ระเบียบวิธีปริพันธ์อันตะร่วมกับการกระจายเชบีเชฟสำหรับการหาผลเฉลยเชิงตัวเลขของสมการเชิงอนุพันธ์ไม่เชิงเส้นและอันดับเศษส่วน

นายอำพล ดวงแป้น

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

FINITE INTEGRATION METHOD WITH CHEBYSHEV EXPANSION FOR FINDING NUMERICAL SOLUTION OF NONLINEAR AND FRACTIONAL ORDER DIFFERENTIAL EQUATIONS

Mr. Ampol Duangpan

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Applied Mathematics and
Computational Science
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2019
Copyright of Chulalongkorn University

Dissertation Title	FINITE INTEGRATION METHOD WITH CHEBYSHEV EXPANSION FOR FINDING NUMERICAL SOLUTION OF NONLINEAR AND FRACTIONAL ORDER DIFFERENTIAL EQUATIONS
By	Mr. Ampol Duangpan
Field of Study	Applied Mathematics and Computational Science
Dissertation Advisor	Associate Professor Ratinan Boonklurb, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

. . . . Dean of the Faculty of Science
(Professor Polkit Sangvanich, Ph.D.)

DISSERTATION COMMITTEE

. . . . Chairman
(Associate Professor Khamron Mekchay, Ph.D.)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

. . . . Dissertation Advisor
(Associate Professor Ratinan Boonklurb, Ph.D.)

. . . . Examiner
(Associate Professor Anusorn Chonwerayuth, Ph.D.)

. . . . Examiner
(Associate Professor Petarpa Boonserm, Ph.D.)

. . . . External Examiner
(Assistant Professor Tawikan Treeyaprasert, Ph.D.)

อำพล ดวงแป้น : ระเบียบวิธีปริพันธ์อันตะร่วมกับการกระจายเชบีเชฟสำหรับการหาผลเฉลยเชิงตัวเลขของสมการเชิงอนุพันธ์ไม่เชิงเส้นและอันดับเศษส่วน. (FINITE INTEGRATION METHOD WITH CHEBYSHEV EXPANSION FOR FINDING NUMERICAL SOLUTION OF NONLINEAR AND FRACTIONAL ORDER DIFFERENTIAL EQUATIONS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร. รตินันท์ บุญเคลือบ, 110 หน้า.

ในวิทยานิพนธ์ฉบับนี้ เราพัฒนาระเบียบวิธีปริพันธ์อันตะโดยใช้การกระจายพหุนามเชบีเชฟ สำหรับการแก้สมการเชิงอนุพันธ์ไม่เชิงเส้นในหนึ่งและสองมิติ ระเบียบวิธีปริพันธ์อันตะที่พัฒนาขึ้นนี้สามารถใช้งานได้บนโดเมนใด ๆ จากนั้นเราใช้ระเบียบวิธีปริพันธ์อันตะที่พัฒนาขึ้นจัดการกับตัวแปรของปริภูมิ และใช้อัตราส่วนเชิงผลต่างสืบเนื่องไปข้างหน้าจัดการกับอนุพันธ์ในตัวแปรของเวลา เพื่อสร้างขั้นตอนวิธีเชิงตัวเลขไปใช้แก้ปัญหาไม่เชิงเส้นสามปัญหา ประกอบด้วย สมการเบอร์เกอร์ที่มีคลื่นกระแทกในหนึ่งมิติ สมการเบนจามิน-โบนา-มาโฮนี-เบอร์เกอร์เชิงเศษส่วนของเวลา และสมการปัวส์ซงไม่เชิงเส้นในสองมิติบนโดเมนไม่ปรกติ นอกจากนี้ ยังได้ทดสอบขั้นตอนวิธีของเราด้วยการทดลองแก้ปัญหาหลายตัวอย่าง แล้วเปรียบเทียบผลลัพธ์โดยประมาณที่ได้จากขั้นตอนวิธีที่เราเสนอและวิธีอื่น ๆ กับผลเฉลยเชิงวิเคราะห์ ตัวอย่างเหล่านั้นแสดงให้เห็นว่า ขั้นตอนวิธีที่นำเสนอ ปรับปรุงค่าประมาณของผลเฉลยให้แม่นยำขึ้นอย่างมีนัยสำคัญ และใช้ต้นทุนในการคำนวณน้อย

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาควิชา	คณิตศาสตร์และ วิทยาการคอมพิวเตอร์	ลายมือชื่อนิสิต	....
		ลายมือชื่อ อ.ที่ปรึกษาหลัก	....
สาขาวิชา	คณิตศาสตร์ประยุกต์ และวิทยาการคณนา		
ปีการศึกษา	2562		

6072835223 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORDS : FINITE INTEGRATION METHOD / CHEBYSHEV EXPANSION / TIME-FRACTIONAL ORDER DERIVATIVE / POISSON EQUATION / BURGERS' EQUATION

AMPOL DUANGPAN : FINITE INTEGRATION METHOD WITH CHEBYSHEV EXPANSION FOR FINDING NUMERICAL SOLUTION OF NONLINEAR AND FRACTIONAL ORDER DIFFERENTIAL EQUATIONS. ADVISOR : ASSOC. PROF. RATINAN BOONKLURB, Ph.D., 110 pp.

In this dissertation, we develop the finite integration method by using Chebyshev polynomial expansion (FIM-CPE) for solving one- and two-dimensional nonlinear differential equations. The developed FIM-CPE can be used on any domains. Then, we utilize our FIM-CPE to deal with the spatial variables and the forward difference quotient to handle the derivative involving temporal variable. Thus, the numerical algorithms based on this idea are devised to overcome three nonlinear problems including one-dimensional Burgers' equation with shock wave, time-fractional Benjamin-Bona-Mahony-Burgers' equation and two-dimensional nonlinear Poisson equation over irregular domains. Moreover, we examine our algorithms with several experimental examples by comparing the approximate results obtained by our methods and other methods with their analytical solutions. Those examples show that the proposed algorithms provide a significant improvement of the approximate solution in terms of accuracy with low computational cost.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Department : Mathematics and	Student's Signature ....
Computer Science	Advisor's Signature ....
Field of Study : Applied Mathematics and	
Computational Science	
Academic Year : 2019	

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my dissertation advisor, Associate Professor Ratinan Boonklurb, Ph.D. for the continuous supporting my studies since the master degree until the doctoral degree. His encouragement, motivation and guidance helped me during the time for writing researches and this dissertation until it was accomplished. I further would like to thank all of my dissertation committees: Associate Professor Khamron Mekchay, Ph.D., Associate Professor Anusorn Chonwerayuth, Ph.D., Associate Professor Petarpa Boonserm, Ph.D. and Assistant Professor Tawikan Treeyaprasert, Ph.D., for their insightful comments and suggestions which motivated me to extend my research from various perspectives.

Moreover, I would like to express my special appreciation and thanks to my financial sponsors, “The 100th Anniversary Chulalongkorn University Fund for Doctoral Scholarship” for the scholarship. My sincere thanks also go to the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University which provided me funding to present my research on international conference and an opportunity that I received throughout my graduate studies.

Finally, I would like to thank my family for supporting me throughout writing this dissertation. Also, I wish to express my gratitude to all friends and colleagues who stayed with me and provided their encouragement, relaxation, great suggestions and supports in many ways during a hard time studying in my doctoral degree.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

CONTENTS

Page

ABSTRACT IN THAI iv

ABSTRACT IN ENGLISH v

ACKNOWLEDGEMENTS vi

CONTENTS vii

LIST OF TABLES ix

LIST OF FIGURES xi

CHAPTER

1 INTRODUCTION **1**

1.1 Motivation and Literature Surveys 1

1.2 Research Objectives 7

1.3 Dissertation Overview 7

2 FIM WITH CHEBYSHEV EXPANSION **8**

2.1 Chebyshev Polynomials 8

2.2 Chebyshev Expansion 12

2.3 Kronecker Product 17

2.4 Developed FIM-CPE 18

2.4.1 One-Dimensional Chebyshev Integration Matrices 18

2.4.2 Two-Dimensional Chebyshev Integration Matrices 21

3 BURGERS’ EQUATION WITH SHOCK WAVE **28**

3.1 Burgers’ Equation 28

3.2 Algorithm for Solving Burgers’ Equation 29

3.3 Numerical Examples for Testing Algorithm 1 33

3.4 Acceleration of Algorithm 1 48

4 TIME-FRACTIONAL BBMB EQUATION **53**

4.1 Benjamin-Bona-Mahony-Burgers Equation 53

4.2 Time-Fractional Order Derivative 54

4.3 Algorithm for Solving Time-Fractional BBMB Equation 55

4.4 Numerical Examples for Testing Algorithm 2 60

CHAPTER Page

5 NONLINEAR POISSON EQUATION **68**

5.1 Poisson-Type Equation 68

5.2 Algorithm for Solving Nonlinear Poisson Equation 70

5.3 Numerical Examples for Testing Algorithm 3 78

6 CONCLUSIONS AND DISCUSSIONS **92**

6.1 Conclusions 92

6.2 Future works 95

REFERENCES 96

APPENDICES 102

BIOGRAPHY 110

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

LIST OF TABLES

Table Page

3.1 MAE at different M and Δt of Example 3.1 34

3.2 Comparison of absolute errors of Example 3.1 35

3.3 MAE at different M and Δt of Example 3.2 38

3.4 Comparison of absolute errors of Example 3.2 38

3.5 MRE at different M and Δt of Example 3.3 41

3.6 Comparison of error norms at different viscosity ν of Example 3.3 41

3.7 Comparison of error norms at different nodes M of Example 3.3 41

3.8 MAE at different M and Δt of Example 3.4 43

3.9 Comparison of error norms at different times T of Example 3.4 44

3.10 Convergence orders p using forward difference of Examples 3.1 - 3.4 49

3.11 Convergence orders p using Crank-Nicolson of Examples 3.1 - 3.4 52

4.1 MAE at various M and Δt for $\alpha = 0.5$ of Example 4.1 61

4.2 MAE at various Δt for $\alpha = 0.5$ and $M = 40$ and of Example 4.1 61

4.3 MAE at various α for $\Delta t = 0.001$ and $M = 40$ of Example 4.1 61

4.4 MAE at various M for $\alpha = 0.5$ and $\Delta t = 0.01$ of Example 4.2 62

4.5 MAE at various Δt for $\alpha = 0.5$ and $M = 40$ of Example 4.2 62

4.6 MAE at various α for $\Delta t = 0.001$ and $M = 40$ of Example 4.2 63

4.7 MAE at various M for $\alpha = 0.5$ and $\Delta t = 0.01$ of Example 4.3 64

4.8 MAE at various Δt for $\alpha = 0.5$ and $M = 40$ of Example 4.3 64

4.9 MAE at various α for $\Delta t = 0.001$ and $M = 40$ of Example 4.3 64

4.10 Convergence orders p at each fractional order α of Example 4.1 66

4.11 Convergence orders p at each fractional order α of Example 4.2 66

4.12 Convergence orders p at each fractional order α of Example 4.3 66

5.1 MAE for each discretizing H at different tolerances of Example 5.1 79

5.2 MAE for each discretizing H at different tolerances of Example 5.2 81

5.3 MAE for each discretizing H at different tolerances of Example 5.3 83

5.4 MAE for each discretizing H at different tolerances of Example 5.4 84

5.5 RMSE for each discretizing H at $TOL = 10^{-5}$ of Example 5.5 86

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

5.6 AME and RMSE for each discretizing H at $TOL = 10^{-2}$ of Example 5.5 . . . 87

5.7 Absolute error at different points (x, y) of Example 5.6 88

5.8 MAE for each discretizing H at different tolerances of Example 5.7 90

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

LIST OF FIGURES

Figure Page

2.1 Chebyshev polynomials $T_n(x)$ for $n \in \{0, 1, 2, 3, 4, 5\}$ 8

2.2 Plotting of the first few polynomials on $[-1, 1]$ 13

2.3 Plotting of the polynomials $w_1(x)$ and $w_2(x)$ on $[-1, 1]$ 15

2.4 The indices of the grid points globally and locally 21

3.1 Physical behavior of our solution in Example 3.1 35

3.2 Behavior of solutions for small values ν in Example 3.1 36

3.3 Profiles of our solution at small viscosity ν in Example 3.1 37

3.4 Physical behavior of our solution in Example 3.2 39

3.5 Behavior of solutions for small values ν in Example 3.2 39

3.6 Profiles of our solution at small viscosity ν in Example 3.2 40

3.7 Physical behavior of our solution in Example 3.3 42

3.8 Profiles of our solution at small viscosity ν in Example 3.3 42

3.9 Physical behavior of our solution in Example 3.4 44

3.10 Profiles of our solution at small viscosity ν in Example 3.4 44

3.11 Profiles of our numerical solution for $\nu = 1$ in Example 3.5 45

3.12 Profiles of our numerical solution for $\nu = 0.1$ in Example 3.5 46

3.13 Profiles of our numerical solution for $\nu = 0.01$ in Example 3.5 46

3.14 Profiles of our numerical solution for $\nu = 0.1$ in Example 3.6 47

3.15 Profiles of our numerical solution for $\nu = 0.01$ in Example 3.6 47

3.16 Graphs of linearly convergence orders p of Examples 3.1 - 3.4 49

3.17 Graphs of quadratically convergence orders p of Examples 3.1 - 3.4 52

4.1 Graphical behavior of our solution in Example 4.1 61

4.2 Graphical behavior of our solution in Example 4.2 63

4.3 Graphical behavior of our solution in Example 4.3 64

4.4 Graphical convergence order p of Example 4.1 67

4.5 Graphical convergence order p of Example 4.2 67

4.6 Graphical convergence order p of Example 4.3 67

5.1 Discretizing each side of boundary over $[a, b] \times [c, d]$ 74

5.2 Order and rate of convergences of Example 5.1 80

5.3 Pentagonal domain and numerical solution of Example 5.1 80

5.4 Order and rate of convergences of Example 5.2 81

5.5 Circular domain and numerical solution of Example 5.2 82

5.6 Order and rate of convergences of Example 5.3 83

5.7 L-shaped domain and numerical solution of Example 5.3 83

5.8 Order and rate of convergences of Example 5.4 85

5.9 Butterfly domain and numerical solution of Example 5.4 85

5.10 Order and rate of convergences of Example 5.5 87

5.11 Peanut-shaped domain and numerical solution of Example 5.5 87

5.12 Order and rate of convergences of Example 5.6 89

5.13 Elliptic domain and numerical solution of Example 5.6 89

5.14 Order and rate of convergences of Example 5.7 91

5.15 Elliptic domain and numerical solution of Example 5.7 91

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

CHAPTER I

INTRODUCTION

1.1 Motivation and Literature Surveys

A differential equation is a mathematical equation depending on the values of functions and their derivatives. Generally, it can be divided into several types such as ordinary and partial; or linear and nonlinear differential equations. In mathematics, the ordinary and partial differential equations (ODEs and PDEs) are the differential equations corresponding to the functions of single and multiple variables, respectively. The nonlinear differential equation is also the differential equation that is not linear in the unknown function and its derivatives. Moreover, these differential equations are used to describe various principles and behaviors of natural phenomena. Especially, most of the real incidents that appear in our daily life are inborn nonlinear. The applications of the nonlinear differential equations play a prominent role in many disciplines including mathematics, physics, economics and engineering.

A fractional differential equation (FDE) is also another type of differential equations used in many fields of science and engineering. In 1695, Leibniz and L'Hôpital [50] have been first introduced the basic concept of FDE which its order of derivative can be taken as an integer or rational number in $(0, 1]$. The applications of FDE have occurred in various real-world problems, such as oscillating dynamical systems [2], thermal conductivity [32], quantum models [35], diffusion processes [44], rheological models [66], etc.

There are many interesting issues in the forms of nonlinear PDEs or FDEs. In this dissertation, we mainly focus on two nonlinear PDEs including one-dimensional Burgers' equation with a shock wave and two-dimensional nonlinear Poisson equation over irregular domains and one nonlinear FDE, namely, time-fractional Benjamin-Bona-Mahony-Burgers' (BBMB) equation. These nonlinear PDEs and FDE have numerous applications

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

in the real world. In order to understand these problems as well as further apply them in practical life, the finding of their solutions is important. However, it is usually very difficult to solve these problems analytically. Thus, numerical methods play an essential role in seeking approximate solutions of these nonlinear problems.

The first interesting issue of nonlinear PDEs in this dissertation is the Burgers' equation that was first introduced by Bateman [7] in 1915. He mentioned that this kind of equation was worthy of study and he gave its steady solutions. In 1948, Burgers [15] studied a mathematical model for turbulence using the equation considered in [7]. This model is then known as "Burgers' equation". The Burgers' equation has some common features with the Navier-Stokes equation. Nowadays, the Burgers' equation, which is a fundamental nonlinear PDE, has been hired in a large variety of applications in applied mathematics, physics and engineering such as a simplified fluid dynamics model, modeling of transport with accumulation, advection and diffusion terms, gas dynamics, traffic flow, modeling of shock waves, heat conduction, acoustic waves, statistics of flow problems, mixing and turbulent diffusion and so on, see [15], [42] and [58] for details. Some numerical methods have been handled with the Burgers' equation such as the Adomian decomposition method [36], homotopy perturbation method [29], variational iteration method [69] and so on.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

The second interesting issue for this dissertation is the time-fractional BBMB which was modified from the nonlinear PDE called the BBMB equation. Originally, the BBMB equation was introduced by Benjamin et al. [9] in 1972. It is the mathematical model of propagation for small-amplitude long waves in nonlinear dispersive media systems which improved from the Korteweg-de Vries (KdV) equation. Normally, the BBMB and KdV equations are relevant to the wave breaking models. The KdV model came from water waves and is used for long waves in many other physical systems. However, in some physical systems of long waves, the KdV equation was not applicable. Hence, the BBMB was proposed instead. It described the unidirectional transmission of long-wave signals in a certain nonlinear dispersive system [30]. For nonlinear FDE, the time-fractional BBMB equation was presented to discuss the dynamic behavior of physical systems. It has been solved by the homotopy analysis transform method [33], the Crank-Nicolson difference scheme [57] and the modified residual power series method [73], etc.

The last interesting issue of nonlinear PDEs in the dissertation is a Poisson equation which can efficiently describe many problems, such as numerical simulation of potential field, gravitational field, thermal field and electrostatic field, see [67]. In reality, for solving the nonlinear Poisson equations, it is very rare to find their solutions in closed forms. Therefore, several effective numerical methods had been developed for solving nonlinear Poisson-type problems such as finite different method (FDM) by Nagel [48], finite element method by Hu et al. [25], boundary element method by Kasab et al. [28], least square method by Arzani and Afshar [3] and so on. In addition, the shape of the domain is one of the main difficulties for constructing a numerical scheme. Thus, we consider the Poisson equation in a more general form over several irregular domains.

As mentioned above, we can see that these three nonlinear issues have been solved by several numerical methods. However, most of these methods have a process of calculating numerical differentiations. It is well-known that the numerical differentiation is very sensitive to round-off errors since its manipulation task involves a division by small step-size. On the other hand, the numerical integration involves a process of multiplication by small step-size, so it is very insensitive to round-off errors. In addition, the integration preserves the approximation accuracy and it is a smoothing process compared with differentiation. Therefore, if we can consider the given problem of PDE in the form of an integral operator instead of a differential operator, the obtained approximate solution should provide much better accuracy.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Recently, the finite integration method (FIM) has been first proposed in 2013 by Wen et al. [63] in order to overcome the boundary value problems for linear PDEs. The concept of this FIM is to transform the given PDE into an equivalent integral equation and apply numerical integrations to solve the integral equation afterward. By constructing the linear integral operator which is called “integration matrix” from the numerical quadrature. Originally, Wen et al. [63] have constructed the integration matrices by using the trapezoidal rule and radial basis functions for solving one-dimensional linear PDEs. They provided more accurate approximate results of these linear PDEs when compared with the FDM. After that Li et al. [38] applied this FIM to solve the nonlocal elastic straight bar under static and dynamic loading conditions. In 2015, Li et al. [37] continued to extend the FIM for finding approximate solutions of two-dimensional linear PDEs. Next, the extended FIM in two dimensions has been used to handle the sideways problem of reconstructing an inaccessible boundary value for parabolic PDEs with variable coefficients proposed by Yu et al. [68]. Then, Yun et al. [70] have developed the two-dimensional FIM combined with the technique of least square method to deal with higher-dimensional singular perturbation problems with multiple boundary layers. In 2016, the FIM has been improved by constructing the novel integration matrices based on three numerical quadratures consisting of the Simpson's rule, Newton-Cotes and Lagrange formula instead of using the trapezoidal rule. These improved FIMs were presented by Li et al. [39] in order to find an approximate solution of the linear PDEs which they provided a higher accuracy than the original FIM for the same number of nodes. Furthermore, we can see that the above-mentioned FIM has been successfully applied to solve various kinds of linear PDEs and it was verified by comparing with several existing methods that it offers a very stable, highly accurate and efficient approach.

In 2018, Boonklurb et al. [14] have been modified the original FIM via Chebyshev polynomial expansion in order to overcome the one- and two-dimensional linear PDEs which provided a much higher accurate solution than the FDM and those traditional FIMs. After that Saengsiritongchai's thesis used this modified FIM to construct the numerical algorithms for solving time-dependent linear PDEs and linear space-fractional order PDEs which obtain the accurate approximate solutions and a part of his work was just recently published in [13]. In 2019, Gugaew's thesis applied the modified FIM to find numerical solutions of direct and inverse problems for linear integro-differential equations (IDEs) that a part of her work was just also recently published in [12]. In 2020, Juytai extended the Gugaew's work by creating the numerical procedures to solve the systems of linear IDEs both Volterra and Fredholm types and also the system of linear ODEs, especially, a stiff type. These procedures are based on the idea of modified FIM, but

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

some of them are slightly adjusted by using the shifted Chebyshev polynomials instead. We can see that the above-mentioned FIMs have been successfully demonstrated the accurate solutions for many kinds of differential equations, especially, linear differential equations. However, we notice here that many applications of this efficiently modified FIM in [14] have not yet been performed to solve the nonlinear differential equations. Therefore, it becomes our major studies for the research in this dissertation.

In this dissertation, we develop the modified FIM [14] by using the Chebyshev polynomial expansion over the general domain in order for the method to be applicable on any given domains which called the “developed FIM-CPE”. It is well known that the Chebyshev polynomials have an orthogonal property which plays an important role in the theory of approximation. The roots of Chebyshev polynomial can be found explicitly and when the equidistant nodes give poor accuracy, the problem can be improved by using the Chebyshev nodes or zeros of Chebyshev polynomial instead. If the function is sampled at the Chebyshev nodes, it has the best approximation under the maximum norm, see [55] for more details. With these advantages, our developed FIM-CPE is constructed by approximating the solutions in terms of the Chebyshev expansion. We use the zeros of the Chebyshev polynomial in the general domain of a certain degree to interpolate the approximate solution. Subsequently, we apply our developed FIM-CPE to devise three numerical algorithms for finding the approximate solutions of our above-interesting issues to the following nonlinear PDEs and FDE.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

- The one-dimensional Burgers’ equation with a shock wave

  $$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} = \nu\frac{\partial^2 v}{\partial x^2}, \quad (x,t) \in (a,b) \times (0,T] \tag{1.1}$$

  subject to the initial condition $v(x,0) = \phi(x)$ for $x \in [a,b]$ and the Dirichlet boundary conditions $v(a,t) = \psi_1(t)$ and $v(b,t) = \psi_2(t)$ for $t \in (0,T]$, where $a < b \in \mathbb{R}$, $T \in \mathbb{R}^+$, ϕ, ψ_1 and ψ_2 are the given sufficiently continuous functions and $\nu > 0$ is a constant coefficient of kinematic viscosity.

- The one-dimensional time-fractional derivative BBMB equation

$$D_t^{\alpha} v - \frac{\partial^3 v}{\partial x^2 \partial t} + \frac{\partial v}{\partial x} + v\frac{\partial v}{\partial x} = f(x,t), \quad (x,t) \in (0,L) \times (0,T] \tag{1.2}$$

  subject to the initial condition $v(x,0) = \phi(x)$ for $x \in [0,L]$ and the Dirichlet boundary conditions $v(0,t) = \psi_1(t)$ and $v(L,t) = \psi_2(t)$ for $t \in (0,T]$, where D_t^{α} is time-fractional differential operator with order $\alpha \in (0,1)$ in the sense of Caputo [51], $L, T \in \mathbb{R}^+$, f, ϕ, ψ_1 and ψ_2 are given sufficiently continuous functions.

- The two-dimensional Poisson equation over irregular domain

$$\nabla^2 v + \alpha(x,y)\frac{\partial v}{\partial x} + \beta(x,y)\frac{\partial v}{\partial y} + \gamma(x,y)v = f(x,y,v), \quad (x,y) \in \Omega \tag{1.3}$$

  subject to the Dirichlet boundary condition $v(x,y) = \psi(x,y)$ for $(x,y) \in \partial\Omega$, where α, β, γ and ψ are the given smooth functions depending on the variables x and y, respectively and f is a nonlinear in terms of v over the irregular domain $\Omega \subseteq \mathbb{R}^2$.

Moreover, we see also that the nonlinear problems (1.1) and (1.2) depend on both space and time variables. Accordingly, their time derivative terms include the first order and fractional order, respectively. These temporal variables are estimated by the forward difference quotient. The spatial variable of these problems is handled by our developed FIM-CPE. Also, we can extend the developed FIM-CPE to deal with two-dimensional spatial domain in (1.3). Finally, we examine our proposed three numerical algorithms through several experimental examples by comparing approximate results obtained by our method and other methods with their analytical solutions. These numerical examples are implemented via MatLab 2016a software and runs on an Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz computer system. These examples demonstrate the ability of our proposed algorithms to produce a significant improvement in terms of accuracy with the low computational cost.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.2 Research Objectives

The objectives of this research have the following goals. The first goal is to develop the FIM by using Chebyshev polynomial expansion in order to be applicable directly on arbitrary domains both in one- and two-dimensional spaces. The next goal is then to devise three numerical algorithms for solving nonlinear PDEs and FDE including one-dimensional Burgers' equation with a shock wave, time-fractional BBMB equation and two-dimensional nonlinear Poisson equation over irregular domains. Finally, the last goal is to obtain a much higher degree of accuracy for our proposed algorithms than other methods with low computational cost under the same parameters and conditions.

1.3 Dissertation Overview

This dissertation is separated into six chapters and it is organized as follows. First, Chapter I is proposed about an introduction of this work including the motivation and literature surveys, the research objectives and the thesis overview. Chapter II presents the developed FIM-CPE by first introducing the background knowledge concerning the definitions and some important properties of both Chebyshev polynomials and Kronecker product. We further discuss about Chebyshev expansion that mentions the good choices of the basis functions and the interpolated nodes in order to obtain the best polynomial approximation in some senses like maximum norm. After that these facts are applied to construct the Chebyshev integration matrices both in one- and two-dimensional spaces. In Chapter III, we have employed the developed FIM-CPE to devise numerical algorithm for solving the one-dimensional Burgers' equation with shock wave. We also utilize our proposed FIM-CPE to create numerical algorithms for finding approximate solutions of one-dimensional time-fractional BBMB equation and two-dimensional nonlinear Poisson equation over the irregular domains as demonstrated in Chapters IV and V, respectively. Besides, comparisons of accuracy and efficiency for each algorithm are displayed in each of its chapters via examining several experimental examples. Finally, the discussions about our obtained results and some conclusions are provided in Chapter VI. The possibly future researches are also suggested.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

CHAPTER II

FIM WITH CHEBYSHEV EXPANSION

In this chapter, the background knowledges about definition and some essential properties of Chebyshev polynomials on the general domain are introduced to be the main materials for developing the FIM. The developed FIM-CPE can be performed over arbitrary domain without any further transformation. Moreover, we also discuss about the Chebyshev expansion which plays an important role in our developed FIM-CPE for constructing the one- and two-dimensional Chebyshev integration matrices.

2.1 Chebyshev Polynomials

First, let us provide the basic definition of Chebyshev polynomials and their useful properties. Chebyshev polynomials are the set of orthogonal polynomials which plays a significant role in the theory of approximation, see [43] for more details. In addition, there are several kinds of Chebyshev polynomials. However, in this work, we only focus on the Chebyshev polynomial of the first kind of degree $n \geq 0$, which is denoted by $T_n(x)$, as defined in Definition 2.1. Then, we can express an instance of the first six Chebyshev polynomials $T_n(x)$ at degree $n \in \{0, 1, 2, 3, 4, 5\}$ for $x \in [-1, 1]$ depicted in Figure 2.1.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

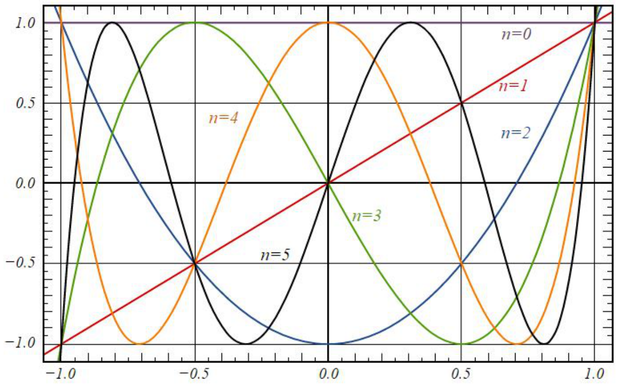


Figure 2.1: Chebyshev polynomials $T_n(x)$ for $n \in \{0, 1, 2, 3, 4, 5\}$

Definition 2.1. ([43]) The Chebyshev polynomial of degree $n \geq 0$ is defined by

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

However, the Chebyshev polynomial $T_n(x)$ can be defined on the general domain $[a, b]$ by

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

Henceforth, the Chebyshev polynomial in this dissertation refers to $R_n(x)$ in (2.2). For the construction of Chebyshev integration matrices in both one and two dimensions, some elementary properties of Chebyshev polynomial are needed to develop. Thus, their useful properties of Chebyshev polynomial are provided as follows.

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

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

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

$$\bar{R}_0(x) = \int_a^x R_0(\xi)d\xi = x - a, \tag{2.4}$$

$$\bar{R}_1(x) = \int_a^x R_1(\xi)d\xi = \frac{(x - a)(x - b)}{b - a}, \tag{2.5}$$

$$\bar{R}_n(x) = \int_a^x R_n(\xi)d\xi = \frac{b - a}{4}\left(\frac{R_{n+1}(x)}{n + 1} - \frac{R_{n-1}(x)}{n - 1} - \frac{2(-1)^n}{n^2 - 1}\right). \tag{2.6}$$

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

$$\sum_{k=1}^{n} R_i(x_k)R_j(x_k) = \begin{cases} 0 & \text{if } i \neq j, \\ n & \text{if } i = j = 0, \\ \frac{n}{2} & \text{if } i = j \neq 0, \end{cases} \tag{2.7}$$

where x_k be the zeros of $R_n(x)$ defined in (2.3) and $i, j \in \{0, 1, 2, \ldots, n\}$.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

(iv) *Let x_k be the zeros of R_n defined in (2.3) and define the Chebyshev matrix $\mathbf{R}$ by*

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

Then, it has the multiplicative inverse

$$\mathbf{R}^{-1} = \frac{1}{n}\mathrm{diag}(1, 2, 2, \ldots, 2)\mathbf{R}^{\top}. \tag{2.8}$$

(v) *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 $R_1(x) = \frac{2x-a-b}{b-a}$.

Proof. (i) It is clear that if $\cos\left(n\arccos\left(\frac{2x-a-b}{b-a}\right)\right) = 0$, then x satisfies (2.3).

(ii) Let $x \in [a, b]$, it is easy to obtain the single-layer integrations of $R_0(x)$ and $R_1(x)$. For $n \geq 2$, we use the integration by substitution. Let $\cos\theta = \frac{2\xi-a-b}{b-a}$, we have

$$\begin{aligned}
\bar{R}_n(x) = \int_a^x R_n(\xi)d\xi &= \frac{b-a}{2}\int_{-1}^{\frac{2x-a-b}{b-a}} T_n(\cos\theta)\, d(\cos\theta) \\
&= \frac{b-a}{2}\int_{\arccos(-1)}^{\arccos(\frac{2x-a-b}{b-a})} \cos(n\theta)(-\sin\theta\, d\theta) \\
&= \frac{b-a}{4}\int_{\arccos(-1)}^{\arccos(\frac{2x-a-b}{b-a})} \big(\sin(n-1)\theta - \sin(n+1)\theta\big)\, d\theta \\
&= \frac{b-a}{4}\left(\frac{\cos(n+1)\theta}{n+1} - \frac{\cos(n-1)\theta}{n-1}\right)\Bigg|_{\theta=\arccos(-1)}^{\theta=\arccos(\frac{2x-a-b}{b-a})} \\
&= \frac{b-a}{4}\left(\frac{R_{n+1}(\xi)}{n+1} - \frac{R_{n-1}(\xi)}{n-1}\right)\Bigg|_{\xi=a}^{\xi=x} \\
&= \frac{b-a}{4}\left(\frac{R_{n+1}(x)}{n+1} - \frac{R_{n-1}(x)}{n-1} - \frac{2(-1)^n}{n^2-1}\right).
\end{aligned}$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

(iii) Recall the trigonometric identity [8],

$$\sum_{k=0}^{n} \cos(a+bk) = \frac{\sin\frac{(n+1)b}{2}\cos(a+\frac{nb}{2})}{\sin\frac{b}{2}}. \tag{2.9}$$

Let $i, j \in \{0,1,2,\ldots,n\}$, for $i \neq j$, by (2.2), (2.3) and (2.9), we have

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

Next, for $i = j = 0$, we have

$$\sum_{k=1}^{n} R_0(x_k)R_0(x_k) = \sum_{k=1}^{n} \cos\left(\frac{2k-1}{2n}(0)\pi\right)\cos\left(\frac{2k-1}{2n}(0)\pi\right) = \sum_{k=1}^{n} 1 = n.$$

Finally, for $i = j \neq 0$, we have

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

(iv) Let $\mathbf{Q} := \frac{1}{n}\mathrm{diag}(1,2,2,\ldots,2)\mathbf{R}^{\top}$. We can prove directly by using (2.7) to compute the products $\mathbf{RQ}$ and $\mathbf{QR}$, we obtain that they are the identity matrices.

(v) From (2.2), let $\frac{2x-a-b}{b-a} = \cos\theta$. Then, $R_n(x) = T_n(\cos\theta) = \cos(n\theta), \theta \in [0,\pi]$. So,

$$R_{n+1}(x) + R_{n-1}(x) = \cos(n+1)\theta + \cos(n-1)\theta = 2(\cos\theta)\cos(n\theta).$$

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

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.2 Chebyshev Expansion

In this section, we mention about the approximation of a given function $f(x)$ by a polynomial $p(x)$ that gives a uniform and accurate description in an interval $[a, b]$. We start by quoting a remarkable theorem by Weierstrass [19] without proof as follows.

