ขั้นตอนวิธีเชิงตัวเลขสำหรับสมการความร้อนที่มีการเคลื่อนตัวของขอบโดยระเบียบวิธีปริพันธ์อันตะร่วมกับการกระจายพหุนามเชบีเชฟ

นางสาววรัญญา วงษ์อุรา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2565

NUMERICAL ALGORITHM FOR HEAT EQUATION WITH MOVING BOUNDARY USING FINITE INTEGRATION METHOD WITH CHEBYSHEV POLYNOMIAL EXPANSION

Miss. Warunya Wong-u-ra

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

Thesis Title	NUMERICAL ALGORITHM FOR HEAT EQUATION WITH MOVING BOUNDARY USING FINITE INTEGRATION METHOD WITH CHEBYSHEV POLYNOMIAL EXPANSION
By	Miss. Warunya Wong-u-ra
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Associate Professor Ratinan Boonklurb, Ph.D.

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

. . . . Dean of the Faculty of Science

(Professor Polkit Sangvanich, Ph.D.)

THESIS COMMITTEE

. . . . Chairman

(Associate Professor Petarpa Boonserm, Ph.D.)

. . . . Thesis Advisor

(Associate Professor Ratinan Boonklurb, Ph.D.)

. . . . Examiner

(Associate Professor Khamron Mekchay, Ph.D.)

. . . . External Examiner

(Associate Professor Tawikan Treeyaprasert, Ph.D.)

วรัญญา วงษ์อุรา : ขั้นตอนวิธีเชิงตัวเลขสำหรับสมการความร้อนที่มีการเคลื่อนตัวของขอบโดยระเบียบวิธีปริพันธ์อันตะร่วมกับการกระจายพหุนามเชบีเชฟ. (NUMERICAL ALGORITHM FOR HEAT EQUATION WITH MOVING BOUNDARY USING FINITE INTEGRATION METHOD WITH CHEBYSHEV POLYNOMIAL EXPANSION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร. รตินันท์ บุญเคลือบ, 52 หน้า.

ในวิทยานิพนธ์ฉบับนี้ ได้สร้างขั้นตอนวิธีเชิงตัวเลขสำหรับหาผลเฉลยโดยประมาณของสมการความร้อนที่มีเงื่อนไขขอบเคลื่อนตัวบางแบบ ขั้นตอนวิธีนี้สามารถนำไปประยุกต์ใช้กับปัญหาค่าขอบเคลื่อนตัวในเฟสเดียวและเงื่อนไขขอบเคลื่อนตัวทั้งสองด้านได้ ขั้นตอนวิธีของเราคำนวณการกระจายของอุณภูมิในโดเมนของปัญหาค่าขอบเคลื่อนตัวแบบสเตฟานเฟสเดียวและเงื่อนไขขอบเคลื่อนตัวทั้งสองด้าน ทั้งนี้ยังคำนวณตำแหน่งของขอบที่เคลื่อนตัวในปัญหาสเตฟานเฟสเดียว การทดสอบความแม่นยำของขั้นตอนวิธีที่นำเสนอทำโดยการเปรียบเทียบผลที่ได้กับผลเฉลยวิเคราะห์ที่มีอยู่

ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	ลายมือชื่อนิสิต	....
		ลายมือชื่อ อ.ที่ปรึกษาหลัก	....
สาขาวิชา	คณิตศาสตร์ประยุกต์และวิทยาการคณนา		
ปีการศึกษา	2565		

6370085923 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE
KEYWORDS : MOVING BOUNDARY / STAFAN PROBLEM / FINITE INTEGRATION METHOD / CHEBYSHEV POLYNOMIAL EXPANSION

WARUNYA WONG-U-RA : NUMERICAL ALGORITHM FOR HEAT EQUATION WITH MOVING BOUNDARY USING FINITE INTEGRATION METHOD WITH CHEBYSHEV POLYNOMIAL EXPANSION. ADVISOR : ASSOC. PROF. RATINAN BOONKLURB, Ph.D., 52 pp.

In this thesis, numerical algorithms for solving the approximate solution of the heat equation together with some moving boundary conditions are constructed. The algorithms can be applied to the one-phase Stefan problem and two-sided moving boundary conditions. Our algorithms provide a distribution of temperature within the domain in the one-phase Stefan problem and two-sided moving boundary conditions and also approximate the position of the moving interfaces in the one-phase Stefan problem. The accuracy of the presented algorithms is examined by comparing the results obtained with existing analytic solutions.

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

ACKNOWLEDGEMENTS

First, I am deeply indebted to my thesis advisor, Associate Professor Dr. Ratinan Boonklurb, for giving encouragement, motivation and guidance. He has helped me during the time for writing research and this thesis until it was accomplished. I also would like to thank Dr. Ampol Duangpan for his motivation and help in finishing this thesis.

Moreover, I would like to express my sincere thanks to my committee members, Associate Professor Dr. Petarpa Boonserm, Associate Professor Dr. Khamron Mekchay and Associate Professor Dr. Tawikan Treeyaprasert, for their invaluable advice and comments on this thesis.

Finally, I am very grateful to all my friends and colleagues who stayed with me and provided their encouragement, relaxation, great suggestions and support in many ways during a hard time studying for my master's degree.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Motivation and literature surveys

Moving boundary problems, also well-known as the Stefan problems, occur in many processes of physics and engineering. Especially, the processes regarding phase changes of materials which are caused by the heat transfer to and from both phases on each side of the interface. As a result, these yield a freezing process if the net heat is subtracted from the liquid part of the interface and a melting process when the net heat is added to the solid part. Actually, the moving boundary problems can be used to understand severally real-world applications such as renewable energy using latent heat storage systems, crystal growth of semiconductors and materials, welding and casting technology, freezing and thawing of foods, production of ice, ice formation on the pipe surface, solidification of alloys, etc., see [17, 18, 20] for more information and references therein.

There are interesting issues in the form of the heat equation with a one-phase Stefan problem and a two-sided moving boundary condition. This problem involves transient heat conduction and a phase change, often referred to as the freezing or melting problem. Mathematically, the solution to such a problem is inherently difficult because of the nonlinearity of the interface conditions.

The first interesting issue of the one-phase Stefan problem was introduced in [20]. The interface region between the liquid and solid phases is also moving as the latent heat is released and absorbed at the interface depicted in Figure 1.1. Therefore, the location of the interface, that is, the boundary of the domain, is also unknown. This means that the problem also has to be solved under the unknown domain. We can see that, in practice, there are many limitations to obtaining analytical solutions to such problems. Hence, the numerical solution has become the principal tool for studying the moving boundary problem. Furthermore, there are many numerical schemes to obtain an approximate solution to the moving boundary problem such as the variable space grid method which was given by Murray and Landis [17], the boundary immobilization method which was given by Crank [8], perturbation technique which was presented in [6, 25], an integral iterative formulation which was shown in [22, 23], and so on.

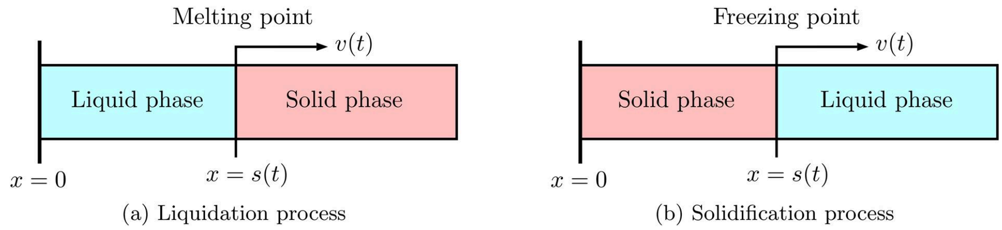


Figure 1.1: Physical configurations for the phase change problem.

Another interesting issue is the two-sided moving boundary conditions where the problem was introduced in [12]. The equation appeared in [12] has an additional coefficient term and two moving boundaries depending on time.

However, in this thesis, we will apply the other numerical method to address these two problems, namely, the finite integration method with Chebyshev polynomial expansion (FIM-CPE), which was proposed by [4] and has been successfully applied to handle various problems. For specific examples, please refer to [2, 3, 5].

Therefore, in this thesis, we propose numerical algorithms for solving the heat conduction equation with a one-phase Stefan problem and two-sided moving boundary conditions. The presented algorithm is designed based on the FIM-CPE to handle the spatial variable, combined with the use of a difference quotient to address the temporal variable. The accuracy of this algorithm is verified by comparing the obtained temperature distributions and locations of the moving interface with results acquired from some existing methods and analytical solutions via three examples of the one-phase Stefan problem. One of them involves a heat source or sink, while the second one consists of an exponentially increasing heat flux at the boundary. The third one consists of a fixed boundary and no forcing term. Together with two examples of two-sided moving boundary conditions, in which the first one involves linear coefficients, whereas the second one involves nonlinear coefficients. Evidently, our numerical algorithms can efficiently and accurately predict the evolution of the temperature distribution as well as the position of each moving front for one phase Stefan problem.

1.2 Research objectives

The goal of this research is to apply the FIM-CPE to construct an accurate numerical solution of the heat equation with a one-phase Stefan problem and two-sided moving boundary conditions.

1.3 Thesis overview

This thesis is separated into five chapters organized as follows. Chapter I is an introduction to this work including the motivation and introduction of the problem, the research objectives and the thesis overview. Chapter II presents the background knowledge used in this thesis, which includes Chebyshev polynomials, and heat equations with one-phase Stefan problem and two-sided moving boundary conditions. In Chapter III, we use the FIM-CPE for solving heat equations with moving boundaries. Also, we present the procedure and some examples for solving one-phase Stefan problems. In Chapter IV, we present the procedure and some examples for solving two-sided moving boundary conditions. Finally, in Chapter V, a discussion of our results is provided, some conclusions are drawn and possible future research is suggested.

CHAPTER II

PRELIMINARIES

In this chapter, we introduce the background knowledge about the definition and properties of the Chebyshev polynomials which play an important role in the FIM-CPE. We also present a heat equation with moving boundary conditions in the one-dimensional domain which is the main problem to be solved by the FIM-CPE. First, let us introduce the Chebyshev polynomials and some useful facts about them.

2.1 Chebyshev polynomials

The Chebyshev polynomials are a set of orthogonal polynomials which play an important role in the interpolation problem. Their roots are used as nodes to calculate a polynomial interpolation which provides the best polynomial approximation under the maximum norm. Normally, the Chebyshev polynomial is defined over $[-1, 1]$. However, in this thesis, we use the Chebyshev polynomial which is defined over $[a, b]$ instead.

Definition 2.1 ([1])**.** The Chebyshev polynomial of degree $n \geq 0$ is defined by

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

The first few Chebyshev polynomials $R_n(x)$ over $[-1, 1]$ are shown in Figure 2.1 for $n \in \{0, 1, 2, 3, 4, 5\}$.

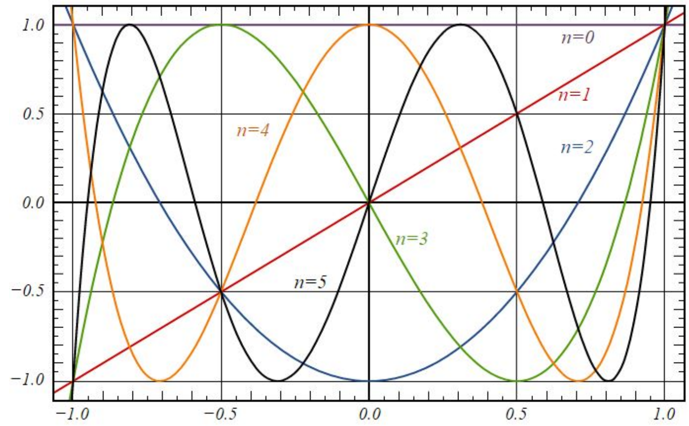


Figure 2.1: Chebyshev polynomials $R_n(x)$ for $n \in \{0, 1, 2, 3, 4, 5\}$.

The followings are the key properties of $R_n(x)$ where the proofs can be reproduced by using a similar idea as shown in Lemma [9].

Lemma 2.1. *The Chebyshev polynomial $R_n(x)$ satisfies the following properties:*

(i) *The zeros of Chebyshev polynomial $R_n(x)$ for $n \in \mathbb{N}$ and $x \in [a, b]$ are*

$$x_k = \frac{1}{2}\left((b-a)\cos\left(\frac{2k-1}{2n}\pi\right) + a + b\right), \ k \in \{1, 2, 3, \ldots, n\}. \tag{2.1}$$

(ii) *The r^{th} order derivatives of $R_n(x)$ at the end points $x = a$ and $x = b$ are*

$$\frac{d^r}{dx^r} R_n(x)\Big|_{x\in\{a,b\}} = (2x-1)^{r+n} \prod_{k=0}^{r-1}\left(\frac{2n^2 - 2k^2}{2k+1}\right). \tag{2.2}$$

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

$$\begin{aligned}
\bar{R}_0(x) &= \int_0^x R_0(\xi)\, d\xi = x - a, \\
\bar{R}_1(x) &= \int_0^x R_1(\xi)\, d\xi = \frac{(x-a)(x-b)}{b-a}, \\
\bar{R}_n(x) &= \int_0^x R_n(\xi)\, d\xi = \frac{b-a}{4}\left(\frac{R_{n+1}(x)}{n+1} - \frac{R_{n-1}(x)}{n-1} - \frac{2(-1)^n}{n^2-1}\right).
\end{aligned} \tag{2.3}$$

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

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

where x_k be the zeros of $R_n(x)$ defined in (2.1) *and $i, j \in \{0, 1, 2, \ldots, n\}$.*

(v) *The Chebyshev matrix $\mathbf{R}$ at each point x_k defined by* (2.1) *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, it has the multiplicative inverse

$$\mathbf{R}^{-1} = \frac{1}{n}\text{diag}\{1, 2, 2, \ldots, 2\}\,\mathbf{R}^{\top}. \tag{2.4}$$

(vi) *The recurrence relation of Chebyshev polynomials $R_{n-1}(x), R_n(x)$ and $R_{n+1}(x)$ is*

$$R_{n+1}(x) = 2\left(\frac{2x - a - b}{b - a}\right) R_n(x) - R_{n-1}(x)$$

with starting from the values $R_0(x) = 1$ and $R_1(x) = \frac{2x-a-b}{b-a}$.

