my friends for supporting and encouraging me at all time.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Motivation and Literature Surveys

The finite integration method (FIM) is one of the recent developed numerical techniques for finding approximate solutions to boundary value problems for ordinary and partial differential equations. Similar to the finite difference method (FDM), we replace the solution domain with a finite number of points known as grid points, and obtain the approximate solution at these points. The grids are generally spaced along the independent coordinates.

The important tool in FIM is the finite integration matrix. Traditionally, the finite integration matrix can be obtained by direct numerical integrations, provided by Wen et al. [12], using both standard trapezoidal integral algorithm and radial basis functions. After that Li et al. [8] used the FIM to solve multi-dimentional problems. Recently, Li et al. [9] developed the FIM for solving partial differential equations (PDEs) by using the Simpson, Cotes and Lagrange formula. In 2016, Duangpan [4] modified the traditional FIM by using the integration of Chebyshev polynomials instead of trapezoidal ([8] and [12]), Simpson, Cotes and Lagrange ([8]) to approximate the solution of linear ordinary differential equations (ODEs) and the steady state linear PDEs. His algorithms gave significant better results compared to all traditional FIMs. However, Duangpan's method cannot be applied directly to the problems depending on time and problems involving linear fractional derivatives.

In this thesis, we would like to construct algorithms for finding approximate solutions of the time-dependent linear PDEs in one- and two-dimensions base on the FIM using Chebyshev polynomials. For the node points, we use the zeros of Chebyshev polynomials which are roots of the Chebyshev polynomial of some order. Moreover, we also construct an algorithm for finding approximate solutions of linear fractional order differential equations base on the FIM using shifted Chebyshev polynomials. For the node points, we use the zeros of shifted Chebyshev polynomials which are roots of the shifted Chebyshev polynomial of some order. We test our algorithms on several examples. In each example for the time-dependent linear PDEs, we implement the MatLab program to compare our results with the results obtained by other traditional FIMs and their analytical solutions. For linear FDEs, we compare our results with the results obtained by other methods from literatures and their analytical solutions.

1.2 Research Objectives

The goals of this research are to construct the FIM using Chebyshev polynomials for solving time-dependent linear PDEs in one- and two-dimensions that gives a higher accuracy than the traditional FIMs when the number of nodes are equal and to construct the FIM using shifted Chebyshev polynomials for solving linear FDEs that gives a good accuracy.

1.3 Thesis Overview

This thesis consists of 5 chapters and is organized as follows. Chapter I is an introduction and the motivation of this thesis, the research objectives and the thesis overview. Chapter II presents preliminaries knowledge used in this thesis, which includes Chebyshev polynomials, shifted Chebyshev polynomials, linear ODEs, second order linear PDEs, linear FDEs, finite integration matrix in one-dimensional space and two-dimensional space. In Chapter III, we present the procedure for solving time-dependent linear PDEs in one- and two-dimensions and provide some numerical examples. In Chapter IV, we present the procedure for linear FDEs via Riemann-Liouville type and provide some numerical examples. Finally, Chapter V consists of conclusion of this thesis and discussion about possibly future work.

CHAPTER II

PRELIMINARIES

In this chapter, background knowledge on the definition and properties of the Chebyshev polynomials and shifted Chebyshev polynomials are provided. The form of time-dependent second order linear PDEs is given. The definitions for fractional order derivatives and the form of linear FDEs are presented. Finally, for ease of reference, we present the details of constructing the finite integration matrices in one- and two-dimensional spaces similar to those obtained by Daungpan [4].

2.1 Chebyshev Polynomials

Definition 2.1. [6] The Chebyshev polynomial of degree $n \geq 0$ is defined as

$$T_n(x) = \cos(n \arccos x), \text{ for } x \in [-1, 1]. \tag{2.1}$$

The Chebyshev polynomials are orthogonal polynomials which play an important role in the theory of approximation. Their roots are used as nodes to construct a polynomial interpolation which provides the best approximation under the maximum norm. The first few Chebyshev polynomials $T_n(x)$ are illustrated in Figure 2.1 for $n \in \{0, 1, 2, 3, 4, 5\}$. It can be seen that the zeros of Chebyshev polynomials are not equally distributed over $[-1, 1]$.

To construct the first and higher order integration matrices as well as to construct the procedure for our FIM using Chebyshev polynomial, we need the following properties.

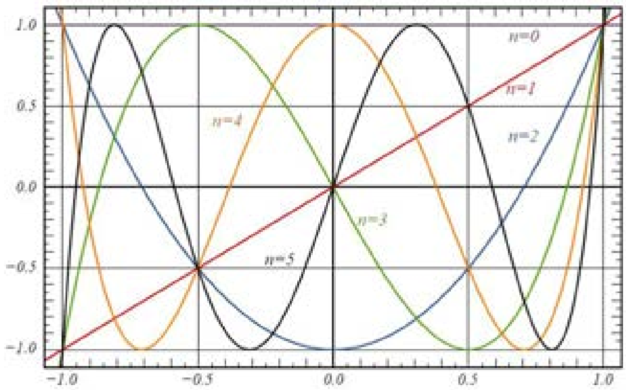


Figure 2.1**:** Chebyshev polynomials $T_n(x)$ for $n \in \{0, 1, 2, 3, 4, 5\}$.

Lemma 2.2. *(i) For $n \in \mathbb{N}$ and $x \in [-1, 1]$, the recurrence relation among Chebyshev polynomials of degree $n-1, n$ and $n+1$ is*

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x),$$

with the starting values $T_0(x) = 1$ and $T_1(x) = x$.

(ii) For $n \in \mathbb{N}$, the zeros of Chebyshev polynomial $T_n(x)$ for $x \in [-1, 1]$ are

$$x_k = \cos\left[\left(\frac{2k-1}{2n}\right)\pi\right], \quad k = \{1, 2, 3, ..., n\}. \tag{2.2}$$

(iii) For $x \in [-1, 1]$, the single integrations of Chebyshev $T_n(x)$ are

$$\bar{T}_0(x) := \int_{-1}^{x} T_0(\xi)d\xi = x + 1, \tag{2.3}$$

$$\bar{T}_1(x) := \int_{-1}^{x} T_1(\xi)d\xi = \frac{x^2 - 1}{2}, \tag{2.4}$$

$$\begin{aligned} \bar{T}_n(x) &:= \int_{-1}^{x} T_n(\xi)d\xi \\ &= \frac{1}{2}\left(\frac{T_{n+1}(x)}{n+1} - \frac{T_{n-1}(x)}{n-1}\right) - \frac{(-1)^n}{n^2 - 1} \textit{ for } \; n \geq 2. \end{aligned} \tag{2.5}$$

(iv) For $n \in \mathbb{N}$, the discrete orthogonality relation of Chebyshev polynomials $T_i(x_k)$ and

$T_j(x_k)$ is

$$\sum_{k=1}^{n} T_i(x_k)T_j(x_k) = \begin{cases} 0 & if \quad i \neq j, \\ n & if \quad i = j = 0, \\ \frac{n}{2} & if \quad i = j \neq 0, \end{cases} \tag{2.6}$$

where x_k, $k \in \{1, 2, 3, ..., n\}$, is defined by (2.2), and $0 \leq i, j \leq n$.

(v) Let the Chebyshev matrix $\boldsymbol{T}$ at each node $\{x_k\}_{k=1}^n$, defined by (2.2), be

$$\boldsymbol{T} = \begin{bmatrix} T_0(x_1) & T_1(x_1) & T_2(x_1) & \dots & T_{n-1}(x_1) \\ T_0(x_2) & T_1(x_2) & T_2(x_2) & \dots & T_{n-1}(x_2) \\ T_0(x_3) & T_1(x_3) & T_2(x_3) & \dots & T_{n-1}(x_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_0(x_n) & T_1(x_n) & T_2(x_n) & \dots & T_{n-1}(x_n) \end{bmatrix}.$$

Then, it has the multiplicative inverse as $(\boldsymbol{T})^{-1} = \frac{1}{n} diag[1, 2, 2, ..., 2](\boldsymbol{T})^T$.

Proof. (i) From (2.1), it is clear that if $x = \cos\theta$, then $T_n(\cos\theta) = \cos(n\theta)$, $\theta \in [0, \pi]$. Let $n \in \mathbb{N}$. By applying the trigonometric addition identities, we obtain

$$\begin{aligned} T_{n+1}(x) + T_{n-1}(x) &= \cos[(n+1)\theta] + \cos[(n-1)\theta] \\ &= \cos(n\theta)\cos\theta - \sin(n\theta)\sin\theta + \cos(n\theta)\cos\theta + \sin(n\theta)\sin\theta \\ &= 2xT_n(x). \end{aligned}$$

(ii) It is done by solving $\cos(n \arccos x) = 0$ for x directly.

(iii) Let $x \in [-1, 1]$. It is easy to see for the cases $n = 0$ and $n = 1$. For $n \geq 2$, we have

$$\begin{aligned} \bar{T}_n(x) &:= \int_{-1}^{x} T_n(\xi)d\xi \\ &= \int_{\arccos(-1)}^{\arccos x} T_n(\cos\theta)d(\cos\theta) \\ &= \int_{\arccos(-1)}^{\arccos x} -\cos(n\theta)\sin\theta d\theta \\ &= \int_{\arccos(-1)}^{\arccos x} -\frac{1}{2}(\sin((n+1)\theta) - \sin((n-1)\theta))d\theta \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\Big(\frac{\cos(n+1)\theta}{n+1} - \frac{\cos(n-1)\theta}{n-1}\Big)_{\theta=\arccos(-1)}^{\theta=\arccos x}\\
&= \frac{1}{2}\Big(\frac{T_{n+1}(\xi)}{n+1} - \frac{T_{n-1}(\xi)}{n-1}\Big)_{\xi=-1}^{\xi=x}\\
&= \frac{1}{2}\left(\frac{T_{n+1}(x)}{n+1} - \frac{T_{n-1}(x)}{n-1}\right) - \frac{(-1)^n}{n^2-1}.
\end{aligned}
$$

(iv) Recall the trigonometric identity [2],

$$
\sum_{k=1}^{n}\cos(\alpha+\beta k) = \frac{\sin\frac{(n+1)\beta}{2}\cos\left(\alpha+\dfrac{n\beta}{2}\right)}{\sin\dfrac{\beta}{2}}. \tag{2.7}
$$

For $0 \le i, j \le n$ and $i \ne j$, by (2.1), (2.2) and (2.7), we obtain

$$
\begin{aligned}
\sum_{k=1}^{n} T_i(x_k)T_j(x_k) &= \sum_{k=1}^{n}\cos\frac{i(2k-1)\pi}{2n}\cos\frac{j(2k-1)\pi}{2n}\\
&= \frac{1}{2}\sum_{k=1}^{n-1}\left(\cos\frac{(i+j)(2k-1)\pi}{2n} + \cos\frac{(i-j)(2k-1)\pi}{2n}\right) \qquad (2.8)\\
&= \frac{1}{2}\sum_{k=1}^{n-1}\left(\cos\Big(\frac{(i+j)\pi}{2n} + \frac{(i+j)\pi}{n}k\Big) + \cos\Big(\frac{(i-j)\pi}{2n} + \frac{(i-j)\pi}{n}k\Big)\right)\\
&= \frac{1}{2}\left(\frac{\sin\frac{n(i+j)\pi}{2n}\cos\Big(\frac{(i+j)\pi}{2n} + \frac{(n-1)(i+j)\pi}{2n}\Big)}{\sin\frac{(i+j)\pi}{2n}}\right.\\
&\qquad \left. + \frac{\sin\frac{n(i-j)\pi}{2n}\cos\Big(\frac{(i-j)\pi}{2n} + \frac{(n-1)(i-j)\pi}{2n}\Big)}{\sin\frac{(i-j)\pi}{2n}}\right)\\
&= \frac{1}{2}\left(\frac{\sin\frac{(i+j)\pi}{2}\cos\frac{(i+j)\pi}{2}}{\sin\frac{(i+j)\pi}{2n}} + \frac{\sin\frac{(i-j)\pi}{2}\cos\frac{(i-j)\pi}{2}}{\sin\frac{(i-j)\pi}{2n}}\right)\\
&= \frac{1}{4}\left(\frac{\sin(i+j)\pi}{\sin\frac{(i+j)\pi}{2n}} + \frac{\sin(i+j)\pi}{\sin\frac{(i-j)\pi}{2n}}\right) = 0.
\end{aligned}
$$

Next, for $i = j = 0$, we have from (2.8) that

$$
\sum_{k=1}^{n} T_0(x_k)T_0(x_k) = \frac{1}{2}\sum_{k=1}^{n-1}\left(\cos\frac{(0+0)(2k-1)\pi}{2n} + \cos\frac{(0-0)(2k-1)\pi}{2n}\right) = n.
$$

Finally, for $0 < i \le n$, we have from (2.8) that

$$\begin{aligned}\sum_{k=1}^{n} T_i(x_k)T_i(x_k) &= \frac{1}{2}\sum_{k=1}^{n-1}\left(\cos\frac{(i+i)(2k-1)\pi}{2n} + \cos\frac{(i-i)(2k-1)\pi}{2n}\right)\\ &= \frac{1}{2}\sum_{k=0}^{n-1}\left(\cos\frac{i(2k-1)\pi)}{n} + 1\right) = \frac{1}{2}(0+n) = \frac{n}{2}.\end{aligned}$$

(v) We can prove directly by letting $\mathbf{Q} = \frac{1}{n}\text{diag}[1, 2, 2, ..., 2]\mathbf{T}^T$ and calculating $\mathbf{QT}$ and $\mathbf{TQ}$ by using Lemma 2.2. (iv) to obtain that they are identity matrices. □

2.2 Shifted Chebyshev Polynomials

In some applications, the interval $[0, 1]$ is more convenient to use than $[-1, 1]$. Thus, we transform the independent variable of $T_n(x)$ from $[-1, 1]$ to x in $[0, 1]$ by the transformation $s = 2x-1$ or $x = \frac{1}{2}(s+1)$. The polynomial obtained after transformation is called a shifted Chebyshev polynomial $T_n^*(x)$ of degree n for $x \in [0, 1]$. That is

$$T_n^*(x) = T_n(2x-1). \tag{2.9}$$

The following lemma is the same as Lemma 2.2 but it is stated in terms of shifted Chebyshev polynomials. Thus, we provide here Lemma 2.3 without proof.

Lemma 2.3. *(i) For $n \in \mathbb{N}$ and $x \in [0, 1]$, the recurrence relation among shifted Chebyshev polynomials of degree $n-1, n$ and $n+1$ is*

$$T_{n+1}^*(x) = 2(2x-1)T_n^*(x) - T_{n-1}^*(x)$$

with the starting values $T_0^*(x) = 1$ and $T_1^*(x) = 2x-1$.

(ii) For $n \in \mathbb{N}$, the zeros of shifted Chebyshev polynomial $T_n^*(x)$ for $x \in [0, 1]$ are

$$x_k = \frac{1}{2}\Big(\cos\Big(\Big(\frac{2k-1}{2n}\Big)\pi\Big) + 1\Big), \quad k \in \{1, 2, 3, ..., n\}. \tag{2.10}$$

(iii) For $x \in [0,1]$, the single integrations of shifted Chebyshev $T_n^*(x)$ are

$$\begin{aligned} \bar{T}_0^*(x) &:= x, && (2.11)\\ \bar{T}_1^*(x) &:= x^2 - x, && (2.12)\\ \bar{T}_n^*(x) &:= \frac{1}{4}\Big(\frac{T_{n+1}^*(x)}{n+1} - \frac{T_{n-1}^*(x)}{n-1}\Big) - \frac{(-1)^n}{2(n^2-1)} \ \textit{for} \ \ n \geq 2. && (2.13)\end{aligned}$$

(iv) For $n \in \mathbb{N}$, the discrete orthogonality relation of shifted Chebyshev polynomials $T_i^*(x_k)$ and $T_j^*(x_k)$ is

$$\sum_{k=1}^{n} T_i^*(x_k)T_j^*(x_k) = \begin{cases} 0 & if \quad i \neq j, \\ n & if \quad i = j = 0, \\ \frac{n}{2} & if \quad i = j \neq 0, \end{cases} \tag{2.14}$$

where x_k, $k \in \{1,2,3,...,n\}$, is defined by (2.10), and $0 \leq i,j \leq n$.

(v) *Let the shifted Chebyshev matrix $\mathbf{T}^*$ at each node $\{x_k\}_{k=1}^n$, defined by (2.10), be*

$$\mathbf{T}^* = \begin{bmatrix} T_0^*(x_1) & T_1^*(x_1) & T_2^*(x_1) & \dots & T_{n-1}^*(x_1) \\ T_0^*(x_2) & T_1^*(x_2) & T_2^*(x_2) & \dots & T_{n-1}^*(x_2) \\ T_0^*(x_3) & T_1^*(x_3) & T_2^*(x_3) & \dots & T_{n-1}^*(x_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_0^*(x_n) & T_1^*(x_n) & T_2^*(x_n) & \dots & T_{n-1}^*(x_n) \end{bmatrix}.$$

Then, by the discrete orthogonality relation of shifted Chebyshev polynomials, it has the multiplicative inverse given by $(\mathbf{T}^*)^{-1} = \frac{1}{n}\mathrm{diag}[1,2,2,...,2](\mathbf{T}^*)^T$.

2.3 Linear ODEs

Let a and b are real numbers such that $a < b$. A linear ODE of order n in the dependent variable u and the independent variable $x \in (a,b)$ is an equation that can be expressed in the form

$$\alpha_n(x)u^{(n)} + \alpha_{n-1}(x)u^{(n-1)} + \alpha_{n-2}(x)u^{(n-2)} + ... + \alpha_0(x)u = f(x), \tag{2.15}$$

where $\alpha_n(x)$ is not identically zero. We shall assume that $\alpha_n, \alpha_{n-1}, \alpha_{n-2}, ..., \alpha_0$ and f are continuous real functions on (a, b). This n^{th} order linear ODE comes with initial conditions which determined at $x = a$ only or boundary conditions which determined at $x = a$ and $x = b$. The right-hand-side function f is called the non-homogeneous term. However, if f is identically zero, (2.15) is called homogeneous linear ordinary differential equation.

2.4 Second Order Linear PDEs

The general second order linear PDEs in two independent variables x and y is an equation of the form

$$\alpha_1\frac{\partial^2 u}{\partial x^2} + \alpha_2\frac{\partial^2 u}{\partial x\partial y} + \alpha_3\frac{\partial^2 u}{\partial y^2} + \alpha_4\frac{\partial u}{\partial x} + \alpha_5\frac{\partial u}{\partial y} + \alpha_6 u = \beta, \quad (x, y) \in \Omega \subseteq \mathbb{R} \tag{2.16}$$

The parameters $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$ and β can be functions of x and y, where α_1, α_2 and α_3 are not all 0. Moreover, the first three terms of (2.16) are called the principal part of the PDE which its coefficients can be used to classify the PDEs as follows:

if $\alpha_2^2 - 4\alpha_1\alpha_3 < 0$, then (2.16) is called the elliptic PDE;

if $\alpha_2^2 - 4\alpha_1\alpha_3 = 0$, then (2.16) is called the parabolic PDE;

if $\alpha_2^2 - 4\alpha_1\alpha_3 > 0$, then (2.16) is called the hyperbolic PDE.

For the second order linear PDEs, there are three types of boundary conditions come with the problems as follows:

1. Dirichlet boundary condition determines the value of the function;
2. Neumann boundary determines the value of the normal derivative of the function, $\frac{\partial u}{\partial n}$;
3. Robin boundary determines the value of the sum of function and its normal derivative, $au + b\frac{\partial u}{\partial n}$ where a and b are constants.

However, in this thesis, we consider only the Dirichlet boundary condition and the time-dependent linear second order PDEs of the form

$$\frac{\partial u}{\partial t} = \alpha_1(x,t)\frac{\partial^2 u}{\partial x^2} + \alpha_2(x,t)\frac{\partial u}{\partial x} + \alpha_3(x,t)u + f(x,t)$$

for one-dimensional and

$$\begin{aligned}\frac{\partial u}{\partial t} &= \alpha_1(x,y,t)\frac{\partial^2 u}{\partial x^2} + \alpha_2(x,y,t)\frac{\partial^2 u}{\partial y^2} + \alpha_3(x,y,t)\frac{\partial^2 u}{\partial x \partial y} + \alpha_4(x,y,t)\frac{\partial u}{\partial x} + \alpha_5(x,y,t)\frac{\partial u}{\partial y} \\ &\quad + \alpha_6(x,y,t)u + f(x,y,t)\end{aligned}$$

for two-dimensional.

2.5 Linear FDEs

Fractional derivative is a derivative which has order as a fraction instead of an integer. Many researchers gave definitions for fractional derivatives. Each definition involves both local and global properties. For examples, the Riemann-Liouville and Caputo definitions involve the local property but the Grunwald-Letnikov definition involves the global property (see [7] and [10] for further details).

In this thesis, we use the Riemann-Liouville definition of fractional derivative as follow.

Definition 2.4. [7] Let m be a positve integer, $\alpha \in (m-1, m)$, b be a positive real number and $x \in [0, b]$. The Riemann-Liouville fractional derivative of order α of a function u is

$$D^\alpha u(x) = \frac{1}{\Gamma(m-\alpha)}\frac{d^m}{dx^m}\int_0^x \frac{u(s)}{(x-s)^{\alpha-m+1}}ds, \qquad (2.17)$$

where $\Gamma(a)$ is a gamma function and $u \in L^1(0,b) = \{u \mid \int_0^b |u|dx < \infty\}$.

A linear FDE of order $m \in \mathbb{N}$ with fractional order α in the dependent variable u

and the independent variable x is an equation that can be expressed in the form

$$D^{\alpha}u(x) + a_m(x)u^{(m)} + a_{m-1}(x)u^{(m-1)} + a_{m-2}(x)u^{(m-2)} + ... + a_0(x)u = f(x),$$

where α is a real number in $(m-i, m-i+1)$, $i \in \{1, 2, 3, ..., m\}$ and D^{α} is an operator for fractional derivative of order α. We shall assume that $a_m(x)$, $a_{m-1}(x)$, $a_{m-2}(x)$,..., $a_0(x)$ and $f(x)$ are continuous real functions on $(0, b)$. We note that, in this thesis we consider only the second order linear FDEs.

Recently, FDEs can be found in various problems such as time delay problem, tautochrone problem, viscoelastic materials, fluid flow, diffusive transport, etc. (see [5] and [10] for further details).

2.6 Finite Integration Matrix in One-Dimensional Space

In this section, we explain how to construct the first order integration matrix based on the Chebyshev polynomials. The m^{th} order integration matrix can be obtained easily as a consequence of the first order. For $N \in \mathbb{N}$, we first approximate the solution of (2.15) by

$$u(x_k) = \sum_{n=0}^{N-1} c_n T_n(x_k),$$

where $x_k \in [-1, 1]$ are the node points defined by (2.2) and T_n is the Chebyshev polynomial of degree n. The approximate solution can be written in the matrix form as (2.18),

$$\begin{bmatrix} u(x_1) \\ u(x_2) \\ u(x_3) \\ \vdots \\ u(x_k) \end{bmatrix} = \begin{bmatrix} T_0(x_1) & T_1(x_1) & T_2(x_1) & \dots & T_{N-1}(x_1) \\ T_0(x_2) & T_1(x_2) & T_2(x_2) & \dots & T_{N-1}(x_2) \\ T_0(x_3) & T_1(x_3) & T_2(x_3) & \dots & T_{N-1}(x_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_0(x_N) & T_1(x_N) & T_2(x_N) & \dots & T_{N-1}(x_N) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_3 \\ \vdots \\ c_{N-1} \end{bmatrix}, \tag{2.18}$$

which is denoted by **u**=**Tc**. Thus, **c**=$\mathbf{T}^{-1}$**u**. Then, we consider

$$U(x_k) \;=\; \int_{-1}^{x_k} u(\xi)d\xi = \sum_{n=0}^{N-1} c_n \int_{-1}^{x_k} T_n(\xi)d\xi = \sum_{n=0}^{N-1} c_n \bar{T}_n(x_k),$$

where $k \in \{1, 2, 3, ..., N\}$ and $\bar{T}_n(x)$ is defined by (2.3), (2.4) and (2.5) or in matrix form,

$$\begin{bmatrix} U(x_1) \\ U(x_2) \\ U(x_3) \\ \vdots \\ U(x_N) \end{bmatrix} = \begin{bmatrix} \bar{T}_0(x_1) & \bar{T}_1(x_1) & \bar{T}_2(x_1) & \dots & \bar{T}_{N-1}(x_1) \\ \bar{T}_0(x_2) & \bar{T}_1(x_2) & \bar{T}_2(x_2) & \dots & \bar{T}_{N-1}(x_2) \\ \bar{T}_0(x_3) & \bar{T}_1(x_3) & \bar{T}_2(x_3) & \dots & \bar{T}_{N-1}(x_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{T}_0(x_N) & \bar{T}_1(x_N) & \bar{T}_2(x_N) & \dots & \bar{T}_{N-1}(x_N) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_3 \\ \vdots \\ c_{N-1} \end{bmatrix}, \tag{2.19}$$

which is denoted by $\mathbf{U} = \bar{\mathbf{T}}\mathbf{c} = \bar{\mathbf{T}}\mathbf{T}^{-1}\mathbf{u} = \mathbf{A}\mathbf{u}$. This $\mathbf{A} = [a_{ki}]_{N\times N}$ is called the *first order integration matrix*, i.e.,

$$U(x_k) = \int_{-1}^{x_k} u(\xi)d\xi = \sum_{i=1}^{N} a_{ki}u(x_i),$$

or in matrix form,

$$\begin{bmatrix} U(x_1) \\ U(x_2) \\ U(x_3) \\ \vdots \\ U(x_N) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1N} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2N} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N1} & a_{N3} & \dots & a_{NN} \end{bmatrix} \begin{bmatrix} u(x_1) \\ u(x_2) \\ u(x_3) \\ \vdots \\ u(x_k) \end{bmatrix}. \tag{2.20}$$

Now, consider the double layer integration of $u(x)$

$$\begin{aligned} U^{(2)}(x_k) \;&:=\; \int_{-1}^{x_k}\int_{-1}^{\xi_2} u(\xi_1)d\xi_1 d\xi_2 = \sum_{i=1}^{N} a_{ki} \int_{-1}^{x_i} u(\xi_1)d\xi_1 \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ki}a_{ij}u(x_j) = \sum_{i=1}^{N} a_{ki}^{(2)} u(x_i) \end{aligned}$$

which can be written in matrix form as $\mathbf{U}^{(2)} = \mathbf{A}^{(2)}\mathbf{u} = \mathbf{A}^2\mathbf{u}$. Similarly, for multi-layer integration of $u(x)$, we have

$$
\begin{aligned}
U^{(m)}(x_k) &:= \int_{-1}^{x_k} \cdots \int_{-1}^{\xi_4} \int_{-1}^{\xi_3} \int_{-1}^{\xi_2} u(\xi_1) d\xi_1 d\xi_2 d\xi_3 \ldots d\xi_m \\
&= \sum_{i_m=1}^{N} \cdots \sum_{i_2=1}^{N} \sum_{i_1=1}^{N} \sum_{j=1}^{N} a_{ki_m} a_{i_1 j} u(x_j) \\
&= \sum_{i=1}^{N} a_{ki}^{(m)} u(x_i) \qquad (2.21)
\end{aligned}
$$

or in matrix form $\mathbf{U}^{(m)} = \mathbf{A}^{(m)}\mathbf{u} = \mathbf{A}^m\mathbf{u}$ and $\mathbf{A}^{(m)} = \mathbf{A}^m$ is called an m^{th} *order integration matrix.*

To construct the first and m^{th} finite integration matrices by using the shifted Chebyshev polynomial, we use the same procedure as we did for constructing the first and m^{th} finite integration matrices by using the Chebyshev polynomial. Then, the first shifted finite integration matrix is defined by $\mathbf{A}^* = \bar{\mathbf{T}}^*(\mathbf{T}^*)^{-1}$ and the m^{th} shifted finite integration method is calculated by $(\mathbf{A}^*)^{(m)} = (\mathbf{A}^*)^m$.

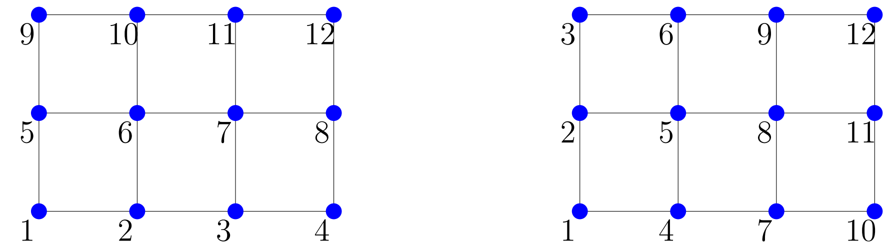
2.7 Finite Integration Matrix in Two-Dimensional Space

In this section, we explain how to construct the first order integration matrix for two-dimensional problems. Then, the higher order integration matrix can be obtained easily in the same manner as we did for one-dimensional space. We consider the second order linear PDE (2.16) under the Dirichelet boundary condition $u(x, y) = \omega(x, y)$, $(x, y) \in \partial\Omega$.

Let a, b, c, d be real numbers such that $a < b$ and $c < d$. We assume that our domain is $\Omega = [a, b] \times [c, d]$ and we transform it into $\bar{\Omega} = [-1, 1] \times [-1, 1]$ which can be discretized by the zeros of the Chebyshev polynomial with the number of points $M = N_1 \times N_2$, where N_1 and N_2 are the total number of horizontal and vertical discretized nodes, respectively. Let $-1 < x_1 < x_2 < x_3 < ... < x_{N_1} < 1$ and $-1 < y_1 < y_2 < y_3 < ... < y_{N_2} < 1$ be grid points that are generated by the zeros of Chebyshev polynomials.

For computational convenience, we index numbering of grid points along the x-

direction by the global numbering system (Figure 2.2(a)) and grid points along y-direction by the local numbering system (Figure 2.2(b)).



(a) Global numbering system **(b)** Local numbering system

Figure 2.2: The indices of the grid points globally and locally.

Let $U_x(x, y)$ and $U_y(x, y)$ be the single layer integrations with respect to variables x and y, respectively. Consider $U_x(x_k, y_s)$ when y_s is fixed. Then, by the idea in Section 2.6, we have

$$U_x(x_k, y_s) \quad := \quad \int_{-1}^{x_k} u(\xi, y_s)d\xi = \sum_{i=1}^{N_1} a_{ki}u(x_i, y_s)$$

for $k \in \{1, 2, 3, ..., N_1\}$. Let $\mathbf{U}_x(\cdot, y_s) = [U_x(x_1, y_s), U_x(x_2, y_s), U_x(x_3, y_s), ..., U_x(x_{N_1}, y_s)]^T$ and $\mathbf{u}(\cdot, y_s) = [u(x_1, y_s), u(x_2, y_s), u(x_3, y_s), ..., u(x_{N_1}, y_s)]^T$. Then, for $s \in \{1, 2, 3, ..., N_2\}$, $\mathbf{U}_x(\cdot, y_s) = \mathbf{A}\mathbf{u}(\cdot, y_s)$ and

$$\begin{bmatrix} \mathbf{U}_x(\cdot, y_1) \\ \mathbf{U}_x(\cdot, y_2) \\ \mathbf{U}_x(\cdot, y_3) \\ \vdots \\ \mathbf{U}_x(\cdot, y_{N_2}) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{A} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{A} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{A} \end{bmatrix}}_{N_2 \text{ blocks}} \begin{bmatrix} \mathbf{u}(\cdot, y_1) \\ \mathbf{u}(\cdot, y_2) \\ \mathbf{u}(\cdot, y_3) \\ \vdots \\ \mathbf{u}(\cdot, y_{N_2}) \end{bmatrix}, \tag{2.22}$$

where $\mathbf{A} = \bar{\mathbf{T}}\mathbf{T}^{-1}$ is an $N_1 \times N_1$ matrix. We will denote (2.22) by $\mathbf{U}_x = \mathbf{A}_x\mathbf{u}$.

Consider $U_y(x_k, y_s)$ when x_k is fixed. Then, by the idea in Section 2.6, we have

$$U_y(x_k, y_s) \quad := \quad \int_{-1}^{y_k} u(x_k, \eta)d\eta = \sum_{j=1}^{N_2} a_{sj}u(x_k, y_j)$$

for $s \in \{1, 2, 3, ..., N_2\}$. Let $\tilde{\mathbf{U}}_y(x_k, \cdot) = [U_y(x_k, y_1), U_y(x_k, y_2), U_y(x_k, y_3), ..., U_k(x_k, y_{N_2})]^T$ and $\tilde{\mathbf{u}}(x_k, \cdot) = [u(x_k, y_1), u(x_k, y_2), u_y(x_k, y_3), ..., u(x_k, y_{N_2})]^T$. Then, for $k \in \{1, 2, 3, ..., N_1\}$, $\tilde{\mathbf{U}}_y(x_k, \cdot) = \tilde{\mathbf{A}}_y \tilde{\mathbf{u}}(x_k, \cdot)$ and

$$\begin{bmatrix} \tilde{\mathbf{U}}_y(x_1, \cdot) \\ \tilde{\mathbf{U}}_y(x_2, \cdot) \\ \tilde{\mathbf{U}}_y(x_3, \cdot) \\ \vdots \\ \tilde{\mathbf{U}}_y(x_{N_1}, \cdot) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{A} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{A} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{A} \end{bmatrix}}_{N_1 \text{ blocks}} \begin{bmatrix} \tilde{\mathbf{u}}(x_1, \cdot) \\ \tilde{\mathbf{u}}(x_2, \cdot) \\ \tilde{\mathbf{u}}(x_3, \cdot) \\ \vdots \\ \tilde{\mathbf{u}}(x_{N_1}, \cdot) \end{bmatrix}, \tag{2.23}$$

where $\mathbf{A} = \bar{\mathbf{T}}\mathbf{T}^{-1}$ is an $N_2 \times N_2$ matrix and denote (2.23) by $\tilde{\mathbf{U}}_y = \tilde{\mathbf{A}}_y \tilde{\mathbf{u}}$.

The integration and integrand vector in the local numbering system can be transformed to the global numbering system by using the transformation matrix $\mathbf{P}$, i.e.,

$$\mathbf{U}_y = \mathbf{P}\tilde{\mathbf{U}}_y \text{ and } \mathbf{u} = \mathbf{P}\tilde{\mathbf{u}}. \tag{2.24}$$

The transformation matrix $\mathbf{P}$ is defined by

$$\mathbf{P}_{mn} = \begin{cases} 1 \;\; ; & \begin{cases} m = N_1 \times (j-1) + i, \\ n = N_2 \times (i-1) + j, \end{cases} \\ 0 \;\; ; & \text{otherwise,} \end{cases} \tag{2.25}$$

for $i \in \{1, 2, 3, ..., N_1\}$ and $j \in \{1, 2, 3, ..., N_2\}$. Therefore, we have the integration matrix with respect to y in the global numbering system as $\mathbf{U}_y = \mathbf{A}_y \mathbf{u}$ where $\mathbf{A}_y = \mathbf{P}\tilde{\mathbf{A}}_y\mathbf{P}^{-1}$ and $\mathbf{P}^{-1} = \mathbf{P}^T$, i.e.,

$$\mathbf{A}_y = \begin{bmatrix} a_{11}\mathbf{I}_{N_1} & a_{12}\mathbf{I}_{N_1} & a_{13}\mathbf{I}_{N_1} & \dots & a_{1N_2}\mathbf{I}_{N_1} \\ a_{21}\mathbf{I}_{N_1} & a_{22}\mathbf{I}_{N_1} & a_{23}\mathbf{I}_{N_1} & \dots & a_{2N_2}\mathbf{I}_{N_1} \\ a_{31}\mathbf{I}_{N_1} & a_{32}\mathbf{I}_{N_1} & a_{33}\mathbf{I}_{N_1} & \dots & a_{3N_2}\mathbf{I}_{N_1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{N_21}\mathbf{I}_{N_1} & a_{N_22}\mathbf{I}_{N_1} & a_{N_23}\mathbf{I}_{N_1} & \dots & a_{N_2N_2}\mathbf{I}_{N_1} \end{bmatrix}, \tag{2.26}$$

where $\mathbf{I}_{N_1}$ is the $N_1 \times N_1$ identity matrix, each a_{ij} is the element of the integration matrix $\mathbf{A} = \bar{\mathbf{T}}\mathbf{T}^{-1}$ with size $N_2 \times N_2$.

