

วิธีปริพันธ์อินต๊ะที่ใช้การกระจายเซบีเซฟสำหรับการแก้สมการความร้อนที่มีเงื่อนไขขอบไม่เฉพาะที่



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2565

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

FINITE INTEGRATION METHOD USING CHEBYSHEV EXPANSION FOR
SOLVING HEAT EQUATION WITH NON-LOCAL BOUNDARY CONDITIONS

Mr. Thanakorn Prasansri



จุฬาลงกรณ์มหาวิทยาลัย

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 2022

Copyright of Chulalongkorn University

ธนากร ประสานศรี : วิธีปริพันธ์อันตะที่ใช้การกระจายเซบีเซฟสำหรับการแก้สมการความร้อนที่มีเงื่อนไขขอบไม่เฉพาะที่. (FINITE INTEGRATION METHOD USING CHEBYSHEV EXPANSION FOR SOLVING HEAT EQUATION WITH NON-LOCAL BOUNDARY CONDITIONS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร.รตินันท์ บุญเคลือบ, 56 หน้า.

ในวิทยานิพนธ์ฉบับนี้ ได้สร้างขั้นตอนวิธีเชิงตัวเลขบนพื้นฐานของระเบียบวิธีปริพันธ์อันตะที่ปรับปรุงรวมกับการกระจายพหุนามเซบีเซฟและระเบียบวิธีแครงค์นิโคลสัน เพื่อจัดการกับตัวแปรเชิงปริภูมิและตัวแปรเชิงเวลา ตามลำดับ ขั้นตอนวิธีเชิงตัวเลขเหล่านี้ ออกแบบมาเพื่อคำนวณผลเฉลยโดยประมาณสำหรับสมการความร้อนในหนึ่งมิติและสองมิติซึ่งเกี่ยวข้องกับเงื่อนไขขอบไม่เฉพาะที่ เช่นเดียวกับสมการความร้อนในหนึ่งมิติที่มีเงื่อนไขขอบแบบโรบิน นอกจากนี้ มีการแสดงตัวอย่างเชิงตัวเลขที่คัดสรรมาเพื่อตรวจสอบประสิทธิภาพและความถูกต้องของขั้นตอนวิธีที่ได้นำเสนอ โดยหาผลเฉลยเชิงวิเคราะห์



ภาควิชา	คณิตศาสตร์และ	ลายมือชื่อนิสิต
	วิทยาการคอมพิวเตอร์	ลายมือชื่อ อ.ที่ปรึกษาหลัก
สาขาวิชา	คณิตศาสตร์ประยุกต์	
	และวิทยาการคณนา	
ปีการศึกษา	2565	

6470028223 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORDS : FINITE INTEGRATION METHOD / CHEBYSHEV EXPANSION / HEAT EQUATION / NON-LOCAL BOUNDARY CONDITION / ROBIN BOUNDARY CONDITION

THANAKORN PRASANSRI: FINITE INTEGRATION METHOD USING CHEBYSHEV EXPANSION FOR SOLVING HEAT EQUATION WITH NON-LOCAL BOUNDARY CONDITIONS. ADVISOR : ASSOC. PROF. RATINAN BOONKLURB, Ph.D., 56 pp.

In this thesis, we devise numerical algorithms based on the finite integration method enhanced with Chebyshev polynomial expansion and also the Crank-Nicolson method to manipulate the spatial and temporal variables, respectively. These algorithms are designed to calculate approximate solutions for one- and two-dimensional heat equations that involve non-local boundary conditions, as well as for one-dimensional heat equations featuring Robin boundary conditions. Furthermore, we illustrate a selection of numerical examples to verify the efficiency and accuracy of the proposed algorithms by their analytical solutions.



Department Mathematics and	Student's Signature
 Computer Science	Advisor's Signature
Field of Study Applied Mathematics and	
 Computational Science	
Academic Year 2022	

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor, Associate Professor Dr. Ratinan Boonklurb. I am immensely grateful to the individual who guided me through my research with his kindness, patience and valuable advice. His suggestions played a crucial role in making my work truly comprehensive. Furthermore, I would like to thank Dr. Ampol Duangpan, Mr. Arnont Saengsiritongchai and Miss Lalita Apisornpanich for helping me and teaching me fundamental knowledge for this thesis.

I also wish to acknowledge my esteemed thesis committee members, Associate Professor Dr. Petarpa Boonserm, Associate Professor Dr. Khamron Mekchay and Associate Professor Dr. Tawikan Treeyaprasert. Their scholarly input and constructive criticism have been indispensable in shaping this thesis, and for this, I am profoundly grateful.

Lastly, I wish to express my heartfelt gratitude towards my family and friends. Their unwavering support and encouragement have been the bedrock of my academic journey, offering me the strength and motivation required during challenging periods.

CONTENTS

	Page
ABSTRACT IN THAI	iv
ABSTRACT IN ENGLISH	v
ACKNOWLEDGEMENTS	vi
CONTENTS	vii
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER	
I INTRODUCTION	1
1.1 Motivation and literature surveys	1
1.2 Research objectives	2
1.3 Thesis overview	3
II PRELIMINARIES	4
2.1 Chebyshev polynomials	4
2.2 Kronecker product	6
2.3 Chebyshev integration matrices	7
2.3.1 FIM-CPE in one-dimension	7
2.3.2 FIM-CPE in two-dimension	10
2.4 Statements of heat equations	15
2.4.1 One-dimensional heat equation	16
2.4.2 Two-dimensional heat equation	18
III ONE-DIMENSIONAL HEAT EQUATION	20
3.1 Formulation of one-dimension heat equation	20
3.2 Algorithm for non-local boundary conditions	22
3.2.1 Numerical algorithm	23
3.2.2 Numerical examples	26
3.3 Algorithm for Robin boundary conditions	30
3.3.1 Numerical algorithm	30
3.3.2 Numerical example	33

CHAPTER	Page
IV TWO-DIMENSIONAL HEAT EQUATION	36
4.1 Formulation of two-dimension heat equation	36
4.2 Algorithm for non-local boundary conditions	39
4.2.1 Numerical algorithm	40
4.2.2 Numerical examples	46
V CONCLUSION	51
5.1 Conclusion and discussion	51
5.2 Future research directions	52
REFERENCES	53
BIOGRAPHY	56



LIST OF TABLES

Table	Page
3.1 The ARE of the numerical solutions $u(x, 1)$ for Example 3.1.	28
3.2 The MAE of the numerical solutions $u(x, 1)$ for Example 3.2.	29
3.3 The ARE of the numerical solutions $u(x, 1)$ for Example 3.3.	35
4.1 The ARE of numerical solutions u at time $T = 1$ for Example 4.1.	47
4.2 The ARE of numerical solutions u at time $T = 1$ for Example 4.2.	49



LIST OF FIGURES

Figure	Page
2.1 Chebyshev polynomials $T_n(x)$ for $n \in \{0, 1, 2, 3, 4\}$	4
2.2 The indices of the grid points globally and locally.	10
3.1 Graphical numerical solutions $u(x, t)$ from Algorithm 1 for Example 3.1.	28
3.2 Graphical numerical solutions $u(x, t)$ in (3.19) from Algorithm 1 for Example 3.2.	30
3.3 Graphical numerical solutions $u(x, t)$ in (3.19) from Algorithm 2 for Example 3.3.	35
4.1 Graphical numerical solutions $u(x, y, t)$ from Algorithm 3 for Example 4.1.	48
4.2 Graphical numerical solutions $u(x, y, t)$ from Algorithm 3 for Example 4.2.	50



CHAPTER I

INTRODUCTION

1.1 Motivation and literature surveys

The finite integration method (FIM) represents a recently developed computational approach for estimating solutions to boundary value problems that stem from ordinary differential equations (ODEs) and partial differential equations (PDEs). The core principle of FIM involves substituting the comprehensive solution domain, i.e., the entire area where the solution is sought, with a finite set of distinct points referred to as grid points. The objective subsequently becomes to find an approximate solution at these grid points, mirroring the process employed in the finite difference method (FDM). It is customary for these grid points to be uniformly distributed along the independent coordinates.

The integration matrix is a key component of the FIM. Traditionally, this matrix is obtained via direct numerical integrations. In 2013, Wen et al. [26] introduced a variation to the FIM, incorporating the standard trapezoidal integral and radial basis functions, a substantial development in the field for solving one-dimensional linear PDEs. In 2015, Li and his colleagues [19] extended the use of FIM, applying it to multi-dimensional problems. This expanded the potential applications of FIM significantly. A year later, Li et al. [20] introduced further improvements to the FIM, incorporating Simpson's rule, Newton-Cotes formulas and Lagrange polynomial methods for solving multi-dimensional linear PDEs. As a result of these advancements, the traditional FIM, where the integration matrix was derived based on quadrature formulas, exhibited superior accuracy when approximating solutions for differential equations, particularly when compared with other techniques like the FDM under identical conditions. In 2018, a notable modification to traditional FIMs was proposed by Duangpan and Boonklurb et al. [7,10]. They integrated Chebyshev polynomials as substitutes for trapezoidal, Simpson, Newton-Cotes and La-

grange functions to approximate solutions for steady-state linear PDEs. This method, later known as the FIM using the Chebyshev polynomial expansion (FIM-CPE), published in 2020 [3], significantly outperformed all preceding versions of FIM, marking a major leap forward in the field. There are applied to solve many problems, see [3–7, 11–16] for more details. Nevertheless, it is important to note that extensive applications of this FIM-CPE have not yet been explored to solve problems involving non-local boundary conditions. Consequently, this constitutes the principal focus of our research.

In this thesis, our objective is to devise numerical algorithms for finding approximate solutions to three types of boundary conditions: one-dimensional heat equations with non-local and Robin boundary conditions together with two-dimensional heat equations with non-local boundary conditions. These proposed algorithms leverage the FIM-CPE and the Crank-Nicolson method to manipulate the spatial and temporal variables, respectively, as a means to address these mathematical challenges. Moreover, we utilize a selection of pertinent examples to validate our proposed methodologies, comparing our findings to those of other established methods as well as their respective analytical solutions. For implementation, we utilize MatLab 2021b software on a system powered by an Intel(R) Core(TM) i7-6700 CPU operating at 3.40 GHz. The resulting data provides compelling evidence of the efficacy of our proposed algorithms, demonstrating their capacity to yield a marked improvement in accuracy, while maintaining low computational costs. This offers the potential for substantial advancements in the processing and understanding of heat equations with varying boundary conditions.

1.2 Research objectives

This research aims to develop numerical algorithms that are based on the FIM-CPE. The primary objective of these algorithms is to find approximate solutions for one- and two-dimensional heat equations that are governed by non-local boundary conditions and also Robin boundary conditions for one-dimensional problem. The ambition is to create a robust numerical model capable of dealing with these types of heat equations and providing reliable and approximate solutions.

1.3 Thesis overview

This thesis is organized into five distinct chapters, each fulfilling a specific purpose. Chapter I provides an introduction and the motivation behind this thesis, outlining the research objectives and offering an overview of the thesis structure. Chapter II delves into the preliminary knowledge essential for understanding this work, including an introduction to Chebyshev polynomials, a presentation on the FIM-CPE in both one- and two-dimensional spaces, and an exploration of one- and two-dimensional heat equations with non-local boundary conditions. Chapter III applies the FIM-CPE and also the Crank-Nicolson method as a solution strategy to construct numerical algorithms for one-dimensional heat equations with non-local and Robin boundary conditions. This chapter includes several numerical examples to illustrate the application of the method. Chapter IV expands on the concept introduced in Chapter III and applies it to two-dimensional heat equations with non-local boundary conditions, offering further numerical examples for clarity and comprehension. Finally, Chapter V concludes the thesis with a comprehensive summary of the findings, a discussion of the research implications and suggestions for possible future avenues of investigation.

CHAPTER II

PRELIMINARIES

In this chapter, we provide background information on the meaning and properties of the Chebyshev polynomials, which are crucial to the FIM-CPE. The form of heat equations with non-local boundary conditions in one- and two-dimensions is given. Heat equations with Robin boundary conditions in one-dimension. Let us first provide the definition and some essential properties of Chebyshev polynomials and some helpful information about them.

2.1 Chebyshev polynomials

Let us start by outlining the fundamental definition of Chebyshev polynomials and some of their beneficial properties. Chebyshev polynomials are set of orthogonal polynomials that are crucial to the theory of approximation; for further information, see [23]. There are several varieties of Chebyshev polynomials as well. However, in this work, we only pay attention to the degree $n \geq 0$ Chebyshev polynomial of the first kind, defined by $T_n(x)$ in Definition 2.1. Figure 2.1 shows the first five Chebyshev polynomials $T_n(x)$ at degree $n \in \{0, 1, 2, 3, 4\}$ for $x \in [-1, 1]$.

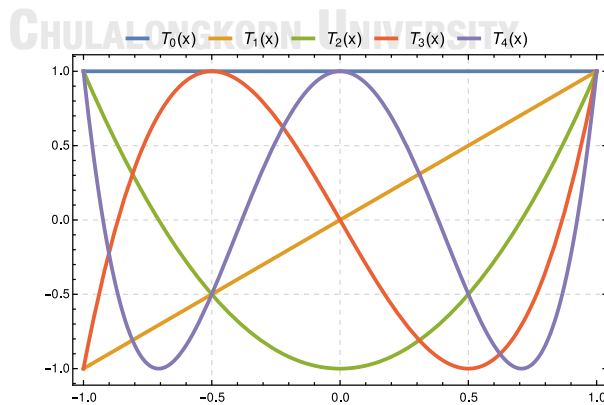


Figure 2.1: Chebyshev polynomials $T_n(x)$ for $n \in \{0, 1, 2, 3, 4\}$.

Usually, the Chebyshev polynomial is defined over $[-1, 1]$. However, in the case of using over the general one-dimensional domain $[a, b]$. The following definition gives the Chebyshev polynomial defined over $[a, b]$ instead.

Definition 2.1 ([3]). The Chebyshev polynomial of degree $n \geq 0$ is defined as

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

However, the formula for the degree $n \geq 0$ Chebyshev polynomial $T_n(x)$ can be defined on the general domain $[a, b]$ by

$$R_n(x) = \cos \left(n \arccos \left(\frac{2x - a - b}{b - a} \right) \right), \quad \text{for } x \in [a, b]. \quad (2.2)$$

Lemma 2.1 ([3]). *The Chebyshev polynomial properties include the followings:*

(i) *The zeros of the Chebyshev polynomial $R_n(x)$ for $x \in [a, b]$ are*

$$x_k = \frac{1}{2} \left((b - a) \cos \left(\frac{2k - 1}{2n} \pi \right) + a + b \right), \quad k \in \{1, 2, 3, \dots, n\}. \quad (2.3)$$

(ii) *The first-order derivatives of $R_n(x)$ at the end point $x \in \{a, b\}$ are*

$$R'_n(x) := \frac{d}{dx} R_n(x) = \frac{2n^2}{b - a} \left(\frac{2x - a - b}{b - a} \right)^{n+1} \quad \text{for } x \in \{a, b\}. \quad (2.4)$$

(iii) *The single-layer integrals of Chebyshev polynomial $R_n(x)$ for $n \geq 2$ is*

$$\bar{R}_n(x) := \int_a^x R_n(\xi) d\xi = \frac{b - a}{4} \left(\frac{R_{n+1}(x)}{n + 1} - \frac{R_{n-1}(x)}{n - 1} - \frac{2(-1)^n}{n^2 - 1} \right), \quad (2.5)$$

where the two initial values are $\bar{R}_0(x) = x - a$ and $\bar{R}_1(x) = \frac{(x-a)(x-b)}{b-a}$. Moreover, the explicit formula of $\bar{R}_n(x)$ at the upper limit $x = b$ is

$$\bar{R}_n(b) := \int_a^b R_n(\xi) d\xi = \frac{b - a}{2} \left(\frac{1 + (-1)^n}{1 - n^2} \right), \quad \text{for } n \geq 2. \quad (2.6)$$

(iv) The discrete orthogonality relationship of Chebyshev polynomials R_i and R_j is

$$\sum_{k=1}^n R_i(x_k)R_j(x_k) = \begin{cases} 0 & \text{if } i \neq j, \\ n & \text{if } i = j = 0, \\ \frac{n}{2} & \text{if } i = j \neq 0, \end{cases}$$

where x_k are the zeroes of $R_n(x)$ defined in (2.2) and $i, j \in \{0, 1, 2, \dots, n\}$.

(v) The Chebyshev matrix \mathbf{R} at each zero x_k as defined in (2.3) is

$$\mathbf{R} = \begin{bmatrix} R_0(x_1) & R_1(x_1) & \cdots & R_{n-1}(x_1) \\ R_0(x_2) & R_1(x_2) & \cdots & R_{n-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_0(x_n) & R_1(x_n) & \cdots & R_{n-1}(x_n) \end{bmatrix}.$$

Then, \mathbf{R} has its multiplicative inverse $\mathbf{R}^{-1} = \frac{1}{n} \text{diag}\{1, 2, 2, \dots, 2\} \mathbf{R}^\top$.

2.2 Kronecker product

The Kronecker product, named after the German mathematician Leopold Kronecker, is a particular kind of tensor product between two matrices. It is used in various fields of mathematics, engineering, physics, and computer science due to its unique properties and the structure it creates [27].

Definition 2.2 ([27]). Let $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$. Then, $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{mp \times nq}$ is the Kronecker product defined by a block matrix as follows:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}.$$

The Kronecker product has a number of important properties, such as compatibility with matrix addition and multiplication, as well as with the transpose and the inverse

operations, given certain conditions. This makes it a valuable tool in many areas. The following theorem is some properties used in this thesis.