Theorem 2.1. *If $f(x)$ is a continuous function in the interval $[a, b]$, then for each $\varepsilon > 0$, there exists a polynomial $p(x)$ such that $|f(x) - p(x)| < \varepsilon$ for all x in the given interval.*

Accordingly, by Theorem 2.1, for knowing $n + 1$ data points of a function $f(x)$, we can express the interpolating function $f(x)$ as a linear combination $p(x)$ of polynomial basis functions of degree less than or equal to n, $\varphi_0, \varphi_1, \varphi_2, \ldots, \varphi_n$ so that

$$f(x) \approx p(x) = \sum_{i=0}^{n} c_i \varphi_i(x).$$

Here, the coefficients $c_0, c_1, c_2, \ldots, c_n$ are to be determined by using the known data points of $f(x)$. Then, we can construct the system of linear equation for solving the unknown coefficients c_i which will be discussed in Section 2.4. Examples of polynomials that are frequently used to be the basis functions $\varphi_i(x)$ for $i \in \{0, 1, 2, \ldots, n\}$ are the followings

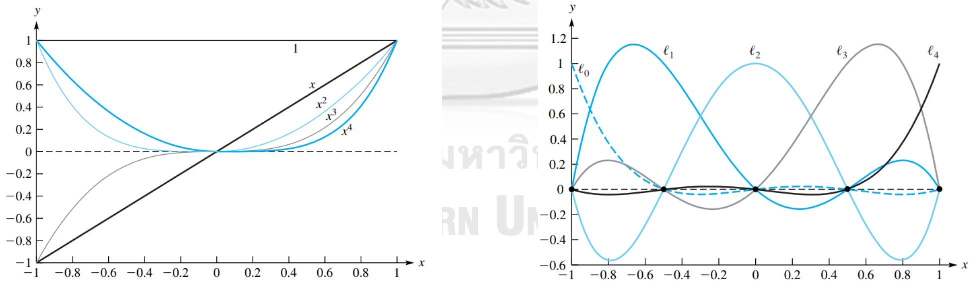
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

- **Monomials:** $\varphi_i(x) = x^i$,
- **Lagrange polynomials:** $\ell_i(x) = \prod_{j=0, j\neq i}^{n} \left(\frac{x - x_j}{x_i - x_j}\right)$,
- **Newton polynomials:** $\pi_i(x) = \prod_{j=0}^{i-1} (x - x_j)$,
- **Chebyshev polynomials:** $T_i(x) = \cos(i \arccos x)$.

Furthermore, we plot the first few mentioned-above polynomials as shown in Figure 2.2. For large degree n, we can observe that the three choices of basis functions: the monomials x^i, the Lagrange polynomials $\ell_i(x)$ and Newton polynomials $\pi_i(x)$, are less distinguishable from one another as depicted in Figures 2.2(a), 2.2(b) and 2.2(c). However, it turns out that there are better choices for the basis functions, namely, the Chebyshev polynomials.

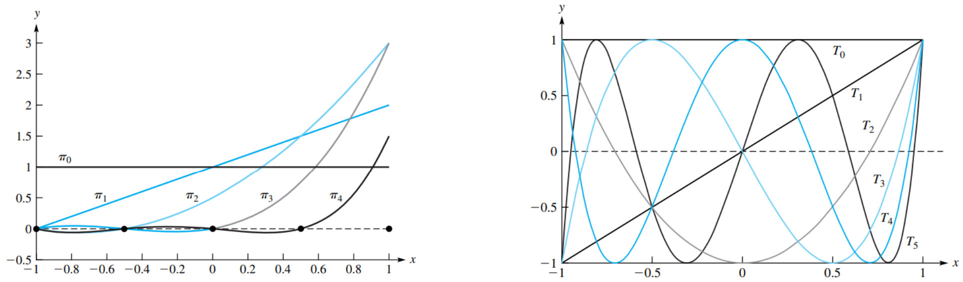
The Chebyshev polynomials play an important role in mathematics because they have several special properties such as the recursive relation and orthogonality. Their first six degrees produce the curves that are quite different from one another as shown in Figure 2.2(d). One of their important properties is the equal oscillation property. Notice from Figure 2.2(d) that successive extreme points of the Chebyshev polynomials are equal in magnitude and alternate in sign. This property tends to distribute the error uniformly when the Chebyshev polynomials are used as the basis functions. In polynomial interpolation for continuous functions, it is particularly advantageous to select each interpolation node as the zeros or the extreme points of a Chebyshev polynomial. This causes the maximum error over the interval of interpolation to be minimized.

In this dissertation, we consequently use the linear combination of Chebyshev polynomial basis functions, which is called the "Chebyshev expansion", to approximate the solution of our considered nonlinear differential equations. We also interpolate the approximate solution at each point by the zeros of Chebyshev polynomial of a certain degree. The reason for interpolating with the zeros will be detailed later.



(a) Monomials x^i

(b) Lagrange polynomials ℓ_i



(c) Newton polynomials π_i

(d) Chebyshev polynomials T_i

Figure 2.2: Plotting of the first few polynomials on $[-1, 1]$

However, the utility of polynomial interpolation cannot be excessively stretched. Next, let us quantify the errors that can occur in polynomial interpolation and develop techniques to minimize such errors. We begin with an interpolation error theorem.

Theorem 2.2. ([22]) *If $p(x)$ is the polynomial of degree at most n that interpolates a function $f \in C^{n+1}(a,b)$ at $n+1$ distinct points $x_0, x_1, x_2, \ldots, x_n \in [a,b]$, then for each $x \in [a,b]$, there exists a point $\xi \in (a,b)$ such that*

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i). \tag{2.10}$$

The preceding theorem not only tells us how large the error could be when a given function is replaced by an interpolating polynomial, but it also gives us a clue that how we might arrange things to make the error as small as possible. Let us write down again the expression (2.10) for the error term

$$E(x) := |f(x) - p(x)| = \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \right|,$$

for some $\xi \in (a,b)$. Now, we do not even know what ξ is, except that it is some point in the interval $[a,b]$ that depends on x. Thus, there is not much we can do with the term $f^{(n+1)}(\xi)$, particularly in physical applications, because we do not even know what f is. However, we can try to make the term $\prod_{i=0}^{n}(x - x_i)$ as small as possible by picking a suitable choice of nodes $\{x_i\}$.

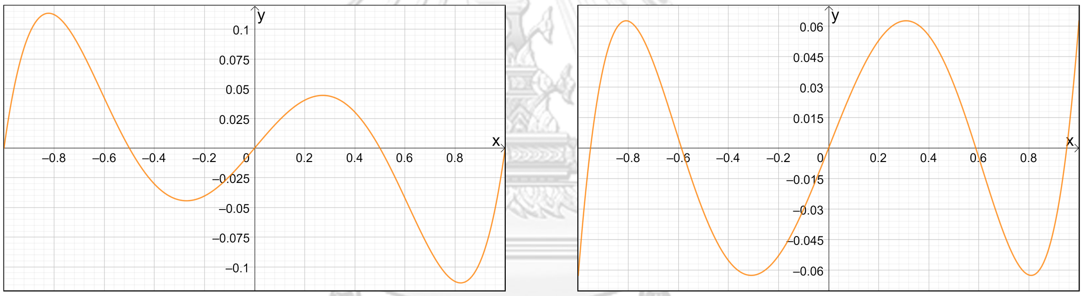
A special case that often arises is the one in which the interpolated nodes are equally spaced $x_i = a + ih$, where $h = \frac{b-a}{n}$. However, this case is the simplest choice of x_i that turns out to be not the best choice. Consider the case where $a = -1$, $b = 1$ and $n = 4$. Then, we have the nodes $x_i = -1 + 0.5i$ for $i = 0, 1, 2, 3, 4$. Thus, we set

$$w_1(x) := \prod_{i=0}^{n} (x - x_i) = (x+1)(x+0.5)x(x-0.5)(x-1)$$

and plot it as shown in Figure 2.3(a). We found that the polynomial $w_1(x)$ has a maximum value on the interval $[-1, 1]$ being around 0.1134. However, if we instead, for some strange reason, choose the points $x_0 = -0.9511$, $x_1 = -0.5878$, $x_2 = 0$, $x_3 = 0.5878$ and $x_4 = 0.9511$, then we have

$$w_2(x) := (x + 0.9511)(x + 0.5878)x(x - 0.5878)(x - 0.9511)$$

and plot this $w_2(x)$ in Figure 2.3(b). We obtain that the maximum values of the polynomial $w_2(x)$ on the interval $[-1, 1]$ is about 0.0624. We can see that the maximum value of $w_2(x)$ is approximately half of the maximum value of $w_1(x)$ that obtained from the equally spaced distribution of the nodes x_i. Thus, by choosing a special set of points x_i, it is possible to reduce the contribution of the factor $\prod_{i=0}^{n}(x - x_i)$ to the error term and also minimize the overall error of the polynomial interpolation.



(a) Polynomial $w_1(x)$ **(b)** Polynomial $w_2(x)$

Figure 2.3: Plotting of the polynomials $w_1(x)$ and $w_2(x)$ on $[-1, 1]$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Therefore, the question now becomes: how to choose a good set of nodal points to sample data, so that a polynomial interpolation is as accurate as possible? This is where Chebyshev polynomials come into play. It is well known that the Chebyshev polynomial $T_n(x)$ in (2.1) has the recursive relation, $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$. Then, we have the first six Chebyshev polynomials as follows: $T_0(x) = 1$, $T_1(x) = x$, $T_2(x) = 2x^2-1$, $T_3(x) = 4x^3-3x$, $T_4(x) = 8x^4-8x^2+1$ and $T_5(x) = 16x^5-20x^3+5x$. We note here that the leading coefficient of the Chebyshev polynomial $T_n(x)$ is 2^{n-1}.

Definition 2.2. A polynomial of degree n is called "monic" if the coefficient of x^n is 1.

Note that expressions of the form $\prod_{i=0}^{n}(x-x_i)$ are monic polynomials as well as the polynomials obtained from the Chebyshev polynomials divided by 2^{n-1} which are written in the form $\hat{T}_n(x) := 2^{1-n}T_n(x)$. Our first application of the Chebyshev polynomials is to prove a lower bound for maximum norm of a monic polynomial on $[-1,1]$.

Theorem 2.3. ([22]) *If* q_n *is a monic polynomial, then* $\|q_n\| = \max\limits_{x\in[-1,1]} |q_n(x)| \geq 2^{1-n}$.

Lemma 2.2. ([22]) *The maximum norm* $\|\hat{T}_n\|$ *on the interval* $[-1,1]$ *is* 2^{1-n}.

From Theorem 2.3 and Lemma 2.2, we have that $\|\hat{T}_n\| \leq \|q_n\|$ which means the monic Chebyshev polynomial $\hat{T}_n(x)$ is the polynomial of degree n that provides the smallest possible maximum norm for any other monic polynomial $q_n(x)$ of degree n.

Lemma 2.3. ([22]) *Let* $x_i = \cos\left(\frac{2i+1}{2n+2}\pi\right)$ *for* $i \in \{0,1,2,\ldots,n\}$ *is the Chebyshev nodes. Then, the monic polynomial* $\prod_{i=0}^{n}(x-x_i) = \hat{T}_{n+1}(x) = 2^{-n}T_{n+1}(x)$.

Consequently, the scaled Chebyshev polynomial $\hat{T}_{n+1}(x) = 2^{-n}T_{n+1}(x)$ is monic of degree $n+1$ with the smallest maximum absolute value in $[-1,1]$. If its zeros are used to be the interpolated nodes, then we can conclude the error bound as follows.

Theorem 2.4. ([22]) *If the nodes* x_i *are chosen as the zeros of Chebyshev polynomial* $T_{n+1}(x)$, *that is* $x_i = \cos\left(\frac{2i+1}{2n+2}\pi\right)$ *for* $i \in \{0,1,2,\ldots,n\}$, *then the error term for the polynomial interpolation using the nodes* x_i *is bounded by*

$$E(x) = |f(x)-p(x)| \leq \frac{1}{2^n(n+1)!} \max_{\xi\in[-1,1]} \left|f^{(n+1)}(\xi)\right|.$$

Moreover, this is the best upper bound, we can achieve by varying the choice of x_i.

Note that for an arbitrary interval $[a,b]$, if we use the zeros x_i of Chebyshev polynomial $R_{n+1}(x)$ as defined in (2.3), then the error bound consuming this zeros x_i is

$$E(x) = |f(x)-p(x)| \leq \frac{1}{2^n(n+1)!}\left(\frac{b-a}{2}\right)^{n+1} \max_{\xi\in[a,b]} \left|f^{(n+1)}(\xi)\right|.$$

Usually, the hiring of Chebyshev nodes is the best practical possibility for interpolation and certainly much better than equispaced interpolation.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.3 Kronecker Product

The Kronecker product in mathematics is an operation on two matrices of arbitrary size resulting in a block matrix. It should not be confused with the usual matrix multiplication, which is an entirely different operation. The Kronecker product was first proposed by Kronecker. Actually, we utilize the definition and some properties of the Kronecker product from [72] to describe a relation of our Chebyshev integration matrices in two-dimensional spaces.

Definition 2.3. ([72]) 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:

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

Let us provide the fact of Kronecker product without proof as follows.

Theorem 2.5. ([72]) *The Kronecker product satisfies the following properties:*

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

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

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

(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* $\mathbf{P} := [\mathbf{I}_n\otimes\mathbf{e}_1, \mathbf{I}_n\otimes\mathbf{e}_2, \mathbf{I}_n\otimes\mathbf{e}_3, \ldots, \mathbf{I}_n\otimes\mathbf{e}_m]$ *be an* $mn \times mn$ *permutation matrix, where* $\mathbf{I}_n$ *is an* $n\times n$ *identity matrix and* $\mathbf{e}_i := [0, 0, \ldots, 0, 1, 0, \ldots, 0]^\top$ *is an* m*-dimensional column vector which has 1 in the* i^{th} *position and 0's elsewhere. Then,* $\mathbf{P}(\mathbf{A}\otimes\mathbf{B})\mathbf{P}^\top = \mathbf{B}\otimes\mathbf{A}$.

2.4 Developed FIM-CPE

In this section, we develop the modified FIM proposed by Boonklurb et al. [14] in order to be applicable on arbitrary domain without any transformation both in one- and two-dimensional spaces. The main material for this scheme is to construct matrix representations for the integral operator of Chebyshev polynomial expansion in the general domain that called “Chebyshev integration matrix”. However, we divide our developed FIM-CPE into two parts for creating its one- and two-dimensional Chebyshev integration matrices for solving the problems of nonlinear PDEs as follows.

2.4.1 One-Dimensional Chebyshev Integration Matrices

First, let $M \in \mathbb{N}$, $a, b \in \mathbb{R}$ such that $a < b$ and $u(x)$ be a linear combination of Chebyshev polynomials $R_0(x), R_1(x), R_2(x), \ldots, R_{M-1}(x)$, that is

$$u(x) = \sum_{n=0}^{M-1} c_n R_n(x), \quad \text{for } x \in [a, b], \tag{2.11}$$

where each c_n is an unknown coefficient. Let $x_1 < x_2 < x_3 < \cdots < x_M$ be nodal points that are generated by the zeros of Chebyshev polynomial $R_M(x)$ as defined in (2.3). After substituting them into (2.11), we have

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$u(x_k) = \sum_{n=0}^{M-1} c_n R_n(x_k)$$

for $k \in \{1, 2, 3, \ldots, M\}$. We can express them into the 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 we denote it by $\mathbf{u} = \mathbf{R}\mathbf{c}$. We notice here that $\mathbf{R}$ is invertible by Lemma 2.1 (iv) and it has the closed form (2.8). Thus, the unknown coefficient vector $\mathbf{c} = \mathbf{R}^{-1}\mathbf{u}$.

Next, we present the construction of one-dimensional Chebyshev integration matrix. Initially, let us consider the single-layer integration of $u(x)$ from a to x_k for $x_k \in [a, b]$, denoted by $U^{(1)}(x_k)$, as the following.

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

where each $\bar{R}_n$ is the integration of Chebyshev polynomial R_n defined in (2.4), (2.5) and (2.6). When each zero x_k for $k \in \{1, 2, 3, \ldots, M\}$ is varied to the above equation, it can be written in the matrix form

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

which we denote it 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" in one-dimensional space for our developed FIM-CPE. However, it can be expressed in another form as

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

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

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.

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

Therefore, we can extend the idea for constructing the Chebyshev integration matrix of the single-layer integration to the higher-layer integrations.

Now, let us consider the double-layer integration of $u(x)$ from a to the zero x_k, which is denoted by $U^{(2)}(x_k)$. Then, we obtain

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

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

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

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

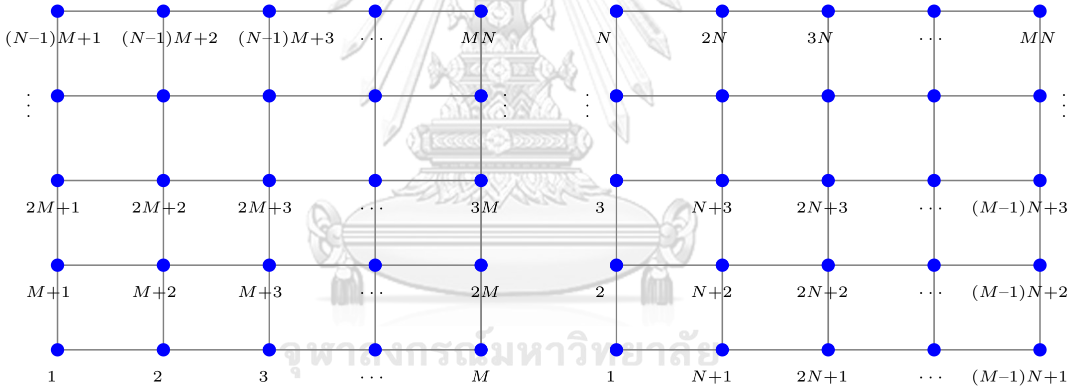
When the zeros x_k for $k \in \{1, 2, 3, \ldots, M\}$ are distributed in the above equation, each equation can be combined and expressed in the matrix form $\mathbf{U}^{(m)} = \mathbf{A}^m\mathbf{u}$ which is the matrix representation for m multiple-layer integration of $u(x)$ in the developed FIM-CPE.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.4.2 Two-Dimensional Chebyshev Integration Matrices

Next, we present the constructing two-dimensional Chebyshev integration matrices with respect to both 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 $x_k,\ k \in \{1, 2, 3, \ldots, M\}$ and $y_h,\ h \in \{1, 2, 3, \ldots, N\}$ be computational grid points over the domain $\Omega = [a, b] \times [c, d]$ along with the horizontal and vertical directions that are discretized by the zeros of the Chebyshev polynomials $R_M(x)$ and $R_N(y)$ defined in (2.3), respectively. Therefore, we have the total number of grid points in the system to be MN nodes.

For calculating convenience, we label the indices in numbering system of the grid points along the x-direction as Figure 2.4(a) and the y-direction as Figure 2.4(b) which are consecutively called the global and local numbering systems.



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

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Figure 2.4: The indices of the grid points globally and locally

First, let us consider the single-layer integrations of $u(x, y)$ with respect to the variables x and y that are denoted by $U_x^{(1)}$ and $U_y^{(1)}$, respectively. For $U_x^{(1)}(x_k, y)$ in the global numbering system, when y is fixed, we can see that its integral depends on only one variable x. Thus, we can utilize the idea in one dimension to construct the Chebyshev integration matrix with respect to x in two dimensions as

$$U_x^{(1)}(x_k, y) := \int_a^{x_k} u(\xi, y)\, d\xi = \sum_{i=1}^{M} a_{ki} u(x_i, y), \tag{2.12}$$

for $k \in \{1, 2, 3, \ldots, M\}$, it can be expressed in the matrix form as $\mathbf{U}_x^{(1)}(\cdot, y) = \mathbf{A}_M \mathbf{u}(\cdot, y)$, where $\mathbf{A}_M = \overline{\mathbf{R}}\mathbf{R}^{-1}$ is an $M \times M$ matrix. Thus, for each $y \in \{y_1, y_2, y_3, \ldots, y_N\}$,

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

which is represented in the matrix form by $\mathbf{U}_x^{(1)} = \mathbf{A}_x \mathbf{u}$, where $\mathbf{A}_x = \mathbf{I}_N \otimes \mathbf{A}_M$ is called the "two-dimensional Chebyshev integration matrix with respect to x" and $\otimes$ is the Kronecker product described in Section 2.3. Similarly, for $U_y^{(1)}(x, y_h)$, when x is fixed, it can be expressed in the local numbering system as

$$U_y^{(1)}(x, y_h) := \int_c^{y_h} u(x, \eta)\, d\eta = \sum_{j=1}^{N} a_{hj} u(x, y_j), \tag{2.13}$$

for $h \in \{1, 2, 3, \ldots, N\}$, it can be written as $\widetilde{\mathbf{U}}_y^{(1)}(x, \cdot) = \mathbf{A}_N \mathbf{u}(x, \cdot)$, where $\mathbf{A}_N = \overline{\mathbf{R}}\mathbf{R}^{-1}$ is an $N \times N$ matrix. Hence, for each $x \in \{x_1, x_2, x_3, \ldots, x_M\}$,

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

which imposed by $\widetilde{\mathbf{U}}_y^{(1)} = \widetilde{\mathbf{A}}_y \widetilde{\mathbf{u}}$, where $\widetilde{\mathbf{A}}_y = \mathbf{I}_M \otimes \mathbf{A}_N$. We notice that the elements of $\mathbf{u}$ and $\widetilde{\mathbf{u}}$ are the same, but the positions are different in the numbering system. Thus, we can transform $\widetilde{\mathbf{U}}_y^{(1)}$ and $\widetilde{\mathbf{u}}$ in local to global numbering system by using the permutation matrix $\mathbf{P} = [p_{ij}]_{MN \times MN}$, where each p_{ij} is defined by

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

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

for all $k \in \{1, 2, 3, \ldots, M\}$ and $h \in \{1, 2, 3, \ldots, N\}$. Note that this permutation matrix $\mathbf{P}$ is equivalent to the permutation matrix $\mathbf{P}$ in Theorem 2.5 (iii). It plays an important role to transform the i^{th} point in local numbering system to the j^{th} point in global numbering system. Moreover, it is not only trivial that $\mathbf{P}$ is nonsingular matrix, but it also has the multiplicative inverse $\mathbf{P}^{-1} = \mathbf{P}^{\top}$. For instance, if $M = 4$ and $N = 3$, then

$$\mathbf{P} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}.$$

Therefore, we obtain that $\mathbf{U}_y^{(1)} = \mathbf{P}\widetilde{\mathbf{U}}_y^{(1)}$ and $\mathbf{u} = \mathbf{P}\tilde{\mathbf{u}}$. Also, we can reformulate them to attain that $\mathbf{U}_y^{(1)} = \mathbf{A}_y\mathbf{u}$, where $\mathbf{A}_y = \mathbf{P}\widetilde{\mathbf{A}}_y\mathbf{P}^{-1} = \mathbf{P}(\mathbf{I}_M \otimes \mathbf{A}_N)\mathbf{P}^{\top} = \mathbf{A}_N \otimes \mathbf{I}_M$ is called the "two-dimensional Chebyshev integration matrix with respect to y" of our developed FIM-CPE in the global numbering system.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Next, let us consider the double-layer integration along both x- and y-directions, which consist of the integrations with respect to the variables x and x, y and y, x and y, and y and x, denoted by $U_x^{(2)}$, $U_y^{(2)}$, $U_{xy}^{(2)}$ and $U_{xy}^{(2)}$, respectively. They can be expressed in the global numbering system as follows.

For the double-layer integration with respect to the variables x and x or $U_x^{(2)}(x_k, y)$ when y is fixed to be a constant and use (2.12), we can perform the process similar to the one in one dimension to obtain that

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

for $k \in \{1, 2, 3, \dots, M\}$, it can be written in the matrix form $\mathbf{U}_x^{(2)}(\cdot, y) = \mathbf{A}_M^2 \mathbf{u}(\cdot, y)$. Therefore, after substituting each the zero $y \in \{y_1, y_2, y_3, \dots, y_N\}$, we have

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

which can be written in the matrix form $\mathbf{U}_x^{(2)} = \mathbf{A}_x^2 \mathbf{u}$, where $\mathbf{A}_x^2 = \mathbf{I}_N \otimes \mathbf{A}_M^2$.

For the double-layer integration with respect to the variables y and y or $U_y^{(2)}(x, y_h)$ when x is fixed to be a constant and use (2.13), then we can operate it similar to the process in one dimension along with the local numbering system as

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

for $h \in \{1, 2, 3, \dots, N\}$, it can be expressed in the matrix form $\mathbf{U}_y^{(2)}(x, \cdot) = \mathbf{A}_N^2 \mathbf{u}(x, \cdot)$. Therefore, after substituting each the zero $x \in \{x_1, x_2, x_3, \dots, x_M\}$, we obtain

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

which can be written in the matrix form $\widetilde{\mathbf{U}}_y^{(2)} = \widetilde{\mathbf{A}}_y^2 \widetilde{\mathbf{u}}$, where $\widetilde{\mathbf{A}}_y^2 = \mathbf{I}_M \otimes \mathbf{A}_N^2$. However, we notice here that the index arrangement of nodes in this matrix equation is labeled along with the local numbering system. Hence, we can transform these nodes globally by consuming the mentioned-above permutation matrix $\mathbf{P}$. Accordingly, we obtain the matrix representation of double-layer integration concerning the variable y only in the global numbering system as $\mathbf{U}_y^{(2)} = \mathbf{A}_y^2 \mathbf{u}$, where $\mathbf{A}_y^2 = \mathbf{P} \widetilde{\mathbf{A}}_y^2 \mathbf{P}^\top = \mathbf{A}_N^2 \otimes \mathbf{I}_M$.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Remark 2.1. Similarly, we can create the matrix representation for m multiple-layer integration by utilizing the same idea as in constructing $\mathbf{U}_x^{(2)}$ and $\mathbf{U}_y^{(2)}$. Then, for the higher-order Chebyshev integration matrices with respect to the variables x only and y only in the global numbering system which can be respectively expressed in the matrix forms, for $m \in \mathbb{N}$, as follow:

$$\begin{aligned}\mathbf{U}_x^{(m)} &= \mathbf{A}_x^m\mathbf{u}, \text{ where } \mathbf{A}_x^m = \mathbf{I}_N \otimes \mathbf{A}_M^m,\\ \mathbf{U}_y^{(m)} &= \mathbf{A}_y^m\mathbf{u}, \text{ where } \mathbf{A}_y^m = \mathbf{A}_N^m \otimes \mathbf{I}_M.\end{aligned}$$

For the double-layer integration with respect to the variables x and y or $U_{xy}^{(2)}(x_k, y_h)$. By using (2.12) and (2.13), we have

$$U_{xy}^{(2)}(x_k, y_h) = \int_c^{y_h}\int_a^{x_k} u(\xi,\eta)\,d\xi d\eta = \sum_{j=1}^{N}\sum_{i=1}^{M} a_{hj}a_{ki}u(x_i, y_j) \tag{2.14}$$

for $k \in \{1,2,3,\ldots,M\}$ and $h \in \{1,2,3,\ldots,N\}$, it can be separated into two types as:

<u>**Type I :**</u> If y_h is fixed, but x_k is varied, then (2.14) can be written in the matrix form

$$\begin{aligned}\mathbf{U}_{xy}^{(2)}(\cdot, y_h) &= \sum_{j=1}^{N} a_{hj}\mathbf{A}_M\mathbf{u}(\cdot, y_j)\\
&= \begin{bmatrix} a_{h1}\mathbf{A}_M & 0 & \cdots & 0\\ 0 & a_{h2}\mathbf{A}_M & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0\\ 0 & \cdots & 0 & a_{hN}\mathbf{A}_M\end{bmatrix}\begin{bmatrix}\mathbf{u}(\cdot, y_1)\\ \mathbf{u}(\cdot, y_2)\\ \vdots\\ \mathbf{u}(\cdot, y_N)\end{bmatrix}\\
&= \begin{bmatrix} a_{h1}\mathbf{I}_M\\ a_{h2}\mathbf{I}_M\\ \vdots\\ a_{hN}\mathbf{I}_M\end{bmatrix}^\top \underbrace{\begin{bmatrix}\mathbf{A}_M & 0 & \cdots & 0\\ 0 & \mathbf{A}_M & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0\\ 0 & \cdots & 0 & \mathbf{A}_M\end{bmatrix}}_{N\text{ blocks}}\begin{bmatrix}\mathbf{u}(\cdot, y_1)\\ \mathbf{u}(\cdot, y_2)\\ \vdots\\ \mathbf{u}(\cdot, y_N)\end{bmatrix},\end{aligned}$$

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

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

By varying all $y_h \in \{y_1, y_2, y_3, \ldots, y_N\}$ in the above equation, we obtain the block matrix

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

which we denote it by $\mathbf{U}_{xy}^{(2)} = (\mathbf{A}_N \otimes \mathbf{I}_M)(\mathbf{I}_N \otimes \mathbf{A}_M)\mathbf{u} = \mathbf{A}_y\mathbf{A}_x\mathbf{u}$ from Remark 2.1.

Type II : If x_k is fixed, but y_h is varied, then (2.14) can be written in the matrix form

$$\begin{aligned} \mathbf{U}_{xy}^{(2)}(x_k, \cdot) &= \sum_{i=1}^{M} a_{ki}\mathbf{A}_N\mathbf{u}(x_i, \cdot) \\ &= \begin{bmatrix} a_{k1}\mathbf{A}_N & 0 & \cdots & 0 \\ 0 & a_{k2}\mathbf{A}_N & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{kM}\mathbf{A}_N \end{bmatrix} \begin{bmatrix} \mathbf{u}(x_1, \cdot) \\ \mathbf{u}(x_2, \cdot) \\ \vdots \\ \mathbf{u}(x_M, \cdot) \end{bmatrix} \\ &= \begin{bmatrix} a_{k1}\mathbf{I}_N \\ a_{k2}\mathbf{I}_N \\ \vdots \\ a_{kM}\mathbf{I}_N \end{bmatrix}^{\top} \underbrace{\begin{bmatrix} \mathbf{A}_N & 0 & \cdots & 0 \\ 0 & \mathbf{A}_N & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{A}_N \end{bmatrix}}_{M \text{ blocks}} \begin{bmatrix} \mathbf{u}(x_1, \cdot) \\ \mathbf{u}(x_2, \cdot) \\ \vdots \\ \mathbf{u}(x_M, \cdot) \end{bmatrix}, \end{aligned}$$

where a_{ki} is an element at k^{th} row and i^{th} column of Chebyshev integration matrix $\mathbf{A}_M$. By varying all $x_k \in \{x_1, x_2, x_3, \ldots, x_M\}$ in the above equation, we get the block matrix

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

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

which we denote it by $\widetilde{\mathbf{U}}_{xy}^{(2)} = (\mathbf{A}_M \otimes \mathbf{I}_N)(\mathbf{I}_M \otimes \mathbf{A}_N)\widetilde{\mathbf{u}} = \widetilde{\mathbf{A}}_x\widetilde{\mathbf{A}}_y\widetilde{\mathbf{u}}$ which is in the local numbering system. Nevertheless, we can transform it globally by hiring the mentioned-above permutation matrix $\mathbf{P}$. Then, we obtain

$$\mathbf{U}_{xy}^{(2)} = \mathbf{P}\widetilde{\mathbf{U}}_{xy}^{(2)} = \mathbf{P}\big(\widetilde{\mathbf{A}}_x\widetilde{\mathbf{A}}_y\widetilde{\mathbf{u}}\big) = \mathbf{P}\left(\mathbf{P}^{-1}\mathbf{A}_x\mathbf{P}\right)\left(\mathbf{P}^{-1}\mathbf{A}_y\mathbf{P}\right)\left(\mathbf{P}^{-1}\mathbf{u}\right) = \mathbf{A}_x\mathbf{A}_y\mathbf{u},$$

where $\mathbf{A}_x$ and $\mathbf{A}_y$ are defined in Remark 2.1. Also, we observe that they are commutative. In fact, $\mathbf{A}_y\mathbf{A}_x = (\mathbf{A}_N \otimes \mathbf{I}_M)(\mathbf{I}_N \otimes \mathbf{A}_M) = \mathbf{A}_N \otimes \mathbf{A}_M = (\mathbf{I}_N \otimes \mathbf{A}_M)(\mathbf{A}_N \otimes \mathbf{I}_M) = \mathbf{A}_x\mathbf{A}_y$ by Theorem 2.5. Hence, we obtain the matrix representation $\mathbf{U}_{xy}^{(2)} = \mathbf{A}_y\mathbf{A}_x\mathbf{u} = \mathbf{A}_x\mathbf{A}_y\mathbf{u}$.

For the double-layer integration with respect to the variables y and x or $U_{yx}^{(2)}(x, y)$. By using (2.12) and (2.13), we have

$$U_{yx}^{(2)}(x_k, y_h) = \int_a^{x_k}\int_c^{y_h} u(\xi, \eta)\, d\eta d\xi = \sum_{i=1}^{M}\sum_{j=1}^{N} a_{ki}a_{hj}u(x_i, y_j) = U_{xy}^{(2)}(x_k, y_h)$$

for all $x_k \in \{x_1, x_2, x_3, \ldots, x_M\}$ and $y_h \in \{y_1, y_2, y_3, \ldots, y_N\}$, which we can see that the double-layer integrations concerning x and y with y and x are equal in the global system. Thereby, the above equation can be written as $\mathbf{U}_{yx}^{(2)} = \mathbf{U}_{xy}^{(2)} = \mathbf{A}_y\mathbf{A}_x\mathbf{u} = \mathbf{A}_x\mathbf{A}_y\mathbf{u}$.

Remark 2.2. The double-layer integration can be easily extended to the multiple-layer integrations, that are the m^{th}-order Chebyshev integration matrix with respect to x and the n^{th}-order Chebyshev integration matrix with respect to y in the global numbering system, which can be represented in the matrix forms

$$\begin{aligned}\mathbf{U}_{xy}^{(m,n)} &= \mathbf{A}_x^m\mathbf{A}_y^n\mathbf{u}, \text{ where } \mathbf{A}_x^m\mathbf{A}_y^n = \mathbf{A}_N^n \otimes \mathbf{A}_M^m,\\ \mathbf{U}_{yx}^{(n,m)} &= \mathbf{A}_y^n\mathbf{A}_x^m\mathbf{u}, \text{ where } \mathbf{A}_y^n\mathbf{A}_x^m = \mathbf{A}_N^n \otimes \mathbf{A}_M^m.\end{aligned}$$

Here, $\mathbf{A}_x$ and $\mathbf{A}_y$ are the first-order Chebyshev integration matrices with respect to the variables x and y defined in Remark 2.1, respectively.