Next, let us consider double layer integrations along the x and y directions. In this thesis we express in the global numbering system as

$$\begin{aligned} U_x^{(2)}(x_k, y_s) &:= \int_{-1}^{x_k} \int_{-1}^{\xi_2} u(\xi_1, y_s) d\xi_1 d\xi_2 \\ &= \sum_{i=1}^{N_1} a_{ki} \int_{-1}^{x_i} u(\xi_1, y_s) d\xi_1 \\ &= \sum_{i=1}^{N_1} \sum_{j=1}^{N_1} a_{ki} a_{ij} u(x_i, y_s) \\ &= \sum_{i=1}^{N_1} a_{ki}^2 u(x_i, y_s) \end{aligned}$$

for $k \in \{1, 2, 3, ..., N_1\}$ when y_s is fixed. Let $\mathbf{U}_x^{(2)}(\cdot, y_s) = [U_x^{(2)}(x_1, y_s),\ U_x^{(2)}(x_2, y_s),\ U_x^{(2)}(x_3, y_s), ..., U_x^{(2)}(x_{N_1}, y_s)]^T$ and $\mathbf{u}(\cdot, y_s) = [u(x_1, y_s), u(x_2, y_s), u(x_3, y_s), ..., u(x_{N_1}, y_s)]^T$. Then, for $s \in \{1, 2, 3, ..., N_2\}$, $\mathbf{U}_x^{(2)}(\cdot, y_s) = \mathbf{A}^2 \mathbf{u}_x^{(2)}(\cdot, y_s)$ and

$$\begin{bmatrix} \mathbf{U}_x^{(2)}(\cdot, y_1) \\ \mathbf{U}_x^{(2)}(\cdot, y_2) \\ \mathbf{U}_x^{(2)}(\cdot, y_3) \\ \vdots \\ \mathbf{U}_x^{(2)}(\cdot, y_{N_2}) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}^2 & 0 & 0 & \dots & 0 \\ 0 & \mathbf{A}^2 & 0 & \dots & 0 \\ 0 & 0 & \mathbf{A}^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{A}^2 \end{bmatrix}}_{N_2 \text{ blocks}} \begin{bmatrix} \mathbf{u}(\cdot, y_1) \\ \mathbf{u}(\cdot, y_2) \\ \mathbf{u}(\cdot, y_3) \\ \vdots \\ \mathbf{u}(\cdot, y_{N_2}) \end{bmatrix}, \tag{2.27}$$

where $\mathbf{A} = \bar{\mathbf{T}}\mathbf{T}^{-1}$ is an $N_1 \times N_1$ matrix and denote (2.27) by $\mathbf{U}_x^{(2)} = \mathbf{A}_x^2 \mathbf{u}$.

Similarly, for the double layer integration with respect to y, we have

$$U_y^{(2)}(x_k, y_s) := \sum_{j=1}^{N_2} a_{sj}^2 u(x_k, y_j)$$

for $s \in \{1, 2, 3, ..., N_2\}$ when x_k is fixed. Let $\tilde{\mathbf{U}}_y^{(2)}(x_k, \cdot) = [U_y^{(2)}(x_k, y_1),\ U_y^{(2)}(x_k, y_2),$

$\mathrm{U}_y^{(2)}(x_k, y_3), ..., U_k^{(2)}(x_k, y_{N_2})]^T$ and $\tilde{\mathbf{u}}(\cdot, y_s) = [\tilde{u}(x_k, y_1), \tilde{u}(x_k, y_2), \tilde{u}(x_k, y_3), ..., \tilde{u}(x_k, y_{N_2})]^T$. Then, for $k \in \{1, 2, 3, ..., N_1\}$, $\tilde{\mathbf{U}}_y^{(2)}(x_k, \cdot) = \mathbf{A}^2 \tilde{\mathbf{u}}_y^{(2)}(x_k, \cdot)$ and

$$\begin{bmatrix} \tilde{\mathbf{U}}_y^{(2)}(x_1, \cdot) \\ \tilde{\mathbf{U}}_y^{(2)}(x_2, \cdot) \\ \tilde{\mathbf{U}}_y^{(2)}(x_3, \cdot) \\ \vdots \\ \tilde{\mathbf{U}}_y^{(2)}(x_{N_1}, \cdot) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}^2 & 0 & 0 & \dots & 0 \\ 0 & \mathbf{A}^2 & 0 & \dots & 0 \\ 0 & 0 & \mathbf{A}^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{A}^2 \end{bmatrix}}_{N_1 blocks} \begin{bmatrix} \tilde{\mathbf{u}}(x_1, \cdot) \\ \tilde{\mathbf{u}}(x_2, \cdot) \\ \tilde{\mathbf{u}}(x_3, \cdot) \\ \vdots \\ \tilde{\mathbf{u}}(x_{N_1}, \cdot) \end{bmatrix}, \tag{2.28}$$

where $\mathbf{A} = \bar{\mathbf{T}}\mathbf{T}^{-1}$ is an $N_2 \times N_2$ matrix and denote (2.28) by $\tilde{\mathbf{U}}_y^{(2)} = \tilde{\mathbf{A}}_y^2 \tilde{\mathbf{u}}$.

We can transform $\tilde{\mathbf{U}}_y^{(2)} = \tilde{\mathbf{A}}_y^2 \tilde{\mathbf{u}}$ into the global numbering system by using the transformation matrix $\mathbf{P}$, then we obtain $\mathbf{U}_y^{(2)} = \mathbf{A}_y^2 \mathbf{u}$, where $\mathbf{A}_y = \mathbf{P}\tilde{\mathbf{A}}_y \mathbf{P}^{-1}$.

For the double layer integration with respect to x and follow with y, which is $U_{xy}^{(2)}(x, y)$, we have

$$\begin{aligned} U_{xy}^{(2)}(x_k, y_s) &:= \int_{-1}^{y_s} \int_{-1}^{x_k} u(\xi, \eta) d\xi d\eta \\ &= \sum_{j=1}^{N_2} \sum_{j=1}^{N_2} a_{sj} a_{ki} u(x_i, y_j) \end{aligned} \tag{2.29}$$

for $k \in \{1, 2, 3, ..., N_1\}$ and $s \in \{1, 2, 3, ..., N_2\}$. We can consider (2.29) into two cases as follow.

<u>Case I</u>: If y_s is fixed, then (2.29) can be written in matrix form as

$$\mathbf{U}_{xy}^{(2)}(\cdot, y_s) = \sum_{j=1}^{N_2} a_{sj} \mathbf{A} \mathbf{u}(\cdot, y_j), \tag{2.30}$$

where $\mathbf{U}_{xy}^{(2)}(\cdot, y_s) = [U_{xy}^{(2)}(x_1, y_s), U_{xy}^{(2)}(x_2, y_s), U_{xy}^{(2)}(x_3, y_s), ..., U_{xy}^{(2)}(x_{N_1}, y_s)]^T$, $\mathbf{u}(\cdot, y_j) = [u(x_1, y_j), u(x_2, y_j), u(x_3, y_j), ..., u(x_{N_1}, y_j)]^T$ and $\mathbf{A} = \bar{\mathbf{T}}\mathbf{T}^{-1}$ is an $N_1 \times N_1$

matrix. That is, for $s \in \{1, 2, 3, ..., N_2\}$,

$$\begin{bmatrix} \mathbf{U}_{xy}^{(2)}(\cdot, y_1) \\ \mathbf{U}_{xy}^{(2)}(\cdot, y_2) \\ \mathbf{U}_{xy}^{(2)}(\cdot, y_3) \\ \vdots \\ \mathbf{U}_{xy}^{(2)}(\cdot, y_{N_2}) \end{bmatrix} = \begin{bmatrix} a_{s1}\mathbf{A} & 0 & 0 & \dots & 0 \\ 0 & a_{s2}\mathbf{A} & 0 & \dots & 0 \\ 0 & 0 & a_{s3}\mathbf{A} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{sN_2}\mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{u}(\cdot, y_1) \\ \mathbf{u}(\cdot, y_2) \\ \mathbf{u}(\cdot, y_3) \\ \vdots \\ \mathbf{u}(\cdot, y_{N_2}) \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}\mathbf{I}_{N_1} & a_{12}\mathbf{I}_{N_1} & a_{13}\mathbf{I}_{N_1} & \dots & a_{1N_1}\mathbf{I}_{N_1} \\ a_{21}\mathbf{I}_{N_1} & a_{22}\mathbf{I}_{N_1} & a_{23}\mathbf{I}_{N_1} & \dots & a_{2N_1}\mathbf{I}_{N_1} \\ a_{31}\mathbf{I}_{N_1} & a_{32}\mathbf{I}_{N_1} & a_{33}\mathbf{I}_{N_1} & \dots & a_{3N_1}\mathbf{I}_{N_1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{N_21}\mathbf{I}_{N_1} & a_{N_22}\mathbf{I}_{N_1} & a_{N_23}\mathbf{I}_{N_1} & \dots & a_{N_2N_1}\mathbf{I}_{N_1} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{A} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{A} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{A} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{A} \end{bmatrix}}_{N_2 \text{ blocks}} \begin{bmatrix} \mathbf{u}(\cdot, y_1) \\ \mathbf{u}(\cdot, y_2) \\ \mathbf{u}(\cdot, y_3) \\ \vdots \\ \mathbf{u}(\cdot, y_{N_2}) \end{bmatrix}$$

or $\mathbf{U}_{xy}^{(2)} = \mathbf{A}_y\mathbf{A}_x\mathbf{u}$, where $\mathbf{I}_{N_1}$ is the $N_1 \times N_1$ identity matrix, $\mathbf{A}_y$ and $\mathbf{A}_x$ defined by (2.26) and (2.22), respectively.

Case II: If x_k is fixed, then (2.29) can be written in matrix form as

$$\tilde{\mathbf{U}}_{xy}^{(2)}(x_k, \cdot) = \sum_{i=1}^{N_1} a_{ki}\mathbf{A}\tilde{\mathbf{u}}(x_i, \cdot), \tag{2.31}$$

where $\tilde{\mathbf{U}}_{xy}^{(2)}(x_k, \cdot) = [U_{xy}^{(2)}(x_k, y_1), U_{xy}^{(2)}(x_k, y_2), U_{xy}^{(2)}(x_k, y_3), ..., U_{xy}^{(2)}(x_k, y_{N_2})]^T$, $\tilde{\mathbf{u}}(x_k, \cdot) = [u(x_k, y_1), u(x_k, y_2), u(x_k, y_3), ..., u(x_k, y_{N_2})]^T$ and $\mathbf{A} = \bar{\mathbf{T}}\mathbf{T}^{-1}$ is an $N_2 \times N_2$ matrix. That is, for $k \in \{1, 2, 3, ..., N_1\}$,

$$\begin{bmatrix} \mathbf{U}_{xy}^{(2)}(x_1, \cdot) \\ \mathbf{U}_{xy}^{(2)}(x_2, \cdot) \\ \mathbf{U}_{xy}^{(2)}(x_3, \cdot) \\ \vdots \\ \mathbf{U}_{xy}^{(2)}(x_k, \cdot) \end{bmatrix} = \begin{bmatrix} a_{k1}\mathbf{A} & 0 & 0 & \dots & 0 \\ 0 & a_{k2}\mathbf{A} & 0 & \dots & 0 \\ 0 & 0 & a_{k3}\mathbf{A} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{kN_1}\mathbf{A} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}(\cdot, y_1) \\ \tilde{\mathbf{u}}(\cdot, y_2) \\ \tilde{\mathbf{u}}(\cdot, y_3) \\ \vdots \\ \tilde{\mathbf{u}}(\cdot, y_{N_1}) \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}\mathbf{I}_{N_2} & a_{12}\mathbf{I}_{N_2} & a_{13}\mathbf{I}_{N_2} & \dots & a_{1N_1}\mathbf{I}_{N_2} \\ a_{21}\mathbf{I}_{N_2} & a_{22}\mathbf{I}_{N_2} & a_{23}\mathbf{I}_{N_2} & \dots & a_{2N_1}\mathbf{I}_{N_2} \\ a_{31}\mathbf{I}_{N_2} & a_{32}\mathbf{I}_{N_2} & a_{33}\mathbf{I}_{N_2} & \dots & a_{3N_1}\mathbf{I}_{N_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{N_11}\mathbf{I}_{N_2} & a_{N_12}\mathbf{I}_{N_2} & a_{N_13}\mathbf{I}_{N_2} & \dots & a_{N_1N_1}\mathbf{I}_{N_2} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{A} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{A} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{A} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{A} \end{bmatrix}}_{N_1 \text{ blocks}} \begin{bmatrix} \tilde{\mathbf{u}}(\cdot, y_1) \\ \tilde{\mathbf{u}}(\cdot, y_2) \\ \tilde{\mathbf{u}}(\cdot, y_3) \\ \vdots \\ \tilde{\mathbf{u}}(\cdot, y_{N_1}) \end{bmatrix},$$

or $\tilde{\mathbf{U}}_{xy}^{(2)} = \tilde{\mathbf{A}}_y\tilde{\mathbf{A}}_x\tilde{\mathbf{u}}$, where $\mathbf{I}_{N_2}$ is the $N_2 \times N_2$ identity matrix. By using (2.24) and (2.26), we obtain

$$\begin{aligned} \mathbf{U}_{xy}^{(2)} &= \mathbf{P}\tilde{\mathbf{U}}_{xy}^{(2)} = \mathbf{P}(\tilde{\mathbf{A}}_x\tilde{\mathbf{A}}_y\tilde{\mathbf{u}}) \\ &= \mathbf{P}((\mathbf{P}^{(-1)}\mathbf{A}_x\mathbf{P})(\mathbf{P}^{(-1)}\mathbf{A}_y\mathbf{P}(\mathbf{P}^{(-1)}\mathbf{u})) \\ &= \mathbf{A}_x\mathbf{A}_y\mathbf{u}, \end{aligned}$$

where $\mathbf{A}_x$ and $\mathbf{A}_y$ are defined by (2.22) and (2.26), respectively.

Similar idea can be applied for the double layer integration with respect to y and follow with x, which is $U_{yx}^{(2)}(x, y)$. Thus, we have $\mathbf{U}_{yx}^{(2)} = \mathbf{U}_{xy}^{(2)} = \mathbf{A}_x\mathbf{A}_y\mathbf{u} = \mathbf{A}_y\mathbf{A}_x\mathbf{u}$.

Finally, for the m-layer integration of $u(x, y)$ with respect to x- or y-axis, we also can use the idea of constructing $\mathbf{U}_x^{(2)}$, $\mathbf{U}_y^{(2)}$, $\mathbf{U}_{xy}^{(2)}$ and $\mathbf{U}_{yx}^{(2)}$ to obtain $\mathbf{U}_x^{(m)} = \mathbf{A}_x^m\mathbf{u}$, $\mathbf{U}_y^{(m)} = \mathbf{A}_y^m\mathbf{u}$ and $\mathbf{U}_{xy}^{(m,n)} = \mathbf{U}_{yx}^{(n,m)} = \mathbf{A}_x^m\mathbf{A}_y^n\mathbf{u} = \mathbf{A}_y^n\mathbf{A}_x^m\mathbf{u}$.

CHAPTER III

NUMERICAL PROCEDURE FOR SOLVING TIME-DEPENDENT LINEAR PARTIAL DIFFERENTIAL EQUATIONS

In this chapter, we construct algorithms based on the FIM using Chebyshev polynomials for finding the approximate solutions of the time-dependent linear PDEs subject to Dirichlet boundary conditions in one- and two-dimensions.

3.1 Algorithm For Solving Time Dependent Problem In One-Dimension

Let $T > 0$, a, b be real numbers such that $a < b$. Consider the one-dimensional time-dependent linear PDE over $(a, b)\times(0, T)$ subject to initial condition and the Dirichlet boundary conditions

$$\frac{\partial u}{\partial t} = \alpha_1(x,t)\frac{\partial^2 u}{\partial x^2} + \alpha_2(x,t)\frac{\partial u}{\partial x} + \alpha_3(x,t)u + f(x,t) \equiv G(t,u), \tag{3.1}$$

$$u(x, 0) = F(x) \text{ for } x \in [a, b] \text{ and } u(a, t) = F_1(t), \quad u(b, t) = F_2(t) \text{ for } t \in [0, T],$$

where $\alpha_1(x, t)$ is are twice continuously differentiable with respect to x on (a, b) and continuous with respect to t on $(0, T)$, $\alpha_2(x, t)$ is continuously differentiable with respect to x on (a, b) and continuous with respect to t on $(0, T)$, $\alpha_3(x, t), f(x, t)$ are continuous over $(a, b) \times (0, T)$, $F(x)$ is continuous over $[a, b]$ and $F_1(t), F_2(t)$ are continuous over $[0, T]$. Moreover, throughout this section, let us assume that the solution of (3.1) exists and unique.

First, we approximate $\frac{\partial u}{\partial t}$ by using the forward difference, i.e., let

$$\frac{\partial u}{\partial t} = \frac{u^{j+1}(x) - u^j(x)}{\tau}, \tag{3.2}$$

where τ is a time step to be determined and $u^j = u^j(x) = u(x, t_j)$. Next, we approximate the function $G(t, u)$ by using the Crank-Nicolson method [11], i.e., let

$$G(t, u) = \frac{1}{2}\Big(G(t_j, u^j) + G(t_{j+1}, u^{j+1})\Big). \tag{3.3}$$

Thus, from (3.1) - (3.3), we have

$$u^{j+1}(x) = u^j(x) + \frac{\tau}{2}\Big(G(t_j, u^j) + G(t_{j+1}, u^{j+1})\Big). \tag{3.4}$$

For convenience, let $f^j = f(x, t_j)$ and $\alpha_i^j = \alpha_i(x, t_j)$ for $i \in \{1, 2, 3\}$. Then, from (3.4), we have

$$\begin{aligned} &-(\alpha_1^{j+1} u_{xx}^{j+1} + \alpha_2^{j+1} u_x^{j+1} + \alpha_3^{j+1} u^{j+1}) + \frac{2}{\tau} u^{j+1} \\ &= (\alpha_1^j u_{xx}^j + \alpha_2^j u_x^j + \alpha_3^j u^j) + \frac{2}{\tau} u^j + (f^j + f^{j+1}). \end{aligned} \tag{3.5}$$

Now, we are ready to apply the FIM using Chebyshev polynomials to devise an algorithm for calculating approximate solution of (3.1) as follows.

Step 1. Transform $x \in [a, b]$ into $\bar{x} \in [-1, 1]$ by using the transformation $\bar{x} = \frac{2x-a-b}{b-a}$. Then, (3.5) becomes

$$\begin{aligned} &-(p^2 \bar{\alpha}_1^{j+1} \bar{u}_{\bar{x}\bar{x}}^{j+1} + p \bar{\alpha}_2^{j+1} \bar{u}_{\bar{x}}^{j+1} + \bar{\alpha}_3^{j+1} \bar{u}^{j+1}) + \frac{2}{\tau} \bar{u}^{j+1} \\ &= (p^2 \bar{\alpha}_1^j \bar{u}_{\bar{x}\bar{x}}^j + p \bar{\alpha}_2^j \bar{u}_{\bar{x}}^j + \bar{\alpha}_3^j \bar{u}^j) + \frac{2}{\tau} \bar{u}^j + (\bar{f}^j + \bar{f}^{j+1}), \end{aligned} \tag{3.6}$$

where $p = \frac{2}{b-a}$, $\bar{f}^j = \bar{f}^j(\bar{x}, t_j) = f^j(\frac{(b-a)\bar{x}+a+b}{2}, t_j)$, $\bar{u}^j = \bar{u}^j(\bar{x}, t_j) = u^j(\frac{(b-a)\bar{x}+a+b}{2}, t_j)$ and $\bar{\alpha}_i^j = \bar{\alpha}^j(\bar{x}, t_j) = \alpha_i^j(\frac{(b-a)\bar{x}+a+b}{2}, t_j)$ for $i \in \{1, 2, 3\}$.

Step 2. Discretize $[-1,1]$ into M nodes by using the zeros of Chebyshev polynomial $T_M(\bar{x})$ as defined by (2.2), i.e.,

$$\bar{x}_k = \cos\Big[\Big(\frac{2k-1}{2M}\Big)\pi\Big],$$

where $k = \{1, 2, 3, ..., M\}$.

Step 3. Eliminate all derivatives with respect to $\bar{x}$ by taking double layers integration from -1 to ξ_2 and from -1 to $\bar{x}_k$, respectively. Then, (3.6) becomes

$$\begin{aligned}
&-\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}(p^2\bar{\alpha}_1^{j+1}\bar{u}_{\bar{x}\bar{x}}^{j+1} + p\bar{\alpha}_2^{j+1}\bar{u}_{\bar{x}}^{j+1})d\xi_1 d\xi_2 - \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}(\bar{\alpha}_3^{j+1}\bar{u}^{j+1} - \frac{2}{\tau}\bar{u}^{j+1})d\xi_1 d\xi_2 \\
&= \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}(p^2\bar{\alpha}_1^{j}\bar{u}_{\bar{x}\bar{x}}^{j} + p\bar{\alpha}_2^{j}\bar{u}_{\bar{x}}^{j} + \bar{\alpha}_3^{j}\bar{u}^{j})d\xi_1 d\xi_2 + \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}(\frac{2}{\tau}\bar{u}^{j} + \bar{f}^{j} + \bar{f}^{j+1})d\xi_1 d\xi_2.
\end{aligned} \tag{3.7}$$

Next, by using the integration by parts, (3.7) becomes

$$\begin{aligned}
&-\Big(p^2\bar{\alpha}_1^{j+1}\bar{u}^{j+1} - 2p^2\int_{-1}^{\bar{x}_k}\bar{\alpha}_{1,\bar{x}}^{j+1}\bar{u}^{j+1}d\xi_2 + \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}p^2\bar{\alpha}_{1,\bar{x}\bar{x}}^{j+1}\bar{u}^{j+1}d\xi_1 d\xi_2 \\
&+\int_{-1}^{\bar{x}_k}p\bar{\alpha}_2^{j+1}\bar{u}^{j+1}d\xi_2 - \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}p\bar{\alpha}_{2,\bar{x}}^{j+1}\bar{u}^{j+1}d\xi_1 d\xi_2 + \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\bar{\alpha}_3^{j+1}\bar{u}^{j+1}d\xi_1 d\xi_2\Big) \\
&+\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\frac{2}{\tau}\bar{u}^{j+1}d\xi_1 d\xi_2 + c_0 + c_1\bar{x}_k \\
&= \Big(p^2\bar{\alpha}_1^{j}\bar{u}^{j} - 2p^2\int_{-1}^{\bar{x}_k}\bar{\alpha}_{1,\bar{x}}^{j}\bar{u}^{j}d\xi_2 + \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}p^2\bar{\alpha}_{1,\bar{x}\bar{x}}^{j}\bar{u}^{j}d\xi_1 d\xi_2 \\
&+\int_{-1}^{\bar{x}_k}p\bar{\alpha}_2^{j}\bar{u}^{j}d\xi_2 - \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}p\bar{\alpha}_{2,\bar{x}}^{j}\bar{u}^{j}d\xi_1 d\xi_2 + \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\bar{\alpha}_3^{j}\bar{u}^{j}d\xi_1 d\xi_2\Big) \\
&+\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\frac{2}{\tau}\bar{u}^{j}d\xi_1 d\xi_2 + \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}(\bar{f}^{j} + \bar{f}^{j+1})d\xi_1 d\xi_2,
\end{aligned} \tag{3.8}$$

where c_0 and c_1 are arbitrary constants.

Step 4. By the first and the second order integration matrices described in Section 2.6, (3.8) can be written in matrix form as

$$
\begin{aligned}
&-(p^2\mathbf{B}_1^{j+1}\mathbf{u}^{j+1} - 2p^2\mathbf{A}\mathbf{B}_{1,\bar{x}}^{j+1}\mathbf{u}^{j+1} + p^2\mathbf{A}^2\mathbf{B}_{1,\bar{x}\bar{x}}^{j+1}\mathbf{u}^{j+1} \\
&+ p\mathbf{A}\mathbf{B}_2^{j+1}\mathbf{u}^{j+1} - p\mathbf{A}^2\mathbf{B}_{2,\bar{x}}^{j+1}\mathbf{u}^{j+1} + \mathbf{A}^2\mathbf{B}_3^{j+1}\mathbf{u}^{j+1}) \\
&+ \frac{2}{\tau}\mathbf{A}^2\mathbf{u}^{j+1} + c_0\mathbf{E} + c_1\bar{\mathbf{x}} \\
&= (p^2\mathbf{B}_1^{j}\mathbf{u}^{j} - 2p^2\mathbf{A}\mathbf{B}_{1,\bar{x}}^{j}\mathbf{u}^{j} + p^2\mathbf{A}^2\mathbf{B}_{1,\bar{x}\bar{x}}^{j}\mathbf{u}^{j} \\
&+ p\mathbf{A}\mathbf{B}_2^{j}\mathbf{u}^{j} - p\mathbf{A}^2\mathbf{B}_{2,\bar{x}}^{j}\mathbf{u}^{j} + \mathbf{A}^2\mathbf{B}_3^{j}\mathbf{u}^{j}) \\
&+ \frac{2}{\tau}\mathbf{A}^2\mathbf{u}^{j} + \mathbf{A}^2(\mathbf{f}^{\,j} + \mathbf{f}^{\,j+1}).
\end{aligned}
$$

That is,

$$
\left(\frac{2}{\tau}\mathbf{A}^2 - \mathbf{L}^{j+1}\right)\mathbf{u}^{j+1} + c_0\mathbf{E} + c_1\bar{\mathbf{x}} = \left(\frac{2}{\tau}\mathbf{A}^2 + \mathbf{L}^{j}\right)\mathbf{u}^{j} + \mathbf{A}^2(\mathbf{f}^{\,j} + \mathbf{f}^{\,j+1}), \tag{3.9}
$$

where, for $r \in \{j, j+1\}$,

$$
\begin{aligned}
\mathbf{u}^r &= [\bar{u}^r(\bar{x}_1), \bar{u}^r(\bar{x}_2), \bar{u}^r(\bar{x}_3), ..., \bar{u}^r(\bar{x}_M)]^T, \\
\mathbf{f}^{\,r} &= [\bar{f}^r(\bar{x}_1), \bar{f}^r(\bar{x}_2), \bar{f}^r(\bar{x}_3), ..., \bar{f}^r(\bar{x}_M)]^T, \\
\mathbf{L}^r &= p^2\mathbf{B}_1^r - 2p^2\mathbf{A}\mathbf{B}_{1,\bar{x}}^r + p^2\mathbf{A}^2\mathbf{B}_{1,\bar{x}\bar{x}}^r + p\mathbf{A}\mathbf{B}_2^r - p\mathbf{A}^2\mathbf{B}_{2,\bar{x}}^r + \mathbf{A}^2\mathbf{B}_3^r, \\
\mathbf{B}_i^r &= \operatorname{diag}(\bar{\alpha}_i^r(\bar{x}_1), \bar{\alpha}_i^r(\bar{x}_2), \bar{\alpha}_i^r(\bar{x}_3), ..., \bar{\alpha}_i^r(\bar{x}_M)) \text{ for } \ i \in \{1,2,3\}, \\
\mathbf{B}_{i,\bar{x}}^r &= \operatorname{diag}(\bar{\alpha}_{i,\bar{x}}^r(\bar{x}_1), \bar{\alpha}_{i,\bar{x}}^r(\bar{x}_2), \bar{\alpha}_{i,\bar{x}}^r(\bar{x}_3), ..., \bar{\alpha}_{i,\bar{x}}^r(\bar{x}_M)) \text{ for } \ i \in \{1,2\}, \\
\mathbf{B}_{1,\bar{x}\bar{x}}^r &= \operatorname{diag}(\bar{\alpha}_{1,\bar{x}\bar{x}}^r(\bar{x}_1), \bar{\alpha}_{1,\bar{x}\bar{x}}^r(\bar{x}_2), \bar{\alpha}_{1,\bar{x}\bar{x}}^r(\bar{x}_3), ..., \bar{\alpha}_{1,\bar{x}\bar{x}}^r(\bar{x}_M)), \\
\mathbf{E} &= [1,1,1,..,1]^T \text{ and } \bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, \bar{x}_3, ..., \bar{x}_M]^T.
\end{aligned}
$$

Step 5. Consider the boundary conditions which are $\bar{u}(-1,t) = F_1(t)$ and $\bar{u}(1,t) = F_2(t)$. Then, we have

$$
F_1(t_{j+1}) = \sum_{n=0}^{M-1} c_n T_n(-1) = \mathbf{t}_l\mathbf{c} = \mathbf{t}_l\mathbf{T}^{-1}\mathbf{u}^{j+1}, \tag{3.10}
$$

$$
F_2(t_{j+1}) = \sum_{n=0}^{M-1} c_n T_n(1) = \mathbf{t}_r\mathbf{c} = \mathbf{t}_r\mathbf{T}^{-1}\mathbf{u}^{j+1}, \tag{3.11}
$$

where $\mathbf{t}_l = [1, -1, 1, ..., (-1)^{M-1}]$ and $\mathbf{t}_r = [1, 1, 1, ..., 1]$.

Step 6. Construct the linear system from (3.9), (3.10) and (3.11) which is

$$\begin{bmatrix} \frac{2}{\tau}\mathbf{A}^2 - \mathbf{L}^{j+1} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l\mathbf{T}^{-1} & 0 & 0 \\ \mathbf{t}_r\mathbf{T}^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{j+1} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\frac{2}{\tau}\mathbf{A}^2 + \mathbf{L}^j)\mathbf{u}^j + \mathbf{A}^2(\mathbf{f}^{\,j} + \mathbf{f}^{\,j+1}) \\ F_1(t_{j+1}) \\ F_2(t_{j+1}) \end{bmatrix}. \tag{3.12}$$

Finally, we can find the approximate solution by solving this linear system (3.12) directly. To obtain the approximate solution $u(x, t_j)$ for $x \in [a, b]$, we use transformation $x = \frac{1}{2}[(b-a)\bar{x} + a + b]$.

3.2 Numerical Examples For One-Dimensional Time-Dependent Linear PDEs

In this section, we use our proposed method to find the approximate solutions of some time-dependent one-dimensional linear PDEs. For an error of our approximate solution, we use the average relative error (ARE) defined by

$$\text{ARE} = \frac{1}{M}\sum_{k=1}^{M} \Big| \frac{u^*(x_k, t) - u(x_k, t)}{u^*(x_k, t)} \Big|,$$

where u^* and u are the exact and numerical solutions, respectively, and x_k for $k \in \{1, 2, 3, ..., M\}$ are the grid point defined by each zero of Chebyshev polynomial $T_M(x)$.

Example 3.1. Consider a time-dependent linear PDE in which the coefficients do not depend on time.

$$\frac{\partial u}{\partial t} = x^2\frac{\partial^2 u}{\partial x^2} + 2x\frac{\partial u}{\partial x} + u - (x^2 + 2x)e^{x+t}, \quad x \in (0, 1), \tag{3.13}$$

$$u(x, 0) = e^x, \quad x \in [0, 1],$$

$$u(0, t) = e^t, \quad u(1, t) = e^{t+1}, \quad t \geq 0.$$

The exact solution is $u(x,t) = e^{x+t}$. By using our numerical algorithm, this problem can be written in matrix form as

$$\Big(\frac{2}{\tau}\mathbf{A}^2 - \mathbf{L}^{j+1}\Big)\mathbf{u}^{j+1} + c_0\mathbf{E} + c_1\bar{\mathbf{x}} = \Big(\frac{2}{\tau}\mathbf{A}^2 + \mathbf{L}^{j}\Big)\mathbf{u}^{j} + \mathbf{A}^2(\mathbf{f}^{\,j} + \mathbf{f}^{\,j+1}),$$

where, for $r \in \{j, j+1\}$,

$$\begin{aligned}
\mathbf{u}^r &= [\bar{u}^r(\bar{x}_1), \bar{u}^r(\bar{x}_2), \bar{u}^r(\bar{x}_3), ..., \bar{u}^r(\bar{x}_M)]^T,\\
\mathbf{f}^r &= \Big[-\Big(\frac{(\bar{x}_1+1)^2}{4} + \bar{x}_1 + 1\Big)e^{\frac{(\bar{x}_1+1)}{2}+t_r}, -\Big(\frac{(\bar{x}_2+1)^2}{4} + \bar{x}_2 + 1\Big)e^{\frac{(\bar{x}_2+1)}{2}+t_r},\\
&\quad ..., -\Big(\frac{(\bar{x}_M+1)^2}{4} + \bar{x}_M + 1\Big)e^{\frac{(\bar{x}_M+1)}{2}+t_r}\Big]^T,\\
\mathbf{L}^r &= (4\mathbf{B}_1^r - 8\mathbf{A}\mathbf{B}_{1,\bar{x}}^r + 4\mathbf{A}^2\mathbf{B}_{1,\bar{x}\bar{x}}^r) + (2\mathbf{A}\mathbf{B}_2^r - 2\mathbf{A}^2\mathbf{B}_{2,\bar{x}}^r) + \mathbf{A}^2\mathbf{B}_3^r,\\
\mathbf{B}_1^r &= \mathrm{diag}((\bar{x}_1+1)^2/4, (\bar{x}_2+1)^2/4, (\bar{x}_3+1)^2/4, ..., (\bar{x}_M+1)^2/4),\\
\mathbf{B}_{1,\bar{x}}^r &= \mathrm{diag}((\bar{x}_1+1)/2, (\bar{x}_2+1)/2, (\bar{x}_3+1)/2, ..., (\bar{x}_M+1)/2),\\
\mathbf{B}_{1,\bar{x}\bar{x}}^r &= \mathrm{diag}(1/2, 1/2, 1/2, ..., 1/2)_{M\times M},\\
\mathbf{B}_2^r &= \mathrm{diag}(\bar{x}_1+1, \bar{x}_2+1, \bar{x}_3+1, ..., \bar{x}_M+1),\\
\mathbf{B}_{2,\bar{x}}^r &= \mathrm{diag}(1,1,1,...,1)_{M\times M},\\
\mathbf{B}_3^r &= \mathrm{diag}(1,1,1,...,1)_{M\times M},\\
\mathbf{E} &= [1,1,1,...,1]^T,\ \ \bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, \bar{x}_3, ..., \bar{x}_M],\\
\text{and } \mathbf{A} &= \bar{\mathbf{T}}\mathbf{T}^{-1} \text{ as defined in Section 2.6.}
\end{aligned}$$

For the boundary conditions, we have $\mathbf{t}_l\mathbf{T}^{-1}\mathbf{u}^{j+1} = e^{t_{j+1}}$ and $\mathbf{t}_r\mathbf{T}^{-1}\mathbf{u}^{j+1} = e^{t_{j+1}+1}$. Therefore, we solve the following linear system

$$\begin{bmatrix} \frac{2}{\tau}\mathbf{A}^2 - \mathbf{L}^{j+1} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l\mathbf{T}^{-1} & 0 & 0 \\ \mathbf{t}_r\mathbf{T}^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{j+1} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} \Big(\frac{2}{\tau}\mathbf{A}^2 + \mathbf{L}^j\Big)\mathbf{u}^j + \mathbf{A}^2(\mathbf{f}^j + \mathbf{f}^{j+1}) \\ e^{t_{j+1}} \\ e^{t_{j+1}+1} \end{bmatrix}. \tag{3.14}$$

Finally, we obtain the approximate solutions $u(x,t_{j+1}) = \bar{u}(\bar{x},t_{j+1})$, where $x = \frac{1}{2}(\bar{x}+1)$ is the transformation from $\bar{x} \in [-1,1]$ into $x \in [0,1]$. Table 3.1 shows the relative errors of our approximate solutions when $M = 6$ at each point in the domain $(0, 1)$ at $t = 1$ and $\tau = 0.1$. Then, the AREs for our FIM using Chebyshev polynomials (CBS) compared with traditional FIMs using trapezoidal (TPZ) and Simpson's rules (SIM) are shown in Tables 3.2 and 3.3 at $t = 1$ with time steps $\tau = 0.1$ and 0.001, respectively. Also, the graphs of exact and approximate solutions are shown in Figure 3.1.