Theorem 2.1 ([27]). *The Kronecker product properties include the followings:*

(i) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$. Then,

$$\mathbf{A} \otimes \mathbf{B} = (\mathbf{A} \otimes \mathbf{I}_p)(\mathbf{I}_n \otimes \mathbf{B}) = (\mathbf{I}_m \otimes \mathbf{B})(\mathbf{A} \otimes \mathbf{I}_q).$$

(ii) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{q \times r}$, $\mathbf{C} \in \mathbb{R}^{n \times p}$ and $\mathbf{D} \in \mathbb{R}^{r \times s}$. Then,

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}).$$

(iii) Let $\mathbf{A} \in \mathbb{R}^{m \times m}$, $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $\mathbf{P} := [\mathbf{I}_n \otimes \mathbf{e}_1, \mathbf{I}_n \otimes \mathbf{e}_2, \dots, \mathbf{I}_n \otimes \mathbf{e}_m]$ be an $mn \times mn$ permutation matrix, where $\mathbf{e}_i := [0, \dots, 0, 1, 0, \dots, 0]^\top$ is an m -dimensional column vector which has 1 in the i^{th} positions and 0's elsewhere and \mathbf{I}_n is an $n \times n$ identity matrix. Then, $\mathbf{P}(\mathbf{A} \otimes \mathbf{B})\mathbf{P}^\top = \mathbf{B} \otimes \mathbf{A}$.

Note that the Kronecker product and its properties will be used in section 2.3.2 to change the local numbering system to the global numbering system.

2.3 Chebyshev integration matrices

In this section, our aim is to succinctly outline the application of the FIM-CPE. The primary focus here is the extraction of Chebyshev integration matrices for both one- and two-dimensional contexts. This approach is the fundamental in our quest for solving heat equations with non-local boundary conditions.

2.3.1 FIM-CPE in one-dimension

The Chebyshev integration matrix described in [3], a major tool for handling the integral term, is then built. We assume that $M \in \mathbb{N}$ and we would like to approximate the

solution of heat equations in terms of a function $u(x)$ by using the Chebyshev expansion which can be written as

$$u(x) = \sum_{n=0}^{M-1} c_n R_n(x) \quad \text{for } x \in [a, b], \quad (2.7)$$

where c_n is unknown coefficients. Let x_k for $k \in \{1, 2, 3, \dots, M\}$ be nodal points that are discretized by the zeros of Chebyshev polynomial $R_M(x)$ defined in (2.3). Substituting each x_k into (2.7), the result can be expressed in matrix form

$$\begin{bmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_M) \end{bmatrix} = \begin{bmatrix} R_0(x_1) & R_1(x_1) & \cdots & R_{M-1}(x_1) \\ R_0(x_2) & R_1(x_2) & \cdots & R_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_0(x_M) & R_1(x_M) & \cdots & R_{M-1}(x_M) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{M-1} \end{bmatrix}$$

which is denoted by $\mathbf{u} = \mathbf{R}\mathbf{c}$. Therefore, $\mathbf{c} = \mathbf{R}^{-1}\mathbf{u}$, where \mathbf{R}^{-1} is defined in Lemma 2.1(v). The single-layer integral of u from a to x_k which is indicated as $U^{(1)}(x_k)$ is the next thing that we should look at. That is,

$$U^{(1)}(x_k) = \int_a^{x_k} u(\xi) d\xi = \sum_{n=0}^{M-1} c_n \int_a^{x_k} R_n(\xi) d\xi = \sum_{n=0}^{M-1} c_n \bar{R}_n(x_k)$$

for $k \in \{1, 2, 3, \dots, M\}$, where $\bar{R}_n(x_k)$ is defined by (2.5) in Lemma 2.1 or as a matrix

$$\begin{bmatrix} U^{(1)}(x_1) \\ U^{(1)}(x_2) \\ \vdots \\ U^{(1)}(x_M) \end{bmatrix} = \begin{bmatrix} \bar{R}_0(x_1) & \bar{R}_1(x_1) & \cdots & \bar{R}_{M-1}(x_1) \\ \bar{R}_0(x_2) & \bar{R}_1(x_2) & \cdots & \bar{R}_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \bar{R}_0(x_M) & \bar{R}_1(x_M) & \cdots & \bar{R}_{M-1}(x_M) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{M-1} \end{bmatrix}$$

which is denoted by $\mathbf{U}^{(1)} = \bar{\mathbf{R}}\mathbf{c} = \bar{\mathbf{R}}\mathbf{R}^{-1}\mathbf{u} := \mathbf{A}\mathbf{u}$, where $\mathbf{A} = \bar{\mathbf{R}}\mathbf{R}^{-1} := [a_{ki}]_{M \times M}$ is called the ‘‘Chebyshev integration matrix’’. For $k \in \{1, 2, 3, \dots, M\}$, it takes another form

$$U^{(1)}(x_k) = \int_a^{x_k} u(\xi) d\xi = \sum_{i=1}^M a_{ki} u(x_i), \quad (2.8)$$

where the matrix form can be written as the following

$$\begin{bmatrix} U^{(1)}(x_1) \\ U^{(1)}(x_2) \\ \vdots \\ U^{(1)}(x_M) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MM} \end{bmatrix} \begin{bmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_M) \end{bmatrix}.$$

The method for constructing the Chebyshev integration matrix of the single-layer integral can be applied to the higher-layer integral. Now, we consider the double-layer integral of $u(x)$ from a to the zero x_k , which is denoted by $U^{(2)}(x_k)$. Therefore, we obtain

$$\begin{aligned} U^{(2)}(x_k) &= \int_a^{x_k} \int_a^\eta u(\xi) d\xi d\eta \\ &= \int_a^{x_k} U^{(1)}(\eta) d\eta \\ &= \sum_{i=1}^M a_{ki} U^{(1)}(x_i) \\ &= \sum_{\ell=1}^M \sum_{i=1}^M a_{ki} a_{i\ell} u(x_\ell) \\ &= \sum_{\ell=1}^M [\mathbf{A}^2]_{k\ell} u(x_\ell). \end{aligned}$$

The above equation can be combined and written in the matrix form $\mathbf{U}^{(2)} = \mathbf{A}^2 \mathbf{u}$ which is the matrix representation for the double-layer integral of $u(x)$ in our developed FIM-CPE by varying the zeros x_k for $k \in \{1, 2, 3, \dots, M\}$.

Remark 2.1. Similarly, by using the principle of mathematical induction, we can express the m multiple-layer integral of $u(x)$ from a to the zero x_k as follows:

$$U^{(m)}(x_k) = \int_a^{x_k} \int_a^{\xi_m} \cdots \int_a^{\xi_3} \int_a^{\xi_2} u(\xi_1) d\xi_1 d\xi_2 \cdots d\xi_{m-1} d\xi_m = \sum_{l=1}^M [\mathbf{A}^m]_{kl} u(x_l). \quad (2.9)$$

When the zeros of x_k for $k \in \{1, 2, 3, \dots, M\}$ are distributed in (2.9), which can be expressed in the matrix form $\mathbf{U}^{(m)} = \mathbf{A}^m \mathbf{u}$.

2.3.2 FIM-CPE in two-dimension

Next, we present the process for constructing two-dimensional Chebyshev integration matrices with respect to the variables x and y . Let $M, N \in \mathbb{N}$ and $a, b, c, d \in \mathbb{R}$ such that $a > b$ and $c > d$. Let x_k for $k \in \{1, 2, 3, \dots, M\}$ and y_h for $h \in \{1, 2, 3, \dots, N\}$ be the computational grid points over the domain $\Omega = [a, b] \times [c, d]$. These grid points are defined using the zeros of the Chebyshev polynomials $R_M(x)$ and $R_N(x)$, to discretize the nodes horizontally and vertically, respectively. Consequently, we presume that the total number of grid points equates to MN nodes. To facilitate computation, grid points along the x and y -directions are indexed using both global and local numbering systems, as depicted in Figures 2.2a and 2.2b, respectively.

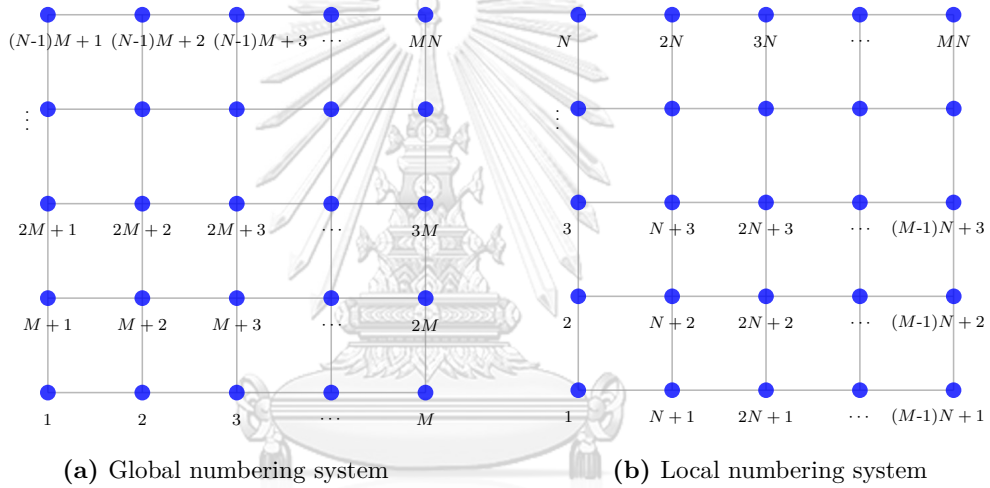


Figure 2.2: The indices of the grid points globally and locally.

Let $U_x^{(1)}(x, y)$ and $U_y^{(1)}(x, y)$ represent the single layer integrations with respect to the variables x and y , respectively. Consider $U_x^{(1)}(x_k, y)$ when y is fixed. To construct the Chebyshev integration matrix with respect to x in two dimensions, we utilize the methodology used for one dimension as follows:

$$U_x^{(1)}(x_k, y) := \int_a^{x_k} u(\xi, y) d\xi = \sum_{i=1}^M a_{ki} u(x_i, y)$$

for $k \in \{1, 2, 3, \dots, M\}$. This can be formulated as a matrix, represented as $\mathbf{U}_x^{(1)}(\cdot, y) = \mathbf{A}_M \mathbf{u}(\cdot, y)$, where $\mathbf{A}_M = \bar{\mathbf{R}}\mathbf{R}^{-1}$ is an $M \times M$ matrix. Then, for each $y \in \{y_1, y_2, y_3, \dots, y_N\}$,

$$\begin{bmatrix} \mathbf{U}_x^{(1)}(\cdot, y_1) \\ \mathbf{U}_x^{(1)}(\cdot, y_2) \\ \vdots \\ \mathbf{U}_x^{(1)}(\cdot, y_N) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_M & 0 & \dots & 0 \\ 0 & \mathbf{A}_M & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{A}_M \end{bmatrix}}_{N \text{ blocks}} \begin{bmatrix} \mathbf{u}(\cdot, y_1) \\ \mathbf{u}(\cdot, y_2) \\ \vdots \\ \mathbf{u}(\cdot, y_N) \end{bmatrix}. \quad (2.10)$$

We denote (2.10) as $\mathbf{U}_x^{(1)} = \mathbf{A}_x \mathbf{u}$, where $\mathbf{A}_x = \mathbf{I}_N \otimes \mathbf{A}_M$ has termed the ‘‘two-dimensional Chebyshev integration matrix with respect to x ’’ in the global numbering system and \otimes is the Kronecker product described in Section 2.2. Next, consider $U_y^{(1)}(x, y_h)$ when x is fixed. Then, we can be expressed in the local numbering system as follows

$$U_y^{(1)}(x, y_h) := \int_c^{y_h} u(x, \eta) d\eta = \sum_{j=1}^N a_{hj} u(x, y_j)$$

for $h \in \{1, 2, 3, \dots, N\}$. It can be expressed as $\mathbf{U}_y^{(1)}(x, \cdot) = \mathbf{A}_N \mathbf{u}(x, \cdot)$. Then, for varying each $x \in \{x_1, x_2, x_3, \dots, x_M\}$, we have

$$\begin{bmatrix} \mathbf{U}_y^{(1)}(x_1, \cdot) \\ \mathbf{U}_y^{(1)}(x_2, \cdot) \\ \vdots \\ \mathbf{U}_y^{(1)}(x_M, \cdot) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_N & 0 & \dots & 0 \\ 0 & \mathbf{A}_N & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{A}_N \end{bmatrix}}_{M \text{ blocks}} \begin{bmatrix} \mathbf{u}(x_1, \cdot) \\ \mathbf{u}(x_2, \cdot) \\ \vdots \\ \mathbf{u}(x_M, \cdot) \end{bmatrix}, \quad (2.11)$$

in which $\mathbf{A}_N = \bar{\mathbf{R}}\mathbf{R}^{-1}$ represents an $N \times N$ matrix. We denote (2.11) as $\tilde{\mathbf{U}}_y^{(1)} = \tilde{\mathbf{A}}_y \tilde{\mathbf{u}}$, where $\tilde{\mathbf{A}}_y = \mathbf{I}_M \otimes \mathbf{A}_N$. It is noteworthy that the elements of \mathbf{u} and $\tilde{\mathbf{u}}$ are identical, although their positions vary between the numbering systems. Hence, we can transition $\tilde{\mathbf{U}}_y^{(1)}$ and $\tilde{\mathbf{u}}$ from the local to the global numbering system using the permutation matrix \mathbf{P} , namely,

$$\mathbf{U}_y^{(1)} = \mathbf{P} \tilde{\mathbf{U}}_y^{(1)} \quad \text{and} \quad \mathbf{u} = \mathbf{P} \tilde{\mathbf{u}}, \quad (2.12)$$

where the permutation matrix \mathbf{P} is an $MN \times MN$ matrix defined by

$$\mathbf{P}_{ij} = \begin{cases} 1 & ; i = (h-1)M + k \text{ and } j = (k-1)N + h, \\ 0 & ; \text{otherwise,} \end{cases}$$

for all $k \in \{1, 2, 3, \dots, M\}$ and $j \in \{1, 2, 3, \dots, N\}$. It is important to note that the permutation matrix \mathbf{P} transforms the i^{th} point in the local numbering system to the j^{th} point in the global numbering system. Furthermore, \mathbf{P} is not only a nonsingular matrix, but it also possesses the multiplicative inverse $\mathbf{P}^{-1} = \mathbf{P}^{\top}$. Therefore, from (2.12), we obtain that $\mathbf{U}_y^{(1)} = \mathbf{A}_y \mathbf{u}$, where $\mathbf{A}_y = \mathbf{P} \tilde{\mathbf{A}}_y \mathbf{P}^{-1} = \mathbf{P}(\mathbf{I}_M \otimes \mathbf{A}_N) \mathbf{P}^{\top} = \mathbf{A}_N \otimes \mathbf{I}_M$ is called the “two-dimensional Chebyshev integration matrix with respect to y ”.

Next, let us consider the double-layer integration of variables along the x and y axes. This thesis uses the global numbering system to communicate the following

$$U_x^{(2)}(x_k, y) := \int_a^{x_k} \int_a^{\xi_2} u(\xi_1, y) d\xi_1 d\xi_2 = \sum_{l=1}^M \sum_{i=1}^M a_{ki} a_{il} u(x_l, y) = \sum_{l=1}^M [\mathbf{A}_M^2]_{kl} u(x_l, y)$$

for $k \in \{1, 2, 3, \dots, M\}$. It can be written into the matrix form $\mathbf{U}_x^{(2)}(\cdot, y) = \mathbf{A}_M^2 \mathbf{u}(\cdot, y)$.

Hence, after substituting each the zero $y \in \{y_1, y_2, y_3, \dots, y_N\}$, we get

$$\begin{bmatrix} \mathbf{U}_x^{(2)}(\cdot, y_1) \\ \mathbf{U}_x^{(2)}(\cdot, y_2) \\ \vdots \\ \mathbf{U}_x^{(2)}(\cdot, y_N) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_M^2 & 0 & \cdots & 0 \\ 0 & \mathbf{A}_M^2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{A}_M^2 \end{bmatrix}}_{N \text{ blocks}} \begin{bmatrix} \mathbf{u}(\cdot, y_1) \\ \mathbf{u}(\cdot, y_2) \\ \vdots \\ \mathbf{u}(\cdot, y_N) \end{bmatrix},$$

which can be represented in matrix notation $\mathbf{U}_x^{(2)} = \mathbf{A}_x^2 \mathbf{u}$, where $\mathbf{A}_x^2 = \mathbf{I}_N \otimes \mathbf{A}_M^2$. Similarly, for the double-layer integration with respect to the variables y , we can operate it with the local numbering system as

$$U_y^{(2)}(x, y_h) := \int_c^{y_h} \int_c^{\eta_2} u(x, \eta_1) d\eta_1 d\eta_2 = \sum_{l=1}^N \sum_{j=1}^N a_{hj} a_{jl} u(x, y_l) = \sum_{l=1}^N [\mathbf{A}_N^2]_{hl} u(x, y_l)$$

for $h \in \{1, 2, 3, \dots, N\}$. It can be written into the matrix form $\mathbf{U}_y^{(2)}(x, \cdot) = \mathbf{A}_N^2 \mathbf{u}(x, \cdot)$.

Hence, after substituting each the zero $x \in \{x_1, x_2, x_3, \dots, x_N\}$, we get

$$\begin{bmatrix} \mathbf{U}_y^{(2)}(x_1, \cdot) \\ \mathbf{U}_y^{(2)}(x_2, \cdot) \\ \vdots \\ \mathbf{U}_y^{(2)}(x_M, \cdot) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_N^2 & 0 & \cdots & 0 \\ 0 & \mathbf{A}_N^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{A}_N^2 \end{bmatrix}}_{M \text{ blocks}} \begin{bmatrix} \mathbf{u}(x_1, \cdot) \\ \mathbf{u}(x_2, \cdot) \\ \vdots \\ \mathbf{u}(x_M, \cdot) \end{bmatrix},$$

which can be represented in matrix notation $\tilde{\mathbf{U}}_y^{(2)} = \tilde{\mathbf{A}}_y^2 \tilde{\mathbf{u}}$, where $\tilde{\mathbf{A}}_y^2 = \mathbf{I}_M \otimes \mathbf{A}_N^2$. However, it should be noted that the node indexing in the local numbering system. Therefore, we can apply the permutation matrix \mathbf{P} mentioned above to transition these nodes to a global system. Consequently, we can represent the double-layer integration in relation to the variable y only, in global numbering, as $\mathbf{U}_y^{(2)} = \mathbf{A}_y^2 \mathbf{u}$. Here, $\mathbf{A}_y^2 = \mathbf{P} \tilde{\mathbf{A}}_y^2 \mathbf{P}^\top = \mathbf{A}_N^2 \otimes \mathbf{I}_M$.