CHAPTER III

BURGERS' EQUATION WITH SHOCK WAVE

In this section, we first provide the briefly physical meaning of Burgers' equation. Then, we apply the developed FIM-CPE to devise a numerical algorithm for finding approximate solutions of the Burgers' equation with shock wave (1.1). After that, several experimental examples are implemented in order to show the accuracy of our algorithm is better than many existing methods through various measurements.

3.1 Burgers' Equation

The Burgers' equation is a simple equation to understand the main properties of the Navier-Stokes equation [24]. In this one-dimensional equation the pressure is neglected, but the effects of the nonlinear and viscous terms remain. Then, it is written as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2},$$

which is usually known as Burgers' equation. This Burgers' equation is balance between time evolution, nonlinearity and diffusion. This is the simplest model of nonlinear equation for diffusive waves in fluid dynamics. In 1948, Burgers [15] first developed this equation primarily to throw light on turbulence described by the interaction of two opposite effects of convection and diffusion. Thenceforth, the Burgers' equation is popular until now to study a thorough behavior of the phenomenon of turbulence. The meaning of each term in this equation is given as follows. The variable u is a velocity of the traveling wave. The term u_t is the acceleration of the traveling wave. The nonlinear convection term uu_x will have a shocking up effect that causes waves to break. The viscous term νu_{xx} is a diffusion term like the one occurring in the heat equation. Other details regarding the Burgers' equation can be found in [11] and the references therein. However, we attempt to find a traveling wave solution u by the developed FIM-CPE in the next section.

3.2 Algorithm for Solving Burgers' Equation

Now, we apply our developed FIM-CPE in one-dimensional space to construct the numerical algorithm for solving the one-dimensional nonlinear Burgers' equation with a shock wave of (1.1) in order to achieve a high accurate approximate result. Let u be an approximate solution of v in (1.1). Then, we have the following one-dimensional nonlinear Burgers' equation with shock wave as

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

subject to the initial condition:

$$u(x,0) = \phi(x), \quad x \in [a,b], \tag{3.2}$$

and the Dirichlet boundary conditions:

$$\begin{aligned} u(a,t) &= \psi_1(t), \quad t \in (0,1], \\ u(b,t) &= \psi_2(t), \quad t \in (0,1]. \end{aligned} \tag{3.3}$$

where $a < b \in \mathbb{R}$, $T \in \mathbb{R}^+$, ϕ, ψ_1 and ψ_2 are the given sufficiently continuous functions, $\nu > 0$ is a constant coefficient of kinematic viscosity, and u is an unknown function of the space x and time t, respectively, to be determined. Assume that u is a smooth real-valued function of the temporal coordinate. Then, we obtain that the functions $u(x,t)$ at any two-consecutive times provide the values closely. In other words, let the two-consecutive times $0 \le t_{m-1} < t_m$ for $m \in \mathbb{N}$. If $|t_{m-1} - t_m| \to 0$, then $|u(x,t_{m-1}) - u(x,t_m)| \to 0$. Hence, this assumption is sufficient to employ the linearization under the time variable.

Let us first uniformly discretizes the temporal domain $(0,T]$ by specifying each time point $t_m = m\Delta t$ for $m \in \mathbb{N}$ into (3.1). Afterward, we linearize the nonlinear term in (3.1). In fact, it can be linearized to many types. In this work, it is linearized by

$$\frac{\partial u^{\langle m\rangle}(x)}{\partial t} + u^{\langle m-1\rangle}(x)\frac{\partial u^{\langle m\rangle}(x)}{\partial x} = \nu\frac{\partial^2 u^{\langle m\rangle}(x)}{\partial x^2},$$

where $u^{\langle m\rangle}(x)=u(x,t_m)$ is the numerical value at m^{th} time iteration. Subsequently, we approximate the time derivative term in the above equation. Usually, an estimation of the derivative term with respect to time has several methods such as forward, backward and central difference schemes. The use of each method affects the convergence speed. The method chosen in this study is the first-order forward difference quotient which actually provides the time complexity $\mathcal{O}(\Delta t)$. Thus, we have

$$\frac{u^{\langle m\rangle}(x)-u^{\langle m-1\rangle}(x)}{\Delta t}+u^{\langle m-1\rangle}(x)\frac{\partial u^{\langle m\rangle}(x)}{\partial x}=\nu\frac{\partial^2 u^{\langle m\rangle}(x)}{\partial x^2}. \tag{3.4}$$

Applying our developed FIM to eliminate all spatial derivatives from (3.4) by taking the double-layer integral from a to $x_k\in(a,b)$, that each x_k is generated by the zeros of Chebyshev polynomial $R_M(x)$ as defined in (2.3), on both sides of (3.4). After that, we obtain the equivalent integral equation as follows,

$$\begin{aligned}&\int_a^{x_k}\int_a^{\xi_2}\left(\frac{u^{\langle m\rangle}(\xi_1)-u^{\langle m-1\rangle}(\xi_1)}{\Delta t}\right)d\xi_1 d\xi_2\\&+\int_a^{x_k}\int_a^{\xi_2}\left(u^{\langle m-1\rangle}(\xi_1)\frac{\partial u^{\langle m\rangle}(\xi_1)}{\partial \xi_1}\right)d\xi_1 d\xi_2=\nu u^{\langle m\rangle}(x_k)+d_1x_k+d_2,\end{aligned} \tag{3.5}$$

where d_1 and d_2 are arbitrary constants emerged from the process of integration.

Then, let us consider the integral of nonlinear term in (3.5) by letting it be $q(x_k)$. Using the linear combination (2.11) and the technique of integration by parts, we have

$$\begin{aligned}q(x_k)&:=\int_a^{x_k}\int_a^{\xi_2}\left(u^{\langle m-1\rangle}(\xi_1)\frac{\partial u^{\langle m\rangle}(\xi_1)}{\partial \xi_1}\right)d\xi_1 d\xi_2\\&=\int_a^{x_k}u^{\langle m-1\rangle}(\xi_2)u^{\langle m\rangle}(\xi_2)\,d\xi_2-\int_a^{x_k}\int_a^{\xi_2}\frac{\partial u^{\langle m-1\rangle}(\xi_1)}{\partial \xi_1}u^{\langle m\rangle}(\xi_1)\,d\xi_1d\xi_2\\&=\int_a^{x_k}u^{\langle m-1\rangle}(\xi_2)u^{\langle m\rangle}(\xi_2)\,d\xi_2-\int_a^{x_k}\int_a^{\xi_2}\sum_{n=0}^{M-1}c_n^{\langle m-1\rangle}R_n'(\xi_1)u^{\langle m\rangle}(\xi_1)\,d\xi_1d\xi_2\\&=\int_a^{x_k}u^{\langle m-1\rangle}(\xi_2)u^{\langle m\rangle}(\xi_2)\,d\xi_2-\int_a^{x_k}\int_a^{\xi_2}\mathbf{R}'(\xi_1)\mathbf{c}^{\langle m-1\rangle}u^{\langle m\rangle}(\xi_1)\,d\xi_1d\xi_2\\&=\int_a^{x_k}u^{\langle m-1\rangle}(\xi_2)u^{\langle m\rangle}(\xi_2)\,d\xi_2-\int_a^{x_k}\int_a^{\xi_2}\mathbf{R}'(\xi_1)\mathbf{R}^{-1}\mathbf{u}^{\langle m-1\rangle}u^{\langle m\rangle}(\xi_1)\,d\xi_1d\xi_2,\end{aligned}$$

where $\mathbf{R}'(\cdot)=\left[R_0'(\cdot),R_1'(\cdot),R_2'(\cdot),\ldots,R_{M-1}'(\cdot)\right]$ is the row vector of first-order derivative vector of Chebyshev polynomial and $\mathbf{R}^{-1}$ is defined by (2.8). Hence, by varying the zeros $x_k \in \{x_1, x_2, x_3, \dots, x_M\}$, the above equation can be written in the matrix form:

$$\begin{bmatrix} q(x_1) \\ q(x_2) \\ \vdots \\ q(x_M) \end{bmatrix} = \mathbf{A} \begin{bmatrix} u^{\langle m-1\rangle}(x_1)u^{\langle m\rangle}(x_1) \\ u^{\langle m-1\rangle}(x_2)u^{\langle m\rangle}(x_1) \\ \vdots \\ u^{\langle m-1\rangle}(x_M)u^{\langle m\rangle}(x_M) \end{bmatrix} - \mathbf{A}^2 \begin{bmatrix} \mathbf{R}'(x_1)\mathbf{R}^{-1}\mathbf{u}^{\langle m-1\rangle}u^{\langle m\rangle}(x_1) \\ \mathbf{R}'(x_2)\mathbf{R}^{-1}\mathbf{u}^{\langle m-1\rangle}u^{\langle m\rangle}(x_2) \\ \vdots \\ \mathbf{R}'(x_M)\mathbf{R}^{-1}\mathbf{u}^{\langle m-1\rangle}u^{\langle m\rangle}(x_M) \end{bmatrix}.$$

For computational convenience, we reduce the above matrix into the simplified form:

$$\mathbf{q} = \mathbf{A}\mathrm{diag}\left(\mathbf{u}^{\langle m-1\rangle}\right)\mathbf{u}^{\langle m\rangle} - \mathbf{A}^2\mathrm{diag}\left(\mathbf{R}'\mathbf{R}^{-1}\mathbf{u}^{\langle m-1\rangle}\right)\mathbf{u}^{\langle m\rangle} := \mathbf{Q}^{\langle m-1\rangle}\mathbf{u}^{\langle m\rangle}, \tag{3.6}$$

where $\mathbf{Q}^{\langle m-1\rangle} = \mathbf{A}\mathrm{diag}(\mathbf{u}^{\langle m-1\rangle}) - \mathbf{A}^2\mathrm{diag}(\mathbf{R}'\mathbf{R}^{-1}\mathbf{u}^{\langle m-1\rangle})$, $\mathbf{q} = [q(x_1), q(x_2), \dots, q(x_M)]^\top$, $\mathbf{u}^{\langle z\rangle} = \left[u^{\langle z\rangle}(x_1), u^{\langle z\rangle}(x_2), \dots, u^{\langle z\rangle}(x_M)\right]^\top$ for $z \in \mathbb{N} \cup \{0\}$, $\mathbf{A} = \bar{\mathbf{R}}\mathbf{R}^{-1}$ is the $M \times M$ one-dimensional Chebyshev integration matrix explained in Section 2.4.1 and

$$\mathbf{R}' = \begin{bmatrix} \mathbf{R}'(x_1) \\ \mathbf{R}'(x_2) \\ \vdots \\ \mathbf{R}'(x_M) \end{bmatrix} = \begin{bmatrix} R_0'(x_1) & R_1'(x_1) & \cdots & R_{M-1}'(x_1) \\ R_0'(x_2) & R_1'(x_2) & \cdots & R_{M-1}'(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_0'(x_M) & R_1'(x_M) & \cdots & R_{M-1}'(x_M) \end{bmatrix}. \tag{3.7}$$

Consequently, by substituting each zero $x_k \in \{x_1, x_2, x_3, \dots, x_M\}$ into the integral equation (3.5), we can convert them, that are substituted by each zero x_k, to the matrix form by employing (3.6) and our developed FIM-CPE as follows:

$$\frac{1}{\Delta t}\left(\mathbf{A}^2\mathbf{u}^{\langle m\rangle} - \mathbf{A}^2\mathbf{u}^{\langle m-1\rangle}\right) + \mathbf{Q}^{\langle m-1\rangle}\mathbf{u}^{\langle m\rangle} = \nu\mathbf{u}^{\langle m\rangle} + d_1\mathbf{x} + d_2\mathbf{i},$$

or it can be simplified as

$$\left(\frac{1}{\Delta t}\mathbf{A}^2 + \mathbf{Q}^{\langle m-1\rangle} - \nu\mathbf{I}\right)\mathbf{u}^{\langle m\rangle} - d_1\mathbf{x} - d_2\mathbf{i} = \frac{1}{\Delta t}\mathbf{A}^2\mathbf{u}^{\langle m-1\rangle}, \tag{3.8}$$

where $\mathbf{I}$ is an $M \times M$ identity matrix, $\mathbf{x} = [x_1, x_2, x_3, \dots, x_M]^\top$ and $\mathbf{i} = [1, 1, 1, \dots, 1]^\top$.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

From the given Dirichlet boundary conditions (3.3), we can change them into the vector forms by using the linear combination of Chebyshev polynomials (2.11) at m^{th} time iteration as follow:

$$u^{\langle m\rangle}(a)=\sum_{n=0}^{M-1}c_n^{\langle m\rangle}R_n(a)=\sum_{n=0}^{M-1}c_n^{\langle m\rangle}(-1)^n:=\mathbf{h}_l\mathbf{c}^{\langle m\rangle}=\mathbf{h}_l\mathbf{R}^{-1}\mathbf{u}^{\langle m\rangle}=\psi_1(t_m), \tag{3.9}$$

$$u^{\langle m\rangle}(b)=\sum_{n=0}^{M-1}c_n^{\langle m\rangle}R_n(b)=\sum_{n=0}^{M-1}c_n^{\langle m\rangle}(+1)^n:=\mathbf{h}_r\mathbf{c}^{\langle m\rangle}=\mathbf{h}_r\mathbf{R}^{-1}\mathbf{u}^{\langle m\rangle}=\psi_2(t_m), \tag{3.10}$$

where $t_m = m\Delta t$ for $m \in \mathbb{N}$, $\mathbf{h}_l = [1, -1, 1, \ldots, (-1)^{M-1}]$ and $\mathbf{h}_r = [1, 1, 1, \ldots, 1]$. Note that, for other types of boundary conditions, they can be similarly transformed the above Dirichlet boundary conditions into vector forms by using the Chebyshev expansion (2.11) together with derivative of Chebyshev polynomials.

Finally, from (3.8), (3.9) and (3.10), we can construct the following system of iterative linear equations for a total of $M+2$ unknowns containing $\mathbf{u}^{\langle m\rangle}$, d_1 and d_2 as

$$\left[\begin{array}{c|cc}\frac{1}{\Delta t}\mathbf{A}^2+\mathbf{Q}^{\langle m-1\rangle}-\nu\mathbf{I} & -\mathbf{x} & -\mathbf{i}\\ \hline \mathbf{h}_l\mathbf{R}^{-1} & 0 & 0\\ \mathbf{h}_r\mathbf{R}^{-1} & 0 & 0\end{array}\right]\left[\begin{array}{c}\mathbf{u}^{\langle m\rangle}\\ \hline d_1\\ d_2\end{array}\right]=\left[\begin{array}{c}\frac{1}{\Delta t}\mathbf{A}^2\mathbf{u}^{\langle m-1\rangle}\\ \hline \psi_1(t_m)\\ \psi_2(t_m)\end{array}\right]. \tag{3.11}$$

Accordingly, the solution $\mathbf{u}^{\langle m\rangle}$ can be approximated by solving the system (3.11) with starting from the given initial condition (3.2), i.e., $\mathbf{u}^{\langle 0\rangle} = [\phi(x_1), \phi(x_2), \phi(x_3), \ldots, \phi(x_M)]^\top$. Note that when we would like to calculate a numerical solution u at any $x \in [a, b]$ for the terminal time T, we can find it by utilizing (2.11) as the following formula:

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

where $\mathbf{R}(x) = [R_0(x), R_1(x), R_2(x), \ldots, R_{M-1}(x)]$ and $\mathbf{u}^{\langle m\rangle}$ is the final iteration of (3.11).

For computational convenience, we summarize all above-mentioned procedures to the following algorithm in form of pseudocode in order to find an approximate solution of the one-dimensional Burgers' equation by using our developed FIM-CPE.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Algorithm 1 Algorithm for solving Burgers' equation by the developed FIM-CPE

```
Input: $a$, $b$, $x$, $\nu$, $T$, $M$, $\Delta t$, $\phi(x)$, $\psi_1(t)$ and $\psi_2(t)$;
Output: An approximate solution $u(x,T)$;
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;
Compute $\mathbf{x}$, $\mathbf{i}$, $\mathbf{h}_l$, $\mathbf{h}_r$, $\mathbf{I}$, $\mathbf{R}'$, $\overline{\mathbf{R}}$, $\mathbf{R}^{-1}$, $\mathbf{R}(x)$ and $\mathbf{A}$;
Construct $\mathbf{u}^{\langle 0\rangle} = [\phi(x_1),\phi(x_2),\phi(x_3),\dots,\phi(x_M)]^\top$;
Set $m = 1$ and $t_1 = \Delta t$;
while $t_m \le T$ do
    Compute $\mathbf{Q}^{\langle m-1\rangle} = \mathbf{A}\mathrm{diag}(\mathbf{u}^{\langle m-1\rangle}) - \mathbf{A}^2\mathrm{diag}(\mathbf{R}'\mathbf{R}^{-1}\mathbf{u}^{\langle m-1\rangle})$;
    Find $\mathbf{u}^{\langle m\rangle}$ by solving the iterative linear system (3.11);
    Update $m = m + 1$;
    Compute $t_m = m\Delta t$;
end while
return $u(x,T) = \mathbf{R}(x)\mathbf{R}^{-1}\mathbf{u}^{\langle m\rangle}$;
```

3.3 Numerical Examples for Testing Algorithm 1

In this section, we apply the proposed Algorithm 1 based on the one-dimensional developed FIM-CPE for finding the approximate solutions of one-dimensional nonlinear Burgers' equations with a shock wave in order to illustrate the efficiency and accuracy. In the following Examples 3.1 and 3.2, the analytical solutions were obtained by using the Hopf-Cole transformation that Benton and Platzman [10] have surveyed. Their analytical solutions involve an infinite series, which may converge very slowly for the small viscosity ν. Then, Miller [45] has shown that these problems produce oscillations and instabilities for $\nu < 0.01$. We can see from our results that the proposed Algorithm 1 can reduce the effect of these problems, especially for small ν. The presented method also can be performed on Examples 3.3 and 3.4 that contain shock waves in their exact solutions. The accuracy of the obtained numerical solutions is measured in terms of the absolute error and also the error norms L_∞ and L_2.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Example 3.1. Consider the Burgers' equation (3.1) with initial and boundary conditions

$$u(x,0)=\sin(\pi x),\quad x\in[0,1],$$
$$u(0,t)=u(1,t)=0,\quad t>0.$$

The analytical solution given by Cole [16] of this equation is

$$u^*(x,t)=\frac{2\pi\nu\sum_{n=1}^{\infty}a_n\exp\left(-n^2\pi^2\nu t\right)n\sin\left(n\pi x\right)}{a_0+\sum_{n=1}^{\infty}a_n\exp\left(-n^2\pi^2\nu t\right)\cos\left(n\pi x\right)},\tag{3.12}$$

where the Fourier coefficients a_0 and a_n are

$$\begin{aligned}a_0&=\int_0^1\exp\left(\frac{\cos\left(\pi x\right)-1}{2\pi\nu}\right)dx,\\ a_n&=2\int_0^1\exp\left(\frac{\cos\left(\pi x\right)-1}{2\pi\nu}\right)\cos\left(n\pi x\right)dx,\quad n\in\{1,2,3,\dots\}.\end{aligned}$$

To find the numerical solutions $u(x,T)$ of Example 3.1 achieved by the proposed Algorithm 1, we can express the mean absolute error (MAE) at different discretizing values of grid numbers M and time steps Δt as displayed in Table 3.1 for the terminal time $T=1$ and the kinematic viscosity $\nu=0.01$. It is demonstrated that the large M and the small Δt affect the MAE to be vanished. Moreover, we choose parameters $\Delta t=10^{-4}$, $M=80$ and $\nu=0.01$. Then, our obtained results are compared with the numerical results obtained from FEM [34], FDM [16], original FIM [40] and their analytical solutions as shown in Table 3.2, which are measured by the absolute error. We can see that our numerical Algorithm 1 provides the accuracy more than those methods.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Table 3.1: MAE at different M and Δt of Example 3.1

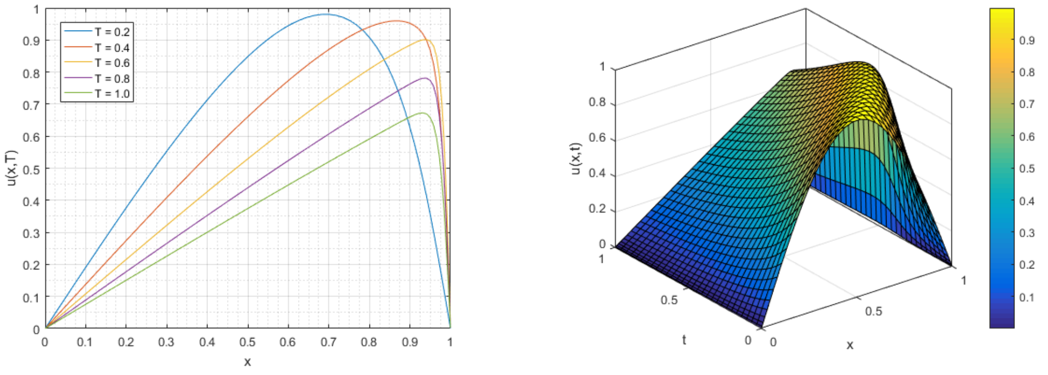
M	$\Delta t=10^{-1}$	$\Delta t=10^{-2}$	$\Delta t=10^{-3}$	$\Delta t=10^{-4}$	$\Delta t=10^{-5}$
10	4.1790×10^{-2}	7.1218×10^{-2}	1.1252×10^{-2}	1.2144×10^{-2}	1.2244×10^{-2}
20	5.8231×10^{-3}	2.5693×10^{-3}	2.4579×10^{-3}	2.4456×10^{-3}	2.4443×10^{-3}
30	5.0198×10^{-3}	4.0507×10^{-4}	1.0844×10^{-4}	1.0775×10^{-4}	1.0816×10^{-4}
40	4.9865×10^{-3}	3.8551×10^{-4}	3.9793×10^{-5}	5.8829×10^{-6}	4.7329×10^{-6}
50	4.9864×10^{-3}	3.8724×10^{-4}	3.8842×10^{-5}	3.9406×10^{-6}	4.6543×10^{-7}

Table 3.2: Comparison of absolute errors of Example 3.1

x	T	FEM [34]	FDM [16]	FIM [40]	Algorithm 1
0.25	0.4	6.28×10^{-3}	5.30×10^{-4}	8.00×10^{-5}	1.1647×10^{-6}
	0.6	6.40×10^{-3}	9.00×10^{-5}	5.00×10^{-5}	5.2590×10^{-7}
	0.8	6.04×10^{-3}	3.00×10^{-5}	3.00×10^{-5}	2.8243×10^{-7}
	1.0	5.56×10^{-3}	6.00×10^{-5}	2.00×10^{-5}	1.7484×10^{-7}
	3.0	2.43×10^{-3}	2.00×10^{-5}	1.00×10^{-5}	2.5503×10^{-8}
0.50	0.4	4.72×10^{-3}	1.08×10^{-2}	1.70×10^{-4}	1.4588×10^{-5}
	0.6	5.83×10^{-3}	4.64×10^{-3}	1.10×10^{-4}	6.4394×10^{-6}
	0.8	6.12×10^{-3}	2.29×10^{-3}	8.00×10^{-5}	3.2064×10^{-6}
	1.0	6.05×10^{-3}	1.26×10^{-3}	5.00×10^{-5}	1.7819×10^{-6}
	3.0	3.44×10^{-3}	2.00×10^{-3}	1.00×10^{-5}	9.2911×10^{-8}
0.75	0.4	1.75×10^{-3}	3.65×10^{-2}	2.80×10^{-4}	7.2687×10^{-5}
	0.6	4.08×10^{-3}	1.75×10^{-2}	1.90×10^{-4}	3.0208×10^{-5}
	0.8	5.14×10^{-3}	9.19×10^{-3}	1.30×10^{-4}	1.3926×10^{-5}
	1.0	5.52×10^{-3}	5.30×10^{-3}	9.00×10^{-5}	7.2688×10^{-6}
	3.0	3.92×10^{-3}	2.10×10^{-3}	2.00×10^{-5}	1.8491×10^{-7}

Then, the graphical solutions are depicted in Figure 3.1 including the propagation of travelling wave at different times T in two-dimensional graph and also at all times $t \in [0, 1]$ in three-dimensional graph by using the parameters $M = 40$, $\Delta t = 10^{-2}$ and $\nu = 10^{-2}$. From Figure 3.1, we can see that our proposed Algorithm 1 can handle the shock wave near $x = 1$ for this problem.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

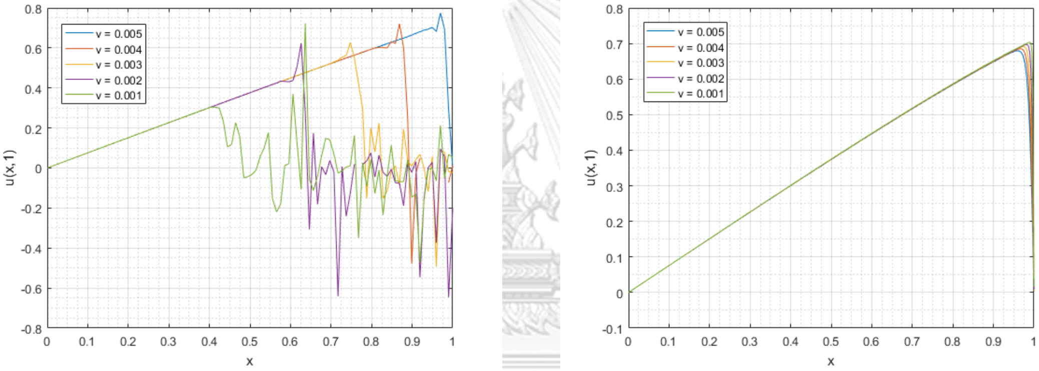


(a) $u(x, T)$ at different times T

(b) Surface plot of $u(x, t)$

Figure 3.1: Physical behavior of our solution in Example 3.1

Although this Example 3.1 has the analytical solution, it is in the form of infinite summations depended on the Fourier coefficients a_0 and a_n which are in the integral form. We can observe that it is difficult to seek the solution, because its coefficients a_0 and a_n cannot be directly integrated. Thus, it needs to apply the numerical integration for finding these values. In addition, the viscous value ν is also at the denominator which produces oscillations and instabilities for small ν. However, our Algorithm 1 can manipulate these issues. We demonstrate the behavior of solutions for various small $\nu = n \times 10^{-3}$, where $n \in \{1, 2, 3, 4, 5\}$ from both the exact formula and our method via plotting graphs as depicted in Figure 3.2. Obviously, the exact formula provides instability and oscillation which contrasts our method, it still preserves the behaving manner of the solution.



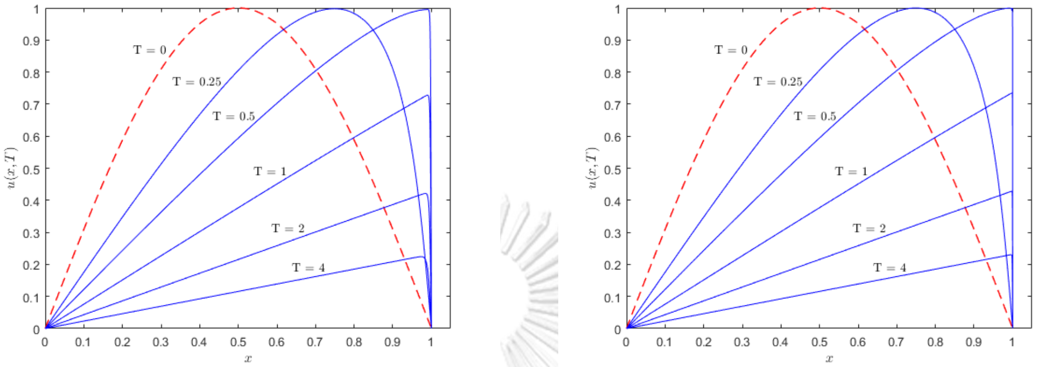
(a) Exact formula **(b)** Our Algorithm 1

Figure 3.2: Behavior of solutions for small values ν in Example 3.1

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Finally, we further illustrate the numerical results from the presented Algorithm 1 by selecting $M = 100$ and $\Delta t = 10^{-2}$ at the small viscous values $\nu \in \{10^{-3}, 10^{-4}\}$ as shown in Figure 3.3 when increasing the terminal times T. However, it is known that the analytical solution for $\nu < 10^{-2}$ is not practical because of the slow convergence of the infinite series. Therefore, these results are not compared to the analytical solution. From Figure 3.3, it demonstrates the development of a sharp front near $x = 1$ and afterwards the amplitude of the sharp front starts to decay. Additionally, the results for $\nu = 10^{-4}$ show the development of a sharp front at more early times earlier than those for $\nu = 10^{-3}$. Our numerical results are in good agreement with the results obtained by [27], [49] and

[62]. Moreover, Figure 3.3 also shows the enlarged profiles of waves in the vicinity of the right boundary at various terminal times T. We can obviously see that Algorithm 1 gives robust results even in the very vicinity of the right margin with a high peak. Thus, it is very advantageous when it comes to deal with a problem involving shock waves.



(a) the viscosity $\nu = 10^{-3}$ **(b)** the viscosity $\nu = 10^{-4}$

Figure 3.3: Profiles of our solution at small viscosity ν in Example 3.1

Example 3.2. Consider the Burgers' equation (3.1) with initial and boundary conditions

$$u(x,0) = 4x(1-x), \quad x \in [0,1],$$
$$u(0,t) = u(1,t) = 0, \quad t > 0.$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

The analytical solution of this equation is given by (3.12) with the Fourier coefficients

$$a_0 = \int_0^1 \exp\left(\frac{2x^3 - 3x^2}{3\nu}\right) dx,$$
$$a_n = 2\int_0^1 \exp\left(\frac{2x^3 - 3x^2}{3\nu}\right) \cos(n\pi x)\, dx, \quad n \in \{1,2,3,\dots\}.$$

In order to show the performance of Algorithm 1, the MAEs is sought at various discretizing values of grid numbers M and time steps Δt as expressed in Table 3.3 for the terminal time $T = 1$ and $\nu = 0.01$ which can be seen that the larger M and the smaller Δt make more accurate solutions increasingly. Moreover, we compare our numerical solutions $u(x,T)$ attained by Algorithm 1 of Example 3.2 at different times T with the approximate solutions obtained from the least-squares with quadratic B-spline FEM (Method 1) [34], the cubic spline quasi-interpolant with multi-node higher order expansion (Method 2) [65], the Taylor series expansion (Method 3) [4] and the cubic Hermite collocation method (Method 4) [21] together with their analytical solutions for using the parameters $\Delta t = 10^{-4}$ and $M = 80$ with the kinematic viscosity $\nu = 0.01$ are demonstrated in Table 3.4 which measured by the absolute error. We can see that our obtained solutions produce the accuracy more than those other methods. Additionally, we also show the physical behavior of our obtained solutions via the plotting graphs in Figure 3.4.

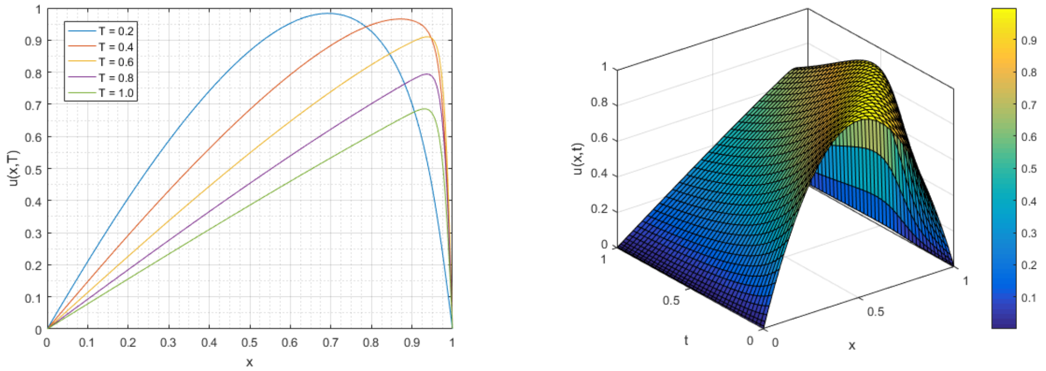
Table 3.3: MAE at different M and Δt of Example 3.2

M	$\Delta t = 10^{-1}$	$\Delta t = 10^{-2}$	$\Delta t = 10^{-3}$	$\Delta t = 10^{-4}$	$\Delta t = 10^{-5}$
10	4.6353×10^{-2}	1.2999×10^{-2}	4.1062×10^{-2}	1.6737×10^{-2}	3.0785×10^{-2}
20	6.6519×10^{-3}	2.7186×10^{-3}	2.6406×10^{-3}	2.6318×10^{-3}	2.6310×10^{-3}
30	5.8743×10^{-3}	5.2310×10^{-4}	1.1837×10^{-4}	1.1746×10^{-4}	1.1776×10^{-4}
40	5.8420×10^{-3}	5.0277×10^{-4}	5.1058×10^{-5}	6.6503×10^{-6}	5.2410×10^{-6}
50	5.8420×10^{-3}	5.0143×10^{-4}	5.0098×10^{-5}	5.0483×10^{-6}	5.8617×10^{-7}

Table 3.4: Comparison of absolute errors of Example 3.2

x	T	Method 1	Method 2	Method 3	Method 4	Algorithm 1
0.25	0.4	6.58×10^{-3}	5.93×10^{-5}	6.06×10^{-5}	9.37×10^{-6}	2.9384×10^{-6}
	0.6	6.99×10^{-3}	3.65×10^{-5}	5.34×10^{-5}	3.40×10^{-6}	1.3140×10^{-6}
	0.8	6.57×10^{-3}	5.11×10^{-5}	3.88×10^{-5}	1.88×10^{-5}	6.8860×10^{-7}
	1.0	5.99×10^{-3}	9.59×10^{-6}	2.95×10^{-5}	2.95×10^{-5}	4.1703×10^{-7}
	3.0	2.51×10^{-3}	3.45×10^{-5}	5.90×10^{-6}	4.09×10^{-6}	5.5235×10^{-8}
0.50	0.4	4.50×10^{-3}	7.86×10^{-5}	1.21×10^{-4}	1.39×10^{-6}	2.0152×10^{-5}
	0.6	5.93×10^{-3}	1.63×10^{-5}	8.36×10^{-5}	4.36×10^{-5}	1.0251×10^{-5}
	0.8	6.39×10^{-3}	1.35×10^{-5}	5.64×10^{-5}	1.64×10^{-5}	5.5454×10^{-6}
	1.0	6.38×10^{-3}	7.57×10^{-5}	4.42×10^{-5}	4.22×10^{-6}	3.2320×10^{-6}
	3.0	3.58×10^{-3}	2.00×10^{-5}	1.00×10^{-5}	1.78×10^{-8}	1.8408×10^{-7}
0.75	0.4	1.43×10^{-3}	3.14×10^{-7}	5.09×10^{-4}	3.14×10^{-7}	7.0463×10^{-5}
	0.6	3.76×10^{-3}	6.05×10^{-6}	2.46×10^{-4}	6.05×10^{-6}	3.4963×10^{-5}
	0.8	5.04×10^{-3}	2.03×10^{-5}	1.29×10^{-5}	3.03×10^{-5}	1.8200×10^{-5}
	1.0	5.59×10^{-3}	1.86×10^{-5}	8.13×10^{-4}	1.32×10^{-5}	1.0371×10^{-5}
	3.0	4.08×10^{-3}	4.30×10^{-5}	1.16×10^{-4}	3.04×10^{-6}	3.9847×10^{-7}

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



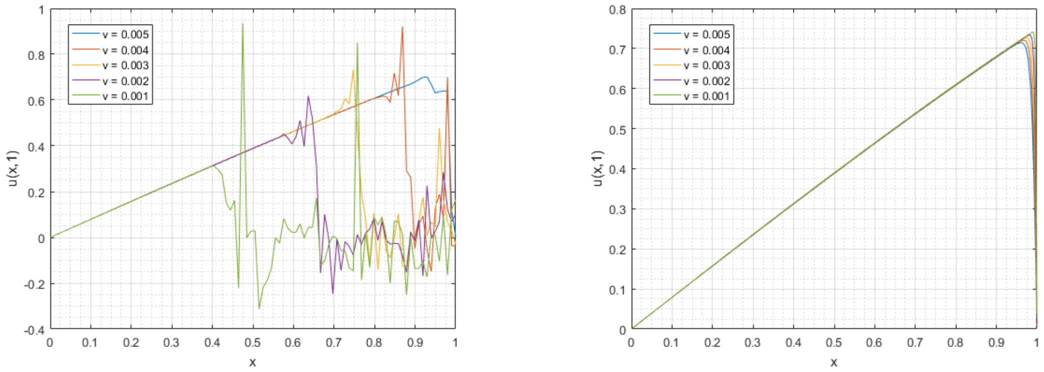
(a) $u(x, T)$ at different times T

(b) Surface plot of $u(x, t)$

Figure 3.4: Physical behavior of our solution in Example 3.2

Similarly, this Example 3.2 has the analytical solution as same as Example 3.1 that it is in the infinite summation form based on the Fourier coefficients a_0 and a_n. These coefficients cannot be directly integrations and also the small viscosity ν affect the solution instability and oscillation as shown in Figure 3.5(a). Anywise, our proposed Algorithm 1 can handle these issues as demonstrated via plotting graph in Figure 3.5(b). We can see that the solutions obtained from our method for small $\nu = n \times 10^{-3}$, where $n \in \{1, 2, 3, 4, 5\}$ has treated the turbulence of behaving solutions smoothly. Moreover, if the kinematic viscosity ν is very small, the obtained solution quite peaks near $x = 1$. Our presented Algorithm 1 can also manipulate it.