Table 3.1: The relative errors of our approximate solutions when $M = 6$ at each point in $(0, 1)$ at $t = 1$ and $\tau = 0.1$

x_k	$u^*(x_k, 1)$	$u(x_k, 1)$	Relative Error
0.1091	3.0316	3.0339	7.5783×10^{-4}
0.2831	3.6077	3.6103	7.2402×10^{-4}
0.5000	4.4817	4.4833	3.6704×10^{-4}
0.7169	5.5675	5.5684	1.6499×10^{-4}
0.8909	6.6254	6.6258	5.1095×10^{-5}
0.9875	7.2970	7.2970	4.1441×10^{-6}

Table 3.2: The AREs for Example 3.1 when $\tau = 0.1$

M	CBS	TPZ	SIM
6	3.4485×10^{-4}	2.7573×10^{-2}	1.2040×10^{-2}
8	3.0501×10^{-4}	1.5109×10^{-2}	5.2615×10^{-3}
10	3.2558×10^{-4}	9.3835×10^{-3}	2.6388×10^{-3}
12	3.3920×10^{-4}	6.3294×10^{-3}	1.4277×10^{-3}

Table 3.3: The AREs for Example 3.1 when $\tau = 0.001$

M	CBS	TPZ	SIM
6	7.7532×10^{-5}	2.7846×10^{-2}	1.2297×10^{-2}
8	3.5817×10^{-7}	1.5400×10^{-2}	5.5570×10^{-3}
10	3.3219×10^{-8}	9.7105×10^{-3}	2.9604×10^{-3}
12	3.3936×10^{-8}	6.6686×10^{-3}	1.7619×10^{-3}

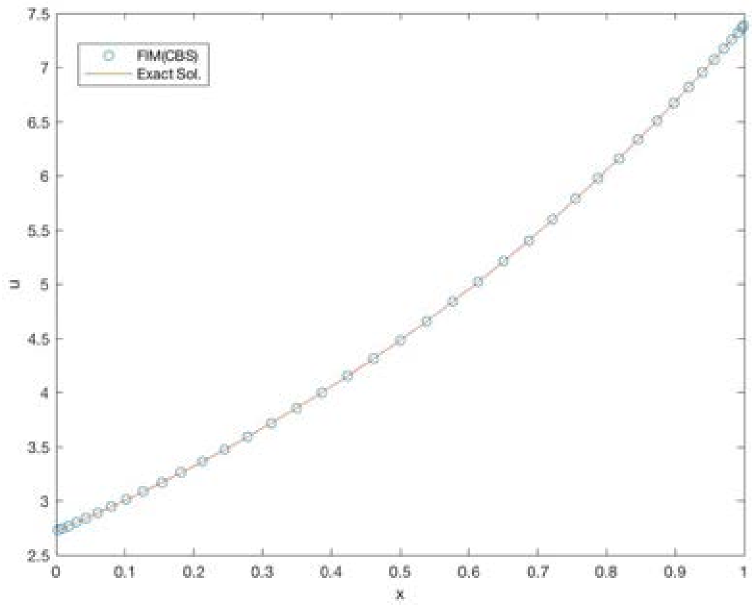


Figure 3.1: The graph of exact and approximate solutions of Example 3.1 at $t = 1$.

Example 3.2. Consider a time-dependent differential equation in which coefficients are functions in terms of both x and t.

$$\begin{aligned}
&\frac{\partial u}{\partial t} = x^2 e^t \frac{\partial^2 u}{\partial x^2} + x e^t \frac{\partial u}{\partial x} + u - 4x^2 e^{2t}, \quad x \in (0, 2), \\
&u(x, 0) = x^2 + 1, \quad x \in [0, 2], \\
&u(0, t) = e^t, \quad u(2, t) = 5e^t, \quad t \geq 0.
\end{aligned}$$

The exact solution is $u(x, t) = e^t(x^2 + 1)$. First, we transform $x \in [0, 2]$ into $\bar{x} \in [-1, 1]$ by using $\bar{x} = x - 1$ and get $p = 1$, we can construct the linear system with boundary conditions $\bar{u}(-1, t_{j+1}) = e^{t_{j+1}}$ and $\bar{u}(1, t_{j+1}) = 5e^{t_{j+1}}$ as

$$\begin{bmatrix} \frac{2}{\tau}\mathbf{A}^2 - \mathbf{L}^{j+1} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l\mathbf{T}^{-1} & 0 & 0 \\ \mathbf{t}_r\mathbf{T}^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{j+1} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\frac{2}{\tau}\mathbf{A}^2 + \mathbf{L}^j)\mathbf{u}^j + \mathbf{A}^2(\mathbf{f}^{\,j} + \mathbf{f}^{\,j+1}) \\ e^{t_{j+1}} \\ 5e^{t_{j+1}} \end{bmatrix},$$

where, for $r \in \{j, j+1\}$,

$$\begin{aligned}
\mathbf{u}^r &= [\bar{u}^r(\bar{x}_1), \bar{u}^r(\bar{x}_2), \bar{u}^r(\bar{x}_3), ..., \bar{u}^r(\bar{x}_M)]^T, \\
\mathbf{f}^{\,r} &= [-4(\bar{x}_1+1)^2 e^{2t_r}, -4(\bar{x}_2+1)^2 e^{2t_r}, -4(\bar{x}_3+1)^2 e^{2t_r}, ..., -4(\bar{x}_M+1)^2 e^{2t_r}]^T, \\
\mathbf{L}^r &= (\mathbf{B}_1^r - 2\mathbf{A}\mathbf{B}_{1,\bar{x}}^r + \mathbf{A}^2\mathbf{B}_{1,\bar{x}\bar{x}}^r) + (\mathbf{A}\mathbf{B}_2^r - \mathbf{A}^2\mathbf{B}_{2,\bar{x}}^r) + \mathbf{A}^2\mathbf{B}_3^r, \\
\mathbf{B}_1^r &= \mathrm{diag}((\bar{x}_1+1)^2 e^{t_r}, (\bar{x}_2+1)^2 e^{t_r}, (\bar{x}_3+1)^2 e^{t_r}, ..., (\bar{x}_M+1)^2 e^{t_r}), \\
\mathbf{B}_{1,\bar{x}}^r &= \mathrm{diag}(2(\bar{x}_1+1)e^{t_r}, 2(\bar{x}_2+1)e^{t_r}, 2(\bar{x}_3+1)e^{t_r}, ..., 2(\bar{x}_M+1)e^{t_r}), \\
\mathbf{B}_{1,\bar{x}\bar{x}}^r &= \mathrm{diag}(2e^{t_r}, 2e^{t_r}, 2e^{t_r}, ..., 2e^{t_r})_{M\times M}, \\
\mathbf{B}_2^r &= \mathrm{diag}((\bar{x}_1+1)e^{t_r}, (\bar{x}_2+1)e^{t_r}, (\bar{x}_3+1)e^{t_r} ..., (\bar{x}_M+1)e^{t_r}), \\
\mathbf{B}_{2,\bar{x}}^r &= \mathrm{diag}(e^{t_r}, e^{t_r}, e^{t_r}, ..., e^{t_r})_{M\times M}, \\
\mathbf{B}_3^r &= \mathrm{diag}(1, 1, 1, ..., 1)_{M\times M}, \\
\mathbf{E} &= [1, 1, 1, ..., 1]^T, \;\; \bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, \bar{x}_3, ..., \bar{x}_M]^T, \\
\text{and } \mathbf{A} &= \bar{\mathbf{T}}\mathbf{T}^{-1} \text{ as defined in Section 2.6.}
\end{aligned}$$

When this linear system is solved, we finally obtain the approximate solutions $u(x, t_{j+1}) = \bar{u}(\bar{x}, t_{j+1})$, where $x = \bar{x} + 1$ is the transformation from $\bar{x} \in [-1, 1]$ into $x \in [0, 2]$. Then, the AREs of the numerical solutions in this problem are computed from three methods including our FIM with CBS, traditional FIMs with TPZ and SIM when $M = 6, 8, 10$ and 12. Then, their AREs are shown in Tables 3.4 and 3.5 at $t = 1$ with time steps $\tau = 0.1$ and 0.001, respectively. Also, the Figure 3.2 displays the graph of solutions at each node in $(0, 2)$.

Table 3.4**:** The AREs for Example 3.2 when $\tau = 0.1$

M	CBS	TPZ	SIM
6	1.9268×10^{-4}	1.4119×10^{-1}	8.5174×10^{-2}
8	2.3260×10^{-4}	8.5196×10^{-2}	4.3706×10^{-2}
10	2.5809×10^{-4}	5.6183×10^{-2}	2.5265×10^{-2}
12	2.7601×10^{-4}	3.9605×10^{-2}	1.5893×10^{-2}

Table 3.5**:** The AREs for Example 3.2 when $\tau = 0.001$

M	CBS	TPZ	SIM
6	1.9298×10^{-8}	1.4138×10^{-1}	8.5316×10^{-2}
8	2.3305×10^{-8}	8.5392×10^{-2}	4.3899×10^{-2}
10	2.5865×10^{-8}	5.6397×10^{-2}	2.5491×10^{-2}
12	2.7656×10^{-8}	3.9838×10^{-2}	1.6144×10^{-2}

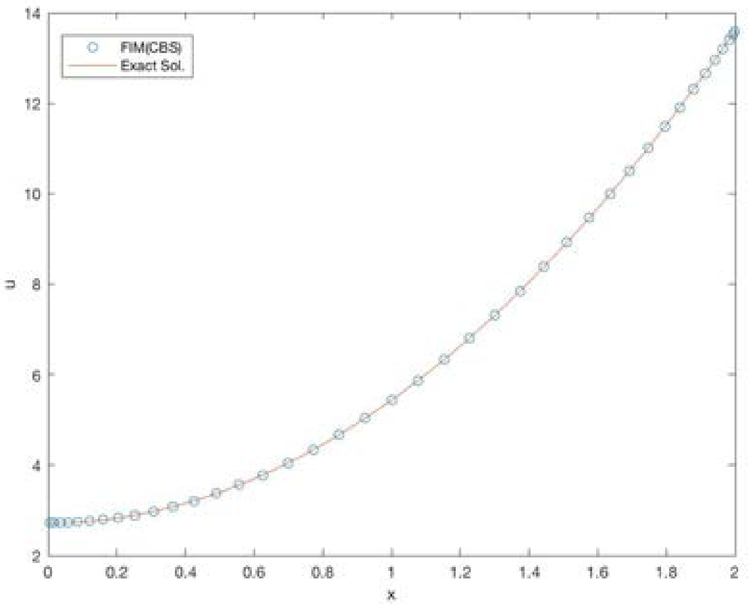


Figure 3.2**:** The graph of exact and approximate solutions of Example 3.2 at $t = 1$.

Example 3.3**.** Consider a time-dependent linear PDE in which coefficients are functions in terms of both x and t and the forcing term involves trigonometric functions.

$$\frac{\partial u}{\partial t} = x^2 \frac{\partial^2 u}{\partial x^2} + t\frac{\partial u}{\partial x} + x^2 u + 2t\sin(x+1) - t^3 cos(x+1), \quad x \in (0,1),$$

$$u(x,0) = 0, \quad x \in [0,1],$$

$$u(0,t) = t^2\sin(1), \quad u(1,t) = t^2\sin(2), \quad t \geq 0.$$

The exact solution is $u(x,t) = t^2\sin(x+1)$. First, we transform $x \in [0,1]$ into $\bar{x} \in [-1,1]$ by using $\bar{x} = 2x - 1$ and get $p = 2$. Now, we can construct the linear system with boundary conditions $\bar{u}(-1,t_{j+1}) = t_{j+1}^2 \sin(1)$ and $\bar{u}(1,t_{j+1}) = t_{j+1}^2 \sin(2)$ as

$$\begin{bmatrix} \frac{2}{\tau}\mathbf{A}^2 - \mathbf{L}^{j+1} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l\mathbf{T}^{-1} & 0 & 0 \\ \mathbf{t}_r\mathbf{T}^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{j+1} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\frac{2}{\tau}\mathbf{A}^2 + \mathbf{L}^j)\mathbf{u}^j + \mathbf{A}^2(\mathbf{f}^{\,j} + \mathbf{f}^{\,j+1}) \\ t_{j+1}^2 \sin(1) \\ t_{j+1}^2 \sin(2) \end{bmatrix},$$

where, for $r \in \{j, j+1\}$,

$$\begin{aligned}
\mathbf{u}^r &= [\bar{u}^r(\bar{x}_1), \bar{u}^r(\bar{x}_2), \bar{u}^r(\bar{x}_3), ..., \bar{u}^r(\bar{x}_M)]^T, \\
\mathbf{f}^{\,r} &= \Big[2t_r \sin\Big(\frac{\bar{x}_1+3}{2}\Big) - t_r^3 \cos\Big(\frac{\bar{x}_1+3}{2}\Big), 2t_r \sin\Big(\frac{\bar{x}_2+3}{2}\Big) - t_r^3 \cos\Big(\frac{\bar{x}_2+3}{2}\Big), \\
&\quad 2t_r \sin\Big(\frac{\bar{x}_3+3}{2}\Big) - t_r^3 \cos\Big(\frac{\bar{x}_3+3}{2}\Big), ..., 2t_r \sin\Big(\frac{\bar{x}_M+3}{2}\Big) - t_r^3 \cos\Big(\frac{\bar{x}_M+3}{2}\Big)\Big]^T, \\
\mathbf{L}^r &= (4\mathbf{B}_1^r - 8\mathbf{A}\mathbf{B}_{1,\bar{x}}^r + 4\mathbf{A}^2\mathbf{B}_{1,\bar{x}\bar{x}}^r) + \mathbf{A}^2\mathbf{B}_3^r, \\
\mathbf{B}_1^r &= \text{diag}\Big(\frac{(\bar{x}_1+1)^2}{4}, \frac{(\bar{x}_2+1)^2}{4}, \frac{(\bar{x}_3+1)^2}{4}, ..., \frac{(\bar{x}_M+1)^2}{4}\Big), \\
\mathbf{B}_{1,\bar{x}}^r &= \text{diag}\Big(\frac{\bar{x}_1+1}{2}, \frac{\bar{x}_2+1}{2}, \frac{\bar{x}_3+1}{2}, ..., \frac{\bar{x}_M+1}{2}\Big), \\
\mathbf{B}_{1,\bar{x}\bar{x}}^r &= \text{diag}\Big(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}\Big)_{M\times M}, \\
\mathbf{B}_2^r &= \text{diag}(t_r, t_t, t_t, ..., t_r), \\
\mathbf{B}_3^r &= \text{diag}\Big(\frac{(\bar{x}_1+1)^2}{4}, \frac{(\bar{x}_2+1)^2}{4}, \frac{(\bar{x}_3+1)^2}{4}, ..., \frac{(\bar{x}_M+1)^2}{4}\Big), \\
\mathbf{E} &= [1, 1, 1, ..., 1]^T, \ \ \bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, \bar{x}_3, ..., \bar{x}_M]^T,
\end{aligned}$$

and $\mathbf{A} = \bar{\mathbf{T}}\mathbf{T}^{-1}$ as defined in Section 2.6.

Finally, we acquire the approximate solutions $u(x, t_{j+1}) = \bar{u}(\bar{x}, t_{j+1})$ by calculating above linear system, where $x = \frac{1}{2}(\bar{x}+1)$ is the transformation from $\bar{x} \in [-1, 1]$ into $x \in [0, 1]$. The AREs of the numerical solutions in this problem are computed from three methods including our FIM with CBS, traditional FIMs with TPZ and SIM when $M = 6, 8, 10$ and 12. Then, their AREs are shown in Tables 3.6 and 3.7 at $t = 1$ with time steps $\tau = 0.1$ and 0.001, respectively. The graphs of exact and approximate solutions are shown in Figure 3.3.

Table 3.6: The AREs for Example 3.3 when $\tau = 0.1$

M	CBS	TPZ	SIM
6	4.3511×10^{-6}	7.4314×10^{-4}	5.0759×10^{-4}
8	6.8971×10^{-9}	4.0726×10^{-4}	1.7135×10^{-4}
10	5.3357×10^{-12}	2.5575×10^{-4}	7.7793×10^{-5}
12	1.0770×10^{-14}	1.7519×10^{-4}	4.1905×10^{-5}

Table 3.7: The AREs for Example 3.3 when $\tau = 0.001$

M	CBS	TPZ	SIM
6	4.3581×10^{-6}	7.4316×10^{-4}	5.0764×10^{-4}
8	6.9037×10^{-9}	4.0727×10^{-4}	1.7137×10^{-4}
10	5.5031×10^{-12}	2.5576×10^{-4}	7.7791×10^{-5}
12	4.1357×10^{-13}	1.7519×10^{-4}	4.1900×10^{-5}

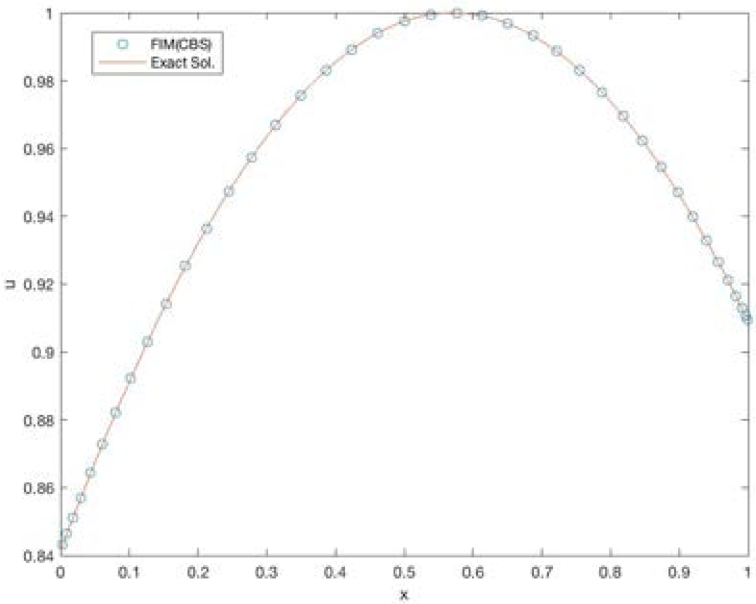


Figure 3.3: The graph of exact and approximate solutions of Example 3.3 at $t = 1$.

3.3 Algorithm For Solving Time Dependent Problem In Two-Dimensions

Let $T > 0$, a, b, c, d be real numbers such that $a < b$ and $c < d$. We consider linear PDE over $((a, b) \times (c, d)) \times (0, T)$ with time dependent variables in two dimensions as follow.

$$\begin{aligned}\frac{\partial u}{\partial t} &= \alpha_1(x,y,t)\frac{\partial^2 u}{\partial x^2}+\alpha_2(x,y,t)\frac{\partial^2 u}{\partial y^2}+\alpha_3(x,y,t)\frac{\partial^2 u}{\partial x\partial y}+\alpha_4(x,y,t)\frac{\partial u}{\partial x}+\alpha_5(x,y,t)\frac{\partial u}{\partial y}\\ &\quad +\alpha_6(x,y,t)u+f(x,y,t)\equiv G(t,u), \qquad (3.15)\end{aligned}$$

with initial condition $u(x,y,0)=F(x,y)$ and the Dirichlet boundary conditions

$$\begin{aligned}u(a,y,t) &= F_1(y,t) \;\; u(b,y,t)=F_2(y,t), \quad y\in(c,d),\\ u(x,c,t) &= H_1(x,t) \;\; u(x,d,t)=H_2(x,t), \quad x\in(a,b) \text{ for } \; t\in[0,T],\end{aligned}$$

where $\alpha_1(x,y,t),\alpha_2(x,y,t),\alpha_3(x,y,t)$ are are twice continuously differentiable with respect to x and y on $(a,b)\times(c,d)$ and continuous with respect to t on $(0,T)$, $\alpha_4(x,y,t),\alpha_5$ (x,y,t) are continuously differentiable with respect to x and y on $(a,b)\times(c,d)$ and continuous with respect to t on $(0,T)$, $\alpha_6(x,y,t),f(x,y,t)$ are continuous over $((a,b)\times(c,d))\times(0,T)$, $F(x,y)$ is continuous over $[a,b]\times[c,d]$, $F_1(y,t),F_2(y,t)$ are continuous over $[c,d]\times[0,T]$ and $H_1(x,t),H_2(x,t)$ are continuous over $[a,b]\times[0,T]$. Throughout this section, let us assume that the solution of (3.15) exists and unique.

First, we approximate $\frac{\partial u}{\partial t}$ by using the forward difference;

$$\frac{\partial u}{\partial t}=\frac{u^{j+1}(x,y)-u^j(x,y)}{\tau}, \qquad (3.16)$$

where τ is a time step to be determined and $u^j=u^j(x,y)=u(x,y,t_j)$. Then, we approximate the function $G(t,u)$ by using the Crank-Nicolson method [11], i.e.,

$$G(t,u)=\frac{1}{2}\Big(G(t_j,u^j)+G(t_{j+1},u^{j+1})\Big). \qquad (3.17)$$

Thus, from (3.15) - (3.17), we have

$$u^{j+1}(x,y)=u^j(x,y)+\frac{\tau}{2}\Big(G(t_j,u^j)+G(t_{j+1},u^{j+1})\Big). \qquad (3.18)$$

For convenience, Let $f^j = f(x, y, t_j)$ and $\alpha_i^j = \alpha_i(x, y, t_j)$ for $i \in \{1, 2, 3, ..., 6\}$, then from (3.18), we have

$$\begin{aligned}
&-(\alpha_1^{j+1} u_{xx}^{j+1} + \alpha_2^{j+1} u_{yy}^{j+1} + \alpha_3^{j+1} u_{xy}^{j+1} + \alpha_4^{j+1} u_x^{j+1} + \alpha_5^{j+1} u_y^{j+1} + \alpha_6^{j+1} u^{j+1}) + \frac{2}{\tau} u^{j+1} \\
&= (\alpha_1^j u_{xx}^j + \alpha_2^j u_{yy}^j + \alpha_3^j u_{xy}^j + \alpha_4^j u_x^j + \alpha_5^j u_y^j + \alpha_6^j u^j) + \frac{2}{\tau} u^j + (f^j + f^{j+1}). \qquad (3.19)
\end{aligned}$$

Now, we are ready to apply the FIM using Chebyshev polynomials to devise an algorithm for calculating the approximate solution of (3.15).

Step 1: Transform $\Omega = [a, b] \times [c, d]$ into $\bar{\Omega} = [-1, 1] \times [-1, 1]$ by the transformations

$$\bar{x} = \frac{2x - a - b}{b - a}, \; \bar{y} = \frac{2y - c - d}{d - c}.$$

Then, (3.19) becomes

$$\begin{aligned}
&- (p^2 \bar{\alpha}_1^{j+1} \bar{u}_{\bar{x}\bar{x}}^{j+1} + q^2 \bar{\alpha}_2^{j+1} \bar{u}_{\bar{y}\bar{y}}^{j+1} + pq \bar{\alpha}_3^{j+1} \bar{u}_{\bar{x}\bar{y}}^{j+1} + p \bar{\alpha}_4{}^{j+1} \bar{u}_{\bar{x}}^{j+1} + q \bar{\alpha}_5^{j+1} \bar{u}_{\bar{y}}^{j+1} + \bar{\alpha}_6^{j+1} \bar{u}^{j+1}) \\
&+ \frac{2}{\tau} \bar{u}^{j+1} = (p^2 \bar{\alpha}_1^j \bar{u}_{\bar{x}\bar{x}}^j + q^2 \bar{\alpha}_2^j \bar{u}_{\bar{y}\bar{y}}^j + pq \bar{\alpha}_3^j \bar{u}_{\bar{x}\bar{y}}^j + p \bar{\alpha}_4{}^j \bar{u}_{\bar{x}}^j + q \bar{\alpha}_5^j \bar{u}_{\bar{y}}^j + \bar{\alpha}_6^j \bar{u}^j) \\
&+ \frac{2}{\tau} \bar{u}^j + (\bar{f}^{j+1} + \bar{f}^j), \qquad (3.20)
\end{aligned}$$

where $p = \frac{2}{b-a}$, $q = \frac{2}{d-c}$, $\bar{f}^j = \bar{f}^j(\bar{x}, \bar{y}, t_j) = f^j(\frac{(b-a)\bar{x}+a+b}{2}, \frac{(d-c)\bar{x}+d+c}{2}, t_j)$, $\bar{u}^j = \bar{u}^j(\bar{x}, \bar{y}, t_j) = u^j(\frac{(b-a)\bar{x}+a+b}{2}, \frac{(d-c)\bar{x}+d+c}{2}, t_j)$ and $\bar{\alpha}_k^j = \bar{\alpha}^j(\bar{x}, \bar{y}, t_j) = \alpha^j(\frac{(b-a)\bar{x}+a+b}{2}, \frac{(d-c)\bar{x}+d+c}{2}, t_j)$ for $k \in \{1, 2, 3, 4, 5, 6\}$.

Step 2: We discretize the domain $\bar{\Omega} = [-1, 1] \times [-1, 1]$ into N_1 and N_2 nodes along x and y directions, which are generated by zeros of Chebyshev polynomials T_{N_1} and T_{N_2} as follows:

$$\bar{x}_k = \cos\left(\frac{(2k-1)\pi}{2N_1}\right), \text{ for } k \in \{1, 2, 3, ..., N_1\}, \qquad (3.21)$$

$$\bar{y}_s = \cos\left(\frac{(2s-1)\pi}{2N_2}\right), \text{ for } s \in \{1, 2, 3, ..., N_2\}, \qquad (3.22)$$

where $-1 < \bar{x}_1 < \bar{x}_2 < \bar{x}_3 < ... < \bar{x}_{N_1} < 1$ and $-1 < \bar{y}_1 < \bar{y}_2 < \bar{y}_3 < ... < \bar{y}_{N_2} < 1$. then, the number of total grid points in the global system is $M = N_1 \times N_2$.

Step 3: Eliminate the derivatives by taking four-layer integrations.

$$\begin{aligned}
\text{Let} \quad Z^{j+1} &:= -(p^2\bar{\alpha}_1^{j+1}\bar{u}_{\bar{x}\bar{x}}^{j+1} + q^2\bar{\alpha}_2^{j+1}\bar{u}_{\bar{y}\bar{y}}^{j+1} + pq\bar{\alpha}_3^{j+1}\bar{u}_{\bar{x}\bar{y}}^{j+1} \\
&\quad + p\bar{\alpha}_4{}^{j+1}\bar{u}_{\bar{x}}^{j+1} + q\bar{\alpha}_5^{j+1}\bar{u}_{\bar{y}}^{j+1} + \bar{\alpha}_6^{j+1}\bar{u}^{j+1}) + \frac{2}{\tau}\bar{u}^{j+1} \\
\text{and} \quad Z^{j} &:= (p^2\bar{\alpha}_1^{j}\bar{u}_{\bar{x}\bar{x}}^{j} + q^2\bar{\alpha}_2^{j}\bar{u}_{\bar{y}\bar{y}}^{j} + pq\bar{\alpha}_3^{j}\bar{u}_{\bar{x}\bar{y}}^{j} + p\bar{\alpha}_4{}^{j}\bar{u}_{\bar{x}}^{j} + q\bar{\alpha}_5^{j}\bar{u}_{\bar{y}}^{j} + \bar{\alpha}_6^{j}\bar{u}^{j}) + \frac{2}{\tau}\bar{u}^{j}.
\end{aligned}$$

Then, (3.20) becomes $Z^{j+1} = Z^j + \bar{f}^{j+1} + \bar{f}^j$.

Therefore,

$$\begin{aligned}
&\int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}(Z^{j+1} - \bar{f}^{j+1})d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
= & -\Big[\int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\Big[p^2\bar{\alpha}_1^{j+1}(\xi_1,\eta_1)\frac{\partial^2\bar{u}^{j+1}(\xi_1,\eta_1)}{\partial\xi_1^2}\Big]d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
+ & \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\Big[q^2\bar{\alpha}_2^{j+1}(\xi_1,\eta_1)\frac{\partial^2\bar{u}^{j+1}(\xi_1,\eta_1)}{\partial\eta_1^2}\Big]d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
+ & \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\Big[pq\bar{\alpha}_3^{j+1}(\xi_1,\eta_1)\frac{\partial^2\bar{u}^{j+1}(\xi_1,\eta_1)}{\partial\xi\partial\eta_1}\Big]d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
+ & \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\Big[p\bar{\alpha}_4^{j+1}(\xi_1,\eta_1)\frac{\partial\bar{u}^{j+1}(\xi_1,\eta_1)}{\partial\xi_1}\Big]d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
+ & \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\Big[q\bar{\alpha}_5^{j+1}(\xi_1,\eta_1)\frac{\partial\bar{u}^{j+1}(\xi_1,\eta_1)}{\partial\eta_1}\Big]d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
+ & \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\Big[\bar{\alpha}_6^{j+1}(\xi_1,\eta_1)\bar{u}^{j+1}(\xi_1,\eta_1)\Big]d\xi_1 d\xi_2 d\eta_1 d\eta_2\Big] \\
+ & \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\frac{2}{\tau}\Big[\bar{u}^{j+1}(\xi_1,\eta_1)\Big]d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
- & \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\Big[\bar{f}^{j+1}(\xi_1,\eta_1)\Big]d\xi_1 d\xi_2 d\eta_1 d\eta_2.
\end{aligned}$$

Using the integrations by parts, we have

$$\begin{aligned}
&\int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}(Z^{j+1} - \bar{f}^{j+1})d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\
= & -\Big[\int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}p^2\Big[\bar{\alpha}_1^{j+1}\bar{u}^{j+1} - 2\int_{-1}^{\bar{x}_k}\frac{\partial\bar{\alpha}_1^{j+1}}{\partial\xi_1}\bar{u}^{j+1}d\xi_2 \\
+ & \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\frac{\partial^2\bar{\alpha}_1^{j+1}}{\partial\xi_1^2}\bar{u}^{j+1}d\xi_1 d\xi_2\Big]d\eta_1 d\eta_2 \\
+ & \int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}q^2\Big[\bar{\alpha}_2^{j+1}\bar{u}^{j+1} - 2\int_{-1}^{\bar{y}_s}\frac{\partial\bar{\alpha}_2^{j+1}}{\partial\eta_1}\bar{u}^{j+1}d\eta_2
\end{aligned}$$

$$
\begin{aligned}
&+ \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\frac{\partial^2\bar{\alpha}_2^{j+1}}{\partial\eta_1^2}\bar{u}^{j+1}d\eta_1 d\eta_2\Big]d\xi_1 d\xi_2\\
&+ \int_{-1}^{\bar{y}_s}\int_{-1}^{\bar{x}_k} pq\Big[\bar{\alpha}_3^{j+1}\bar{u}^{j+1} - \int_{-1}^{\xi_2}\frac{\partial\bar{\alpha}_3^{j+1}}{\partial\xi_1}\bar{u}^{j+1}d\xi_1 - \int_{-1}^{\eta_2}\frac{\partial\bar{\alpha}_3^{j+1}}{\partial\eta_1}\bar{u}^{j+1}d\eta_1\\
&+ \int_{-1}^{\eta_2}\int_{-1}^{\xi_2}\frac{\partial^2\bar{\alpha}_3^{j+1}}{\partial\xi_1\partial\eta_1}\bar{u}^{j+1}d\xi_1 d\eta_1\Big]d\xi_2 d\eta_2\\
&+ \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k} p\Big[\bar{\alpha}_4^{j+1}\bar{u}^{j+1} - \int_{-1}^{\xi_2}\frac{\partial\bar{\alpha}_4^{j+1}}{\partial\xi_1}\bar{u}^{j+1}d\xi_1\Big]d\xi_2 d\eta_1\eta_2\\
&+ \int_{-1}^{\bar{y}_s}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2} q\Big[\bar{\alpha}_5^{j+1}\bar{u}^{j+1} - \int_{-1}^{\eta_2}\frac{\partial\bar{\alpha}_5^{j+1}}{\partial\eta_1}\bar{u}^{j+1}d\eta_1\Big]d\xi_1 d\xi_2\eta_2\\
&+ \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\bar{\alpha}_6^{j+1}\bar{u}^{j+1}d\xi_1 d\xi_2 d\eta_1 d\eta_2\Big]\\
&+ \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\frac{2}{\tau}\bar{u}^{j+1}d\xi_1 d\xi_2 d\eta_1 d\eta_2\\
&- \int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}\bar{f}^{j+1}d\xi_1 d\xi_2 d\eta_1 d\eta_2\\
&+ \bar{x}_k f_0(\bar{y}_s) + f_1(\bar{y}_s) + \bar{y}_s g_0(\bar{x}_k) + g_1(\bar{x}_k), \qquad (3.23)
\end{aligned}
$$

where $f_0(\bar{y}_s)$, $f_1(\bar{y}_s)$, $g_0(\bar{x}_k)$ and $g_1(\bar{x}_k)$ are arbitrary functions of integration assumed to be approximated by Chebyshev interpolating polynomials,

$$
g_l(\bar{x}_k) = \sum_{i=0}^{P_1-1} g_i^{(l)} T_i(\bar{x}_k), \text{ for } l \in \{0,1\}, \qquad (3.24)
$$

$$
f_l(\bar{y}_s) = \sum_{j=0}^{P_2-1} f_j^{(l)} T_j(\bar{y}_s), \text{ for } l \in \{0,1\}, \qquad (3.25)
$$

where $g_0^l, g_1^l, g_2^l, ..., g_{P_1-1}^l$ and $f_0^l, f_1^l, f_2^l, ..., f_{P_2-1}^l$ are the unknown values of those interpolated points which are determined from the given boundary conditions.

Remark: The number of these unknown values should be equal to the number of boundary points in order that a system of linear equations is solvable. Thus, the number of these unknown values and the number of the boundary points are $2P_1 + 2P_2$ and $2P_1 + 2P_2$, respectively. Thus, we select $P_1 = N_1$ and $P_2 = N_2$.

For $\int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}(Z^j + \bar{f}^j)d\xi_1 d\xi_2 d\eta_1 d\eta_2$, we obtain the expression similarly to the case of $\int_{-1}^{\bar{y}_s}\int_{-1}^{\eta_2}\int_{-1}^{\bar{x}_k}\int_{-1}^{\xi_2}(Z^{j+1} - \bar{f}^{j+1})d\xi_1 d\xi_2 d\eta_1 d\eta_2$.