Remark 2.2. In a similar manner, we can formulate the matrix representation for m multiple-layer integration. This is achieved by following the same concept employed in the construction of $\mathbf{U}_x^{(2)}$ and $\mathbf{U}_y^{(2)}$. Hence, for the higher-order Chebyshev integration matrices with respect to only the variables x and y in the global numbering system, we can represent these matrices as follows for any $m \in \mathbb{N}$:

- $\mathbf{U}_x^{(m)} = \mathbf{A}_x^m \mathbf{u}$, where $\mathbf{A}_x^m = \mathbf{I}_N \otimes \mathbf{A}_M^m$;
- $\mathbf{U}_y^{(m)} = \mathbf{A}_y^m \mathbf{u}$, where $\mathbf{A}_y^m = \mathbf{A}_N^m \otimes \mathbf{I}_M$.

For the double-layer integration with respect to the variable x and follow with the variable y , which is denoted by $U_{xy}^{(1,1)}(x_k, y_h)$, we get

$$U_{xy}^{(1,1)}(x_k, y_h) := \int_c^{y_h} \int_a^{x_k} u(\xi, \eta) d\xi d\eta = \sum_{j=1}^N \sum_{i=1}^M a_{hj} a_{ki} u(x_i, y_j) \quad (2.13)$$

for $k \in \{1, 2, 3, \dots, M\}$ and $h \in \{1, 2, 3, \dots, N\}$. Moreover, we can consider (2.13) into two types as follows:

Type I : If y_h is fixed, but x_k is varied, then (2.13) can be written in matrix form as

$$\mathbf{U}_{xy}^{(1,1)}(\cdot, y_h) = \sum_{j=1}^N a_{hj} \mathbf{A}_M \mathbf{u}(\cdot, y_j),$$

where a_{hj} is an element at h^{th} row and j^{th} column of Chebyshev integration matrix \mathbf{A}_N .

By varying all fixed variables $y_h \in \{y_1, y_2, y_3, \dots, y_N\}$, we have

$$\begin{bmatrix} \mathbf{U}_{xy}^{(1,1)}(\cdot, y_1) \\ \mathbf{U}_{xy}^{(1,1)}(\cdot, y_2) \\ \vdots \\ \mathbf{U}_{xy}^{(1,1)}(\cdot, y_N) \end{bmatrix} = \begin{bmatrix} a_{11}\mathbf{I}_M & a_{12}\mathbf{I}_M & \dots & a_{1N}\mathbf{I}_M \\ a_{21}\mathbf{I}_M & a_{22}\mathbf{I}_M & \dots & a_{2N}\mathbf{I}_M \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1}\mathbf{I}_M & a_{N2}\mathbf{I}_M & \dots & a_{NN}\mathbf{I}_M \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{A}_M & 0 & \dots & 0 \\ 0 & \mathbf{A}_M & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{A}_M \end{bmatrix}}_{N \text{ blocks}} \begin{bmatrix} \mathbf{u}(\cdot, y_1) \\ \mathbf{u}(\cdot, y_2) \\ \vdots \\ \mathbf{u}(\cdot, y_N) \end{bmatrix},$$

which is denoted by $\mathbf{U}_{xy}^{(1,1)} = (\mathbf{A}_N \otimes \mathbf{I}_M)(\mathbf{I}_N \otimes \mathbf{A}_M)\mathbf{u} = \mathbf{A}_y \mathbf{A}_x \mathbf{u}$ from Remark 2.2.

Type II : If x_k is fixed, but y_h is varied, then (2.13) can be written in matrix form as

$$\mathbf{U}_{xy}^{(1,1)}(x_k, \cdot) = \sum_{i=1}^M a_{ki} \mathbf{A}_N \mathbf{u}(x_i, \cdot),$$

where a_{ki} is an element at k^{th} row and i^{th} column of Chebyshev integration matrix \mathbf{A}_M .

By varying all fixed variables $x_k \in \{x_1, x_2, x_3, \dots, x_M\}$, we have

$$\begin{bmatrix} \mathbf{U}_{xy}^{(1,1)}(x_1, \cdot) \\ \mathbf{U}_{xy}^{(1,1)}(x_2, \cdot) \\ \vdots \\ \mathbf{U}_{xy}^{(1,1)}(x_M, \cdot) \end{bmatrix} = \begin{bmatrix} a_{11}\mathbf{I}_N & a_{12}\mathbf{I}_N & \dots & a_{1M}\mathbf{I}_N \\ a_{21}\mathbf{I}_N & a_{22}\mathbf{I}_N & \dots & a_{2M}\mathbf{I}_N \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1}\mathbf{I}_N & a_{M2}\mathbf{I}_N & \dots & a_{MM}\mathbf{I}_N \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{A}_N & 0 & \dots & 0 \\ 0 & \mathbf{A}_N & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{A}_N \end{bmatrix}}_{M \text{ blocks}} \begin{bmatrix} \mathbf{u}(x_1, \cdot) \\ \mathbf{u}(x_2, \cdot) \\ \vdots \\ \mathbf{u}(x_M, \cdot) \end{bmatrix},$$

which is denoted by $\tilde{\mathbf{U}}_{xy}^{(1,1)} = (\mathbf{A}_M \otimes \mathbf{I}_N)(\mathbf{I}_M \otimes \mathbf{A}_N)\tilde{\mathbf{u}} = \tilde{\mathbf{A}}_x \tilde{\mathbf{A}}_y \tilde{\mathbf{u}}$, where we currently utilize the local numbering system. However, we can convert it to the global numbering system by employing the previously mentioned permutation matrix \mathbf{P} . Consequently, we obtain that $\mathbf{U}_{xy}^{(1,1)} = \mathbf{P}\tilde{\mathbf{U}}_{xy}^{(1,1)} = \mathbf{P}(\tilde{\mathbf{A}}_x \tilde{\mathbf{A}}_y \tilde{\mathbf{u}}) = \mathbf{P}(\mathbf{P}^{-1}\mathbf{A}_x\mathbf{P})(\mathbf{P}^{-1}\mathbf{A}_y\mathbf{P})(\mathbf{P}^{-1}\mathbf{u}) = \mathbf{A}_x \mathbf{A}_y \mathbf{u}$. Notably, the

matrices \mathbf{A}_x and \mathbf{A}_y are also commutative. Indeed, $\mathbf{A}_y\mathbf{A}_x = (\mathbf{A}_N \otimes \mathbf{I}_M)(\mathbf{I}_N \otimes \mathbf{A}_M) = \mathbf{A}_N \otimes \mathbf{A}_M = (\mathbf{I}_N \otimes \mathbf{A}_M)(\mathbf{A}_N \otimes \mathbf{I}_M) = \mathbf{A}_x\mathbf{A}_y$ according to Theorem 2.1. Thus, we arrive at the matrix representation $\mathbf{U}_{xy}^{(1,1)} = \mathbf{A}_y\mathbf{A}_x\mathbf{u} = \mathbf{A}_x\mathbf{A}_y\mathbf{u}$.

Similarly, by using the mathematical induction, the idea can be applied for the double layer integration with respect to y and followed with x , which is $U_{yx}^{(1,1)}(x, y)$. Thus, we have $\mathbf{U}_{yx}^{(1,1)} = \mathbf{U}_{xy}^{(1,1)} = \mathbf{A}_x\mathbf{A}_y\mathbf{u} = \mathbf{A}_y\mathbf{A}_x\mathbf{u}$.

Remark 2.3. The concept of double-layer integration can be conveniently expanded to accommodate multiple-layer integrations. This leads to the formation of the m^{th} -order and n^{th} -order Chebyshev integration matrices with respect to x and y . Respectively, in the global numbering system. These can be represented in the matrix forms as:

- $\mathbf{U}_{xy}^{(m,n)} = \mathbf{A}_x^m \mathbf{A}_y^n \mathbf{u}$, where $\mathbf{A}_x^m \mathbf{A}_y^n = \mathbf{A}_N^m \otimes \mathbf{A}_M^n$;
- $\mathbf{U}_{yx}^{(n,m)} = \mathbf{A}_y^n \mathbf{A}_x^m \mathbf{u}$, where $\mathbf{A}_y^n \mathbf{A}_x^m = \mathbf{A}_N^n \otimes \mathbf{A}_M^m$.

In this context, \mathbf{A}_x and \mathbf{A}_y denote the first-order Chebyshev integration matrices in relation to the variables x and y , as defined in Remark 2.2.

2.4 Statements of heat equations

Heat equations are PDEs that describe the variation of temperature in a particular region over time. There are some fundamental concept in conductive heat transfer, which is derived from the law of conservation of energy. Moreover, these equations are an essential subject in many fields of science and engineering, especially involving non-local boundary conditions. They are used in a wide range of real-world scenarios for example:

- In materials science [1], they can be used to understand and predict how heat treatment will affect the properties of a material.
- In environmental science [2], they can model how heat spreads in various media, such as the ground, the ocean or the atmosphere. This can help predict the temperature changes in soil or bodies of water over time.

- In electronic engineering [18], heat equations can be used to manage and predict the heat distribution in electronic systems and components, helping to improve the design of heat sinks and cooling systems.
- In electronic engineering [22], the heat equation can be used to model heat transfer within tissues, which is critical in applications such as hyperthermia treatment or the design of prosthetics.

This kind of problem is often encountered in complex physical systems, where the state at a particular location is affected by the states of other locations, or where the heat flux at a boundary is not only related to the local temperature gradient, but also depends on the distribution of temperature throughout the domain. Therefore, these problems culminate in our primary challenge in this study.

2.4.1 One-dimensional heat equation

In this study, our attention is drawn towards the one-dimensional heat equation with non-local and Robin boundary conditions coupled with forcing terms. This equation serves as a mathematical model to depict the spread of heat across a one-dimensional medium under the effect of external factors. Note that, the heat equation that we consider in this thesis is the one with constant coefficients. Thus, we assume that all coefficients are scaled to 1. Therefore, the considered heat equation can be expressed in the following form:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + q(x, t), \quad (x, t) \in (a, b) \times (0, T], \quad (2.14)$$

where $a, b, T \in \mathbb{R}$ such that $a < b$ and $T > 0$. The function $u(x, t)$ is the temperature at position x and time t . This is the unknown function that we would like to find. The function $q(x, t)$ represents a heat source within the material and we generally assume it is a known, smooth function of x and t . The considered problem (2.14) is subject to the

initial condition:

$$u(x, 0) = f(x), \quad x \in [a, b], \quad (2.15)$$

where $f(x)$ is a given continuous function at the initial time.

We consider two non-classical boundary conditions that are introduced in [25, 28].

The first one is in the integration forms which are the non-local boundary conditions:

$$u(a, t) = \int_a^b k_1(x, t)u(x, t) dx + g_1(t), \quad t \in (0, T], \quad (2.16)$$

$$u(b, t) = \int_a^b k_2(x, t)u(x, t) dx + g_2(t), \quad t \in (0, T], \quad (2.17)$$

where k_1, k_2, g_1 and g_2 and are known functions, while u is an unknown solution to be determined. The functions $k_1(x, t)$ and $k_2(x, t)$ are the weight functions for the boundary condition, which we typically assume to be integrable over the interval (a, b) and possibly time-dependent. The functions $g_1(t)$ and $g_2(t)$ represent heat sources at the boundary which is a smooth function of t . These non-local boundary conditions (2.16) and (2.17) usually emerge in the investigation of a thermoelastic bar's quasi-static flexure where they represent the weighted average of the entropy $u(x, t)$ given in [28].

The second one is in the differential forms together with the function value itself at the boundary point known as Robin boundary conditions:

$$\alpha_1(t) \left. \frac{\partial u(x, t)}{\partial x} \right|_{x=a} + \beta_1(t)u(a, t) = h_1(t), \quad t \in (0, T], \quad (2.18)$$

$$\alpha_2(t) \left. \frac{\partial u(x, t)}{\partial x} \right|_{x=b} + \beta_2(t)u(b, t) = h_2(t), \quad t \in (0, T], \quad (2.19)$$

where $\alpha_1, \alpha_2, \beta_1, \beta_2, h_1$ and h_2 are known continuous functions, while u is an unknown solution to be determined. The functions $\alpha_1(t)$ and $\alpha_2(t)$ be related to the area or a characteristic length. The functions $\beta_1(t)$ and $\beta_2(t)$ represent convection heat transfer coefficient. The functions $h_1(t)$ and $h_2(t)$ represent the heat flux at the boundary and are generally assumed to be a known, continuous function of time t . Note that if

$\beta_1(t), h_1(t), \beta_2(t)$ and $h_2(t)$ are all zero, then (2.18) and (2.19) correspond to the heat conduction problem with insulated at both ends. Moreover, these Robin boundary conditions (2.18) and (2.19) are actually the local boundaries, however, it is in terms of Robin boundary conditions that can be seen as a heat transfer process where the temperature distribution is represented by $u(x, t)$.

2.4.2 Two-dimensional heat equation

The two-dimensional heat equation is a PDE that describes the distribution of heat in a given region (in two directions) over time and over a rectangle domain $\Omega = [a, b] \times [c, d]$. Note that, the heat equation that we consider in this thesis is the one with constant coefficients. Thus, we assume that all coefficients are scaled to 1. In mathematical form, it is usually represented as follows:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + q(x, y, t), \quad (x, y, t) \in (a, b) \times (c, d) \times (0, T], \quad (2.20)$$

where $a, b, c, d, T \in \mathbb{R}$ such that $a < b$, $c < d$ and $T > 0$. The considered problem (2.20) is an extension of (2.14), augmented by the inclusion of the variable y , representing the distance along the secondary axis of heat conduction. The other parameters in (2.20) can be understood in the same way as in (2.14), namely, $u(x, y, t)$ is the temperature at position (x, y) and time t that we wish to find. $q(x, y, t)$ is a given smooth function of x , y and t representing a heat source within the material. The problem in focus is subject to a specific initial condition given by

$$u(x, y, 0) = f(x, y), \quad (x, y) \in [a, b] \times [c, d] \quad (2.21)$$

with the non-local boundary specifications described in [24] given by

$$u(a, y, t) = \int_c^d \int_a^b k_1(a, y, \xi, \eta) u(\xi, \eta, t) d\xi d\eta, \quad y \in [c, d], \quad (2.22)$$

$$u(b, y, t) = \int_c^d \int_a^b k_2(b, y, \xi, \eta) u(\xi, \eta, t) d\xi d\eta, \quad y \in [c, d], \quad (2.23)$$

$$u(x, c, t) = \int_c^d \int_a^b k_3(x, c, \xi, \eta) u(\xi, \eta, t) d\xi d\eta, \quad x \in [a, b], \quad (2.24)$$

$$u(x, d, t) = \int_c^d \int_a^b k_4(x, d, \xi, \eta) u(\xi, \eta, t) d\xi d\eta, \quad x \in [a, b], \quad (2.25)$$

for $t \in (0, T]$. The function $f(x, y)$ is a given continuous function and $k(x, y)$ is the weight function for the boundary conditions, which we typically assume to be integrable over the domain $(a, b) \times (c, d)$. Comprehensive studies on the existence, uniqueness and data-dependent continuity of solutions to this problem, along with related issues, have been conducted and are documented in [8, 9, 21] and the references included therein.

CHAPTER III

ONE-DIMENSIONAL HEAT EQUATION

The focus of this chapter is the formulation of numerical algorithms designed to solve the one-dimensional heat equation (2.14). This equation includes an initial condition (2.15) and takes into account two types of unconventional boundaries - specifically, non-local boundary conditions (2.16)–(2.17) and Robin boundary conditions (2.18)–(2.19). Our algorithms utilize the FIM-CPE for handling the spatial domain, while implementing the Crank-Nicolson method for the temporal variable. Furthermore, we supply numerical examples to evaluate the effectiveness and precision of the algorithms we propose.

3.1 Formulation of one-dimension heat equation

Within this section, we will recast the one-dimensional heat equation (2.14) into matrix form, utilizing both the FIM-CPE and the Crank-Nicolson method. To initiate this process, let us revisit the problems we are addressing, which are outlined below:

$$\frac{\partial u(x, t)}{\partial t} = \frac{\partial^2 u(x, t)}{\partial x^2} + q(x, t), \quad (x, t) \in (a, b) \times (0, T]. \quad (3.1)$$

The proposed numerical algorithms are based on the FIM-CPE in Section 2.3.1 to deal with the spatial variable x and use the Crank-Nicolson method to handle the temporal variable t . First, let us consider the heat equation (3.1), we start by uniformly discretizing the temporal domain $(0, T]$ by specifying each time point $t_m = m\Delta t$ for $m \in \mathbb{N}$ into (3.1), where Δt is a given time step. Then, we have

$$\left. \frac{\partial u(x, t)}{\partial t} \right|_{t=t_m} = \frac{\partial^2 u(x, t_m)}{\partial x^2} + q(x, t_m) := G(x, t_m, u(x, t_m)), \quad (3.2)$$

where $G(x, t, u)$ represents the right-hand-side term of (3.1).