2.2 Statement of heat equation with one-phase Stefan problem

In this thesis, we are interested in the temperature distribution $u(x, t)$ and the position of the moving boundary or moving interface or moving front $s(t)$ for the liquidation or solidification process as shown in Figure 1.1 over a semi-infinite slab $0 \leq x < \infty$ of material and time $t > 0$. The concerning problem is the one-phase Stefan problem with a forcing term $f(x, t)$ which was given in [20]. It is governed by the following heat conduction equation in the first phase, namely,

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} + f(x,t), \quad 0 < x < s(t),\ t > 0 \tag{2.5}$$

with a uniform temperature k, which refers to the traditional temperature of a material, beyond the melting or freezing point $s(t)$ within the second phase, that is,

$$u(x,t) = k, \quad x > s(t),\ t > 0, \tag{2.6}$$

where x is a spatial variable, t is a temporal variable, k is a constant, $u(x,t)$ is a temperature distribution, $f(x,t)$ is a forcing term acting on the solid or liquid region, which is sufficiently smooth and nonnegative, and $s(t)$ is a position of moving front at time t.

From (2.5) and (2.6), we can see that this problem involves moving boundary conditions at the interface $x = s(t)$. In this case, we consider two boundary conditions at $x = s(t)$. The first one provides the temperature at the interface $x = s(t)$ equivalent to the traditional temperature of the considered material which is

$$u(x,t) = k \ \text{ at } \ x = s(t) \ \text{ for } \ t > 0. \tag{2.7}$$

The second one locates the interface itself through a relationship defining the front velocity $v(t)$. It is actually the heat balance equation known as the Stefan condition [20], i.e.,

$$v(t) = \frac{ds(t)}{dt} = -\frac{\partial u(x,t)}{\partial x} \ \text{ at } \ x = s(t) \ \text{ for } \ t > 0. \tag{2.8}$$

In addition, this problem requires two more conditions for the solution of the second-order partial differential equation (PDE) in (2.5). First, the initial temperature at $t = 0$ is defined by

$$u(x,0) = g(x), \quad 0 \le x \le s(0), \tag{2.9}$$

where $g(x)$ is a prescribed function that is sufficiently smooth and nonnegative. Next, for another condition, we need a fixed boundary condition (FBC) at $x = 0$ with respect to a function $u(x,t)$. However, this FBC can vary depending on the problem considered.

2.3 Statement of heat equation with two-sided moving boundary condition

Consider the one-dimensional two-sided moving boundary condition given in [12]:

$$\frac{\partial u}{\partial t} = a(x,t)\frac{\partial^2 u}{\partial x^2} + b(x,t)\frac{\partial u}{\partial x} + c(t)u + f(x,t), \quad (x,t) \in \Omega_T, \tag{2.10}$$

where the free domain $\Omega_T = \{(x,t) : h_1(t) < x < h_2(t), 0 < t < T\}$, subject to the initial condition

$$u(x,0) = g(x), \quad x \in [h_1(0), h_2(0)] \tag{2.11}$$

and the non-homogeneous Dirichlet boundary conditions which are

$$u(h_1(t),t) = \mu_1(t), \quad u(h_2(t),t) = \mu_2(t), \quad t \in [0,T], \tag{2.12}$$

where $u(x,t)$ is the temperature distribution, while $g(x)$, $\mu_1(t)$, and $\mu_2(t)$ are theoretically verified as continuous functions by [21], the coefficients $a(x,t), b(x,t), c(t),\ f(x,t), h_1(t)$ and $h_2(t)$ have been reported to exhibit continuity [15].

We will utilize the properties described in Section 2.1 to develop an algorithm based on Chebyshev polynomial expansion. This algorithm will be used to solve the problems discussed in Sections 2.2 and 2.3, which are of our interest.

CHAPTER III

FIM-CPE FOR ONE-PHASE STEFAN PROBLEM

In this chapter, we describe the technique of FIM-CPE for one-dimensional and construct the Chebyshev integration matrix to manipulate the derivative with respect to the spatial variable in (2.5) and (2.6). Then, based on this FIM-CPE, we can devise a numerical algorithm for solving the heat equation with moving boundary conditions as stated in Section 2.2. Finally, our numerical algorithm is also provided.

3.1 The FIM-CPE in one dimension

We construct the Chebyshev integration matrix, which is the main tool for dealing with the integral term. Let $M \in \mathbb{N}$. We would like to approximate the solution of the problem in Section 2.2 which depends on the spatial variable in terms of a function $w(x)$ that can be expressed by the Chebyshev polynomial expansion as follows

$$w(x) = \sum_{n=0}^{M-1} c_n R_n(x), \quad \text{for } \ x \in [a, b], \tag{3.1}$$

where c_n is unknown coefficients to be considered. Next, let x_k be grid points generated by the zeros of the Chebyshev polynomial R_M defined by (2.1) in ascending order. When we substitute each x_k into (3.1), those equations can be expressed in the matrix form

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

which is denoted by $\mathbf{w} = \mathbf{R}\mathbf{c}$. Since $\mathbf{R}$ is invertible by Lemma 2.1(v), we have $\mathbf{c} = \mathbf{R}^{-1}\mathbf{w}$. Next, we consider the single integral of $w(x)$ from a to x_k for $x_k \in [a, b]$, denoted $W^{(1)}(x_k)$, to obtain

$$W^{(1)}(x_k) := \int_a^{x_k} w(\xi)\,d\xi = \sum_{n=0}^{M-1} c_n \int_a^{x_k} R_n(\xi)\,d\xi = \sum_{n=0}^{M-1} c_n \bar{R}_n(x_k),$$

where $\bar{R}_n$ is denoted to be a single-layer integration of R_n that can be directly obtained by (2.3) depending on its degree n. After substituting each node x_k into the above equation, it can be written in the matrix form

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

which is denoted by $\mathbf{W}^{(1)} = \overline{\mathbf{R}}\mathbf{c} = \overline{\mathbf{R}}\mathbf{R}^{-1}\mathbf{w} := \mathbf{A}\mathbf{w}$, where $\mathbf{A} = \overline{\mathbf{R}}\mathbf{R}^{-1} := [a_{ki}]_{M\times M}$ is the integral operational matrix that is called the Chebyshev integration matrix. Thus, it can be also expressed in another form

$$W^{(1)}(x_k) = \int_a^{x_k} w(\xi)\,d\xi = \sum_{i=1}^{M} a_{ki} w(x_i). \tag{3.2}$$

for varying each zero $x_k, k \in \{1, 2, 3, \ldots, M\}$ to the above equation, the matrix form can be written as the following

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

After that, we consider the double-layer integration of $w(x)$ from a to x_k, denoted by

$W^{(2)}(x_k)$, by using (3.2). Then, we obtain

$$
\begin{aligned}
W^{(2)}(x_k) &:= \int_a^{x_k} \int_a^{\xi_2} w(\xi_1)\, d\xi_1 d\xi_2 \\
&= \int_a^{x_k} W^{(1)}(\xi_2) d\xi_2 \\
&= \sum_{i=1}^{M} a_{ki} W^{(1)}(x_i) \\
&= \sum_{l=1}^{M} \sum_{i=1}^{M} a_{ki} a_{il} w(x_l) \\
&= \sum_{l=1}^{M} [\mathbf{A}^2]_{kl} w(x_l).
\end{aligned}
$$

When we vary the zeros x_k for $k \in \{1, 2, 3, \ldots, M\}$ in the above $W^{(2)}(x_k)$, each equation can be combined and written in the matrix form $\mathbf{W}^{(2)} = \mathbf{A}^2\mathbf{w}$ which represents the integral matrix for double-layer integration of $w(x)$.

Similarly, by using the mathematical induction, we have the m multiple-layer integration of $w(x)$ from a to the zero x_k, denoted by $W^{(m)}(x_k)$, as follows

$$
\begin{aligned}
W^{(m)}(x_k) &:= \int_a^{x_k} \int_a^{\xi_m} \cdots \int_a^{\xi_3} \int_a^{\xi_2} w(\xi_1)\, d\xi_1 d\xi_2 \cdots d\xi_{m-1} d\xi_m \\
&= \int_a^{x_k} W^{(m-1)}(\xi_m) d\xi_m \\
&= \sum_{i=1}^{M} a_{ki} W^{(m-1)}(x_i) \\
&= \sum_{l=1}^{M} \sum_{i=1}^{M} a_{ki} [\mathbf{A}^{m-1}]_{il} w(x_l) \\
&= \sum_{l=1}^{M} [\mathbf{A}^m]_{kl} w(x_l).
\end{aligned}
$$

When the zeros x_k for $k \in \{1, 2, 3, \ldots, M\}$ are distributed in the above equation, each equation can be combined and expressed in the matrix form $\mathbf{W}^{(m)} = \mathbf{A}^m\mathbf{u}$ which is the matrix representation for m multiple-layer integration of $w(x)$ in the FIM-CPE.

3.2 Procedure for solving one-phase Stefan problem

In this part, the numerical algorithm based on the FIM-CPE explained in Section 3.1 is devised for solving the heat equation with one phase moving boundary as stated in Section 2.2. First, by using the idea of [20], let us use the spatial coordinate transformation $\eta = \frac{x}{s(t)}$. We obtain the new coordinate system (η, t) and the moving front is fixed at $\eta = 1$. Let us define the solution $u(x,t) = w(\eta,t)$ which corresponds to the new coordinate. Then, by employing the chain rule of partial derivatives, we get

$$\frac{\partial u}{\partial x} = \frac{1}{s}\frac{\partial w}{\partial \eta}, \quad \frac{\partial^2 u}{\partial x^2} = \frac{1}{s^2}\frac{\partial^2 w}{\partial \eta^2} \quad \text{and} \quad \frac{\partial u}{\partial t} = \frac{\partial w}{\partial t} - \frac{\eta}{s}\frac{ds}{dt}\frac{\partial w}{\partial \eta}.$$

Thus, by using the above partial differential relations, the considered problem in Section 2.2 given by (2.5)–(2.9) with the transformation $\eta = \frac{x}{s(t)}$ can be rewritten as follows.

$$\frac{\partial w}{\partial t} = \frac{\eta}{s}\frac{ds}{dt}\frac{\partial w}{\partial \eta} + \frac{1}{s^2}\frac{\partial^2 w}{\partial \eta^2} + f(\eta s, t), \quad (\eta, t) \in (0,1) \times (0,T], \tag{3.3}$$

$$\frac{ds}{dt} = -\frac{1}{s}\frac{\partial w}{\partial \eta} = v(t), \quad (\eta, t) \in \{1\} \times (0,T], \tag{3.4}$$

where $w = w(\eta, t)$, $s = s(t)$ and $T \in \mathbb{R}^+$ is denoted to be a terminal time. Their initial condition is $w(\eta, 0) = g(\eta s_0)$ for $\eta \in [0,1]$ where $s_0 = s(0)$. The boundary conditions are $w(1,t) = k$ and the FBC at $\eta = 0$ for $t \in (0,T]$. Next, let us construct a numerical algorithm. We start from uniformly discretizing the temporal domain $(0,T]$ by specifying each time point $t_m = m\Delta t$ for $m \in \mathbb{N}$ into (3.3) and (3.4), where Δt is a given time step. Moreover, in order to find the moving front $s(t_m)$, we linearize the nonlinear term of (3.4) by letting the functions s and w be at the previous time t_{m-1}. Then, we have

$$\frac{\partial w^{\langle m\rangle}(\eta)}{\partial t} = \frac{\eta v^{\langle m\rangle}}{s^{\langle m\rangle}}\frac{\partial w^{\langle m\rangle}(\eta)}{\partial \eta} + \frac{1}{(s^{\langle m\rangle})^2}\frac{\partial^2 w^{\langle m\rangle}(\eta)}{\partial \eta^2} + f^{\langle m\rangle}(\eta s^{\langle m\rangle}), \tag{3.5}$$

$$\frac{ds^{\langle m\rangle}}{dt} = -\frac{1}{s^{\langle m-1\rangle}}\frac{\partial w^{\langle m-1\rangle}(\eta)}{\partial \eta}\Big|_{\eta=1} = v^{\langle m\rangle}, \tag{3.6}$$

where the functions with superscript $\langle m\rangle$ mean those functions are indicated at time t_m. After that, we approximate the derivative terms with respect to time t of (3.5) and (3.6) by applying the forward difference quotient which provides the time complexity $\mathcal{O}(\Delta t)$. Then, we have

$$\frac{w^{\langle m\rangle}(\eta)-w^{\langle m-1\rangle}(\eta)}{\Delta t}=\frac{\eta v^{\langle m\rangle}}{s^{\langle m\rangle}}\frac{\partial w^{\langle m\rangle}(\eta)}{\partial\eta}+\frac{1}{(s^{\langle m\rangle})^2}\frac{\partial^2 w^{\langle m\rangle}(\eta)}{\partial\eta^2}+f^{\langle m\rangle}(\eta s^{\langle m\rangle}), \tag{3.7}$$

$$\frac{s^{\langle m\rangle}-s^{\langle m-1\rangle}}{\Delta t}=-\frac{1}{s^{\langle m-1\rangle}}\frac{\partial w^{\langle m-1\rangle}(\eta)}{\partial\eta}\Big|_{\eta=1}=v^{\langle m\rangle}. \tag{3.8}$$

Now, we can see that our considered problem depends only on the spatial variable η. Hence, the FIM-CPE can be applied to the problem which assumes that a problem solution $w^{\langle m\rangle}(\eta)$ can be approximated by the Chebyshev polynomial expansion (3.1),

$$w^{\langle m\rangle}(\eta)=\sum_{n=0}^{M-1}c_n^{\langle m\rangle}R_n(\eta). \tag{3.9}$$