Step 4: We can transform (3.23) into the matrix form,

$$
\begin{aligned}
- &\quad [p^2\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_1^{j+1}\mathbf{u}^{j+1} - 2\mathbf{A}_x\bar{\boldsymbol{\alpha}}_{1,\bar{x}}^{j+1}\mathbf{u}^{j+1} + \mathbf{A}_x^2\bar{\boldsymbol{\alpha}}_{1,\bar{x}\bar{x}}^{j+1}\mathbf{u}^{j+1}) \\
+ &\quad q^2\mathbf{A}_x^2(\bar{\boldsymbol{\alpha}}_2^{j+1}\mathbf{u}^{j+1} - 2\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{2,\bar{y}}^{j+1}\mathbf{u}^{j+1} + \mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_{2,\bar{y}\bar{y}}^{j+1}\mathbf{u}^{j+1}) \\
+ &\quad pq\mathbf{A}_x\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_3^{j+1}\mathbf{u}^{j+1} - \mathbf{A}_x\bar{\boldsymbol{\alpha}}_{3,\bar{x}}^{j+1}\mathbf{u}^{j+1} - \mathbf{A}_y\bar{\boldsymbol{\alpha}}_{3,\bar{y}}^{j+1}\mathbf{u}^{j+1} + \mathbf{A}_x\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{3,\bar{x}\bar{y}}^{j+1}\mathbf{u}^{j+1}) \\
+ &\quad p\mathbf{A}_x\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_4^{j+1}\mathbf{u}^{j+1} - \mathbf{A}_x\bar{\boldsymbol{\alpha}}_{4,\bar{x}}^{j+1}\mathbf{u}^{j+1}) \\
+ &\quad q\mathbf{A}_x^2\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_5^{j+1}\mathbf{u}^{j+1} - \mathbf{A}_y\bar{\boldsymbol{\alpha}}_{5,\bar{y}}^{j+1}\mathbf{u}^{j+1})] \\
+ &\quad \mathbf{A}_x^2\mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_6^{j+1}\mathbf{u}^{j+1}] + \mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau}\mathbf{u}^{j+1} + \mathbf{X}\boldsymbol{\Phi}_y\mathbf{f}_0 + \boldsymbol{\Phi}_y\mathbf{f}_1 + \mathbf{Y}\boldsymbol{\Phi}_x\mathbf{g}_0 + \boldsymbol{\Phi}_x\mathbf{g}_1 \\
= &\quad p^2\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_1^{j}\mathbf{u}^{j} - 2\mathbf{A}_x\bar{\boldsymbol{\alpha}}_{1,\bar{x}}^{j}\mathbf{u}^{j} + \mathbf{A}_x^2\bar{\boldsymbol{\alpha}}_{1,\bar{x}\bar{x}}^{j}\mathbf{u}^{j}) \\
+ &\quad q^2\mathbf{A}_x^2(\bar{\boldsymbol{\alpha}}_2^{j}\mathbf{u}^{j} - 2\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{2,\bar{y}}^{j}\mathbf{u}^{j} + \mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_{2,\bar{y}\bar{y}}^{j}\mathbf{u}^{j}) \\
+ &\quad pq\mathbf{A}_x\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_3^{j}\mathbf{u}^{j} - \mathbf{A}_x\bar{\boldsymbol{\alpha}}_{3,\bar{x}}^{j}\mathbf{u}^{j} - \mathbf{A}_y\bar{\boldsymbol{\alpha}}_{3,\bar{y}}^{j}\mathbf{u}^{j} + \mathbf{A}_x\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{3,\bar{x}\bar{y}}^{j}\mathbf{u}^{j}) \\
+ &\quad p\mathbf{A}_x\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_4^{j}\mathbf{u}^{j} - \mathbf{A}_x\bar{\boldsymbol{\alpha}}_{4,\bar{x}}^{j}\mathbf{u}^{j}) \\
+ &\quad q\mathbf{A}_x^2\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_5^{j}\mathbf{u}^{j} - \mathbf{A}_y\bar{\boldsymbol{\alpha}}_{5,\bar{y}}^{j}\mathbf{u}^{j}) \\
+ &\quad \mathbf{A}_x^2\mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_6^{j}\mathbf{u}^{j}] + \mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau}\mathbf{u}^{j} + \mathbf{A}_x^2\mathbf{A}_y^2\bar{\mathbf{F}}^{j+1} + \mathbf{A}_x^2\mathbf{A}_y^2\bar{\mathbf{F}}^{j},
\end{aligned}
\tag{3.26}
$$

where for $r \in \{j, j+1\}$, $l \in \{0,1\}$ and $i \in \{1,2,3,4,5,6\}$, $\bar{\boldsymbol{\alpha}}_i^r = \mathrm{diag}[\bar{\alpha}_i^r(\bar{x}_1,\bar{y}_1), \bar{\alpha}_i^r(\bar{x}_2,\bar{y}_2), \bar{\alpha}_i^r(\bar{x}_3,\bar{y}_3), ..., \bar{\alpha}_i^r(\bar{x}_M,\bar{y}_M)]_{M\times M}^T$, $\mathbf{g}_l = [g_0^l, g_1^l, g_2^l, ..., g_{P_1-1}^l]^T$, $\mathbf{f}_l = [f_0^l, f_1^l, f_2^l, ..., f_{P_2-1}^l]^T$, $\mathbf{u}^r = [\mathbf{u}^r(\cdot,\bar{y}_1), \mathbf{u}^r(\cdot,\bar{y}_2), ..., \mathbf{u}^r(\cdot,\bar{y}_{N_2})]^T$ with $\mathbf{u}^r(\cdot,\bar{y}_s) = [\mathbf{u}^r(\bar{x}_1,\bar{y}_s), \mathbf{u}^r(\bar{x}_2,\bar{y}_s), ..., \mathbf{u}^r(\bar{x}_{N_1},\bar{y}_s)]^T$, $\mathbf{F}^r = [\mathbf{f}^r(\cdot,\bar{y}_1), \mathbf{f}^r(\cdot,\bar{y}_2), ..., \mathbf{f}^r(\cdot,\bar{y}_{N_2})]^T$ with $\mathbf{f}^{\,r}(\cdot,\bar{y}_s) = [\mathbf{f}^r(\bar{x}_1,\bar{y}_s), \mathbf{f}^r(\bar{x}_2,\bar{y}_s), ..., \mathbf{f}^r(\bar{x}_{N_1},\bar{y}_s)]^T$, $\mathbf{A}_x$ and $\mathbf{A}_y$ are defined in (2.22) and (2.26), respectively. The diagonal matrices $\mathbf{X}$ and $\mathbf{Y}$ are defined by

$$
\mathbf{X} = \underbrace{\begin{bmatrix} \mathbf{X}_0 & 0 & 0 & \dots & 0 \\ 0 & \mathbf{X}_0 & 0 & \dots & 0 \\ 0 & 0 & \mathbf{X}_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{X}_0 \end{bmatrix}}_{N_2 \text{ blocks}}{}_{M\times M} \quad \text{with } \mathbf{X}_0 = \begin{bmatrix} \bar{x}_1 & 0 & 0 & \dots & 0 \\ 0 & \bar{x}_2 & 0 & \dots & 0 \\ 0 & 0 & \bar{x}_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \bar{x}_{N_1} \end{bmatrix}_{N_1\times N_1},
$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 & 0 & 0 & \dots & 0 \\ 0 & \mathbf{Y}_2 & 0 & \dots & 0 \\ 0 & 0 & \mathbf{Y}_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{Y}_{N_2} \end{bmatrix}_{M\times M} \quad \text{with } \mathbf{Y}_s = \begin{bmatrix} \bar{y}_s & 0 & 0 & \dots & 0 \\ 0 & \bar{y}_s & 0 & \dots & 0 \\ 0 & 0 & \bar{y}_s & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \bar{y}_s \end{bmatrix}_{N_1\times N_1} .$$

Let $\hat{x}_m = \mathbf{X}_{mm}$ and $\hat{y}_m = \mathbf{Y}_{mm}$ for $m \in \{1, 2, 3, ..., M\}$. From (3.24) and (3.25), we obtain $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_y$, which their elements can be found by (2.1), as follow:

$$\mathbf{\Phi}_x = \begin{bmatrix} T_0(\hat{x}_1) & T_1(\hat{x}_1) & T_2(\hat{x}_1) & \dots & T_{P_1-1}(\hat{x}_1) \\ T_0(\hat{x}_2) & T_1(\hat{x}_2) & T_2(\hat{x}_2) & \dots & T_{P_1-1}(\hat{x}_2) \\ T_0(\hat{x}_3) & T_1(\hat{x}_3) & T_2(\hat{x}_3) & \dots & T_{P_1-1}(\hat{x}_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_0(\hat{x}_M) & T_1(\hat{x}_M) & T_2(\hat{x}_M) & \dots & T_{P_1-1}(\hat{x}_M) \end{bmatrix}_{M\times P_1} ,$$

$$\mathbf{\Phi}_y = \begin{bmatrix} T_0(\hat{y}_1) & T_1(\hat{y}_1) & T_2(\hat{y}_1) & \dots & T_{P_2-1}(\hat{y}_1) \\ T_0(\hat{y}_2) & T_1(\hat{y}_2) & T_2(\hat{y}_2) & \dots & T_{P_2-1}(\hat{y}_2) \\ T_0(\hat{y}_3) & T_1(\hat{y}_3) & T_2(\hat{y}_3) & \dots & T_{P_2-1}(\hat{y}_3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_0(\hat{y}_M) & T_1(\hat{y}_M) & T_2(\hat{y}_M) & \dots & T_{P_2-1}(\hat{y}_M) \end{bmatrix}_{M\times P_2} .$$

Let, for $r \in \{j, j+1\}$,

$$\begin{aligned} \mathbf{Z}^r \;=\;& p^2\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_1^r - 2\mathbf{A}_x\bar{\boldsymbol{\alpha}}_{1,\bar{x}}^r + \mathbf{A}_x^2\bar{\boldsymbol{\alpha}}_{1,\bar{x}\bar{x}}^r) + q^2\mathbf{A}_x^2(\bar{\boldsymbol{\alpha}}_2^r - 2\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{2,\bar{y}}^r + \mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_{2,\bar{y}\bar{y}}^r) \\ &+pq\mathbf{A}_x\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_3^r - \mathbf{A}_x\bar{\boldsymbol{\alpha}}_{3,\bar{x}}^r - \mathbf{A}_y\bar{\boldsymbol{\alpha}}_{3,\bar{y}}^r + \mathbf{A}_x\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{3,\bar{x}\bar{y}}^r) + p\mathbf{A}_x\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_4^r - \mathbf{A}_x\bar{\boldsymbol{\alpha}}_{4,\bar{x}}^r) \\ &+q\mathbf{A}_x^2\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_5^r - \mathbf{A}_y\bar{\boldsymbol{\alpha}}_{5,\bar{y}}^r\mathbf{u}^r) + \mathbf{A}_x^2\mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_6^r. \end{aligned}$$

Then, from (3.26), we have

$$\Big(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} - \mathbf{Z}^{j+1}\Big)\mathbf{u}^{j+1} + \mathbf{X}\mathbf{\Phi}_y\mathbf{f}_0 + \mathbf{\Phi}_y\mathbf{f}_1 + \mathbf{Y}\mathbf{\Phi}_x\mathbf{g}_0 + \mathbf{\Phi}_x\mathbf{g}_1$$

$$= \left(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} + \mathbf{Z}^j\right)\mathbf{u}^j + \mathbf{A}_x^2\mathbf{A}_y^2(\bar{\mathbf{F}}^{j+1} + \bar{\mathbf{F}}^j). \tag{3.27}$$

Step 5: We consider the four boundary conditions.

For the left side boundary condition, we let $\bar{u}(-1, \bar{y}, t) = \bar{F}_1(\bar{y}, t)$. Then, we have

$$\bar{u}(-1, \bar{y}_s, t_{j+1}) = \sum_{n=0}^{N_1-1} c_n T_n(-1) = \sum_{n=0}^{N_1-1} c_n(-1)^n = \mathbf{t}_l\mathbf{c} = \mathbf{t}_l\mathbf{T}^{-1}\mathbf{u}^{j+1}(\cdot, \bar{y}_s) = \bar{F}_1^{j+1}(\bar{y}_s)$$

for $s \in \{1, 2, 3, ..., N_2\}$ or in matrix form

$$\begin{bmatrix} \mathbf{t}_l\mathbf{T}^{-1} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{t}_l\mathbf{T}^{-1} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{t}_l\mathbf{T}^{-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{t}_l\mathbf{T}^{-1} \end{bmatrix}_{N_2 \times M} \begin{bmatrix} \mathbf{u}^{j+1}(\cdot, \bar{y}_1) \\ \mathbf{u}^{j+1}(\cdot, \bar{y}_2) \\ \mathbf{u}^{j+1}(\cdot, \bar{y}_3) \\ \vdots \\ \mathbf{u}^{j+1}(\cdot, \bar{y}_{N_2}) \end{bmatrix} = \begin{bmatrix} \bar{F}_1^{j+1}(\bar{y}_1) \\ \bar{F}_1^{j+1}(\bar{y}_2) \\ \bar{F}_1^{j+1}(\bar{y}_2) \\ \vdots \\ \bar{F}_1^{j+1}(\bar{y}_{N_2}) \end{bmatrix},$$

which we denote it by $\mathbf{T}_l\mathbf{u} = \bar{\mathbf{F}}_1^{j+1}$, where the row vector $\mathbf{t}_l = [1, -1, 1, -1, ..., (-1)^{N_1-1}]$, $\mathbf{u}(\cdot, \bar{y}_s) = [\bar{u}(\bar{x}_1, \bar{y}_s), \bar{u}(\bar{x}_2, \bar{y}_s), \bar{u}(\bar{x}_3, \bar{y}_s), ..., \bar{u}(\bar{x}_{N_1}, \bar{y}_s)]^T$ and $\mathbf{T}^{-1}$ is an $N_1 \times N_1$ matrix.

For the right side boundary condition, we let $\bar{u}(1, \bar{y}, t) = \bar{F}_2(\bar{y}, t)$. Then, we have

$$\bar{u}(1, \bar{y}_s, t_{j+1}) = \sum_{n=0}^{N_1-1} c_n T_n(1) = \sum_{n=0}^{N_1-1} c_n(1)^n = \mathbf{t}_r\mathbf{c} = \mathbf{t}_r\mathbf{T}^{-1}\mathbf{u}^{j+1}(\cdot, \bar{y}_s) = \bar{F}_2^{j+1}(\bar{y}_s)$$

for $s \in \{1, 2, 3, ..., N_2\}$ or in matrix form

$$\begin{bmatrix} \mathbf{t}_r\mathbf{T}^{-1} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{t}_r\mathbf{T}^{-1} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{t}_r\mathbf{T}^{-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{t}_r\mathbf{T}^{-1} \end{bmatrix}_{N_2 \times M} \begin{bmatrix} \mathbf{u}^{j+1}(\cdot, \bar{y}_1) \\ \mathbf{u}^{j+1}(\cdot, \bar{y}_2) \\ \mathbf{u}^{j+1}(\cdot, \bar{y}_3) \\ \vdots \\ \mathbf{u}^{j+1}(\cdot, \bar{y}_{N_2}) \end{bmatrix} = \begin{bmatrix} \bar{F}_2^{j+1}(\bar{y}_1) \\ \bar{F}_2^{j+1}(\bar{y}_2) \\ \bar{F}_2^{j+1}(\bar{y}_2) \\ \vdots \\ \bar{F}_2^{j+1}(\bar{y}_{N_2}) \end{bmatrix},$$

which we denote it by $\mathbf{T}_r\mathbf{u} = \bar{\mathbf{F}}_2^{j+1}$, where the row vector $\mathbf{t}_r = [1, 1, 1, 1, ..., 1^{N_1-1}]$, $\mathbf{u}(\cdot, \bar{y}_s) = [\bar{u}(\bar{x}_1, \bar{y}_s), \bar{u}(\bar{x}_2, \bar{y}_s), \bar{u}(\bar{x}_3, \bar{y}_s), ..., \bar{u}(\bar{x}_{N_1}, \bar{y}_s)]^T$ and $\mathbf{T}^{-1}$ is an $N_1 \times N_1$ matrix.

For the bottom side boundary condition, we let $\bar{u}(\bar{x}, -1, t) = \bar{G}_1(\bar{x}, t)$. Then, we have

$$\bar{u}(\bar{x}_k, -1, t_{j+1}) = \sum_{n=0}^{N_2-1} c_n T_n(-1) = \sum_{n=0}^{N_2-1} c_n (-1)^n = \mathbf{t}_b\mathbf{c} = \mathbf{t}_b\mathbf{T}^{-1}\tilde{\mathbf{u}}^{j+1}(\bar{x}_k, \cdot) = \bar{G}_1^{j+1}(\bar{x}_k)$$

for $s \in \{1, 2, 3, ..., N_2\}$ or in matrix form

$$\begin{bmatrix} \mathbf{t}_b\mathbf{T}^{-1} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{t}_b\mathbf{T}^{-1} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{t}_b\mathbf{T}^{-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{t}_b\mathbf{T}^{-1} \end{bmatrix}_{N_1 \times M} \begin{bmatrix} \tilde{\mathbf{u}}^{j+1}(\bar{x}_1, \cdot) \\ \tilde{\mathbf{u}}^{j+1}(\bar{x}_2, \cdot) \\ \tilde{\mathbf{u}}^{j+1}(\bar{x}_3, \cdot) \\ \vdots \\ \tilde{\mathbf{u}}^{j+1}(\bar{x}_{N_1}, \cdot) \end{bmatrix} = \begin{bmatrix} \bar{G}_1^{j+1}(\bar{x}_1) \\ \bar{G}_1^{j+1}(\bar{x}_2) \\ \bar{G}_1^{j+1}(\bar{x}_3) \\ \vdots \\ \bar{G}_1^{j+1}(\bar{x}_{N_1}) \end{bmatrix},$$

which we denote it by $\mathbf{T}_b\tilde{\mathbf{u}} = \bar{\mathbf{G}}_1^{j+1}$. By using (2.24), we obtain $\mathbf{T}_b\mathbf{P}^{-1}\mathbf{u} = \bar{\mathbf{G}}_1^{j+1}$, where the row vector $\mathbf{t}_b = [1, -1, 1, -1, ..., (-1)^{N_2-1}]$, $\mathbf{u}(\bar{x}_k, \cdot) = [\bar{u}(\bar{x}_k, \bar{y}_1), \bar{u}(\bar{x}_k, \bar{y}_2), \bar{u}(\bar{x}_k, \bar{y}_3), ..., \bar{u}(\bar{x}_k, \bar{y}_{N_2})]^T$ and $\mathbf{T}^{-1}$ is an $N_2 \times N_2$ matrix.

For the upper side boundary condition, we let $\bar{u}(\bar{x}, 1, t) = \bar{G}_2(\bar{x}, t)$. Then, we have

$$\bar{u}(\bar{x}_k, 1, t_{j+1}) = \sum_{n=0}^{N_2-1} c_n T_n(1) = \sum_{n=0}^{N_2-1} c_n (1)^n = \mathbf{t}_u\mathbf{c} = \mathbf{t}_u\mathbf{T}^{-1}\tilde{\mathbf{u}}^{j+1}(\bar{x}_k, \cdot) = \bar{G}_2^{j+1}(\bar{x}_k)$$

for $s \in \{1, 2, 3, ..., N_2\}$ or in matrix form

$$\begin{bmatrix} \mathbf{t}_u\mathbf{T}^{-1} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{t}_u\mathbf{T}^{-1} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{t}_u\mathbf{T}^{-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{t}_u\mathbf{T}^{-1} \end{bmatrix}_{N_1 \times M} \begin{bmatrix} \tilde{\mathbf{u}}^{j+1}(\bar{x}_1, \cdot) \\ \tilde{\mathbf{u}}^{j+1}(\bar{x}_2, \cdot) \\ \tilde{\mathbf{u}}^{j+1}(\bar{x}_3, \cdot) \\ \vdots \\ \tilde{\mathbf{u}}^{j+1}(\bar{x}_{N_1}, \cdot) \end{bmatrix} = \begin{bmatrix} \bar{G}_2^{j+1}(\bar{x}_1) \\ \bar{G}_2^{j+1}(\bar{x}_2) \\ \bar{G}_2^{j+1}(\bar{x}_3) \\ \vdots \\ \bar{G}_2^{j+1}(\bar{x}_{N_1}) \end{bmatrix},$$

which we denote it by $\mathbf{T}_u\tilde{\mathbf{u}} = \bar{\mathbf{G}}_2^{j+1}$. By using (2.24), we obtain $\mathbf{T}_u\mathbf{P}^{-1}\mathbf{u} = \bar{\mathbf{G}}_2^{j+1}$, where the row vector $\mathbf{t}_u = [1, 1, 1, 1, ..., 1^{N_2-1}], \mathbf{u}(\bar{x}_k, \cdot) = [\bar{u}(\bar{x}_k, \bar{y}_1), \bar{u}(\bar{x}_k, \bar{y}_2), \bar{u}(\bar{x}_k, \bar{y}_3), ...,$ $\bar{u}(\bar{x}_k, \bar{y}_{N_2})]^T$ and $\mathbf{T}^{-1}$ is an $N_2 \times N_2$ matrix.

Thus, all boundary conditions can be represented in matrix forms as $\mathbf{T}_l\mathbf{u} = \bar{\mathbf{F}}_1^{j+1}, \mathbf{T}_r\mathbf{u} = \bar{\mathbf{F}}_2^{j+1}, \mathbf{T}_b\mathbf{P}^{-1}\mathbf{u} = \bar{\mathbf{G}}_1^{j+1}$ and $\mathbf{T}_u\mathbf{P}^{-1}\mathbf{u} = \bar{\mathbf{G}}_2^{j+1}$.

Step 6: Construct the system of linear equations from (3.27) in Step 4 and all boundary conditions in Step 5. Then, we obtain

$$\begin{bmatrix} \mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} - \mathbf{Z}^{j+1} & \mathbf{X}\mathbf{\Phi}_y & \mathbf{\Phi}_y & \mathbf{Y}\mathbf{\Phi}_x & \mathbf{\Phi}_x \\ \mathbf{T}_l & 0 & 0 & \dots & 0 \\ \mathbf{T}_r & 0 & 0 & \dots & 0 \\ \mathbf{T}_b\mathbf{P}^{-1} & \vdots & \vdots & \ddots & \vdots \\ \mathbf{T}_u\mathbf{P}^{-1} & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{j+1} \\ \mathbf{f}_0 \\ \mathbf{f}_1 \\ \mathbf{g}_0 \\ \mathbf{g}_1 \end{bmatrix}$$
$$= \begin{bmatrix} \left(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} + \mathbf{Z}^j\right)\mathbf{u}^j + \mathbf{A}_x^2\mathbf{A}_y^2(\bar{\mathbf{F}}^{j+1} + \bar{\mathbf{F}}^j) \\ \bar{\mathbf{F}}_1^{j+1} \\ \bar{\mathbf{F}}_2^{j+1} \\ \bar{\mathbf{G}}_1^{j+1} \\ \bar{\mathbf{G}}_2^{j+1} \end{bmatrix}. \tag{3.28}$$

We can solve for the solutions $\bar{u}(\bar{x}, \bar{y}, t_{j+1})$ for $(\bar{x}, \bar{y}) \in [-1, 1] \times [-1, 1]$. Then, by using the transformation $x = \frac{1}{2}[(b-a)\bar{x} + a + b]$ and $y = \frac{1}{2}[(d-c)\bar{y} + c + d]$, we can obtain the approximate solution $u(x, y)$ for $(x, y) \in [a, b] \times [c, d]$.

3.4 Numerical Examples For Two-Dimensional Time-Dependent linear PDEs

In this section, we use our proposed method to find the approximate solutions of some time-dependent two-dimensional linear PDEs. For an error of solution, we use the average relative error (ARE) defined by

$$\text{ARE} = \frac{1}{M}\sum_{m=1}^{M} \Big| \frac{u^*(x_m, y_m, , t) - u(x_m, y_m, t)}{u^*(x_m, y_m, t)} \Big|,$$

where u^* and u are the exact and numerical solutions, respectively.

Example 3.4. Consider a time-dependent linear PDE in which the coefficients do not depend on time.

$$\begin{aligned}
\frac{\partial u}{\partial t} &= (x^2+y^2+1)\frac{\partial^2 u}{\partial x^2} + (x^2+y^2+1)\frac{\partial^2 u}{\partial y^2} + 2x\frac{\partial u}{\partial x} + 2y\frac{\partial u}{\partial y} \\
&\quad - 2u + [1 - 2(x^2+y^2+x+y)]e^{x+y+t}, \quad (x,y) \in (0,1)\times(0,1), \quad t \in (0,1), \\
u(x,y,0) &= e^{x+y}, \quad (x,y) \in [0,1]\times[0,1], \\
u(x,0,t) &= e^{x+t}, \quad u(x,1,t) = e^{x+1+t}, \quad x \in [0,1], \quad t \in [0,1], \\
u(0,y,t) &= e^{y+t}, \quad u(1,y,t) = e^{1+y+t}, \quad y \in [0,1], \quad t \in [0,1].
\end{aligned}$$

The exact solution for this problem is $u(x, y, t) = e^{x+y+t}$. We first transform our domain $\Omega = [0, 1] \times [0, 1]$ by using the transformations $\bar{x} = 2x - 1$ and $\bar{y} = 2y - 1$. By our numerical algorithm, this problem can be written in matrix form as

$$\begin{aligned}
&\Big(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} - \mathbf{Z}^{j+1}\Big)\mathbf{u}^{j+1} + \mathbf{X}\mathbf{\Phi}_y\mathbf{f}_0 + \mathbf{\Phi}_y\mathbf{f}_1 + \mathbf{Y}\mathbf{\Phi}_x\mathbf{g}_0 + \mathbf{\Phi}_x\mathbf{g}_1 \\
&= \Big(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} + \mathbf{Z}^j\Big)\mathbf{u}^j + \mathbf{A}_x^2\mathbf{A}_y^2(\bar{\mathbf{F}}^{j+1} + \bar{\mathbf{F}}^j).
\end{aligned}$$

From the boundary conditions, we have $\mathbf{T}_l\mathbf{u} = \bar{\mathbf{F}}_1^{j+1}$, $\mathbf{T}_r\mathbf{u} = \bar{\mathbf{F}}_2^{j+1}$, $\mathbf{T}_b\mathbf{P}^{-1}\mathbf{u} = \bar{\mathbf{G}}_1^{j+1}$ and $\mathbf{T}_u\mathbf{P}^{-1}\mathbf{u} = \bar{\mathbf{G}}_2^{j+1}$. Thus, we can construct the linear system as

$$\begin{bmatrix}
\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} - \mathbf{Z}^{j+1} & \mathbf{X}\mathbf{\Phi}_y & \mathbf{\Phi}_y & \mathbf{Y}\mathbf{\Phi}_x & \mathbf{\Phi}_x \\
\mathbf{T}_l & 0 & 0 & \cdots & 0 \\
\mathbf{T}_r & 0 & 0 & \cdots & 0 \\
\mathbf{T}_b\mathbf{P}^{-1} & \vdots & \vdots & \ddots & \vdots \\
\mathbf{T}_u\mathbf{P}^{-1} & 0 & 0 & \cdots & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{u}^{j+1} \\ \mathbf{f}_0 \\ \mathbf{f}_1 \\ \mathbf{g}_0 \\ \mathbf{g}_1
\end{bmatrix}$$

$$= \begin{bmatrix} \Big(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} + \mathbf{Z}^j\Big)\mathbf{u}^j + \mathbf{A}_x^2\mathbf{A}_y^2(\bar{\mathbf{F}}^{j+1} + \bar{\mathbf{F}}^j) \\ \bar{\mathbf{F}}_1^{j+1} \\ \bar{\mathbf{F}}_2^{j+1} \\ \bar{\mathbf{G}}_1^{j+1} \\ \bar{\mathbf{G}}_2^{j+1} \end{bmatrix},$$

where, for $r \in \{j, j+1\}$,

$$\begin{aligned}
\mathbf{u}^r &= [\bar{u}^r(\bar{x}_1,\bar{y}_1), \bar{u}^r(\bar{x}_2,\bar{y}_2), \bar{u}^r(\bar{x}_3,\bar{y}_3), ..., \bar{u}^r(\bar{x}_{N_1},\bar{y}_{N_2})]^T, \\
\bar{\mathbf{F}}^{\ r} &= \Big[(1-2(x_1^2+y_1^2+x_1+y_1))e^{x_1+y_1+t_r}, (1-2(x_2^2+y_2^2+x_2+y_2))e^{x_2+y_2+t_r}, \\
&\quad (1-2(x_3^2+y_3^2+x_3+y_3))e^{x_3+y_3+t_r}, ..., \\
&\quad (1-2(x_{N_1}^2+y_{N_2}^2+x_{N_1}+y_{N_2}))e^{x_{N_1}+y_{N_2}+t_r}\Big]^T, \\
\mathbf{Z}^r &= p^2\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_1^r - 2\mathbf{A}_x\bar{\boldsymbol{\alpha}}_{1,\bar{x}}^r + \mathbf{A}_x^2\bar{\boldsymbol{\alpha}}_{1,\bar{x}\bar{x}}^r) + q^2\mathbf{A}_x^2(\bar{\boldsymbol{\alpha}}_2^r - 2\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{2,\bar{y}}^r + \mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_{2,\bar{y}\bar{y}}^r) \\
&\quad + p\mathbf{A}_x\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_3^r - \mathbf{A}_x\bar{\boldsymbol{\alpha}}_{3,\bar{x}}^r) + q\mathbf{A}_x^2\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_4^r - \mathbf{A}_y\bar{\boldsymbol{\alpha}}_{4,\bar{y}}^r\mathbf{u}^r) + \mathbf{A}_x^2\mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_5^r, \\
\bar{\boldsymbol{\alpha}}_1^r &= \mathrm{diag}\Big(\frac{(\bar{x}_1+1)^2}{4} + \frac{(\bar{y}_1+1)^2}{4} + 1, \frac{(\bar{x}_2+1)^2}{4} + \frac{(\bar{y}_2+1)^2}{4} + 1 \\
&\quad \frac{(\bar{x}_3+1)^2}{4} + \frac{(\bar{y}_3+1)^2}{4} + 1, ..., \frac{(\bar{x}_{N_1}+1)^2}{4} + \frac{(\bar{y}_{N_2}+1)^2}{4} + 1\Big), \\
\bar{\boldsymbol{\alpha}}_{1,\bar{x}}^r &= \mathrm{diag}\Big(\frac{\bar{x}_1+1}{2}, \frac{\bar{x}_2+1}{2}, \frac{\bar{x}_3+1}{2}, ..., \frac{\bar{x}_{N_1}+1}{2}\Big), \\
\bar{\boldsymbol{\alpha}}_{1,\bar{x}\bar{x}}^r &= \mathrm{diag}\Big(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}\Big), \\
\bar{\boldsymbol{\alpha}}_2^r &= \mathrm{diag}\Big(\frac{(\bar{x}_1+1)^2}{4} + \frac{(\bar{y}_1+1)^2}{4} + 1, \frac{(\bar{x}_2+1)^2}{4} + \frac{(\bar{y}_2+1)^2}{4} + 1 \\
&\quad \frac{(\bar{x}_3+1)^2}{4} + \frac{(\bar{y}_3+1)^2}{4} + 1, ..., \frac{(\bar{x}_{N_1}+1)^2}{4} + \frac{(\bar{y}_{N_2}+1)^2}{4} + 1\Big), \\
\bar{\boldsymbol{\alpha}}_{2,\bar{y}}^r &= \mathrm{diag}\Big(\frac{\bar{y}_1+1}{2}, \frac{\bar{y}_2+1}{2}, \frac{\bar{y}_3+1}{2}, ..., \frac{\bar{y}_{N_2}+1}{2}\Big), \\
\bar{\boldsymbol{\alpha}}_{2,\bar{y}\bar{y}}^r &= \mathrm{diag}\Big(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}\Big), \\
\bar{\boldsymbol{\alpha}}_3^r &= \mathrm{diag}(\bar{x}_1+1, \bar{x}_2+1, \bar{x}_3+1, ..., \bar{x}_{N_1}+1), \\
\bar{\boldsymbol{\alpha}}_{3,\bar{x}}^r &= \mathrm{diag}(2, 2, 2, ..., 2), \\
\bar{\boldsymbol{\alpha}}_4^r &= \mathrm{diag}(\bar{y}_1+1, \bar{y}_2+1, \bar{y}_3+1, ..., \bar{y}_{N_2}+1), \\
\bar{\boldsymbol{\alpha}}_{4,\bar{y}}^r &= \mathrm{diag}(2, 2, 2, ..., 2), \\
\bar{\boldsymbol{\alpha}}_5^r &= \mathrm{diag}(-2, -2, -2, ..., -2),
\end{aligned}$$

$$\begin{aligned}
\bar{\mathbf{F}}_1^{j+1} &= [e^{\frac{\bar{x}_1+1}{2}+t_{j+1}}, e^{\frac{\bar{x}_2+1}{2}+t_{j+1}}, e^{\frac{\bar{x}_3+1}{2}+t_{j+1}}, ..., e^{\frac{\bar{x}_{N_1}+1}{2}+t_{j+1}}]^T,\\
\bar{\mathbf{F}}_2^{j+1} &= [e^{\frac{\bar{x}_1+1}{2}+t_{j+1}+1}, e^{\frac{\bar{x}_2+1}{2}+t_{j+1}+1}, e^{\frac{\bar{x}_3+1}{2}+t_{j+1}+1}, ..., e^{\frac{\bar{x}_{N_1}+1}{2}+t_{j+1}+1}]^T,\\
\bar{\mathbf{G}}_1^{j+1} &= [e^{\frac{\bar{y}_1+1}{2}+t_{j+1}}, e^{\frac{\bar{y}_2+1}{2}+t_{j+1}}, e^{\frac{\bar{y}_3+1}{2}+t_{j+1}}, ..., e^{\frac{\bar{y}_{N_2}+1}{2}+t_{j+1}}]^T,\\
\bar{\mathbf{G}}_2^{j+1} &= [e^{\frac{\bar{y}_1+1}{2}+t_{j+1}+1}, e^{\frac{\bar{y}_2+1}{2}+t_{j+1}+1}, e^{\frac{\bar{y}_3+1}{2}+t_{j+1}+1}, ..., e^{\frac{\bar{y}_{N_2}+1}{2}+t_{j+1}+1}]^T.
\end{aligned}$$

Table 3.8**:** The AREs for FIM (TPZ) in Example 3.4 when $\tau = 0.1$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	2.9684×10^{-2}	3.0513×10^{-2}	3.1105×10^{-2}	3.1536×10^{-2}
8	3.0513×10^{-2}	3.1323×10^{-2}	3.1924×10^{-2}	3.2364×10^{-2}
10	3.1105×10^{-2}	3.1924×10^{-2}	3.2534×10^{-2}	3.2981×10^{-2}
12	3.1536×10^{-2}	3.2364×10^{-2}	3.2981×10^{-2}	3.3434×10^{-2}

Table 3.9**:** The AREs for FIM (SIM) Example 3.4 when $\tau = 0.1$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	2.6237×10^{-2}	2.7622×10^{-2}	2.8513×10^{-2}	2.9114×10^{-2}
8	2.7622×10^{-2}	2.9074×10^{-2}	3.0013×10^{-2}	3.0647×10^{-2}
10	2.8513×10^{-2}	3.0013×10^{-2}	3.0984×10^{-2}	3.1639×10^{-2}
12	2.9114×10^{-2}	3.0647×10^{-2}	3.1639×10^{-2}	3.2308×10^{-2}

Table 3.10**:** The AREs for FIM (CBS) in Example 3.4 when $\tau = 0.1$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	1.1486×10^{-5}	1.0720×10^{-5}	1.0720×10^{-5}	1.0722×10^{-5}
8	1.0720×10^{-5}	9.9216×10^{-6}	9.9185×10^{-6}	9.9182×10^{-6}
10	1.0720×10^{-5}	9.9185×10^{-6}	9.9158×10^{-6}	9.9157×10^{-6}
12	1.0722×10^{-5}	9.9182×10^{-6}	9.9157×10^{-6}	9.9158×10^{-6}

Table 3.11**:** The AREs for FIM (CBS) in Example 3.4 when $\tau = 0.01$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	2.3278×10^{-6}	1.5463×10^{-6}	1.5601×10^{-6}	1.5632×10^{-6}
8	1.5463×10^{-6}	1.0197×10^{-7}	1.0099×10^{-7}	1.0099×10^{-7}
10	1.5601×10^{-6}	1.0099×10^{-7}	9.9978×10^{-8}	9.9978×10^{-8}
12	1.5632×10^{-6}	1.0099×10^{-7}	9.9978×10^{-8}	9.9981×10^{-8}

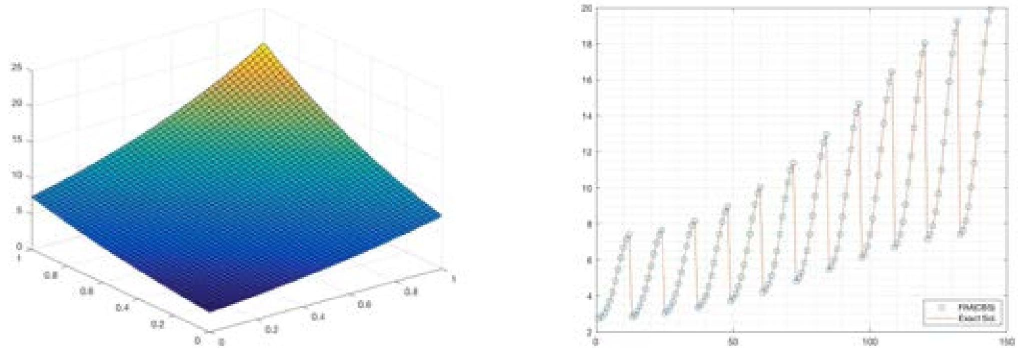


Figure 3.4**:** The surface and the grid points of the solution in Example 3.4 at $t = 1$.