Next, we can see that (3.2) has a derivative with respect to time t . We then approximate it by employing the Crank-Nicolson method which provides the time complexity of $\mathcal{O}(\Delta t^2)$. We can also use the difference quotients to approximate the time derivative, but it results in less accuracy and cannot guarantee stability compared to the Crank-Nicolson method. The idea of the Crank-Nicolson method is to approximate a solution by its average between backward and forward times. Then, (3.2) can be estimated in the form

$$\frac{u(x, t_m) - u(x, t_{m-1})}{\Delta t} = \frac{G(x, t_{m-1}, u(x, t_{m-1})) + G(x, t_m, u(x, t_m))}{2}, \quad (3.3)$$

where the functions G at times t_m and t_{m-1} are respectively defined by

$$\begin{aligned} G(x, t_m, u(x, t_m)) &= \frac{\partial^2 u(x, t_m)}{\partial x^2} + q(x, t_m) \quad \text{and} \\ G(x, t_{m-1}, u(x, t_{m-1})) &= \frac{\partial^2 u(x, t_{m-1})}{\partial x^2} + q(x, t_{m-1}). \end{aligned}$$

Let a function u with superscript $\langle m \rangle$ be that function u indicated at time t_m . When the functions G at times t_m and t_{m-1} are plugged in (3.3), we have

$$u^{\langle m \rangle}(x) - u^{\langle m-1 \rangle}(x) = \frac{\Delta t}{2} \left(\frac{\partial^2 u^{\langle m-1 \rangle}(x)}{\partial x^2} + q^{\langle m-1 \rangle}(x) + \frac{\partial^2 u^{\langle m \rangle}(x)}{\partial x^2} + q^{\langle m \rangle}(x) \right). \quad (3.4)$$

Now, the considered problem (3.4) is depending on the spatial variable x only. Hence, the FIM-CPE can be applied to solve the problem which approximates the solution $u(x, t)$ by the Chebyshev expansion (2.7). By the idea of FIM-CPE, we eliminate derivatives with respect to x from (3.4) by taking the double-layer integrals from a to x_k defined in (2.3) on both sides of (3.4). Thus, we obtain the equivalent integral equation

$$\begin{aligned} & \int_a^{x_k} \int_a^\eta u^{\langle m \rangle}(\xi) d\xi d\eta - \int_a^{x_k} \int_a^\eta u^{\langle m-1 \rangle}(\xi) d\xi d\eta \\ &= \frac{\Delta t}{2} \left(u^{\langle m-1 \rangle}(x_k) + \int_a^{x_k} \int_a^\eta q^{\langle m-1 \rangle}(\xi) d\xi d\eta \right) \\ &+ \frac{\Delta t}{2} \left(u^{\langle m \rangle}(x_k) + \int_a^{x_k} \int_a^\eta q^{\langle m \rangle}(\xi) d\xi d\eta \right) + h_1 x_k + h_2, \end{aligned} \quad (3.5)$$

where h_1 and h_2 are arbitrary constants that emerged from the process of integration.

Next, by substituting each zero $x_k \in \{x_1, x_2, x_3, \dots, x_M\}$ into the integral equation (3.5), we can express them into the matrix form as follows

$$\mathbf{A}^2 \mathbf{u}^{(m)} - \mathbf{A}^2 \mathbf{u}^{(m-1)} = \frac{\Delta t}{2} \left(\mathbf{u}^{(m)} + \mathbf{u}^{(m-1)} + \mathbf{A}^2 \mathbf{q}^{(m)} + \mathbf{A}^2 \mathbf{q}^{(m-1)} \right) + h_1 \mathbf{x} + h_2 \mathbf{e}$$

which can be simplified to

$$\left(\mathbf{A}^2 - \frac{\Delta t}{2} \mathbf{I} \right) \mathbf{u}^{(m)} - h_1 \mathbf{x} - h_2 \mathbf{e} = \left(\mathbf{A}^2 + \frac{\Delta t}{2} \mathbf{I} \right) \mathbf{u}^{(m-1)} + \frac{\Delta t}{2} \mathbf{A}^2 \left(\mathbf{q}^{(m)} + \mathbf{q}^{(m-1)} \right), \quad (3.6)$$

where \mathbf{I} is an $M \times M$ identity matrix, $\mathbf{A} = \bar{\mathbf{R}}\mathbf{R}^{-1}$ is the Chebyshev integration matrix as described in Section 2.3.1, $\mathbf{e} = [1, 1, 1, \dots, 1]^\top$ is the all-ones vector with size $M \times 1$,

$$\begin{aligned} \mathbf{x} &= [x_1, x_2, x_3, \dots, x_M]^\top, \\ \mathbf{u}^{(m)} &= [u(x_1, t_m), u(x_2, t_m), u(x_3, t_m), \dots, u(x_M, t_m)]^\top, \\ \mathbf{q}^{(m)} &= [q(x_1, t_m), q(x_2, t_m), q(x_3, t_m), \dots, q(x_M, t_m)]^\top. \end{aligned}$$

Observing (3.6), it is evident that there are two variables, h_1 and h_2 , that remain unknown. As such, we require two additional equations to form a complete system. These can be created based on the given boundary conditions. In our study, we focus on two types: non-local and Robin boundary conditions. These will be elaborated upon in Sections 3.2 and 3.3, respectively, facilitating the construction of the numerical algorithms.

3.2 Algorithm for non-local boundary conditions

In this section, we formulate and illustrate the numerical procedure for solving the heat equation (3.1), conforming to the non-local boundary conditions (2.16) and (2.17), which are introduced in [25, 28]. They are in the following integration forms as

$$u(a, t) = \int_a^b k_1(x, t) u(x, t) dx + g_1(t), \quad t \in (0, T], \quad (3.7)$$

$$u(b, t) = \int_a^b k_2(x, t) u(x, t) dx + g_2(t), \quad t \in (0, T]. \quad (3.8)$$

3.2.1 Numerical algorithm

Consider the non-local boundary conditions (3.7) and (3.8). They are in the integration forms. We can rewrite them at time t_m into general form for both the left and right boundary conditions as follows

$$u(\gamma_j, t_m) = \int_a^b k_j(x, t_m)u(x, t_m) dx + g_j(t_m), \quad j \in \{1, 2\}, \quad (3.9)$$

where $\gamma_1 = a$ and $\gamma_2 = b$. Note that for $j = 1$ in (3.9), it becomes the left boundary condition, conversely, the right boundary condition when $j = 2$. Next, we transform (3.9) at time t_m into the vector form by using the Chebyshev polynomial expansion (2.7). Consider the left-hand-side term of (3.9), we have

$$u(\gamma_j, t_m) = u^{(m)}(\gamma_j) = \sum_{n=0}^{M-1} c_n^{(m)} R_n(\gamma_j) := \mathbf{R}_j \mathbf{c}^{(m)} = \mathbf{R}_j \mathbf{R}^{-1} \mathbf{u}^{(m)}, \quad (3.10)$$

where $\mathbf{R}_j = [R_0(\gamma_j), R_1(\gamma_j), R_2(\gamma_j), \dots, R_{M-1}(\gamma_j)]$ and \mathbf{R}^{-1} is defined in Lemma 2.1(v).

We next turn our attention to the integral term of (3.9). To treat this, we adopt the Chebyshev polynomial expansion (2.7). Hence, we obtain

$$k_j(x, t_m)u(x, t_m) = \sum_{n=0}^{M-1} p_n^{(m)} R_n(x), \quad (3.11)$$

where $p_n^{(m)}$ is unknown coefficients. Let x_k , where $k \in \{1, 2, 3, \dots, M\}$, signify nodal points which are discretized via the zeros of the Chebyshev polynomial $R_M(x)$ as defined in (2.3). Inserting each x_k into (3.11) permits the outcome to be cast in matrix form.

$$\begin{bmatrix} k_j(x_1, t_m)u(x_1, t_m) \\ k_j(x_2, t_m)u(x_2, t_m) \\ \vdots \\ k_j(x_M, t_m)u(x_M, t_m) \end{bmatrix} = \begin{bmatrix} R_0(x_1) & R_1(x_1) & \cdots & R_{M-1}(x_1) \\ R_0(x_2) & R_1(x_2) & \cdots & R_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_0(x_M) & R_1(x_M) & \cdots & R_{M-1}(x_M) \end{bmatrix} \begin{bmatrix} p_0^{(m)} \\ p_1^{(m)} \\ \vdots \\ p_{M-1}^{(m)} \end{bmatrix} := \mathbf{R} \mathbf{p}^{(m)}. \quad (3.12)$$

Indeed, the left-hand side of equation (3.12) can be expressed in an alternate form

$$\begin{bmatrix} k_j(x_1, t_m)u(x_1, t_m) \\ k_j(x_2, t_m)u(x_2, t_m) \\ \vdots \\ k_j(x_M, t_m)u(x_M, t_m) \end{bmatrix} = \begin{bmatrix} k_j^{(m)}(x_1) & 0 & \dots & 0 \\ 0 & k_j^{(m)}(x_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & k_j^{(m)}(x_M) \end{bmatrix} \begin{bmatrix} u^{(m)}(x_1) \\ u^{(m)}(x_2) \\ \vdots \\ u^{(m)}(x_M) \end{bmatrix} := \mathbf{K}_j^{(m)} \mathbf{u}^{(m)}. \quad (3.13)$$

Thus, we have that (3.12) is equal to (3.13), namely,

$$\mathbf{K}_j^{(m)} \mathbf{u}^{(m)} = \mathbf{R} \mathbf{p}^{(m)} \quad \text{or} \quad \mathbf{p}^{(m)} = \mathbf{R}^{-1} \mathbf{K}_j^{(m)} \mathbf{u}^{(m)}, \quad (3.14)$$

where $\mathbf{K}_j^{(m)} = \text{diag} \{k_j(x_1, t_m), k_j(x_2, t_m), k_j(x_3, t_m), \dots, k_j(x_M, t_m)\}$ and \mathbf{R}^{-1} is defined in Lemma 2.1(v). Then, for the single-layer integral term appearing in (3.9), we use the relationship outlined in (3.14) to convert it into the matrix form, which can be as the following

$$\begin{aligned} \int_a^b k_j(x, t_m)u(x, t_m) dx &= \sum_{n=0}^{M-1} p_n^{(m)} \int_a^b R_n(\xi) d\xi \\ &= \sum_{n=0}^{M-1} p_n^{(m)} \bar{R}_n(b) \\ &= \bar{\mathbf{R}}(b) \mathbf{p}^{(m)} \\ &= \bar{\mathbf{R}}(b) \mathbf{R}^{-1} \mathbf{K}_j^{(m)} \mathbf{u}^{(m)}, \end{aligned} \quad (3.15)$$

where $\bar{\mathbf{R}}(b) = [\bar{R}_0(b), \bar{R}_1(b), \bar{R}_2(b), \dots, \bar{R}_{M-1}(b)]$, in which $\bar{R}_n(b)$ is defined in (2.6).

Thus, by substituting (3.10) and (3.15) into (3.9), we have

$$\mathbf{R}_j \mathbf{R}^{-1} \mathbf{u}^{(m)} = \bar{\mathbf{R}}(b) \mathbf{R}^{-1} \mathbf{K}_j^{(m)} \mathbf{u}^{(m)} + g_j(t_m).$$

Thus, for $j \in \{1, 2\}$, we obtain the following left and right boundary conditions as

$$\left(\mathbf{R}_1 \mathbf{R}^{-1} - \bar{\mathbf{R}}(b) \mathbf{R}^{-1} \mathbf{K}_1^{(m)} \right) \mathbf{u}^{(m)} = g_1(t_m), \quad (3.16)$$

$$\left(\mathbf{R}_2 \mathbf{R}^{-1} - \bar{\mathbf{R}}(b) \mathbf{R}^{-1} \mathbf{K}_2^{(m)} \right) \mathbf{u}^{(m)} = g_2(t_m), \quad (3.17)$$

where

$$\begin{aligned} \mathbf{R}_1 &= [R_0(a), R_1(a), R_2(a), \dots, R_{M-1}(a)] = [1, -1, 1, \dots, (-1)^{M-1}], \\ \mathbf{R}_2 &= [R_0(b), R_1(b), R_2(b), \dots, R_{M-1}(b)] = [1, 1, 1, \dots, 1] \text{ has } M \text{ terms,} \\ \mathbf{K}_1^{(m)} &= \text{diag} \{k_1(x_1, t_m), k_1(x_2, t_m), k_1(x_3, t_m), \dots, k_1(x_M, t_m)\} \text{ and} \\ \mathbf{K}_2^{(m)} &= \text{diag} \{k_2(x_1, t_m), k_2(x_2, t_m), k_2(x_3, t_m), \dots, k_2(x_M, t_m)\}. \end{aligned}$$

Now, we completely obtain all of the equations for solving $\mathbf{u}^{(m)}$. Consequently, we can combine (3.6) with the boundary conditions (3.16) and (3.17) to construct the system of linear equations. Therefore, we have a linear system of $M + 2$ unknowns, namely $\mathbf{u}^{(m)}$, h_1 and h_2 , depending on the type of non-local boundary conditions as follows:

$$\begin{bmatrix} \mathbf{A}^2 - \frac{\Delta t}{2} \mathbf{I} & -\mathbf{x} & -\mathbf{e} \\ \mathbf{R}_1 \mathbf{R}^{-1} - \bar{\mathbf{R}}(b) \mathbf{R}^{-1} \mathbf{K}_1^{(m)} & 0 & 0 \\ \mathbf{R}_2 \mathbf{R}^{-1} - \bar{\mathbf{R}}(b) \mathbf{R}^{-1} \mathbf{K}_2^{(m)} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(m)} \\ h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} \mathbf{z}^{(m)} \\ g_1(t_m) \\ g_2(t_m) \end{bmatrix}, \quad (3.18)$$

where $[\mathbf{z}^{(m)}]_{M \times 1} := \left(\mathbf{A}^2 + \frac{\Delta t}{2} \mathbf{I} \right) \mathbf{u}^{(m-1)} + \frac{\Delta t}{2} \mathbf{A}^2 (\mathbf{q}^{(m)} + \mathbf{q}^{(m-1)})$. Accordingly, the solution $\mathbf{u}^{(m)}$ can be approximated by solving (3.18) together with (2.15) that starts from the given initial condition $\mathbf{u}^{(0)} = [f(x_1), f(x_2), f(x_3), \dots, f(x_M)]^\top$. Note that, performing the final iteration, the obtained solution $\mathbf{u}^{(m)} = [u(x_1, T), u(x_2, T), u(x_3, T), \dots, u(x_M, T)]^\top$ can be actually expressed corresponding to the function $u(x, T)$, $x \in [a, b]$, by using (2.7), i.e.,

$$u(x, T) = \sum_{n=0}^{M-1} c_n^{(m)} R_n(x) = \mathbf{R}(x) \mathbf{c}^{(m)} = \mathbf{R}(x) \mathbf{R}^{-1} \mathbf{u}^{(m)}, \quad (3.19)$$

where $\mathbf{R}(x) = [R_0(x), R_1(x), R_2(x), \dots, R_{M-1}(x)]$.

For computational convenience, we summarize all the procedures mentioned above into the pseudocode algorithms for finding approximate solutions of the heat equation with non-local boundary by using FIM-CPE combined with the Crank-Nicolson method.

Algorithm 1 One-dimensional heat equation with non-local boundary conditions

Input: $a, b, M, T, \Delta t, q(x, t), f(x), g_1(t), g_2(t), k_1(x, t)$ and $k_2(x, t)$;

Output: The approximate solution $\mathbf{u}^{(m)}$;

- 1: Set $x_k \leftarrow \frac{1}{2} \left((b - a) \cos \left(\frac{2k-1}{2n} \pi \right) + a + b \right)$ for $k \in \{1, 2, 3, \dots, M\}$ in ascending order;
 - 2: Compute $\mathbf{x}, \mathbf{e}, \mathbf{I}, \mathbf{A}, \mathbf{R}, \mathbf{R}_1, \mathbf{R}_2, \bar{\mathbf{R}}, \mathbf{R}^{-1}$ and $\bar{\mathbf{R}}(b)$;
 - 3: Construct $\mathbf{u}^{(0)} \leftarrow [f(x_1), f(x_2), f(x_3), \dots, f(x_M)]^\top$;
 - 4: Set $m \leftarrow 1$ and $t_1 \leftarrow \Delta t$;
 - 5: **while** $t_m \leq T$ **do**
 - 6: Compute $\mathbf{K}_1^{(m)} \leftarrow \text{diag} \{k_1(x_1, t_m), k_1(x_2, t_m), k_1(x_3, t_m), \dots, k_1(x_M, t_m)\}$;
 - 7: Compute $\mathbf{K}_2^{(m)} \leftarrow \text{diag} \{k_2(x_1, t_m), k_2(x_2, t_m), k_2(x_3, t_m), \dots, k_2(x_M, t_m)\}$;
 - 8: Compute $\mathbf{q}^{(m-1)} \leftarrow [q(x_1, t_m - \Delta t), q(x_2, t_m - \Delta t), \dots, q(x_M, t_m - \Delta t)]^\top$;
 - 9: Compute $\mathbf{q}^{(m)} \leftarrow [q(x_1, t_m), q(x_2, t_m), q(x_3, t_m), \dots, q(x_M, t_m)]^\top$;
 - 10: Compute $\mathbf{z}^{(m)} \leftarrow (\mathbf{A}^2 + \frac{\Delta t}{2} \mathbf{I}) \mathbf{u}^{(m-1)} + \frac{\Delta t}{2} \mathbf{A}^2 (\mathbf{q}^{(m)} + \mathbf{q}^{(m-1)})$;
 - 11: Find $\mathbf{u}^{(m)}$ by solving the iterative linear system (3.18);
 - 12: Update $m \leftarrow m + 1$ and $t_m \leftarrow m \Delta t$;
 - 13: **end while**
 - 14: **return** The final iteration of $\mathbf{u}^{(m)}$ is the approximate solution;
-

3.2.2 Numerical examples

In this section, we demonstrate the accuracy and efficiency of the proposed numerical Algorithms 1 in order to find the approximate solutions of the heat equations with non-local boundary conditions via Examples 3.1 and 3.2. For measuring the accuracy of the obtained solution, we use the average relative error (ARE) and the maximal absolute error (MAE) which are respectively defined by

$$\text{ARE} = \frac{1}{M} \sum_{k=1}^M \left| \frac{u^*(x_k, T) - u(x_k, T)}{u^*(x_k, T)} \right| \quad \text{and} \quad \text{MAE} = \max_{1 \leq k \leq M} |u^*(x_k, T) - u(x_k, T)|,$$

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

Example 3.1 ([28]). We consider the heat equation without heat source

$$\frac{\partial u(x, t)}{\partial t} = \frac{\partial^2 u(x, t)}{\partial x^2}, \quad (x, t) \in (0, 1) \times (0, 1],$$

subject to the initial condition

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

with non-local boundary conditions

$$u(0, t) = \int_0^1 (x + t)u(x, t) dx + \frac{1}{2} + e^{-t} - t - e^{-t} (\cos(1) + \sin(1) + t \sin(1) - 1) \text{ and}$$

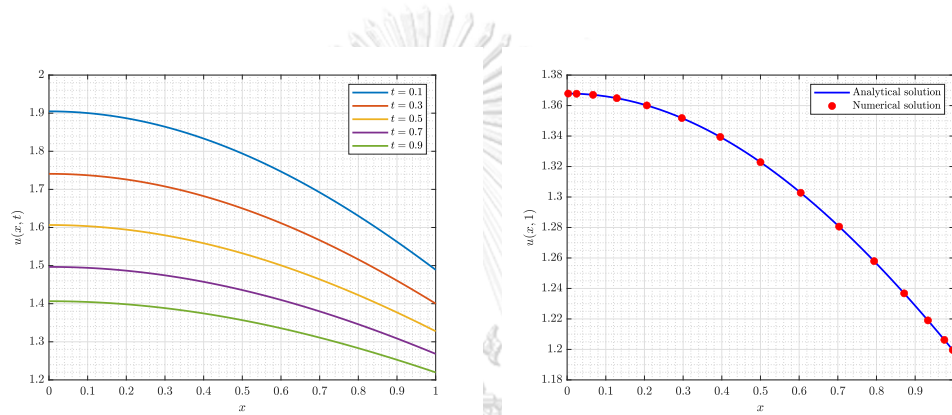
$$u(1, t) = \int_0^1 (te^{-t})u(x, t) dx + 1 + e^{-t} \cos(1) - \frac{t}{2e} (2e - 2 + e^{-t}(\sin(1) - \cos(1) + e)),$$

for $t \in (0, 1]$. The analytical solution is $u^*(x, t) = 1 + e^{-t} \cos(x)$.