CHULALONGKORN UNIVERSITY

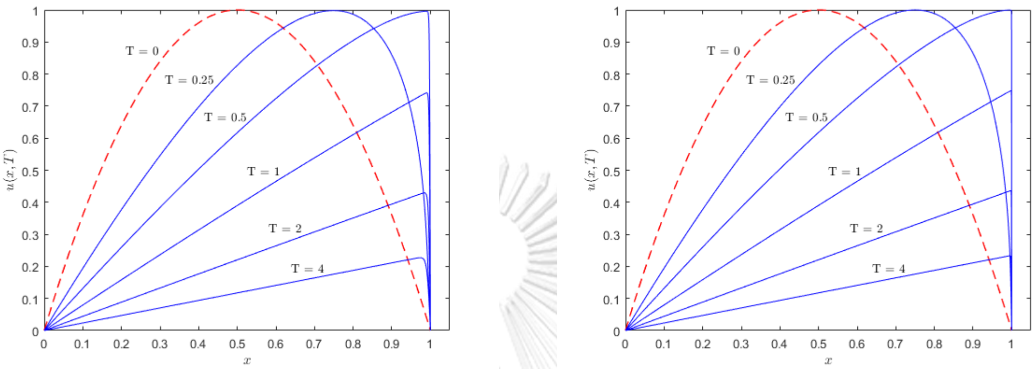


(a) Exact formula

(b) Our Algorithm 1

Figure 3.5: Behavior of solutions for small values ν in Example 3.2

Finally, we also demonstrate the numerical solutions obtained from Algorithm 1 by using $M = 100$ and $\Delta t = 10^{-2}$ at the small viscous values $\nu \in \{10^{-3}, 10^{-4}\}$ as depicted in Figure 3.3 when varying the terminal times T. These obtained numerical solutions of Example 3.2 provide the physical profiles of travelling wave as well as Example 3.1 which has the shocking up near $x = 1$.



(a) the viscosity $\nu = 10^{-3}$ **(b)** the viscosity $\nu = 10^{-4}$

Figure 3.6: Profiles of our solution at small viscosity ν in Example 3.2

Example 3.3. Consider the Burgers' equation (3.1) with initial and boundary condition

$$u(x,0) = \frac{2\pi\nu \sin(\pi x)}{\sigma + \cos(\pi x)}, \quad x \in [0,1]$$

$$u(0,t) = u(1,t) = 0, \quad t > 0.$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

The analytical solution given by Wood [64] for an arbitrary constant σ of this equation is

$$u^*(x,t) = \frac{2\pi\nu \exp\left(-\pi^2\nu t\right) \sin(\pi x)}{\sigma + \exp\left(-\pi^2\nu t\right) \cos(\pi x)}.$$

In order to demonstrate the efficiency of Algorithm 1, we express the accuracy via measuring a mean relative error (MRE) since this analytical solution of Example 3.3 closely approaches to zero. Table 3.5 shows the MREs at a variety of discretizing values to grid numbers M and time steps Δt for choosing parameters $\alpha = 2$, terminal time $T = 1$ and viscosity $\nu = 0.001$. The approximate results of Example 3.3 obtained by our presented Algorithm 1 for selecting parameters $\sigma = 2$, $M = 40$, $T = 0.001$ and $\Delta t = 10^{-4}$

with the different viscosity values $\nu \in \{0.5, 0.2, 0.1\}$ are compared with those achieved by Mittal [46] and Ganaie [21]. We can see in Table 3.6 that the developed FIM-CPE has less error in both L_∞ and L_2 norms than the other two methods. In Table 3.7, we compare our solutions versus the solutions reported by Mittal [46] and Rahman [53] for $\sigma = 100$, $T = 1$ and $\Delta t = 0.01$ with the viscosity $\nu = 0.005$ at the different nodal numbers $M \in \{10, 20, 40, 80\}$ which observe that the L_∞ and L_2 errors of our proposed FIM-CPE still provides slightly less than the errors of both Mittal [46] and Rahman [53]. Also, the graphical behaviors of our results with $\nu = 0.1$ and $\sigma = 2$ are displayed in Figure 3.7.

Table 3.5: MRE at different M and Δt of Example 3.3

M	$\Delta t = 10^{-1}$	$\Delta t = 10^{-2}$	$\Delta t = 10^{-3}$	$\Delta t = 10^{-4}$	$\Delta t = 10^{-5}$
10	2.2092×10^{-3}	2.2347×10^{-3}	2.2383×10^{-3}	2.2386×10^{-3}	2.2386×10^{-3}
20	2.0434×10^{-5}	2.1515×10^{-6}	3.4126×10^{-7}	1.6803×10^{-7}	1.6732×10^{-7}
30	2.0499×10^{-5}	2.0604×10^{-6}	2.0596×10^{-7}	2.0251×10^{-8}	4.3790×10^{-8}
40	2.0497×10^{-5}	2.0603×10^{-6}	2.0499×10^{-7}	2.2363×10^{-8}	1.6546×10^{-7}
50	2.0496×10^{-5}	2.0602×10^{-6}	2.0508×10^{-7}	3.2022×10^{-8}	5.7757×10^{-7}

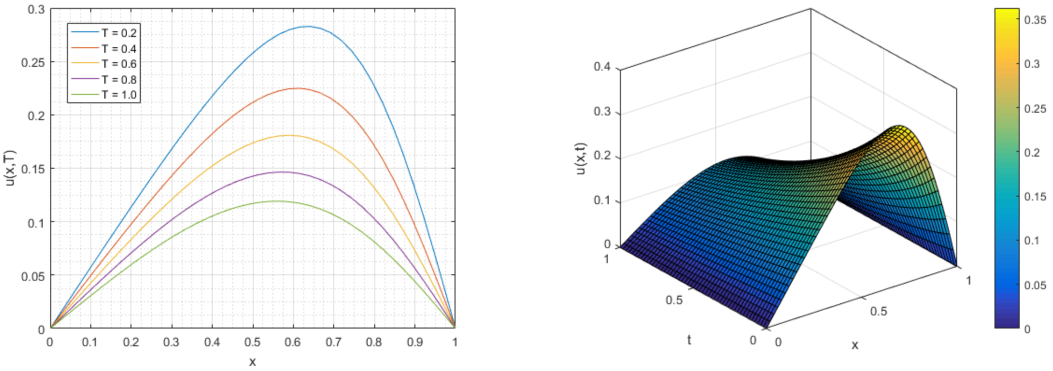
Table 3.6: Comparison of error norms at different viscosity ν of Example 3.3

ν	Mittal [46]		Ganaie [21]		Our Algorithm 1	
	L_∞	L_2	L_∞	L_2	L_∞	L_2
0.5	7.44×10^{-5}	2.79×10^{-5}	2.00×10^{-5}	3.54×10^{-6}	1.2721×10^{-5}	2.1025×10^{-6}
0.2	1.22×10^{-5}	4.57×10^{-6}	3.00×10^{-6}	5.24×10^{-7}	8.2543×10^{-7}	3.9663×10^{-7}
0.1	3.08×10^{-6}	1.15×10^{-6}	2.00×10^{-6}	3.54×10^{-7}	1.0395×10^{-7}	4.9837×10^{-8}

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Table 3.7: Comparison of error norms at different nodes M of Example 3.3

M	Mittal [46]		Rahman [53]		Our Algorithm 1	
	L_∞	L_2	L_∞	L_2	L_∞	L_2
10	1.21×10^{-7}	8.63×10^{-8}	1.24×10^{-7}	8.81×10^{-8}	3.6359×10^{-9}	2.5761×10^{-9}
20	3.06×10^{-8}	2.15×10^{-8}	3.39×10^{-8}	2.40×10^{-8}	3.6387×10^{-9}	2.5760×10^{-9}
40	7.64×10^{-9}	5.37×10^{-9}	1.12×10^{-8}	7.94×10^{-9}	3.6485×10^{-9}	2.5760×10^{-9}
80	1.91×10^{-9}	1.34×10^{-9}	5.54×10^{-9}	3.91×10^{-9}	3.6485×10^{-9}	2.5760×10^{-9}

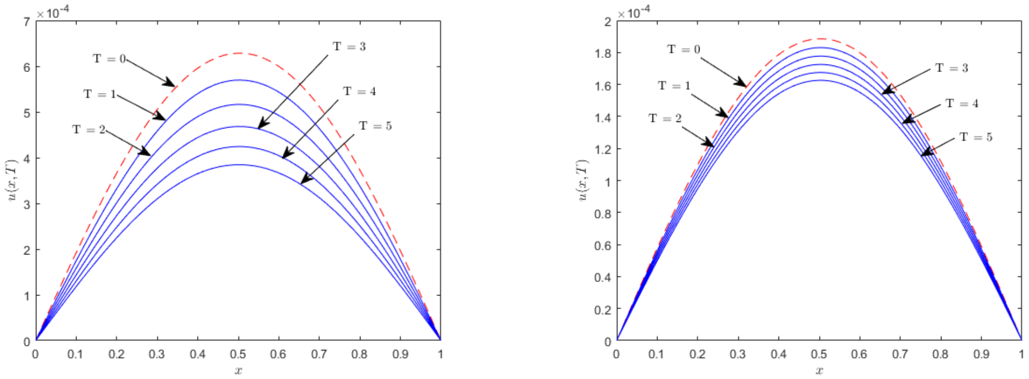


(a) $u(x, T)$ at different times T **(b)** Surface plot of $u(x, t)$

Figure 3.7: Physical behavior of our solution in Example 3.3

Finally, we further focus on the behavior profiles of travelling waves for Example 3.3 by using $M = 100$, $\Delta t = 10^{-2}$ and $\sigma = 100$ when decreasing the small kinematic viscosity values $\nu \in \{10^{-2}, 10^{-3}\}$ at different times $T \in \{1, 2, 3, 4, 5\}$. From Figure 3.8, we can see that our numerical solutions tend to decline as time passes. Moreover, the motion speed of the solutions, that moves away from the initial solution, depends on the viscous values ν. The approximate solutions for $\nu = 10^{-2}$ at the different final times T produce the distribution more than those for $\nu = 10^{-3}$ as presented in Figure 3.8. Hence, if the kinematic viscosity ν of Example 3.3 is very small, the obtained solutions also slowly decrease from the initial solution.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



(a) the viscosity $\nu = 10^{-2}$ **(b)** the viscosity $\nu = 10^{-3}$

Figure 3.8: Profiles of our solution at small viscosity ν in Example 3.3

Example 3.4. Consider the Burgers' equation (3.1) for $t \geq 1$ with initial condition

$$u(x,1) = \frac{x}{1+\exp\left(\frac{4x^2-1}{16\nu}\right)}, \quad x \in [0,1]$$

and the Dirichlet boundary conditions are

$$u(0,t) = 0 \ \text{ and } \ u(1,t) = \left(t + t^{\frac{3}{2}} \exp\left(-\frac{1}{16\nu}\right)\right)^{-1}, \quad t \geq 1.$$

The analytical solution given by Harris [23] is

$$u^*(x,t) = \frac{x}{t + t^{\frac{3}{2}} \exp\left(\frac{x^2}{4\nu t} - \frac{1}{16\nu}\right)}.$$

In this Example 3.4, we examine Algorithm 1 by varying the increasing number of grid points M and the decreasing time step Δt as shown in Table 3.8 for fixing the terminal time $T = 2$ and $\nu = 0.01$. From Table 3.8, we can see that the larger M and the smaller Δt produce the MAEs which tend to decline. Moreover, in our computation of Example 3.4 for the numerous times $T \in \{1.7, 2.4, 3.1\}$ by using Algorithm 1, we choose $M = 100$, $\Delta t = 10^{-5}$ with the viscosity $\nu = 0.005$. The L_∞ and L_2 errors are compared with the numerical solutions obtained by procedures of Ashpazzadeh [5] and Dogan [17] as shown in Table 3.9. From this table, it is clearly seen that our method produces much better solutions than [5] and [17]. The physical behavior of our approximate solution with viscosity $\nu = 0.003$ and time $t \in [1,2]$ is illustrated in Figure 3.9.

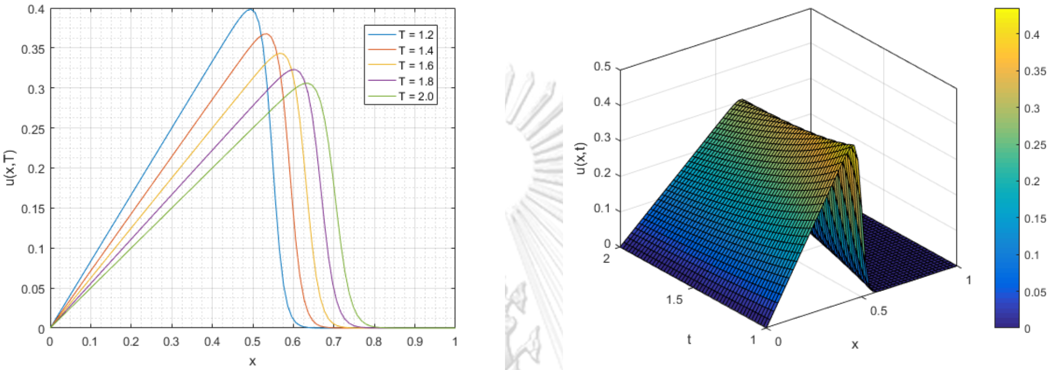
จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Table 3.8: MAE at different M and Δt of Example 3.4

M	$\Delta t = 10^{-1}$	$\Delta t = 10^{-2}$	$\Delta t = 10^{-3}$	$\Delta t = 10^{-4}$	$\Delta t = 10^{-5}$
10	1.2127×10^{-2}	5.2961×10^{-3}	5.3212×10^{-3}	5.3453×10^{-3}	5.3477×10^{-3}
20	5.1856×10^{-3}	7.3940×10^{-4}	2.0987×10^{-4}	1.7183×10^{-4}	1.6960×10^{-4}
30	5.1562×10^{-3}	6.4212×10^{-4}	6.7560×10^{-5}	8.7468×10^{-6}	3.5443×10^{-6}
40	5.1599×10^{-3}	6.4149×10^{-4}	6.5891×10^{-5}	6.6297×10^{-6}	6.9639×10^{-7}
50	5.1609×10^{-3}	6.4155×10^{-4}	6.5876×10^{-5}	6.6053×10^{-6}	6.6233×10^{-7}

Table 3.9: Comparison of error norms at different times T of Example 3.4

T	**Ashpazzadeh [5]**		**Dogan [17]**		**Our Algorithm 1**	
	L_∞	L_2	L_∞	L_2	L_∞	L_2
1.7	2.94×10^{-3}	1.11×10^{-3}	8.09×10^{-3}	2.10×10^{-3}	1.9031×10^{-5}	3.9616×10^{-5}
2.4	2.08×10^{-3}	9.83×10^{-4}	1.16×10^{-2}	3.34×10^{-3}	2.1251×10^{-5}	5.1921×10^{-5}
3.1	4.79×10^{-3}	2.19×10^{-3}	1.58×10^{-2}	4.82×10^{-3}	2.1016×10^{-5}	6.1827×10^{-5}

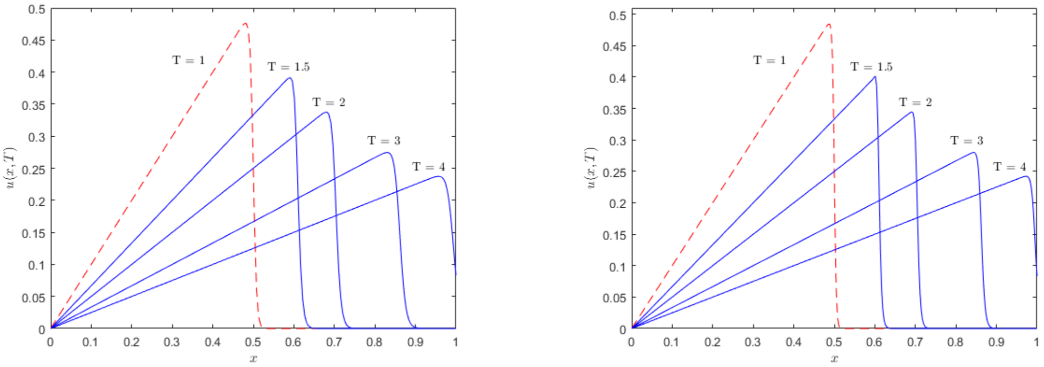


(a) $u(x,T)$ at different times T **(b)** Surface plot of $u(x,t)$

Figure 3.9: Physical behavior of our solution in Example 3.4

Finally, we also show the profiles of travelling waves by using parameters $M = 100$ and $\Delta t = 10^{-2}$ for small viscosity $\nu \in \{10^{-3}, 10^{-4}\}$ at different times T as displayed in Figure 3.10. We can see that Algorithm 1 can handle the problem with small ν.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



(a) the viscosity $\nu = 10^{-3}$ **(b)** the viscosity $\nu = 10^{-4}$

Figure 3.10: Profiles of our solution at small viscosity ν in Example 3.4

Furthermore, we can also show the utility and adaptability of the proposed Algorithm 1 for solving the Burgers' equation with an initial condition that is in terms of a continuous piecewise-defined function.

Example 3.5. Consider the Burgers' equation (3.1) with initial and boundary conditions given by Mittal and Singhal [47] as follows.

$$u(x,0) = \begin{cases} 1 & \text{for} \quad x \in [0,5), \\ 6-x & \text{for} \quad x \in [5,6), \\ 0 & \text{for} \quad x \in [6,12), \end{cases}$$

$u(0,t) = 1$ and $u(12,t) = 0$ for $t \in (0,T]$, respectively. The numerical solution produced by our Algorithm 1 exhibits correct physical behavior for several values of the terminal times T. In Figures 3.11 - 3.13, our obtained numerical solutions are depicted corresponding to $\nu \in \{1, 0.1, 0.01\}$, respectively. We take the parameters $M = 240$ and $\Delta t = 10^{-3}$ at different terminal times $T \in \{0,1,2,3,4\}$. With same parameters, similar patterns have been depicted by Asaithambi [4], Mittal and Jain [46], and Mittal and Singhal [47].

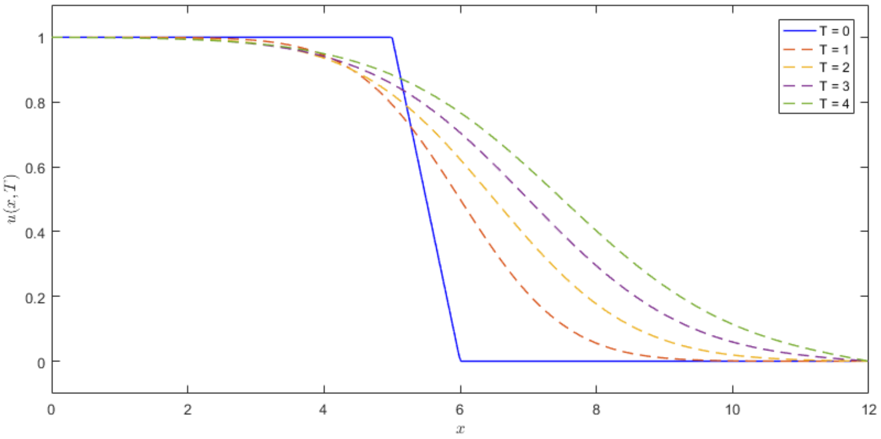


Figure 3.11: Profiles of our numerical solution for $\nu = 1$ in Example 3.5

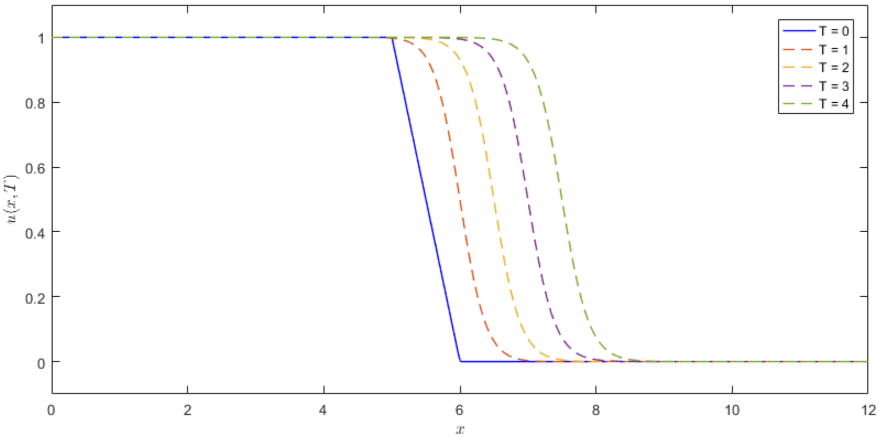
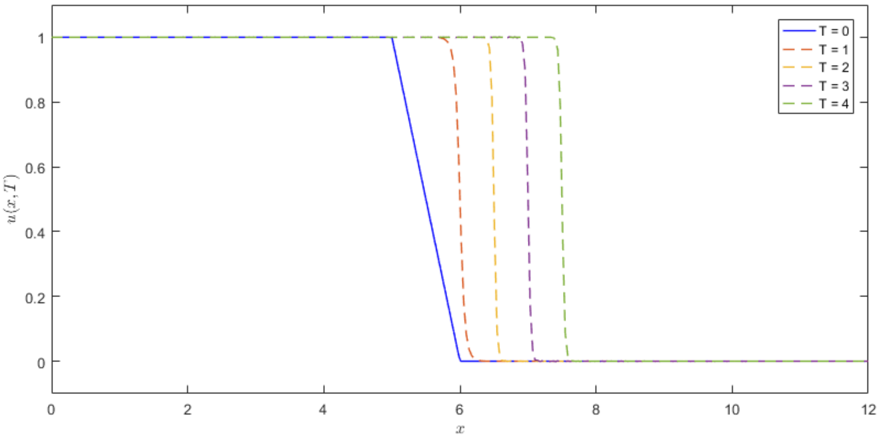


Figure 3.12: Profiles of our numerical solution for $\nu = 0.1$ in Example 3.5



CHULALONGKORN UNIVERSITY

Figure 3.13: Profiles of our numerical solution for $\nu = 0.01$ in Example 3.5

Example 3.6. Consider the Burgers' equation (3.1) with initial and boundary conditions given by Mittal and Singhal [47] as follows.

$$u(x,0) = \begin{cases} \sin(\pi x) & \text{for} \quad x \in [0,1), \\ -\frac{1}{2}\sin(\pi x) & \text{for} \quad x \in [1,2), \\ 0 & \text{for} \quad x \in [2,5), \end{cases}$$

$u(0,t) = 0$ and $u(5,t) = 0$ for $t \in (0,T]$, respectively. The numerical solution produced by our Algorithm 1 exhibits correct physical behavior for several values of the terminal times T. In Figures 3.14 and 3.15, our obtained numerical solutions are depicted corresponding to the kinematic viscosity $\nu \in \{0.1, 0.01\}$, respectively. We choose the following parameters $M = 100$ and $\Delta t = 10^{-3}$ at different terminal times $T \in \{0, 1, 2, \dots, 10\}$. With same parameters, similar figures have been demonstrated by Mittal and Jain [46] with Mittal and Singhal [47].

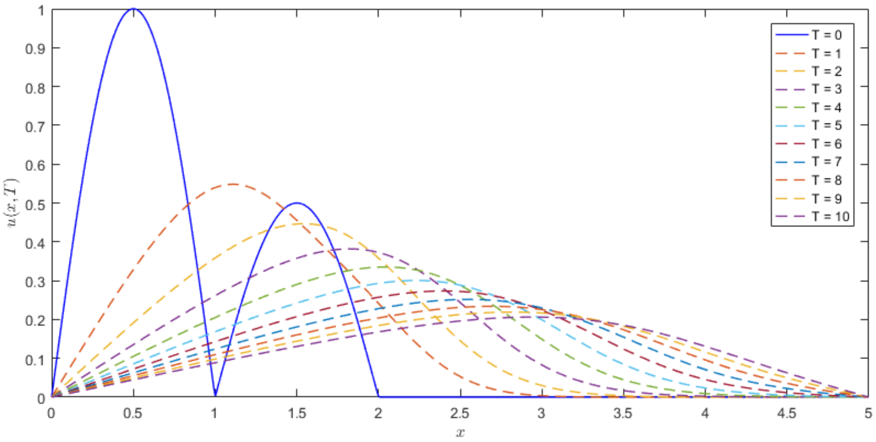


Figure 3.14: Profiles of our numerical solution for $\nu = 0.1$ in Example 3.6

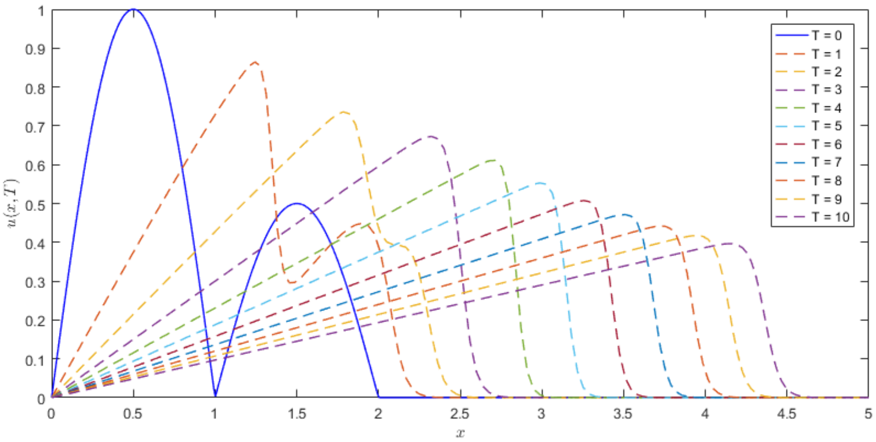


Figure 3.15: Profiles of our numerical solution for $\nu = 0.01$ in Example 3.6

3.4 Acceleration of Algorithm 1

Moreover, we can find the convergence speed of this Algorithm 1 via discretization parameters. Usually, the convergence of Algorithm 1 depends on the parameters of time step Δt and grid spacing which is inversely proportional to the number of grid points M. In this work, we observe that the grid spacing is not uniformly discretizing because of using the zeros of Chebyshev polynomial, which differs from the time step Δt that is equally discretizing. Therefore, we only illustrate the order of convergence for equally discretizing parameter $\Delta t := \tau$ when the number of grid points M is fixed.

In this case, a sequence $\left(\mathbf{u}^{\langle m \rangle}\right)$ is said to converge to the solution $\mathbf{u}^*$ with order p if there exists a constant C such that

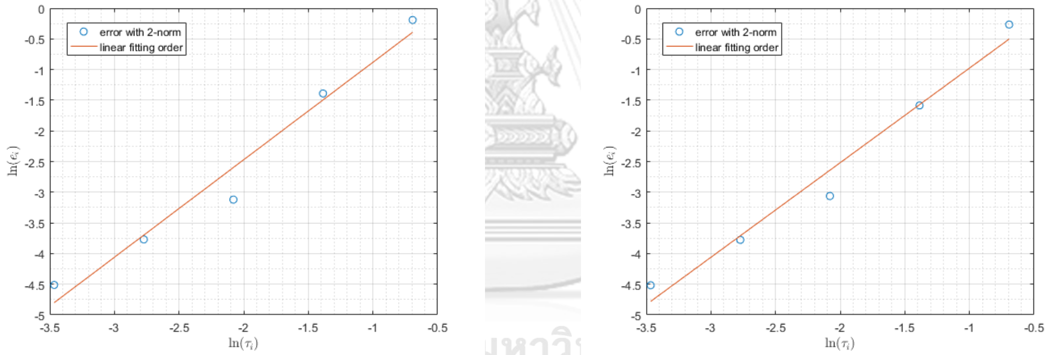
$$\left\| \mathbf{u}^{\langle m \rangle} - \mathbf{u}^* \right\| < C\tau^p.$$

It can be written to another form as $\|\mathbf{u}^{\langle m \rangle} - \mathbf{u}^*\| = \mathcal{O}\left(\tau^p\right)$ using the big O notation. In practical, we take the natural logarithmic function ln on both sides of the above expression in order to approximate the order of convergence p. Thus, we obtain the linear equation $\ln \|\mathbf{u}^{\langle m \rangle} - \mathbf{u}^*\| = p \ln(\tau) + \ln(C)$ with the slope p. However, when we have many time steps τ, we can transform the above linear equation for estimating the order p to the following formula: $p \approx \frac{\ln(e_{\text{new}}/e_{\text{old}})}{\ln(\tau_{\text{new}}/\tau_{\text{old}})}$, where e_{new} and e_{old} denote the errors with respect to the new and old time steps τ_{new} and τ_{old}, respectively.

Hence, we can seek the orders of convergence p of the previous Examples 3.1 - 3.4 by varying the time steps $\tau = 2^{-n}$ for $n \in \{1, 2, 3, 4, 5\}$. These implementations use the following parameters: the number of discretizing nodes $M = 40$, the kinematic viscosity $\nu = 0.01$ at the terminal time $T = 1$ for Examples 3.1 - 3.3 and $T = 2$ for Example 3.4 which is demonstrated in Table 3.10 and Figure 3.16. The convergence orders $p := \lim_{i \to \infty} p_i$ in Table 3.10 are computed based on the Euclidean norm $e_i := \|\mathbf{u}^{\langle m \rangle} - \mathbf{u}^*\|_2$. From Table 3.10, we can see that the convergence orders p_i approach one for all examples which they are really corresponding to the process used, i.e., the forward difference quotient. Usually, the forward difference method produces an algorithm complexity $\mathcal{O}(\tau)$ or linear order of convergence. However, we can accelerate the speed of this Algorithm 1 by using other methods to deal with the temporal variable, instead of the forward difference method, such as the Crank-Nicolson method [20].