Example 3.5**.** Consider a time-dependent linear PDE in which coefficients are functions in terms of both x and t.

$$
\begin{aligned}
&\frac{\partial u}{\partial t} = x^2 e^t \frac{\partial^2 u}{\partial x^2} + y^2 e^t \frac{\partial^2 u}{\partial y^2} + x e^t \frac{\partial u}{\partial x} + y e^t \frac{\partial u}{\partial y} + u - (4x^2 e^{2t} + 4y^2 e^{2t}),\\
&\qquad (x, y) \in (0, 2) \times (0, 2), \quad t \in (0, 1),\\
&u(x, y, 0) = x^2 + y^2 + 1, \quad (x, y) \in [0, 2] \times [0, 2],\\
&u(x, 0, t) = e^t(x^2 + 1), \quad u(x, 1, t) = e^t(x^2 + 5), \quad x \in [0, 2], \quad t \in [0, 1],\\
&u(0, y, t) = e^t(y^2 + 1), \quad u(1, y, t) = e^t(y^2 + 5), \quad y \in [0, 2], \quad t \in [0, 1].
\end{aligned}
$$

The exact solution for this problem is $u(x, y, t) = e^t(x^2 + y^2 + 1)$. We first transform our domain $\Omega = [0, 2] \times [0, 2]$ by using the transformations $\bar{x} = x - 1$ and $\bar{y} = y - 1$. By our numerical algorithm, this problem can be written in matrix form as

$$
\begin{aligned}
&\Big(\mathbf{A}_x^2 \mathbf{A}_y^2 \frac{2}{\tau} - \mathbf{Z}^{j+1}\Big)\mathbf{u}^{j+1} + \mathbf{X}\boldsymbol{\Phi}_y \mathbf{f}_0 + \boldsymbol{\Phi}_y \mathbf{f}_1 + \mathbf{Y}\boldsymbol{\Phi}_x \mathbf{g}_0 + \boldsymbol{\Phi}_x \mathbf{g}_1\\
&= \Big(\mathbf{A}_x^2 \mathbf{A}_y^2 \frac{2}{\tau} + \mathbf{Z}^{j}\Big)\mathbf{u}^{j} + \mathbf{A}_x^2 \mathbf{A}_y^2 (\bar{\mathbf{F}}^{j+1} + \bar{\mathbf{F}}^{j}).
\end{aligned}
$$

From the boundary conditions, we have $\mathbf{T}_l \mathbf{u} = \bar{\mathbf{F}}_1^{j+1}$, $\mathbf{T}_r \mathbf{u} = \bar{\mathbf{F}}_2^{j+1}$, $\mathbf{T}_b \mathbf{P}^{-1} \mathbf{u} = \bar{\mathbf{G}}_1^{j+1}$ and $\mathbf{T}_u \mathbf{P}^{-1} \mathbf{u} = \bar{\mathbf{G}}_2^{j+1}$. Thus, we can construct the linear system as

$$\begin{bmatrix} \mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau}-\mathbf{Z}^{j+1} & \mathbf{X}\mathbf{\Phi}_y & \mathbf{\Phi}_y & \mathbf{Y}\mathbf{\Phi}_x & \mathbf{\Phi}_x \\ \mathbf{T}_l & 0 & 0 & \cdots & 0 \\ \mathbf{T}_r & 0 & 0 & \cdots & 0 \\ \mathbf{T}_b\mathbf{P}^{-1} & \vdots & \vdots & \ddots & \vdots \\ \mathbf{T}_u\mathbf{P}^{-1} & 0 & 0 & \cdots & 0 \end{bmatrix}\begin{bmatrix} \mathbf{u}^{j+1} \\ \mathbf{f}_0 \\ \mathbf{f}_1 \\ \mathbf{g}_0 \\ \mathbf{g}_1 \end{bmatrix}$$

$$= \begin{bmatrix} \left(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau}+\mathbf{Z}^j\right)\mathbf{u}^j+\mathbf{A}_x^2\mathbf{A}_y^2(\bar{\mathbf{F}}^{j+1}+\bar{\mathbf{F}}^j) \\ \bar{\mathbf{F}}_1^{j+1} \\ \bar{\mathbf{F}}_2^{j+1} \\ \bar{\mathbf{G}}_1^{j+1} \\ \bar{\mathbf{G}}_2^{j+1} \end{bmatrix},$$

where, for $r \in \{j, j+1\}$,

$$\begin{aligned}
\mathbf{u}^r &= [\bar{u}^r(\bar{x}_1,\bar{y}_1),\bar{u}^r(\bar{x}_2,\bar{y}_2),\bar{u}^r(\bar{x}_3,\bar{y}_3),...,\bar{u}^r(\bar{x}_{N_1},\bar{y}_{N_2})]^T, \\
\bar{\mathbf{F}}^{\ r} &= [-4e^{2t_r}((\bar{x}_1+1)^2+(\bar{y}_1+1)^2),-4e^{2t_r}((\bar{x}_2+1)^2+(\bar{y}_2+1)^2), \\
&\quad -4e^{2t_r}((\bar{x}_3+1)^2+(\bar{y}_3+1)^2),...,-4e^{2t_r}((\bar{x}_{N_1}+1)^2+(\bar{y}_{N_2}+1)^2)], \\
\mathbf{Z}^r &= p^2\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_1^r-2\mathbf{A}_x\bar{\boldsymbol{\alpha}}_{1,\bar{x}}^r+\mathbf{A}_x^2\bar{\boldsymbol{\alpha}}_{1,\bar{x}\bar{x}}^r)+q^2\mathbf{A}_x^2(\bar{\boldsymbol{\alpha}}_2^r-2\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{2,\bar{y}}^r+\mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_{2,\bar{y}\bar{y}}^r) \\
&\quad +p\mathbf{A}_x\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_3^r-\mathbf{A}_x\bar{\boldsymbol{\alpha}}_{3,\bar{x}}^r)+q\mathbf{A}_x^2\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_4^r-\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{4,\bar{y}}^r\mathbf{u}^r)+\mathbf{A}_x^2\mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_5^r, \\
\bar{\boldsymbol{\alpha}}_1^r &= \mathrm{diag}((\bar{x}_1+1)^2e^{t_r},(\bar{x}_2+1)^2e^{t_r},(\bar{x}_3+1)^2e^{t_r},...,(\bar{x}_{N_1}+1)^2e^{t_r}), \\
\bar{\boldsymbol{\alpha}}_{1,\bar{x}}^r &= \mathrm{diag}(2(\bar{x}_1+1)e^{t_r},2(\bar{x}_2+1)e^{t_r},2(\bar{x}_3+1)e^{t_r},...,2(\bar{x}_{N_1}+1)e^{t_r}), \\
\bar{\boldsymbol{\alpha}}_{1,\bar{x}\bar{x}}^r &= \mathrm{diag}(2e^{t_r},2e^{t_r},2e^{t_r},...,2e^{t_r}), \\
\bar{\boldsymbol{\alpha}}_2^r &= \mathrm{diag}((\bar{y}_1+1)^2e^{t_r},(\bar{y}_2+1)^2e^{t_r},(\bar{y}_3+1)^2e^{t_r},...,(\bar{y}_{N_2}+1)^2e^{t_r}), \\
\bar{\boldsymbol{\alpha}}_{2,\bar{y}}^r &= \mathrm{diag}(2(\bar{y}_1+1)e^{t_r},2(\bar{y}_2+1)e^{t_r},2(\bar{y}_3+1)e^{t_r},...,2(\bar{y}_{N_2}+1)e^{t_r}), \\
\bar{\boldsymbol{\alpha}}_{2,\bar{y}\bar{y}}^r &= \mathrm{diag}(2e^{t_r},2e^{t_r},2e^{t_r},...,2e^{t_r}), \\
\bar{\boldsymbol{\alpha}}_3^r &= \mathrm{diag}((\bar{x}_1+1)e^{t_r},(\bar{x}_2+1)e^{t_r},(\bar{x}_2+1)e^{t_r},...,(\bar{x}_{N_1}+1)e^{t_r}), \\
\bar{\boldsymbol{\alpha}}_{3,\bar{x}}^r &= \mathrm{diag}(e^{t_r},e^{t_r},e^{t_r},...,e^{t_r}), \\
\bar{\boldsymbol{\alpha}}_4^r &= \mathrm{diag}((\bar{y}_1+1)e^{t_r},(\bar{y}_2+1)e^{t_r},(\bar{y}_2+1)e^{t_r},...,(\bar{y}_{N_2}+1)e^{t_r}),
\end{aligned}$$

$$
\begin{aligned}
\bar{\boldsymbol{\alpha}}_{4,\bar{y}}^r &= \text{diag}(e^{t_r}, e^{t_r}, e^{t_r}, ..., e^{t_r}), \\
\bar{\boldsymbol{\alpha}}_5^r &= \text{diag}(1, 1, 1, ..., 1), \\
\bar{\mathbf{F}}_1^{j+1} &= [e^{t_{j+1}}((\bar{x}_1+1)^2+1), e^{t_{j+1}}((\bar{x}_2+1)^2+1), \\
&\quad e^{t_{j+1}}((\bar{x}_3+1)^2+1), ..., e^{t_{j+1}}((\bar{x}_{N_1}+1)^2+1)]^T, \\
\bar{\mathbf{F}}_2^{j+1} &= [e^{t_{j+1}}((\bar{x}_1+1)^2+5), e^{t_{j+1}}((\bar{x}_2+1)^2+5), \\
&\quad e^{t_{j+1}}((\bar{x}_3+1)^2+5), ..., e^{t_{j+1}}((\bar{x}_{N_1}+5)^2+1)]^T, \\
\bar{\mathbf{G}}_1^{j+1} &= [e^{t_{j+1}}((\bar{y}_1+1)^2+1), e^{t_{j+1}}((\bar{y}_2+1)^2+1), \\
&\quad e^{t_{j+1}}((\bar{y}_3+1)^2+1), ..., e^{t_{j+1}}((\bar{y}_{N_2}+1)^2+1)]^T, \\
\bar{\mathbf{G}}_2^{j+1} &= [e^{t_{j+1}}((\bar{y}_1+1)^2+5), e^{t_{j+1}}((\bar{y}_2+1)^2+5), \\
&\quad e^{t_{j+1}}((\bar{y}_3+1)^2+5), ..., e^{t_{j+1}}((\bar{y}_{N_2}+5)^2+1)]^T.
\end{aligned}
$$

Table 3.12: The AREs for FIM (TPZ) in Example 3.5 when $\tau = 0.1$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	5.7281×10^{-2}	4.4053×10^{-2}	3.7873×10^{-2}	3.4610×10^{-2}
8	4.4053×10^{-2}	2.6256×10^{-2}	1.7627×10^{-2}	1.3041×10^{-2}
10	3.7873×10^{-2}	1.7627×10^{-2}	8.4489×10^{-3}	5.1827×10^{-3}
12	3.4610×10^{-2}	1.3041×10^{-2}	5.1827×10^{-3}	5.7623×10^{-3}

Table 3.13: The AREs for FIM (SIM) in Example 3.5 when $\tau = 0.1$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	2.5104×10^{-2}	1.5501×10^{-2}	1.2211×10^{-2}	1.0854×10^{-2}
8	1.5501×10^{-2}	7.6268×10^{-2}	8.2983×10^{-2}	1.0055×10^{-2}
10	1.2211×10^{-2}	8.2983×10^{-2}	1.3237×10^{-2}	1.6804×10^{-2}
12	1.0854×10^{-2}	1.0055×10^{-2}	1.6804×10^{-2}	2.0794×10^{-2}

Table 3.14: The AREs for FIM (CBS) in Example 3.5 when $\tau = 0.1$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	7.8415×10^{-5}	8.7648×10^{-5}	9.3284×10^{-5}	9.7073×10^{-5}
8	8.7648×10^{-5}	9.8155×10^{-5}	1.0458×10^{-4}	1.0890×10^{-4}
10	9.3284×10^{-5}	1.0458×10^{-4}	1.1148×10^{-4}	1.1613×10^{-4}
12	9.7073×10^{-5}	1.0890×10^{-4}	1.1613×10^{-4}	1.2101×10^{-4}

Table 3.15: The AREs for FIM (CBS) Example 3.5 when $\tau = 0.01$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	7.8566×10^{-7}	8.7823×10^{-7}	9.3476×10^{-7}	9.7271×10^{-7}
8	8.7823×10^{-7}	9.8348×10^{-7}	1.0478×10^{-6}	1.0911×10^{-6}
10	9.3476×10^{-7}	1.0478×10^{-6}	1.1170×10^{-6}	1.1636×10^{-6}
12	9.7271×10^{-7}	1.0911×10^{-6}	1.1636×10^{-6}	1.2122×10^{-6}

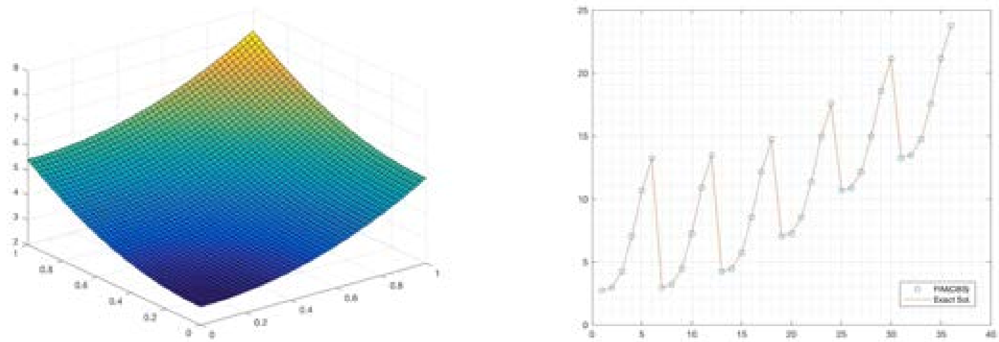


Figure 3.5: The surface and the grid points of the solution in Example 3.5.

Example 3.6. Consider a time-dependent differential equation in which coefficients are functions in terms of both x and t and the forcing term involves trigonometric functions.

$$\begin{aligned}
\frac{\partial u}{\partial t} &= (x^2+t^2)\frac{\partial^2 u}{\partial x^2} + (y^2+t^2)\frac{\partial^2 u}{\partial y^2} + (2x+t)\frac{\partial u}{\partial x} + (2y+t)\frac{\partial u}{\partial y} \\
&\quad + [(x^2+y^2+2t^2)t^2+2t]\sin(x+y+1) - (2x+2y+2t)t^2\cos(x+y+1), \\
&\quad (x,y) \in (0,1)\times(0,1), \quad t \in (0,1), \\
u(x,y,0) &= 0, \quad x \in [0,1]\times[0,1], \\
u(x,0,t) &= t^2\sin(x+1), \quad u(x,1,t) = t^2\sin(x+2), \quad x \in [0,1], \quad t \in [0,1], \\
u(0,y,t) &= t^2\sin(y+1), \quad u(1,y,t) = t^2\sin(y+2), \quad y \in [0,1], \quad t \in [0,1].
\end{aligned}$$

The exact solution for this problem is $u(x,y,t) = t^2\sin(x+y+1)$. We first transform our domain $\Omega = [0,1]\times[0,1]$ by using the transformations $\bar{x} = 2x-1$ and $\bar{y} = 2y-1$. By our numerical algorithm, this problem can be written in matrix form as

$$\begin{aligned}
&\Big(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} - \mathbf{Z}^{j+1}\Big)\mathbf{u}^{j+1} + \mathbf{X}\mathbf{\Phi}_y\mathbf{f}_0 + \mathbf{\Phi}_y\mathbf{f}_1 + \mathbf{Y}\mathbf{\Phi}_x\mathbf{g}_0 + \mathbf{\Phi}_x\mathbf{g}_1 \\
&= \Big(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} + \mathbf{Z}^{j}\Big)\mathbf{u}^{j} + \mathbf{A}_x^2\mathbf{A}_y^2(\bar{\mathbf{F}}^{j+1} + \bar{\mathbf{F}}^{j}).
\end{aligned}$$

From the boundary conditions, we have $\mathbf{T}_l\mathbf{u} = \bar{\mathbf{F}}_1^{j+1}$, $\mathbf{T}_r\mathbf{u} = \bar{\mathbf{F}}_2^{j+1}$, $\mathbf{T}_b\mathbf{P}^{-1}\mathbf{u} = \bar{\mathbf{G}}_1^{j+1}$ and $\mathbf{T}_u\mathbf{P}^{-1}\mathbf{u} = \bar{\mathbf{G}}_2^{j+1}$. Thus, we can construct the linear system as

$$\begin{bmatrix} \mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} - \mathbf{Z}^{j+1} & \mathbf{X}\mathbf{\Phi}_y & \mathbf{\Phi}_y & \mathbf{Y}\mathbf{\Phi}_x & \mathbf{\Phi}_x \\ \mathbf{T}_l & 0 & 0 & \cdots & 0 \\ \mathbf{T}_r & 0 & 0 & \cdots & 0 \\ \mathbf{T}_b\mathbf{P}^{-1} & \vdots & \vdots & \ddots & \vdots \\ \mathbf{T}_u\mathbf{P}^{-1} & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{j+1} \\ \mathbf{f}_0 \\ \mathbf{f}_1 \\ \mathbf{g}_0 \\ \mathbf{g}_1 \end{bmatrix}$$

$$= \begin{bmatrix} \left(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} + \mathbf{Z}^j\right)\mathbf{u}^j + \mathbf{A}_x^2\mathbf{A}_y^2(\bar{\mathbf{F}}^{j+1} + \bar{\mathbf{F}}^j) \\ \bar{\mathbf{F}}_1^{j+1} \\ \bar{\mathbf{F}}_2^{j+1} \\ \bar{\mathbf{G}}_1^{j+1} \\ \bar{\mathbf{G}}_2^{j+1} \end{bmatrix},$$

where, for $r \in \{j, j+1\}$,

$$\begin{aligned}
\mathbf{u}^r &= [\bar{u}^r(\bar{x}_1,\bar{y}_1), \bar{u}^r(\bar{x}_2,\bar{y}_2), \bar{u}^r(\bar{x}_3,\bar{y}_3), ..., \bar{u}^r(\bar{x}_{N_1},\bar{y}_{N_2})]^T, \\
\bar{\mathbf{F}}^{\,r} &= \Bigg[\bigg\{\bigg(\frac{(\bar{x}_1+1)^2}{4} + \frac{(\bar{y}_1+1)^2}{4} + 2t_r^2\bigg)t_r^2 + 2t_r\bigg\}\sin\left(\frac{\bar{x}_1+1}{2} + \frac{\bar{y}_1+1}{2} + 1\right) \\
&\quad - (\bar{x}_1 + \bar{y}_1 + 2 + 2t_r)t_r^2\cos\left(\frac{\bar{x}_1+1}{2} + \frac{\bar{y}_1+1}{2} + 1\right), \\
&\quad \bigg\{\bigg(\frac{(\bar{x}_2+1)^2}{4} + \frac{(\bar{y}_2+1)^2}{4} + 2t_r^2\bigg)t_r^2 + 2t_r\bigg\}\sin\left(\frac{\bar{x}_2+1}{2} + \frac{\bar{y}_2+1}{2} + 1\right) \\
&\quad - (\bar{x}_2 + \bar{y}_2 + 2 + 2t_r)t_r^2\cos\left(\frac{\bar{x}_2+1}{2} + \frac{\bar{y}_2+1}{2} + 1\right), ..., \\
&\quad \bigg\{\bigg(\frac{(\bar{x}_{N_1}+1)^2}{4} + \frac{(\bar{y}_{N_2}+1)^2}{4} + 2t_r^2\bigg)t_r^2 + 2t_r\bigg\}\sin\left(\frac{\bar{x}_{N_1}+1}{2} + \frac{\bar{y}_{N_2}+1}{2} + 1\right) \\
&\quad - (\bar{x}_{N_1} + \bar{y}_{N_2} + 2 + 2t_r)t_r^2\cos\left(\frac{\bar{x}_{N_1}+1}{2} + \frac{\bar{y}_{N_2}+1}{2} + 1\right)\Bigg], \\
\mathbf{Z}^r &= p^2\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_1^r - 2\mathbf{A}_x\bar{\boldsymbol{\alpha}}_{1,\bar{x}}^r + \mathbf{A}_x^2\bar{\boldsymbol{\alpha}}_{1,\bar{x}\bar{x}}^r) + q^2\mathbf{A}_x^2(\bar{\boldsymbol{\alpha}}_2^r - 2\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{2,\bar{y}}^r + \mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_{2,\bar{y}\bar{y}}^r) \\
&\quad + p\mathbf{A}_x\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_3^r - \mathbf{A}_x\bar{\boldsymbol{\alpha}}_{3,\bar{x}}^r) + q\mathbf{A}_x^2\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_4^r - \mathbf{A}_y\bar{\boldsymbol{\alpha}}_{4,\bar{y}}^r\mathbf{u}^r), \\
\bar{\boldsymbol{\alpha}}_1^r &= \mathrm{diag}\bigg(\frac{(\bar{x}_1+1)^2}{4} + t_r^2, \frac{(\bar{x}_2+1)^2}{4} + t_r^2 + \frac{(\bar{x}_3+1)^2}{4} + t_r^2, ..., \frac{(\bar{x}_{N_1}+1)^2}{4} + t_r^2\bigg), \\
\bar{\boldsymbol{\alpha}}_{1,\bar{x}}^r &= \mathrm{diag}\bigg(\frac{\bar{x}_1+1}{2}, \frac{\bar{x}_2+1}{2}, \frac{\bar{x}_3+1}{2}, ..., \frac{\bar{x}_{N_1}+1}{2}\bigg), \\
\bar{\boldsymbol{\alpha}}_{1,\bar{x}\bar{x}}^r &= \mathrm{diag}\bigg(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}\bigg),
\end{aligned}$$

$$
\begin{aligned}
\bar{\boldsymbol{\alpha}}_2^r &= \operatorname{diag}\Big(\frac{(\bar{y}_1+1)^2}{4}+t_r^2, \frac{(\bar{y}_2+1)^2}{4}+t_r^2+\frac{(\bar{y}_3+1)^2}{4}+t_r^2, ..., \frac{(\bar{y}_{N_2}+1)^2}{4}+t_r^2\Big),\\
\bar{\boldsymbol{\alpha}}_{2,\bar{y}}^r &= \operatorname{diag}\Big(\frac{\bar{y}_1+1}{2}, \frac{\bar{y}_2+1}{2}, \frac{\bar{y}_3+1}{2}, ..., \frac{\bar{y}_{N_2}+1}{2}\Big),\\
\bar{\boldsymbol{\alpha}}_{2,\bar{y}\bar{y}}^r &= \operatorname{diag}\Big(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}\Big),\\
\bar{\boldsymbol{\alpha}}_3^r &= \operatorname{diag}(\bar{x}_1+1+t_r, \bar{x}_2+1+t_r, \bar{x}_3+1+t_r, ..., \bar{x}_{N_1}+1+t_r),\\
\bar{\boldsymbol{\alpha}}_{3,\bar{y}}^r &= \operatorname{diag}(1,1,1,...,1),\\
\bar{\boldsymbol{\alpha}}_4^r &= \operatorname{diag}(\bar{y}_1+1+t_r, \bar{y}_2+1+t_r, \bar{y}_3+1+t_r, ..., \bar{y}_{N_2}+1+t_r),\\
\bar{\boldsymbol{\alpha}}_{4,\bar{y}}^r &= \operatorname{diag}(1,1,1,...,1),\\
\bar{\mathbf{F}}_1^{j+1} &= [t_{j+1}^2 \sin\Big(\frac{\bar{x}_1+3}{2}\Big), t_{j+1}^2 \sin\Big(\frac{\bar{x}_2+3}{2}\Big), t_{j+1}^2 \sin\Big(\frac{\bar{x}_3+3}{2}\Big), ..., t_{j+1}^2 \sin\Big(\frac{\bar{x}_{N_1}+3}{2}\Big)]^T,\\
\bar{\mathbf{F}}_2^{j+1} &= [t_{j+1}^2 \sin\Big(\frac{\bar{x}_1+5}{2}\Big), t_{j+1}^2 \sin\Big(\frac{\bar{x}_2+5}{2}\Big), t_{j+1}^2 \sin\Big(\frac{\bar{x}_3+5}{2}\Big), ..., t_{j+1}^2 \sin\Big(\frac{\bar{x}_{N_1}+5}{2}\Big)]^T,\\
\bar{\mathbf{G}}_1^{j+1} &= [t_{j+1}^2 \sin\Big(\frac{\bar{y}_1+3}{2}\Big), t_{j+1}^2 \sin\Big(\frac{\bar{y}_2+3}{2}\Big), t_{j+1}^2 \sin\Big(\frac{\bar{y}_3+3}{2}\Big), ..., t_{j+1}^2 \sin\Big(\frac{\bar{y}_{N_2}+3}{2}\Big)]^T,\\
\bar{\mathbf{G}}_2^{j+1} &= t_{j+1}^2 \sin\Big(\frac{\bar{y}_1+5}{2}\Big), t_{j+1}^2 \sin\Big(\frac{\bar{y}_2+5}{2}\Big), t_{j+1}^2 \sin\Big(\frac{\bar{y}_3+5}{2}\Big), ..., t_{j+1}^2 \sin\Big(\frac{\bar{y}_{N_2}+5}{2}\Big)]^T.
\end{aligned}
$$

Table 3.16: The AREs for FIM (TPZ) in Example 3.6 when $\tau = 0.1$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	1.1343×10^{-2}	1.1629×10^{-2}	1.1844×10^{-2}	1.2003×10^{-2}
8	1.1629×10^{-2}	1.1191×10^{-2}	1.2134×10^{-2}	1.2296×10^{-2}
10	1.1844×10^{-2}	1.2134×10^{-2}	1.2355×10^{-2}	1.2518×10^{-2}
12	1.2003×10^{-2}	1.2296×10^{-2}	1.2518×10^{-2}	1.2684×10^{-2}

Table 3.17: The AREs for FIM (SIM) in Example 3.6 when $\tau = 0.1$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	9.6989×10^{-3}	1.0244×10^{-2}	1.0603×10^{-2}	1.0846×10^{-2}
8	1.0244×10^{-2}	1.0837×10^{-2}	1.1121×10^{-2}	1.1145×10^{-2}
10	1.0603×10^{-2}	1.1121×10^{-2}	1.1652×10^{-2}	1.1915×10^{-2}
12	1.0846×10^{-2}	1.1145×10^{-2}	1.1915×10^{-2}	1.3024×10^{-2}

Table 3.18: The AREs for FIM (CBS) in Example 3.6 when $\tau = 0.1$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	1.3795×10^{-6}	9.0151×10^{-7}	9.0667×10^{-7}	9.0734×10^{-7}
8	9.0151×10^{-7}	2.1107×10^{-9}	1.3195×10^{-9}	1.3279×10^{-9}
10	9.0667×10^{-7}	1.3195×10^{-9}	1.9190×10^{-12}	1.1931×10^{-12}
12	9.0734×10^{-7}	1.3279×10^{-9}	1.1931×10^{-12}	2.7685×10^{-12}

Table 3.19: The AREs for FIM (CBS) in Example 3.6 when $\tau = 0.01$

N_1	$N_2 = 6$	$N_2 = 8$	$N_2 = 10$	$N_2 = 12$
6	1.3795×10^{-6}	9.0152×10^{-7}	9.0668×10^{-7}	9.0735×10^{-7}
8	9.0152×10^{-7}	2.1092×10^{-9}	1.3199×10^{-9}	1.3274×10^{-9}
10	9.0668×10^{-7}	1.3199×10^{-9}	8.6832×10^{-12}	1.0379×10^{-11}
12	9.0735×10^{-7}	1.3274×10^{-9}	1.0379×10^{-11}	6.0370×10^{-12}

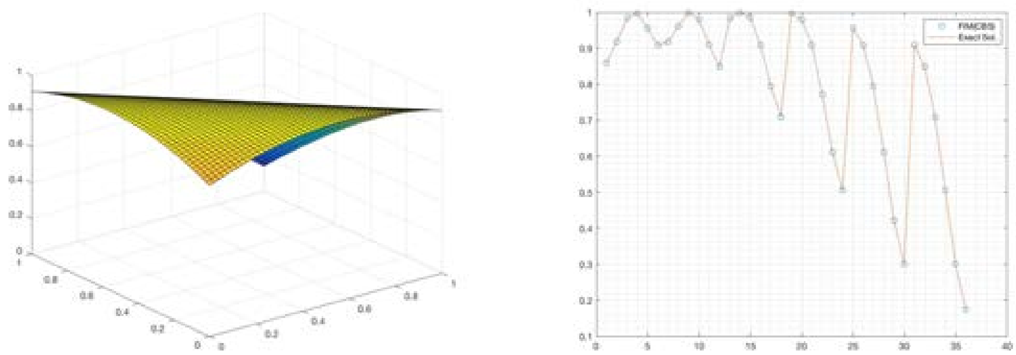


Figure 3.6: The surface and the grid points of the solution in Example 3.6 at $t = 1$.

From Examples 3.1-3.6, we can see that our proposed algorithm gives a lot better accurate results comparing to the other traditional FIMs when using the same number of nodes. In addition, if we increase the time step, then we get better approximate solutions.

CHAPTER IV

NUMERICAL PROCEDURE FOR SOLVING LINEAR FRACTIONAL DIFFERENTIAL EQUATIONS

In this chapter, we construct an algorithm based on FIM using shifted Chebyshev polynomials for finding the approximate solutions of linear FDEs using Riemann-Liouville definition of fractional derivative. To simplify our construction, let us consider the following the linear FDE.

$$D^{\alpha}u(x) + a_2(x)u''(x) + a_1(x)u'(x) + a_0(x)u(x) = f(x), \quad x \in (0, b), \tag{4.1}$$

where $m \in \{1, 2\}$ and $\alpha \in (m-1, m)$ with boundary conditions $u(0) = 0$ and $u(b) = b_r$. Throughout this section, let us assume that the solution of (4.1) exists and unique. In addition, to apply FIM, we have to assume further that $lim_{x\to 0^+} u'(x)(-x)^{m-\alpha}$ exists.

First of all, for $m \in \{1, 2\}$, let $\beta = \alpha - m + 1$, $x \in (0, 1)$, $F(x) = \int_0^x \frac{u(s)}{(x-s)^{\beta}} ds$ and $\omega \in [0, 1]$. We approximate $F(x)$ at each computational node $x_k \in [0, 1]$ for $k \in \{1, 2, 3, ..., M\}$, where $x_0 = 0$ as follow.

$$\begin{aligned}
F(x_k) &= \int_0^{x_k} (x_k - s)^{-\beta} u(s) ds \\
&= \sum_{i=0}^{k-1} \int_{x_i}^{x_{i+1}} (x_k - s)^{-\beta} u(s) ds \\
&\approx \sum_{i=0}^{k-1} \int_{x_i}^{x_{i+1}} (x_k - s)^{-\beta} (\omega u(x_i) + (1-\omega) u(x_{i+1})) ds \\
&= \sum_{i=0}^{k-1} (\omega u(x_i) + (1-\omega) u(x_{i+1})) \int_{x_i}^{x_{i+1}} (x_k - s)^{-\beta} ds \\
&= \sum_{i=0}^{k-1} (\omega u(x_i) + (1-\omega) u(x_{i+1})) \Big(\frac{(x_k - x_i)^{1-\beta}}{1-\beta} - \frac{(x_k - x_{i+1})^{1-\beta}}{1-\beta} \Big).
\end{aligned}$$

By letting $q_{i,i+1}(x_k) = (x_k - x_i)^{1-\beta} - (x_k - x_{i+1})^{1-\beta}$, we have from the fact of $1-\beta = m-\alpha$ that

$$F(x_k) = \frac{1}{m-\alpha}\sum_{i=0}^{k-1}(\omega u(x_i) + (1-\omega)u(x_{i+1}))q_{i,i+1}(x_k). \tag{4.2}$$

Now, we are ready to apply the FIM using Chebyshev polynomials to devise an algorithm of calculating the approximate solution of (4.1).

4.1 Algorithm For Solving Linear Fractional Differential Equations

Step 1. Transform $x \in [0, b]$ into $\bar{x} \in [0, 1]$ by using the transformation $\bar{x} = \frac{x}{b}$ and $\rho = \frac{s}{b}$. Then, (4.1) becomes

$$\frac{p^{\alpha}}{\Gamma(m-\alpha)}\frac{d^m}{d\bar{x}^m}\int_0^{\bar{x}}\frac{\bar{u}(\rho)}{(\bar{x}-\rho)^{\beta}}d\rho + p^2\bar{a}_2(\bar{x})\bar{u}''(\bar{x}) + p\bar{a}_1(\bar{x})\bar{u}'(\bar{x}) + \bar{a}_0(\bar{x})\bar{u}(\bar{x}) = \bar{f}(\bar{x}), \tag{4.3}$$

where $p = \frac{1}{b}$, $\bar{u}^{(j)}(\bar{x}) = \frac{d^{(j)}\bar{u}}{d\bar{x}^{(j)}}$ for $j \in \{1, 2\}$, $\bar{f}(\bar{x}) = f(b\bar{x})$, $\bar{u}(\bar{x}) = u(b\bar{x})$ and $\bar{a}_i(\bar{x}) = a_i(b\bar{x})$ for $i \in \{0, 1, 2\}$. By our assumptions, $\bar{u}(0) = 0$.

Step 2. Discretize $[0, 1]$ into M nodes by using zeros of the shifted Chebyshev polynomial $T^*_M(\bar{x})$ as defined in (2.10), i.e.,

$$\bar{x}_k = \frac{1}{2}\Big(\cos\Big(\Big(\frac{2k-1}{2M}\Big)\pi\Big) + 1\Big),$$

where $k = \{1, 2, 3, ..., M\}$.

Step 3. Eliminate all derivatives with respect to $\bar{x}$ by taking the double layer integration from 0 to ξ_2 and from 0 to $\bar{x}_k$, respectively and using integration by parts with integer order terms. Then, (4.3) becomes

$$\frac{p^\alpha}{\Gamma(m-\alpha)}\int_0^{\bar{x}_k}\int_0^{\xi 2}\frac{d^m}{d\xi_1^m}\int_0^{\xi_1}\frac{\bar{u}(\rho)}{(\xi_1-\rho)^\beta}d\rho d\xi_1 d\xi_2+p^2\bar{a}_2(\bar{x}_k)\bar{u}(\bar{x}_k)$$
$$-2p^2\int_0^{\bar{x}_k}\bar{a}_{2,\bar{x}}(\xi_2)\bar{u}(\xi_2)d\xi_2+p^2\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{a}_{2,\bar{x}\bar{x}}(\xi_1)\bar{u}(\xi_1)d\xi_1 d\xi_2+p\int_0^{\bar{x}_k}\bar{a}_1(\xi_2)\bar{u}(\xi_2)d\xi_2$$
$$-p\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{a}_{1,\bar{x}}(\xi_1)\bar{u}(\xi_1)d\xi_1 d\xi_2+\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{a}_0(\xi_2)\bar{u}(\xi_2)d\xi_1 d\xi_2$$
$$=\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{f}(\xi_2)d\xi_1 d\xi_2+c_0+c_1\bar{x}_k. \tag{4.4}$$

where $c_0=-a_2(0)\bar{u}(0),\quad c_1=-a_2(0)\bar{u}'(0)+a_{2,\bar{x}}(0)\bar{u}(0)-a_1(0)\bar{u}(0).$

Case 1 $m=1$. We can get rid of the first derivative of the first term of (4.4) by integrating once with respect to ξ_1 from 0 to ξ_2. Then by using assumption that $lim_{x\to 0^+}u'(x)(-x)^{m-\alpha}$ exists then we have $lim_{\rho\to 0^+}\bar{u}'(\rho)(-\rho)^{1-\beta}$ exists and by (4.2), (4.4) becomes

$$\frac{p^\alpha}{\Gamma(m-\alpha)}\int_0^{\bar{x}_k}\frac{1}{m-\alpha}\sum_{i=0}^{k-1}(\omega\bar{u}(\bar{x}_i)+(1-\omega)\bar{u}(\bar{x}_{i+1}))q_{i,i+1}(\xi_2)d\xi_2$$
$$+p^2\bar{a}_2(\bar{x}_k)\bar{u}(\bar{x}_k)-2p^2\int_0^{\bar{x}_k}\bar{a}_{2,\bar{x}}(\xi_2)\bar{u}(\xi_2)d\xi_2+p^2\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{a}_{2,\bar{x}\bar{x}}(\xi_1)\bar{u}(\xi_1)d\xi_1 d\xi_2$$
$$+p\int_0^{\bar{x}_k}\bar{a}_1(\xi_2)\bar{u}(\xi_2)d\xi_2-p\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{a}_{1,\bar{x}}(\xi_1)\bar{u}(\xi_1)d\xi_1 d\xi_2+\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{a}_0(\xi_2)\bar{u}(\xi_2)d\xi_1 d\xi_2$$
$$=\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{f}(\xi_2)d\xi_1 d\xi_2+c_0+c_1\bar{x}_k, \tag{4.5}$$

Case 2 $m=2$. We can get rid of the second derivative of the first term of (4.4) by integrating twice with respect to ξ_1 and ξ_2 from 0 to ξ_2 and 0 to $\bar{x}_k$, respectively. Then by using (4.2), (4.4) becomes

$$\frac{p^\alpha}{\Gamma(m-\alpha)}\frac{1}{m-\alpha}\sum_{i=0}^{k-1}(\omega\bar{u}(\bar{x}_i)+(1-\omega)\bar{u}(\bar{x}_{i+1}))q_{i,i+1}(\bar{x}_k)$$
$$+p^2\bar{a}_2(\bar{x}_k)\bar{u}(\bar{x}_k)-2p^2\int_0^{\bar{x}_k}\bar{a}_{2,\bar{x}}(\xi_2)\bar{u}(\xi_2)d\xi_2+p^2\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{a}_{2,\bar{x}\bar{x}}(\xi_1)\bar{u}(\xi_1)d\xi_1 d\xi_2$$
$$+p\int_0^{\bar{x}_k}\bar{a}_1(\xi_2)\bar{u}(\xi_2)d\xi_2-p\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{a}_{1,\bar{x}}(\xi_1)\bar{u}(\xi_1)d\xi_1 d\xi_2+\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{a}_0(\xi_2)\bar{u}(\xi_2)d\xi_1 d\xi_2$$
$$=\int_0^{\bar{x}_k}\int_0^{\xi_2}\bar{f}(\xi_2)d\xi_1 d\xi_2+c_0+c_1\bar{x}_k, \tag{4.6}$$

where c_0 and c_1 are arbitrary constants.