This problem is considered under the non-local boundary conditions. Thus, we can find its approximate solutions using our numerical Algorithm 1. When Algorithm 1 is performed, we have the numerical solutions $\mathbf{u}^{(m)}$. Then, their accuracy of the obtained solutions at time $t = 1$ measured by ARE are shown in Table 3.1 for $M \in \{5, 10, 15, 20, 25\}$ and $\Delta t \in \{0.1, 0.05, 0.01, 0.005\}$. This table shows that the errors slightly decrease as the nodal number M increases, but they get significantly smaller as Δt decreases. Finally, Figure 3.1 displays the graphical solutions solved by Algorithm 1 at various $t \in (0, 1)$ and the obtained solutions compared to the exact solution with $M = 15$ and $t = 1$.

Table 3.1: The ARE of the numerical solutions $u(x, 1)$ for Example 3.1.

M	$\Delta t = 0.1$	$\Delta t = 0.05$	$\Delta t = 0.01$	$\Delta t = 0.005$
5	1.3437×10^{-4}	3.3601×10^{-5}	1.6162×10^{-6}	1.2665×10^{-6}
10	1.3406×10^{-4}	3.3323×10^{-5}	1.3303×10^{-6}	1.3303×10^{-8}
15	1.3406×10^{-4}	3.3321×10^{-5}	1.3303×10^{-6}	1.3303×10^{-8}
20	1.3406×10^{-4}	3.3321×10^{-5}	1.3303×10^{-6}	1.3301×10^{-8}
25	1.3406×10^{-4}	3.3321×10^{-5}	1.3303×10^{-6}	1.3302×10^{-8}

**(a)** Numerical solutions at various $t \in (0, 1)$ **(b)** Compared solutions with $M = 15$ and $t = 1$ **Figure 3.1:** Graphical numerical solutions $u(x, t)$ from Algorithm 1 for Example 3.1.

Example 3.2 ([17]). We consider the heat equation with heat source

$$\frac{\partial u(x, t)}{\partial t} = \frac{\partial^2 u(x, t)}{\partial x^2} + e^{-t} \left(x(x-1) + \frac{\delta^2}{6(1+\delta^2)} + 2 \right), \quad (x, t) \in (0, 1) \times (0, 1],$$

subject to the initial condition

$$u(x, 0) = x(x-1) + \frac{\delta^2}{6(1+\delta^2)}, \quad x \in [0, 1],$$

with non-local boundary conditions

$$u(0, t) = -\delta^2 \int_0^1 u(x, t) dx, \quad t \in (0, 1],$$

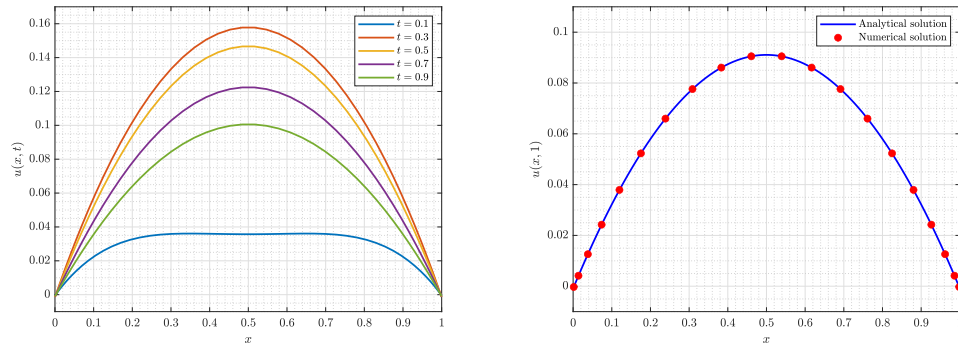
$$u(1, t) = -\delta^2 \int_0^1 u(x, t) dx, \quad t \in (0, 1],$$

where $\delta = 0.12$ and the boundary kernels, $k_0 = k_1 = -\delta^2$, are taken from an example in Day [9]. The analytical solution is $u^*(x, t) = -e^{-t} \left(x(x-1) + \frac{\delta^2}{6(1+\delta^2)} \right)$.

The problem in consideration here is associated with non-local boundary conditions, thus permitting the derivation of approximate solutions through our numerical Algorithm 1. Upon execution of Algorithm 1, we procure the numerical solutions denoted by $\mathbf{u}^{(m)}$. The accuracy of the solutions achieved at time $t = 1$ is subsequently quantified using MAE, with results showcased in Table 3.2 for various values of $M \in \{10, 15, 20, 25, 30\}$ and $\Delta t \in \{0.1, 0.05, 0.01, 0.005\}$. An observation from the table reveals that errors marginally reduce with an increase in the nodal number M , whereas they decrease considerably as Δt diminishes. Lastly, Figure 3.2 presents the graphical illustrations of solutions computed by Algorithm 1 at various instances $t \in (0, 1)$ and a comparative visualization of the obtained solutions versus the exact solution when $M = 20$ and $t = 1$.

Table 3.2: The MAE of the numerical solutions $u(x, 1)$ for Example 3.2.

M	$\Delta t = 0.1$	$\Delta t = 0.05$	$\Delta t = 0.01$	$\Delta t = 0.005$
10	0.0021	1.8958×10^{-4}	2.5056×10^{-5}	2.5055×10^{-5}
15	0.0020	5.2492×10^{-4}	2.3437×10^{-5}	2.5263×10^{-5}
20	0.0021	4.4978×10^{-4}	2.4389×10^{-5}	2.5335×10^{-5}
25	0.0022	5.2424×10^{-4}	3.1606×10^{-5}	2.5605×10^{-5}
30	0.0021	4.9161×10^{-4}	2.6435×10^{-5}	2.4979×10^{-5}



(a) Numerical solutions in (3.19) at various $t \in (0, 1)$ (b) Compared solutions with $M = 20$ and $t = 1$

Figure 3.2: Graphical numerical solutions $u(x, t)$ in (3.19) from Algorithm 1 for Example 3.2.

3.3 Algorithm for Robin boundary conditions

In this section, we present the formulation and illustration of a numerical procedure designed to solve the one-dimensional heat equation (3.1), in accordance with the Robin boundary conditions (2.18) and (2.19). These conditions can be expressed in the subsequent integral forms:

$$\alpha_1(t) \left. \frac{\partial u(x, t)}{\partial x} \right|_{x=a} + \beta_1(t) u(a, t) = g_1(t), \quad t \in (0, T], \quad (3.20)$$

$$\alpha_2(t) \left. \frac{\partial u(x, t)}{\partial x} \right|_{x=b} + \beta_2(t) u(b, t) = g_2(t), \quad t \in (0, T]. \quad (3.21)$$

3.3.1 Numerical algorithm

Consider the Robin boundary conditions (3.20) and (3.21). They are in differential forms with respect to the spatial variable. We can rewrite them at time t_m into general

form for both the left and right boundary conditions as follows

$$\alpha_j(t_m) \frac{\partial u(x, t_m)}{\partial x} \Big|_{x=\gamma_j} + \beta_j(t_m) u(\gamma_j, t_m) = g_j(t_m), \quad j \in \{1, 2\}, \quad (3.22)$$

where $\gamma_1 = a$ and $\gamma_2 = b$. Note that (3.22) is the left boundary condition if $j = 1$ and the right boundary condition if $j = 2$. Next, we transform (3.22) at time t_m into the vector form by using the Chebyshev polynomial expansion (2.7). Consider the derivative term with respect to x contained in (3.22). Then, we obtain

$$\frac{\partial u(x, t_m)}{\partial x} \Big|_{x=\gamma_j} = u_x^{(m)}(\gamma_j) = \sum_{n=0}^{M-1} c_n^{(m)} R_n'(\gamma_j) := \mathbf{R}'_j \mathbf{c}^{(m)} = \mathbf{R}'_j \mathbf{R}^{-1} \mathbf{u}^{(m)}, \quad (3.23)$$

where $\mathbf{R}'_j = [R_0'(\gamma_j), R_1'(\gamma_j), R_2'(\gamma_j), \dots, R_{M-1}'(\gamma_j)]$ in which the first-order derivative $R_n'(\cdot)$ is defined in (2.4) and \mathbf{R}^{-1} is defined in Lemma 2.1(iv). Finally, by substituting (3.10) and (3.23) into (3.22), we have

$$\alpha_j(t_m) \mathbf{R}'_j \mathbf{R}^{-1} \mathbf{u}^{(m)} + \beta_j(t_m) \mathbf{R}_j \mathbf{R}^{-1} \mathbf{u}^{(m)} = g_j(t_m).$$

Thus, for $j \in \{1, 2\}$, we obtain the following left and right boundary conditions as

$$\left(\alpha_1(t_m) \mathbf{R}'_1 + \beta_1(t_m) \mathbf{R}_1 \right) \mathbf{R}^{-1} \mathbf{u}^{(m)} = g_1(t_m), \quad (3.24)$$

$$\left(\alpha_2(t_m) \mathbf{R}'_2 + \beta_2(t_m) \mathbf{R}_2 \right) \mathbf{R}^{-1} \mathbf{u}^{(m)} = g_2(t_m), \quad (3.25)$$

where

$$\mathbf{R}_1 = [R_0(a), R_1(a), R_2(a), \dots, R_{M-1}(a)],$$

$$\mathbf{R}'_1 = [R_0'(a), R_1'(a), R_2'(a), \dots, R_{M-1}'(a)],$$

$$\mathbf{R}_2 = [R_0(b), R_1(b), R_2(b), \dots, R_{M-1}(b)],$$

$$\mathbf{R}'_2 = [R_0'(b), R_1'(b), R_2'(b), \dots, R_{M-1}'(b)].$$

Having fully derived all equations necessary for determining $\mathbf{u}^{(m)}$, we can now incorporate (3.6) with the boundary conditions, or with (3.24) and (3.25), to establish a

system of linear equations. As a result, we end up with two different linear systems of $M + 2$ unknowns, namely $\mathbf{u}^{(m)}$, h_1 and h_2 , contingent on the kind of Robin boundary conditions, which are defined as follows:

$$\left[\begin{array}{c|cc} \mathbf{A}^2 - \frac{\Delta t}{2}\mathbf{I} & -\mathbf{x} & -\mathbf{e} \\ \hline (\alpha_1(t_m)\mathbf{R}'_1 + \beta_1(t_m)\mathbf{R}_1)\mathbf{R}^{-1} & 0 & 0 \\ (\alpha_2(t_m)\mathbf{R}'_2 + \beta_2(t_m)\mathbf{R}_2)\mathbf{R}^{-1} & 0 & 0 \end{array} \right] \begin{bmatrix} \mathbf{u}^{(m)} \\ h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} \mathbf{z}^{(m)} \\ g_1(t_m) \\ g_2(t_m) \end{bmatrix}, \quad (3.26)$$

where $[\mathbf{z}^{(m)}]_{M \times 1} := (\mathbf{A}^2 + \frac{\Delta t}{2}\mathbf{I})\mathbf{u}^{(m-1)} + \frac{\Delta t}{2}\mathbf{A}^2(\mathbf{q}^{(m)} + \mathbf{q}^{(m-1)})$. Therefore, we can approximate the solution $\mathbf{u}^{(m)}$ by solving the system (3.26) and (2.15), beginning from the provided initial condition $\mathbf{u}^{(0)} = [f(x_1), f(x_2), f(x_3), \dots, f(x_M)]^\top$. It is noteworthy that when executing the final iteration, the obtained numerical solution $\mathbf{u}^{(m)} = [u(x_1, T), u(x_2, T), u(x_3, T), \dots, u(x_M, T)]^\top$ can be actually expressed corresponding to the function $u(x, T)$, $x \in [a, b]$, by using (3.19).

To facilitate computation, we encapsulate the aforementioned processes into pseudocode algorithms, which aim to find approximate solutions of the heat equation with the Robin boundary using a combination of FIM-CPE and the Crank-Nicolson method.

Algorithm 2 One-dimensional heat equation with Robin boundary conditions

Input: $a, b, M, T, \Delta t, q(x, t), f(x), g_1(t), g_2(t), \alpha_1(t), \alpha_2(t), \beta_1(t)$ and $\beta_2(t)$;

Output: The approximate solution $\mathbf{u}^{(m)}$;

- 1: Set $x_k \leftarrow \frac{1}{2} \left((b - a) \cos \left(\frac{2k-1}{2n} \pi \right) + a + b \right)$ for $k \in \{1, 2, 3, \dots, M\}$ in ascending order;
 - 2: Compute $\mathbf{x}, \mathbf{e}, \mathbf{I}, \mathbf{A}, \mathbf{R}, \mathbf{R}_1, \mathbf{R}_2, \mathbf{R}'_1, \mathbf{R}'_2, \bar{\mathbf{R}}$ and \mathbf{R}^{-1} ;
 - 3: Construct $\mathbf{u}^{(0)} \leftarrow [f(x_1), f(x_2), f(x_3), \dots, f(x_M)]^\top$;
 - 4: Set $m \leftarrow 1$ and $t_1 \leftarrow \Delta t$;
 - 5: **while** $t_m \leq T$ **do**
 - 6: Compute $\mathbf{q}^{(m-1)} \leftarrow [q(x_1, t_m - \Delta t), q(x_2, t_m - \Delta t), \dots, q(x_M, t_m - \Delta t)]^\top$;
 - 7: Compute $\mathbf{q}^{(m)} \leftarrow [q(x_1, t_m), q(x_2, t_m), q(x_3, t_m), \dots, q(x_M, t_m)]^\top$;
 - 8: Compute $\mathbf{z}^{(m)} \leftarrow (\mathbf{A}^2 + \frac{\Delta t}{2} \mathbf{I}) \mathbf{u}^{(m-1)} + \frac{\Delta t}{2} \mathbf{A}^2 (\mathbf{q}^{(m)} + \mathbf{q}^{(m-1)})$;
 - 9: Find $\mathbf{u}^{(m)}$ by solving the iterative linear system (3.26);
 - 10: Update $m \leftarrow m + 1$ and $t_m \leftarrow m \Delta t$;
 - 11: **end while**
 - 12: **return** The final iteration of $\mathbf{u}^{(m)}$ is the approximate solution;
-

3.3.2 Numerical example

Within this section, we validate the accuracy and efficiency of the proposed numerical Algorithms 2 through their application to approximate solutions of the heat equation with Robin boundary conditions. This validation is conducted via Example 3.3. In order to quantify the accuracy of the solutions derived, we employ two metrics, namely the ARE and the MAE, as defined by

$$\text{ARE} = \frac{1}{M} \sum_{k=1}^M \left| \frac{u^*(x_k, T) - u(x_k, T)}{u^*(x_k, T)} \right| \quad \text{and} \quad \text{MAE} = \max_{1 \leq k \leq M} |u^*(x_k, T) - u(x_k, T)|,$$

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

Example 3.3 ([24]). We consider the heat equation

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

subject to the initial condition

$$u(x, 0) = \sin(\pi x) + \cos(\pi x), \quad x \in [0, 1],$$

with the Robin boundary conditions

$$u_x(0, t) + t^2 u(0, t) = e^{-t}(t^2 + \pi), \quad t \in (0, 1],$$

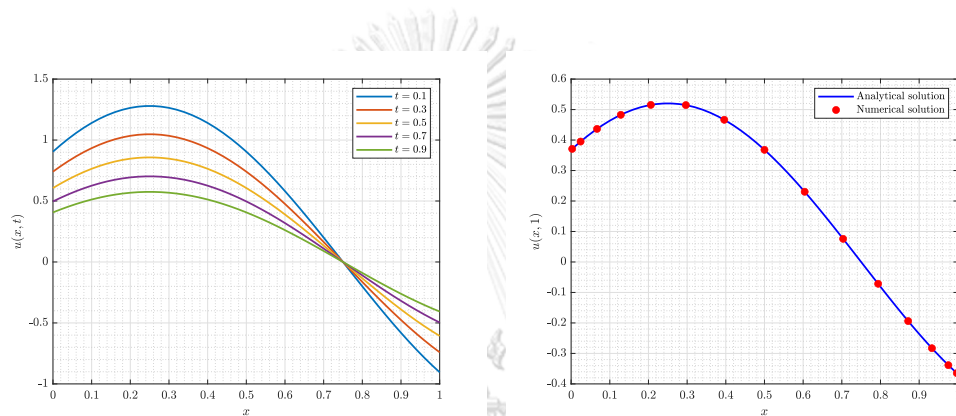
$$u_x(1, t) + tu(1, t) = -e^{-t}(t + \pi), \quad t \in (0, 1].$$

The analytical solution is $u(x, t) = e^{-t}(\sin(\pi x) + \cos(\pi x))$.