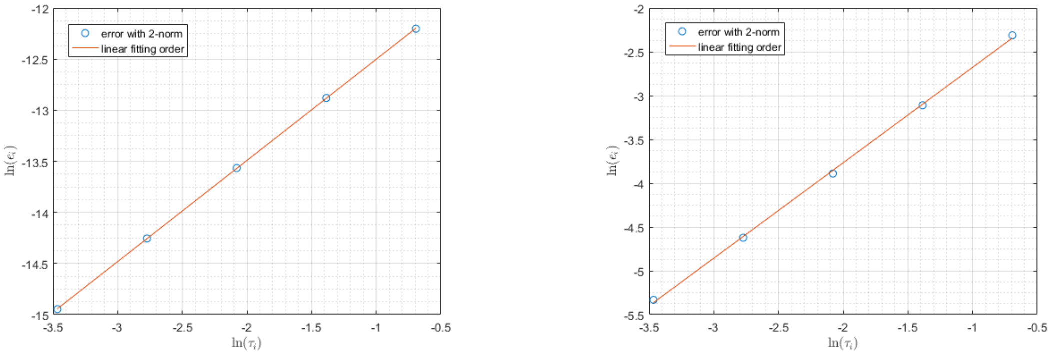
Table 3.10: Convergence orders p using forward difference of Examples 3.1 - 3.4

τ	**Example 3.1**		**Example 3.2**		**Example 3.3**		**Example 3.4**	
	e_i	p_i	e_i	p_i	e_i	p_i	e_i	p_i
2^{-1}	8.255×10^{-1}	-	7.661×10^{-1}	-	5.016×10^{-6}	-	9.908×10^{-2}	-
2^{-2}	2.490×10^{-1}	1.7287	2.049×10^{-1}	1.9025	2.547×10^{-6}	0.97776	4.458×10^{-2}	1.1522
2^{-3}	4.412×10^{-2}	2.4971	4.678×10^{-2}	2.1311	1.283×10^{-6}	0.98871	2.044×10^{-2}	1.1244
2^{-4}	2.303×10^{-2}	0.9374	2.289×10^{-2}	1.0311	6.443×10^{-7}	0.99431	9.832×10^{-3}	1.0564
2^{-5}	1.097×10^{-2}	1.0700	1.092×10^{-2}	1.0675	3.228×10^{-7}	0.99715	4.836×10^{-3}	1.0236



(a) Example 3.1 **(b)** Example 3.2

CHULALONGKORN UNIVERSITY



(c) Example 3.3 **(d)** Example 3.4

Figure 3.16: Graphs of linearly convergence orders p of Examples 3.1 - 3.4

Next, in this section, we accelerate the convergence speed of the proposed Algorithm 1 by using the Crank-Nicolson method [20] which is an average between the forward difference method at iteration $m-1$ and the backward difference method at iteration m. It is well-known that the Crank-Nicolson method always provides the time complexity $\mathcal{O}(\tau^2)$ or quadratic order of convergence. Thus, we start the process by reforming (3.1) that its left-hand-side term remains the time derivative term only and moves others to the right-hand side. Then, we have

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x} := G(x,t,u).$$

After that, we apply the Crank-Nicolson method to the above equation, that is

$$\frac{u^{\langle m\rangle} - u^{\langle m-1\rangle}}{\tau} = \frac{1}{2}\left(G^{\langle m-1\rangle} + G^{\langle m\rangle}\right), \tag{3.13}$$

where τ is the time step, $G^{\langle m-1\rangle} := G(x,t_{m-1},u^{\langle m-1\rangle})$ is the right-hand-side term of the forward difference process at iteration $m-1$ and $G^{\langle m\rangle} := G(x,t_m,u^{\langle m\rangle})$ is the right-hand-side term of the backward difference process at iteration m. We observe that some terms contained in G consist of the nonlinear term $u\frac{\partial u}{\partial x}$ which is manipulated by using the linearization method. In fact, the nonlinear term can be linearized in many ways. For convenience, we linearize the nonlinear term in the same way as in (3.4). Then, we obtain the terms $G^{\langle m-1\rangle}$ and $G^{\langle m\rangle}$ in (3.13) which are

$$\begin{aligned} G^{\langle m-1\rangle} &= \nu\frac{\partial^2 u^{\langle m-1\rangle}}{\partial x^2} - u^{\langle m-1\rangle}\frac{\partial u^{\langle m\rangle}}{\partial x}, \\ G^{\langle m\rangle} &= \nu\frac{\partial^2 u^{\langle m\rangle}}{\partial x^2} - u^{\langle m-1\rangle}\frac{\partial u^{\langle m\rangle}}{\partial x}. \end{aligned}$$

When replacing $G^{\langle m-1\rangle}$ and $G^{\langle m\rangle}$ to (3.13), it becomes

$$\frac{u^{\langle m\rangle}(x) - u^{\langle m-1\rangle}(x)}{\tau} = \frac{\nu}{2}\left(\frac{\partial^2 u^{\langle m-1\rangle}(x)}{\partial x^2} + \frac{\partial^2 u^{\langle m\rangle}(x)}{\partial x^2}\right) - u^{\langle m-1\rangle}(x)\frac{\partial u^{\langle m\rangle}(x)}{\partial x}.$$

Next, we apply our developed FIM-CPE to eliminate derivatives from the above equation by taking twice integrals on both sides. Then, we have

$$\int_a^x \int_a^{\xi_2} \left(\frac{u^{\langle m \rangle}(\xi_1) - u^{\langle m-1 \rangle}(\xi_1)}{\tau} \right) d\xi_1 d\xi_2 = \frac{\nu}{2} \left(u^{\langle m-1 \rangle}(x) + u^{\langle m \rangle}(x) \right) - q(x) + d_1 x + d_2,$$

where $q(x)$ is the twice integrations of the nonlinear term which is performed in the same way as in Section 3.2, d_1 and d_2 are arbitrary constants emerged from the process of integrations. After that, we vary $x \in \{x_1, x_2, x_3, \ldots, x_M\}$ that are generated by the zeros of Chebyshev polynomial $R_M(x)$ to the above equation and rearrange them to construct the matrix form

$$\frac{\mathbf{A}^2}{\tau} \left(\mathbf{u}^{\langle m \rangle} - \mathbf{u}^{\langle m-1 \rangle} \right) = \frac{\nu}{2} \left(\mathbf{u}^{\langle m-1 \rangle} + \mathbf{u}^{\langle m \rangle} \right) - \mathbf{Q}^{\langle m-1 \rangle} \mathbf{u}^{\langle m \rangle} + d_1 \mathbf{x} + d_2 \mathbf{i}.$$

It can be simplified to

$$\left(\frac{\mathbf{A}^2}{\tau} - \frac{\nu \mathbf{I}}{2} + \mathbf{Q}^{\langle m-1 \rangle} \right) \mathbf{u}^{\langle m \rangle} - d_1 \mathbf{x} - d_2 \mathbf{i} = \left(\frac{\mathbf{A}^2}{\tau} + \frac{\nu \mathbf{I}}{2} \right) \mathbf{u}^{\langle m-1 \rangle}, \tag{3.14}$$

where $\mathbf{Q}^{\langle m-1 \rangle} = \mathbf{A}\text{diag}(\mathbf{u}^{\langle m-1 \rangle}) - \mathbf{A}^2\text{diag}(\mathbf{R}'\mathbf{R}^{-1}\mathbf{u}^{\langle m-1 \rangle})$ and the other parameters are defined as in Section 3.2. Finally, we construct the linear system from (3.14) and the boundary conditions (3.9) and (3.10) which is

$$\left[\begin{array}{c|cc} \frac{\mathbf{A}^2}{\tau} - \frac{\nu \mathbf{I}}{2} + \mathbf{Q}^{\langle m-1 \rangle} & -\mathbf{x} & -\mathbf{i} \\ \hline \mathbf{h}_l \mathbf{R}^{-1} & 0 & 0 \\ \mathbf{h}_r \mathbf{R}^{-1} & 0 & 0 \end{array} \right] \left[\begin{array}{c} \mathbf{u}^{\langle m \rangle} \\ \hline d_1 \\ d_2 \end{array} \right] = \left[\begin{array}{c} \left(\frac{\mathbf{A}^2}{\tau} + \frac{\nu \mathbf{I}}{2} \right) \mathbf{u}^{\langle m-1 \rangle} \\ \hline \psi_1(t_m) \\ \psi_2(t_m) \end{array} \right]. \tag{3.15}$$

Therefore, we can solve the linear system (3.15) in order to find an approximate solution $\mathbf{u}^{\langle m \rangle}$ which it produces the accurate solution higher than the previous procedure described by (3.11) under the same parameters.

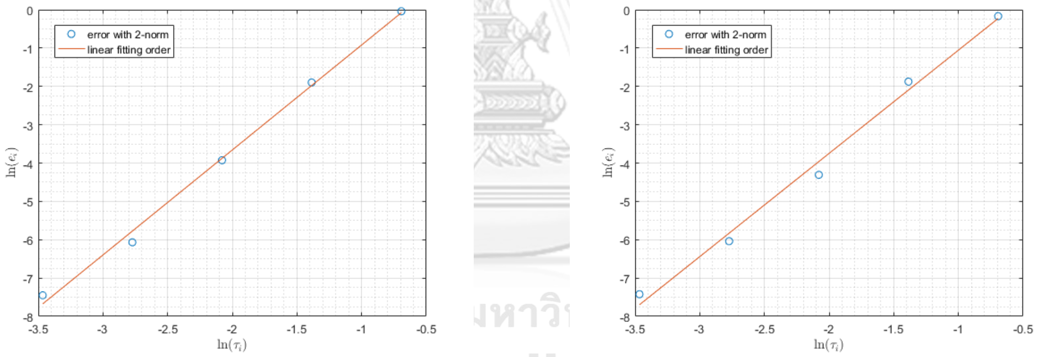
In order to show that the order of convergence p for the accelerated Algorithm 1 is significantly improved, we chose the same parameters as we have used in Table 3.10 to examine this method described by (3.15) which is demonstrated in Table 3.11 and Figure 3.17. In Table 3.11, we find the convergence order $p := \lim\limits_{i \to \infty} p_i$ corresponding to

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

the Euclidean error norm $e_i := \|\mathbf{u}^{\langle m\rangle} - \mathbf{u}^*\|_2$, where $\mathbf{u}^{\langle m\rangle}$ is approximate solution at the terminal time. We can see that the obtained convergence orders p for all Examples 3.1 - 3.4 approach two or $\mathcal{O}(\tau^2)$. It is really coinciding the time complexity of the Crank-Nicolson method that actually gives the quadratically convergence order.

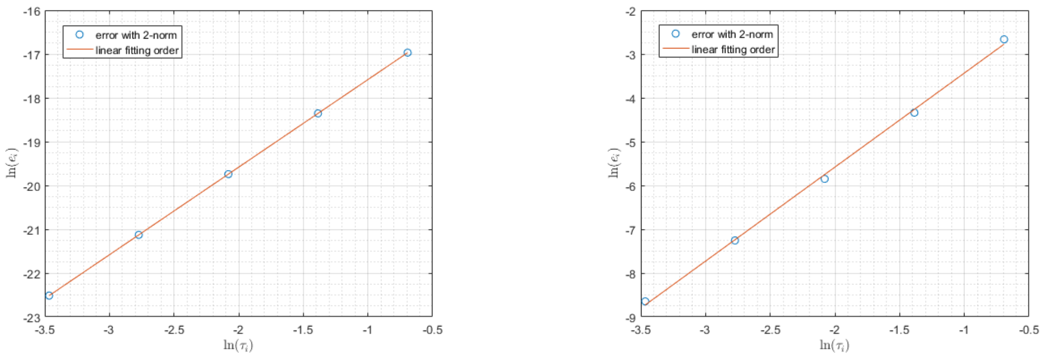
Table 3.11: Convergence orders p using Crank-Nicolson of Examples 3.1 - 3.4

τ	**Example 3.1**		**Example 3.2**		**Example 3.3**		**Example 3.4**	
	e_i	p_i	e_i	p_i	e_i	p_i	e_i	p_i
2^{-1}	9.607×10^{-1}	-	8.453×10^{-1}	-	4.272×10^{-8}	-	6.962×10^{-2}	-
2^{-2}	1.497×10^{-1}	2.6821	1.532×10^{-1}	2.4635	1.067×10^{-8}	2.0004	1.305×10^{-2}	2.4151
2^{-3}	1.971×10^{-2}	2.9249	1.346×10^{-2}	3.5089	2.669×10^{-9}	2.0001	2.878×10^{-3}	2.1812
2^{-4}	2.316×10^{-3}	3.0888	2.383×10^{-3}	2.4978	6.672×10^{-10}	2.0000	7.060×10^{-4}	2.0274
2^{-5}	5.813×10^{-4}	1.9946	2.383×10^{-4}	1.9949	1.668×10^{-10}	2.0000	1.756×10^{-4}	2.0067



(a) Example 3.1 **(b)** Example 3.2

CHULALONGKORN UNIVERSITY



(c) Example 3.3 **(d)** Example 3.4

Figure 3.17: Graphs of quadratically convergence orders p of Examples 3.1 - 3.4

CHAPTER IV

TIME-FRACTIONAL BBMB EQUATION

In this section, we briefly explain the physical meaning of the time-fractional BBMB eqution. We then give some definitions of time-fractional order derivative. Next, we create a numerical algorithm for solving time-fractional BBMB equation (1.2) by extending the idea form Algorithm 1 that uses to solving the Burgers' equation. Finally, we implement the obtained algorithm via several experimental examples in order to demonstrate that our proposed algorithm provides a good accuracy.

4.1 Benjamin-Bona-Mahony-Burgers Equation

The Benjamin-Bona-Mahony-Burgers or BBMB equation was first investigated by Benjamin et al. [9]. It describes the model for propagation of long waves which incorporates nonlinear and dissipative effects. Moreover, it is used in the analysis of the surface waves for long wavelength in liquids, hydromagnetic waves in cold plasma, acoustic-gravity waves in compressible fluids and acoustic waves in harmonic crystals [1]. Many mathematicians paid their attention to the dynamics of the BBMB equation in the form

CHULALONGKORN UNIVERSITY

$$\frac{\partial u}{\partial t} - \frac{\partial^3 u}{\partial x^2 \partial t} + \frac{\partial u}{\partial x} + u\frac{\partial u}{\partial x} = 0,$$

where u represents the fluid velocity, u_t is an acceleration term, u_{xxt} is a dispersive term, u_x is an advection term and uu_x is a nonlinear convection term.

However, in this dissertation, we attempt to adjust the BBMB equation by following Kumar and Kumar [33] in order to increase the attractiveness of this equation and for testing our algorithm that we will create in Section 4.3. This BBMB equation is adjusted by supplementing the external forced function $f(x,t)$, like pressure, on the right-hand side and modifying the acceleration term u_t to be the time-fractional derivative term

$D_t^\alpha u$ instead that accounts simulating the effect of a decrease of the permeability in time delay, see [20]. Therefore, the studied equation in this chapter becomes

$$D_t^\alpha u - \frac{\partial^3 u}{\partial x^2 \partial t} + \frac{\partial u}{\partial x} + u\frac{\partial u}{\partial x} = f(x,t)$$

which called time-fractional BBMB equation. Next, let us clarify the definition of the fractional derivative.

4.2 Time-Fractional Order Derivative

Before embarking into the details for constructing a numerical algorithm via the developed FIM-CPE to overcome the time-fractional BBMB equations in one dimension, we provide the basic definitions of fractional derivatives. The necessary notations and some important facts used throughout this chapter are also given. More details on definitions and basic results of fractional calculus can be found in [51].

Definition 4.1. A real-valued function $u(t)$, $t > 0$ can be defined on the space C_μ, $\mu \in \mathbb{R}$, if there exist a real number $\rho > \mu$ such that $u(t) = t^\rho u_1(t)$, where $u_1(t) \in C[0,\infty)$ and it is defined on the space C_μ^n, if and only if $u^{(n)} \in C_\mu$, $n \in \mathbb{N}$.

Definition 4.2. The Riemann-Liouville fractional integral operator of order $\alpha \geq 0$ of an integrable function $u \in C_\mu$, $\mu > -1$ is defined by

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$I^\alpha u(t) = \begin{cases} \frac{1}{\Gamma(\alpha)} \int_0^t \frac{u(s)}{(t-s)^{1-\alpha}} ds & \text{for} \quad \alpha > 0, \\ u(t) & \text{for} \quad \alpha = 0. \end{cases}$$

Actually, there are several definition for fractional-order derivative. However, here, we give one definition that is used in this chapter.

Definition 4.3. The Caputo fractional derivative D^α of $u \in C_{-1}^n$, $n \in \mathbb{N}$ is defined by

$$D^\alpha u(t) = I^{n-\alpha} D^n u(t) = \begin{cases} \frac{1}{\Gamma(n-\alpha)} \int_0^t \frac{u^{(n)}(s)}{(t-s)^{1-n+\alpha}} ds & \text{for} \quad \alpha \in (n-1, n), \\ u^{(n)}(t) & \text{for} \quad \alpha = n. \end{cases}$$

4.3 Algorithm for Solving Time-Fractional BBMB Equation

Next, we devise the numerical algorithm for seeking approximate solutions of the time-fractional BBMB equation (1.2) in one-dimensional space as considered by Kumar and Kumar [33] in 2014. Let u be an approximate solution of v in (1.2), then our determined BBMB equation can be written in the following form as

$$D_t^{\alpha} u - \frac{\partial^3 u}{\partial x^2 \partial t} + \frac{\partial u}{\partial x} + u\frac{\partial u}{\partial x} = f(x,t), \quad (x,t) \in (0,L) \times (0,T], \tag{4.1}$$

subject to the initial condition:

$$u(x,0) = \phi(x), \quad x \in [0,L], \tag{4.2}$$

and the Dirichlet boundary conditions:

$$\begin{aligned} u(0,t) &= \psi_1(t), \quad t \in (0,T], \\ u(L,t) &= \psi_2(t), \quad t \in (0,T], \end{aligned} \tag{4.3}$$

where L and T are positive real numbers, f, ϕ, ψ_1 and ψ_2 are given sufficiently smooth functions, $\alpha \in (0,1)$ is an order of time fractional derivative term and also u is unknown function of spatial variable x and temporal variable t to be solved. Suppose that u is a smooth real-valued function of the temporal coordinate. Let us first use the technique of linearization to handle with (4.1) by taking the iteration at time $t_m = m\Delta t$, where Δt is a time step and $m \in \mathbb{N}$. Then, we obtain

$$D_t^{\alpha} u(x,t)|_{t=t_m} - \left.\frac{\partial^3 u(x,t)}{\partial x^2 \partial t}\right|_{t=t_m} + \frac{\partial u^{\langle m\rangle}(x)}{\partial x} + u^{\langle m-1\rangle}(x)\frac{\partial u^{\langle m\rangle}(x)}{\partial x} = f(x,t_m), \tag{4.4}$$

where $u^{\langle m\rangle}(x) = u(x,t_m)$ is the numerical solution at m^{th} time iteration. Next, consider the time-fractional derivative term of (4.4) by using the Caputo sense in Definition 4.3, then we have

$$D_t^{\alpha} u(x,t)|_{t=t_m} = \frac{1}{\Gamma(1-\alpha)} \int_0^{t_m} \frac{u_s(x,s)}{(t_m-s)^{\alpha}} ds = \frac{1}{\Gamma(1-\alpha)} \sum_{i=0}^{m-1} \int_{t_i}^{t_{i+1}} \frac{u_s(x,s)}{(t_m-s)^{\alpha}} ds.$$

After that, we use the first-order forward difference quotient to approximate the time derivative term in the above equation. Note that we can employ other methods to approximate the derivative term with respect to time. For convenience, in this work, we choose the forward difference method and also let an index $j := m - i - 1$ in order to use between deriving the following process as

$$\begin{aligned} D_t^{\alpha} u(x,t)\big|_{t=t_m} &\approx \frac{1}{\Gamma(1-\alpha)} \sum_{i=0}^{m-1} \int_{t_i}^{t_{i+1}} (t_m - s)^{-\alpha} \left(\frac{u^{\langle i+1 \rangle}(x) - u^{\langle i \rangle}(x)}{\Delta t} \right) ds \\ &= \frac{1}{\Gamma(1-\alpha)} \sum_{i=0}^{m-1} \left(\frac{u^{\langle i+1 \rangle}(x) - u^{\langle i \rangle}(x)}{\Delta t} \right) \left(\frac{(t_m - t_i)^{1-\alpha} - (t_m - t_{i+1})^{1-\alpha}}{1-\alpha} \right) \\ &= \frac{(\Delta t)^{1-\alpha}}{\Gamma(2-\alpha)} \sum_{i=0}^{m-1} \left(\frac{u^{\langle i+1 \rangle}(x) - u^{\langle i \rangle}(x)}{\Delta t} \right) \left((m-i)^{1-\alpha} - (m-i-1)^{1-\alpha} \right) \\ &= \frac{(\Delta t)^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{m-1} \left(u^{\langle m-j \rangle}(x) - u^{\langle m-j-1 \rangle}(x) \right) \left((j+1)^{1-\alpha} - j^{1-\alpha} \right) \\ &= \sum_{j=0}^{m-1} w_j \left(u^{\langle m-j \rangle}(x) - u^{\langle m-j-1 \rangle}(x) \right), \end{aligned} \tag{4.5}$$

where $w_j = \frac{(\Delta t)^{-\alpha}}{\Gamma(2-\alpha)} \left((j+1)^{1-\alpha} - j^{1-\alpha} \right)$. Now, we consider the third-order derivative term with respect to twice-spatial and single-temporal variables in (4.4). We estimate the time derivative of this term by hiring the same method as mentioned in Burgers' equation, i.e., the first-order forward difference quotient. Then, we have

$$\left. \frac{\partial^3 u(x,t)}{\partial x^2 \partial t} \right|_{t=t_m} = \left. \frac{\partial^2}{\partial x^2} \left(\frac{\partial u(x,t)}{\partial t} \right) \right|_{t=t_m} \approx \frac{\partial^2}{\partial x^2} \left(\frac{u^{\langle m \rangle}(x) - u^{\langle m-1 \rangle}(x)}{\Delta t} \right). \tag{4.6}$$

Hence, we can replace (4.5) and (4.6) into (4.4) to obtain

$$\begin{aligned} & w_0 \left(u^{\langle m \rangle}(x) - u^{\langle m-1 \rangle}(x) \right) + \sum_{j=1}^{m-1} w_j \left(u^{\langle m-j \rangle}(x) - u^{\langle m-j-1 \rangle}(x) \right) \\ & - \frac{1}{\Delta t} \left(\frac{\partial^2 u^{\langle m \rangle}(x)}{\partial x^2} - \frac{\partial^2 u^{\langle m-1 \rangle}(x)}{\partial x^2} \right) + \frac{\partial u^{\langle m \rangle}(x)}{\partial x} + u^{\langle m-1 \rangle}(x) \frac{\partial u^{\langle m \rangle}(x)}{\partial x} = f(x, t_m). \end{aligned}$$

Next, to eliminate all derivatives concerning the spatial variable out of the above equation, we consume the developed FIM-CPE in Section 2.4.1 by taking the twice-layer integrals from 0 to the zero x_k of the Chebyshev polynomial $R_M(x)$ that is generated by (2.3).

Then, we get the following equation

$$\begin{aligned}
&w_0 \int_0^{x_k} \int_0^{\xi_2} \Big(u^{\langle m \rangle}(\xi_1) - u^{\langle m-1 \rangle}(\xi_1) \Big)\, d\xi_1 d\xi_2 \\
&+ \sum_{j=1}^{m-1} w_j \int_0^{x_k} \int_0^{\xi_2} \Big(u^{\langle m-j \rangle}(\xi_1) - u^{\langle m-j-1 \rangle}(\xi_1) \Big)\, d\xi_1 d\xi_2 \\
&+ \frac{1}{\Delta t}\Big(u^{\langle m-1 \rangle}(x_k) - u^{\langle m \rangle}(x_k) \Big) + \int_0^{x_k} u^{\langle m \rangle}(\xi_2)\, d\xi_2 + d_1 x_k + d_2 \\
&+ \int_0^{x_k} \int_0^{\xi_2} \Big(u^{\langle m-1 \rangle}(\xi_1) \frac{\partial u^{\langle m \rangle}(\xi_1)}{\partial \xi_1} \Big)\, d\xi_1 d\xi_2 = \int_0^{x_k} \int_0^{\xi_2} f(\xi_1, t_m)\, d\xi_1 d\xi_2, \qquad (4.7)
\end{aligned}$$

where d_1 and d_2 are arbitrary constants emerged in the process of integration. After that, we consider the nonlinear term in (4.7) which is denoted by $q(x_k)$. Anywise, we have used to consider this nonlinear term $q(x_k)$ for the Burgers' equation in previous chapter. Thus, this nonlinear term $q(x_k)$ is approximated by (3.6), that is,

$$\mathbf{q} = \mathbf{A}\mathrm{diag}\big(\mathbf{u}^{\langle m-1 \rangle}\big)\mathbf{u}^{\langle m \rangle} - \mathbf{A}^2\mathrm{diag}\big(\mathbf{R}'\mathbf{R}^{-1}\mathbf{u}^{\langle m-1 \rangle}\big)\mathbf{u}^{\langle m \rangle}, \qquad (4.8)$$

where $\mathbf{q} = [q(x_1), q(x_2), q(x_3), \dots, q(x_M)]^\top$, $\mathbf{u}^{\langle z \rangle} = \big[u^{\langle z \rangle}(x_1), u^{\langle z \rangle}(x_2), \dots, u^{\langle z \rangle}(x_M)\big]^\top$ for $z \in \mathbb{N} \cup \{0\}$, $\mathbf{A} = \bar{\mathbf{R}}\mathbf{R}^{-1}$ is the $M \times M$ one-dimensional Chebyshev integration matrix explained in Section 2.4.1 and

$$\mathbf{R}' = \begin{bmatrix} R'_0(x_1) & R'_1(x_1) & \cdots & R'_{M-1}(x_1) \\ R'_0(x_2) & R'_1(x_2) & \cdots & R'_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ R'_0(x_M) & R'_1(x_M) & \cdots & R'_{M-1}(x_M) \end{bmatrix}.$$

Consequently, by varying the zero $x_k \in \{x_1, x_2, x_3, \dots, x_M\}$ in (4.7) and consuming (4.8), we can transform the integral equation (4.7) to the matrix form by using our developed FIM-CPE as follows

$$\begin{gathered}
w_0\mathbf{A}^2\big(\mathbf{u}^{\langle m \rangle} - \mathbf{u}^{\langle m-1 \rangle}\big) + \sum_{j=1}^{m-1} w_j \mathbf{A}^2\big(\mathbf{u}^{\langle m-j \rangle} - \mathbf{u}^{\langle m-j-1 \rangle}\big) + \frac{1}{\Delta t}\mathbf{u}^{\langle m-1 \rangle} - \frac{1}{\Delta t}\mathbf{u}^{\langle m \rangle} \\
+ \mathbf{A}\mathbf{u}^{\langle m \rangle} + d_1\mathbf{x} + d_2\mathbf{i} + \mathbf{A}\mathrm{diag}\big(\mathbf{u}^{\langle m-1 \rangle}\big)\mathbf{u}^{\langle m \rangle} - \mathbf{A}^2\mathrm{diag}\big(\mathbf{R}'\mathbf{R}^{-1}\mathbf{u}^{\langle m-1 \rangle}\big)\mathbf{u}^{\langle m \rangle} = \mathbf{A}^2\mathbf{f}^{\langle m \rangle}
\end{gathered}$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

or it can be simplified as

$$\begin{aligned}\Big(w_0\mathbf{A}^2 - \frac{1}{\Delta t}\mathbf{I} + \mathbf{A} + \mathbf{A}\mathrm{diag}\big(\mathbf{u}^{\langle m-1\rangle}\big) - \mathbf{A}^2\mathrm{diag}\big(\mathbf{R}'\mathbf{R}^{-1}\mathbf{u}^{\langle m-1\rangle}\big)\Big)\mathbf{u}^{\langle m\rangle} + d_1\mathbf{x} + d_2\mathbf{i}& \\ = \mathbf{A}^2\mathbf{f}^{\langle m\rangle} - \sum_{j=1}^{m-1} w_j\mathbf{A}^2\big(\mathbf{u}^{\langle m-j\rangle} - \mathbf{u}^{\langle m-j-1\rangle}\big) + \Big(w_0\mathbf{A}^2 - \frac{1}{\Delta t}\mathbf{I}\Big)\mathbf{u}^{\langle m-1\rangle},& \qquad (4.9)\end{aligned}$$

where $\mathbf{I}$ is an $M \times M$ identity matrix, $\mathbf{x} = [x_1, x_2, x_3, \dots, x_M]^\top$, $\mathbf{i} = [1, 1, 1, \dots, 1]^\top$, and $\mathbf{f}^{\langle m\rangle} = [f(x_1, t_m), f(x_2, t_m), f(x_3, t_m), \dots, f(x_M, t_m)]^\top$. From the given Dirichlet boundary conditions (4.3), we can convert them into vector forms by using the linear combination of Chebyshev polynomial (2.11) at the m^{th} iteration as follow:

$$u^{\langle m\rangle}(0) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle}R_n(0) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle}(-1)^n := \mathbf{h}_l\mathbf{c}^{\langle m\rangle} = \mathbf{h}_l\mathbf{R}^{-1}\mathbf{u}^{\langle m\rangle} = \psi_1(t_m), \qquad (4.10)$$

$$u^{\langle m\rangle}(L) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle}R_n(L) = \sum_{n=0}^{M-1} c_n^{\langle m\rangle}(1)^n := \mathbf{h}_r\mathbf{c}^{\langle m\rangle} = \mathbf{h}_r\mathbf{R}^{-1}\mathbf{u}^{\langle m\rangle} = \psi_2(t_m), \qquad (4.11)$$

where $t_m = m\Delta t$ for $m \in \mathbb{N}$, $\mathbf{h}_l = [1, -1, 1, \dots, (-1)^{M-1}]$ and $\mathbf{h}_r = [1, 1, 1, \dots, 1]$.

Finally, from (4.9), (4.10) and (4.11), 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 $\mathbf{u}^{\langle m\rangle}$, d_1 and d_2, as follows:

$$\left[\begin{array}{c|cc} \mathbf{K}^{\langle m-1\rangle} & \mathbf{x} & \mathbf{i} \\ \hline \mathbf{h}_l\mathbf{R}^{-1} & 0 & 0 \\ \mathbf{h}_r\mathbf{R}^{-1} & 0 & 0 \end{array}\right]\left[\begin{array}{c} \mathbf{u}^{\langle m\rangle} \\ \hline d_1 \\ d_2 \end{array}\right] = \left[\begin{array}{c} \mathbf{A}^2\mathbf{f}^{\langle m\rangle} - \mathbf{s}^{\langle m-1\rangle} + \left(w_0\mathbf{A}^2 - \frac{1}{\Delta t}\mathbf{I}\right)\mathbf{u}^{\langle m-1\rangle} \\ \hline \psi_1(t_m) \\ \psi_2(t_m) \end{array}\right], \qquad (4.12)$$

where $\mathbf{K}^{\langle m-1\rangle} := w_0\mathbf{A}^2 - \frac{1}{\Delta t}\mathbf{I} + \mathbf{A} + \mathbf{A}\mathrm{diag}\big(\mathbf{u}^{\langle m-1\rangle}\big) - \mathbf{A}^2\mathrm{diag}\big(\mathbf{R}'\mathbf{R}^{-1}\mathbf{u}^{\langle m-1\rangle}\big)$ is the coefficient matrix of $\mathbf{u}^{\langle m\rangle}$ and $\mathbf{s}^{\langle m-1\rangle} := \sum_{j=1}^{m-1} w_j\mathbf{A}^2\big(\mathbf{u}^{\langle m-j\rangle} - \mathbf{u}^{\langle m-j-1\rangle}\big)$ for $m \in \mathbb{N}$. Anywise, we remark that for $\mathbf{s}^{\langle 0\rangle} = \mathbf{0}$, because the initial step of this summation is already separated during the process of transforming to the integral equation. Therefore, the solution $\mathbf{u}^{\langle m\rangle}$ can be found by solving the system of linear equation (4.12) with starting by $\mathbf{u}^{\langle 0\rangle} = [\phi(x_1), \phi(x_2), \phi(x_3), \dots, \phi(x_M)]^\top$. We notice here that at the terminal time T,

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

the numerical solution $u(x,T)$ for each arbitrary $x \in (0,L)$ can be computed by

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

where $\mathbf{R}(x) = [R_0(x), R_1(x), R_2(x), \dots, R_{M-1}(x)]$, $\mathbf{R}^{-1}$ is the $M \times M$ matrix defined by (2.8) and $\mathbf{u}^{\langle m \rangle}$ is the final iteration of (4.12).

For computational convenience, we provide the following algorithm in the form of pseudocode for finding an approximate solution of the one-dimensional time-fractional BBMB equation by using our developed FIM-CPE.