Step 4. Transform (4.5) or (4.6) into the matrix form.

First, we transform the term

$$\sum_{i=0}^{k-1}(\omega\bar{u}(\bar{x}_i)+(1-\omega)\bar{u}(\bar{x}_{i+1}))q_{i,i+1}(\bar{x}_k),$$

for $k\in\{1,2,3,...,M\}$, into $\mathbf{Qu}$, where $\mathbf{Q}=[Q_{ij}]_{M\times M}$, with

$$Q_{ij}=\begin{cases}(1-\omega)q_{i-1,i}(\bar{x}_i) & \text{if } i=j,\\ (1-\omega)q_{i-1,j}(\bar{x}_i)+\omega q_{j,i}(\bar{x}_i) & \text{if } i>j,\\ 0 & \text{if } i<j,\end{cases} \tag{4.7}$$

$$\text{and } \mathbf{u}=\begin{bmatrix}\bar{u}(\bar{x}_1) & \bar{u}(\bar{x}_2) & \bar{u}(\bar{x}_3) & \dots & \bar{u}(\bar{x}_M)\end{bmatrix}^T.$$

By the first and the second order integration matrices using the shifted Chebyshev polynomial which is described in Section 2.6 and equation (4.7), (4.5) or (4.6) can be written in matrix form as

$$\frac{p^\alpha}{\Gamma(m+1-\alpha)}(\mathbf{A}^*)^{2-m}\mathbf{Qu}+p^2\mathbf{B}_2\mathbf{u}-2p^2\mathbf{A}^*\mathbf{B}_{2,\bar{x}}\mathbf{u}+p^2(\mathbf{A}^*)^2\mathbf{B}_{2,\bar{x}\bar{x}}\mathbf{u}$$
$$+p\mathbf{A}^*\mathbf{B}_1\mathbf{u}-p(\mathbf{A}^*)^2\mathbf{B}_{1,\bar{x}}\mathbf{u}+(\mathbf{A}^*)^2\mathbf{B}_0\mathbf{u}+c_0\mathbf{E}+c_1\bar{\mathbf{x}}=(\mathbf{A}^*)^2\mathbf{F}.$$

That is,

$$\mathbf{Ku}+c_0\mathbf{E}+c_1\bar{\mathbf{x}}=(\mathbf{A}^*)^2\mathbf{F}, \tag{4.8}$$

where, for $m\in\{1,2\},\alpha\in(m-1,m)$ and $\beta=\alpha-m+1$

$$\begin{aligned}
\mathbf{F} &= [\bar{f}(\bar{x}_1),\bar{f}(\bar{x}_2),\bar{f}(\bar{x}_3),...,\bar{f}(\bar{x}_M)]^T,\\
\mathbf{K} &= \frac{p^\alpha}{\Gamma(m+1-\alpha)}(\mathbf{A}^*)^{2-m}\mathbf{Q}+p^2\mathbf{B}_2-2p^2\mathbf{A}^*\mathbf{B}_{2,\bar{x}}+p^2(\mathbf{A}^*)^2\mathbf{B}_{2,\bar{x}\bar{x}}\\
&\quad +p\mathbf{A}^*\mathbf{B}_1-p(\mathbf{A}^*)^2\mathbf{B}_{1,\bar{x}}+(\mathbf{A}^*)^2\mathbf{B}_0,\\
\mathbf{B}_i &= \text{diag}(\bar{a}_i(\bar{x}_1),\bar{a}_i(\bar{x}_2),\bar{a}_i(\bar{x}_3),...,\bar{a}_i(\bar{x}_M)) \text{ for } \ i\in\{0,1,2\},\\
\mathbf{B}_{i,\bar{x}} &= \text{diag}(\bar{a}_{i,\bar{x}}(\bar{x}_1),\bar{a}_{i,\bar{x}}(\bar{x}_2),\bar{a}_{i,\bar{x}}(\bar{x}_3),...,\bar{a}_{i,\bar{x}}(\bar{x}_M)) \text{ for } \ i\in\{1,2\},\\
\mathbf{B}_{2,\bar{x}\bar{x}} &= \text{diag}(\bar{a}_{2,\bar{x}\bar{x}}(\bar{x}_1),\bar{a}_{2,\bar{x}\bar{x}}(\bar{x}_2),\bar{a}_{2,\bar{x}\bar{x}}(\bar{x}_3),...,\bar{a}_{2,\bar{x}\bar{x}}(\bar{x}_M)),\\
\mathbf{E} &= [1,1,1,..,1]^T \text{and } \bar{\mathbf{x}}=[\bar{x}_1,\bar{x}_2,\bar{x}_3,...,\bar{x}_M]^T.
\end{aligned}$$

Step 5. Consider the boundary conditions which are $\bar{u}(0) = 0$ and $\bar{u}(1) = b_r$. Then, we have

$$\bar{u}(0) = \sum_{n=0}^{M-1} c_n T_n^*(0) = \mathbf{t}_l \mathbf{c} = \mathbf{t}_l (\mathbf{T}^*)^{-1} \mathbf{u}, \tag{4.9}$$

$$\bar{u}(1) = \sum_{n=0}^{M-1} c_n T_n^*(1) = \mathbf{t}_r \mathbf{c} = \mathbf{t}_r (\mathbf{T}^*)^{-1} \mathbf{u}, \tag{4.10}$$

where $\mathbf{t}_l = [1, -1, 1, ..., (-1)^{M-1}]$ and $\mathbf{t}_r = [1, 1, 1, ..., 1]$.

Step 6. Construct the linear system (4.11) from (4.8), (4.9) and (4.10).

$$\begin{bmatrix} \mathbf{K} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l(\mathbf{T}^*)^{-1} & 0 & 0 \\ \mathbf{t}_r(\mathbf{T}^*)^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\mathbf{A}^*)^2\mathbf{F} \\ 0 \\ b_r \end{bmatrix}. \tag{4.11}$$

Then, the approximate solution can be found by solving the linear system (4.11). To obtain the approximate solution $u(x)$ for $x \in [0, b]$, we use transformation $x = b\bar{x}$.

4.2 Numerical Examples

In this section, we use our proposed method to find the approximate solutions of some linear FDEs. In each example, we use different error based on the results from each corresponding paper that we would like to compare with.

Example 4.1**.** Consider a linear FDE with $\alpha \in (1, 2)$.

$$D^\alpha(u) + \frac{d^2u}{dx^2} + u = \frac{2x^{2-\alpha}}{\Gamma(3-\alpha)} + x^2 + 2, \quad x \in (0, 1), \tag{4.12}$$

with boundary conditions $u(0) = 0$ and $u(1) = 1$. The exact solution is $u^*(x) = x^2$. By using our numerical algorithm, this problem can be written in the matrix form as

$$\mathbf{K}\mathbf{u} + c_0\mathbf{E} + c_1\bar{\mathbf{x}} = (\mathbf{A}^*)^2\mathbf{F},$$

where

$$\begin{aligned}
\mathbf{F} &= \Big[\frac{2x_1^{2-\alpha}}{\Gamma(3-\alpha)} + x_1^2 + 2, \frac{2x_2^{2-\alpha}}{\Gamma(3-\alpha)} + x_2^2 + 2, \\
&\quad \frac{2x_3^{2-\alpha}}{\Gamma(3-\alpha)} + x_3^2 + 2, ..., \frac{2x_M^{2-\alpha}}{\Gamma(3-\alpha)} + x_M^2 + 2\Big]^T, \\
\mathbf{K} &= \frac{1}{\Gamma(3-\alpha)}\mathbf{Q} + \mathbf{I} + (\mathbf{A}^*)^2.
\end{aligned}$$

For the boundary conditions, we have $\mathbf{t}_l(\mathbf{T}^*)^{-1}\mathbf{u} = 0$ and $\mathbf{t}_r(\mathbf{T}^*)^{-1}\mathbf{u} = 1$. Therefore, we solve the following linear system

$$\begin{bmatrix} \mathbf{K} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l(\mathbf{T}^*)^{-1} & 0 & 0 \\ \mathbf{t}_r(\mathbf{T}^*)^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\mathbf{A}^*)^2\mathbf{F} \\ 0 \\ 1 \end{bmatrix}. \tag{4.13}$$

Finally, we obtain the approximate solutions $u(x)$. In this example according to Wen et al. [12], they used the FIM with trapezoidal rule and the error defined by

$$E = \frac{1}{M}\sum_{k=1}^{M} \left| \frac{u_k(x) - u_k^*(x)}{u_{max}^*(x)} \right|,$$

where u^* and u are the exact and numerical solutions, respectively. Table 4.1 shows the error E for our FIM using Chebyshev polynomials (CBS) when $M = 10$ with several values of α. For each values of α, we give the optimal ω that achieve the highest accuracy. Then, the error E for our FIM using Chebyshev polynomials (CBS) compared with Wen et al.'s FIMs results [12] are shown in Tables 4.2 and 4.3 when the number of nodes are equal. Also, the graph of the exact and approximate solutions are shown in Figure 4.1. Note that ω in Tables 4.2 - 4.3 are the optimal ω that give the best accuracy for each α.

Table 4.1**:** The errors E for FIM (CBS) in Example 4.1 when $M = 10$

α	ω	E
1.001	0.5	1.5150×10^{-4}
1.1	0.51	1.2287×10^{-4}
1.3	0.52	1.3868×10^{-3}
1.5	0.25	1.8318×10^{-3}

α	ω	E
1.7	0.14	1.5272×10^{-3}
1.9	0.05	6.5496×10^{-4}
1.99	0	1.4920×10^{-4}
1.999	0	2.2748×10^{-4}

Table 4.2**:** The errors E for FIM (CBS) and FIM (TPZ) in Example 4.1 when $M = 10$

α	ω	FIM (CBS)	FIM (TPZ)
1.001	0.5	1.5150×10^{-4}	2.5492×10^{-6}
1.1	0.51	1.2287×10^{-4}	3.1509×10^{-4}
1.5	0.25	1.8318×10^{-3}	3.6165×10^{-3}
1.9	0.05	6.5496×10^{-4}	1.3637×10^{-2}
1.999	0	2.2748×10^{-4}	1.7736×10^{-2}

Table 4.3**:** The errors E for FIM (CBS) and FIM (TPZ) in Example 4.1 when $M = 20$

α	ω	FIM (CBS)	FIM (TPZ)
1.001	0.5	1.5070×10^{-5}	8.4525×10^{-7}
1.1	0.51	5.4347×10^{-5}	1.0870×10^{-4}
1.5	0.29	8.1860×10^{-4}	1.5302×10^{-3}
1.9	0.06	3.5514×10^{-4}	7.4245×10^{-3}
1.999	0	1.2338×10^{-5}	1.0311×10^{-2}

Example 4.2**.** Consider a linear FDE with $\alpha \in (1, 2)$.

$$D^{\alpha}(u) + \frac{d^2u}{dx^2} + 2\frac{du}{dx} - u = \frac{6x^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{2x^{2-\alpha}}{\Gamma(3-\alpha)} - x^3 + 5x^2 + 10x + 2, \quad x \in (0, 1), \tag{4.14}$$

with boundary conditions $u(0) = 0$ and $u(1) = 2$. The exact solution is $u^*(x) = x^3 + x^2$. By using our numerical algorithm, this problem can be written in the matrix form as

$$\mathbf{K}\mathbf{u} + c_0\mathbf{E} + c_1\bar{\mathbf{x}} = (\mathbf{A}^*)^2\mathbf{F},$$

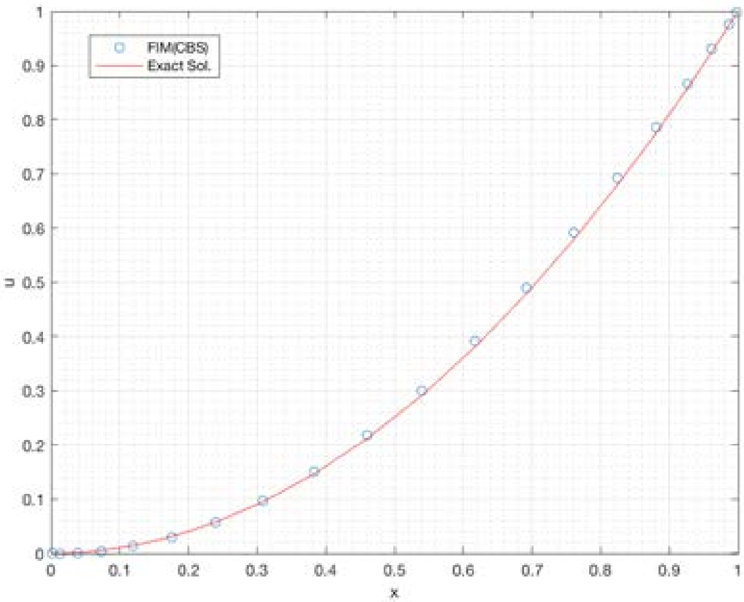


Figure 4.1: The graph of exact and approximate solutions of Example 4.1 with $\alpha = 1.5$.

where

$$
\begin{aligned}
\mathbf{F} &= \Big[\frac{6x_1^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{2x_1^{2-\alpha}}{\Gamma(3-\alpha)} - x_1^3 + 5x_1^2 + 10x_1 + 2, \\
&\quad \frac{6x_2^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{2x_2^{2-\alpha}}{\Gamma(3-\alpha)} - x_2^3 + 5x_2^2 + 10x_2 + 2, \\
&\quad \frac{6x_3^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{2x_3^{2-\alpha}}{\Gamma(3-\alpha)} - x_3^3 + 5x_3^2 + 10x_3 + 2, \\
&\quad \ldots, \frac{6x_M^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{2x_M^{2-\alpha}}{\Gamma(3-\alpha)} - x_M^3 + 5x_M^2 + 10x_M + 2\Big]^T, \\
\mathbf{K} &= \frac{1}{\Gamma(2-\alpha)}(\mathbf{A}^*)\mathbf{Q} + \mathbf{I} + 2(\mathbf{A}^*) - (\mathbf{A}^*)^2.
\end{aligned}
$$

For the boundary conditions, we have $\mathbf{t}_l(\mathbf{T}^*)^{-1}\mathbf{u} = 0$ and $\mathbf{t}_r(\mathbf{T}^*)^{-1}\mathbf{u} = 2$. Therefore, we solve the following linear system

$$
\begin{bmatrix} \mathbf{K} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l(\mathbf{T}^*)^{-1} & 0 & 0 \\ \mathbf{t}_r(\mathbf{T}^*)^{-1} & 0 & 0 \end{bmatrix}
\begin{bmatrix} \mathbf{u} \\ c_0 \\ c_1 \end{bmatrix}
=
\begin{bmatrix} (\mathbf{A}^*)^2\mathbf{F} \\ 0 \\ 2 \end{bmatrix}. \tag{4.15}
$$

The error E are defined as the one in Example 4.1. Table 4.4 shows the error E for our

FIM using Chebyshev polynomials (CBS) when $M = 10$ with different α and ω. Also, the graph of exact and approximate solutions are shown in Figure 4.2. Note that ω in Table 4.4 are the optimal ω that give the best accuracy.

Table 4.4: The errors E for FIM (CBS) in Example 4.2 when $M = 10$

α	ω	E
1.001	0.51	1.5010×10^{-4}
1.1	0.52	9.5674×10^{-5}
1.3	0.54	1.2643×10^{-3}
1.5	0.25	2.1052×10^{-3}

α	ω	E
1.7	0.14	1.9663×10^{-3}
1.9	0.04	9.3191×10^{-3}
1.99	0	1.4630×10^{-4}
1.999	0	1.4758×10^{-5}

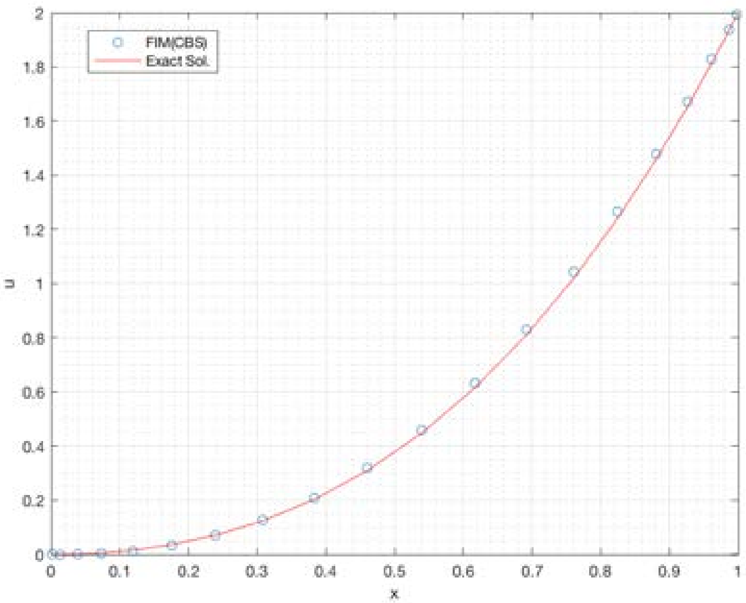


Figure 4.2: The graph of exact and approximate solutions of Example 4.2 with $\alpha = 1.5$.

Example 4.3. Consider a linear FDE with $\alpha \in (0, 1)$.

$$D^{\alpha}(u) + \frac{d^2u}{dx^2} + u = \frac{6x^{3-\alpha}}{\Gamma(4-\alpha)} + x^3 + 6x, \quad x \in (0, 1), \tag{4.16}$$

with boundary conditions $u(0) = 0$ and $u(1) = 1$. The exact solution is $u^*(x) = x^3$. By using our numerical algorithm, this problem can be written in the matrix form as

$$\mathbf{K}\mathbf{u} + c_0\mathbf{E} + c_1\bar{\mathbf{x}} = (\mathbf{A}^*)^2\mathbf{F},$$

where

$$\begin{aligned}\mathbf{F} &= \Big[\frac{6x_1^{3-\alpha}}{\Gamma(4-\alpha)} + x_1^3 + 6x_1, \frac{6x_2^{3-\alpha}}{\Gamma(4-\alpha)} + x_2^3 + 6x_2, \\ &\quad \frac{6x_3^{3-\alpha}}{\Gamma(4-\alpha)} + x_3^3 + 6x_3, ..., \frac{6x_M^{3-\alpha}}{\Gamma(4-\alpha)} + x_M^3 + 6x_M\Big]^T, \\ \mathbf{K} &= \frac{1}{\Gamma(2-\alpha)}\mathbf{A}^*\mathbf{Q} + \mathbf{I} + (\mathbf{A}^*)^2.\end{aligned}$$

For the boundary conditions, we have $\mathbf{t}_l(\mathbf{T}^*)^{-1}\mathbf{u} = 0$ and $\mathbf{t}_r(\mathbf{T}^*)^{-1}\mathbf{u} = 1$. Therefore, we solve the following linear system

$$\begin{bmatrix} \mathbf{K} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l(\mathbf{T}^*)^{-1} & 0 & 0 \\ \mathbf{t}_r(\mathbf{T}^*)^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\mathbf{A}^*)^2\mathbf{F} \\ 0 \\ 1 \end{bmatrix}. \tag{4.17}$$

Finally, we obtain the approximate solutions $u(x)$. In this example, according to Bhrawy et al. [3], they used the shifted Chebyshev spectral tau (SCT) method and define the error to be

$$E_{max} = \max \mid u_k(x) - u_k^*(x) \mid,$$

where u^* and u are the exact and numerical solutions, respectively. Table 4.5 shows the error E for our FIM using Chebyshev polynomials (CBS) when $M = 10$ with with several values of α. For each values of α, we give the optimal ω that achieve the highest accuracy. Then, the error E_{max} for our FIM using Chebyshev polynomials (CBS) comparing with Bhrawy et al.'s SCT method [3] are shown in Tables 4.6 when the number of nodes are equal and $\alpha = 0.5$. Also, the graph of the exact and approximate solutions are shown in Figure 4.3. Note that ω in Tables 4.6 is the optimal ω for $\alpha = 0.5$ that give the best accuracy.

Table 4.5**:** The errors E for FIM(CBS) in Example 4.3 when $M = 10$

α	ω	E	α	ω	E
0.001	0.54	3.4276×10^{-5}	0.7	0.32	2.1657×10^{-4}
0.1	0.53	3.3185×10^{-5}	0.9	0.14	1.8333×10^{-4}
0.3	0.49	3.4051×10^{-5}	0.99	0.02	7.4989×10^{-5}
0.5	0.43	1.3316×10^{-4}	0.999	0	3.4180×10^{-5}

Table 4.6**:** The errors E_{max} for FIM (CBS) and SCT in Example 4.3 when $\alpha = 0.5$

M	ω	FIM (CBS)	SCT
4	0.43	2.0190×10^{-4}	1.6×10^{-5}
8	0.43	3.4531×10^{-4}	5.3×10^{-8}
16	0.44	2.1759×10^{-4}	2.3×10^{-10}
32	0.45	1.2561×10^{-4}	9.9×10^{-13}
64	0.47	5.3740×10^{-5}	1.0×10^{-14}

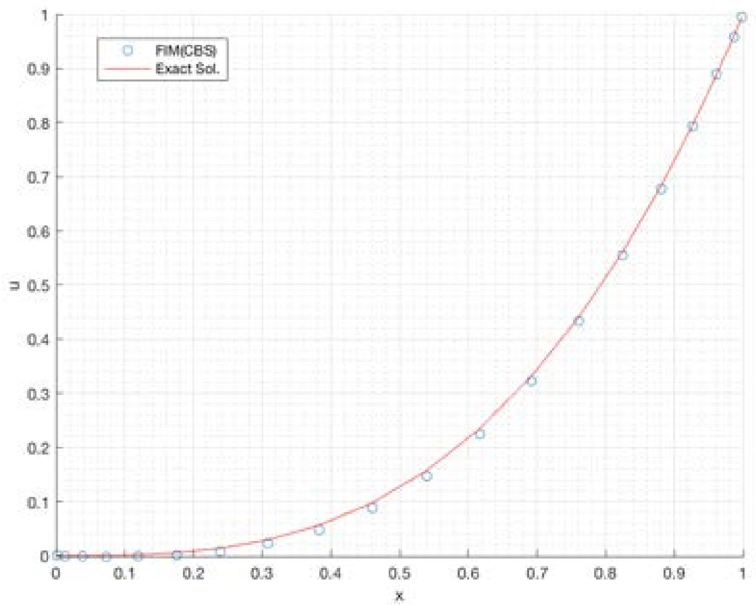


Figure 4.3**:** The graph of exact and approximate solutions of Example 4.3 with $\alpha = 0.5$.

Example 4.4**.** Consider a linear FDE with $\alpha \in (0, 1)$.

$$D^{\alpha}(u) + \frac{d^2u}{dx^2} + \frac{6}{5}\frac{du}{dx} + \frac{1}{5}u = \frac{3.5x^{\frac{5}{2}-\alpha}}{\Gamma(3.5-\alpha)} + \frac{2x^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{1}{5}x^{\frac{5}{2}} + \frac{1}{5}x^2 + 3x^{\frac{3}{2}} + \frac{15}{4}x^{\frac{1}{2}} + \frac{12}{5}x + 2, \quad x \in (0, 1), \tag{4.18}$$

with boundary conditions $u(0) = 0$ and $u(1) = 2$. The exact solution is $u^*(x) = x^{\frac{5}{2}} + x^2$.

By using our numerical algorithm, this problem can be written in the matrix form as

$$\mathbf{K}\mathbf{u} + c_0\mathbf{E} + c_1\bar{\mathbf{x}} = (\mathbf{A}^*)^2\mathbf{F},$$

where

$$\begin{aligned}
\mathbf{F} &= \Big[\frac{3.5x_1^{\frac{5}{2}-\alpha}}{\Gamma(3.5-\alpha)} + \frac{2x_1^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{1}{5}x_1^{\frac{5}{2}} + \frac{1}{5}x_1^2 + 3x_1^{\frac{3}{2}} + \frac{15}{4}x_1^{\frac{1}{2}} + \frac{12}{5}x_1 + 2,\\
&\quad \frac{3.5x_2^{\frac{5}{2}-\alpha}}{\Gamma(3.5-\alpha)} + \frac{2x_w^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{1}{5}x_2^{\frac{5}{2}} + \frac{1}{5}x_2^2 + 3x_2^{\frac{3}{2}} + \frac{15}{4}x_2^{\frac{1}{2}} + \frac{12}{5}x_2 + 2,\\
&\quad \frac{3.5x_3^{\frac{5}{2}-\alpha}}{\Gamma(3.5-\alpha)} + \frac{2x_3^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{1}{5}x_3^{\frac{5}{2}} + \frac{1}{5}x_3^2 + 3x_3^{\frac{3}{2}} + \frac{15}{4}x_3^{\frac{1}{2}} + \frac{12}{5}x_3 + 2, ...,\\
&\quad \frac{3.5x_M^{\frac{5}{2}-\alpha}}{\Gamma(3.5-\alpha)} + \frac{2x_M^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{1}{5}x_M^{\frac{5}{2}} + \frac{1}{5}x_M^2 + 3x_M^{\frac{3}{2}} + \frac{15}{4}x_M^{\frac{1}{2}} + \frac{12}{5}x_M + 2\Big]^T,\\
\mathbf{K} &= \frac{1}{\Gamma(2-\alpha)}\mathbf{A}^*\mathbf{Q} + \mathbf{I} + \frac{6}{5}\mathbf{A}^* + \frac{1}{5}(\mathbf{A}^*)^2.
\end{aligned}$$

For the boundary conditions, we have $\mathbf{t}_l(\mathbf{T}^*)^{-1}\mathbf{u} = 0$ and $\mathbf{t}_r(\mathbf{T}^*)^{-1}\mathbf{u} = 2$. Therefore, we solve the following linear system

$$\begin{bmatrix} \mathbf{K} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l(\mathbf{T}^*)^{-1} & 0 & 0 \\ \mathbf{t}_r(\mathbf{T}^*)^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\mathbf{A}^*)^2\mathbf{F} \\ 0 \\ 2 \end{bmatrix}. \tag{4.19}$$

Finally, we obtain the approximate solutions $u(x)$. Table 4.7 shows the error E for our FIM using Chebyshev polynomials (CBS) when $M = 10$ with different α and ω. Also, the graph of the exact and approximate solutions are shown in Figure 4.4. Note that ω in Table 4.7 are the optimal ω that give the best accuracy.

Table 4.7: The errors E for FIM(CBS) in Example 4.4 when $M = 10$

α	ω	E
0.001	0.53	3.5582×10^{-5}
0.1	0.52	4.0456×10^{-5}
0.3	0.48	8.0875×10^{-5}
0.5	0.43	1.6888×10^{-4}

α	ω	E
0.7	0.33	2.9007×10^{-4}
0.9	0.15	2.4695×10^{-4}
0.99	0.02	5.0977×10^{-5}
0.999	0	2.3218×10^{-5}

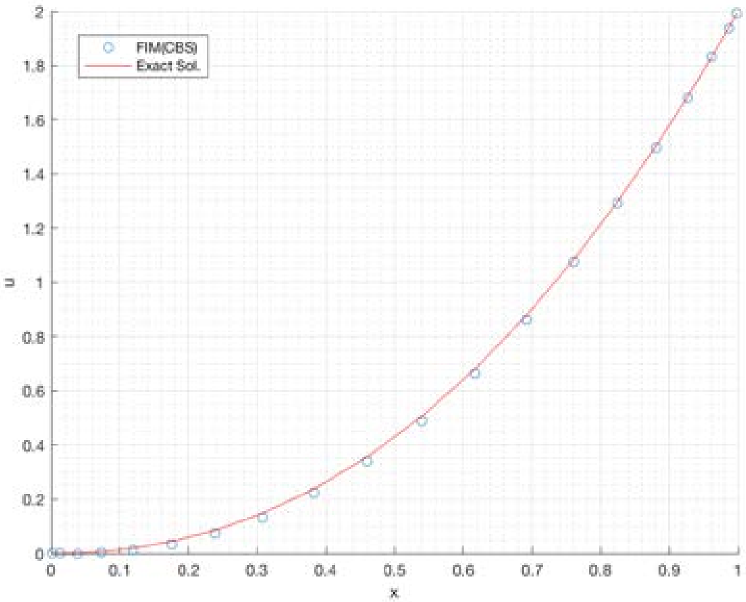


Figure 4.4**:** The graph of exact and approximate solutions of Example 4.4 with $\alpha = 0.5$.

4.3 Further Observation

From Examples 4.1 and 4.2, for $1 < \alpha < 2$, we can see from Table 4.1 and 4.4 that there is a relationship between α and the optimal ω that gives the best accuracy as shown in Figure 4.5. Similarly, for $0 < \alpha < 1$, Figure 4.6 shows the relationship between α and the optimal ω for Examples 4.3 and 4.4.

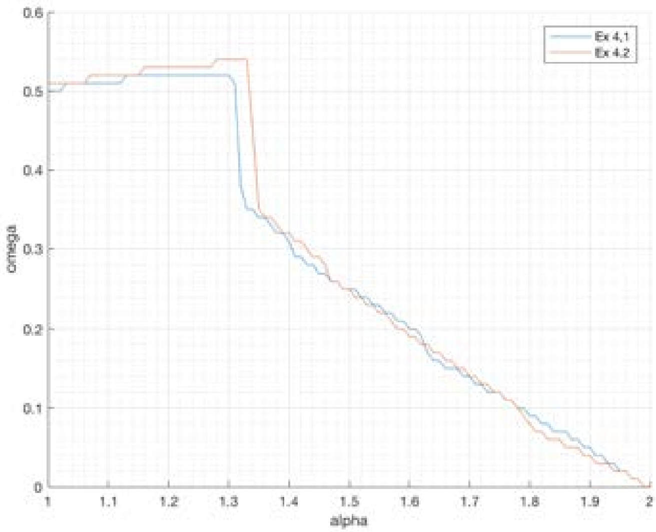


Figure 4.5**:** The optimal ω versus α from Examples 4.1-4.2 when $M = 10$.

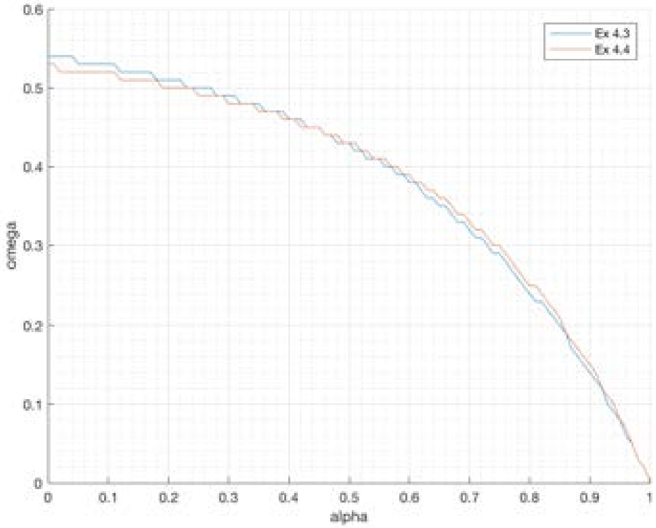


Figure 4.6**:** The optimal ω versus α from Examples 4.3-4.4 when $M = 10$.

Therefore, we automate choose ω according to the interpolation polynomials that interpolate the average of ω from each to examples corresponding to the value of α. Thus, we add Step 0 for choosing ω in our numerical algorithm from section 4.2 as follow

Step 0 Choose ω

For $\alpha \in (0, 1)$,

$$\omega = -0.6238\alpha^3 + 0.3120\alpha^2 - 0.2175\alpha + 0.5397.$$

For $\alpha \in (1, 2)$,

$$\omega = \begin{cases} 0.0775\alpha + 0.4298 & \text{if } 1 \leq \alpha \leq 1.31, \\ -0.5643\alpha + 1.1080 & \text{if } 1.31 < \alpha \leq 2. \end{cases}$$

However, if ω obtained from these formulas is less than 0, then take $\omega = 0$. The next two examples verify our suggestion about the optimal ω.

Example 4.5**.** Consider a linear FDE with $\alpha \in (0,1)$.

$$D^{\alpha}(u) + \frac{d^2u}{dx^2} + \frac{du}{dx} - u = \frac{24x^{4-\alpha}}{\Gamma(5-\alpha)} + \frac{2x^{2-\alpha}}{\Gamma(3-\alpha)} + 4x^3 + 11x^2 + 2x - x^4 + 2, \quad x \in (0,1), \tag{4.20}$$

with boundary conditions $u(0) = 0$ and $u(1) = 2$. The exact solution is $u^*(x) = x^4 + x^2$. By using our numerical algorithm with Step 0, we can see that our suggested ω is a good approximation for the optimal ω of this problem as shown in Table 4.8 and Figure 4.7.

Table 4.8**:** The errors E for FIM(CBS) in Example 4.5 when $M = 10$

α	optimal ω	E
0.001	0.54	3.4169×10^{-5}
0.01	0.53	3.8895×10^{-5}
0.1	0.52	3.9675×10^{-5}
0.3	0.49	4.2996×10^{-5}
0.5	0.43	1.2478×10^{-4}
0.7	0.33	2.1812×10^{-4}
0.9	0.14	1.9063×10^{-4}
0.99	0.02	5.1359×10^{-5}
0.999	0	2.6051×10^{-5}

α	suggested ω	E
0.001	0.5395	3.0160×10^{-5}
0.01	0.5376	2.4006×10^{-5}
0.1	0.5204	3.3041×10^{-5}
0.3	0.4857	5.6484×10^{-5}
0.5	0.4310	1.2458×10^{-4}
0.7	0.3264	2.1634×10^{-4}
0.9	0.1419	1.8140×10^{-4}
0.99	0.0249	1.1259×10^{-4}
0.999	0.0119	1.4497×10^{-4}

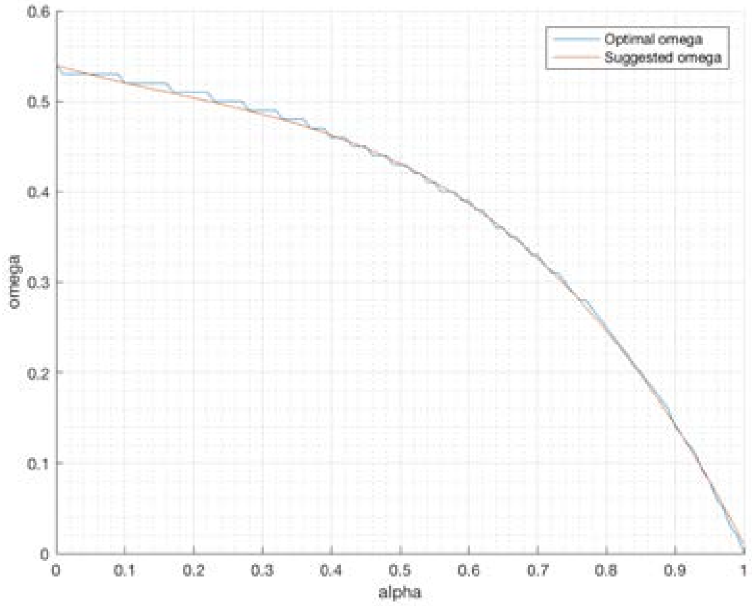


Figure 4.7**:** The optimal ω and suggested ω in Examples 4.5 when $M = 10$.