The presented example takes into account Robin boundary conditions. Therefore, numerical solutions can be computed utilizing Algorithm 2. As a result, we acquire the approximate solutions denoted by $\mathbf{u}^{(m)}$. The precision of these achieved solutions at time $t = 1$, quantified by the ARE, is displayed in Table 3.3 for the range of $M \in \{10, 15, 20, 25, 30\}$ and time increments $\Delta t \in \{0.1, 0.05, 0.01, 0.005\}$. This table reveals that the ARE diminishes modestly as the number of nodes M amplifies. Nonetheless, the significant reduction is noted as the time increment Δt contracts. Subsequently, the graphical solutions derived from Algorithm 2 at different time instances $t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, along with comparisons with the analytical solution at $M = 15$ and $t = 1$, are portrayed in Figure 3.3.

Table 3.3: The ARE of the numerical solutions $u(x, 1)$ for Example 3.3.

M	$\Delta t = 0.1$	$\Delta t = 0.05$	$\Delta t = 0.01$	$\Delta t = 0.005$
10	0.0019	5.2250×10^{-4}	7.8118×10^{-5}	6.4422×10^{-6}
15	0.0015	3.8031×10^{-4}	1.5193×10^{-5}	3.7982×10^{-6}
20	0.0020	5.0906×10^{-4}	2.0337×10^{-5}	5.0840×10^{-6}
25	0.0021	5.2526×10^{-4}	2.0984×10^{-5}	5.2459×10^{-6}
30	0.0017	4.2664×10^{-4}	1.7044×10^{-5}	4.2591×10^{-6}



(a) Numerical solutions in (3.19) at various $t \in (0, 1)$
 (b) Compared solutions with $M = 15$ and $t = 1$

Figure 3.3: Graphical numerical solutions $u(x, t)$ in (3.19) from Algorithm 2 for Example 3.3.

CHAPTER IV

TWO-DIMENSIONAL HEAT EQUATION

In the present chapter, we develop a numerical algorithm designed to solve the two-dimensional heat equation over a rectangular domain (2.20), which includes the initial condition (2.21) and takes into consideration the non-local boundary conditions (2.22)–(2.25). Our proposed algorithm hinges on a two-dimensional application of the FIM-CPE and incorporates the Crank-Nicolson method. Additionally, we provide numerical examples to verify the accuracy of the algorithms that we propose.

4.1 Formulation of two-dimension heat equation

Initially, we aim to transform the two-dimensional heat equation (2.20) into a matrix formulation by applying the two-dimensional FIM-CPE for the spatial domain and employing the Crank-Nicolson method for the temporal variable. To commence this procedure, we recall the problems under examination, as detailed in the following:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + q(x, y, t), \quad (x, y, t) \in (a, b) \times (c, d) \times (0, T]. \quad (4.1)$$

Subsequently, we focus on the heat equation (4.1). We initiate by discretizing the time domain $(0, T]$ uniformly, designating each time point as $t_m = m\Delta t$ for $m \in \mathbb{N}$ in (4.1), where Δt represents the designated time step. This results in

$$\begin{aligned} \left. \frac{\partial u(x, y, t)}{\partial t} \right|_{t=t_m} &= \frac{\partial^2 u(x, y, t_m)}{\partial x^2} + \frac{\partial^2 u(x, y, t_m)}{\partial y^2} + q(x, y, t_m) \\ &:= G(x, y, t_m, u(x, y, t_m)), \end{aligned} \quad (4.2)$$

where $G(x, y, t, u)$ represents the right-hand-side term of (4.1).

Following, we observe that (4.2) contains a derivative with respect to time t . We can also use the difference quotients to approximate time derivative but it will result the less accuracy and cannot guarantee the stability comparing to the Crank-Nicolson method. So, we approximate this derivative using the Crank-Nicolson method, which delivers a time complexity of $\mathcal{O}(\Delta t^2)$. Consequently, we can approximate (4.2) as follows

$$\frac{u^{(m)}(x, y) - u^{(m-1)}(x, y)}{\Delta t} = \frac{G(x, y, t_{m-1}, u^{(m-1)}(x, y)) + G(x, y, t_m, u^{(m)}(x, y))}{2}, \quad (4.3)$$

where $u^{(m)}(x, y) := u(x, y, t_m)$ and G at the times t_m and t_{m-1} are defined by

$$\begin{aligned} G(x, y, t_m, u^{(m)}(x, y)) &= \frac{\partial^2 u^{(m)}(x, y)}{\partial x^2} + \frac{\partial^2 u^{(m)}(x, y)}{\partial y^2} + q^{(m)}(x, y) \quad \text{and} \\ G(x, y, t_{m-1}, u^{(m-1)}(x, y)) &= \frac{\partial^2 u^{(m-1)}(x, y)}{\partial x^2} + \frac{\partial^2 u^{(m-1)}(x, y)}{\partial y^2} + q^{(m-1)}(x, y). \end{aligned}$$

When the functions G at times t_m and t_{m-1} are substituted in (4.3), we have

$$\begin{aligned} u^{(m)}(x, y) - u^{(m-1)}(x, y) &= \frac{\Delta t}{2} \left(\frac{\partial^2 u^{(m-1)}(x, y)}{\partial x^2} + \frac{\partial^2 u^{(m-1)}(x, y)}{\partial y^2} + q^{(m-1)}(x, y) \right) \\ &\quad + \frac{\Delta t}{2} \left(\frac{\partial^2 u^{(m)}(x, y)}{\partial x^2} + \frac{\partial^2 u^{(m)}(x, y)}{\partial y^2} + q^{(m)}(x, y) \right). \quad (4.4) \end{aligned}$$

Next, the FIM-CPE is utilized to remove all spatial derivatives from (4.4) by performing double integrals with respect to variables x and y , respectively. This operation transforms the differential equation (4.4) into its equivalent integral equation.

$$\begin{aligned} &\int_c^y \int_c^{\eta_2} \int_a^x \int_a^{\xi_2} \left(u^{(m)}(\xi_1, \eta_1) - u^{(m-1)}(\xi_1, \eta_1) \right) d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\ &= \frac{\Delta t}{2} \int_c^y \int_c^{\eta_2} \left(u^{(m-1)}(x, \eta_1) + u^{(m)}(x, \eta_1) \right) d\eta_1 d\eta_2 \\ &\quad + \frac{\Delta t}{2} \int_a^x \int_a^{\xi_2} \left(u^{(m-1)}(\xi_1, y) + u^{(m)}(\xi_1, y) \right) d\xi_1 d\xi_2 \\ &\quad + \frac{\Delta t}{2} \int_c^y \int_c^{\eta_2} \int_a^x \int_a^{\xi_2} \left(q^{(m-1)}(\xi_1, \eta_1) + q^{(m)}(\xi_1, \eta_1) \right) d\xi_1 d\xi_2 d\eta_1 d\eta_2 \\ &\quad + xb_1(y) + b_2(y) + yd_1(x) + d_2(x), \quad (4.5) \end{aligned}$$

where $b_1(y)$, $b_2(y)$, $d_1(x)$ and $d_2(x)$ are arbitrary functions that arise during the integration process. To address these undetermined functions, Chebyshev interpolation is employed for their approximation as follows:

$$b_r(y) = \sum_{n=0}^{N-1} b_{r,n} R_n(y) \quad \text{and} \quad d_r(x) = \sum_{n=0}^{M-1} d_{r,n} R_n(x) \quad (4.6)$$

for $r \in \{1, 2\}$, with $\{b_{r,n}\}_{n=0}^{N-1}$ and $\{d_{r,n}\}_{n=0}^{M-1}$ being the unknown values at these interpolated points, determined in accordance with the specified non-local boundary conditions (2.22)–(2.25). Subsequently, we discretize both the horizontal and vertical dimensions of the rectangular domain $[a, b] \times [c, d]$ into M and N points, respectively, using the zeros of Chebyshev polynomials $R_M(x)$ and $R_N(y)$. These are defined by sets $X = \{x_1, x_2, x_3, \dots, x_M\}$ and $Y = \{y_1, y_2, y_3, \dots, y_N\}$, respectively. Thus, the total count of grid points in the global numbering system amounts to $H = MN$ nodes. Note that each node in the system originates from an element in the Cartesian product set $X \times Y$, ordered as per the global-type system, i.e., $(x_i, y_i) \in X \times Y$ for $i \in \{1, 2, 3, \dots, H\}$. Upon substituting each node (x_i, y_i) into (4.5) and transforming it into the matrix form using the approach of the FIM-CPE in two-dimensional spaces as described in Section 2.3.2, then we obtain

$$\begin{aligned} & \left(\mathbf{A}_x^2 \mathbf{A}_y^2 - \frac{\Delta t}{2} (\mathbf{A}_x^2 + \mathbf{A}_y^2) \right) \mathbf{u}^{(m)} - \mathbf{X} \Phi_y \mathbf{b}_1 - \Phi_y \mathbf{b}_2 - \mathbf{Y} \Phi_x \mathbf{d}_1 - \Phi_x \mathbf{d}_2 \\ & = \left(\mathbf{A}_x^2 \mathbf{A}_y^2 + \frac{\Delta t}{2} (\mathbf{A}_x^2 + \mathbf{A}_y^2) \right) \mathbf{u}^{(m-1)} + \frac{\Delta t}{2} \mathbf{A}_x^2 \mathbf{A}_y^2 (\mathbf{q}^{(m-1)} + \mathbf{q}^{(m)}), \end{aligned} \quad (4.7)$$

where the Chebyshev integration matrices \mathbf{A}_x and \mathbf{A}_y are defined in Remark 2.2. Other

parameters contained in (4.7) are defined by

$$\begin{aligned}
\mathbf{X} &= \text{diag}\{x_1, x_2, x_3, \dots, x_H\}, \\
\mathbf{Y} &= \text{diag}\{y_1, y_2, y_3, \dots, y_H\}, \\
\mathbf{b}_r &= [b_{r,0}, b_{r,1}, b_{r,2}, \dots, b_{r,N-1}]^\top \quad \text{for } r \in \{1, 2\}, \\
\mathbf{d}_r &= [d_{r,0}, d_{r,1}, d_{r,2}, \dots, d_{r,M-1}]^\top \quad \text{for } r \in \{1, 2\}, \\
\mathbf{u}^{(m)} &= [u(x_1, y_1, t_m), u(x_2, y_2, t_m), u(x_3, y_3, t_m), \dots, u(x_H, y_H, t_m)]^\top, \\
\mathbf{q}^{(m)} &= [q(x_1, y_1, t_m), q(x_2, y_2, t_m), q(x_3, y_3, t_m), \dots, q(x_H, y_H, t_m)]^\top,
\end{aligned}$$

where each (x_i, y_i) be an element in a set of the Cartesian product $X \times Y$ as mentioned above. From (4.6), we can obtain Φ_x and Φ_y , where

$$\Phi_x = \begin{bmatrix} R_0(x_1) & R_1(x_1) & \cdots & R_{M-1}(x_1) \\ R_0(x_2) & R_1(x_2) & \cdots & R_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_0(x_H) & R_1(x_H) & \cdots & R_{M-1}(x_H) \end{bmatrix}$$

and

$$\Phi_y = \begin{bmatrix} R_0(y_1) & R_1(y_1) & \cdots & R_{N-1}(y_1) \\ R_0(y_2) & R_1(y_2) & \cdots & R_{N-1}(y_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_0(y_H) & R_1(y_H) & \cdots & R_{N-1}(y_H) \end{bmatrix}.$$

4.2 Algorithm for non-local boundary conditions

In this part of our study, we transform the two-dimensional non-local boundary conditions (2.22)–(2.25) into their respective matrix forms. By amalgamating these with the matrix from the prior section, we assemble the numerical algorithm. Moreover, we explore the efficacy and precision of the algorithm through various numerical examples. The non-local boundary conditions (2.22)–(2.25) in two dimensions can be expressed in

the following manner:

$$u(a, y, t) = \int_c^d \int_a^b k_1(a, y, \xi, \eta) u(\xi, \eta, t) d\xi d\eta, \quad y \in [c, d], \quad (4.8)$$

$$u(b, y, t) = \int_c^d \int_a^b k_2(b, y, \xi, \eta) u(\xi, \eta, t) d\xi d\eta, \quad y \in [c, d], \quad (4.9)$$

$$u(x, c, t) = \int_c^d \int_a^b k_3(x, c, \xi, \eta) u(\xi, \eta, t) d\xi d\eta, \quad x \in [a, b], \quad (4.10)$$

$$u(x, d, t) = \int_c^d \int_a^b k_4(x, d, \xi, \eta) u(\xi, \eta, t) d\xi d\eta, \quad x \in [a, b], \quad (4.11)$$

for $t \in (0, T]$. The function $k_i(x, y)$ for $i \in \{1, 2, 3, 4\}$ symbolize given functions.

4.2.1 Numerical algorithm

When dealing with the two-dimensional non-local boundary conditions (4.8)–(4.11), they adopt the form of double-layer integrals in relation to the spatial variables. We can recast these conditions at time t_m into a general format as follows:

$$u^{(m)}(x, y) = \int_c^d \int_a^b k(x, y, \xi, \eta) u^{(m)}(\xi, \eta) d\xi d\eta, \quad (4.12)$$

where the elements (x, y) in the equation pertain to the points (a, y) , (b, y) , (x, c) , and (x, d) , denoting the left, right, bottom, and top boundaries, respectively, on the left-hand side of (4.12). As for the right-hand side of (4.12), at time instance t_m , it is viable to transform it into a matrix form. The transformation can be facilitated by invoking the associations detailed in (3.11) and (3.14), following a similar rationale to (3.15). Hence,

the right-hand side of (4.12) can be articulated as

$$\begin{aligned}
\text{RHS} &= \int_c^d \int_a^b k(x, y, \xi, \eta) u^{(m)}(\xi, \eta) d\xi d\eta \\
&= \int_c^d \sum_{i=0}^{M-1} p_i \bar{R}_i(b) d\eta, \text{ where } p_i \text{ is unknown coefficients to approximate } k(\cdot)u^{(m)}(\cdot) \\
&= \int_c^d \bar{\mathbf{R}}(b) \mathbf{p} d\eta, \text{ where } \mathbf{p} = [p_0, p_1, p_2, \dots, p_{M-1}]^\top \\
&= \int_c^d \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1} \underbrace{\text{diag}\{k(x_1, \eta), k(x_2, \eta), \dots, k(x_M, \eta)\}}_{\mathbf{k}(\cdot, \eta)} \mathbf{u}^{(m)}(\cdot, \eta) d\eta \\
&= \sum_{j=0}^{N-1} p_j \bar{R}_j(d), \text{ where } p_j \text{ is unknown coefficients for } \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1} \mathbf{k}(\cdot) \mathbf{u}^{(m)}(\cdot) \\
&= \bar{\mathbf{R}}(d) \mathbf{p}, \text{ where } \mathbf{p} = [p_0, p_1, p_2, \dots, p_{N-1}]^\top \\
&= \bar{\mathbf{R}}(d) \mathbf{R}_N^{-1} \left[\bar{\mathbf{R}}(b) \mathbf{R}_M^{-1} \mathbf{k}(\cdot, y_1) \mathbf{u}^{(m)}(\cdot, y_1), \dots, \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1} \mathbf{k}(\cdot, y_N) \mathbf{u}^{(m)}(\cdot, y_N) \right]^\top \\
&= \bar{\mathbf{R}}(d) \mathbf{R}_N^{-1} \begin{bmatrix} \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1} & & & \\ & \ddots & & \\ & & \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1} & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{k}(\cdot, y_1) & & & \\ & \ddots & & \\ & & \mathbf{k}(\cdot, y_N) & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(m)}(\cdot, y_1) \\ \vdots \\ \mathbf{u}^{(m)}(\cdot, y_N) \end{bmatrix} \\
&:= \bar{\mathbf{R}}(d) \mathbf{R}_N^{-1} (\mathbf{I}_N \otimes \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1}) \mathbf{K} \mathbf{u}^{(m)}, \tag{4.13}
\end{aligned}$$

where \mathbf{R}_M^{-1} and \mathbf{R}_N^{-1} represent the inverses of the $M \times M$ and $N \times N$ Chebyshev matrices, respectively, as outlined in Lemma 2.1(v). The additional parameters included in (4.13) are defined as follows:

$$\begin{aligned}
\bar{\mathbf{R}}(b) &= [\bar{R}_0(b), \bar{R}_1(b), \bar{R}_2(b), \dots, \bar{R}_{M-1}(b)], \\
\bar{\mathbf{R}}(d) &= [\bar{R}_0(d), \bar{R}_1(d), \bar{R}_2(d), \dots, \bar{R}_{M-1}(d)], \\
\mathbf{K}_i &= \text{diag}\{k_i(x_1, y_1), k_i(x_2, y_2), k_i(x_3, y_3), \dots, k_i(x_H, y_H)\} \quad \text{where } i \in \{1, 2, 3, 4\},
\end{aligned}$$

where in each coordinate pair (x_i, y_i) belongs to the Cartesian product set $X \times Y$ and is ordered in accordance with the global numbering system. Here, $X = \{x_1, x_2, x_3, \dots, x_M\}$ and $Y = \{y_1, y_2, y_3, \dots, y_N\}$ denote the set of zeros of the Chebyshev polynomials R_M and R_N , correspondingly.