Then, by the idea of FIM-CPE, we eliminate all derivatives with respect to the space variable η out of (3.7) by taking double-layer integrals from 0 to $\eta_k\in(0,1)$ on both sides of (3.7), where η_k's are generated by the zeros of the Chebyshev polynomial R_M, as defined in (2.1). Thus, we obtain the equivalent integral equation that is used for the integration by parts as follows

$$\begin{aligned}&\int_0^{\eta_k}\int_0^{\xi_2}\left(\frac{w^{\langle m\rangle}(\xi_1)-w^{\langle m-1\rangle}(\xi_1)}{\Delta t}\right)d\xi_1 d\xi_2\\&=\frac{v^{\langle m\rangle}}{s^{\langle m\rangle}}\left(\int_0^{\eta_k}\xi_2 w^{\langle m\rangle}(\xi_2)\,d\xi_2-\int_0^{\eta_k}\int_0^{\xi_2}w^{\langle m\rangle}(\xi_1)\,d\xi_1 d\xi_2\right)\\&+\frac{w^{\langle m\rangle}(\eta_k)}{(s^{\langle m\rangle})^2}+\int_0^{\eta_k}\int_0^{\xi_2}f^{\langle m\rangle}(\xi_1 s^{\langle m-1\rangle})\,d\xi_1 d\xi_2+d_1\eta_k+d_2,\end{aligned} \tag{3.10}$$

where d_1 and d_2 are arbitrary constants that emerged from the process of integrations. Next, by substituting each zero $\eta_k\in\{\eta_1,\eta_2,\eta_3,\ldots,\eta_M\}$ into the integral equation (3.10), we can express them into the matrix form as

$$\frac{\mathbf{A}^2}{\Delta t}\left(\mathbf{w}^{\langle m\rangle}-\mathbf{w}^{\langle m-1\rangle}\right)=\frac{v^{\langle m\rangle}}{s^{\langle m\rangle}}\left(\mathbf{A}(\boldsymbol{\eta}\odot\mathbf{w}^{\langle m\rangle})-\mathbf{A}^2\mathbf{w}^{\langle m\rangle}\right)+\frac{\mathbf{w}^{\langle m\rangle}}{(s^{\langle m\rangle})^2}+\mathbf{A}^2\mathbf{f}^{\langle m\rangle}+d_1\boldsymbol{\eta}+d_2\mathbf{i}$$

which can be simplified to

$$\left(\frac{\mathbf{A}^2}{\Delta t} - \frac{v^{\langle m\rangle}}{s^{\langle m\rangle}}\left(\mathbf{A}\mathrm{diag}\{\boldsymbol{\eta}\} - \mathbf{A}^2\right) - \frac{\mathbf{I}}{(s^{\langle m\rangle})^2}\right)\mathbf{w}^{\langle m\rangle} - d_1\boldsymbol{\eta} - d_2\mathbf{i} = \frac{\mathbf{A}^2\mathbf{w}^{\langle m-1\rangle}}{\Delta t} + \mathbf{A}^2\mathbf{f}^{\langle m\rangle}, \tag{3.11}$$

where the operator $\odot$ is Hadamard product defined in [7], $\mathbf{A} = \overline{\mathbf{R}}\mathbf{R}^{-1}$ is the Chebyshev integration matrix defined in Section 3.1, $\mathbf{I}$ is an $M \times M$ identity matrix,

$$\begin{aligned}
\mathbf{i}^\top &= [1, 1, 1, \ldots, 1], \\
\boldsymbol{\eta}^\top &= [\eta_1, \eta_2, \eta_3, \ldots, \eta_M], \\
\mathbf{w}^{\langle m\rangle\,\top} &= [w(\eta_1, t_m), w(\eta_2, t_m), w(\eta_3, t_m), \ldots, w(\eta_M, t_m)] \quad \text{and} \\
\mathbf{f}^{\langle m\rangle\,\top} &= \left[f(\eta_1 s^{\langle m\rangle}, t_m), f(\eta_2 s^{\langle m\rangle}, t_m), f(\eta_3 s^{\langle m\rangle}, t_m), \ldots, f(\eta_M s^{\langle m\rangle}, t_m)\right].
\end{aligned}$$

Here, we observe that the matrix equation (3.11) does not account for the moving location $s^{\langle m\rangle}$ and front velocity $v^{\langle m\rangle}$, which can be estimated by utilizing (3.6) in conjunction with the Chebyshev polynomial expansion (3.9) and the differential relation (2.2). Therefore, it can be written in the matrix form as follows

$$\begin{aligned}
v^{\langle m\rangle} &= -\frac{1}{s^{\langle m-1\rangle}}\frac{\partial w^{\langle m-1\rangle}(\eta)}{\partial \eta}\bigg|_{\eta=1} \\
&= -\frac{1}{s^{\langle m-1\rangle}}\sum_{n=0}^{M-1} c_n^{\langle m-1\rangle} R_n'(1) \\
&= -\frac{2}{s^{\langle m-1\rangle}}\sum_{n=0}^{M-1} c_n^{\langle m-1\rangle} n^2 \\
&= -\frac{2\mathbf{q}^\top\mathbf{c}^{\langle m-1\rangle}}{s^{\langle m-1\rangle}} = -\frac{2\mathbf{q}^\top\mathbf{R}^{-1}\mathbf{w}^{\langle m-1\rangle}}{s^{\langle m-1\rangle}},
\end{aligned} \tag{3.12}$$

where $\mathbf{q}^\top = \left[0, 1, 4, 9, \ldots, (M-1)^2\right]$ and $\mathbf{R}^{-1}$ is defined in Lemma 2.1(v) with size $M \times M$. When we substitute the known values $s^{\langle m-1\rangle}$ and $\mathbf{w}^{\langle m-1\rangle}$ from previous time t_{m-1} into (3.12), we can obtain the front velocity $v^{\langle m\rangle}$. As a consequence, the moving location $s^{\langle m\rangle}$ can be approximated by using (3.8) and (3.12), that is

$$s^{\langle m\rangle} = s^{\langle m-1\rangle} + v^{\langle m\rangle}\Delta t. \tag{3.13}$$

However, we can see that (3.11) has unknown vectors apart from $\mathbf{w}^{\langle m\rangle}$, i.e., d_1 and d_2, which are emerged from the process of integrations. Thus, we require more two equations that are constructed from the given boundary conditions $w(1,t) = k$ and FBC at $\eta = 0$. At the time t_m, we can use (3.9) to change these conditions into the vector form as follows:

$$w^{\langle m\rangle}(1) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle} R_n(1) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle} = \mathbf{i}^\top \mathbf{c}^{\langle m\rangle} = \mathbf{i}^\top \mathbf{R}^{-1}\mathbf{w}^{\langle m\rangle} = k. \tag{3.14}$$

Another condition is FBC at $\eta = 0$, which depends on the considered problem. In this case, we provide two examples of the left FBC, namely, $u(0,t) = \phi_0(t)$ and $u_x(x,t)|_{x=0} = \phi_1(t)$. By using the spatial coordinate transformation $\eta = \frac{x}{s(t)}$, these conditions become $w(0,t) = \phi_0(t)$ and $w_\eta(\eta,t)|_{\eta=0} = s(t)\phi_1(t)$, respectively. Thus, at fixing time t_m, we apply (3.9) and (2.2) for conversing the obtained conditions into the vector forms as follows:

$$w^{\langle m\rangle}(0) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle} R_n(0) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle}(-1)^n = \boldsymbol{\ell}_0^\top \mathbf{c}^{\langle m\rangle} = \boldsymbol{\ell}_0^\top \mathbf{R}^{-1}\mathbf{w}^{\langle m\rangle} = \phi_0(t_m) \tag{3.15}$$

and

$$w_\eta^{\langle m\rangle}(0) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle} R_n'(0) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle}(-1)^{n+1}(2n^2) = \boldsymbol{\ell}_1^\top \mathbf{c}^{\langle m\rangle} = \boldsymbol{\ell}_1^\top \mathbf{R}^{-1}\mathbf{w}^{\langle m\rangle} = s^{\langle m\rangle}\phi_1(t_m), \tag{3.16}$$

where $\boldsymbol{\ell}_0^\top = \left[1, -1, 1, -1, \ldots, (-1)^{M-1}\right]$ and $\boldsymbol{\ell}_1^\top = \left[0, 2, -8, 18, \ldots, 2(-1)^M(M-1)^2\right]$. Note that, for other FBCs at $\eta = 0$, we can transform them into matrix forms similar to the process of being formed (3.15) and (3.16).

Now, we completely obtain all of the equations for solving $\mathbf{w}^{\langle m\rangle}$. Consequently, we can combine those equations given by (3.11), (3.14) and (3.15) or (3.16), which depend on the considered problem, into the matrix form of a linear system. Thus, we have two linear systems depending on the type of the left FBC considered as follows:

- **Case I:** The FBC is $u(0,t)=\phi_0(t)$,

$$\left[\begin{array}{c|cc} \mathbf{K}^{\langle m\rangle} & -\boldsymbol{\eta} & -\mathbf{i} \\ \hline \mathbf{i}^\top\mathbf{R}^{-1} & 0 & 0 \\ \boldsymbol{\ell}_0^\top\mathbf{R}^{-1} & 0 & 0 \end{array}\right]\left[\begin{array}{c} \mathbf{w}^{\langle m\rangle} \\ \hline d_1 \\ d_2 \end{array}\right]=\left[\begin{array}{c} \frac{\mathbf{A}^2\mathbf{w}^{\langle m-1\rangle}}{\Delta t}+\mathbf{A}^2\mathbf{f}^{\langle m\rangle} \\ \hline k \\ \phi_0(t_m) \end{array}\right]; \tag{3.17}$$

- **Case II:** The FBC is $u_x(x,t)|_{x=0}=\phi_1(t)$,

$$\left[\begin{array}{c|cc} \mathbf{K}^{\langle m\rangle} & -\boldsymbol{\eta} & -\mathbf{i} \\ \hline \mathbf{i}^\top\mathbf{R}^{-1} & 0 & 0 \\ \boldsymbol{\ell}_1^\top\mathbf{R}^{-1} & 0 & 0 \end{array}\right]\left[\begin{array}{c} \mathbf{w}^{\langle m\rangle} \\ \hline d_1 \\ d_2 \end{array}\right]=\left[\begin{array}{c} \frac{\mathbf{A}^2\mathbf{w}^{\langle m-1\rangle}}{\Delta t}+\mathbf{A}^2\mathbf{f}^{\langle m\rangle} \\ \hline k \\ s^{\langle m\rangle}\phi_1(t_m) \end{array}\right]; \tag{3.18}$$

where $\mathbf{K}^{\langle m\rangle} := \frac{\mathbf{A}^2}{\Delta t}-\frac{v^{\langle m\rangle}}{s^{\langle m\rangle}}\left(\mathbf{A}\mathrm{diag}\{\boldsymbol{\eta}\}-\mathbf{A}^2\right)-\frac{\mathbf{I}}{(s^{\langle m\rangle})^2}$. Accordingly, the solution $\mathbf{w}^{\langle m\rangle}$ can be approximated by solving the system (3.17) or (3.18) together with (3.12) and (3.13) that start from the given initial conditions ${\mathbf{w}^{\langle 0\rangle}}^\top = [g(\eta_1 s_0), g(\eta_2 s_0), g(\eta_3 s_0), \ldots, g(\eta_M s_0)]$ and $s^{\langle 0\rangle}=s_0$. Note that, upon performing the final iteration, the obtained numerical solutions $s^{\langle m\rangle}=s(T)$ and $\mathbf{w}^{\langle m\rangle}$ can be expressed as corresponding to the function $u(x,T)$ that is

$${\mathbf{w}^{\langle m\rangle}}^\top = \left[u(\eta_1 s^{\langle m\rangle},T), u(\eta_2 s^{\langle m\rangle},T), u(\eta_3 s^{\langle m\rangle},T), \ldots, u(\eta_M s^{\langle m\rangle},T),\right].$$

For computational convenience, we summarize all the above-mentioned procedures in terms of the pseudocode algorithm in order to find an approximate solution of the heat equation with moving boundary in Section 2.2 by using the FIM-CPE.

Algorithm 1 Numerical algorithm for solving the heat equation with moving boundary

Input: s_0, k, T, M, Δt, $g(x)$, $f(x,t)$ and $\phi_0(t)$ or $\phi_1(t)$;

Output: The approximate solutions $s^{\langle m\rangle}$ and $\mathbf{w}^{\langle m\rangle}$;

Set $\eta_k \leftarrow \frac{1}{2}\left(1+\cos\left(\frac{2k-1}{2M}\pi\right)\right)$ for $k \in \{1,2,3,\dots,M\}$ in ascending order;

Compute $\boldsymbol{\eta}$, $\mathbf{q}$, $\mathbf{i}$, $\mathbf{I}$, $\mathbf{R}$, $\overline{\mathbf{R}}$, $\mathbf{R}^{-1}$, $\mathbf{A}$ and $\boldsymbol{\ell}_0$ or $\boldsymbol{\ell}_1$;

Construct $s^{\langle 0\rangle} \leftarrow s_0$ and $\mathbf{w}^{\langle 0\rangle} \leftarrow [g(\eta_1 s_0), g(\eta_2 s_0), g(\eta_3 s_0), \dots, g(\eta_M s_0)]^\top$;

Set $m \leftarrow 1$ and $t_1 \leftarrow \Delta t$;

while $t_m \leq T$ **do**

    Compute $v^{\langle m\rangle} \leftarrow -\frac{2\mathbf{q}^\top \mathbf{R}^{-1}\mathbf{w}^{\langle m-1\rangle}}{s^{\langle m-1\rangle}}$;

    Compute $s^{\langle m\rangle} \leftarrow s^{\langle m-1\rangle} + v^{\langle m\rangle}\Delta t$;

    Compute $\mathbf{K}^{\langle m\rangle} \leftarrow \frac{\mathbf{A}^2}{\Delta t} - \frac{v^{\langle m\rangle}}{s^{\langle m\rangle}}\left(\mathbf{A}\mathrm{diag}\{\boldsymbol{\eta}\} - \mathbf{A}^2\right) - \frac{\mathbf{I}}{(s^{\langle m\rangle})^2}$;

    Compute $\mathbf{f}^{\langle m\rangle} \leftarrow \left[f(\eta_1 s^{\langle m\rangle}, t_m), f(\eta_2 s^{\langle m\rangle}, t_m), \dots, f(\eta_M s^{\langle m\rangle}, t_m)\right]^\top$;

    Find $\mathbf{w}^{\langle m\rangle}$ by solving the iterative linear system (3.17) or (3.18);

    Update $m \leftarrow m+1$;

    Compute $t_m \leftarrow m\Delta t$;

end while

return The final iteration of $s^{\langle m\rangle}$ and $\mathbf{w}^{\langle m\rangle}$;

3.3 Numerical examples for one-phase Stefan problem

In this section, we apply the proposed Algorithm 1 based on the FIM-CPE for finding numerical results of the heat equation with moving boundary in order to demonstrate its efficiency and accuracy by measuring with average relative error via three examples. Examples 3.1, 3.2 and 3.3 are the one-phase Stefan problems with forcing term, time-dependent heat flux, and fixed boundary and no forcing term, respectively. All the experiments are carried out by MatLab R2021b on a computer equipped with a CPU Processor: 11th Gen Intel(R) Core(TM) i7-1165G7 at 2.80 GHz running on Windows 11.