Example 4.6**.** Consider a linear FDE with $\alpha \in (1, 2)$.

$$D^{\alpha}(u) + \frac{d^2u}{dx^2} + 2u = \frac{6x^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{2x^{2-\alpha}}{\Gamma(3-\alpha)} + \frac{x^{1-\alpha}}{\Gamma(2-\alpha)} + 2x^3 + 2x^2 + 8x + 2, \quad x \in (0, 1), \tag{4.21}$$

with boundary conditions $u(0) = 0$ and $u(1) = 3$. The exact solution is $u^*(x) = x^3 + x^2 + x$. By using our numerical algorithm with Step 0, we can see that our suggested ω is a good approximation for the optimal ω of this problem as shown in Table 4.9 and Figure 4.8.

Table 4.9: The errors E for FIM(CBS) in Example 4.6 when $M = 10$

α	optimal ω	E
1.001	0.51	9.6872×10^{-5}
1.01	0.51	8.6439×10^{-5}
1.1	0.53	1.6900×10^{-4}
1.3	0.26	1.5695×10^{-3}
1.5	0.24	1.6789×10^{-3}
1.7	0.12	1.1829×10^{-3}
1.9	0.04	4.2573×10^{-4}
1.99	0	1.1761×10^{-4}
1.999	0	1.1957×10^{-5}

α	suggested ω	E
1.001	0.50738	1.0872×10^{-5}
1.01	0.50808	9.5303×10^{-5}
1.1	0.48727	4.6312×10^{-4}
1.3	0.37441	1.5738×10^{-3}
1.5	0.26155	1.7960×10^{-3}
1.7	0.14869	1.2926×10^{-3}
1.9	0.03538	4.4215×10^{-4}
1.99	0	1.1761×10^{-4}
1.999	0	1.1957×10^{-5}

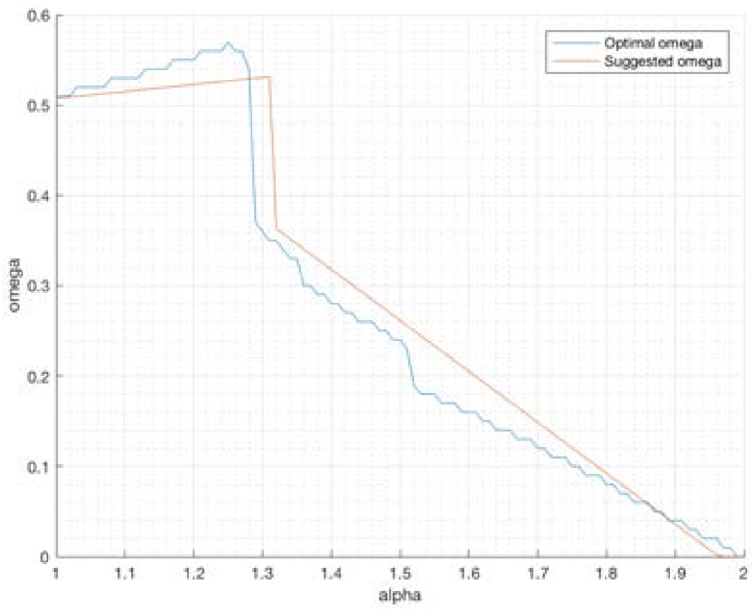


Figure 4.8: The optimal ω and suggested ω in Examples 4.6 when $M = 10$.

CHAPTER V

CONCLUSIONS

5.1 Conclusion of This Work

As we can see in Chapter III that by using forward difference and Crank-Nicolson method, we can construct the numerical procedures based on the FIM using Chebyshev polynomials suggested by Duangpan [4] to give a lot better accurate result comparing to other traditional FIMs. This better accuracy can be obtained even through we use a few numbers of computational points and a few number of time step. This thesis shows how to handle time-dependent second order linear PDEs. However, the same idea can be extended to the time-dependent linear PDEs of higher order as well.

In Chapter IV, we obtain a preliminary result in terms of having an algorithm to find the numerical solution for second order linear FDEs. The results show that for $\alpha \in (0, 1)$ and $\alpha \in (1, 2)$ then our method works very well. We also notice that in terms of the procedure, our numerical algorithm can be extended to solve FDEs of higher order.

5.2 Future Works

In this section, we propose the plan for the future works to extend our research as follows:

1. We would like to apply FIM by using Chebyshev polynomials to solve nonlinear differential equations.

2. We would like to improve our FIM by using Chebyshev polynomials for solving differential equations with other kind of boundary conditions namely, Neumann and Robin boundary conditions.

3. We would like to improve our FIM by using Chebyshev polynomials for solving

linear FDEs for multi-order in Riemann-Liouville sense.

4. We would like to improve our FIM by using Chebyshev polynomials for solving partial fractional order linear differential equations.

REFERENCES

[1] M.H. Annaby anf Z.S. Mansour, *q-Fractional Calculus and Equations*, Springer-Verlag, Heidelberg,2012.

[2] O.A. Bauchu, *Dymore User's Manual Chebyshev polynomials*, (1st ed.), Georgia Institute of Technology, Atlanta, 2007.

[3] A.H. Bhrawy, M.M. Tharwat, A. Yildirim, A new formula for fractional integrals of Chebyshev polynomials: Application for solving multi-term fractional differential equations, *Appl. Math. Model.*, **37** (2013) 4245-4252.

[4] A. Duangpan, R. Boonklurb, *Solving differential equations by finite integration method with orthogonal functions*, M.Sc. Thesis, Chulalongkorn University (2016).

[5] R.M. Evans, U.N. Katugampola, D.A. Edwards. Applications of fractional calculus in solving Abel-type integral equations: Surface-volume reaction problem. *Comput. Math. Appl.*, **00** (2016) 1-19.

[6] A. Gil, J. Segura and N. Temme, *Numerical Methods for Special Functions*, the Society for Industrial and Applied Mathematics, Philadelphia, 2007.

[7] R. Khalil, M. All Horani, A. Yousef, M. Sababheh, A new definition of fractional derivative, *J. Comput. Appl. Math.*, **264**(2014) 65-70.

[8] M. Li, C.S. Chen, Y.C. Hon, P.H. Wen, Finite integration method for solving multi-dimensional partial differential equations, *Appl. Math. Model.*, **39** (2015) 4979-4994.

[9] M. Li, Z.L. Tian, Y.C. Hon, C.S. Chen, P.H. Wen, Improved finite integration method for partial differential equations, *Eng. Anal. Boundary Elem.*, **64**(2016) 230-236.

[10] M. Rahimy, Applications of Fractional Differential Equations, *Appl. Math. Sci.*, **50** (2010) 2453-2461.

[11] G. Recktenwald, *Crank Nicolson Solution to the Heat Equation*. Retrieved from http://web.cecs.pdx.edu/ gerry/class/ME448/notes/pdf/CN_slides.pdf.

[12] P.H. Wen, Y.C. Hon, M. Li, T. Korakianitis, Finite integration method for partial differential equations, *Appl. Math. Model.*, **37**(2013) 10092-10106.

APPENDICES

We use MatLab to devise the code for computing our numerical solution using our proposed algorithms for each example in this research. In this appendix we would like to show the examples of code, which is the linear systems that are solved by the Gaussian elimination method in one-dimensional linear PDEs and using the Moore-Penrose Pseudoinverse for finding inverse to solving the linear systems in two-dimensional linear PDEs and linear FDEs.

APPENDIX A: MatLab code for calculating one-dimensional problems

Example A1. We consider Example 3.1

$$
\begin{aligned}
&\frac{\partial u}{\partial t} = x^2\frac{\partial^2 u}{\partial x^2} + 2x\frac{\partial u}{\partial x} + u - (x^2+2x)e^{x+t}, \quad x \in (0,1),\\
&u(x,0) = e^x, \quad x \in [0,1],\\
&u(0,t) = e^t, \quad u(1,t) = e^{t+1}, \quad t \geq 0.
\end{aligned}
$$

The exact solution is $u(x,t) = e^{x+t}$. Thus, for $\mathbf{L}^r = (4\mathbf{B}_1^r - 8\mathbf{A}\mathbf{B}_{1,\bar{x}}^r + 4\mathbf{A}^2\mathbf{B}_{1,\bar{x}\bar{x}}^r) + (2\mathbf{A}\mathbf{B}_2^r - 2\mathbf{A}^2\mathbf{B}_{2,\bar{x}}^r) + \mathbf{A}^2\mathbf{B}_3^r$, where $r \in \{j, j+1\}$, we can construct the linear system as following:

$$
\begin{bmatrix} \frac{2}{\tau}\mathbf{A}^2 - \mathbf{L}^{j+1} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l\mathbf{T}^{-1} & 0 & 0 \\ \mathbf{t}_r\mathbf{T}^{-1} & 0 & 0 \end{bmatrix}
\begin{bmatrix} \mathbf{u}^{j+1} \\ c_0 \\ c_1 \end{bmatrix}
=
\begin{bmatrix} \left(\frac{2}{\tau}\mathbf{A}^2 + \mathbf{L}^j\right)\mathbf{u}^j + \mathbf{A}^2(\mathbf{f}^j + \mathbf{f}^{j+1}) \\ e^{t_{j+1}} \\ e^{t_{j+1}+1} \end{bmatrix}.
$$

```
function Ex1CBS
N = 40;                                  % Number of nodes
a = 0;                                   % Left boundary
b = 1;                                   % Right boundary
time = 1;
tao = 0.01;
t = 0:tao:time;
xbar = -cos((2*(1:N)+1)/(2*(N+1))*pi);
% Construct A ---------------------------------------------------
```

```
T(:,1) = ones(N,1);
T(:,2) = xbar;
for i = 2:N
    T(:,i+1) = 2*xbar'.*T(:,i)-T(:,i-1);
end
Tbar(:,1) = xbar+1;
Tbar(:,2) = (xbar.^2-1)/2;
for i = 2:N-1
    Tbar(:,i+1) = (T(:,i+2)/(i+1)-T(:,i)/(i-1))/2-(-1)^i/(i^2-1);
end
Tinv = inv(T(:,1:N));
A = Tbar*Tinv;                  % First order integration matrix
% Boundary Conditions -------------------------------------------
for i = 1:N
    t0l(i) = (-1)^(i-1);
    t0r(i) = (1)^(i-1);
end
% Construct Block Matrix ----------------------------------------
X = [xbar' ones(N,1)];                   % Block matrix at P(1,2)
Y = [t0l*Tinv; t0r*Tinv];                % Block matrix at P(2,1)
Z = [0 0; 0 0];                          % Block matrix at P(2,2)
% Set Parameters -------------------------------
k = 2/(b-a);
x = ((b-a)*xbar+a+b)/2;           %Transform x into [a,b]
% Solve u and Find Error ----------------------------------------
u = exp(x');                       %Initial condition
U = u;
tj= 0;
for i = 2:length(t)
    B1 = diag(((xbar'+1)/2).^2);
    B2 = diag(2*(xbar'+1)/2);
```

```
    B3 = diag(ones(1,N));
    B11 = diag((xbar'+1)/2);
    B22 = diag(ones(1,N));
    B111= diag(1/2*ones(1,N));
    L1 = ((k^2*B1)-(2*k^2*A*B11)+(k^2*A^2*B111))+((k*A*B2)-(k*A^2*
        B22))+(A^2*B3);
    L0 = ((k^2*B1)-(2*k^2*A*B11)+(k^2*A^2*B111))+((k*A*B2)-(k*A^2*
        B22))+(A^2*B3);
    f0 = -(x'.^2+(2*x')).*exp(x'+tj);
    f1 = -(x'.^2+(2*x')).*exp(x'+tj+tao);
    W = (2/tao)*A^2-L1;
    F = ((2/tao)*A^2+L0)*u(1:N)+(A^2*(f0+f1));
    P = [W X; Y Z];                    %Construct Block Matrix
    q = [F; exp(tj+tao); exp(tj+tao+1)];
    u = pinv(P)*q;                     % Approximate solutions.
    tj= tj+tao;
end
ex = exp(x'+time);                     % Analytical solutions
E = 1/N*sum(abs((ex-u(1:N))./ex))      % average relative error
[x' ex u(1:N)];
% Plot Solution -------------------------------------------------
plot(x,u(1:N),'o');
hold on;
y = linspace(0,1,100);
e = exp(y+time)';
plot(y,e);
```

Example A2. We consider Example 3.2

$$\frac{\partial u}{\partial t} = x^2 e^t \frac{\partial^2 u}{\partial x^2} + x e^t \frac{\partial u}{\partial x} + u - 4x^2 e^{2t}, \quad x \in (0,2),$$
$$u(x,0) = x^2 + 1, \quad x \in [0,2],$$
$$u(0,t) = e^t, \quad u(2,t) = 5e^t, \quad t \geq 0.$$

The exact solution is $u(x,t) = e^t(x^2+1)$. Thus, for $\mathbf{L}^r = (\mathbf{B}_1^r - 2\mathbf{A}\mathbf{B}_{1,\bar{x}}^r + \mathbf{A}^2\mathbf{B}_{1,\bar{x}\bar{x}}^r) + (\mathbf{A}\mathbf{B}_2^r - \mathbf{A}^2\mathbf{B}_{2,\bar{x}}^r) + \mathbf{A}^2\mathbf{B}_3^r$, where $r \in \{j, j+1\}$, we can construct the linear system as following:

$$\begin{bmatrix} \frac{2}{\tau}\mathbf{A}^2 - \mathbf{L}^{j+1} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l\mathbf{T}^{-1} & 0 & 0 \\ \mathbf{t}_r\mathbf{T}^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{j+1} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\frac{2}{\tau}\mathbf{A}^2 + \mathbf{L}^j)\mathbf{u}^j + \mathbf{A}^2(\mathbf{f}^{\,j} + \mathbf{f}^{\,j+1}) \\ e^{t_{j+1}} \\ 5e^{t_{j+1}} \end{bmatrix}.$$

```
function Ex2CBS
N = 40;                                        % Number of nodes
a = 0;                                         % Left boundary
b = 2;                                         % Right boundary
time = 1;
tao = 0.001;
t = 0:tao:time;
xbar = -cos((2*(1:N)+1)/(2*(N+1))*pi);
% Construct A ---------------------------------------------------
T(:,1) = ones(N,1);
T(:,2) = xbar;
for i = 2:N
    T(:,i+1) = 2*xbar'.*T(:,i)-T(:,i-1);
end
Tbar(:,1) = xbar+1;
Tbar(:,2) = (xbar.^2-1)/2;
for i = 2:N-1
```

```
    Tbar(:,i+1) = (T(:,i+2)/(i+1)-T(:,i)/(i-1))/2-(-1)^i/(i^2-1);
end
Tinv = inv(T(:,1:N));
A = Tbar*Tinv;                  % First order integration matrix
% Boundary Conditions -------------------------------------------
for i = 1:N
    t0l(i) = (-1)^(i-1);
    t0r(i) = (1)^(i-1);
end
% Construct Block Matrix ----------------------------------------
X = [xbar' ones(N,1)];                  % Block matrix at P(1,2)
Y = [t0l*Tinv; t0r*Tinv];               % Block matrix at P(2,1)
Z = [0 0; 0 0];                         % Block matrix at P(2,2)
% Set Parameters ---------------------------------------------
k = 2/(b-a);
x = ((b-a)*xbar+a+b)/2;             %Transform x into [a,b]
% Solve u and Find Error ----------------------------------------
u = (x'.^2)+1;                       %Initial condition
U = u;
tj= 0;
for i = 2:length(t)
    B1 = diag(((xbar'+1).^2)*exp(tj));
    B2 = diag((xbar'+1)*exp(tj));
    B3 = diag(ones(1,N));
    B11 = diag(2*(xbar'+1)*exp(tj));
    B22 = diag(exp(tj)*ones(1,N));
    B111= diag(2*exp(tj)*ones(1,N));
    B1j = diag(((xbar'+1).^2)*exp(tj+tao));
    B2j = diag((xbar'+1)*exp(tj+tao));
    B3j = diag(ones(1,N));
    B11j = diag(2*(xbar'+1)*exp(tj+tao));
```

```
    B22j = diag(exp(tj+tao)*ones(1,N));
    B111j= diag(2*exp(tj+tao)*ones(1,N));
    L1 = ((k^2*B1j)-(2*k^2*A*B11j)+(k^2*A^2*B111j))+((k*A*B2j)-(k*
        A^2*B22j))+(A^2*B3j);
    L0 = ((k^2*B1)-(2*k^2*A*B11)+(k^2*A^2*B111))+((k*A*B2)-(k*A^2*
        B22))+(A^2*B3);
    f0 = -(4*x'.^2).*exp(2*tj);
    f1 = -(4*x'.^2).*exp(2*(tj+tao));
    W = (2/tao)*A^2-L1;
    F = ((2/tao)*A^2+L0)*u(1:N)+(A^2*(f0+f1));
    P = [W X; Y Z];          %Construct Block Matrix
    q = [F; exp(tj+tao); 5*exp(tj+tao)];
    u = pinv(P)*q;           % Approximate solutions.
    tj= tj+tao;
end
ex = (x'.^2+1)*exp(time);          % Analytical solutions
E = 1/N*sum(abs((ex-u(1:N))./ex)) % average relative error
[x' ex u(1:N)];
% Plot Solution ---------------------------------------------------
plot(x,u(1:N),'o');
hold on;
y = linspace(0,2,100);
e = (y.^2+1)*exp(time)';
plot(y,e);
```

Example A3. We consider Example 3.3

$$\frac{\partial u}{\partial t} = x^2\frac{\partial^2 u}{\partial x^2} + t\frac{\partial u}{\partial x} + x^2 u + 2t\sin(x+1) - t^3 cos(x+1), \quad x \in (0,1),$$

$$u(x,0) = 0, \quad x \in [0,1],$$

$$u(0,t) = t^2\sin(1), \quad u(1,t) = t^2\sin(2), \quad t \geq 0.$$

The exact solution is $u(x,t) = t^2 \sin(x+1)$. Thus, for $\mathbf{L}^r = (4\mathbf{B}_1^r - 8\mathbf{A}\mathbf{B}_{1,\bar{x}}^r + 4\mathbf{A}^2\mathbf{B}_{1,\bar{x}\bar{x}}^r) + \mathbf{A}^2\mathbf{B}_3^r$, where $r \in \{j, j+1\}$, we can construct the linear system as following:

$$\begin{bmatrix} \frac{2}{\tau}\mathbf{A}^2 - \mathbf{L}^{j+1} & \mathbf{E} & \bar{\mathbf{x}} \\ \mathbf{t}_l\mathbf{T}^{-1} & 0 & 0 \\ \mathbf{t}_r\mathbf{T}^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{j+1} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\frac{2}{\tau}\mathbf{A}^2 + \mathbf{L}^j)\mathbf{u}^j + \mathbf{A}^2(\mathbf{f}^{\,j} + \mathbf{f}^{\,j+1}) \\ t_{j+1}^2 \sin(1) \\ t_{j+1}^2 \sin(2) \end{bmatrix}.$$

```
function Ex3CBS
N = 40;                                         % Number of nodes
a = 0;                                          % Left boundary
b = 1;                                          % Right boundary
time = 1;
tao = 0.001;
t = 0:tao:time;
xbar = -cos((2*(1:N)+1)/(2*(N+1))*pi);
% Construct A ---------------------------------------------------
T(:,1) = ones(N,1);
T(:,2) = xbar;
for i = 2:N
    T(:,i+1) = 2*xbar'.*T(:,i)-T(:,i-1);
end
Tbar(:,1) = xbar+1;
Tbar(:,2) = (xbar.^2-1)/2;
for i = 2:N-1
    Tbar(:,i+1) = (T(:,i+2)/(i+1)-T(:,i)/(i-1))/2-(-1)^i/(i^2-1);
end
Tinv = inv(T(:,1:N));
A = Tbar*Tinv;                        % First order integration matrix
% Boundary Conditions -------------------------------------------
for i = 1:N
    t0l(i) = (-1)^(i-1);
```

```
    t0r(i) = (1)^(i-1);
end
% Construct Block Matrix ----------------------------------------
X = [xbar' ones(N,1)];                    % Block matrix at P(1,2)
Y = [t0l*Tinv; t0r*Tinv];                 % Block matrix at P(2,1)
Z = [0 0; 0 0];                           % Block matrix at P(2,2)
% Set Parameters ----------------------------------------------
k = 2/(b-a);
x = ((b-a)*xbar+a+b)/2;            %Transform x into [a,b]
% Solve u and Find Error ----------------------------------------
u = zeros(1,N)';                         %Initial condition
U = u;
tj= 0;
for i = 2:length(t)
    B1 = diag(((xbar'+1)/2).^2);
    B2 = diag(ones(1,N)*tj);
    B3 = diag(((xbar'+1)/2).^2);
    B11 = diag((xbar'+1)/2);
    B22 = diag(ones(1,N)*0);
    B111= diag(1/2*ones(1,N));
    B1j = diag(((xbar'+1)/2).^2);
    B2j = diag(ones(1,N)*(tj+tao));
    B3j = diag(((xbar'+1)/2).^2);
    B11j = diag((xbar'+1)/2);
    B22j = diag(ones(1,N)*0);
    B111j= diag(1/2*ones(1,N));
    L1 = ((k^2*B1j)-(2*k^2*A*B11j)+(k^2*A^2*B111j))+((k*A*B2j)-(k*
        A^2*B22j))+(A^2*B3j);
    L0 = ((k^2*B1)-(2*k^2*A*B11)+(k^2*A^2*B111))+((k*A*B2)-(k*A^2*
        B22))+(A^2*B3);
    f0 = 2*tj*sin(x'+1)-tj^3*cos(x'+1);
```

```
    f1 = 2*(tj+tao)*sin(x'+1)-(tj+tao)^3*cos(x'+1);
    W = (2/tao)*A^2-L1;
    F = ((2/tao)*A^2+L0)*u(1:N)+(A^2*(f0+f1));
    P = [W X; Y Z];          %Construct Block Matrix
    q = [F; (tj+tao)^2*sin(1); (tj+tao)^2*sin(2)];
    u = pinv(P)*q;           % Approximate solutions.
    tj= tj+tao;
end
ex = time^2*sin(x'+1);              % Analytical solutions
E = 1/N*sum(abs((ex-u(1:N))./ex)) % average relative error
[x' ex u(1:N)];
% Plot Solution -------------------------------------------------
plot(x,u(1:N),'o');
hold on;
y = linspace(0,1,100);
e = time^2*sin(y'+1);
plot(y,e);
```

APPENDIX B: MatLab code for calculating 2-dimensional problems

Example B1. We consider Example 3.4

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= (x^2+y^2+1)\frac{\partial^2 u}{\partial x^2} + (x^2+y^2+1)\frac{\partial^2 u}{\partial y^2} + 2x\frac{\partial u}{\partial x} + 2y\frac{\partial u}{\partial y} \\
&\quad - 2u + [1-2(x^2+y^2+x+y)]e^{x+y+t}, \\
u(x,y,0) &= e^{x+y}, \\
u(x,0,t) &= e^{x+t}, \quad u(x,1,t) = e^{x+1+t}, \quad x\in[0,1], \\
u(0,y,t) &= e^{y+t}, \quad u(1,y,t) = e^{1+y+t}, \quad y\in[0,1],
\end{aligned}
$$

where $t \in [0,1]$ and exact solution is $u(x,y,t) = e^{x+y+t}$. Thus, for $\mathbf{Z}^r = p^2\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_1^r - 2\mathbf{A}_x\bar{\boldsymbol{\alpha}}^r_{1,\bar{x}} + \mathbf{A}_x^2\bar{\boldsymbol{\alpha}}^r_{1,\bar{x}\bar{x}}) + q^2\mathbf{A}_x^2(\bar{\boldsymbol{\alpha}}_2^r - 2\mathbf{A}_y\bar{\boldsymbol{\alpha}}^r_{2,\bar{y}} + \mathbf{A}_y^2\bar{\boldsymbol{\alpha}}^r_{2,\bar{y}\bar{y}}) + p\mathbf{A}_x\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_3^r - \mathbf{A}_x\bar{\boldsymbol{\alpha}}^r_{3,\bar{x}}) + q\mathbf{A}_x^2\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_4^r - \mathbf{A}_y\bar{\boldsymbol{\alpha}}^r_{4,\bar{y}}\mathbf{u}^r) + \mathbf{A}_x^2\mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_5^r$, where $r \in \{j, j+1\}$, we can construct the linear system as following:

$$
\begin{bmatrix}
\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} - \mathbf{Z}^{j+1} & \mathbf{X}\boldsymbol{\Phi}_y & \boldsymbol{\Phi}_y & \mathbf{Y}\boldsymbol{\Phi}_x & \boldsymbol{\Phi}_x \\
\mathbf{T}_l & 0 & 0 & \cdots & 0 \\
\mathbf{T}_r & 0 & 0 & \cdots & 0 \\
\mathbf{T}_b\mathbf{P}^{-1} & \vdots & \vdots & \ddots & \vdots \\
\mathbf{T}_u\mathbf{P}^{-1} & 0 & 0 & \cdots & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{u}^{j+1} \\ \mathbf{f}_0 \\ \mathbf{f}_1 \\ \mathbf{g}_0 \\ \mathbf{g}_1
\end{bmatrix}
=
\begin{bmatrix}
\left(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} + \mathbf{Z}^j\right)\mathbf{u}^j + \mathbf{A}_x^2\mathbf{A}_y^2(\bar{\mathbf{F}}^{j+1} + \bar{\mathbf{F}}^j) \\
\bar{\mathbf{F}}_1^{j+1} \\
\bar{\mathbf{F}}_2^{j+1} \\
\bar{\mathbf{G}}_1^{j+1} \\
\bar{\mathbf{G}}_2^{j+1}
\end{bmatrix},
$$

```
function PDECBS1
N1 = input('Enter N1 : ');                  % # Nodes along x
N2 = input('Enter N2 : ');                  % # Nodes along y
M = N1*N2;                                  % # Total points
N = 2*(N1+N2);                              % # Boundary points
time = 1;                                   % # Time
```

```
tao = 0.01;
t = 0:tao:time;
e = @(x,y) exp(x+y+time);                      % Exact solution
% Discretize Domain ---------------------------------------------
a = 0; b = 1;                             % Left & Right bounded
c = 0; d = 1;                             % Bottom & Upper bounded
k = 2/(b-a); h = 2/(d-c); p = [];
xbar = -cos((2*(1:N1)-1)/(2*N1)*pi);   % Discretize x in [-1,1]
ybar = -cos((2*(1:N2)-1)/(2*N2)*pi);   % Discretize y in [-1,1]
x = ((b-a)*xbar+a+b)/2;                % Transform xbar into [a,b]
y = ((d-c)*ybar+c+d)/2;                % Transform ybar into [c,d]
for j = 1:N2
    for i = 1:N1
        p = [p; xbar(i) ybar(j)];
    end
end
x1 = p(:,1)'; x2 = ((b-a)*x1+a+b)/2;
y1 = p(:,2)'; y2 = ((d-c)*y1+c+d)/2;
% First Order Integration Matrix --------------------------------
P = Transformation(N1,N2);
Ax = BlockCBS(xbar,N1,N2);
Ay = P*BlockCBS(ybar,N2,N1)*P';
% Set Parameters ------------------------------------------------
X = diag(x2);                             % Diagonal matrix of xbar
Y = diag(y2);                             % Diagonal matrix of ybar
Px = PhiC(x1,N1,M);                       % Phix matrix
Py = PhiC(y1,N2,M);                       % Phiy matrix
% Construct Block Matrix ----------------------------------------
u = exp(x2+y2)';                           % Initial value
U = u;
tj= 0;
```

```
for i = 2:length(t)
    al1j = diag(x2.^2+y2.^2+1);
    al1xj = diag(x2);
    al1xxj = diag(0.5*ones(1,M));
    al2j = diag(x2.^2+y2.^2+1);
    al2yj = diag(y2);
    al2yyj = diag(0.5*ones(1,M));
    al3j = diag(2*x2);
    al3xj = diag(ones(1,M));
    al4j = diag(2*y2);
    al4yj = diag(ones(1,M));
    al5j = diag(-2*ones(1,M));
    al1j1 = diag(x2.^2+y2.^2+1);
    al1xj1 = diag(x2);
    al1xxj1 = diag(0.5*ones(1,M));
    al2j1 = diag(x2.^2+y2.^2+1);
    al2yj1 = diag(y2);
    al2yyj1 = diag(0.5*ones(1,M));
    al3j1 = diag(2*x2);
    al3xj1 = diag(ones(1,M));
    al4j1 = diag(2*y2);
    al4yj1 = diag(ones(1,M));
    al5j1 = diag(-2*ones(1,M));
    Z1 = ((k^2*Ay^2)*(al1j1-2*Ax*al1xj1+Ax^2*al1xxj1)+(h^2*Ax^2)*(
        al2j1-2*Ay*al2yj1+Ay^2*al2yyj1)+(k*Ax*Ay^2)*(al4j1-Ax*
        al4xj1)+(h*Ay*Ax^2)*(al5j1-Ay*al5yj1)+(Ax^2*Ay^2*al6j1));
    Z0 = ((k^2*Ay^2)*(al1j-2*Ax*al1xj+Ax^2*al1xxj)+(h^2*Ax^2)*(
        al2j-2*Ay*al2yj+Ay^2*al2yyj)+(k*Ax*Ay^2)*(al4j-Ax*al4xj)+(h
        *Ay*Ax^2)*(al5j-Ay*al5yj)+(Ax^2*Ay^2*al6j));
    fj1 = (1-2*(x2.^2+y2.^2+x2+y2)).*exp(x2+y2+tj+tao);
    % Right side fn j+1.
```

```
    fj = (1-2*(x2.^2+y2.^2+x2+y2)).*exp(x2+y2+tj);
    % Right side fn j.
    % Boundary Conditions --------------------------------------
    gl = @(y) exp(y+tj+tao);          % Left bound fn.
    gr = @(y) exp(y+tj+tao+1);        % Right bound fn.
    gb = @(x) exp(x+tj+tao);          % Bottom bound fn.
    gu = @(x) exp(x+tj+tao+1);        % Upper bound fn.
    [Tl wl] = BoundC5(xbar,y,-1,N1,N2,gl);
    [Tr wr] = BoundC5(xbar,y, 1,N1,N2,gr);
    [Tb wb] = BoundC5(ybar,x,-1,N2,N1,gb);
    [Tu wu] = BoundC5(ybar,x, 1,N2,N1,gu);
    B11 = Ax^2*Ay^2*(2/tao)-Z1;          % Block matrix at B(1,1)
    B12 = [X*Py Py Y*Px Px];             % Block matrix at B(1,2)
    B21 = [Tl; Tr; Tb*P'; Tu*P'];        % Block matrix at B(2,1)
    B22 = zeros(N);                      % Block matrix at B(2,2)
    B = [B11 B12; B21 B22];
    c = [(Ax^2*Ay^2*(2/tao)+Z0)*u(1:M)+(Ax^2*Ay^2)*(fj1'+fj'); wl
        '; wr'; wb'; wu'];
% Solve u and Find Error ----------------------------------------
    u = pinv(B)*c;
    tj= tj+tao;
end
uf = u(1:M);                             % Approximate solutions
s = e(x2,y2);                            % Analytical solutions
E = 1/M*sum(abs((s'-uf)./s'))            % Average relative error
% Plot Solution and Surface ------------------------------------
plot(1:M,uf,'o'); hold on;
plot(1:M,s);
grid on; grid minor;
[X,Y] = meshgrid(0:0.02:1,0:0.02:1);
Z = e(X,Y); figure; surf(X,Y,Z);
```

```
disp([x2' y2' s' uf]);                    % Display the solutions
```

Example B2. We consider Example 3.5

$$
\begin{aligned}
&\frac{\partial u}{\partial t} = x^2e^t\frac{\partial^2 u}{\partial x^2} + y^2e^t\frac{\partial^2 u}{\partial y^2} + xe^t\frac{\partial u}{\partial x} + ye^t\frac{\partial u}{\partial y} + u - (4x^2e^{2t} + 4y^2e^{2t}),\\
&u(x,y,0) = x^2 + y^2 + 1,\\
&u(x,0,t) = e^t(x^2+1), \quad u(x,1,t) = e^t(x^2+5), \quad x \in [0,2],\\
&u(0,y,t) = e^t(y^2+1), \quad u(1,y,t) = e^t(y^2+5), \quad y \in [0,2],
\end{aligned}
$$

where $t \in [0,1]$ and exact solution is $u(x,y,t) = e^t(x^2+y^2+1)$. Thus, for $\mathbf{Z}^r = p^2\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_1^r - 2\mathbf{A}_x\bar{\boldsymbol{\alpha}}_{1,\bar{x}}^r + \mathbf{A}_x^2\bar{\boldsymbol{\alpha}}_{1,\bar{x}\bar{x}}^r) + q^2\mathbf{A}_x^2(\bar{\boldsymbol{\alpha}}_2^r - 2\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{2,\bar{y}}^r + \mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_{2,\bar{y}\bar{y}}^r) + p\mathbf{A}_x\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_3^r - \mathbf{A}_x\bar{\boldsymbol{\alpha}}_{3,\bar{x}}^r) + q\mathbf{A}_x^2\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_4^r - \mathbf{A}_y\bar{\boldsymbol{\alpha}}_{4,\bar{y}}^r\mathbf{u}^r) + \mathbf{A}_x^2\mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_5^r$, where $r \in \{j, j+1\}$, we can construct the linear system as following:

$$
\begin{bmatrix}
\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} - \mathbf{Z}^{j+1} & \mathbf{X}\boldsymbol{\Phi}_y & \boldsymbol{\Phi}_y & \mathbf{Y}\boldsymbol{\Phi}_x & \boldsymbol{\Phi}_x\\
\mathbf{T}_l & 0 & 0 & \cdots & 0\\
\mathbf{T}_r & 0 & 0 & \cdots & 0\\
\mathbf{T}_b\mathbf{P}^{-1} & \vdots & \vdots & \ddots & \vdots\\
\mathbf{T}_u\mathbf{P}^{-1} & 0 & 0 & \cdots & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{u}^{j+1}\\ \mathbf{f}_0\\ \mathbf{f}_1\\ \mathbf{g}_0\\ \mathbf{g}_1
\end{bmatrix}
=
\begin{bmatrix}
\left(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} + \mathbf{Z}^j\right)\mathbf{u}^j + \mathbf{A}_x^2\mathbf{A}_y^2(\bar{\mathbf{F}}^{j+1} + \bar{\mathbf{F}}^j)\\
\bar{\mathbf{F}}_1^{j+1}\\
\bar{\mathbf{F}}_2^{j+1}\\
\bar{\mathbf{G}}_1^{j+1}\\
\bar{\mathbf{G}}_2^{j+1}
\end{bmatrix},
$$

```
function PDECBS2
N1 = input('Enter N1 : ');                       % # Nodes along x
N2 = input('Enter N2 : ');                       % # Nodes along y
M = N1*N2;                                       % # Total points
N = 2*(N1+N2);                                   % # Boundary points
```

```
time = 1;                                              % # Time
tao = 0.1;
t = 0:tao:time;
e = @(x,y) exp(time).*(x.^2+y.^2+1);           % Exact solution
% Discretize Domain ---------------------------------------------
a = 0; b = 2;                          % Left & Right bounded
c = 0; d = 2;                          % Bottom & Upper bounded
k = 2/(b-a); h = 2/(d-c); p = [];
xbar = -cos((2*(1:N1)-1)/(2*N1)*pi);  % Discretize x in [-1,1]
ybar = -cos((2*(1:N2)-1)/(2*N2)*pi);  % Discretize y in [-1,1]
x = ((b-a)*xbar+a+b)/2;               % Transform xbar into [a,b]
y = ((d-c)*ybar+c+d)/2;               % Transform ybar into [c,d]
for j = 1:N2
    for i = 1:N1
        p = [p; xbar(i) ybar(j)];
    end
end
x1 = p(:,1)'; x2 = ((b-a)*x1+a+b)/2;
y1 = p(:,2)'; y2 = ((d-c)*y1+c+d)/2;
% First Order Integration Matrix --------------------------------
P = Transformation(N1,N2);
Ax = BlockCBS(xbar,N1,N2);
Ay = P*BlockCBS(ybar,N2,N1)*P';
% Set Parameters ------------------------------------------------
X = diag(x2);                          % Diagonal matrix of xbar
Y = diag(y2);                          % Diagonal matrix of ybar
Px = PhiC(x1,N1,M);                    % Phix matrix
Py = PhiC(y1,N2,M);                    % Phiy matrix
% Construct Block Matrix ----------------------------------------
u = (x2.^2+y2.^2+1)';                        % Initial value
U = u;
```

```
tj= 0;
for i = 2:length(t)
    al1j = diag(exp(tj).*(x2.^2));
    al1xj = diag(2*exp(tj).*x2);
    al1xxj = diag(2*exp(tj).*ones(1,M));
    al2j = diag(exp(tj).*(y2.^2));
    al2yj = diag(2*exp(tj).*y2);
    al2yyj = diag(2*exp(tj).*ones(1,M));
    al3j = diag(exp(tj).*x2);
    al3xj = diag(exp(tj).*ones(1,M));
    al4j = diag(exp(tj).*y2);
    al4yj = diag(exp(tj).*ones(1,M));
    al5j = diag(ones(1,M));
    al1j1 = diag(exp(tj+tao).*(x2.^2));
    al1xj1 = diag(2*exp(tj+tao).*x2);
    al1xxj1 = diag(2*exp(tj+tao).*ones(1,M));
    al2j1 = diag(exp(tj+tao).*(y2.^2));
    al2yj1 = diag(2*exp(tj+tao).*y2);
    al2yyj1 = diag(2*exp(tj+tao).*ones(1,M));
    al3j1 = diag(exp(tj+tao).*x2);
    al3xj1 = diag(exp(tj+tao).*ones(1,M));
    al4j1 = diag(exp(tj+tao).*y2);
    al4yj1 = diag(exp(tj+tao).*ones(1,M));
    al5j1 = diag(ones(1,M));
    Z1 = ((k^2*Ay^2)*(al1j1-2*Ax*al1xj1+Ax^2*al1xxj1)+(h^2*Ax^2)*(
        al2j1-2*Ay*al2yj1+Ay^2*al2yyj1)+(k*Ax*Ay^2)*(al3j1-Ax*
        al3xj1)+(h*Ay*Ax^2)*(al4j1-Ay*al4yj1)+(Ax^2*Ay^2*al5j1));
    Z0 = ((k^2*Ay^2)*(al1j-2*Ax*al1xj+Ax^2*al1xxj)+(h^2*Ax^2)*(
        al2j-2*Ay*al2yj+Ay^2*al2yyj)+(k*Ax*Ay^2)*(al3j-Ax*al3xj)+(h
        *Ay*Ax^2)*(al4j-Ay*al4yj)+(Ax^2*Ay^2*al5j));
    fj1 = (-4*exp(2*(tj+tao)).*(x2.^2+y2.^2)); % Right side fn j+1
```

```
        .
    fj = (-4*exp(2*tj).*(x2.^2+y2.^2));  % Right side fn j.
    % Boundary Conditions ----------------------------------------
    gl = @(y) exp(tj+tao).*(y.^2+1);      % Left bound fn.
    gr = @(y) exp(tj+tao).*(y.^2+5);      % Right bound fn.
    gb = @(x) exp(tj+tao).*(x.^2+1);      % Bottom bound fn.
    gu = @(x) exp(tj+tao).*(x.^2+5);      % Upper bound fn.
    [Tl wl] = BoundC5(xbar,y,-1,N1,N2,gl);
    [Tr wr] = BoundC5(xbar,y, 1,N1,N2,gr);
    [Tb wb] = BoundC5(ybar,x,-1,N2,N1,gb);
    [Tu wu] = BoundC5(ybar,x, 1,N2,N1,gu);
    B11 = Ax^2*Ay^2*(2/tao)-Z1;           % Block matrix at B(1,1)
    B12 = [X*Py Py Y*Px Px];              % Block matrix at B(1,2)
    B21 = [Tl; Tr; Tb*P'; Tu*P'];         % Block matrix at B(2,1)
    B22 = zeros(N);                       % Block matrix at B(2,2)
    B = [B11 B12; B21 B22];
    c = [(Ax^2*Ay^2*(2/tao)+Z0)*u(1:M)+(Ax^2*Ay^2)*(fj1'+fj'); wl'; wr'; wb'; wu'];
    % Solve u and Find Error ------------------------------------
    u = pinv(B)*c;
    tj= tj+tao;
end
uf = u(1:M);                              % Approximate solutions
s = e(x2,y2);                             % Analytical solutions
E = 1/M*sum(abs((s'-uf)./s'))             % Average relative error
% Plot Solution and Surface ------------------------------------
plot(1:M,uf,'o'); hold on;
plot(1:M,s);
grid on; grid minor;
[X,Y] = meshgrid(0:0.02:2,0:0.02:2);
Z = e(X,Y); figure; surf(X,Y,Z);
```

```
disp([x2' y2' s' uf]);                    % Display the solutions
```