Algorithm 2 Algorithm for solving time-fractional BBMB equation by the FIM-CPE

```
Input: $\alpha$, $x$, $L$, $T$, $M$, $\Delta t$, $\phi(x)$, $\psi_1(t)$, $\psi_2(t)$ and $f(x,t)$;
Output: An approximate solution $u(x,T)$;
Set $x_k = \frac{L}{2}\left[\cos\left(\frac{2k-1}{2M}\pi\right)+1\right]$ for $k \in \{1,2,3,\dots,M\}$ in descending order;
Compute $\mathbf{x}$, $\mathbf{i}$, $\mathbf{h}_l$, $\mathbf{h}_r$, $\mathbf{I}$, $\mathbf{A}$, $\mathbf{R}'$, $\overline{\mathbf{R}}$, $\mathbf{R}^{-1}$, $\mathbf{R}(x)$ and $w_0$;
Construct $\mathbf{u}^{\langle 0 \rangle} = [\phi(x_1), \phi(x_2), \phi(x_3), \dots, \phi(x_M)]^\top$;
Set $m = 1$ and $t_1 = \Delta t$;
while $t_m \le T$ do
    Set $\mathbf{s}^{\langle m-1 \rangle} = \mathbf{0}$;
    for $j = 1$ to $m-1$ do
        Compute $w_j = \frac{(\Delta t)^{-\alpha}}{\Gamma(2-\alpha)}\left((j+1)^{1-\alpha} - j^{1-\alpha}\right)$;
        Compute $\mathbf{s}^{\langle m-1 \rangle} = \mathbf{s}^{\langle m-1 \rangle} + w_j\mathbf{A}^2\left(\mathbf{u}^{\langle m-j \rangle} - \mathbf{u}^{\langle m-j-1 \rangle}\right)$;
    end for
    Compute $\mathbf{K}^{\langle m-1 \rangle} = w_0\mathbf{A}^2 - \frac{1}{\Delta t}\mathbf{I} + \mathbf{A} + \mathbf{A}\mathrm{diag}\left(\mathbf{u}^{\langle m-1 \rangle}\right) - \mathbf{A}^2\mathrm{diag}\left(\mathbf{R}'\mathbf{R}^{-1}\mathbf{u}^{\langle m-1 \rangle}\right)$;
    Compute $f^{\langle m \rangle} = [f(x_1,t_m), f(x_2,t_m), f(x_3,t_m), \dots, f(x_M,t_m)]^\top$;
    Find $\mathbf{u}^{\langle m \rangle}$ by solving the iterative linear system (4.12);
    Update $m = m + 1$;
    Compute $t_m = m\Delta t$;
end while
return $u(x,T) = \mathbf{R}(x)\mathbf{R}^{-1}\mathbf{u}^{\langle m \rangle}$;
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

4.4 Numerical Examples for Testing Algorithm 2

In this section, we implement the proposed Algorithm 2 based on our developed FIM-CPE for solving the time-fractional BBMB equations in order to show the efficiency and effectiveness of our scheme through several numerical examples which are measured the accurate results by the MAE $= \frac{1}{M}\sum_{i=1}^{M}|u^*(x_i,t)-u(x_i,t)|$, where u^* and u are the analytical and numerical solutions, respectively.

Example 4.1. Consider the time-fractional BBMB equation (4.1) with the source term

$$f(x,t)=\frac{3\sqrt{\pi}x^4(x-1)t^{\frac{3}{2}-\alpha}}{4\Gamma(\frac{5}{2}-\alpha)}+x^2t^{\frac{1}{2}}\left(5x^7t^{\frac{5}{2}}-9x^6t^{\frac{5}{2}}+4x^5t^{\frac{5}{2}}+5x^2t-4xt-30x+18\right)$$

subject to the initial condition:

$$u(x,0)=0,\quad x\in[0,1], \tag{4.13}$$

and the Dirichlet boundary conditions:

$$u(0,t)=0 \text{ and } u(1,t)=0,\ \ t\in(0,1]. \tag{4.14}$$

The analytical solution given by Shen and Zhu [57] is $u^*(x,t)=x^4(x-1)t^{\frac{3}{2}}$. In the numerical testing, we compare the approximate results obtained by our Algorithm 2 with a Crank-Nicolson linear difference scheme (CNLDS) proposed by Shen and Zhu [57] in 2018 which measured by the MAE for $\alpha=0.5$ at various M and Δt as shown in Table 4.1. We can see that our presented Algorithm 2 gives higher accuracy than the CNLDS under the same parameters and conditions. Moreover, we also illustrate the MAE at the final time $T=1$ for $\alpha=0.5$, $M=40$ and the different time steps $\Delta t\in\{0.05,0.01,0.005,0.001\}$ in Table 4.2 and for $\Delta t=0.001$, $M=40$ and the various fractional orders of derivative $\alpha\in\{0.1,0.3,0.7,0.9\}$ in Table 4.3. Finally, the graph of our numerical solutions $u(x,T)$ at the different terminal times T and the surface plot of our numerical solutions $u(x,t)$ are provided in Figure 4.1 for $\alpha=0.5$, $M=40$ and $\Delta t=0.001$.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Table 4.1: MAE at various M and Δt for $\alpha = 0.5$ of Example 4.1

M	Our Algorithm 2			CNLDS [57]
	$\Delta t = 0.01$	$\Delta t = 0.005$	$\Delta t = 0.001$	$\Delta t = 0.001$
10	1.7347×10^{-4}	8.7727×10^{-5}	1.7810×10^{-5}	1.8414×10^{-3}
20	1.7301×10^{-4}	8.7495×10^{-5}	1.7763×10^{-5}	5.0479×10^{-4}
40	1.7289×10^{-4}	8.7436×10^{-5}	1.7751×10^{-5}	1.3219×10^{-4}
80	1.7286×10^{-4}	8.7421×10^{-5}	1.7748×10^{-5}	3.3657×10^{-5}

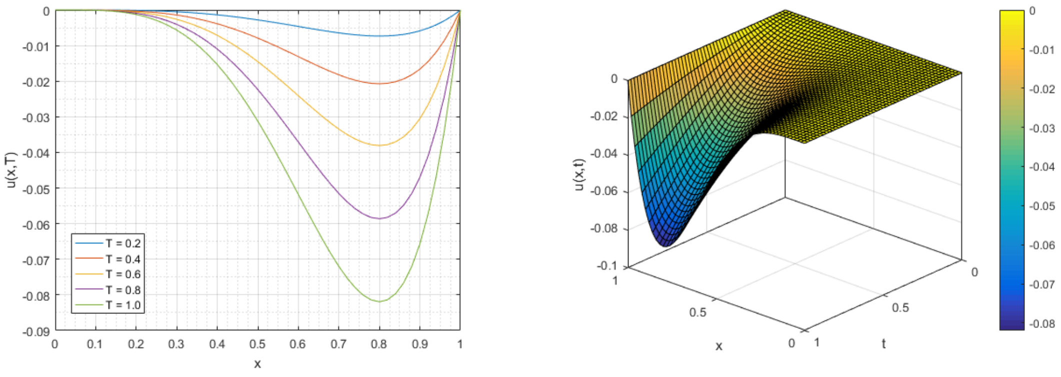
Table 4.2: MAE at various Δt for $\alpha = 0.5$ and $M = 40$ and of Example 4.1

x	$\Delta t = 0.05$	$\Delta t = 0.01$	$\Delta t = 0.005$	$\Delta t = 0.001$
0.2	1.3072×10^{-3}	2.7980×10^{-5}	1.4266×10^{-5}	2.9342×10^{-6}
0.4	1.8644×10^{-3}	3.8426×10^{-5}	1.9242×10^{-5}	3.8423×10^{-6}
0.6	1.3842×10^{-3}	2.9035×10^{-4}	1.4669×10^{-4}	2.9731×10^{-5}
0.8	2.5887×10^{-3}	5.4398×10^{-4}	2.7512×10^{-4}	5.5861×10^{-5}

Table 4.3: MAE at various α for $\Delta t = 0.001$ and $M = 40$ of Example 4.1

x	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.7$	$\alpha = 0.9$
0.2	2.8624×10^{-6}	2.9140×10^{-6}	2.8699×10^{-6}	2.4680×10^{-6}
0.4	4.0394×10^{-6}	3.9104×10^{-6}	3.9486×10^{-6}	4.7687×10^{-6}
0.6	3.0072×10^{-5}	2.9863×10^{-5}	2.9838×10^{-5}	3.0944×10^{-5}
0.8	5.6196×10^{-5}	5.5998×10^{-5}	5.5923×10^{-5}	5.6838×10^{-5}

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



(a) The solution at different times T **(b)** The surface plot of solution

Figure 4.1: Graphical behavior of our solution in Example 4.1

Example 4.2. Consider the time-fractional BBMB equation (4.1) with the source term

$$f(x,t) = \frac{2e^x t^{2-\alpha}}{\Gamma(3-\alpha)} + te^x \left(t^3 e^x + t - 2\right)$$

subject to the initial condition (4.13) and the Dirichlet boundary conditions:

$$u(0,t) = t^2 \text{ and } u(1,t) = et^2, \quad t \in (0,1].$$

The analytical solution given by Esen and Tasbozan [18] is $u^*(x,t) = t^2 e^x$. For the numerical examination using Algorithm 2, we choose parameters $\alpha = 0.5$ and $\Delta t = 0.01$ to show the MAEs at different nodes M in Table 4.4. We also display the MAEs at many time steps Δt for $\alpha = 0.5$ and $M = 40$ in Table 4.5. We can see that for increasing of nodal point M, it almost does not affect to the accuracy which obviously contracts the decreasing of time step Δt. Further, we vary the fractional orders of derivative α at $\Delta t = 0.001$ and $M = 40$ to show the MAE in Table 4.6. Figure 4.2 provides our graphical solutions of this problem including the plotting solutions at different times T and the surface plot under the parameters $\alpha = 0.5$, $M = 40$ and $\Delta t = 0.001$.

Table 4.4: MAE at various M for $\alpha = 0.5$ and $\Delta t = 0.01$ of Example 4.2

x	$M = 5$	$M = 10$	$M = 15$	$M = 20$
0.2	7.9553×10^{-5}	6.4013×10^{-5}	6.4013×10^{-5}	6.4013×10^{-5}
0.4	6.3990×10^{-5}	6.1063×10^{-5}	6.1063×10^{-5}	6.1063×10^{-5}
0.6	3.4929×10^{-5}	1.1182×10^{-5}	1.1182×10^{-5}	1.1182×10^{-5}
0.8	4.4970×10^{-5}	3.9815×10^{-5}	3.9815×10^{-5}	3.9815×10^{-5}

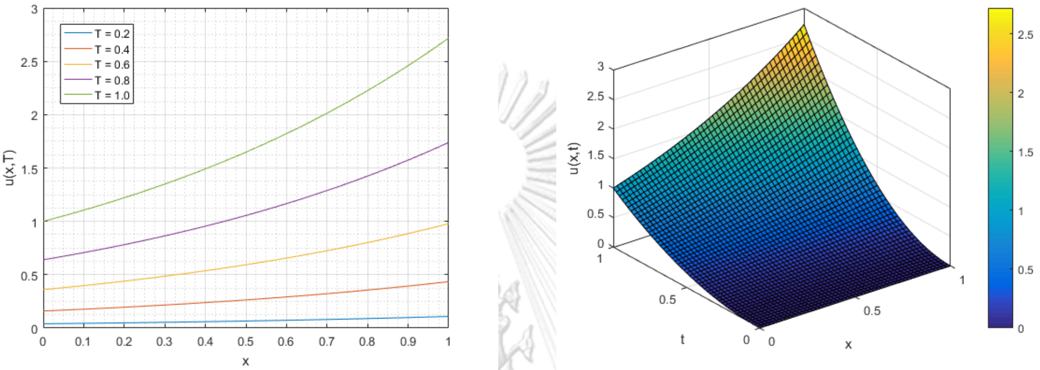
Table 4.5: MAE at various Δt for $\alpha = 0.5$ and $M = 40$ of Example 4.2

x	$\Delta t = 0.05$	$\Delta t = 0.01$	$\Delta t = 0.005$	$\Delta t = 0.001$
0.2	1.4907×10^{-4}	6.4013×10^{-5}	4.1055×10^{-5}	1.0379×10^{-5}
0.4	5.1386×10^{-4}	6.1063×10^{-5}	4.6138×10^{-5}	1.2941×10^{-5}
0.6	9.1238×10^{-4}	1.1182×10^{-5}	2.3799×10^{-5}	9.0593×10^{-6}
0.8	9.7423×10^{-4}	3.9815×10^{-5}	5.5279×10^{-6}	2.2631×10^{-6}

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Table 4.6: MAE at various α for $\Delta t = 0.001$ and $M = 40$ of Example 4.2

x	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.7$	$\alpha = 0.9$
0.2	1.1518×10^{-5}	1.1292×10^{-5}	4.5130×10^{-6}	2.8758×10^{-5}
0.4	1.4991×10^{-5}	1.4551×10^{-5}	2.9661×10^{-6}	5.3208×10^{-5}
0.6	1.1558×10^{-5}	1.1002×10^{-5}	2.4827×10^{-6}	6.6751×10^{-5}
0.8	4.3173×10^{-6}	3.8567×10^{-6}	6.8200×10^{-6}	5.6685×10^{-5}



(a) The solution at different times T **(b)** The surface plot of solution

Figure 4.2: Graphical behavior of our solution in Example 4.2

Example 4.3. Consider the time-fractional BBMB equation (4.1) with the source term

$$f(x,t) = \frac{t^{1-\alpha}\sin(\pi x)}{\Gamma(2-\alpha)} + \pi^2 \sin(\pi x) + \pi t \sin(\pi x) + \frac{\pi t^2}{2} \sin(2\pi x)$$

subject to the same initial condition (4.13) and the Dirichlet boundary conditions (4.14). The analytical solution given by Zarebnia and Parvaz [71] is $u^*(x,t) = t\sin(\pi x)$. In the numerical testing of this problem with Algorithm 2, we consider the MAE of this problem by selecting the same parameters as Example 4.2 which varies along the nodal grid numbers M, the time steps Δt and the fractional orders of derivative α as demonstrated in Tables 4.7, 4.8 and 4.9, respectively. We can observe that the obtained solutions have produced a consequences similar to Examples 4.2. Finally, we depict the plotting of numerical solutions $u(x,T)$ at the different terminal times T and the surface plot of our solution $u(x,t)$ in Figure 4.3 for selecting $\alpha = 0.5$, $M = 40$ and $\Delta t = 0.001$.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Table 4.7: MAE at various M for $\alpha = 0.5$ and $\Delta t = 0.01$ of Example 4.3

x	$M = 5$	$M = 10$	$M = 15$	$M = 20$
0.2	7.9553×10^{-5}	6.4013×10^{-5}	6.4013×10^{-5}	6.4013×10^{-5}
0.4	6.3990×10^{-5}	6.1063×10^{-5}	6.1063×10^{-5}	6.1063×10^{-5}
0.6	3.4929×10^{-5}	1.1182×10^{-5}	1.1182×10^{-5}	1.1182×10^{-5}
0.8	4.4970×10^{-5}	3.9815×10^{-5}	3.9815×10^{-5}	3.9815×10^{-5}

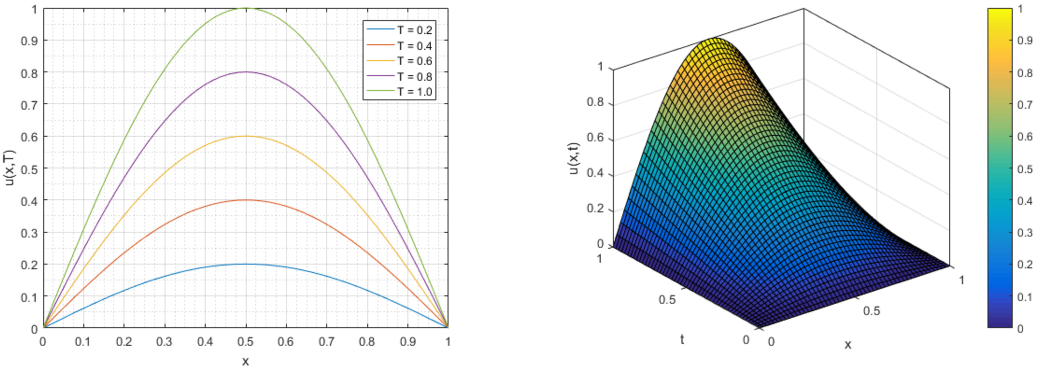
Table 4.8: MAE at various Δt for $\alpha = 0.5$ and $M = 40$ of Example 4.3

x	$\Delta t = 0.05$	$\Delta t = 0.01$	$\Delta t = 0.005$	$\Delta t = 0.001$
0.2	1.0218×10^{-3}	1.9609×10^{-4}	9.7531×10^{-5}	1.9424×10^{-5}
0.4	7.5199×10^{-4}	1.4293×10^{-4}	7.1000×10^{-5}	1.4126×10^{-5}
0.6	4.4809×10^{-4}	8.8329×10^{-5}	4.4082×10^{-5}	8.8029×10^{-6}
0.8	9.2119×10^{-4}	1.7832×10^{-4}	8.8787×10^{-5}	1.7698×10^{-5}

Table 4.9: MAE at various α for $\Delta t = 0.001$ and $M = 40$ of Example 4.3

x	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.7$	$\alpha = 0.9$
0.2	1.9561×10^{-5}	1.9499×10^{-5}	1.9335×10^{-5}	1.9232×10^{-5}
0.4	1.4256×10^{-5}	1.4198×10^{-5}	1.4039×10^{-5}	1.3936×10^{-5}
0.6	8.7989×10^{-6}	8.7996×10^{-6}	8.8102×10^{-6}	8.8226×10^{-6}
0.8	1.7765×10^{-5}	1.7734×10^{-5}	1.7658×10^{-5}	1.7615×10^{-5}

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



(a) The solution at different times T **(b)** The surface plot of solution

Figure 4.3: Graphical behavior of our solution in Example 4.3

Furthermore, we can find the convergence speed of the proposed Algorithm 2 corresponding to the time step $\Delta t := \tau$. We observe that this time-fractional BBMB problem includes two derivative terms with respect to time that are $D_t^{\alpha} u$ and u_{xxt}. When we approximate these terms by using the forward difference method, it produces the time complexities $\mathcal{O}(\tau^{2-\alpha})$ and $\mathcal{O}(\tau)$, respectively. Since $\alpha \in (0,1)$, the combination of those terms provide the larger complexity, i.e., $\mathcal{O}(\tau)$. Accordingly, our presented Algorithm 2 converges with linear order to the exact solution without a fractional order α which is displayed via all previous Examples 4.1 - 4.3.

Next, we show the order of convergence $p := \lim_{i \to \infty} p_i$ based on the Euclidean error norm $e_i := \|\mathbf{u}^{\langle m \rangle} - \mathbf{u}^*\|_2$, where $\mathbf{u}^{\langle m \rangle}$ is a numerical solution at the terminal time and $\mathbf{u}^*$ is the exact solution, in Tables 4.10 - 4.12 and Figures 4.4 - 4.6. In each table, we vary the time steps $\tau = 2^{-n}$ for $n \in \{1, 2, 3, 4, 5\}$ and the fractional order $\alpha \in \{0.0001, 0.5, 0.9999\}$ by using the number of grid points $M = 20$ at the terminal time $T = 1$. From all Tables 4.10 - 4.12, no matter the fractional orders α approach the zero side, one side or it is in between the zero and one sides, we can see that the obtained convergence orders p corresponding to the time steps τ are indeed linearly order or $\mathcal{O}(\tau)$, which they are independent of the fractional order α for all three Examples 4.1 - 4.3.

Additionally, if we would like to accelerate the convergence speed of Algorithm 2, then we can modify it in the same way as in Section 3.4 which we can use the Crack-Nicolson method [20] instead of the forward difference method. Then, the obtained Algorithm 2 certainly provides the time complexity $\mathcal{O}(\tau^2)$ or quadratically order of convergence. However, it is easy to construct the accelerated Algorithm 2 with the Crack-Nicolson method by directly imitating the created process as shown in Section 3.4. Thus, in this chapter, we omit the construction of procedure for solving the time-fractional BBMB equation based on the Crank-Nicolson method.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Table 4.10: Convergence orders p at each fractional order α of Example 4.1

$\boldsymbol{\tau}$	$\boldsymbol{\alpha = 0.0001}$		$\boldsymbol{\alpha = 0.5}$		$\boldsymbol{\alpha = 0.9999}$	
	$\boldsymbol{e_i}$	$\boldsymbol{p_i}$	$\boldsymbol{e_i}$	$\boldsymbol{p_i}$	$\boldsymbol{e_i}$	$\boldsymbol{p_i}$
2^{-1}	4.4674×10^{-2}	-	4.5057×10^{-2}	-	4.6876×10^{-2}	-
2^{-2}	2.4644×10^{-2}	0.8581	2.4718×10^{-2}	0.8662	2.5763×10^{-2}	0.8635
2^{-3}	1.3176×10^{-2}	0.9034	1.3165×10^{-2}	0.9088	1.3748×10^{-2}	0.9060
2^{-4}	6.8964×10^{-3}	0.9339	6.8731×10^{-3}	0.9376	7.1904×10^{-3}	0.9350
2^{-5}	3.5585×10^{-3}	0.9545	3.5401×10^{-3}	0.9571	3.7093×10^{-3}	0.9549

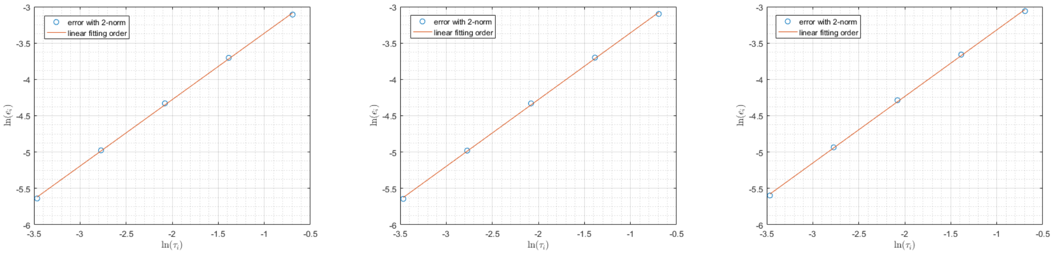
Table 4.11: Convergence orders p at each fractional order α of Example 4.2

$\boldsymbol{\tau}$	$\boldsymbol{\alpha = 0.0001}$		$\boldsymbol{\alpha = 0.5}$		$\boldsymbol{\alpha = 0.9999}$	
	$\boldsymbol{e_i}$	$\boldsymbol{p_i}$	$\boldsymbol{e_i}$	$\boldsymbol{p_i}$	$\boldsymbol{e_i}$	$\boldsymbol{p_i}$
2^{-1}	5.6501×10^{-1}	-	6.2905×10^{-1}	-	8.9250×10^{-1}	-
2^{-2}	2.5668×10^{-1}	1.1383	2.8080×10^{-1}	1.1637	4.6797×10^{-1}	0.9314
2^{-3}	1.1476×10^{-1}	1.1614	1.2184×10^{-1}	1.2045	1.6922×10^{-1}	1.4675
2^{-4}	5.3472×10^{-2}	1.1017	5.5856×10^{-2}	1.1252	7.2280×10^{-2}	1.2272
2^{-5}	2.5794×10^{-2}	1.0517	2.6690×10^{-2}	1.0654	3.3603×10^{-2}	1.1050

Table 4.12: Convergence orders p at each fractional order α of Example 4.3

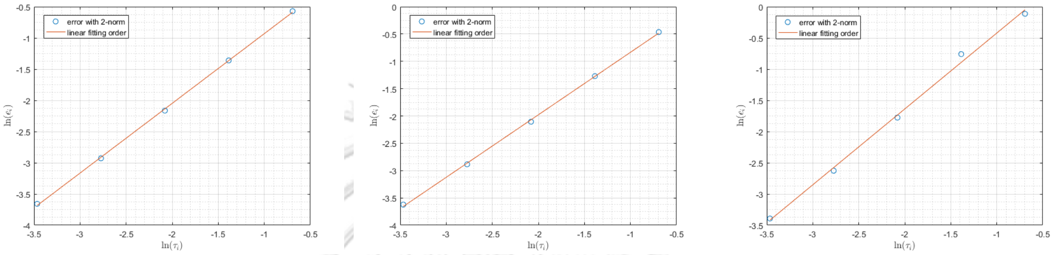
$\boldsymbol{\tau}$	$\boldsymbol{\alpha = 0.0001}$		$\boldsymbol{\alpha = 0.5}$		$\boldsymbol{\alpha = 0.9999}$	
	$\boldsymbol{e_i}$	$\boldsymbol{p_i}$	$\boldsymbol{e_i}$	$\boldsymbol{p_i}$	$\boldsymbol{e_i}$	$\boldsymbol{p_i}$
2^{-1}	4.1168×10^{-2}	-	4.0900×10^{-2}	-	4.0797×10^{-2}	-
2^{-2}	1.7207×10^{-2}	1.2586	1.7083×10^{-2}	1.2596	1.6980×10^{-2}	1.2646
2^{-3}	7.7528×10^{-3}	1.1502	7.6966×10^{-3}	1.1502	7.6357×10^{-3}	1.1530
2^{-4}	3.6632×10^{-3}	1.0816	3.6370×10^{-3}	1.0815	3.6044×10^{-3}	1.0830
2^{-5}	1.7783×10^{-3}	1.0426	1.7657×10^{-3}	1.0425	1.7489×10^{-3}	1.0433

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



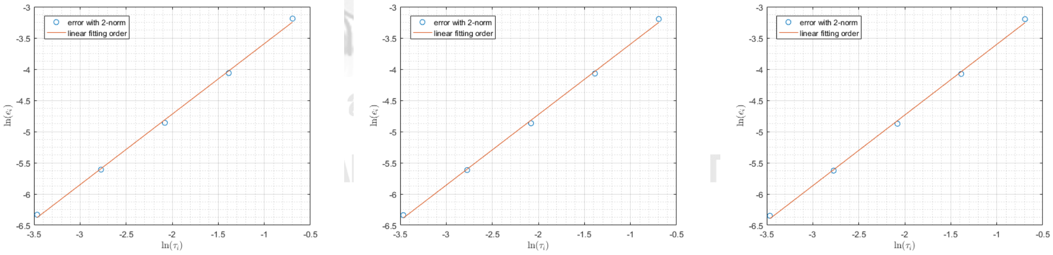
(a) $\alpha = 0.0001$ **(b)** $\alpha = 0.5$ **(c)** $\alpha = 0.9999$

Figure 4.4: Graphical convergence order p of Example 4.1



(a) $\alpha = 0.0001$ **(b)** $\alpha = 0.5$ **(c)** $\alpha = 0.9999$

Figure 4.5: Graphical convergence order p of Example 4.2



(a) $\alpha = 0.0001$ **(b)** $\alpha = 0.5$ **(c)** $\alpha = 0.9999$

Figure 4.6: Graphical convergence order p of Example 4.3

CHAPTER V

NONLINEAR POISSON EQUATION

In this chapter, we briefly account some physical meaning of Poisson-type equation. Then, we construct a numerical algorithm for solving two-dimensional nonlinear Poisson equation (1.3) over irregular domains by using the developed FIM-CPE in two-dimensional spatial coordinates. Next, we test accuracy and efficiency of the obtained algorithm via several examples for different irregular regions.

5.1 Poisson-Type Equation

Poisson equation is an elliptic PDE with broad utility in mechanical engineering and theoretical physics. For instance, it arises to describe the potential field caused by a given charge or mass density distribution. Then, one can compute the electrostatic or gravitational field. The Poisson equation is a generalization of the Laplace's equation, which is also frequently seen in physics and it is written as

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$\nabla^2 u = f,$$

where ∇^2 is the Laplace operator, f is given a real-valued function and u is an unknown function to be sought. This Poisson equation has many applications, for examples;

- **Newtonian gravity** [52]: In the case of a gravitational field $\mathbf{g}$ due to an attracting massive object of density ρ, the Gauss's law for gravity in differential form can be used to obtain the corresponding equation $\nabla \cdot \mathbf{g} = -4\pi G\rho$, where G is the gravitational constant. Since the gravitational field $\mathbf{g}$ is conservative, it can be expressed in terms of a scalar potential ϕ, i.e., $\mathbf{g} = -\nabla\phi$. After substituting it into the Gauss's law, we yield the Poisson equation for gravity as $\nabla^2\phi = 4\pi G\rho$. Solving this Poisson equation for the potential ϕ requires knowing the density ρ.

- **Electrostatics** [26]: Starting with the Gauss's law for electricity in differential form, one of the Maxwell's equations, that is $\nabla \cdot \mathbf{D} = \rho$, where $\mathbf{D}$ is an electric displacement field and ρ is charge density that brought from outside. Assuming that the medium is linear, isotropic and homogeneous, we have the constitutive equation $\mathbf{D} = \varepsilon \mathbf{E}$, where ε is a permittivity of the medium and $\mathbf{E}$ is an electric field. After substituting this into the Gauss's law, we have $\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon}$. In electrostatic, we assume that there is no magnetic field. Then, we obtain $\nabla \times \mathbf{E} = \mathbf{0}$. When the curl of any gradient is zero, we can write the electric field $\mathbf{E}$ as the gradient of a scalar function φ called the electric potential, i.e., $\mathbf{E} = -\nabla \varphi$. Substituting the potential gradient for the electric field, then this directly produces Poisson equation for electrostatics, which is $\nabla^2 \varphi = -\frac{\rho}{\varepsilon}$. This Poisson equation can be solved to seek the electric potential φ when the charge density ρ and the permittivity ε are known.

In addition, there are many other applications regarding the Poisson-type equation, see [56] and references therein. However, in this dissertation, we study the generalized two-dimensional nonlinear Poisson equation, that is, the lower-order derivatives u_x, u_y and u are added on the left-hand side and the forcing term f is also considered to be a nonlinear function in terms of u. Then, it is expressed as

$$\nabla^2 u + \alpha \frac{\partial u}{\partial x} + \beta \frac{\partial u}{\partial y} + \gamma u = f(u),$$

CHULALONGKORN UNIVERSITY

where α, β and γ are given coefficient functions depending on the variables x and y, respectively. In order to increase the attractiveness of this nonlinear Poisson equation, we further consider it over an irregular two-dimensional domain. For the irregular domain used in this work, it means any arbitrarily shaped regions in two-dimensional domains which are the connected region without a hole with a closed boundary curve, except the rectangular-shaped domain. The variety of irregular domains that we choose to study in this work including the pentagonal, circular, L-shaped, butterfly, peanut-shaped and elliptic regions as demonstrated the figures in Section 5.3. Let us devise a numerical algorithm for estimating an approximate solution of the nonlinear Poisson-type equation over irregular two-dimensional domains in the next section.

5.2 Algorithm for Solving Nonlinear Poisson Equation

In this section, we propose the numerical procedure based on our two-dimensional developed FIM-CPE for finding approximate solutions of the two-dimensional nonlinear Poisson-type equation (1.3) over an irregular domain. Now, we let u be an approximate result of v in (1.3). Then, we consider the following steady state nonlinear Poisson-type equation in two dimensions on irregular domain Ω

$$\nabla^2 u + \alpha(x,y)\frac{\partial u}{\partial x} + \beta(x,y)\frac{\partial u}{\partial y} + \gamma(x,y)u = f(x,y,u), \quad (x,y) \in \Omega \tag{5.1}$$

subject to the Dirichlet boundary condition

$$u(x,y) = \psi(x,y), \quad (x,y) \in \partial\Omega, \tag{5.2}$$

where $\nabla^2 := \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplace operator, α, β, γ and ψ are given the smooth functions depending on the variables x and y, respectively. The function f is a nonlinear in term of u, that u is an unknown function in terms of x and y to be sought over the irregular domain $\Omega \subseteq \mathbb{R}^2$, where Ω is a connected region with closed boundary curve $\partial\Omega$. Before we continue, let $a, b, c, d \in \mathbb{R}$ such that $a < b$ and $c < d$. Assume that $[a,b] \times [c,d]$ is the smallest rectangular region that covers the domain Ω along the horizontal and vertical directions, respectively.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

First, the technique of linearization is used to deal with the nonlinear term of (5.1) in order to find the solution iteratively. Then, we obtain

$$\frac{\partial^2 u^{\langle m\rangle}}{\partial x^2} + \frac{\partial^2 u^{\langle m\rangle}}{\partial y^2} + \alpha\frac{\partial u^{\langle m\rangle}}{\partial x} + \beta\frac{\partial u^{\langle m\rangle}}{\partial y} + \gamma u^{\langle m\rangle} = f(x,y,u^{\langle m-1\rangle}), \tag{5.3}$$

where $u^{\langle m\rangle}$ is a numerical value in the m^{th} iterations for $m \in \mathbb{N}$. Next, applying the developed FIM-CPE to eliminate all spatial derivatives from (5.3) by taking twice integrals with respect to the variables x and y, respectively. Then, the differential equation (5.3) is transformed into the equivalent integral equation and utilizing the technique of integration by parts, it is the following form as

$$\begin{aligned}
&\int_c^y \int_c^{\eta_2} u^{\langle m \rangle}(x,\eta_1)\, d\eta_1 d\eta_2 + \int_a^x \int_a^{\xi_2} u^{\langle m \rangle}(\xi_1, y)\, d\xi_1 d\xi_2 \\
&+ \int_c^y \int_c^{\eta_2} \int_a^x \Big(\alpha(\xi_2,\eta_1) u^{\langle m \rangle}(\xi_2,\eta_1) - \int_a^{\xi_2} \frac{\partial \alpha}{\partial \xi_1}(\xi_1,\eta_1) u^{\langle m \rangle}(\xi_1,\eta_1)\, d\xi_1 \Big) d\xi_2 d\eta_1 d\eta_2 \\
&+ \int_c^y \int_a^x \int_a^{\xi_2} \Big(\beta(\xi_1,\eta_2) u^{\langle m \rangle}(\xi_1,\eta_2) - \int_c^{\eta_2} \frac{\partial \beta}{\partial \eta_1}(\xi_1,\eta_1) u^{\langle m \rangle}(\xi_1,\eta_1)\, d\eta_1 \Big) d\xi_1 d\xi_2 d\eta_2 \\
&+ \int_c^y \int_c^{\eta_2} \int_a^x \int_a^{\xi_2} \gamma(\xi_1,\eta_1) u^{\langle m \rangle}(\xi_1,\eta_1)\, d\xi_1 d\xi_2 d\eta_1 d\eta_2 + x b_1(y) + b_2(y) + y d_1(x) + d_2(x) \\
&= \int_c^y \int_c^{\eta_2} \int_a^x \int_a^{\xi_2} f(\xi_1,\eta_1, u^{\langle m-1 \rangle}(\xi_1,\eta_1))\, d\xi_1 d\xi_2 d\eta_1 d\eta_2, \qquad (5.4)
\end{aligned}$$

where $b_1(y)$, $b_2(y)$, $d_1(x)$ and $d_2(x)$ are arbitrary functions emerged in the process of integration. To handle with these unknown functions, the Chebyshev interpolation is used to approximate them by

$$b_r(y) = \sum_{n=0}^{N-1} b_{r,n} R_n(y) \quad \text{and} \quad d_r(x) = \sum_{n=0}^{M-1} d_{r,n} R_n(x) \qquad (5.5)$$

for $r \in \{1, 2\}$, where $\{b_{r,n}\}_{n=0}^{N-1}$ and $\{d_{r,n}\}_{n=0}^{M-1}$ are unknown values on these interpolated points which will be determined according to the given boundary condition (5.2). Then, we discretize both horizontal and vertical directions of the rectangular domain $[a,b] \times [c,d]$ into M and N points, respectively, through the zeros of Chebyshev polynomials $R_M(x)$ and $R_N(y)$, which are defined by $\mathrm{X} = \{x_1, x_2, x_3, \ldots, x_M\}$ and $\mathrm{Y} = \{y_1, y_2, y_3, \ldots, y_N\}$, respectively. Therefore, the total number of grid points in global numbering system is $H := MN$ nodes. We note that each node in the system obtains from an element in the set of Cartesian product $\mathrm{X} \times \mathrm{Y}$ ordering as the global-type system, i.e., $(x_i, y_i) \in \mathrm{X} \times \mathrm{Y}$ for $i \in \{1, 2, 3, \ldots, H\}$. Next, we substitute each node (x_i, y_i) to (5.4) and transform it into the matrix form by using the idea of developed FIM-CPE in two-dimensional spaces described in Section 2.4.2. Then, we obtain

$$\begin{aligned}
&\mathbf{A}_y^2 \mathbf{u}^{\langle m \rangle} + \mathbf{A}_x^2 \mathbf{u}^{\langle m \rangle} + \mathbf{A}_y^2 \mathbf{A}_x \big(\boldsymbol{\alpha} \mathbf{u}^{\langle m \rangle} - \mathbf{A}_x \boldsymbol{\alpha}_x \mathbf{u}^{\langle m \rangle}\big) + \mathbf{A}_y \mathbf{A}_x^2 \big(\boldsymbol{\beta} \mathbf{u}^{\langle m \rangle} - \mathbf{A}_y \boldsymbol{\beta}_y \mathbf{u}^{\langle m \rangle}\big) \\
&+ \mathbf{A}_y^2 \mathbf{A}_x^2 \boldsymbol{\gamma} \mathbf{u}^{\langle m \rangle} + \mathbf{X} \boldsymbol{\Phi}_y \mathbf{b}_1 + \boldsymbol{\Phi}_y \mathbf{b}_2 + \mathbf{Y} \boldsymbol{\Phi}_x \mathbf{d}_1 + \boldsymbol{\Phi}_x \mathbf{d}_2 = \mathbf{A}_y^2 \mathbf{A}_x^2 \mathbf{f}^{\langle m-1 \rangle}
\end{aligned}$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

or it can be simplified as

$$\mathbf{K}\mathbf{u}^{\langle m\rangle} + \mathbf{X}\boldsymbol{\Phi}_y\mathbf{b}_1 + \boldsymbol{\Phi}_y\mathbf{b}_2 + \mathbf{Y}\boldsymbol{\Phi}_x\mathbf{d}_1 + \boldsymbol{\Phi}_x\mathbf{d}_2 = \mathbf{A}_y^2\mathbf{A}_x^2\mathbf{f}^{\langle m-1\rangle}, \tag{5.6}$$

where $\mathbf{K} := \mathbf{A}_y^2 + \mathbf{A}_x^2 + \mathbf{A}_y^2\mathbf{A}_x\boldsymbol{\alpha} - \mathbf{A}_y^2\mathbf{A}_x^2\boldsymbol{\alpha}_x + \mathbf{A}_y\mathbf{A}_x^2\boldsymbol{\beta} - \mathbf{A}_y^2\mathbf{A}_x^2\boldsymbol{\beta}_y + \mathbf{A}_y^2\mathbf{A}_x^2\boldsymbol{\gamma}$, $\mathbf{A}_x$ and $\mathbf{A}_y$ are defined in Remark 2.1. Other parameters contained in (5.6) and $\mathbf{K}$ are defined by

$$\begin{array}{lcll}
\mathbf{X} & = & \operatorname{diag}(x_1, x_2, x_3, \ldots, x_H), & \\
\mathbf{Y} & = & \operatorname{diag}(y_1, y_2, y_3, \ldots, y_H), & \\
\boldsymbol{\alpha} & = & \operatorname{diag}(\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_H) & \text{for } \alpha_i = \alpha(x_i, y_i), \\
\boldsymbol{\beta} & = & \operatorname{diag}(\beta_1, \beta_2, \beta_3, \ldots, \beta_H) & \text{for } \beta_i = \beta(x_i, y_i), \\
\boldsymbol{\gamma} & = & \operatorname{diag}(\gamma_1, \gamma_2, \gamma_3, \ldots, \gamma_H) & \text{for } \gamma_i = \gamma(x_i, y_i), \\
\boldsymbol{\alpha}_x & = & \operatorname{diag}(\alpha_{x,1}, \alpha_{x,2}, \alpha_{x,3}, \ldots, \alpha_{x,H}) & \text{for } \alpha_{x,i} = \alpha_x(x_i, y_i), \\
\boldsymbol{\beta}_y & = & \operatorname{diag}(\beta_{y,1}, \beta_{y,2}, \beta_{y,3}, \ldots, \beta_{y,H}) & \text{for } \beta_{y,i} = \beta_y(x_i, y_i), \\
\mathbf{b}_r & = & \left[b_{r,0}, b_{r,1}, b_{r,2}, \ldots, b_{r,N-1}\right]^\top & \text{for } r \in \{1, 2\}, \\
\mathbf{d}_r & = & \left[d_{r,0}, d_{r,1}, d_{r,2}, \ldots, d_{r,M-1}\right]^\top & \text{for } r \in \{1, 2\}, \\
\mathbf{u}^{\langle m\rangle} & = & \left[u_1^{\langle m\rangle}, u_2^{\langle m\rangle}, u_3^{\langle m\rangle}, \ldots, u_H^{\langle m\rangle}\right]^\top & \text{for } u_i^{\langle\cdot\rangle} = u^{\langle\cdot\rangle}(x_i, y_i), \\
\mathbf{f}^{\langle m-1\rangle} & = & \left[f_1^{\langle m-1\rangle}, f_2^{\langle m-1\rangle}, f_3^{\langle m-1\rangle}, \ldots, f_H^{\langle m-1\rangle}\right]^\top & \text{for } f_i^{\langle\cdot\rangle} = f(x_i, y_i, u_i^{\langle\cdot\rangle}).
\end{array}$$

From (5.5), we can obtain $\boldsymbol{\Phi}_x$ and $\boldsymbol{\Phi}_y$, where

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

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

and

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

Next, for the given Dirichlet boundary condition (5.2), it can be transformed into matrix equations by hiring the linear combination of Chebyshev polynomials (2.11) which can be split into four cases as follows:

Left & Right boundary conditions: For y being fixed, we consider

$$u^{\langle m\rangle}(x,y)=\sum_{n=0}^{M-1}c_n^{\langle m\rangle}R_n(x):=\mathbf{z}_M(x)\mathbf{R}_M^{-1}\mathbf{u}^{\langle m\rangle}(\cdot,y)=\psi(x,y) \tag{5.7}$$

where $\mathbf{z}_M(x)=[R_0(x),R_1(x),R_2(x),\dots,R_{M-1}(x)]$. As we vary the fixed variable $y\in\{y_1,y_2,y_3,\dots,y_N\}$ which is meshed by the zeros of $R_N(y)$. For $h\in\{1,2,3,\dots,N\}$, we consider the nodal points (x_h^l,y_h) and (x_h^r,y_h) which are the corresponding left and right vertical projections along all points of the zeros y_h, respectively, on the boundary region Ω, see Figures 5.1(a) and 5.1(b). Then, we can express (5.7) to the left ($\mu=l$) or right ($\mu=r$) boundary condition in the matrix form by

$$\begin{bmatrix}\mathbf{z}_M(x_1^\mu)\mathbf{R}_M^{-1} & 0 & \cdots & 0\\ 0 & \mathbf{z}_M(x_2^\mu)\mathbf{R}_M^{-1} & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0\\ 0 & \cdots & 0 & \mathbf{z}_M(x_N^\mu)\mathbf{R}_M^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{u}^{\langle m\rangle}(\cdot,y_1)\\ \mathbf{u}^{\langle m\rangle}(\cdot,y_2)\\ \vdots\\ \mathbf{u}^{\langle m\rangle}(\cdot,y_N)\end{bmatrix}=\begin{bmatrix}\psi(x_1^\mu,y_1)\\ \psi(x_2^\mu,y_2)\\ \vdots\\ \psi(x_N^\mu,y_N)\end{bmatrix},$$

which denoted by $\mathbf{Z}_\mu\mathbf{u}^{\langle m\rangle}=\mathbf{\Psi}_\mu$, where $\mathbf{Z}_\mu=\operatorname{diag}\big(\mathbf{z}_M(x_1^\mu),\mathbf{z}_M(x_2^\mu),\dots,\mathbf{z}_M(x_N^\mu)\big)\otimes\mathbf{R}_M^{-1}$ for $\mu\in\{l,r\}$. Then, to be specific, we denote the above boundary condition which corresponds to the left- and right-hand sides of the rectangular domain $[a,b]\times[c,d]$ by $\mathbf{Z}_l\mathbf{u}^{\langle m\rangle}=\mathbf{\Psi}_l$ and $\mathbf{Z}_r\mathbf{u}^{\langle m\rangle}=\mathbf{\Psi}_r$, respectively.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

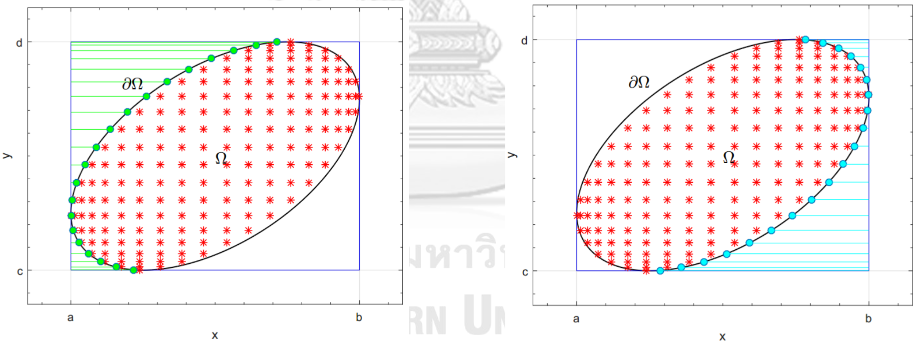
Bottom & Top boundary conditions: For x being fixed, we consider

$$u^{\langle m\rangle}(x,y)=\sum_{n=0}^{N-1}c_n^{\langle m\rangle}R_n(y):=\mathbf{z}_N(y)\mathbf{R}_N^{-1}\mathbf{u}^{\langle m\rangle}(x,\cdot)=\psi(x,y), \tag{5.8}$$

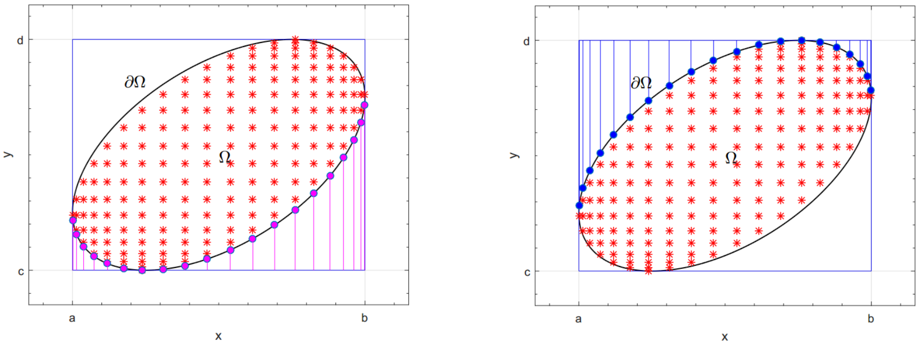
where $\mathbf{z}_N(y)=[R_0(y),R_1(y),R_2(y),\dots,R_{N-1}(y)]$. As we vary the fixed variable $x\in\{x_1,x_2,x_3,\dots,x_M\}$ which is generated by the zeros of $R_M(x)$. For $k\in\{1,2,3,\dots,M\}$, we can consider the nodal points (x_k, y_k^b) and (x_k, y_k^t) which are the corresponding bottom and top horizontal projections along all zeros x_k, respectively, on the boundary region Ω, see Figures 5.1(c) and 5.1(d). Then, (5.8) can be expressed to the bottom $(\nu = b)$ or top $(\nu = t)$ boundary condition in the matrix form by

$$\begin{bmatrix} \mathbf{z}_N(y_1^\nu)\mathbf{R}_N^{-1} & 0 & \cdots & 0 \\ 0 & \mathbf{z}_N(y_2^\nu)\mathbf{R}_N^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{z}_N(y_M^\nu)\mathbf{R}_N^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{\langle m\rangle}(x_1,\cdot) \\ \mathbf{u}^{\langle m\rangle}(x_2,\cdot) \\ \vdots \\ \mathbf{u}^{\langle m\rangle}(x_M,\cdot) \end{bmatrix} = \begin{bmatrix} \psi(x_1, y_1^\nu) \\ \psi(x_2, y_2^\nu) \\ \vdots \\ \psi(x_M, y_M^\nu) \end{bmatrix},$$

which is denoted by $\mathbf{Z}_\nu \tilde{\mathbf{u}}^{\langle m\rangle} = \mathbf{\Psi}_\nu$ or ordering the index as the global numbering system: $\mathbf{Z}_\nu \mathbf{P}^{-1}\mathbf{u}^{\langle m\rangle} = \mathbf{\Psi}_\nu$, where $\mathbf{Z}_\nu = \operatorname{diag}\big(\mathbf{z}_N(y_1^\nu), \mathbf{z}_N(y_2^\nu), \ldots, \mathbf{z}_N(y_M^\nu)\big) \otimes \mathbf{R}_N^{-1}$ for $\nu \in \{b, t\}$. Therefore, to be specific, we denote the above boundary condition which corresponds to the bottom and top sides of the rectangular domain $[a, b] \times [c, d]$ by $\mathbf{Z}_b \mathbf{P}^{-1}\mathbf{u}^{\langle m\rangle} = \mathbf{\Psi}_b$ and $\mathbf{Z}_t \mathbf{P}^{-1}\mathbf{u}^{\langle m\rangle} = \mathbf{\Psi}_t$, respectively.



(a) Left boundary condition **(b)** Right boundary condition



(c) Bottom boundary condition **(d)** Top boundary condition

Figure 5.1: Discretizing each side of boundary over $[a, b] \times [c, d]$

Finally, we can construct the system of iterative linear equations for finding numerical results of the nonlinear Poisson equation from (5.6) and four above-mentioned boundary conditions. This linear system has the total number of $MN + 2(M + N)$ unknown variables including $\mathbf{u}^{\langle m\rangle}$, $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{d}_1$ and $\mathbf{d}_2$ as follows:

$$\left[\begin{array}{c|cccc} \mathbf{K} & \mathbf{X}\boldsymbol{\Phi}_y & \boldsymbol{\Phi}_y & \mathbf{Y}\boldsymbol{\Phi}_x & \boldsymbol{\Phi}_x \\ \hline \mathbf{Z}_l & 0 & 0 & \cdots & 0 \\ \mathbf{Z}_r & 0 & 0 & \cdots & 0 \\ \mathbf{Z}_b\mathbf{P}^{-1} & \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_t\mathbf{P}^{-1} & 0 & 0 & \cdots & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{u}^{\langle m\rangle} \\ \hline \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{d}_1 \\ \mathbf{d}_2 \end{array}\right] = \left[\begin{array}{c} \mathbf{A}_x^2\mathbf{A}_y^2\mathbf{f}^{\langle m-1\rangle} \\ \hline \boldsymbol{\Psi}_l \\ \boldsymbol{\Psi}_r \\ \boldsymbol{\Psi}_b \\ \boldsymbol{\Psi}_t \end{array}\right]. \tag{5.9}$$

Consequently, the iterative approximate solutions $\mathbf{u}^{\langle m\rangle}$ can be found by solving (5.9) in conjunction with an arbitrary initial guess of the iteration $\mathbf{u}^{\langle 0\rangle}$ that makes $\mathbf{f}^{\langle 0\rangle}$ to be available. Note that the stopping criterion for finding the m^{th} iterative approximate solution $\mathbf{u}^{\langle m\rangle}$ is stopped when the m^{th} error norm $\mathrm{e}^{\langle m\rangle} := \|\mathbf{u}^{\langle m\rangle} - \mathbf{u}^{\langle m-1\rangle}\|$ of difference between the current and consecutively previous solutions is less than the given convergent tolerance TOL. Therefore, an approximate solution $u(x, y)$ at arbitrary point $(x, y) \in \Omega$ can be estimated by using the transformation processes in the same way as (5.7) and (5.8). Then, we have

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

where $\mathbf{u}^{\langle m\rangle}$ is the m^{th} terminal solution in solving (5.9). In addition, we can consider the convergent sequence $\{\mathbf{u}^{\langle m\rangle}\}$ from the iterative linear system (5.9) which is respectively denoted by $\mathbf{H}\mathbf{u}^{\langle m\rangle} = \mathcal{F}(\mathbf{u}^{\langle m-1\rangle})$ or

$$\mathbf{u}^{\langle m\rangle} = \mathcal{G}(\mathbf{u}^{\langle m-1\rangle}), \tag{5.10}$$

where $\mathcal{G}(\mathbf{u}^{\langle m-1\rangle}) := \mathbf{H}^{-1}\mathcal{F}(\mathbf{u}^{\langle m-1\rangle})$ and $\mathbf{u}^{\langle m\rangle}$ also contains the unknowns $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{d}_1$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

and $\mathbf{d}_2$. We can see that (5.10) is in the form of fixed-point iteration, where $\mathcal{G}$ is called the iterative function. Let the exact solution $\mathbf{u}^*$ be a fixed point of $\mathcal{G}$, i.e., $\mathcal{G}(\mathbf{u}^*) = \mathbf{u}^*$. However, for considering the convergent sequence $\left\{\mathbf{u}^{\langle m \rangle}\right\}$ in this case, we assume that $\mathbf{H}$ is invertible. Then, we discuss regarding the convergence via the following theorem.

Theorem 5.1. ([19]) *Let $D \subset \mathbb{R}^n$ be closed and $\mathcal{G} : D \to D$. Assume that the mapping $\mathcal{G}$ is a contraction on D, i.e., there exists $\lambda < 1$ such that*

$$\|\mathcal{G}(\mathbf{u}) - \mathcal{G}(\mathbf{v})\| \leq \lambda \|\mathbf{u} - \mathbf{v}\|, \tag{5.11}$$

for all $\mathbf{u}, \mathbf{v} \in D$. Then, there exists unique $\mathbf{u}^* \in D$ such that $\mathbf{u}^* = \mathcal{G}(\mathbf{u}^*)$ and the fixed-point iterations converge to $\mathbf{u}^*$ for any choice of $\mathbf{u}^{\langle 0 \rangle} \in D$.

Theorem 5.2. ([19]) *Let $D \subset \mathbb{R}^n$ be closed and $\mathcal{G} : D \to D$ have continuous partial derivatives in D. Assume that there exists $\lambda < 1$ such that the natural norm of Jacobian matrix $\|\nabla \mathcal{G}(\mathbf{u})\| < \lambda$, for all $\mathbf{u} \in D$. Then, $\mathcal{G}$ is a contraction in D and satisfies (5.11).*

Therefore, the existence and uniqueness of the fixed-point iteration (5.10) can be described by using Theorems 5.1 and 5.2. In verification, we have to construct the domain $D \subseteq \mathbb{R}^{MN+2(M+N)}$. If the iterative function $\mathcal{G}$ for (5.10) is the contraction and maps D into D, it has a fixed point in the domain D. Furthermore, if we can show that there exists a constant $\lambda < 1$ such that for some natural matrix norm of the Jacobian, $\|\nabla \mathcal{G}(\mathbf{u})\| < \lambda$ for all $\mathbf{u} \in D$, then $\mathcal{G}$ has a unique fixed point $\mathbf{u}^*$ in D, and the fixed-point iteration is guaranteed to converge to $\mathbf{u}^*$ for any initial guess chosen in D. However, the aim of this dissertation is to construct numerical procedures for solving nonlinear PDEs. Thus, we do not provide theoretical verifications concerning fixed point for each example presented at the end of this chapter.

Moreover, we can also observe a convergence speed for the iterative method (5.9) by considering the following definition.

Definition 5.1. ([19]) The sequence $\left(\mathbf{u}^{\langle m\rangle}\right)$ converges with order p to the solution $\mathbf{u}^*$ if

$$\lim_{m\to\infty}\frac{\|\mathbf{u}^{\langle m+1\rangle}-\mathbf{u}^*\|}{\|\mathbf{u}^{\langle m\rangle}-\mathbf{u}^*\|^p}\le\mu$$

for some positive constant μ and $p\ge 1$. Here, p is always called the order of convergence. Obviously, the larger p and the smaller μ, they can affect the more quickly the sequence converges. In particular, convergence with order

- $p = 1$ and $\mu < 1$ is called linear convergence with rate of convergence μ,
- $p = 2$ and $\mu > 0$ is called quadratic convergence,
- $p = 3$ and $\mu > 0$ is called cubic convergence.

Usually, we scarcely know the analytical solution $\mathbf{u}^*$, however, a practical method to calculate the order of convergence p for the sequence $(\mathbf{u}^{\langle m\rangle})$ can be approximated from Definition 5.1 which its expression may be better understood when it is interpreted as $\left\|\mathbf{u}^{\langle m+1\rangle}-\mathbf{u}^{\langle m\rangle}\right\|=\mu\left\|\mathbf{u}^{\langle m\rangle}-\mathbf{u}^{\langle m-1\rangle}\right\|^p$. The first way to compute the order of convergence p, we take the natural logarithm function ln on both sides of this equation. Then, we have

$$\ln\left\|\mathbf{u}^{\langle m+1\rangle}-\mathbf{u}^{\langle m\rangle}\right\|=p\ln\left\|\mathbf{u}^{\langle m\rangle}-\mathbf{u}^{\langle m-1\rangle}\right\|+\ln\mu \tag{5.12}$$

or simplified form: $\ln\mathrm{e}^{\langle m+1\rangle}=p\ln\mathrm{e}^{\langle m\rangle}+\ln\mu$. We can see that this is a linear equation, where $\ln\mathrm{e}^{\langle m\rangle}$ vs $\ln\mathrm{e}^{\langle m+1\rangle}$ are corresponding with the variables x vs y in the Cartesian coordinate, respectively. Therefore, the order of convergence p is a slope in plotting linear graph between $\ln\mathrm{e}^{\langle m\rangle}$ vs $\ln\mathrm{e}^{\langle m+1\rangle}$.

Another way is to eliminate $\ln\mu$ by constructing two consecutive linear equations from (5.12). Next, we subtract both linear equations together, the term $\ln\mu$ is then removed. After that we rearrange the subtracted equation, we finally get the equation for estimating the order of convergence p which is converging to

$$p\approx\frac{\ln\left\|\mathbf{u}^{\langle m+1\rangle}-\mathbf{u}^{\langle m\rangle}\right\|-\ln\left\|\mathbf{u}^{\langle m\rangle}-\mathbf{u}^{\langle m-1\rangle}\right\|}{\ln\left\|\mathbf{u}^{\langle m\rangle}-\mathbf{u}^{\langle m-1\rangle}\right\|-\ln\left\|\mathbf{u}^{\langle m-1\rangle}-\mathbf{u}^{\langle m-2\rangle}\right\|}=\frac{\ln\left(\mathrm{e}^{\langle m+1\rangle}/\mathrm{e}^{\langle m\rangle}\right)}{\ln\left(\mathrm{e}^{\langle m\rangle}/\mathrm{e}^{\langle m-1\rangle}\right)}.$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

For computational convenience, we provide the following algorithm in the form of pseudocode for finding an approximate solution of the two-dimensional nonlinear Poisson equation over the irregular domains by using our developed FIM-CPE.

Algorithm 3 Algorithm for solving nonlinear Poisson equation by the FIM-CPE