Example 3.1 (One-phase Stefan problem with forcing term)**.** This problem considers the one-phase Stefan problem with a forcing term and imposed zero temperature at the boundaries $x = 0$ and $x = s(t)$. The solidification process initiation is due to the initial temperature distribution in the solid region leading mathematically to the following:

$$s_0 = 1, \quad k = 0, \quad f(x,t) = xe^t + 2, \quad g(x) = x(1-x) \quad \text{and} \quad \text{FBC: } u(0,t) = 0.$$

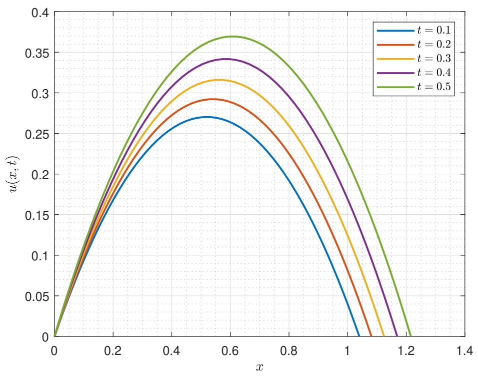
The analytical solution for the temperature distribution and the freezing front location, given by Fasano and Primicerio [10], was recently obtained by applying the heat balance integral method [19], as follows: for $0 \leq x \leq s(t)$ and $t \geq 0$,

$$u(x,t) = x(e^t - x) \quad \text{and} \quad s(t) = e^t.$$

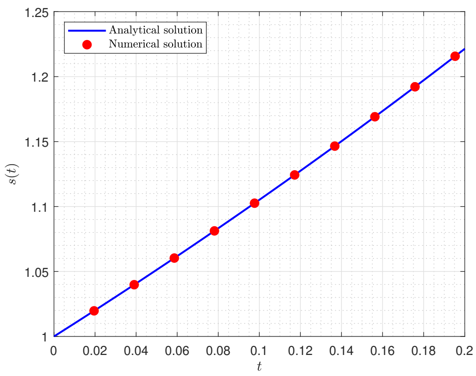
We can see that the FBC of this problem is $u(0,t) = 0$ which corresponds to the linear system (3.17). Thus, by using Algorithm 1, the numerical results of moving location $s(t)$ and temperature distribution $u(x,t)$ are obtained and measured in their accuracies via the average relative error as shown in Table 3.1. In this table, we find approximate solutions $s(t)$ and $u(x,t)$ at the terminal time $T = 0.5$ by using the discretization nodes $M \in \{10, 20, 40, 80\}$ and the time step size $\Delta t = \frac{1}{2M^2}$. We compare the obtained solutions with the existing methods such as the resulting numerical scheme (ResNS) [13], the modified numerical scheme (ModNS) [14] and the refined numerical scheme (RefNS) [24]. We can see that the average relative errors of $s(T)$ and $u(x,T)$, respectively defined by E_s and E_u, from Algorithm 1 are lower than other methods. Moreover, we also depicted the graphical behavior of temperature $u(x,t)$ at various times $t \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ in Figure 3.1(a) and plot the comparison of moving front $s(t)$ between exact and numerical solutions as shown in Figure 3.1(b) which are well-performed matching.

Table 3.1: Predicted moving location s and average relative errors E_s and E_u at the time $T = 0.5$ in Example 3.1.

M	Schemes	Location $s(0.5)$	Average relative errors E_s	Average relative errors E_u
10	ResNS	1.646602	1.29×10^{-3}	4.32×10^{-3}
	ModNS	1.644863	2.34×10^{-3}	6.82×10^{-3}
	RefNS	1.646613	1.28×10^{-3}	4.31×10^{-3}
	Algorithm 1	1.647908	8.13×10^{-4}	5.20×10^{-4}
20	ResNS	1.648193	3.20×10^{-4}	1.27×10^{-3}
	ModNS	1.647753	5.87×10^{-4}	2.07×10^{-3}
	RefNS	1.648194	3.20×10^{-4}	4.31×10^{-3}
	Algorithm 1	1.648520	2.02×10^{-4}	1.29×10^{-4}
40	ResNS	1.648589	8.00×10^{-5}	3.67×10^{-4}
	ModNS	1.648479	1.47×10^{-4}	6.10×10^{-4}
	RefNS	1.648589	8.00×10^{-5}	3.67×10^{-4}
	Algorithm 1	1.648671	5.03×10^{-5}	3.24×10^{-5}
80	ResNS	1.648688	2.00×10^{-3}	1.05×10^{-4}
	ModNS	1.648661	3.67×10^{-3}	1.77×10^{-4}
	RefNS	1.648688	2.00×10^{-3}	1.05×10^{-4}
	Algorithm 1	1.648709	1.26×10^{-5}	8.09×10^{-6}



(a) Temperature distribution $u(x, t)$



(b) Moving front location $s(t)$

Figure 3.1: Graphical solutions u and s obtained by Algorithm 1 in Example 3.1.

Example 3.2 (One-phase Stefan problem with time-dependent heat flux at the boundary)**.** The classical one-phase Stefan problem (no forcing term) with time-dependent heat flux, instead of a fixed temperature, at the boundary $x = 0$ is considered. The half space $x \geq 0$ is entirely liquid and subjected to an exponential time-decreasing heat flux at its boundary. Mathematically, that is expressed by the following equations:

$$s_0 = 0, \quad k = 0, \quad f(x,t) = 0, \quad g(x) = 0 \quad \text{and} \quad \left.\frac{\partial u}{\partial x}\right|_{x=0} = -e^t.$$

The analytical solution given by Hoffmann [11] for $0 \leq x \leq s(t)$ and $0 < t < 1$ holds:

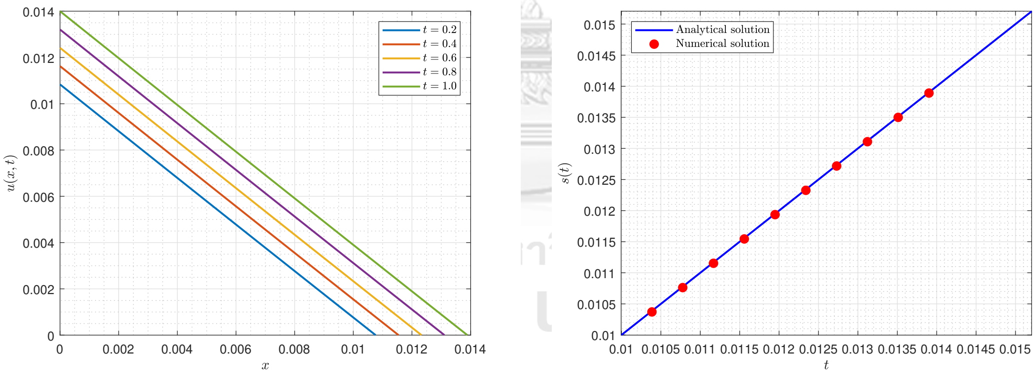
$$u(x,t) = e^{t-x} - 1 \quad \text{and} \quad s(t) = t.$$

By employing Algorithm 1 with linear system (3.18) for solving this problem, we notice that its initial moving front is $s_0 = 0$. As a result, our Algorithm 1 cannot be used because the value of s_0 is used as the denominator in the first step to compute velocity $v^{\langle 1 \rangle}$, as seen in line 6 of Algorithm 1. To resolve this issue, it is treated by using s_0 close to zero instead. We attempt to vary the values of $s_0 \in \{0.5, 0.1, 0.05, 0.01, 0.005, 0.001\}$ with the nodal number $M = 20$ and time step $\Delta t = 0.0001$. The obtained average relative errors E_s and E_u at the terminal time $T = 1$ are demonstrated in Table 3.2. We can see that the errors decrease when $s_0 \to 0$.

Thus, in this case, we choose $s_0 = 0.001$. Now, our Algorithm 1 can already handle this problem, as demonstrated by the results obtained in Table 3.3. The table showcases the predicted moving location $s(T)$ and the average relative errors E_s and E_u at the terminal time $T = 1$, with the discretization nodes $M \in \{10, 20, 40, 80\}$ and a time step $\Delta t = \frac{0.1^2}{2M^2}$. Under the same parameters M and Δt, we found that our solutions $s(T)$ and $u(x,T)$ provided significantly higher accuracy compared to other methods, namely, ResNS, ModNS and RefNS, as displayed in Table 3.3. In addition, the behavior of temperature $u(x,t)$ for different $t \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ is illustrated in Figure 3.2(a), together with the comparison of moving interface $s(t)$ is shown in Figure 3.2(b), which are very matching.

Table 3.2: Predicted moving location s and average relative errors E_s and E_u varied by s_0 in Example 3.2.

Initial moving s_0	Location $s(1)$	Average relative errors	
		E_s	E_u
0.5	0.931124	6.89×10^{-2}	8.24×10^{-2}
0.1	0.997583	2.42×10^{-3}	2.72×10^{-3}
0.05	0.999410	5.90×10^{-4}	6.58×10^{-4}
0.01	0.999981	1.94×10^{-5}	1.56×10^{-5}
0.005	0.999998	1.93×10^{-6}	4.81×10^{-6}
0.001	1.000000	2.36×10^{-7}	1.04×10^{-6}
Exact solution	1.000000	–	–



(a) Temperature distribution $u(x, t)$ **(b)** Moving front location $s(t)$

Figure 3.2: Graphical solutions u and s obtained by Algorithm 1 in Example 3.2.

Table 3.3: Predicted moving location s and average relative errors E_s and E_u at the time $T = 1$ in Example 3.2.

M	Schemes	Location $s(1)$	Average relative errors	
			E_s	E_u
10	ResNS	0.999047	9.53×10^{-4}	2.80×10^{-3}
	ModNS	0.999024	9.76×10^{-4}	2.85×10^{-3}
	RefNS	1.000023	2.30×10^{-5}	5.26×10^{-4}
	Algorithm 1	1.000000	1.65×10^{-6}	4.93×10^{-6}
20	ResNS	0.999766	2.34×10^{-4}	8.43×10^{-4}
	ModNS	0.999761	2.39×10^{-4}	8.61×10^{-4}
	RefNS	0.999997	3.05×10^{-6}	1.41×10^{-4}
	Algorithm 1	1.000000	2.36×10^{-7}	1.04×10^{-6}
40	ResNS	0.999942	5.78×10^{-5}	2.48×10^{-4}
	ModNS	0.999941	5.92×10^{-5}	2.54×10^{-4}
	RefNS	0.999998	1.80×10^{-6}	3.90×10^{-5}
	Algorithm 1	1.000000	1.17×10^{-7}	1.31×10^{-7}
80	ResNS	0.999963	1.44×10^{-5}	7.20×10^{-5}
	ModNS	0.999962	1.47×10^{-5}	7.30×10^{-5}
	RefNS	0.999999	5.70×10^{-7}	1.10×10^{-5}
	Algorithm 1	1.000000	9.92×10^{-8}	4.74×10^{-8}

Example 3.3 (one phase Stefan problem with fixed boundary and no forcing term)**.** The classical one-phase Stefan problem, instead of a fixed temperature, at the boundary $x = 0$ is considered. The half space $x \geq 0$ is entirely liquid. Mathematically, that is expressed by the following equations:

$$s_0 = 0, \quad k = 0, \quad f(x,t) = 0, \quad g(x) = 0 \quad \text{and} \quad \text{FBC: } u(0,t) = e^t - 1.$$

The analytical solution given by Hoffmann [11] for $0 \leq x \leq s(t)$ and $0 < t < 1$ holds:

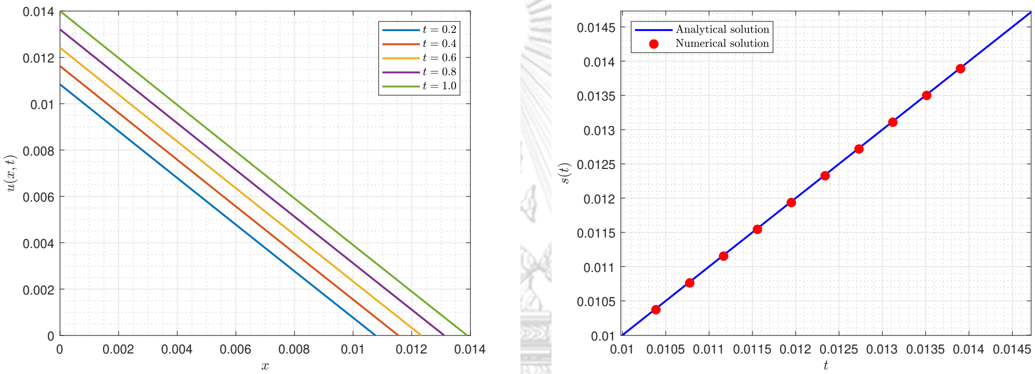
$$u(x,t) = e^{t-x} - 1 \quad \text{and} \quad s(t) = t.$$

By employing Algorithm 1 with linear system (3.17) for solving this problem, we notice that its initial moving front is $s_0 = 0$. As a result, our Algorithm 1 cannot be used because the value of s_0 is used as the denominator in the first step to compute velocity $v^{\langle 1 \rangle}$, as seen in line 6 of Algorithm 1. We use the idea of Example 3.2 to resolve this issue. We thus let s_0 close to zero instead. We attempt to vary the values of $s_0 \in \{0.5, 0.1, 0.05, 0.01, 0.005, 0.001\}$ with the nodal number $M = 20$ and time step $\Delta t = 0.0001$. The obtained average relative errors E_s and E_u at the terminal time $T = 1$ are demonstrated in Table 3.4. We can see that the errors decrease when $s_0 \to 0$.