We then employ the Chebyshev polynomial expansion (2.7) to modify the left-hand side of (4.12). This process can be segregated into two parts. Specifically, for the left and right boundary conditions, we hold the value of $x \in \{a, b\}$ constant while varying the value of y . Conversely, for the bottom and top boundary conditions, we keep the value of $y \in \{c, d\}$ constant and change the value of x . This results in the following expressions:

$$u^{(m)}(a, y) = \sum_{n=0}^{M-1} c_n^{(m)}(y) R_n(a) = \sum_{n=0}^{M-1} c_n^{(m)}(y) (-1)^n := \mathbf{s}_M \mathbf{R}_M^{-1} \mathbf{u}^{(m)}(\cdot, y), \quad (4.14)$$

$$u^{(m)}(b, y) = \sum_{n=0}^{M-1} c_n^{(m)}(y) R_n(b) = \sum_{n=0}^{M-1} c_n^{(m)}(y) (+1)^n := \mathbf{e}_M \mathbf{R}_M^{-1} \mathbf{u}^{(m)}(\cdot, y), \quad (4.15)$$

$$u^{(m)}(x, c) = \sum_{n=0}^{N-1} c_n^{(m)}(x) R_n(c) = \sum_{n=0}^{N-1} c_n^{(m)}(x) (-1)^n := \mathbf{s}_N \mathbf{R}_N^{-1} \mathbf{u}^{(m)}(x, \cdot), \quad (4.16)$$

$$u^{(m)}(x, d) = \sum_{n=0}^{N-1} c_n^{(m)}(x) R_n(d) = \sum_{n=0}^{N-1} c_n^{(m)}(x) (+1)^n := \mathbf{e}_N \mathbf{R}_N^{-1} \mathbf{u}^{(m)}(x, \cdot), \quad (4.17)$$

where $\mathbf{s}_n = [1, -1, 1, \dots, (-1)^{n-1}]$ and $\mathbf{e}_n = [1, 1, 1, \dots, 1]$ represent the n -element row vectors, and \mathbf{R}_n^{-1} is an inverse Chebyshev matrix as defined in Lemma 2.1(v). Then, we apply the zeros of the Chebyshev $y \in \{y_1, y_2, y_3, \dots, y_N\}$ to (4.14) and (4.15), and likewise apply $x \in \{x_1, x_2, x_3, \dots, x_M\}$ to (4.16) and (4.17) as illustrated below:

- **Left boundary conditions:**

$$\begin{bmatrix} \mathbf{u}^{(m)}(a, y_1) \\ \mathbf{u}^{(m)}(a, y_2) \\ \vdots \\ \mathbf{u}^{(m)}(a, y_N) \end{bmatrix} = \begin{bmatrix} \mathbf{s}_M \mathbf{R}_M^{-1} & 0 & \cdots & 0 \\ 0 & \mathbf{s}_M \mathbf{R}_M^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{s}_M \mathbf{R}_M^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(m)}(\cdot, y_1) \\ \mathbf{u}^{(m)}(\cdot, y_2) \\ \vdots \\ \mathbf{u}^{(m)}(\cdot, y_N) \end{bmatrix}.$$

It can be simplified to $\mathbf{u}^{(m)}(a, \cdot) = (\mathbf{I}_N \otimes \mathbf{s}_M \mathbf{R}_M^{-1}) \mathbf{u}^{(m)}$ that is equivalent to RHS in (4.13). Thus, we have

$$\mathbf{B}_\ell^{(m)} \mathbf{u}^{(m)} := [(\mathbf{I}_N \otimes \mathbf{s}_M \mathbf{R}_M^{-1}) - \bar{\mathbf{R}}(d) \mathbf{R}_N^{-1} (\mathbf{I}_N \otimes \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1}) \mathbf{K}_1] \mathbf{u}^{(m)} = \mathbf{0}. \quad (4.18)$$

- **Right boundary conditions:**

$$\begin{bmatrix} \mathbf{u}^{(m)}(b, y_1) \\ \mathbf{u}^{(m)}(b, y_2) \\ \vdots \\ \mathbf{u}^{(m)}(b, y_N) \end{bmatrix} = \begin{bmatrix} \mathbf{e}_M \mathbf{R}_M^{-1} & 0 & \cdots & 0 \\ 0 & \mathbf{e}_M \mathbf{R}_M^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{e}_M \mathbf{R}_M^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(m)}(\cdot, y_1) \\ \mathbf{u}^{(m)}(\cdot, y_2) \\ \vdots \\ \mathbf{u}^{(m)}(\cdot, y_N) \end{bmatrix}.$$

It can be simplified to $\mathbf{u}^{(m)}(b, \cdot) = (\mathbf{I}_N \otimes \mathbf{e}_M \mathbf{R}_M^{-1}) \mathbf{u}^{(m)}$. That is equivalent to RHS in (4.13). Thus, we have

$$\mathbf{B}_r^{(m)} \mathbf{u}^{(m)} := [(\mathbf{I}_N \otimes \mathbf{e}_M \mathbf{R}_M^{-1}) - \bar{\mathbf{R}}(d) \mathbf{R}_N^{-1} (\mathbf{I}_N \otimes \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1}) \mathbf{K}_2] \mathbf{u}^{(m)} = \mathbf{0}. \quad (4.19)$$

- **Bottom boundary conditions:**

$$\begin{bmatrix} \mathbf{u}^{(m)}(x_1, c) \\ \mathbf{u}^{(m)}(x_2, c) \\ \vdots \\ \mathbf{u}^{(m)}(x_M, c) \end{bmatrix} = \begin{bmatrix} \mathbf{s}_N \mathbf{R}_N^{-1} & 0 & \cdots & 0 \\ 0 & \mathbf{s}_N \mathbf{R}_N^{-1} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{s}_N \mathbf{R}_N^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(m)}(x_1, \cdot) \\ \mathbf{u}^{(m)}(x_2, \cdot) \\ \vdots \\ \mathbf{u}^{(m)}(x_N, \cdot) \end{bmatrix}.$$

It can be simplified to $\mathbf{u}^{(m)}(\cdot, c) = (\mathbf{I}_M \otimes \mathbf{s}_N \mathbf{R}_N^{-1}) \tilde{\mathbf{u}}^{(m)} = (\mathbf{I}_M \otimes \mathbf{s}_N \mathbf{R}_N^{-1}) \mathbf{P}^{-1} \mathbf{u}^{(m)}$ that is equivalent to RHS in (4.13). Thus, we have

$$\mathbf{B}_b^{(m)} \mathbf{u}^{(m)} := [(\mathbf{I}_M \otimes \mathbf{s}_N \mathbf{R}_N^{-1}) \mathbf{P}^{-1} - \bar{\mathbf{R}}(d) \mathbf{R}_N^{-1} (\mathbf{I}_N \otimes \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1}) \mathbf{K}_3] \mathbf{u}^{(m)} = \mathbf{0}. \quad (4.20)$$

- **Top boundary conditions:**

$$\begin{bmatrix} \mathbf{u}^{(m)}(x_1, d) \\ \mathbf{u}^{(m)}(x_2, d) \\ \vdots \\ \mathbf{u}^{(m)}(x_M, d) \end{bmatrix} = \begin{bmatrix} \mathbf{e}_N \mathbf{R}_N^{-1} & 0 & \cdots & 0 \\ 0 & \mathbf{e}_N \mathbf{R}_N^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{e}_N \mathbf{R}_N^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(m)}(x_1, \cdot) \\ \mathbf{u}^{(m)}(x_2, \cdot) \\ \vdots \\ \mathbf{u}^{(m)}(x_N, \cdot) \end{bmatrix}.$$

It can be simplified to $\mathbf{u}^{(m)}(\cdot, d) = (\mathbf{I}_M \otimes \mathbf{e}_N \mathbf{R}_N^{-1}) \tilde{\mathbf{u}}^{(m)} = (\mathbf{I}_M \otimes \mathbf{e}_N \mathbf{R}_N^{-1}) \mathbf{P}^{-1} \mathbf{u}^{(m)}$ that is equivalent to RHS in (4.13). Thus, we have

$$\mathbf{B}_t^{(m)} \mathbf{u}^{(m)} := [(\mathbf{I}_M \otimes \mathbf{e}_N \mathbf{R}_N^{-1}) \mathbf{P}^{-1} - \bar{\mathbf{R}}(d) \mathbf{R}_N^{-1} (\mathbf{I}_N \otimes \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1}) \mathbf{K}_4] \mathbf{u}^{(m)} = \mathbf{0}. \quad (4.21)$$

We have now gathered all the necessary components to construct the system of iterative linear equations, which will help us to compute the numerical solutions of the two-dimensional heat equation from (4.7), given the four non-local boundary conditions (4.18)–(4.21). This linear system comprises a total of $MN + 2(M + N)$ unknowns, which includes $\mathbf{u}^{(m)}$, \mathbf{b}_1 , \mathbf{b}_2 , \mathbf{d}_1 , and \mathbf{d}_2 . Here is how it looks:

$$\left[\begin{array}{c|cccc} \mathbf{A}_x^2 \mathbf{A}_y^2 - \frac{\Delta t}{2} (\mathbf{A}_x^2 + \mathbf{A}_y^2) & -\mathbf{X} \Phi_y & -\Phi_y & -\mathbf{Y} \Phi_x & -\Phi_x \\ \hline \mathbf{B}_\ell^{(m)} & 0 & 0 & \cdots & 0 \\ \mathbf{B}_r^{(m)} & 0 & 0 & \cdots & 0 \\ \mathbf{B}_b^{(m)} & \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_t^{(m)} & 0 & 0 & \cdots & 0 \end{array} \right] \begin{bmatrix} \mathbf{u}^{(m)} \\ \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{z}^{(m)} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad (4.22)$$

where $\mathbf{z}^{(m)} := (\mathbf{A}_x^2 \mathbf{A}_y^2 + \frac{\Delta t}{2} (\mathbf{A}_x^2 + \mathbf{A}_y^2)) \mathbf{u}^{(m-1)} + \frac{\Delta t}{2} \mathbf{A}_x^2 \mathbf{A}_y^2 (\mathbf{q}^{(m-1)} + \mathbf{q}^{(m)})$. Consequently, the solution $\mathbf{u}^{(m)}$ can be deduced by solving (4.22) in conjunction with (2.21), beginning from the specified initial condition $\mathbf{u}^{(0)} = [f(x_1, y_1), f(x_2, y_2), f(x_3, y_3), \dots, f(x_H, y_H)]^\top$. It is important to highlight that, upon completion of the final iteration, the derived numerical solution $\mathbf{u}^{(m)} = [u(x_1, y_1, T), u(x_2, y_2, T), u(x_3, y_3, T), \dots, u(x_H, y_H, T)]^\top$ can be accurately equated to the function $u(x, y, T)$ by using the transformation processes in the same way as above the boundary conditions. Then, we have

$$\begin{aligned} u(x, y, T) &= \mathbf{R}_N(y) \mathbf{R}_N^{-1} \mathbf{u}^{(m)}(x, \cdot) \\ &= \mathbf{R}_N(y) \mathbf{R}_N^{-1} (\mathbf{I}_N \otimes \mathbf{R}_M(x) \mathbf{R}_M^{-1}) \mathbf{u}^{(m)} \\ &= (\mathbf{R}_N(y) \mathbf{R}_N^{-1} \otimes \mathbf{R}_M(x) \mathbf{R}_M^{-1}) \mathbf{u}^{(m)}, \end{aligned}$$

where $\mathbf{R}_N(y) = [R_0(y), R_1(y), \dots, R_{N-1}(y)]$ and $\mathbf{R}_M(x) = [R_0(x), R_1(x), \dots, R_{M-1}(x)]$.

To facilitate computation, we consolidate all previously discussed steps into pseudocode algorithms. These algorithms are devised to calculate approximate solutions for the two-dimensional heat equation with non-local boundaries. We utilize a combined approach of FIM-CPE and the Crank-Nicolson method in this process.

Algorithm 3 Two-dimensional heat equation with non-local boundary conditions

Input: $a, b, c, d, M, N, T, \Delta t, q(x, y, t), f(x, y)$ and $k(x, y)$;

Output: The approximate solution $\mathbf{u}^{(m)}$;

- 1: Set $x_k = \frac{1}{2} \left((b - a) \cos \left(\frac{2k-1}{2M} \pi \right) + a + b \right)$ for $k \in \{1, 2, 3, \dots, M\}$ in descending order;
 - 2: Set $y_h = \frac{1}{2} \left((d - c) \cos \left(\frac{2h-1}{2N} \pi \right) + c + d \right)$ for $h \in \{1, 2, 3, \dots, N\}$ in descending order;
 - 3: Calculate the total number of grid points $H = MN$;
 - 4: Construct x_i and y_i in the global numbering system for $i \in \{1, 2, 3, \dots, H\}$;
 - 5: Compute $\mathbf{P}, \mathbf{K}, \mathbf{X}, \mathbf{Y}, \Phi_x, \Phi_y, \mathbf{A}_x, \mathbf{A}_y, \mathbf{e}_M, \mathbf{e}_N, \mathbf{s}_M, \mathbf{s}_N, \mathbf{I}_M, \mathbf{I}_N, \mathbf{R}_M, \mathbf{R}_N, \bar{\mathbf{R}}(b), \bar{\mathbf{R}}(d)$
 - 6: Construct $\mathbf{u}^{(0)} \leftarrow [f(x_1, y_1), f(x_2, y_2), f(x_3, y_3), \dots, f(x_H, y_H)]^\top$;
 - 7: Set $m \leftarrow 1$ and $t_1 \leftarrow \Delta t$;
 - 8: **while** $t_m \leq T$ **do**
 - 9: Compute $\mathbf{B}_\ell^{(m)} \leftarrow (\mathbf{I}_N \otimes \mathbf{s}_M \mathbf{R}_M^{-1}) - \bar{\mathbf{R}}(d) \mathbf{R}_N^{-1} (\mathbf{I}_N \otimes \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1}) \mathbf{K}$;
 - 10: Compute $\mathbf{B}_r^{(m)} \leftarrow (\mathbf{I}_N \otimes \mathbf{e}_M \mathbf{R}_M^{-1}) - \bar{\mathbf{R}}(d) \mathbf{R}_N^{-1} (\mathbf{I}_N \otimes \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1}) \mathbf{K}$;
 - 11: Compute $\mathbf{B}_b^{(m)} \leftarrow (\mathbf{I}_M \otimes \mathbf{s}_N \mathbf{R}_N^{-1}) \mathbf{P}^{-1} - \bar{\mathbf{R}}(d) \mathbf{R}_N^{-1} (\mathbf{I}_N \otimes \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1}) \mathbf{K}$;
 - 12: Compute $\mathbf{B}_t^{(m)} \leftarrow (\mathbf{I}_M \otimes \mathbf{e}_N \mathbf{R}_N^{-1}) \mathbf{P}^{-1} - \bar{\mathbf{R}}(d) \mathbf{R}_N^{-1} (\mathbf{I}_N \otimes \bar{\mathbf{R}}(b) \mathbf{R}_M^{-1}) \mathbf{K}$;
 - 13: Compute $\mathbf{q}^{(m-1)} \leftarrow [q(x_1, y_1, t_m - \Delta t), q(x_2, y_2, t_m - \Delta t), \dots, q(x_H, y_H, t_m - \Delta t)]^\top$;
 - 14: Compute $\mathbf{q}^{(m)} \leftarrow [q(x_1, y_1, t_m), q(x_2, y_2, t_m), q(x_3, y_3, t_m), \dots, q(x_H, y_H, t_m)]^\top$;
 - 15: Compute $\mathbf{z}^{(m)} \leftarrow (\mathbf{A}_x^2 \mathbf{A}_y^2 + \frac{\Delta t}{2} (\mathbf{A}_x^2 + \mathbf{A}_y^2)) \mathbf{u}^{(m-1)} + \frac{\Delta t}{2} \mathbf{A}_x^2 \mathbf{A}_y^2 (\mathbf{q}^{(m-1)} + \mathbf{q}^{(m)})$;
 - 16: Find $\mathbf{u}^{(m)}$ by solving the iterative linear system (4.22);
 - 17: Update $m \leftarrow m + 1$ and $t_m \leftarrow m \Delta t$;
 - 18: **end while**
 - 19: **return** The final iteration of $\mathbf{u}^{(m)}$ is the approximate solution;
-

4.2.2 Numerical examples

In this part of the thesis, we present the results of employing the proposed numerical Algorithm 3 to derive approximate solutions to the heat equation with non-local boundary conditions in a two-dimensional context, as illustrated in Examples 4.1 and 4.2 for the heat equation without linear forcing term. However, our algorithm can also be applied for the heat equation with linear forcing term too. The ARE is used as a metric to determine the accuracy of the obtained solutions and it is given by

$$\text{ARE} = \frac{1}{H} \sum_{k=1}^H \left| \frac{u^*(x_k, y_k, T) - u(x_k, y_k, T)}{u^*(x_k, y_k, T)} \right|,$$

where u^* denotes the exact solution, and u is the numerical solution. Also, x_k and y_k for $k \in \{1, 2, 3, \dots, H\}$ are the grid points defined by each zero of the Chebyshev polynomial.