Example B3. We consider Example 3.6

$$\begin{aligned}
\frac{\partial u}{\partial t} &= (x^2+t^2)\frac{\partial^2 u}{\partial x^2} + (y^2+t^2)\frac{\partial^2 u}{\partial y^2} + (2x+t)\frac{\partial u}{\partial x} + (2y+t)\frac{\partial u}{\partial y} \\
&\quad + [(x^2+y^2+2t^2)t^2+2t]\sin(x+y+1) - (2x+2y+2t)t^2\cos(x+y+1), \\
u(x,y,0) &= 0, \\
u(x,0,t) &= t^2\sin(x+1), \quad u(x,1,t) = t^2\sin(x+2), \quad x\in[0,1], \\
u(0,y,t) &= t^2\sin(y+1), \quad u(1,y,t) = t^2\sin(y+2), \quad y\in[0,1],
\end{aligned}$$

where $t \in [0,1]$ and exact solution is $u(x,y,t) = t^2\sin(x+y+1)$. Thus, for $\mathbf{Z}^r = p^2\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_1^r - 2\mathbf{A}_x\bar{\boldsymbol{\alpha}}_{1,\bar{x}}^r + \mathbf{A}_x^2\bar{\boldsymbol{\alpha}}_{1,\bar{x}\bar{x}}^r) + q^2\mathbf{A}_x^2(\bar{\boldsymbol{\alpha}}_2^r - 2\mathbf{A}_y\bar{\boldsymbol{\alpha}}_{2,\bar{y}}^r + \mathbf{A}_y^2\bar{\boldsymbol{\alpha}}_{2,\bar{y}\bar{y}}^r) + p\mathbf{A}_x\mathbf{A}_y^2(\bar{\boldsymbol{\alpha}}_3^r - \mathbf{A}_x\bar{\boldsymbol{\alpha}}_{3,\bar{x}}^r) + q\mathbf{A}_x^2\mathbf{A}_y(\bar{\boldsymbol{\alpha}}_4^r - \mathbf{A}_y\bar{\boldsymbol{\alpha}}_{4,\bar{y}}^r\mathbf{u}^r)$, where $r \in \{j, j+1\}$, we can construct the linear system as following:

$$\begin{bmatrix}
\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} - \mathbf{Z}^{j+1} & \mathbf{X}\boldsymbol{\Phi}_y & \boldsymbol{\Phi}_y & \mathbf{Y}\boldsymbol{\Phi}_x & \boldsymbol{\Phi}_x \\
\mathbf{T}_l & 0 & 0 & \cdots & 0 \\
\mathbf{T}_r & 0 & 0 & \cdots & 0 \\
\mathbf{T}_b\mathbf{P}^{-1} & \vdots & \vdots & \ddots & \vdots \\
\mathbf{T}_u\mathbf{P}^{-1} & 0 & 0 & \cdots & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{u}^{j+1} \\ \mathbf{f}_0 \\ \mathbf{f}_1 \\ \mathbf{g}_0 \\ \mathbf{g}_1
\end{bmatrix}
= \begin{bmatrix}
\left(\mathbf{A}_x^2\mathbf{A}_y^2\frac{2}{\tau} + \mathbf{Z}^j\right)\mathbf{u}^j + \mathbf{A}_x^2\mathbf{A}_y^2(\bar{\mathbf{F}}^{j+1} + \bar{\mathbf{F}}^j) \\
\bar{\mathbf{F}}_1^{j+1} \\
\bar{\mathbf{F}}_2^{j+1} \\
\bar{\mathbf{G}}_1^{j+1} \\
\bar{\mathbf{G}}_2^{j+1}
\end{bmatrix},$$

```
function PDECBS3
N1 = input('Enter N1 : ');                % # Nodes along x
N2 = input('Enter N2 : ');                % # Nodes along y
M = N1*N2;                                % # Total points
```

```
N = 2*(N1+N2);                                      % # Boundary points
time = 1;                                           % # Time
tao = 0.1;
t = 0:tao:time;
e = @(x,y) time^2*sin(x+y+1);                          % Exact solution
% Discretize Domain ---------------------------------------------
a = 0; b = 1;                               % Left & Right bounded
c = 0; d = 1;                               % Bottom & Upper bounded
k = 2/(b-a); h = 2/(d-c); p = [];
xbar = -cos((2*(1:N1)-1)/(2*N1)*pi);  % Discretize x in [-1,1]
ybar = -cos((2*(1:N2)-1)/(2*N2)*pi);  % Discretize y in [-1,1]
x = ((b-a)*xbar+a+b)/2;               % Transform xbar into [a,b]
y = ((d-c)*ybar+c+d)/2;               % Transform ybar into [c,d]
for j = 1:N2
    for i = 1:N1
        p = [p; xbar(i) ybar(j)];
    end
end
x1 = p(:,1)'; x2 = ((b-a)*x1+a+b)/2;
y1 = p(:,2)'; y2 = ((d-c)*y1+c+d)/2;
% First Order Integration Matrix --------------------------------
P = Transformation(N1,N2);
Ax = BlockCBS(xbar,N1,N2);
Ay = P*BlockCBS(ybar,N2,N1)*P';
% Set Parameters ------------------------------------------------
X = diag(x2);                         % Diagonal matrix of xbar
Y = diag(y2);                         % Diagonal matrix of ybar
Px = PhiC(x1,N1,M);                   % Phix matrix
Py = PhiC(y1,N2,M);                   % Phiy matrix
% Construct Block Matrix ----------------------------------------
u = zeros(M,1);                       % Initial value
```

```
U = u;
tj= 0;
for i = 2:length(t)
    al1j = diag(x2.^2+tj^2);
    al1xj = diag(x2);
    al1xxj = diag(0.5*ones(1,M));
    al2j = diag(y2.^2+tj^2);
    al2yj = diag(y2);
    al2yyj = diag(0.5*ones(1,M));
    al3j = diag(2*x2+tj);
    al3xj = diag(ones(1,M));
    al4j = diag(2*y2+tj);
    al4yj = diag(ones(1,M));
    al1j1 = diag(x2.^2+(tj+tao)^2);
    al1xj1 = diag(x2);
    al1xxj1 = diag(0.5*ones(1,M));
    al2j1 = diag(y2.^2+(tj+tao)^2);
    al2yj1 = diag(y2);
    al2yyj1 = diag(0.5*ones(1,M));
    al3j1 = diag(2*x2+tj+tao);
    al3xj1 = diag(ones(1,M));
    al4j1 = diag(2*y2+tj+tao);
    al4yj1 = diag(ones(1,M));
    Z1 = ((k^2*Ay^2)*(al1j1-2*Ax*al1xj1+Ax^2*al1xxj1)+(h^2*Ax^2)*(
        al2j1-2*Ay*al2yj1+Ay^2*al2yyj1)+(k*Ax*Ay^2)*(al3j1-Ax*
        al3xj1)+(h*Ay*Ax^2)*(al4j1-Ay*al4yj1));
    Z0 = ((k^2*Ay^2)*(al1j-2*Ax*al1xj+Ax^2*al1xxj)+(h^2*Ax^2)*(
        al2j-2*Ay*al2yj+Ay^2*al2yyj)+(k*Ax*Ay^2)*(al3j-Ax*al3xj)+(h
        *Ay*Ax^2)*(al4j-Ay*al4yj));
    fj1 = ((x2.^2+y2.^2+(2*(tj+tao)^2))*(tj+tao)^2+(2*(tj+tao))).*
        sin(x2+y2+1)-((2*x2+2*y2+(2*(tj+tao)))*(tj+tao)^2).*cos(x2+
```

```
        y2+1);     % Right side fn j+1.
    fj = ((x2.^2+y2.^2+(2*tj^2))*tj^2+(2*tj)).*sin(x2+y2+1)-((2*x2
        +2*y2+2*tj)*tj^2).*cos(x2+y2+1);     % Right side fn j.
    % Boundary Conditions ---------------------------------------
    gl = @(y) (tj+tao)^2.*sin(y+1);    % Left bound fn.
    gr = @(y) (tj+tao)^2.*sin(y+2);    % Right bound fn.
    gb = @(x) (tj+tao)^2.*sin(x+1);    % Bottom bound fn.
    gu = @(x) (tj+tao)^2.*sin(x+2);    % Upper bound fn.
    [Tl wl] = BoundC5(xbar,y,-1,N1,N2,gl);
    [Tr wr] = BoundC5(xbar,y, 1,N1,N2,gr);
    [Tb wb] = BoundC5(ybar,x,-1,N2,N1,gb);
    [Tu wu] = BoundC5(ybar,x, 1,N2,N1,gu);
    B11 = Ax^2*Ay^2*(2/tao)-Z1;            % Block matrix at B(1,1)
    B12 = [X*Py Py Y*Px Px];               % Block matrix at B(1,2)
    B21 = [Tl; Tr; Tb*P'; Tu*P'];          % Block matrix at B(2,1)
    B22 = zeros(N);                        % Block matrix at B(2,2)
    B = [B11 B12; B21 B22];
    c = [(Ax^2*Ay^2*(2/tao)+Z0)*u(1:M)+(Ax^2*Ay^2)*(fj1'+fj'); wl
        '; wr'; wb'; wu'];
    % Solve u and Find Error ------------------------------------
    u = pinv(B)*c;
    tj= tj+tao;
end
uf = u(1:M);                               % Approximate solutions
s = e(x2,y2);                              % Analytical solutions
E = 1/M*sum(abs((s'-uf)./s'))              % Average relative error
% Plot Solution and Surface ------------------------------------
plot(1:M,uf,'o'); hold on;
plot(1:M,s);
grid on; grid minor;
[X,Y] = meshgrid(0:0.02:1,0:0.02:1);
```

```
Z = e(X,Y); figure; surf(X,Y,Z);
disp([x2' y2' s' uf]);                % Display the solutions
```

The following MatLab code are sub procedures that are used to calculate the second order linear partial differential equations problems for finite integration method using Chebyshev polynomial.

```
function P = Transformation(N1,N2)
P = zeros(N1*N2);
for i = 1:N1
    for j = 1:N2
        m = N1*(j-1)+i;
        n = N2*(i-1)+j;
        P(m,n) = 1;
    end
end
```

```
function [B v] = BoundC5(x,y,xb,N1,N2,g)
% vector t -----------------------------------
for i = 1:N1
    t(i) = (xb)^(i-1);
end
% Tinv ---------------------------------------
T(:,1) = ones(N1,1);
T(:,2) = x;
for i = 2:N1-1
    T(:,i+1) = 2*x'.*T(:,i)-T(:,i-1);
end
Tinv = [T(:,1)'; 2*T(:,2:N1)']/N1;
% Block B ------------------------------------
```

```
B = [];
for i = 1:N2
    B = blkdiag(B,t*Tinv);
    v(i) = g(y(i));
end
```

```
function T = PhiC(x,N,M)
T(:,1) = ones(M,1); T(:,2) = x;
for i = 2:N-1
    T(:,i+1) = 2*x'.*T(:,i)-T(:,i-1);
end
```

```
function A = BlockCBS(x,N1,N2)
A = [];
for i = 1:N2
    A = blkdiag(A,A_CBS(x,N1));
end
```

```
function A = A_CBS(x,N)
T(:,1) = ones(N,1); T(:,2) = x;
for i = 2:N
    T(:,i+1) = 2*x'.*T(:,i)-T(:,i-1);
end
Tbar(:,1) = x+1;
Tbar(:,2) = (x.^2-1)/2;
for i = 2:N-1
    Tbar(:,i+1) = (T(:,i+2)/(i+1)-T(:,i)/(i-1))/2-(-1)^i/(i^2-1);
end
T = T(:,1:N);
```

```
Tinv = [T(:,1)'; 2*T(:,2:N)']/N;
A = Tbar*Tinv;
```

APPENDIX C: MatLab code for calculating fractional differential equations

Example C1. We consider Example 4.1

$$D^{\alpha}(u) + \frac{d^2u}{dx^2} + u = \frac{2x^{2-\alpha}}{\Gamma(3-\alpha)} + x^2 + 2, \quad x \in (0,1),$$

with boundary conditions $u(0) = 0$ and $u(1) = 1$. The exact solution is $u^*(x) = x^2$. Thus, for $\mathbf{K} = \frac{1}{\Gamma(3-\alpha)}\mathbf{Q} + \mathbf{I} + (\mathbf{A}^*)^2$, we can construct the linear system as following:

$$\begin{bmatrix} \mathbf{K} & \mathbf{E} & \mathbf{x} \\ \mathbf{t}_l(\mathbf{T}^*)^{-1} & 0 & 0 \\ \mathbf{t}_r(\mathbf{T}^*)^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\mathbf{A}^*)^2\mathbf{F} \\ 0 \\ 1 \end{bmatrix}.$$

```
function [bomega, alpha]=Ex1()
N = 20;                                        % Number of nodes
a = 0;                                         % Left boundary
b = 1;                                         % Right boundary
u0 = 0;
ub = 1;
m=2;
xbar1 = 0.5*cos((2*(1:N)-1)/(2*(N))*pi)+0.5; % Discretize x
xbar = sort(xbar1);
% Construct A ---------------------------------------------------
T(:,1) = ones(N,1);
T(:,2) = 2*xbar-1;
for i = 2:N
    T(:,i+1) = 2*(2*xbar'-1).*T(:,i)-T(:,i-1);
end
Tbar(:,1) = xbar;
```

```
Tbar(:,2) = (xbar.^2-xbar);
for i = 2:N-1
    Tbar(:,i+1) = 0.5*((T(:,i+2)/(i+1)-T(:,i)/(i-1))/2-(-1)^i/(i^2-1));
end
Tinv = inv(T(:,1:N));
A = Tbar*Tinv;                   % First order integration matrix
% Boundary Conditions -------------------------------------------
for i = 1:N
    t0l(i) = (-1)^(i-1);
    t0r(i) = 1;
end
% Set Parameters --------------------------------------------
k = 1/(b-a);
x = (b-a)*xbar'+a;               % Transform xbar into [a,b]
% Construct Q------------------------------------------------
h = [0.01];
for ii = 1:length(h)
alpha = 1:h(ii):2;
%alpha = 1.5;
omega = 0:0.01:1;
for j = 1:length(alpha)
    beta = alpha(j)-m+1;
for i = 1:length(omega)
Q = Gam3(N,xbar,beta,omega(i));
f = 2*x.^(2-alpha(j))/gamma(3-alpha(j))+x.^2+2;
W = 1/(gamma(m+1-alpha(j)))*Q+A^2+eye(N);% Block matrix at P(1,1)
X = [xbar' ones(N,1)];                    % Block matrix at P(1,2)
Y = [t0l*Tinv; t0r*Tinv];                 % Block matrix at P(2,1)
Z = [0 0;0 0];                            % Block matrix at P(2,2)
% Solve u and Find Error ----------------------------------------
```

```
P = [W X; Y Z];
q = [A^2*f; u0; ub];
u = pinv(P)*q;                          % Approximate solutions
e = (x.^2);                               % Analytical solutions
omega(i);
E1(i) = mean(abs((e-u(1:N))./max(e))); % Average relative error
end
[M,I] = min(E1)
bomega(j) = omega(I)
end
plot(alpha,bomega)
hold on
end
for i = 1:length(bomega)
    if abs(bomega(i+1)-bomega(i))>0.05
        [alpha(i) bomega(i)];
        [alpha(i+1) bomega(i+1)];
        break
    end
end
pp1 = polyfit(alpha(1:i),bomega(1:i),1);
pp2 = polyfit(alpha(i+1:end),bomega(i+1:end),1);
pp11 = polyval(pp1,1:h:alpha(i));
pp22 = polyval(pp2,alpha(i+1):h:alpha(end));
plot(1:h:alpha(i),pp11,'o')
hold on
plot(alpha(i+1):h:alpha(end),pp22,'o')
[x u(1:N) e];
% Plot Solution -------------------------------------------------
%plot(x,u(1:N),'o');
%hold on;
```

```
%plot(x,e,'red');
grid on;
grid minor;
```

Example C2. We consider Example 4.2

$$D^{\alpha}(u) + \frac{d^2u}{dx^2} + 2\frac{du}{dx} - u = \frac{6x^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{2x^{2-\alpha}}{\Gamma(3-\alpha)} - x^3 + 5x^2 + 10x + 2, \quad x \in (0,1),$$

with boundary conditions $u(0) = 0$ and $u(1) = 2$. The exact solution is $u^*(x) = x^3 + x^2$. Thus, for $\mathbf{K} = \frac{1}{\Gamma(2-\alpha)}(\mathbf{A}^*)\mathbf{Q} + \mathbf{I} + 2(\mathbf{A}^*) - (\mathbf{A}^*)^2$, we can construct the linear system as following:

$$\begin{bmatrix} \mathbf{K} & \mathbf{E} & \mathbf{x} \\ \mathbf{t}_l(\mathbf{T}^*)^{-1} & 0 & 0 \\ \mathbf{t}_r(\mathbf{T}^*)^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\mathbf{A}^*)^2\mathbf{F} \\ 0 \\ 2 \end{bmatrix}.$$

```
function [bomega, alpha]=Ex2()
N = 20;                                     % Number of nodes
a = 0;                                      % Left boundary
b = 1;                                      % Right boundary
u0 = 0;
ub = 2;
m=2;
xbar1 = 0.5*cos((2*(1:N)-1)/(2*(N))*pi)+0.5; % Discretize x
xbar = sort(xbar1);
% Construct A ---------------------------------------------------
T(:,1) = ones(N,1);
T(:,2) = 2*xbar-1;
for i = 2:N
    T(:,i+1) = 2*(2*xbar'-1).*T(:,i)-T(:,i-1);
end
Tbar(:,1) = xbar;
```

```
Tbar(:,2) = (xbar.^2-xbar);
for i = 2:N-1
    Tbar(:,i+1) = 0.5*((T(:,i+2)/(i+1)-T(:,i)/(i-1))/2-(-1)^i/(i^2-1));
end
Tinv = inv(T(:,1:N));
A = Tbar*Tinv;                    % First order integration matrix
% Boundary Conditions -------------------------------------------
for i = 1:N
    t0l(i) = (-1)^(i-1);
    t0r(i) = 1;
end
% Set Parameters ----------------------------------------------
k = 1/(b-a);
x = (b-a)*xbar'+a;                % Transform xbar into [a,b]
% Construct Q--------------------------------------------------
h = [0.2];
for ii = 1:length(h)
alpha = 1.1:h(ii):2;
omega = 0:0.01:1;
for j = 1:length(alpha)
    beta = alpha(j)-m+1;
for i = 1:length(omega)
Q = Gam3(N,xbar,beta,omega(i));
f = (5*x.^2)-(x.^3)+(10*x)+2+(gamma(4)/gamma(4-alpha(j)))*(x.^(3-alpha(j)))+(gamma(3)/gamma(3-alpha(j)))*(x.^(2-alpha(j)));
W = 1/gamma(m+1-alpha(j))*Q+eye(N)+(2*A)-(A^2);% Block matrix at P(1,1)
X = [xbar' ones(N,1)];                    % Block matrix at P(1,2)
Y = [t0l*Tinv; t0r*Tinv];                 % Block matrix at P(2,1)
Z = [0 0;0 0];                              % Block matrix at P(2,2)
```

```
% Solve u and Find Error ----------------------------------------
P = [W X; Y Z];
q = [A^2*f; u0; ub];
u = pinv(P)*q;                     % Approximate solutions
%u = inv(W)*(z-X*inv(Y*inv(W)*X)*(Y*inv(W)*z-b2));
e = x.^3+x.^2;                     % Analytical solutions
omega(i);
E1(i) = mean(abs((e-u(1:N))./max(e))); % Average relative error
end
[M,I] = min(E1)
bomega(j) = omega(I)
eomega(j) = M;
end
plot(alpha,bomega)
hold on
end
for i = 1:length(bomega)
    if abs(bomega(i+1)-bomega(i))>0.05
        [alpha(i) bomega(i) eomega(i)];
        [alpha(i+1) bomega(i+1) eomega(i+1)];
        break
    end
end
plot(1:h:2,eomega)
pp1 = polyfit(alpha(1:i),bomega(1:i),1);
pp2 = polyfit(alpha(i+1:end),bomega(i+1:end),1);
pp11 = polyval(pp1,1:h:alpha(i));
pp22 = polyval(pp2,alpha(i+1):h:alpha(end));
plot(1:h:alpha(i),pp11,'o')
hold on
plot(alpha(i+1):h:alpha(end),pp22,'o')
```

```
[x u(1:N) e];
% Plot Solution ------------------------------------------------
%plot(x,u(1:N),'o');
%hold on;
%plot(x,e,'red');
grid on;
grid minor;
```

Example C3. We consider Example 4.3

$$D^{\alpha}(u) + \frac{d^2u}{dx^2} + u = \frac{6x^{3-\alpha}}{\Gamma(4-\alpha)} + x^3 + 6x, \quad x \in (0,1),$$

with boundary conditions $u(0) = 0$ and $u(1) = 1$. The exact solution is $u^*(x) = x^3$. Thus, for $\mathbf{K} = \frac{1}{\Gamma(2-\alpha)}\mathbf{A}^*\mathbf{Q} + \mathbf{I} + (\mathbf{A}^*)^2$, we can construct the linear system as following:

$$\begin{bmatrix} \mathbf{K} & \mathbf{E} & \mathbf{x} \\ \mathbf{t}_l(\mathbf{T}^*)^{-1} & 0 & 0 \\ \mathbf{t}_r(\mathbf{T}^*)^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} (\mathbf{A}^*)^2\mathbf{F} \\ 0 \\ 1 \end{bmatrix}.$$

```
function [bomega, alpha]=Ex3()
m=1;
N = 4;                                       % Number of nodes
a = 0;                                        % Left boundary
b = 1;                                        % Right boundary
u0 = 0;
ub = 1;
xbar1 = 0.5*cos((2*(1:N)-1)/(2*(N))*pi)+0.5; % Discretize x
xbar = sort(xbar1);
% Construct A --------------------------------------------------
T(:,1) = ones(N,1);
T(:,2) = 2*xbar-1;
```

```
for i = 2:N
    T(:,i+1) = 2*(2*xbar'-1).*T(:,i)-T(:,i-1);
end
Tbar(:,1) = xbar;
Tbar(:,2) = (xbar.^2-xbar);
for i = 2:N-1
    Tbar(:,i+1) = 0.5*((T(:,i+2)/(i+1)-T(:,i)/(i-1))/2-(-1)^i/(i
        ^2-1));
end
Tinv = inv(T(:,1:N));
A = Tbar*Tinv;                    % First order integration matrix
% Boundary Conditions -------------------------------------------
for i = 1:N
    t0l(i) = (-1)^(i-1);
    t0r(i) = 1;
end
% Set Parameters ---------------------------------------------
k = 1/(b-a);
x = (b-a)*xbar'+a;                % Transform xbar into [a,b]
% Construct Q---------------------------------------------
h = [0.2];
for ii = 1:length(h)
alpha = 0.1:h(ii):1;
%alpha = 0.5;
omega = 0:0.01:1;
for j = 1:length(alpha)
    beta = alpha(j)-m+1;
for i = 1:length(omega)
Q = Gam3(N,xbar,beta,omega(i));
f = x.^3+(gamma(4)/gamma(4-alpha(j)))*x.^(3-alpha(j))+6*x;
W = 1/(gamma(m+1-alpha(j)))*A*Q+A^2+eye(N);% Block matrix atP(1,1)
```

```
X = [xbar' ones(N,1)];        % Block matrix at P(1,2)
Y = [t0l*Tinv; t0r*Tinv];               % Block matrix at P(2,1)
Z = [0 0;0 0];                          % Block matrix at P(2,2)
% Solve u and Find Error ----------------------------------------
P = [W X; Y Z];
q = [A^2*f; u0; ub];
u = pinv(P)*q;                 % Approximate solutions
%u = inv(W)*(z-X*inv(Y*inv(W)*X)*(Y*inv(W)*z-b2));
e = x.^3;       % Analytical solutions
omega(i);
E1(i) = mean(abs((e-u(1:N))./max(e))); % Average relative error
%E1(i) = max(abs((e-u(1:N))));
end
[M,I] = min(E1)
bomega(j) = omega(I)
end
plot(alpha,bomega)
hold on
end
pp = polyfit(alpha,bomega,3);
pp2 = polyval(pp,0:0.01:1);
plot(0:0.01:1,pp2,'o')
%EE = mean(abs(pp2-bomega));
[x u(1:N) e]
% Plot Solution -------------------------------------------------
%plot(x,u(1:N),'o');
%hold on;
%plot(x,e,'red');
grid on;
grid minor;
```

Example C4. We consider Example 4.4

$$D^{\alpha}(u)+\frac{d^2u}{dx^2}+\frac{6}{5}\frac{du}{dx}+\frac{1}{5}u=\frac{3.5x^{\frac{5}{2}-\alpha}}{\Gamma(3.5-\alpha)}+\frac{2x^{2-\alpha}}{\Gamma(3-\alpha)}+\frac{1}{5}x^{\frac{5}{2}}+\frac{1}{5}x^2$$
$$+3x^{\frac{3}{2}}+\frac{15}{4}x^{\frac{1}{2}}+\frac{12}{5}x+2,\quad x\in(0,1),$$

with boundary conditions $u(0)=0$ and $u(1)=2$. The exact solution is $u^*(x)=x^{\frac{5}{2}}+x^2$. Thus, for $\mathbf{K}=\frac{1}{\Gamma(2-\alpha)}\mathbf{A}^*\mathbf{Q}+\mathbf{I}+(\mathbf{A}^*)^2$, we can construct the linear system as following:

$$\begin{bmatrix}\mathbf{K} & \mathbf{E} & \mathbf{x}\\ \mathbf{t}_l(\mathbf{T}^*)^{-1} & 0 & 0\\ \mathbf{t}_r(\mathbf{T}^*)^{-1} & 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ c_0\\ c_1\end{bmatrix}=\begin{bmatrix}(\mathbf{A}^*)^2\mathbf{F}\\ 0\\ 2\end{bmatrix}.$$

```
function [bomega, alpha]=Ex4()
m=1;
N = 10;                                  % Number of nodes
a = 0;                                   % Left boundary
b = 1;                                   % Right boundary
u0 = 0;
ub = 2;
xbar1 = 0.5*cos((2*(1:N)-1)/(2*(N))*pi)+0.5; % Discretize x
xbar = sort(xbar1);
% Construct A ---------------------------------------------------
T(:,1) = ones(N,1);
T(:,2) = 2*xbar-1;
for i = 2:N
    T(:,i+1) = 2*(2*xbar'-1).*T(:,i)-T(:,i-1);
end
Tbar(:,1) = xbar;
Tbar(:,2) = (xbar.^2-xbar);
for i = 2:N-1
    Tbar(:,i+1) = 0.5*((T(:,i+2)/(i+1)-T(:,i)/(i-1))/2-(-1)^i/(i
```

```
        ^2-1));
end
Tinv = inv(T(:,1:N));
A = Tbar*Tinv;                  % First order integration matrix
% Boundary Conditions -------------------------------------------
for i = 1:N
    t0l(i) = (-1)^(i-1);
    t0r(i) = 1;
end
% Set Parameters --------------------------------------------
k = 1/(b-a);
x = (b-a)*xbar'+a;              % Transform xbar into [a,b]
% Construct Q----------------------------------------------
h = [0.01];
for ii = 1:length(h)
alpha = 0:h(ii):1;
%alpha = 0.999
omega = 0:0.01:1;
for j = 1:length(alpha)
    beta = alpha(j)-m+1;
for i = 1:length(omega)
Q = Gam3(N,xbar,beta,omega(i));
f = (1/5)*x.^(5/2)+(1/5)*x.^2+3*x.^(3/2)+(15/4)*x.^(1/2)+(12/5)*x
    +2+...
(gamma(7/2)/gamma((7/2)-alpha(j)))*x.^((5/2)-alpha(j))+2/gamma(3-
    alpha(j))*x.^(2-alpha(j));
W = 1/(gamma(m+1-alpha(j)))*A*Q+(1/5)*A^2+(6/5)*A+eye(N);
                        % Block matrix at P(1,1)
X = [xbar' ones(N,1)];         % Block matrix at P(1,2)
Y = [t0l*Tinv; t0r*Tinv];               % Block matrix at P(2,1)
Z = [0 0;0 0];                          % Block matrix at P(2,2)
```

```
% Solve u and Find Error ----------------------------------------
P = [W X; Y Z];
q = [A^2*f; u0; ub];
u = pinv(P)*q;                  % Approximate solutions
%u = inv(W)*(z-X*inv(Y*inv(W)*X)*(Y*inv(W)*z-b2));
e = x.^(5/2)+x.^2;      % Analytical solutions
omega(i);
E1(i) = mean(abs((e-u(1:N))./max(e))); % Average relative error
end
[M,I] = min(E1);
bomega(j) = omega(I);
end
plot(alpha,bomega)
hold on
end
pp = polyfit(alpha,bomega,3);
pp2 = polyval(pp,0:0.01:1);
plot(0:0.01:1,pp2,'o')
%EE = mean(abs(pp2-bomega));
%[x u(1:N) e]
% Plot Solution -------------------------------------------------
%plot(x,u(1:N),'o');
%hold on;
%plot(x,e,'red');
grid on;
grid minor;
```

The following MatLab code are sub procedures that are used to calculate the linear fractional differential equations problems for finite integration method using shifted Chebyshev polynomial.

```
function Q = Gam3(N,xbar,beta,omega)
Q = zeros(N);
Q(1,1) = (1-omega)*xbar(1)^(1-beta);
for k = 2:length(xbar)
    for i = 1:k
        if i == 1
            Q(k,i) = omega*((xbar(k)-xbar(i))^(1-beta)-(xbar(k)-xbar(i+1))^(1-beta))...
                +(1-omega)*((xbar(k))^(1-beta)-(xbar(k)-xbar(i))^(1-beta));
        elseif i == k
            Q(k,i) = (1-omega)*((xbar(k)-xbar(k-1))^(1-beta));
        else
            Q(k,i) = (1-omega)*((xbar(k)-xbar(i-1))^(1-beta)-(xbar(k)-xbar(i))^(1-beta))...
                        +omega*((xbar(k)-xbar(i))^(1-beta)-(xbar(k)-xbar(i+1))^(1-beta));
        end
    end
end
```

APPENDIX D: MatLab code for calculating average of ω

```
h = 0.01;
[bomega1, alpha]=Ex1();
[bomega2, alpha]=Ex2();
plot(alpha,bomega1)
hold on
plot(alpha,bomega2)
A = (bomega1+bomega2)/2;
for i = 1:length(A)
    if abs(A(i+1)-A(i))>0.05
        [alpha(i) A(i)]
        [alpha(i+1) A(i+1)]
        break
    end
end
pp1 = polyfit(alpha(1:i),A(1:i),1)
pp2 = polyfit(alpha(i+1:end),A(i+1:end),1)
pp11 = polyval(pp1,1:h:alpha(i));
pp22 = polyval(pp2,alpha(i+1):h:alpha(end));
%plot(alpha,A)
grid on
grid minor
```

```
[bomega1, alpha]=Ex3();
[bomega2, alpha]=Ex4();
plot(alpha,bomega1)
hold on
plot(alpha,bomega2)
A = (bomega1+bomega2)/2;
```

```
pp = polyfit(alpha,A,3)
pp2 = polyval(pp,0:0.01:1);
%plot(alpha,A)
grid on
grid minor
```

BIOGRAPHY

Name Mr. Arnont Saengsiritongchai

Date of Birth 5 May 1994

Place of Birth Bangkok, Thailand

Education B.Sc. (Applied Mathematics) (First Class Honours), King Mongkut Institute of Technology Ladkrabang, 2015

Publication

- A. Saengsiritongchai and R. Booklurb, Finite Integration Method Using Chebyshev Polynomials for Solving Linear Time-Dependent Differential Equations, *Proceeding of The 23rd Annual Meeting in Mathematics* (2018), 232-238.