Thus, in this case, we choose $s_0 = 0.01$. Now, our Algorithm 1 can already work with this problem as shown in the obtained results in Table 3.5. This table demonstrates the predicted moving location $s(T)$ and the average relative errors E_s and E_u at the terminal time $T = 1$ with the discretization nodes $M \in \{10, 20, 40, 80\}$ and time step $\Delta t = \frac{0.1^2}{2M^2}$. Under the same parameters M and Δt, we found that our solutions $u(x,T)$ provided much higher accuracy than other methods in [16], namely, Semi-implicit, Keller box and Crank-Nicolson as displayed in Table 3.5. In addition, the behavior of temperature $u(x,t)$ for different $t \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ is illustrated in Figure 3.3(a), together with the comparison of moving interface $s(t)$ is shown in Figure 3.3(b), which are very matching.

Table 3.4: Predicted moving location s and average relative errors E_s and E_u varied by s_0 in Example 3.3.

Initial moving s_0	Location $s(1)$	Average relative errors	
		E_s	E_u
0.5	0.982732	1.73×10^{-2}	9.94×10^{-3}
0.1	0.999878	1.22×10^{-4}	6.84×10^{-5}
0.05	0.999986	1.41×10^{-5}	6.72×10^{-6}
0.01	1.000001	9.80×10^{-7}	1.89×10^{-6}
0.005	1.000001	1.08×10^{-6}	1.95×10^{-6}
0.001	1.000001	1.10×10^{-6}	1.96×10^{-6}
Exact solution	1.000001	–	–



(a) Temperature distribution $u(x,t)$ **(b)** Moving front location $s(t)$

Figure 3.3: Graphical solutions u and s obtained by Algorithm 1 in Example 3.3.

Table 3.5: Average relative error E_u at the time $T = 1$ in Example 3.3.

M	Schemes	Average relative error E_u
10	Semi-implicit	1.05×10^{-2}
	Keller box	1.17×10^{-4}
	Crank-Nicolson	8.95×10^{-5}
	Algorithm 1	9.11×10^{-7}
20	Semi-implicit	5.16×10^{-3}
	Keller box	2.93×10^{-5}
	Crank-Nicolson	2.25×10^{-5}
	Algorithm 1	1.76×10^{-7}
40	Semi-implicit	2.56×10^{-3}
	Keller box	7.34×10^{-6}
	Crank-Nicolson	5.63×10^{-6}
	Algorithm 1	4.58×10^{-8}
80	Semi-implicit	1.28×10^{-3}
	Keller box	1.83×10^{-6}
	Crank-Nicolson	1.41×10^{-6}
	Algorithm 1	4.74×10^{-8}

In conclusion, it can be seen from the examples we have shown that the numerical solution of the FIM-CPE is more accurate compared to other methods. Next, we will consider the issue of two-sided moving boundary conditions.

CHAPTER IV

FIM-CPE FOR TWO-SIDED MOVING BOUNDARY CONDITION

In this chapter, based on the idea of FIM-CPE in one-dimensional as presented in Section 3.1, we construct the Chebyshev integration matrix to manipulate the derivative with respect to spatial variable in (2.10). Then, based on this FIM-CPE, we can devise a numerical algorithm for solving the heat equation with a two-sided moving boundary as stated in Section 2.3. Finally, our numerical algorithm is also provided.

4.1 Procedure for solving two-sided moving boundary condition

In this section, the numerical algorithm based on the FIM-CPE explained in Section 3.1 is devised for solving the heat equation with moving boundary as stated in Section 2.1. First, by hiring the idea given in [12] we change the variables $y = \frac{x-h_1(t)}{h_2(t)-h_1(t)}$, let $h_3(t) = h_2(t) - h_1(t)$. We obtain the new coordinate system (y, t) and the area with the fixed domain is $y \in (0, 1)$. Let us define the solution $u(x, t) = w(y, t)$ which corresponds to the new coordinate. Then, by employing the chain rule of partial derivatives, we get

$$\frac{\partial u}{\partial x} = \frac{1}{h_3}\frac{\partial w}{\partial y}, \quad \frac{\partial^2 u}{\partial x^2} = \frac{1}{h_3^2}\frac{\partial^2 w}{\partial y^2} \quad \text{and} \quad \frac{\partial u}{\partial t} = \frac{\partial w}{\partial t} - \frac{1}{h_3}\frac{\partial w}{\partial y}\left(yh_3' + h_1'\right).$$

Thus, using the above partial differential relations, the considered problem in Section 2.3 given by (2.10)–(2.12) with the transformation $y = \frac{x-h_1(t)}{h_3(t)}$ can be rewritten as follows.

$$\frac{\partial w}{\partial t} = \frac{a}{h_3^2}\cdot\frac{\partial^2 w}{\partial y^2} + \frac{b + yh_3' + h_1'}{h_3}\cdot\frac{\partial w}{\partial y} + cw + f, \quad (y,t) \in (0,1)\times(0,T], \tag{4.1}$$

$$w(y, 0) = g(yh_3(0) + h_1(0)), \quad y \in [0, 1],$$

$$w(0,t) = \mu_1(t), \quad w(1,t) = \mu_2(t), \quad t \in [0,T], \tag{4.2}$$

where $w = w(y,t)$, $h_1 = h_1(t)$, $h_3 = h_3(t)$, $a = a(yh_3(t)+h_1(t),t)$, $b = b(yh_3(t)+h_1(t),t)$, $c = c(t), f(yh_3(t) + h_1(t), t)$ and $T \in \mathbb{R}^+$ is denoted to be a terminal time. Next, we construct a numerical algorithm. We start from uniformly discretizing the temporal domain $[0,T]$ by specifying each time point $t_m = m\Delta t$ for $m \in \mathbb{N}$ into (4.1), where Δt is a given time step. Then, we have

$$\begin{aligned}\frac{\partial w^{\langle m\rangle}(y)}{\partial t} = {} & \frac{a^{\langle m\rangle}(yh_{3m}+h_{1m})}{h_{3m}^2}\cdot\frac{\partial^2 w^{\langle m\rangle}(y)}{\partial y^2} \\ & + \frac{b^{\langle m\rangle}(yh_{3m}+h_{1m}) + yh'_{3m} + h'_{1m}}{h_{3m}}\cdot\frac{\partial w^{\langle m\rangle}(y)}{\partial y} \\ & + c^{\langle m\rangle}w^{\langle m\rangle}(y) + f^{\langle m\rangle}(yh_{3m}+h_{1m}), \end{aligned} \tag{4.3}$$

where $h_{im} = h_i(t_m)$ for $i \in \{1,2,3\}$ and the functions with superscript $\langle m\rangle$ mean those functions are indicated at time t_m. After that, we approximate the derivative terms with respect to time t of (4.3) by applying the forward difference quotient which provides the time complexity $\mathcal{O}(\Delta t)$. Then, we have

$$\begin{aligned}\frac{w^{\langle m\rangle}(y) - w^{\langle m-1\rangle}(y)}{\Delta t} = {} & \frac{1}{h_{3m}^2}\left(a^{\langle m\rangle}(yh_{3m}+h_{1m})\cdot\frac{\partial^2 w^{\langle m\rangle}(y)}{\partial y^2}\right) \\ & + \frac{1}{h_{3m}}\left(b^{\langle m\rangle}(yh_{3m}+h_{1m})\cdot\frac{\partial w^{\langle m\rangle}(y)}{\partial y}\right) \\ & + \frac{h'_{3m}}{h_{3m}}\left(y\cdot\frac{\partial w^{\langle m\rangle}(y)}{\partial y}\right) + \frac{h'_{1m}}{h_{3m}}\cdot\frac{\partial w^{\langle m\rangle}(y)}{\partial y} \\ & + c^{\langle m\rangle}w^{\langle m\rangle}(y) + f^{\langle m\rangle}(yh_{3m}+h_{1m}). \end{aligned} \tag{4.4}$$

Now, we can see that our considered problem depends only on the spatial variable y. Hence, the FIM-CPE can be applied to the problem which assumes that a problem solution $w^{\langle m\rangle}(y)$ can be approximated by the Chebyshev polynomial expansion (3.1)

$$w^{\langle m\rangle}(y) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle} R_n(y). \tag{4.5}$$

Then, by the idea of FIM-CPE, we eliminate all derivatives with respect to the space variable y out of (4.4) by taking double-layer integrals from 0 to $y_k \in (0,1)$ on both sides of (4.4), where y_k is generated by the zeros of the Chebyshev polynomial R_M defined in (2.1). Thus, we obtain the equivalent integral equation with using the integration by parts as follows

$$
\begin{aligned}
&\int_0^{y_k}\int_0^{\xi_2}\left(\frac{w^{\langle m\rangle}(\xi_1)-w^{\langle m-1\rangle}(\xi_1)}{\Delta t}\right)d\xi_1 d\xi_2\\
&\quad=\frac{1}{h_{3m}^2}\left(a^{\langle m\rangle}(y_k h_{3m}+h_{1m})w^{\langle m\rangle}(y_k)-2\int_0^{y_k}\frac{\partial a^{\langle m\rangle}(\xi_2 h_{3m}+h_{1m})}{\partial \xi_2}w^{\langle m\rangle}(\xi_2)d\xi_2\right.\\
&\qquad\qquad\left.+\int_0^{y_k}\int_0^{\xi_2}\frac{\partial^2 a^{\langle m\rangle}(\xi_1 h_{3m}+h_{1m})}{\partial\xi_1\partial\xi_2}w^{\langle m\rangle}(\xi_1)d\xi_1 d\xi_2\right)\\
&\qquad+\frac{1}{h_{3m}}\left(\int_0^{y_k} b^{\langle m\rangle}(\xi_2 h_{3m}+h_{1m})w^{\langle m\rangle}(\xi_2)d\xi_2\right.\\
&\qquad\qquad\left.-\int_0^{y_k}\int_0^{\xi_2}\frac{\partial^2 b^{\langle m\rangle}(\xi_1 h_{3m}+h_{1m})}{\partial\xi_1\partial\xi_2}w^{\langle m\rangle}(\xi_1)d\xi_1 d\xi_2\right)\\
&\qquad+\frac{h_{3m}'}{h_{3m}}\left(\int_0^{y_k}\xi_2 w^{\langle m\rangle}(\xi_2)d\xi_2-\int_0^{y_k}\int_0^{\xi_2} w^{\langle m\rangle}(\xi_1)d\xi_1 d\xi_2\right)+\frac{h_{1m}'}{h_{3m}}\int_0^{y_k} w^{\langle m\rangle}(\xi_2)d\xi_2\\
&\qquad+c^{\langle m\rangle}\int_0^{y_k}\int_0^{\xi_2} w^{\langle m\rangle}(\xi_1)d\xi_1 d\xi_2+\int_0^{y_k}\int_0^{\xi_2} f^{\langle m\rangle}(\xi_1 h_{3m}+h_{1m})d\xi_1 d\xi_2+d_1 y_k+d_2,
\end{aligned}
\tag{4.6}
$$

where d_1 and d_2 are arbitrary constants that emerged from the process of integrations. Next, by substituting each zero $y_k \in \{y_1, y_2, y_3, \ldots, y_M\}$ into the integral equation (4.6), we can express them into the matrix form as

$$
\begin{aligned}
\frac{\mathbf{A}^2}{\Delta t}\left(\mathbf{w}^{\langle m\rangle}-\mathbf{w}^{\langle m-1\rangle}\right)&=\frac{1}{h_{3m}^2}\left(\mathrm{diag}\{\mathbf{a}^{\langle m\rangle}\}\mathbf{w}^{\langle m\rangle}-2\mathbf{A}\mathrm{diag}\{\mathbf{a}_y^{\langle m\rangle}\}\mathbf{w}^{\langle m\rangle}+\mathbf{A}^2\mathrm{diag}\{\mathbf{a}_{yy}^{\langle m\rangle}\}\mathbf{w}^{\langle m\rangle}\right)\\
&\quad+\frac{1}{h_{3m}}\left(\mathbf{A}\mathrm{diag}\{\mathbf{b}^{\langle m\rangle}\}\mathbf{w}^{\langle m\rangle}-\mathbf{A}^2\mathrm{diag}\{\mathbf{b}_y^{\langle m\rangle}\}\mathbf{w}^{\langle m\rangle}\right)\\
&\quad+\frac{h_{3m}'}{h_{3m}}\left(\mathbf{A}\mathrm{diag}\{\mathbf{y}\}\mathbf{w}^{\langle m\rangle}-\mathbf{A}^2\mathbf{w}^{\langle m\rangle}\right)+\frac{h_{1m}'}{h_{3m}}\mathbf{A}\mathbf{w}^{\langle m\rangle}\\
&\quad+c^{\langle m\rangle}\mathbf{A}^2\mathbf{w}^{\langle m\rangle}+\mathbf{A}^2\mathbf{f}^{\langle m\rangle}+d_1\mathbf{y}+d_2\mathbf{i}
\end{aligned}
$$