```
Input: $a$, $b$, $c$, $d$, $\alpha$, $\beta$, $\gamma$, $\psi$, $f$, $M$, $N$, $TOL$ and $\mathbf{u}^{\langle 0\rangle}$;
Output: An approximate solution $u(x,y)$;
Set $x_k = \frac{1}{2}\left[(b-a)\cos\left(\frac{2k-1}{2M}\pi\right)+a+b\right]$ for $k \in \{1,2,3,\ldots,M\}$ in descending order;
Set $y_h = \frac{1}{2}\left[(d-c)\cos\left(\frac{2h-1}{2N}\pi\right)+c+d\right]$ for $h \in \{1,2,3,\ldots,N\}$ in descending order;
Calculate the total number of grid points $H = M \times N$;
Construct $x_i$ and $y_i$ in the global numbering system for $i \in \{1,2,3,\ldots,H\}$;
Compute $\mathbf{K}$, $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{P}$, $\mathbf{\Phi}_x$, $\mathbf{\Phi}_y$, $\mathbf{A}_x$, $\mathbf{A}_y$, $\mathbf{Z}_l$, $\mathbf{Z}_r$, $\mathbf{Z}_b$, $\mathbf{Z}_t$, $\mathbf{\Psi}_l$, $\mathbf{\Psi}_r$, $\mathbf{\Psi}_b$ and $\mathbf{\Psi}_t$;
Set $m = 1$;
do
    Compute $\mathbf{f}^{\langle m-1\rangle} = \left[f_1^{\langle m-1\rangle}, f_2^{\langle m-1\rangle}, f_3^{\langle m-1\rangle}, \ldots, f_H^{\langle m-1\rangle}\right]^\top$;
    Find $\mathbf{u}^{\langle m\rangle}$ by solving the iterative linear system (5.9);
    Compute $\mathrm{e}^{\langle m\rangle} = \|\mathbf{u}^{\langle m\rangle} - \mathbf{u}^{\langle m-1\rangle}\|$;
    Update $m = m+1$;
while $\mathrm{e}^{\langle m-1\rangle} \geq TOL$
return $u(x,y) = \left(\mathbf{z}_N(y)\mathbf{R}_N^{-1} \otimes \mathbf{z}_M(x)\mathbf{R}_M^{-1}\right)\mathbf{u}^{\langle m\rangle}$;
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

5.3 Numerical Examples for Testing Algorithm 3

In this section, we show the efficiency and effectiveness of our constructed Algorithm 3 through seven numerical examples for Poisson-type problems over irregular-shaped domains, including pentagonal, circular, L-shaped, butterfly, peanut-shaped and elliptic regions. Moreover, we display the comparisons between analytical solution u^* and numerical solution u measured by the MAE $= \frac{1}{H}\sum_{i=1}^{H}|u^*(x_i,y_i)-u(x_i,y_i)|$, together with iterative numbers m and CPU time(s) in order to demonstrate the computational cost. We further express the order and rate of convergences of Algorithm 3 via plotting graphs. Finally, we also depict a plotting of numerical result in three-dimensional diagram.

Example 5.1. Consider the nonlinear Poisson-type equation in [31] over the pentagonal domain Ω as shown in Figure 5.3(a) as follows.

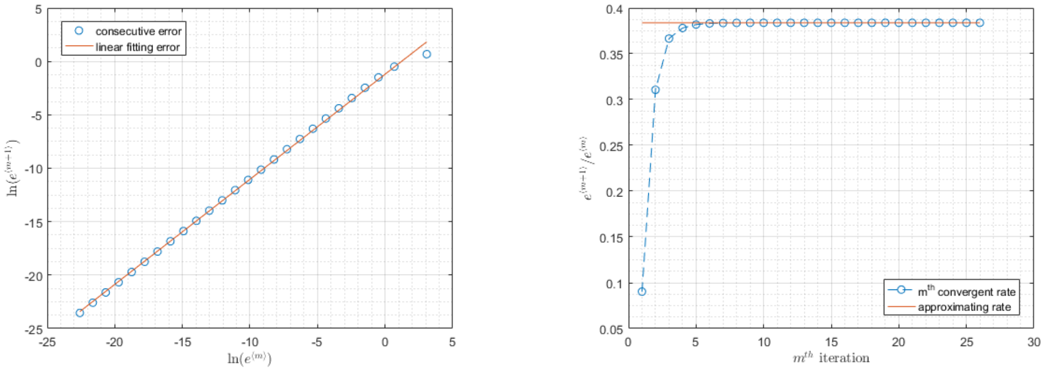
$$\nabla^2 u + u^2 = e^{2x} \cos^2 y, \quad (x, y) \in \Omega, \tag{5.13}$$

with the Dirichlet boundary condition corresponds to the analytical solution $u^*(x, y) = e^x \cos y$. When our proposed Algorithm 3 is employed to solve this problem, (5.13) can be transformed into the matrix form (5.6), where $\mathbf{K} = \mathbf{A}_y^2 + \mathbf{A}_x^2$ and each element of $\mathbf{f}^{\langle m-1\rangle}$ is $f_i^{\langle m-1\rangle} = e^{2x_i} \cos^2 y_i - (u_i^{\langle m-1\rangle})^2$ for $i \in \{1, 2, 3, \ldots, H\}$. The initial guess of iteration $\mathbf{u}^{\langle 0\rangle}$ is chosen to be the zero vector. To test the accuracy of our method, we perform by varying the convergent tolerances which are taken as $TOL = 10^{-n}$ for $n \in \{1, 2, 3, 4, 5\}$ under the discretizing domain $M = N$, namely, the nodal points $H \in \{10 \times 10, 12 \times 12, 14 \times 14\}$ that demonstrates the MAE in Table 5.1 with their numbers of iteration m and CPU time(s). Moreover, we also find the order and rate of convergences via plotting graphs in Figures 5.2(a) and 5.2(b), respectively, for the discretizing grid numbers $H = 10 \times 10$ and the convergent tolerance $TOL = 10^{-10}$. However, their convergences produce the order $p = 0.9916 \approx 1$ and the rate $\mu = 0.3838$. Therefore, a consequence of the proposed Algorithm 3 converges with a linear order and a quick rate because μ quite approaches to the zero side. Finally, we plot the three-dimensional graph of our approximate solutions as depicted in Figure 5.3(b).

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

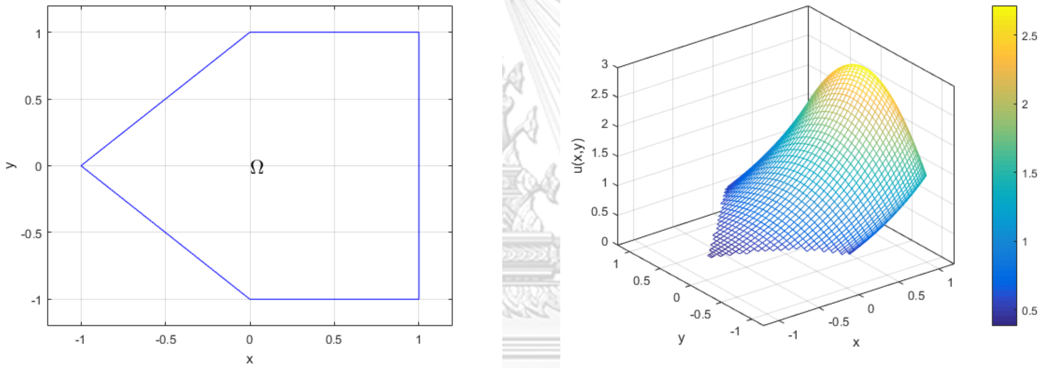
Table 5.1: MAE for each discretizing H at different tolerances of Example 5.1

TOL	$H = 10 \times 10$			$H = 12 \times 12$			$H = 14 \times 14$		
	m	MAE	Time(s)	m	MAE	Time(s)	m	MAE	Time(s)
10^{-1}	5	1.0397×10^{-3}	0.3661	6	3.8033×10^{-4}	0.4826	7	1.5128×10^{-4}	0.5812
10^{-2}	7	1.5311×10^{-4}	0.4016	9	2.1407×10^{-5}	0.5445	10	8.5425×10^{-6}	0.6547
10^{-3}	10	8.6638×10^{-6}	0.4429	11	3.1447×10^{-6}	0.5762	12	1.2571×10^{-6}	0.7324
10^{-4}	12	1.2771×10^{-6}	0.4691	14	1.7704×10^{-7}	0.5866	15	7.1399×10^{-8}	0.8144
10^{-5}	14	1.8822×10^{-7}	0.4854	16	2.6015×10^{-8}	0.6273	17	1.1618×10^{-8}	0.9071



(a) Order of convergence **(b)** Rate of convergence

Figure 5.2: Order and rate of convergences of Example 5.1



(a) Pentagonal domain Ω **(b)** Graphical solution u

Figure 5.3: Pentagonal domain and numerical solution of Example 5.1

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Example 5.2. Consider the nonlinear Poisson-type equation as in [6] over the circular domain as Figure 5.5(a) as follows.

$$\begin{aligned} -\nabla^2 u &= e^u + \beta(x,y) \quad \text{in} \quad x^2+y^2 \le 1, \\ u &= 0 \qquad\qquad\quad \text{on} \quad x^2+y^2 = 1, \end{aligned} \tag{5.14}$$

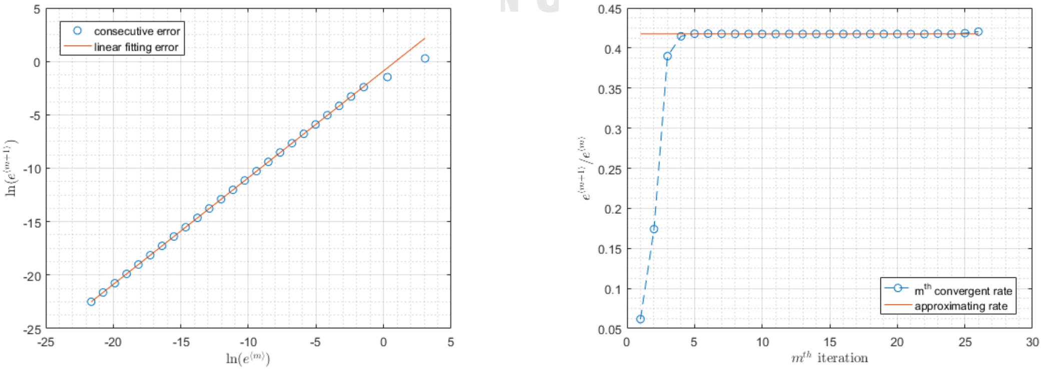
where $\beta(x,y)$ is chosen corresponding to the exact solution is $u^*(x,y) = (1-x^2-y^2)e^{x\cos y}$. Our presented numerical Algorithm 3 is used to find the approximate solution of (5.14), where $\mathbf{K} = -\mathbf{A}_y^2 - \mathbf{A}_x^2$ and each component of $\mathbf{f}^{\langle m-1\rangle}$ is the right-hand-side term of

(5.14). The initial guess of iteration $\mathbf{u}^{\langle 0\rangle}$ is selected to be the unit-element vector. The accuracies of our method for each convergent tolerance $TOL = 10^{-n}$ for $n \in \{1,2,3,4,5\}$ and the computational points by $H \in \{10\times 12, 12\times 14, 14\times 16\}$ are shown in Table 5.2 together with their numbers of iteration m and CPU time(s). We further obtain the convergent order $p = 0.9991 \approx 1$ and convergent rate $\mu = 0.4180$ as demonstrated in Figures 5.4(a) and 5.4(b), respectively, for the parameters discretizing grid numbers $H = 10\times 10$ and the convergent tolerance $TOL = 10^{-10}$. We can see that the rate of convergence μ quite approaches to the zero side, hence it quickly converges to the exact solution. Moreover, we also display the behavior of our approximate solutions via the plotting of three-dimensional graph in Figure 5.5(b).

Table 5.2: MAE for each discretizing H at different tolerances of Example 5.2

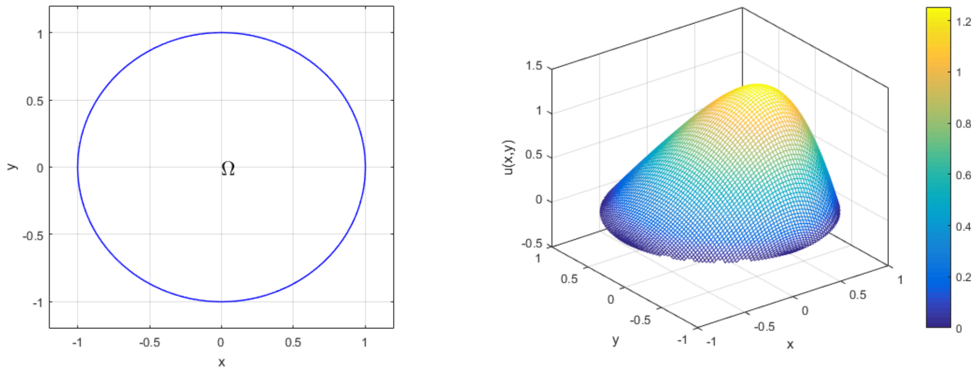
TOL	$H = 10\times 12$			$H = 12\times 14$			$H = 14\times 16$		
	m	MAE	Time(s)	m	MAE	Time(s)	m	MAE	Time(s)
10^{-1}	4	3.2425×10^{-3}	0.3447	4	3.2411×10^{-3}	0.3786	4	3.2401×10^{-3}	0.4149
10^{-2}	6	5.6464×10^{-4}	0.3643	7	2.3565×10^{-4}	0.4075	7	2.3558×10^{-4}	0.4603
10^{-3}	9	4.1090×10^{-5}	0.4078	9	4.1072×10^{-5}	0.4469	9	4.1060×10^{-5}	0.5272
10^{-4}	12	2.9894×10^{-6}	0.4281	12	2.9878×10^{-6}	0.4946	12	2.9869×10^{-6}	0.6003
10^{-5}	14	5.2207×10^{-7}	0.4317	14	5.2064×10^{-7}	0.5321	15	2.1727×10^{-7}	0.6628

จุฬาลงกรณ์มหาวิทยาลัย



(a) Order of convergence (b) Rate of convergence

Figure 5.4: Order and rate of convergences of Example 5.2



(a) Circular domain Ω **(b)** Graphical solution u

Figure 5.5: Circular domain and numerical solution of Example 5.2

We can see that two previous examples, considered also in [31] and [6], contain only one term of Laplace operator. Therefore, the next two examples, we construct the nonlinear Poisson-type equations in fully form based on (5.1) by modifying from the linear Poisson-type problems in [41] and [31] which are added the nonlinear forcing terms with constant and variable coefficients for Examples 5.3 and 5.4, respectively, in order to verify the performance of our numerical Algorithm 3.