Example 4.1 ([24]). We consider the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \quad (x, y, t) \in (0, 2\pi) \times (0, 2\pi) \times (0, T],$$

subject to the initial condition

$$u(x, y, 0) = \sin(x) \sin(y), \quad (x, y) \in [0, 2\pi] \times [0, 2\pi],$$

with non-local boundary conditions

$$\begin{aligned} u(0, y, t) &= \frac{1}{4\pi} \int_0^{2\pi} \int_0^{2\pi} u(\xi, \eta, t) d\xi d\eta, & y \in [0, 2\pi], \\ u(2\pi, y, t) &= \frac{1}{4\pi} \int_0^{2\pi} \int_0^{2\pi} u(\xi, \eta, t) d\xi d\eta, & y \in [0, 2\pi], \\ u(x, 0, t) &= \frac{1}{4\pi} \int_0^{2\pi} \int_0^{2\pi} u(\xi, \eta, t) d\xi d\eta, & x \in [0, 2\pi], \\ u(x, 2\pi, t) &= \frac{1}{4\pi} \int_0^{2\pi} \int_0^{2\pi} u(\xi, \eta, t) d\xi d\eta, & x \in [0, 2\pi], \end{aligned}$$

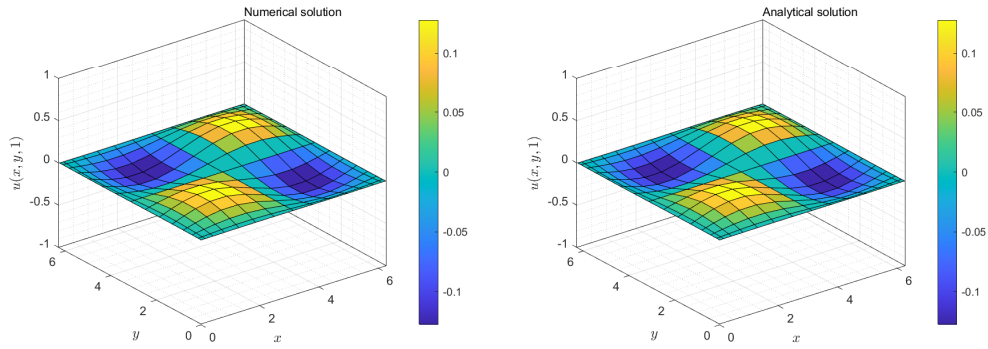
for $k(x, y, \xi, \eta) = \frac{1}{4\pi}$ where k is a positive constant. The analytical solution to this problem is given by $u^*(x, y, t) = e^{-2t} \sin(x) \sin(y)$.

This problem is investigated under non-local boundary conditions and approximate solutions can be computed by employing the numerical Algorithm 3. Upon the execution of Algorithm 3, we obtain numerical solutions denoted as $\mathbf{u}^{(m)}$. The precision of these solutions at time $T = 1$, measured by the ARE, is tabulated in Table 4.1 for $H := M \times N \in \{10 \times 10, 15 \times 15, 20 \times 20, 25 \times 25, 30 \times 30\}$, time step $\Delta t \in \{0.1, 0.05, 0.01, 0.005\}$, and $k = 0.1$. The table indicates that the errors minimally reduce as the nodal number $M \times N$ increases, while a significant reduction is observed as Δt decreases.

Lastly, Figure 4.1 presents the graphical solutions obtained from Algorithm 3 at time $T = 1$ for all domain $[0, 2\pi] \times [0, 2\pi]$ by using nodal point $H = 15 \times 15$ and time step $\Delta t = 0.01$, as well as the comparison of these solutions with the analytical solution.

Table 4.1: The ARE of numerical solutions u at time $T = 1$ for Example 4.1.

$M \times N$	$\Delta t = 0.1$	$\Delta t = 0.05$	$\Delta t = 0.01$	$\Delta t = 0.005$
10×10	2.5065×10^{-4}	6.5722×10^{-5}	2.6023×10^{-6}	9.4076×10^{-7}
15×15	2.3572×10^{-4}	6.1750×10^{-5}	2.3746×10^{-6}	5.9078×10^{-7}
20×20	2.4452×10^{-4}	6.4056×10^{-5}	2.4631×10^{-6}	6.1273×10^{-7}
25×25	2.4011×10^{-4}	6.2900×10^{-5}	2.4187×10^{-6}	6.0166×10^{-7}
30×30	2.4344×10^{-4}	6.3711×10^{-5}	2.4522×10^{-6}	6.1003×10^{-7}



(a) Our numerical solution from Algorithm 3 (b) Compared to analytical solution

Figure 4.1: Graphical numerical solutions $u(x, y, t)$ from Algorithm 3 for Example 4.1.

Example 4.2. We consider the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \quad (x, y, t) \in \left(0, \frac{\pi}{\sqrt{10}}\right) \times \left(0, \frac{\pi}{\sqrt{10}}\right) \times (0, T], \quad (4.23)$$

subject to the initial condition

$$u(x, y, 0) = \cos(x) \cos(y), \quad (x, y) \in \left[0, \frac{\pi}{\sqrt{10}}\right] \times \left[0, \frac{\pi}{\sqrt{10}}\right], \quad (4.24)$$

with non-local boundary conditions

$$\begin{aligned} u(0, y, t) &= \int_0^{\frac{\pi}{\sqrt{10}}} \int_0^{\frac{\pi}{\sqrt{10}}} k(0, y, \xi, \eta) u(\xi, \eta, t) d\xi d\eta, & y \in \left[0, \frac{\pi}{\sqrt{10}}\right], \\ u\left(\frac{\pi}{\sqrt{10}}, y, t\right) &= \int_0^{\frac{\pi}{\sqrt{10}}} \int_0^{\frac{\pi}{\sqrt{10}}} k\left(\frac{\pi}{\sqrt{10}}, y, \xi, \eta\right) u(\xi, \eta, t) d\xi d\eta, & y \in \left[0, \frac{\pi}{\sqrt{10}}\right], \\ u(x, 0, t) &= \int_0^{\frac{\pi}{\sqrt{10}}} \int_0^{\frac{\pi}{\sqrt{10}}} k(x, 0, \xi, \eta) u(\xi, \eta, t) d\xi d\eta, & x \in \left[0, \frac{\pi}{\sqrt{10}}\right], \\ u\left(x, \frac{\pi}{\sqrt{10}}, t\right) &= \int_0^{\frac{\pi}{\sqrt{10}}} \int_0^{\frac{\pi}{\sqrt{10}}} k\left(x, \frac{\pi}{\sqrt{10}}, \xi, \eta\right) u(\xi, \eta, t) d\xi d\eta, & x \in \left[0, \frac{\pi}{\sqrt{10}}\right], \end{aligned}$$

for the boundary kernels defined as $k(x, y, \xi, \eta) = \frac{5}{2} \cos\left(\sqrt{\frac{5}{2}}x\right) \cos\left(\sqrt{\frac{5}{2}}y\right)$, the analytical solution takes the form $u^*(x, y, t) = e^{-5t} \cos\left(\sqrt{\frac{5}{2}}x\right) \cos\left(\sqrt{\frac{5}{2}}y\right)$.

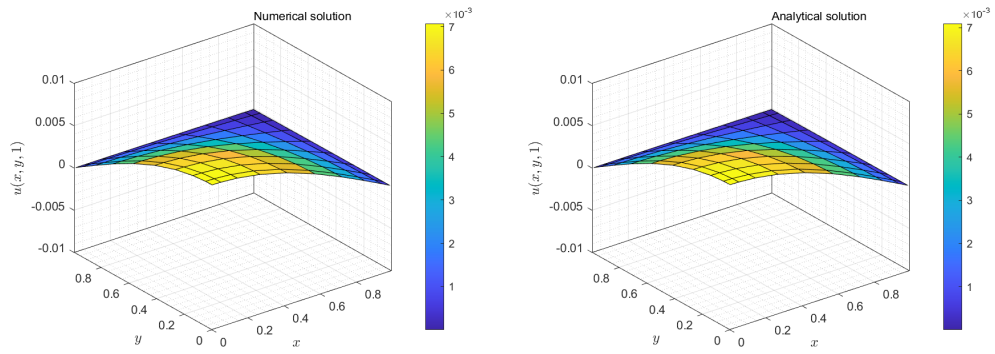
The problem is posed within the framework of non-local boundary conditions. Thus, Algorithm 3 can be utilized to derive approximate solutions. The application of Algorithm

3 provides numerical solutions denoted by $\mathbf{u}^{(m)}$. The precision of these solutions at time $t = 1$ evaluated via the ARE is reported in Table 4.2 for $H := M \times N \in \{10 \times 10, 15 \times 15, 20 \times 20, 25 \times 25, 30 \times 30\}$ and $\Delta t \in \{0.1, 0.05, 0.01, 0.005\}$. It can be observed from the table that the errors tend to slightly diminish with an increase in nodal number $M \times N$, but exhibit a more pronounced reduction as Δt becomes smaller.

Ultimately, the visualizations of solutions derived from Algorithm 3 are portrayed in Figure 4.2. These solutions are achieved at time $T = 1$ across the entire domain $[0, \frac{\pi}{\sqrt{10}}] \times [0, \frac{\pi}{\sqrt{10}}]$, employing a grid of $H = 15 \times 15$ nodes and a time increment $\Delta t = 0.01$. Furthermore, this figure incorporates the comparison between these computed solutions and the corresponding analytical solutions.

Table 4.2: The ARE of numerical solutions u at time $T = 1$ for Example 4.2.

$M \times N$	$\Delta t = 0.1$	$\Delta t = 0.05$	$\Delta t = 0.01$	$\Delta t = 0.005$
10×10	2.5508×10^{-4}	6.7992×10^{-5}	7.2715×10^{-6}	5.3764×10^{-6}
15×15	2.5157×10^{-4}	7.8482×10^{-5}	3.6233×10^{-6}	1.6074×10^{-6}
20×20	2.5098×10^{-4}	7.7732×10^{-5}	2.9573×10^{-6}	9.5263×10^{-7}
25×25	2.5082×10^{-4}	7.7527×10^{-5}	2.7750×10^{-6}	7.7343×10^{-7}
30×30	2.5076×10^{-4}	7.7453×10^{-5}	2.7096×10^{-6}	7.0970×10^{-7}



(a) Our numerical solution from Algorithm 3

(b) Compared to analytical solution

Figure 4.2: Graphical numerical solutions $u(x, y, t)$ from Algorithm 3 for Example 4.2.

CHAPTER V

CONCLUSION

5.1 Conclusion and discussion

Throughout this research, we have leveraged the FIM-CPE in conjunction with the Crank-Nicolson method to establish numerical algorithms capable of resolving one-dimensional heat equations (2.14) constrained by the initial condition (2.15), as well as non-local boundary conditions (2.16)–(2.17), and the Robin boundary conditions (3.20)–(3.21). These methodologies are presented as Algorithms 1 and 2, respectively. Additionally, Algorithm 3 has been developed to address the two-dimensional heat equation (2.20), subjected to the initial condition (2.21) and non-local boundary conditions (2.22)–(2.25).

In the context of these algorithms, the derivative associated with the temporal variable has been managed via the Crank-Nicolson method, which results in a time complexity of $\mathcal{O}(\Delta t^2)$. For the spatial variable, we adopt the solution via Chebyshev expansion and discard all derivatives in relation to the spatial variable by performing integrals within the heat equations, encompassing both one and two dimensions, and taking into account both non-local and Robin boundary conditions. Subsequently, we approximate the remaining integration terms using the Chebyshev integration matrix, based on FIM-CPE. The specifics of this process are outlined in the sections of Chapters III and IV pertaining to numerical algorithms.

Furthermore, the efficacy and precision of the proposed Algorithms 1–3 are highlighted through numerous experimental examples within the numerical examples sections of Chapters III and IV. The resulting data clearly affirms that these algorithms exhibit increased accuracy in terms of ARE and MAE as the number of nodes increases and the time step Δt decreases, especially with regards to Δt .

5.2 Future research directions

As part of our ongoing commitment to scholarly advancement in this field, we have identified several potential avenues for future research. Our investigations have unveiled exciting possibilities that could be explored to enhance our understanding and augment our existing body of knowledge. Below, we delineate these prospective research directions:

- Our current research invites the application of FIM-CPE to multi-dimensional non-linear heat equations. This extension will enable us to probe deeper into the intricate dynamics of these systems, thus further validating and enriching the robustness of the FIM-CPE methodology.
- We anticipate that our FIM-CPE approach can be further optimized for the resolution of heat equations under various non-classical boundary conditions. This broadened exploration has the potential to foster a more nuanced comprehension of heat equations across a wider array of complex boundary conditions.
- Our work consider the heat equations with non-classical boundary conditions on the rectangular domain. Then, we can extend on the heat equations with non-classical boundary conditions on the non-rectangular domain.

Pursuing these promising directions will enable us to perpetuate our contribution to this field and perpetuate the development of increasingly effective and versatile numerical solutions to heat equations.

REFERENCES

- [1] S. O. Adesanya and J. A. Falade. A new analytical method for the solutions of a non-linear heat-like equation in a finite medium. *Global Journal of Pure and Applied Mathematics*, 10(4):479–490, 2014.
- [2] B. Amaziane, M. Jurak, and H. Zakerzadeh. One-dimensional non-local model of heat transfer in porous media: application to geothermal reservoirs. *Nonlinear Analysis: Real World Applications*, 12(1):745–764, 2011.
- [3] R. Boonklurb and A. Duangpan. Finite integration method using Chebyshev expansion for solving nonlinear Poisson equations on irregular domains. *Journal of Numerical Analysis, Industrial and Applied Mathematics*, 14(1–2):7–24, 2020.
- [4] R. Boonklurb, A. Duangpan, and P. Gugaew. Numerical solution of direct and inverse problems for time-dependent Volterra integro-differential equation using finite integration method with shifted Chebyshev polynomials. *Symmetry*, 12(4):497, 2020.
- [5] R. Boonklurb, A. Duangpan, U. Rakwongwan, and P. Sutthimat. A novel analytical formula for the discounted moments of the ECIR process and interest rate swaps pricing. *Fractal and Fractional*, 6(2):58, 2022.
- [6] R. Boonklurb, A. Duangpan, and A. Saengsitongchai. Finite integration method via Chebyshev polynomial expansion for solving 2-D linear time-dependent and linear space-fractional differential equations. *Thai Journal of Mathematics*, pages 103–131, 2020.
- [7] R. Boonklurb, A. Duangpan, and T. Treeyaprasert. Modified finite integration method using Chebyshev polynomial for solving linear differential equations. *Journal of Numerical Analysis, Industrial and Applied Mathematics*, 12(3–4):1–19, 2018.
- [8] J. R. Cannon and J. van der Hoek. Diffusion subject to the specification of mass. *Journal of Mathematical Analysis and Applications*, 115(2):517–529, 1986.
- [9] W. Day. Extensions of a property of the heat equation to linear thermoelasticity and other theories. *Quarterly of Applied Mathematics*, 40(3):319–330, 1982.
- [10] A. Duangpan. Solving differential equations by finite integration method with orthogonal functions. Master’s thesis, Chulalongkorn University, 2018.

- [11] A. Duangpan and R. Boonklurb. Modified finite integration method using Chebyshev polynomial expansion for solving one-dimensional nonlinear Burgers' equations with shock wave. *Thai Journal of Mathematics*, pages 63–73, 2021.
- [12] A. Duangpan and R. Boonklurb. Numerical solution of time-fractional Benjamin-Bona-Mahony-Burgers equation via finite integration method by using Chebyshev expansion. *Songklanakarin Journal of Science & Technology*, 43(3), 2021.
- [13] A. Duangpan, R. Boonklurb, K. Chumpong, and P. Sutthimat. Analytical formulas for conditional mixed moments of generalized stochastic correlation process. *Symmetry*, 14(5):897, 2022.
- [14] A. Duangpan, R. Boonklurb, and M. Juytai. Numerical solutions for systems of fractional and classical integro-differential equations via finite integration method based on shifted Chebyshev polynomials. *Fractal and Fractional*, 5(3):103, 2021.
- [15] A. Duangpan, R. Boonklurb, U. Rakwongwan, and P. Sutthimat. Analytical formulas using affine transformation for pricing generalized swaps in commodity markets with stochastic convenience yields. *Symmetry*, 14(11):2385, 2022.
- [16] A. Duangpan, R. Boonklurb, and T. Treeyaprasert. Finite integration method with shifted Chebyshev polynomials for solving time-fractional Burgers' equations. *Mathematics*, 7(12):1201, 2019.
- [17] G. Ekolin. Finite difference methods for a nonlocal boundary value problem for the heat equation. *BIT Numerical Mathematics*, 31:245–261, 1991.
- [18] P. Khan and D. Vieru. Non-local boundary layer feedback in stability analysis of a class of spatially distributed systems. *Automatica*, 53:135–142, 2015.
- [19] M. Li, C. S. Chen, Y. C. Hon, and P. H. Wen. Finite integration method for solving multi-dimensional partial differential equations. *Applied Mathematical Modelling*, 39(17):4979–4994, 2015.
- [20] M. Li, Z. L. Tian, Y. C. Hon, C. S. Chen, and P. H. Wen. Improved finite integration method for partial differential equations. *Engineering Analysis with Boundary Elements*, 64:230–236, 2016.
- [21] Y. Lin, S. Xu, and H.-M. Yin. Finite difference approximations for a class of non-local parabolic equations. *International Journal of Mathematics and Mathematical Sciences*, 20(1):147–163, 1997.

- [22] D. M. Ludwig and W. L. Miranker. The use of heat equations in tumor growth models. *SIAM Journal on Applied Mathematics*, 17(4):842–855, 1969.
- [23] J. C. Mason and D. C. Handscomb. *Chebyshev Polynomials*. CRC Press, 2002.
- [24] P. R. Sharma and G. Methi. Solution of two-dimensional parabolic equation subject to non-local boundary conditions using homotopy perturbation method. *Journal of Applied Computer Science and Mathematics*, 6(1):64–68, 2012.
- [25] E. Tohidi. Application of Chebyshev collocation method for solving two classes of non-classical parabolic PDEs. *Ain Shams Engineering Journal*, 6(1):373–379, 2015.
- [26] P. H. Wen, Y. C. Hon, M. Li, and T. Korakianitis. Finite integration method for partial differential equations. *Applied Mathematical Modelling*, 37(24):10092–10106, 2013.
- [27] H. Zhang and F. Ding. On the Kronecker products and their applications. *Journal of Applied Mathematics*, 2013:1–8, 2013.
- [28] R. Zolfaghari and A. Shidfar. Solving a parabolic pde with nonlocal boundary conditions using the Sinc method. *Numerical Algorithms*, 62:411–427, 2013.

BIOGRAPHY

Name Mr. Thanakorn Prasansri

Date of Birth February 22, 1999

Place of Birth Udon Thani, Thailand

Education B.Sc. (Applied Mathematics), King Mongkut Institute of Technology Ladkrabang, 2020

Publication

- T. Prasansri and R. Booklurb, Finite integration method with Chebyshev polynomial expansion for solving heat equation with non-classical boundary conditions, *Proceeding of The 27th Annual Meeting in Mathematics and International Conference in Number Theory and Applications (AMM & ICNA 2023)*, 2023.