which can be simplified to

$$
\mathbf{K}^{\langle m\rangle}\mathbf{w}^{\langle m\rangle}-d_1\mathbf{y}-d_2\mathbf{i}=\frac{\mathbf{A}^2}{\Delta t}\mathbf{w}^{\langle m-1\rangle}+\mathbf{A}^2\mathbf{f}^{\langle m\rangle},
\tag{4.7}
$$

where $\mathbf{K}^{\langle m\rangle} := \frac{\mathbf{A}^2}{\Delta t}-\frac{\mathbf{D}_1^{\langle m\rangle}-2\mathbf{A}\mathbf{D}_2^{\langle m\rangle}+\mathbf{A}^2\mathbf{D}_3^{\langle m\rangle}}{h_{3m}^2}-\frac{\mathbf{A}\mathbf{D}_4^{\langle m\rangle}-\mathbf{A}^2\mathbf{D}_5^{\langle m\rangle}}{h_{3m}}-\frac{h_{3m}'(\mathbf{A}\mathbf{Y}-\mathbf{A}^2)}{h_{3m}}-\frac{h_{1m}'\mathbf{A}}{h_{3m}}-c^{\langle m\rangle}\mathbf{A}^2$, $\mathbf{D}_1=\mathrm{diag}\{\mathbf{a}^{\langle m\rangle}\}$, $\mathbf{D}_2=\mathrm{diag}\{\mathbf{a}_y^{\langle m\rangle}\}$, $\mathbf{D}_3=\mathrm{diag}\{\mathbf{a}_{yy}^{\langle m\rangle}\}$, $\mathbf{D}_4=\mathrm{diag}\{\mathbf{b}^{\langle m\rangle}\}$, $\mathbf{D}_5=\mathrm{diag}\{\mathbf{b}^{\langle m\rangle}\}$

and $\mathbf{Y} = \mathrm{diag}\{\mathbf{y}\}$. The other parameters contained in (4.7) are as follows: $c^{\langle m\rangle} = c(t_m)$, $\mathbf{A} = \overline{\mathbf{R}}\mathbf{R}^{-1}$ is the Chebyshev integration matrix defined in Section 3.1,

$$
\begin{aligned}
\mathbf{i}^\top &= [1, 1, 1, \ldots, 1]\,, \\
\mathbf{y}^\top &= [y_1, y_2, y_3, \ldots, y_M]\,, \\
\mathbf{Y} &= \mathrm{diag}\{y_1, y_2, y_3, \ldots, y_M\}, \\
\mathbf{w}^{\langle m\rangle\top} &= [w(y_1h_{3m} + h_{1m}, t_m), w(y_2h_{3m} + h_{1m}, t_m), \ldots, w(y_Mh_{3m} + h_{1m}, t_m)]\,, \\
\mathbf{f}^{\langle m\rangle\top} &= [f(y_1h_{3m} + h_{1m}, t_m), f(y_2h_{3m} + h_{1m}, t_m), \ldots, f(y_Mh_{3m} + h_{1m}, t_m)]\,.
\end{aligned}
$$

For the $M \times M$ diagonal matrices $\mathbf{D}_1^{\langle m\rangle}, \mathbf{D}_2^{\langle m\rangle}, \mathbf{D}_3^{\langle m\rangle}, \mathbf{D}_4^{\langle m\rangle}$ and $\mathbf{D}_5^{\langle m\rangle}$ are defined by

$$
\mathbf{D}_1^{\langle m\rangle} = \begin{bmatrix} a(y_1h_{3m} + h_{1m}, t_m) & 0 & \cdots & 0 \\ 0 & a(y_2h_{3m} + h_{1m}, t_m) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & a(y_Mh_{3m} + h_{1m}, t_m) \end{bmatrix},
$$

$$
\mathbf{D}_2^{\langle m\rangle} = \begin{bmatrix} \frac{\partial a}{\partial y}(y_1h_{3m} + h_{1m}, t_m) & 0 & \cdots & 0 \\ 0 & \frac{\partial a}{\partial y}(y_2h_{3m} + h_{1m}, t_m) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial a}{\partial y}(y_Mh_{3m} + h_{1m}, t_m) \end{bmatrix},
$$

$$
\mathbf{D}_3^{\langle m\rangle} = \begin{bmatrix} \frac{\partial^2 a}{\partial y^2}(y_1h_{3m} + h_{1m}, t_m) & 0 & \cdots & 0 \\ 0 & \frac{\partial^2 a}{\partial y^2}(y_2h_{3m} + h_{1m}, t_m) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial^2 a}{\partial y^2}(y_Mh_{3m} + h_{1m}, t_m) \end{bmatrix},
$$

$$
\mathbf{D}_4^{\langle m\rangle} = \begin{bmatrix} b(y_1h_{3m} + h_{1m}, t_m) & 0 & \cdots & 0 \\ 0 & b(y_2h_{3m} + h_{1m}, t_m) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & b(y_Mh_{3m} + h_{1m}, t_m) \end{bmatrix},
$$

$$\mathbf{D}_5^{\langle m\rangle} = \begin{bmatrix} \frac{\partial b}{\partial y}(y_1h_{3m}+h_{1m},t_m) & 0 & \cdots & 0 \\ 0 & \frac{\partial b}{\partial y}(y_2h_{3m}+h_{1m},t_m) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial b}{\partial y}(y_Mh_{3m}+h_{1m},t_m) \end{bmatrix}.$$

From the given non-homogeneous Dirichlet boundary conditions (4.2), we can convert them into vector forms by using the linear combination of Chebyshev polynomial expansion (4.5) at the time t_m as follows:

$$w^{\langle m\rangle}(0) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle} R_n(0) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle}(-1)^n = \boldsymbol{\ell}_0^\top \mathbf{c}^{\langle m\rangle} = \boldsymbol{\ell}_0^\top \mathbf{R}^{-1}\mathbf{w}^{\langle m\rangle} = \mu_1(t_m), \tag{4.8}$$

$$w^{\langle m\rangle}(1) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle} R_n(1) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle} = \mathbf{i}^\top \mathbf{c}^{\langle m\rangle} = \mathbf{i}^\top \mathbf{R}^{-1}\mathbf{w}^{\langle m\rangle} = \mu_2(t_m), \tag{4.9}$$

where $\boldsymbol{\ell}_0^\top = \left[1,-1,1,-1,\ldots,(-1)^{M-1}\right]$.

Finally, from (4.7), (4.8) and (4.9), we can combine them to construct the system of linear equations at the iterative time t_m for $m \in \mathbb{N}$, which contains $M+2$ unknown variables including $\mathbf{w}^{\langle m\rangle}, d_1$ and d_2, as follows:

$$\left[\begin{array}{c|cc} \mathbf{K}^{\langle m\rangle} & -\mathbf{y} & -\mathbf{i} \\ \hline \boldsymbol{\ell}_0^\top \mathbf{R}^{-1} & 0 & 0 \\ \mathbf{i}^\top \mathbf{R}^{-1} & 0 & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{w}^{\langle m\rangle} \\ \hline d_1 \\ d_2 \end{array}\right] = \left[\begin{array}{c} \frac{\mathbf{A}^2\mathbf{w}^{\langle m-1\rangle}}{\Delta t} + \mathbf{A}^2\mathbf{f}^{\langle m\rangle} \\ \hline \mu_1(t_m) \\ \mu_2(t_m) \end{array}\right], \tag{4.10}$$

where $\mathbf{K}^{\langle m\rangle} := \frac{\mathbf{A}^2}{\Delta t} - \frac{\mathbf{D}_1^{\langle m\rangle} - 2\mathbf{A}\mathbf{D}_2^{\langle m\rangle} + \mathbf{A}^2\mathbf{D}_3^{\langle m\rangle}}{h_{3m}^2} - \frac{\mathbf{A}\mathbf{D}_4^{\langle m\rangle} - \mathbf{A}^2\mathbf{D}_5^{\langle m\rangle}}{h_{3m}} - \frac{h'_{3m}(\mathbf{A}\mathbf{Y} - \mathbf{A}^2)}{h_{3m}} - \frac{h'_{1m}\mathbf{A}}{h_{3m}} - c^{\langle m\rangle}\mathbf{A}^2$. Accordingly, the solution $\mathbf{w}^{\langle m\rangle}$ can be approximated by solving the system (4.10) that starts from the given initial conditions $h_1(0) = h_{10}, h_2(0) = h_{20}, h_3(0) = h_{30} = h_{20} - h_{10}$ and $\mathbf{w}^{\langle 0\rangle^\top} = \left[g(y_1h_{30}+h_{10}), g(y_2h_{03}+h_{10}), \ldots, g(y_Mh_{30}+h_{10})\right]$. Note that, performing of the final iteration, the obtained numerical solutions $h_{1m} = h_1(T), h_{3m} = h_3(T)$ and $\mathbf{w}^{\langle m\rangle}$ can be actually expressed corresponding to the function $u(x,T)$ that is

$$\mathbf{w}^{\langle m\rangle^\top} = \left[u(y_1h_{3m}+h_{1m},T), u(y_2h_{3m}+h_{1m},T), \ldots, u(y_Mh_{3m}+h_{1m},T),\right].$$

For computational convenience, we summarize all the above-mentioned procedures in terms of the pseudocode algorithm in order to find an approximate solution of the heat equation with moving boundary in Section 2.3 by using the FIM-CPE.

Algorithm 2 Numerical algorithm for solving the heat equation with moving boundary

---

Input: $a(x,t)$, $b(x,t)$, $c(t)$, T, M, Δt, $g(x)$, $f(x,t)$, $h_1(t)$, $h_2(t)$, $\mu_1(t)$ and $\mu_2(t)$;
Output: The approximate solutions $\mathbf{w}^{\langle m\rangle}$, $h_1^{\langle m\rangle}$, $h_2^{\langle m\rangle}$ and $h_3^{\langle m\rangle}$;

Set $y_k \leftarrow \frac{1}{2}\left(1+\cos\left(\frac{2k-1}{2M}\pi\right)\right)$ for $k \in \{1,2,3,\ldots,M\}$ in ascending order;
Compute $\mathbf{y}$, $\mathbf{i}$, $\mathbf{R}$, $\overline{\mathbf{R}}$, $\mathbf{R}^{-1}$, $\mathbf{A}$ and $\boldsymbol{\ell}_0$;
Construct $\mathbf{w}^{\langle 0\rangle} \leftarrow [g(y_1h_{30}+h_{10}), g(y_2h_{03}+h_{10}), \ldots, g(y_Mh_{30}+h_{10})]^\top$;
Set $m \leftarrow 1$ and $t_1 \leftarrow \Delta t$;
while $t_m \leq T$ **do**
    Compute $x^{\langle m\rangle} \leftarrow yh_3^{\langle m\rangle} + h_1^{\langle m\rangle}$;
    Compute $\mathbf{D}_1^{\langle m\rangle}, \mathbf{D}_2^{\langle m\rangle}, \mathbf{D}_3^{\langle m\rangle}, \mathbf{D}_4^{\langle m\rangle}$, $\mathbf{D}_5^{\langle m\rangle}$ and $\mathbf{K}^{\langle m\rangle}$;
    Compute $\mathbf{f}^{\langle m\rangle} \leftarrow \left[f(x_1^{\langle m\rangle}, t_m), f(x_2^{\langle m\rangle}, t_m), \ldots, f(x_M^{\langle m\rangle}, t_m)\right]^\top$;
    Find $\mathbf{w}^{\langle m\rangle}$ by solving the iterative linear system (4.10);
    Update $m \leftarrow m+1$;
    Compute $t_m \leftarrow m\Delta t$;
end while
return The final iteration of $\mathbf{w}^{\langle m\rangle}$, $h_1^{\langle m\rangle}$, $h_2^{\langle m\rangle}$ and $h_3^{\langle m\rangle}$;

4.2 Numerical examples for two-sided moving boundary condition

In this section, we apply the proposed Algorithm 2 based on the FIM-CPE for finding numerical results of the heat equation with a two-sided moving boundary in order to demonstrate its efficiency and accuracy by measuring with average relative error via two examples. Examples 4.1 and 4.2 are the two-sided moving boundary conditions with forcing term and non-homogeneous Dirichlet boundary conditions. All the experiments are carried out by MatLab R2021b on a computer equipped with a CPU Processor: 11th Gen Intel(R) Core(TM) i7-1165G7 at 2.80 GHz running on Windows 11.

Example 4.1. In this example, we consider the case when the coefficients in (2.10) are a set of polynomials of the first order in x and t. Moreover, the free boundaries are linear functions in time, as illustrated in the following quantities.

$$
\begin{aligned}
&a(x,t) = 1 + xt, \quad b(x,t) = 1 + x, \quad c(t) = 1 + t, \\
&h_1(t) = 1 + t, \quad h_2(t) = 2 + 2t, \quad h_3(t) = h_2(t) - h_1(t) = 1 + t, \\
&\mu_1(t) = (1+t)^2 + 2t + 1, \quad \mu_2(t) = 4(1+t)^2 + 2t + 1, \\
&f(x,t) = 2 - 2(1+xt) - 2x(1+x) - (1+t)(x^2 + 2t + 1) \text{ and } u(x,t) = x^2 + 2t + 1.
\end{aligned}
$$

After performing the transformation, we have

$$
\begin{aligned}
a(y,t) &= 1 + (y+1)(1+t)t, \\
b(y,t) &= 1 + (y+1)(1+t), \\
w(y,t) &= (y+1)^2(1+t)^2 + 1 + 2t \quad \text{and} \\
f(y,t) &= 2 - 2(1 + t(y+1)(1+t)) - 2(y+1)(t+1)(1 + (y+1)(1+t)) \\
&\quad - (1+t)((y+1)^2(1+t)^2 + 2t + 1).
\end{aligned}
$$

By using our Algorithm 2, the numerical results of moving location $h_1(t), h_2(t)$ and temperature distribution $u(x,t)$ are obtained and measured in their accuracies via the average relative error as shown in Table 4.1. In this table, we compare the exact solution with our obtained approximate solution $u(x,t)$ at the terminal time $T = 1$ by using the discretization nodes $M \in \{10, 20, 40, 80\}$ and time step size $\Delta t = \frac{T}{M}$. We find the average relative errors of $u(x,T)$, defined by E_u, from Algorithm 2 provided are low. Moreover, we also depicted the graphical behavior of temperature $u(x,t)$ at various times $t \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ in Figure 4.1 which is a good performance.

Table 4.1: Average relative error E_u at time $T = 1$ in Example 4.1.

M	Exact	Algorithm 2	Average relative error E_u
10	18.9017	18.9049	3.30×10^{-3}
20	18.9753	18.9761	2.50×10^{-3}
40	18.9938	18.9940	2.10×10^{-3}
80	18.9985	18.9985	2.00×10^{-3}
Location	$h_1(1) = 2$	$h_2(1) = 4$	–

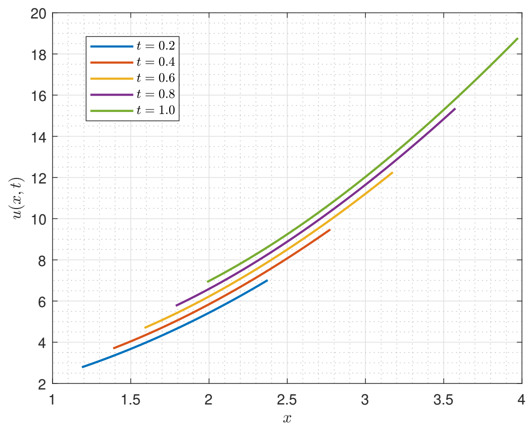


Figure 4.1: Temperature distribution $u(x, t)$ obtained by Algorithm 2 in Example 4.1.

Example 4.2. In this example, we consider a nonlinear case for the coefficients and a linear one for the free boundaries h_1 and h_2.

$$
\begin{aligned}
&a(x,t) = (1+x+t)^2, \quad b(x,t) = x^2 + \sin(t), \quad c(t) = t + t^2,\\
&h_1(t) = 1 + t^3, \quad h_2(t) = 2 + t^2, \quad h_3(t) = h_2(t) - h_1(t) = 1 + t^2 - t^3,\\
&\mu_1(t) = 1 + 2t^2 + (1+t^3)^3, \quad \mu_2(t) = 1 + 2t^2 + (2+t^2)^3,\\
&f(x,t) = 4t - 6t(1+t+x)^2 - (t+t^2)(1+2t^2+x^3) - 3x^2(x^2+\sin(t)) \text{ and}\\
&u(x,t) = x^3 + 2t^2 + 1.
\end{aligned}
$$

After performing the transformation, we have

$$
\begin{aligned}
a(y,t) &= (2 + t + t^3 + (1 + t^2 + x^3)y)^2,\\
b(y,t) &= (1 + t^3 + (1 + t^2 - t^3)y)^2 + \sin(t),\\
w(y,t) &= 1 + 2t^2 + (1 + t^3 + (1 + t^2 - t^3)y)^3 \quad \text{and}\\
f(y,t) &= 4t - 6(1 + t^3 + (1 + t^2 - t^3)y)(2 + t + t^3 + (1 + t^2 - t^3)y)^2\\
&\quad - (t + t^2)(1 + 2t^2 + (1 + t^3 + (1 + t^2 - t^3)y)^3)\\
&\quad - 3(1 + t^3 + (1 + t^2 - t^3)y)^2((1 + t^3 + (1 + t^2 - t^3)y)^2 + \sin(t)).
\end{aligned}
$$

By using our Algorithm 2, the numerical results of moving location $h_1(t), h_2(t)$ and temperature distribution $u(x,t)$ are obtained and measured in their accuracies via the average relative error as shown in Table 4.2. In this table, we compare the exact solution with our obtained approximate solution $u(x,t)$ at the terminal time $T = 1$ by using the discretization nodes $M \in \{10, 20, 40, 80\}$ and time step size $\Delta t = \frac{T}{M}$. We find the average relative errors of $u(x,T)$, defined by E_u, from Algorithm 2 provided are low. Moreover, we also depicted the graphical behavior of temperature $u(x,t)$ at various times $t \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ in Figure 4.2 which is a good performance.

Table 4.2: Average relative error E_u at time $T = 1$ in Example 4.2.

M	Exact	Algorithm 2	Average relative error E_u
10	29.8341	29.8298	5.00×10^{-3}
20	29.9584	29.9572	5.60×10^{-3}
40	29.9896	29.9893	6.00×10^{-3}
80	29.9974	29.9973	6.10×10^{-3}
Location	$h_1(1) = 2$	$h_2(1) = 3$	–

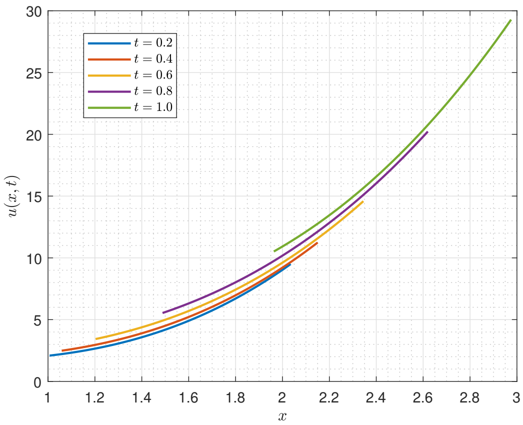


Figure 4.2: Temperature distribution $u(x, t)$ obtained by Algorithm 2 in Example 4.2.

CHAPTER V

CONCLUSIONS

5.1 Conclusions and discussions

In this thesis, the main idea is to construct a numerical algorithm for finding approximate solutions of the heat equation with moving boundary described in Sections 2.2 and 2.3, we first transform the problem from moving boundary into the fixed boundary by using the spatial coordinate transformation. Afterward, we manipulate the derivative with respect to the time variable by using the forward different quotient. Then, the FIM-CPE is applied to handle the derivative with respect to the space variable.

In Chapter III, we constructed Algorithm 1, expressed in pseudocode form for easy implementation. Furthermore, this Algorithm 1 remains flexible for the given FBC at $x = 0$, which can be transformed into a vector form based on the established concepts of (3.15) and (3.16) as well. In addition, the performance of Algorithm 1 is demonstrated through numerical experiments in three examples: Example 3.1, Example 3.2, and Example 3.3. These examples involve one-phase Stefan problems with a forcing term, time-dependent heat flux at the boundary, and a fixed boundary with no forcing term, respectively. The results of these examples show that our algorithm accurately predicts the evolution of the temperature distribution $u(x, t)$ and the moving front location $s(t)$. Furthermore, our algorithm achieves lower average relative errors compared to other schemes such as ResNS, ModNS, and RefNS in Examples 3.1 and 3.2, as demonstrated in Tables 3.1 and 3.3. Additionally, in Example 3.3, our algorithm outperforms other schemes, namely Semi-implicit, Keller box, and Crank-Nicolson, with lower average relative errors, as shown in Tables 3.5. However, Algorithm 1 has a limitation: it cannot operate when the initial position of the moving position interface, s_0, is set to zero. This is due to the fact that s_0 is used as the denominator in the first step to compute the velocity $v^{\langle 1 \rangle}$. Therefore, the treatment for this issue is by choosing the value of s_0 close to zero as obviously illustrated in Examples 3.2 and 3.3 that we use $s_0 = 0.001$ and $s_0 = 0.01$, respectively. Consequently, the obtained numerical solutions still exhibit a high level of precision.

In Chapter IV, we use FIM-CPE to devise the numerical Algorithm 2 for solving two-sided moving boundary conditions (2.10) as demonstrated in Section 2.3. The numerical examples demonstrate that our Algorithm 2 gives a good performance via Examples 4.1 and 4.2. These examples show that our algorithm can accurately predict the evolution of the temperature distribution $u(x,t)$ with a comparison exact solution and the moving front location $h_1(t), h_2(t)$ and $h_3(t)$ and also provide the average relative errors, in Examples 4.1 and 4.2 which can be seen in Tables 4.1 and 4.2, respectively. We further depict the graphical behaviors of temperature distribution $u(x,t)$ at different times t together in Figures 4.1 and 4.2.

5.2 Future works

In future work, we really hope that our proposed FIM-CPE can be applied to interesting problems. The lists of our future plan include the following:

- Improve our FIM-CPE to be more accurate in one-dimensional one-phase Stefan problem and two-sided moving boundary conditions.

- Extend our FIM-CPE to the multi-dimensional moving boundary problems.

REFERENCES

[1] R. Boonklurb and A. Duangpan. Finite integration method using chebyshev expansion for solving nonlinear poisson equations on irregular domains. *J. Numer. Anal. Ind. Appl. Math.*, 14 (1–2):7–24, 2020.

[2] 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):1–19, 2020.

[3] R. Boonklurb, A. Duangpan, and M. Juytai. Numerical solutions for systems of fractional and classical integro-differential equations via finite integration method based on shifted chebyshev polynomials. *Fractal Fract.*, 5(3):1–21, 2021.

[4] R. Boonklurb, A. Duangpan, and T. Treeyaprasert. Modified finite integration method using chebyshev polynomial for solving linear differential equations. *J. Numer. Ind. Appl. Math.*, 12(3–4):1–19, 2018.

[5] R. Boonklurb, A. Duangpan, and T. Treeyaprasert. Finite integration method with shifted chebyshev polynomials for solving time-fractional burgers' equations. *Mathematics*, 7(12):1–24, 2019.

[6] J. Caldwell and Y. Kwan. On the perturbation method for the stefan problem with time-dependent boundary conditions. *Int. J. Heat Mass Transf.*, 46:1497–1501, 2003.

[7] F. J. Caro-Lopera, V. Leiva, and N. Balakrishnan. Connection between the hadamard and matrix products with an application to matrix-variate birnbaum–saunders distributions. *J. Multivar. Anal.*, 104(1):126–139, 2012.

[8] J. Crank. Two methods for the solution of moving boundary problems in diffusion and heat flow. *Q. J. Mech. Appl. Math.*, 10:202–231, 1957.

[9] A. Duangpan. *Finite integration method with Chebyshev expansion for finding nuerical solution of nonlinear and fractional order differential equations.* Chilalongkorn University, Bangkok, Thailand, 2019.

[10] A. Fasano and M. Primicerio. Free boundary problems for nonlinear parabolic equations with nonlinear free boundary conditions. *J. Math. Anal. Appl.*, 72(1):247–273, 1979.

[11] K. H. Hoffmann. *Freie Randwert problem I, II, III.* Freie University Berlin, Berlin, 1977.

[12] M. S. Hussein and Z. Adil. Numerical solution for two-sided stefan problem. *Iraqi J. Sci.*, 61(2):444–452, 2020.

[13] S. Kutluay. Numerical schemes for one-dimensional stefan-like problems with a forcing term. *Appl. Math. Comput.*, 168(2):1159–1168, 2005.

[14] S. Kutluay, A. R. Bahadir, and A. Özdeş. The numerical solution of one-phase classical stefan problem. *J. Comput. Appl. Math.*, 81(1):135–144, 1997.

[15] O. Ladyzenskaja, V. Solonnikov, and N. Ural'ceva. *Linear and Quasilinear Equation of Parabolic Type.* American Mathematical Society, 1968.

[16] S. L. Mitchell and M. Vynnycky. Finite-difference methods with increased accuracy and correct initialization for one-dimensional stefan problems. *Appl. Math. Comput.*, 215:1609–1621, 2009.

[17] W. D. Murray and F. Landis. Numerical and machine solutions of transient heat-conduction problems involving melting or freezing: Part i – method of analysis and sample solutions. *J. Heat Transf.*, 81(2):106–112, 1959.

[18] P. Rattanadecho and S. Wongwises. Moving boundary-moving mesh analysis of freezing process in water-saturated porous media using a combined transfinite interpolation and pde mapping methods. *J. Heat Transf.*, 130(1):1–10, 2008.

[19] N. Sadoun, E.-K. Si-Ahmed, P. Colinet, and J. Legrand. On the goodman heat-balance integral method for stefan like-problems: further considerations and refinements. *Thermal Sci.*, 13(2):81–96, 2009.

[20] N. Sadoun, E.-K. Si-Ahmed, P. Colinet, and J. Legrand. On the boundary immobilization and variable space grid methods for transient heat conduction problems with phase change: Discussion and refinement. *Comptes Rendus Mécanique.*, 340(7):501–511, 2012.

[21] H. Snitko. Determination of the minor coefficient in a parabolic equation in a free boundary domain. *Ukrainian mathematical*, 1:60–67, 2009.

[22] B. Sudhakar. An integral method for non-linear moving boundary problems. *Chem. Eng. Sci.*, 47:475–479, 1992.

[23] B. Sudhakar. On integral iterative formulation in classical stefan problem. *Chem. Eng. Sci.*, 47:3158–3162, 1992.

[24] A. S. Wood. A new look at the heat balance integral method. *Appl. Math. Model.*, 25(10):815–824, 2001.

[25] Z. Yang, M. Sen, and S. Paolucci. Solidification in finite slab with convective cooling and shrinkage. *Appl. Math. Model.*, 27:733–762, 2003.

APPENDICES

APPENDIX A : Examples of MatLab code for one-phase Stefan problems

In calculating the approximate solutions of each example in this research, we implement the MatLab code for FIM-CPE to find the result. In this appendix, we would like to show examples of code, and the command for solving a system of linear equations.

Example A1 (Stefan problem with forcing term). Consider the problem in Example 3.1.

$$s_0 = 1, \quad k = 0, \quad f(x,t) = xe^t + 2, \quad g(x) = x(1-x) \quad \text{and} \quad \text{FBC: } u(0,t) = 0.$$

The analytical solution is $u(x,t) = x(e^t - x)$ and the freezing front location is $s(t) = e^t$. Thus, we can construct the linear system in case (3.17) as follows:

$$\left[\begin{array}{c|cc} \mathbf{K}^{\langle m\rangle} & -\boldsymbol{\eta} & -\mathbf{i} \\ \hline \mathbf{i}^{\top}\mathbf{R}^{-1} & 0 & 0 \\ \boldsymbol{\ell}_0^{\top}\mathbf{R}^{-1} & 0 & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{w}^{\langle m\rangle} \\ \hline d_1 \\ d_2 \end{array}\right] = \left[\begin{array}{c} \frac{\mathbf{A}^2\mathbf{w}^{\langle m-1\rangle}}{\Delta t} + \mathbf{A}^2\mathbf{f}^{\langle m\rangle} \\ \hline k \\ \phi_0(t_m) \end{array}\right].$$

```
% -- Set initial parameters ------------------------------------
H = 500:500:2500;
J = 250:250:2500;
M  = 80;                        % number of node
T  = 0.5;                       % terminal time
t0 = 0;                         % initial time
dt = 1/(2*M^2);                 % time step
s0 = exp(t0);                   % initial position moving
k  = 0;                         % right boundary
phi0 = @(t) 0;                  % left boundary
s  = @(t) exp(t);               % position moving front
g  = @(x) x.*(exp(t0)-x);       % initial condition
f  = @(x,t) x*exp(t)+2;         % forcing term
ue = @(x,t) x.*(exp(t)-x);      % exact solution
```

```
% -- Construct Chebyshev integration matrix A ----------------
eta = flip(1/2*(cos((2*(1:M)'-1)/(2*M)*pi)+1)); % zero of CBS
R(:,1) = ones(M,1);
R(:,2) = 2*eta-1;
for n = 2:M
    R(:,n+1) = 2*(2*eta-1).*R(:,n)-R(:,n-1);
end
Rbar(:,1) = eta;
Rbar(:,2) = eta.^2-eta;
for n = 2:M-1
    Rbar(:,n+1) = 1/4*(R(:,n+2)/(n+1)-R(:,n)/(n-1)-2*(-1)^n/(n^2-1));
end
Rinv = 1/M*diag([1 2*ones(1,M-1)])*R(:,1:M)';
A = Rbar*Rinv;
% -- Construct Block Matrix -----------------------------------
W = []; S = []; S1 = [];
wm = g(eta*s0);
sm = s0;
l0 = (-1).^(0:M-1)';
q = (0:M-1)'.^2;
i = ones(M,1);
t = t0+dt:dt:T;
for m = 1:length(t)
    vm = -2*q'*Rinv*wm/sm;
    sm = sm + vm*dt;
    Km = A^2/dt-vm/sm*(A*diag(eta)-A^2)-eye(M)/(sm^2);
    fm = f(eta*sm,t(m));
    Q = [Km -eta -i; i'*Rinv 0 0; l0'*Rinv 0 0];
    F = [A^2*wm/dt+A^2*fm; k; phi0(t(m))];
    w = Rinv(Q)*F;
    wm = w(1:M);
```

```
    if find(m==H)
        t(m)
        W = [W wm]
        S = [S sm]
    end
    if find(m==J)
        t(m)
        S1 = [S1 sm]
    end
end
% Approximate solution W and moving front location s ---------
x = eta*sm;
[x ue(x,T) wm abs((ue(x,T)-wm))]
average_error_u = mean(abs((ue(x,T)-wm)))
location_s = round(sm,6)
average_error_s = abs(sm-s(T))
% -- Plot solution ---------------------------------------------
figure
for i = 1:5
    plot(eta*S(i),W(:,i),'LineWidth',1.5)
    hold on;
end
grid on; grid minor;
figure
plot(t,s(t),'b-','LineWidth',1.5)
hold on;
plot(t(J),S1','ro','LineWidth',1.5,'MarkerFaceColor','r')
grid on; grid minor;
hold off;
```

Example A2 (Stefan problem with time-dependent heat flux at the boundary). We consider the problem in Example 3.2.

$$s_0 = 0, \quad k = 0, \quad f(x,t) = 0, \quad g(x) = 0 \quad \text{and} \quad \frac{\partial u}{\partial x}\Big|_{x=0} = -e^t.$$

The analytical solution is $u(x,t) = e^{t-x} - 1$ and the freezing front location is $s(t) = t$. Thus, we can construct the linear system in case (3.18) as follows:

$$\left[\begin{array}{c|cc} \mathbf{K}^{\langle m\rangle} & -\boldsymbol{\eta} & -\mathbf{i} \\ \hline \mathbf{i}^\top\mathbf{R}^{-1} & 0 & 0 \\ \boldsymbol{\ell}_1^\top\mathbf{R}^{-1} & 0 & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{w}^{\langle m\rangle} \\ \hline d_1 \\ d_2 \end{array}\right] = \left[\begin{array}{c} \frac{\mathbf{A}^2\mathbf{w}^{\langle m-1\rangle}}{\Delta t} + \mathbf{A}^2\mathbf{f}^{\langle m\rangle} \\ \hline k \\ s^{\langle m\rangle}\phi_1(t_m) \end{array}\right].$$

```
% -- Set initial parameters ----------------------------------
H = 996:1000:4996
J = 496:500:4996
M  = 80;                       % number of node
T  = 1;                        % teminal time
dt = 0.1^2/(2*M^2);            % time step
s0 = 0.001;                    % initial position moving front
k  = 0;                        % right boundary
phi1 = @(t) -exp(t);           % left boundary
s  = @(t) t;                   % position moving front
g  = @(x) 0*x;                 % initial condition
f  = @(x,t) 0*x;               % forcing term
ue = @(x,t) exp(t-x)-1;        % exact solution
% -- Construct Chebyshev integration matrix A ----------------
eta = flip(1/2*(cos((2*(1:M)'-1)/(2*M)*pi)+1)); % zero of CBS
R(:,1) = ones(M,1);
R(:,2) = 2*eta-1;
```

```
for n = 2:M
    R(:,n+1) = 2*(2*eta-1).*R(:,n)-R(:,n-1);
end
Rbar(:,1) = eta;
Rbar(:,2) = eta.^2-eta;
for n = 2:M-1
    Rbar(:,n+1) = 1/4*(R(:,n+2)/(n+1)-R(:,n)/(n-1)-2*(-1)^n/(n^2-1));
end
Rinv = 1/M*diag([1 2*ones(1,M-1)])*R(:,1:M)';
A = Rbar*Rinv;
% -- Construct Block Matrix ----------------------------------
W = []; S = []; S1 = [];
wm = g(eta*s0);
sm = s0;
l1 = 2*(-1).^(1:M)'.*(0:M-1)'.^2;
q = (0:M-1)'.^2;
i = ones(M,1);
t = s0:dt:T;
for m = 1:length(t)
    vm = -2*q'*Rinv*wm/sm;
    sm = sm + vm*dt;
    Km = A^2/dt-vm/sm*(A*diag(eta)-A^2)-eye(M)/(sm^2);
    fm = f(eta*sm,t(m));
    Q = [Km -eta -i; i'*Rinv 0 0; l1'*Pinv 0 0];
    F = [A^2*wm/dt+A^2*fm; k; sm*phi1(t(m))];
    w = Rinv(Q)*F;
    wm = w(1:M);
    [vm t(m) sm]
```

```
    if find(m==H)
        t(m)
        W = [W wm]
        S = [S sm]
    end
    if find(m==J)
        t(m)
        S1 = [S1 sm]
    end
end
% Approximate solution W and moving front location s -----------
x = eta*sm;
[x ue(x,T) wm abs((ue(x,T)-wm))]
average_error_u = mean(abs((ue(x,T)-wm)))
location_s = round(sm,6)
average_error_s = abs(sm-s(T))
% -- Plot solution -------------------------------------------
figure
for i = 1:5
    plot(eta*S(i),W(:,i),'LineWidth',1.5)
    hold on
end
grid on; grid minor;
figure
plot(t,s(t),'b-','LineWidth',1.5)
hold on;
plot(t(J),S1','ro','LineWidth',1.5,'MarkerFaceColor','r')
grid on; grid minor;
hold off;
```

APPENDIX B : Examples of MatLab code for two-sided moving boundary conditions

Example B1. We consider the problem in Example 4.1.

$$
\begin{aligned}
&a(x,t) = 1 + xt, \quad b(x,t) = 1 + x, \quad c(t) = 1 + t, \\
&h_1(t) = 1 + t, \quad h_2(t) = 2 + 2t, \quad h_3(t) = h_2(t) - h_1(t) = 1 + t, \\
&\mu_1(t) = (1+t)^2 + 2t + 1, \quad \mu_2(t) = 4(1+t)^2 + 2t + 1, \\
&f(x,t) = 2 - 2(1+xt) - 2x(1+x) - (1+t)(x^2 + 2t + 1).
\end{aligned}
$$

The analytical solution is $u(x,t) = x^2 + 2t + 1$. Thus, we can construct the linear system in (4.10) as follows:

$$
\left[\begin{array}{c|cc}
\mathbf{K}^{\langle m\rangle} & -\boldsymbol{y} & -\mathbf{i} \\
\hline
\boldsymbol{\ell}_0^\top \mathbf{R}^{-1} & 0 & 0 \\
\mathbf{i}^\top \mathbf{R}^{-1} & 0 & 0
\end{array}\right]
\left[\begin{array}{c}
\mathbf{w}^{\langle m\rangle} \\
\hline
d_1 \\
d_2
\end{array}\right]
=
\left[\begin{array}{c}
\frac{\mathbf{A}^2\mathbf{w}^{\langle m-1\rangle}}{\Delta t} + \mathbf{A}^2\mathbf{f}^{\langle m\rangle} \\
\hline
\mu_1(t_m) \\
\mu_2(t_m)
\end{array}\right].
$$

```
% -- Set initial --------------------------
H = 0:16:80;
M  = 80;                          % number of nodes
T  = 1;                           % teminal time
t0 = 0;                           % initial time
dt = T/M;                         % time step
a = @(x,t) 1+x*t;                 % coefficient of u_xx
b = @(x,t) 1+x;                   % coefficient of u_x
c = @(t) 1+t;                     % coefficient of u
h1 = @(t) 1+t;                    % position left moving front
h2 = @(t) 2+2*t;                  % position right moving front
h3 = @(t) 1+t;                    % h2-h1
mu1 = @(t) (1+t)^2+2*t+1;         % left boundary
mu2 = @(t) 4*(1+t)^2+2*t+1;       % right boundary
```

```
g = @(x) x.^2+1;                  % initial condition
f = @(x,t) 2-2*(1+x*t)-2*x.*(1+x)-(1+t)*(x.^2+2*t+1); % forcing
    term
ue = @(x,t) x.^2+2*t+1;           % exact solution
% -- Construct matrix A -----------------------------------
y = flip(1/2*(cos((2*(1:M)'-1)/(2*M)*pi)+1)); % zero of CBS
R(:,1) = ones(M,1);
R(:,2) = 2*y-1;
for n = 2:M
    R(:,n+1) = 2*(2*y-1).*R(:,n)-R(:,n-1);
end
Rbar(:,1) = y;
Rbar(:,2) = y.^2-y;
for n = 2:M-1
    Rbar(:,n+1) = 1/4*(R(:,n+2)/(n+1)-R(:,n)/(n-1)-2*(-1)^n/(n
        ^2-1));
end
Rinv = 1/M*diag([1 2*ones(1,M-1)])*R(:,1:M)';
A = Rbar*Rinv;
% -- Derivative of coefficient functions -----------------------
h1t = @(t) 1;
h3t = @(t) 1;
ay  = @(x,t) (1+t)*t+0*x;
ayy = @(x,t) 0*x;
by  = @(x,t) 1+t+0*x;
% Block Matrix --------------------------------------------------
W = []; H1 = []; H2 = []; H3 = [];
wm = g(y.*h3(t0)+h1(t0));
l0 = (-1).^(0:M-1)';
```

```
i = ones(M,1);
t = t0:dt:T;
for m = 1:length(t)
    x = y.*h3(t(m))+h1(t(m));
    Km = A^2/dt-(diag(a(x,t(m)))-2*A*diag(ay(x,t(m))) ...
        +A^2*diag(ayy(x,t(m))))./(h3(t(m)))^2 ...
        -(A*diag(b(x,t(m)))-A*diag(by(x,t(m))) ...
        +h3t(t(m))*(A*diag(y)-A^2)+h1t(t(m))*A)./(h3(t(m))) ...
        -A^2*c(t(m));
    fm = f(x,t(m));
    Q = [Km -y -i; 10'*Rinv 0 0; i'*Rinv 0 0];
    F = [A^2*wm/dt+A^2*fm; mu1(t(m)); mu2(t(m))];
    w = pinv(Q)*F;
    wm = w(1:M);
    h1m = h1(t(m));
    h2m = h2(t(m));
    h3m = h3(t(m));
    if find(m==H)
        t(m);
        W = [W wm];
        H1 = [H1 h1m];
        H2 = [H2 h2m];
        H3 = [H3 h3m];
    end
end
% -- Approximate solution W and moving front location h_1,h_2
x = y.*h3(T)+h1(T);
[x ue(x,T) wm abs((ue(x,T)-wm)./ue(x,T))]
average_error_u = mean(abs((ue(x,T)-wm)./ue(x,T)))
```

```
location_h1 = h1(T)
location_h2 = h2(T)
% -- Plot solution --------------------
figure
for i = 1:5
    plot(y*H3(i)+H1(i),W(:,i),'LineWidth',1.5)
    hold on
end
grid on
grid minor
xlabel('$x$','Interpreter','latex')
ylabel('$u(x,t)$','Interpreter','latex')
legend({'$t=0.2$','$t=0.4$','$t=0.6$','$t=0.8$','$t=1.0$'},'Interpreter','latex','Location', 'northeast')
```

BIOGRAPHY

Name Miss Warunya Wong-u-ra

Date of Birth July 13, 1997

Place of Birth Bangkok, Thailand

Educations B.Sc. (Mathematics), Thammasat University, 2019

Publications

- W. Wong-u-ra and R. Boonklurb, Numerical Algorithm Based on Finite Integration Method using Shifted Chebyshev Expansion for Solving Moving Boundary Problems, *Proceeding of the* 26^{th} *Annual Meeting in Mathematics conference* (2022), 169–180.