Example 5.3. Consider the modified nonlinear Poisson equation obtained [41] on the L-shaped domain Ω shown in Figure 5.7(a) with the constant coefficients as follows.

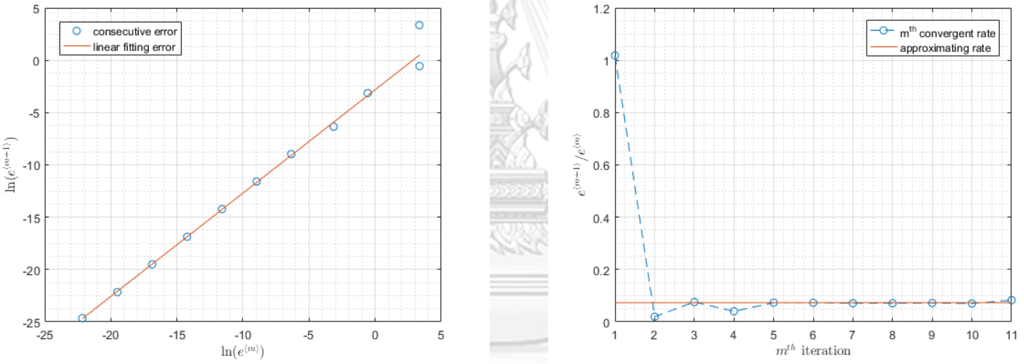
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$\nabla^2 u + 4u_x + 4u_y + 2u = 4\cos(x+y) + \frac{2u}{\sin^2 u + 1} - \frac{4\cos x \sin y}{3 - \cos(2\cos x \sin y)}, \quad (x,y) \in \Omega, \quad (5.15)$$

with the Dirichlet boundary condition corresponds to exact solution $u^*(x,y) = \cos x \sin y$. Then, we use our improved Algorithm 3 to solve the problem (5.15), which can be written in the matrix form (5.6), where $\mathbf{K} = \mathbf{A}_y^2 + \mathbf{A}_x^2 + 4\mathbf{A}_x\mathbf{A}_y^2 + 4\mathbf{A}_x^2\mathbf{A}_y + 2\mathbf{A}_x^2\mathbf{A}_y^2$ and each component of $\mathbf{f}^{\langle m-1 \rangle}$ is explicitly the right-hand-side term of (5.15). The initial guess $\mathbf{u}^{\langle 0 \rangle}$ is chosen to be the zero vector. We test the effectiveness and efficiency of our method by selecting the convergent tolerance $TOL = 10^{-n}$ for $n \in \{1,2,3,4,5\}$ under the meshing domain $M = N$, that is $H \in \{11 \times 11, 13 \times 13, 15 \times 15\}$ as shown in Table 5.3 together with their iterative numbers m and CPU time(s). We further obtain the order $p = 0.9876 \approx 1$ and rate $\mu = 0.0620$ of convergences as shown in Figures 5.6(a) and 5.6(b), respectively, for parameters $H = 11 \times 11$ and $TOL = 10^{-10}$. We can see that the rate of convergence μ approaches to the zero side. Hence, it very quickly converges to the solution. Finally, the three-dimensional graph of the numerical solution is also plotted in Figure 5.7(b).

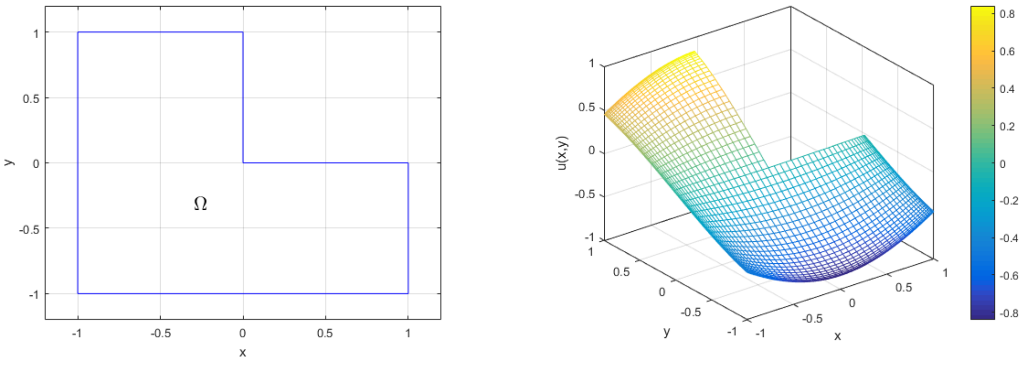
Table 5.3: MAE for each discretizing H at different tolerances of Example 5.3

TOL	$H = 11 \times 11$			$H = 13 \times 13$			$H = 15 \times 15$		
	m	MAE	Time(s)	m	MAE	Time(s)	m	MAE	Time(s)
10^{-1}	4	1.8951×10^{-5}	0.3396	5	1.4978×10^{-6}	0.3782	5	9.8880×10^{-7}	0.4295
10^{-2}	5	1.3980×10^{-6}	0.3458	6	1.0862×10^{-7}	0.3910	6	6.3262×10^{-8}	0.4479
10^{-3}	6	1.0183×10^{-7}	0.3519	7	7.8660×10^{-9}	0.4093	7	4.0312×10^{-9}	0.4608
10^{-4}	7	7.3534×10^{-9}	0.3654	8	5.6536×10^{-10}	0.4107	8	2.5461×10^{-10}	0.4876
10^{-5}	7	7.3534×10^{-9}	0.3743	8	5.6536×10^{-10}	0.4180	9	1.6620×10^{-11}	0.5258



(a) Order of convergence **(b)** Rate of convergence

Figure 5.6: Order and rate of convergences of Example 5.3



(a) L-shaped domain Ω **(b)** Graphical solution u

Figure 5.7: L-shaped domain and numerical solution of Example 5.3

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Example 5.4. Consider the modified nonlinear Poisson equation obtained from [31] with the variable coefficients and nonlinear singular forcing term over the butterfly domain Ω as shown in Figure 5.9(a) as follows.

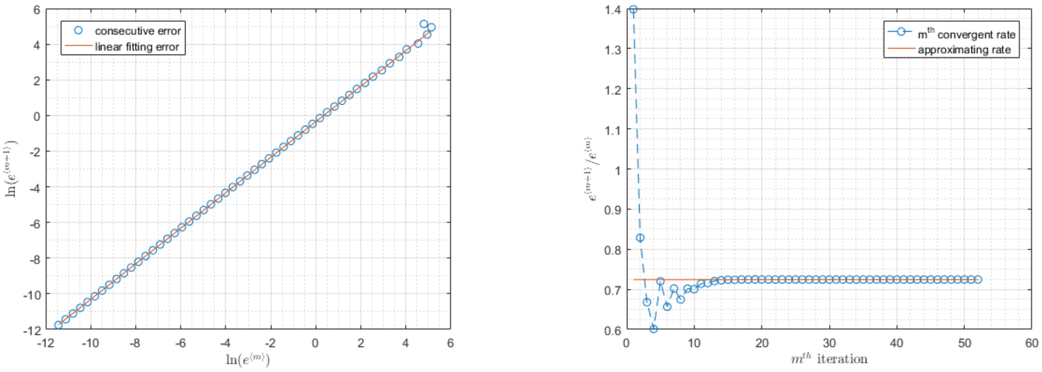
$$\nabla^2 u + x^2 u_x - y^2 u_y = 2u + (x-y)u\ln u \quad \text{in} \quad x^4 - 4x^2 + y^2 \le 0, \tag{5.16}$$

with the Dirichlet boundary condition corresponds to analytical solution $u^*(x,y) = e^{x+y}$. First, we can transform (5.16) into the matrix form of (5.6) by using the numerical Algorithm 3, where $\mathbf{K} = \mathbf{A}_y^2 + \mathbf{A}_x^2 + \mathbf{A}_x\mathbf{A}_y^2\mathbf{X}^2 - 2\mathbf{A}_x^2\mathbf{A}_y^2\mathbf{X} - \mathbf{A}_x^2\mathbf{A}_y\mathbf{Y}^2 + 2\mathbf{A}_x^2\mathbf{A}_y^2\mathbf{Y}$ and each component of $\mathbf{f}^{\langle m-1\rangle}$ is explicitly the right-hand-side term of (5.16). We notice that the forcing term f is singular for $u \le 0$. Thus, the initial guess of the iteration $\mathbf{u}^{\langle 0\rangle}$ is chosen to be the unit-element vector. Next, we show the performance of this method via the convergent tolerance $TOL = 10^{-n}$ for $n \in \{1,2,3,4,5\}$ under the discretizing domain $M \neq N$, namely, the computational nodes $H \in \{10\times 12, 12\times 14, 14\times 16\}$ as shown in Table 5.4 together with the numbers of iteration m and CPU time(s). We further obtain the order of convergent $p = 0.9962 \approx 1$ and rate of convergence $\mu = 0.7235$ as demonstrated in Figures 5.8(a) and 5.8(b), respectively, for the parameters discretizing grid numbers $H = 10\times 10$ and the convergent tolerance $TOL = 10^{-5}$. We can see that the rate of convergence μ quite approaches to the one side, thus this Example 5.4 converges to the solution slower than the previous examples. Finally, we show the graph of the numerical solution in three dimensions as Figure 5.9(b).

Table 5.4: MAE for each discretizing H at different tolerances of Example 5.4

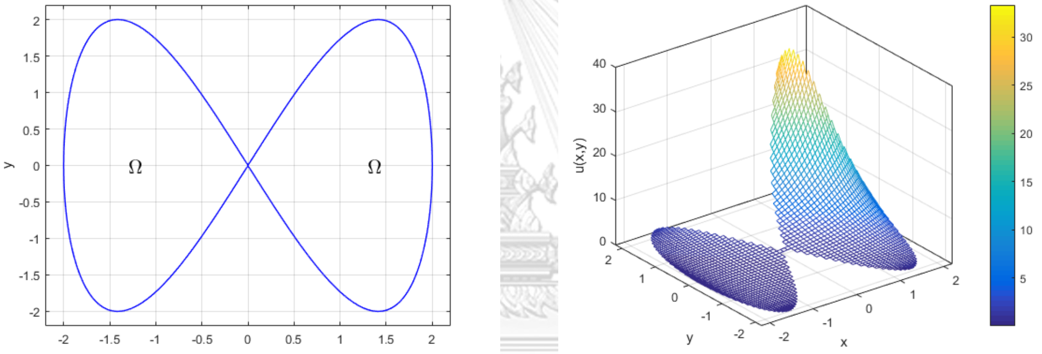
TOL	$H = 10\times 12$			$H = 12\times 14$			$H = 14\times 16$		
	m	MAE	Time(s)	m	MAE	Time(s)	m	MAE	Time(s)
10^{-1}	23	2.9980×10^{-4}	0.4883	32	5.1969×10^{-5}	0.7872	29	4.8389×10^{-5}	0.9561
10^{-2}	30	2.2497×10^{-5}	0.5734	38	5.9046×10^{-6}	1.0168	35	5.8914×10^{-6}	1.0725
10^{-3}	36	3.5654×10^{-6}	0.5943	45	4.6999×10^{-7}	1.0892	42	5.0498×10^{-7}	1.3126
10^{-4}	42	2.0847×10^{-6}	0.6494	51	6.1648×10^{-8}	1.1168	48	6.1503×10^{-8}	1.4271
10^{-5}	48	2.0620×10^{-6}	0.7452	57	2.1871×10^{-8}	1.5885	55	5.2418×10^{-9}	1.6056

CHULALONGKORN UNIVERSITY



(a) Order of convergence **(b)** Rate of convergence

Figure 5.8: Order and rate of convergences of Example 5.4



(a) Butterfly domain Ω **(b)** Graphical solution u

Figure 5.9: Butterfly domain and numerical solution of Example 5.4

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

From the four previous examples, we solved the two-dimensional nonlinear Poisson-type equations over the numerous irregular domains, including the pentagonal, circular, L-shaped and butterfly regions. They provide the high accurate solutions compared to their analytical solutions in term of the MAE with small computational nodes. Next, we consider the problems as presented in [54] and compare results obtained from our Algorithm 3 and other methods on the complicated irregular domains such as the peanut-shaped and elliptic regions in the polar coordinate form as follows.

Example 5.5. Consider the nonlinear Poisson equation given by [54] over the peanut-shaped domain Ω as depicted in Figure 5.11(a) as follows.

$$\nabla^2 u = 4u^3, \quad (x, y) \in \Omega, \tag{5.17}$$

where $\Omega = \{(\theta, r) \mid r(\theta) = 0.3\sqrt{\cos 2\theta + \sqrt{1.1 - \sin^2 2\theta}},\ 0 \le \theta \le 2\pi\}$ with the Dirichlet boundary condition corresponds to the analytical solution $u^*(x, y) = \frac{1}{4+x+y}$. By our proposed numerical Algorithm 3, (5.17) can be transformed into the matrix form (5.6), where $\mathbf{K} = \mathbf{A}_y^2 + \mathbf{A}_x^2$ and each component of $\mathbf{f}^{\langle m-1\rangle}$ is $f_i^{\langle m-1\rangle} = 4\big(u_i^{\langle m-1\rangle}\big)^3$ for $i \in \{1, 2, 3, \ldots, H\}$. The iteratively initial guess $\mathbf{u}^{\langle 0\rangle}$ is chosen to be the zero vector. We test the performance of our algorithm by comparing with the homotopy analysis method (HAM), the fictitious time integration method (FTIM) and the dual reciprocity method (DRM) which are reported in [60], [61] and [54], respectively. According to the HAM [60] with grid point $H = 11 \times 11$, their results are demonstrated that the best maximum error of the 8th iterations is 1.24×10^{-10}, while our Algorithm 3 gives 2.1561×10^{-11} using merely 4 iterations. Next, the comparison of root mean square error (RMSE) between our method with $TOL = 10^{-5}$ and the FTIM [61] is shown in Table 5.5. Furthermore, Table 5.6 shows absolute maximal error (AME) and RMSE of our method with $TOL = 10^{-2}$ and the DRM [54]. We also express the order of convergence $p = 0.9979 \approx 1$ and the rate of convergence $\mu = 0.0109$ via plotting graph as shown in Figures 5.10(a) and 5.10(b), respectively, for $H = 5 \times 5$ and $TOL = 10^{-15}$ at the initial guess $\mathbf{u}^{\langle 0\rangle}$ is unit-element vector. This rate μ is very near the zero side, then it rapidly converges to the solution. Finally, the graph of the numerical solution is depicted in Figure 5.11(b).

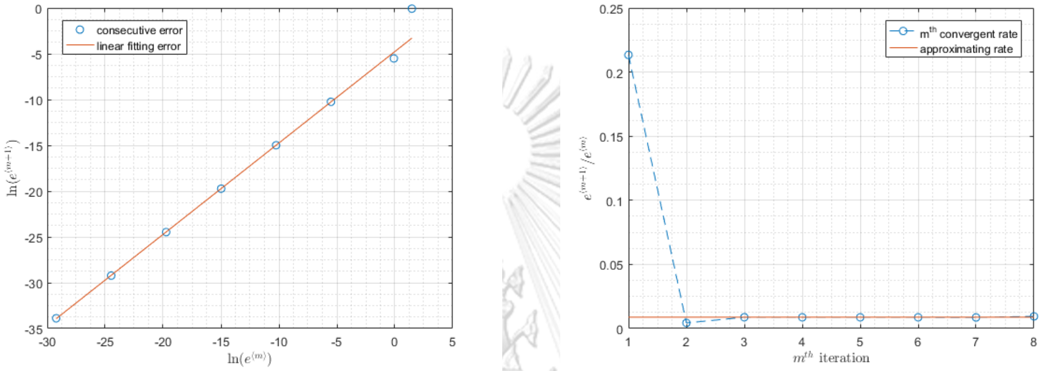
CHULALONGKORN UNIVERSITY

Table 5.5: RMSE for each discretizing H at $TOL = 10^{-5}$ of Example 5.5

H	RMSE		m	Time(s)
	FTIM [61]	Our Algorithm 3		
4×4	$\approx 1.0 \times 10^{-6}$	3.3928×10^{-6}	3	0.1474
8×8	$\approx 1.0 \times 10^{-7}$	4.0990×10^{-11}	4	0.1622
10×10	$\approx 1.0 \times 10^{-9}$	1.5483×10^{-11}	4	0.1756

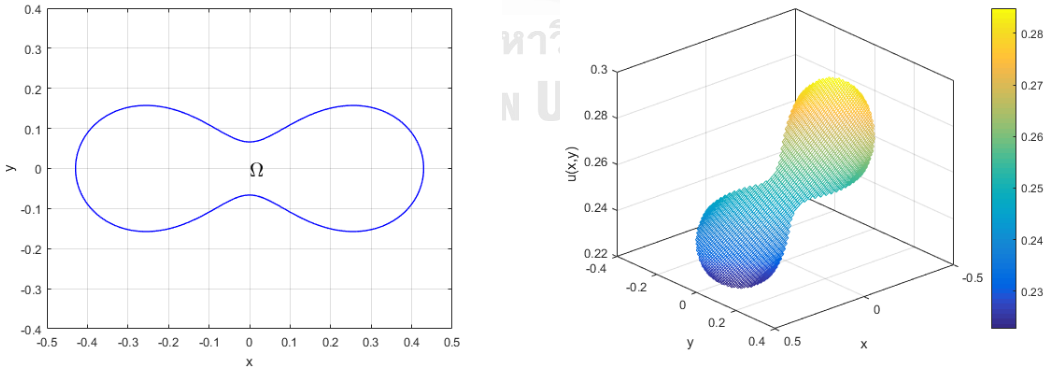
Table 5.6: AME and RMSE for each discretizing H at $TOL = 10^{-2}$ of Example 5.5

$\boldsymbol{H}$	**AME**		**RMSE**	
	DRM [54]	**Our Algorithm 3**	**DRM [54]**	**Our Algorithm 3**
10×10	5.3×10^{-7}	9.7458×10^{-9}	1.6×10^{-7}	3.7660×10^{-9}
20×20	3.3×10^{-7}	1.6809×10^{-8}	7.9×10^{-8}	4.4999×10^{-9}
30×30	2.4×10^{-8}	9.0301×10^{-10}	7.3×10^{-9}	9.5182×10^{-11}



(a) Order of convergence **(b)** Rate of convergence

Figure 5.10: Order and rate of convergences of Example 5.5



(a) Peanut-shaped domain Ω **(b)** Graphical solution u

Figure 5.11: Peanut-shaped domain and numerical solution of Example 5.5

Example 5.6. Consider the nonlinear Poisson-type equation given by [54] on boundary of the elliptic domain Ω as shown in Figure 5.13(a) as follows.

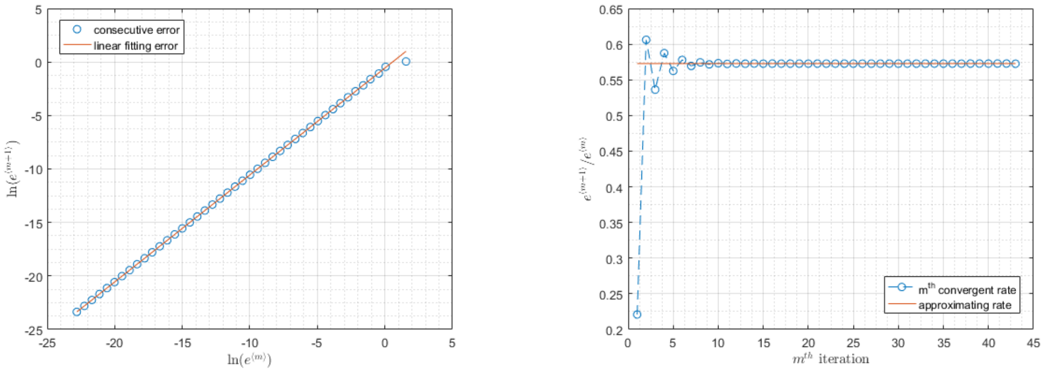
$$\nabla^2 u = u^3 - \frac{5}{2} - \left(1 - \frac{x^2}{4} - y^2\right)^3, \quad (x,y) \in \Omega, \tag{5.18}$$

where $\Omega = \{(x,y) \ \ x = 2\cos\theta \text{ and } y = \sin\theta,\ 0 \leq \theta \leq 2\pi\}$ with the Dirichlet boundary condition corresponds to the analytical solution $u^*(x,y) = 1 - \frac{x^2}{4} - y^2$. We can transform (5.18) into the matrix form (5.6), where $\mathbf{K} = \mathbf{A}_y^2 + \mathbf{A}_x^2$ and each entry of $\mathbf{f}^{\langle m-1\rangle}$ is explicitly the right-hand-side term of (5.18). The initial guess of the iteration $\mathbf{u}^{\langle 0\rangle}$ is selected to be the zero vector. Next, we compare the absolute error obtained from our developed method with $TOL = 10^{-5}$ to the asymptotic numerical method combined with the method of fundamental solution (ANM-MFS) with the thin plate splines RBFs (TPS-RBF) and the multiquadric RBFs (MQ-RBF) in [59] and the DRM in [54] as demonstrated in Table 5.7. Additionally, we can illustrate the order of convergence $p = 0.9996 \approx 1$ in Figure 5.12(a) and the rate of convergence $\mu = 0.5728$ in Figure 5.12(b) for the discretizing grid number $H = 10 \times 10$ and the convergent tolerance $TOL = 10^{-10}$. We can see that the convergence rate of μ is approximately the middle between zero and one. Thus, it approaches to the exact solution with an acceptable rate Finally, we show the graph of the numerical solution in Figure 5.13(b).

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

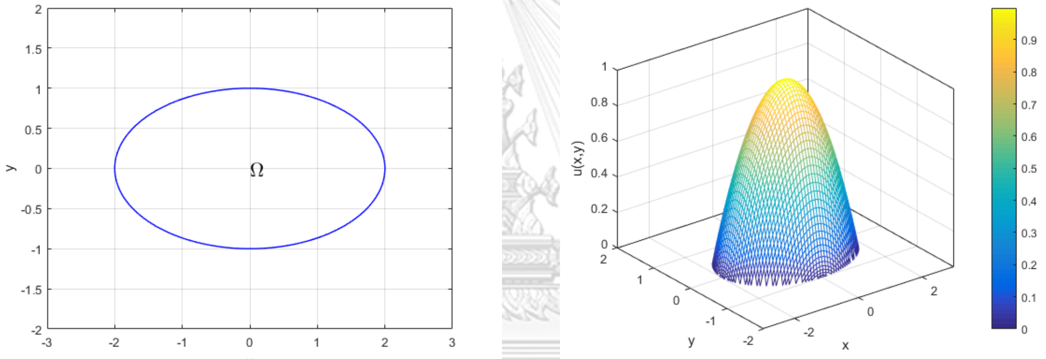
Table 5.7: Absolute error at different points (x,y) of Example 5.6

(x,y)	ANM-MFS [59]		DRM [54]		Our Algorithm 3	
	TPS-RBF	MQ-RBF	$H = 5 \times 5$	$H = 10 \times 10$	$H = 5 \times 5$	$H = 10 \times 10$
$(1.5, 0.00)$	3.9430×10^{-3}	3.3290×10^{-3}	3.0×10^{-6}	3.0×10^{-6}	3.6917×10^{-7}	1.7089×10^{-7}
$(0.0, 0.45)$	6.9580×10^{-3}	5.8120×10^{-3}	8.4×10^{-5}	1.7×10^{-5}	1.5027×10^{-6}	8.0096×10^{-7}
$(0.6, 0.45)$	1.8275×10^{-2}	1.5634×10^{-2}	2.5×10^{-5}	2.0×10^{-6}	1.2122×10^{-6}	5.9233×10^{-7}
$(1.2, 0.35)$	2.2270×10^{-3}	1.4760×10^{-3}	6.8×10^{-5}	9.0×10^{-6}	6.3852×10^{-7}	2.6843×10^{-7}
$(0.9, 0.00)$	5.4930×10^{-3}	4.4580×10^{-3}	9.3×10^{-5}	1.0×10^{-6}	1.2987×10^{-6}	6.0220×10^{-7}



(a) Order of convergence **(b)** Rate of convergence

Figure 5.12: Order and rate of convergences of Example 5.6



(a) Elliptic domain Ω **(b)** Graphical solution u

Figure 5.13: Elliptic domain and numerical solution of Example 5.6

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

In fact, our procedure of Algorithm 3 can be performed not only the presented Poisson-type equation (5.1) with nonlinear forcing term $f(x, y, u)$, but it is also can carry out in the more general nonlinear Poisson-type equation that the physical field u is governed by $\nabla^2 u = F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy})$. However, because F can be any type of nonlinear functions, it is impossible to write one procedure to fit all kinds of F. Thus, to illustrate our idea, we choose one of the form shown in Example 5.7. We devise the numerical algorithm based on our idea of developed FIM-CPE for finding approximate solution and show that our method can handle this kind of fully nonlinear problem effectively.

Example 5.7. Consider a fully nonlinear Poisson-type equation with nonlinear singular forcing term on the elliptic domain Ω as shown in Figure 5.15(a) as follows.

$$\nabla^2 u + uu_x - uu_y - u = \left(x^{-1} - 1\right) u^2 + \ln\left|ux^{-1}\right| - y \quad \text{in} \quad x^2 - xy + y^2 \le 1, \qquad (5.19)$$

where $x \neq 0$ and the Dirichlet boundary conditions coincident to the analytical solution $u^*(x,y) = xe^y$. Since we know that $uu_x = (\frac{u^2}{2})_x$ and $uu_y = (\frac{u^2}{2})_y$. The linearization process can be adapted to (5.19), by letting the left-hand-side term to be

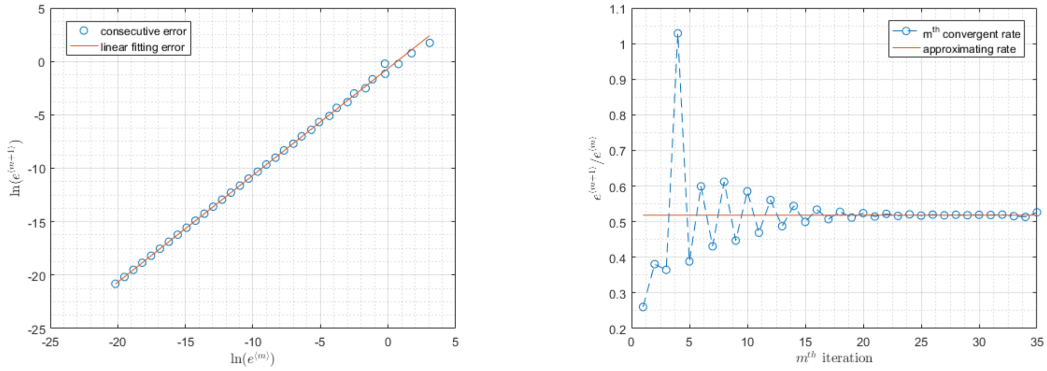
$$u_{xx}^{\langle m\rangle} + u_{yy}^{\langle m\rangle} + \left(\frac{1}{2}u^{\langle m-1\rangle}u^{\langle m\rangle}\right)_x - \left(\frac{1}{2}u^{\langle m-1\rangle}u^{\langle m\rangle}\right)_y - u^{\langle m\rangle}.$$

Thus, by hiring our numerical Algorithm 3 to solve this problem, we have (5.6), where $\mathbf{K} = \mathbf{A}_y^2 + \mathbf{A}_x^2 + \mathbf{A}_x\mathbf{A}_y^2\mathbf{D} - \mathbf{A}_x^2\mathbf{A}_y\mathbf{D} - \mathbf{A}_x^2\mathbf{A}_y^2$ when $\mathbf{D} = \frac{1}{2}\mathrm{diag}(u_1^{\langle m-1\rangle}, u_2^{\langle m-1\rangle}, \ldots, u_H^{\langle m-1\rangle})$ and each element of $\mathbf{f}^{\langle m-1\rangle}$ is explicitly its right-hand-side term. Moreover, we can see that the forcing term f is singular when $u = 0$. Hence, the selected initial guess $\mathbf{u}^{\langle 0\rangle}$ is the unit-element vector. Then, we test the accuracy of the results by varying the tolerance $TOL = 10^{-n}$ for $n \in \{1,2,3,4,5\}$ under discretizing $H \in \{12 \times 12, 14 \times 14, 16 \times 16\}$ as shown the MAE and CPU time(s) in Table 5.8. Furthermore, the convergent order is $p = 1.0003 \approx 1$ and the convergent rate is $\mu = 0.5274$ as shown in Figure 5.14 for parameters $H = 10 \times 10$ and $TOL = 10^{-9}$. Also, we plot the three-dimensional graph of numerical results in Figure 5.15(b).

Table 5.8: MAE for each discretizing H at different tolerances of Example 5.7

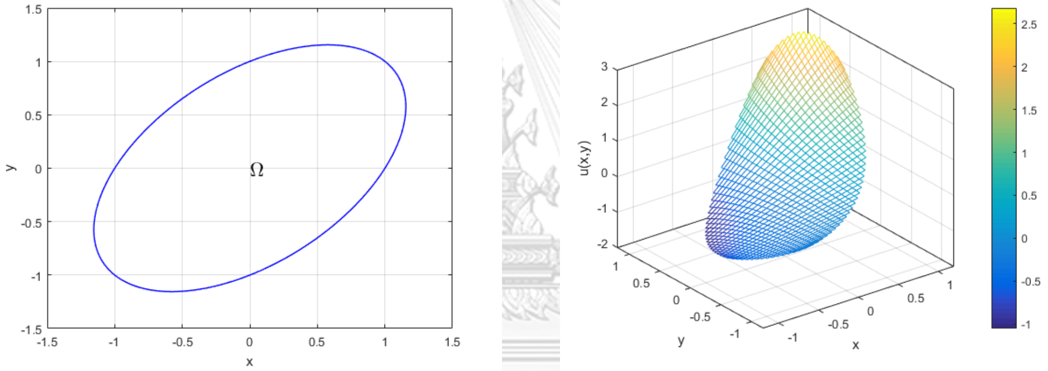
$\boldsymbol{TOL}$	$\boldsymbol{H = 12 \times 12}$			$\boldsymbol{H = 14 \times 14}$			$\boldsymbol{H = 16 \times 16}$		
	$\boldsymbol{m}$	**MAE**	**Time(s)**	$\boldsymbol{m}$	**MAE**	**Time(s)**	$\boldsymbol{m}$	**MAE**	**Time(s)**
10^{-1}	8	1.3566×10^{-3}	0.3787	10	3.4830×10^{-4}	0.5661	40	1.5667×10^{-7}	1.7427
10^{-2}	12	8.9169×10^{-5}	0.5022	11	2.9352×10^{-5}	0.5942	41	8.5052×10^{-8}	1.7507
10^{-3}	15	1.1437×10^{-5}	0.5447	17	2.2632×10^{-6}	0.6943	43	2.6786×10^{-8}	1.7862
10^{-4}	19	8.4218×10^{-7}	0.5856	21	2.0292×10^{-7}	0.7935	45	8.4077×10^{-9}	1.8478
10^{-5}	22	1.1952×10^{-7}	0.6240	25	1.9052×10^{-8}	0.8312	46	4.7365×10^{-9}	1.8574

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



(a) Order of convergence **(b)** Rate of convergence

Figure 5.14: Order and rate of convergences of Example 5.7



(a) Elliptic domain Ω **(b)** Graphical solution u

Figure 5.15: Elliptic domain and numerical solution of Example 5.7

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

CHAPTER VI

CONCLUSIONS AND DISCUSSIONS

In this chapter, we summarize the overview and highlight of our works from each previous chapter. Starting from the development of FIM via Chebyshev expansion. This developed FIM is then utilized to handle nonlinear differential equations, that consist of the Burgers' equation with shock wave, the time-fractional BBMB equation and the Poisson equation over irregular domains.

6.1 Conclusions

In this research, we begin to develop the traditional FIM by applying the Chebyshev expansion in order to be applicable on arbitrary domains without any transformations as demonstrated in Chapter 2. We have then mentioned about the advantages of choosing Chebyshev expansion which interpolated by zeros of the Chebyshev polynomial of a certain degree. Subsequently, we have constructed the Chebyshev integration matrices both one- and two-dimensional regions which are matrix representations of the integral operator. For the one-dimensional Chebyshev integration matrix, we achieve the relationship that the number of integral layers is equal to the exponential number of Chebyshev integration matrix. For the two-dimensional Chebyshev integration matrices, we can express their relationships via the Kronecker product as followed by Remark 2.1 for integrating with respect to only one variable x or y and also Remark 2.2 for integrating with respect to both variables x and y.

In Chapter 3, we utilize our developed FIM-CPE to devise the numerical algorithm for solving one-dimensional nonlinear Burgers' equation with a shock wave (3.1) as demonstrated in Section 3.2. The presented Algorithm 1 can reduce the oscillations and instabilities for small viscosity ν that can observe from the graphical behavior of several numerical examples in Section 3.3. Moreover, we notice that Algorithm 1 can

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

overcome the problems that their analytical solutions involve an infinite series like Examples 3.1 and 3.2 which may converge very slowly for the small viscosity ν. Also, it can perform with the problems that their initial conditions contain a kinematic viscosity like Examples 3.3 and 3.4. Illustrative implementations for many experiments in Section 3.3 demonstrate that our Algorithm 1 outperforms the traditional FIM and other methods in terms of accuracy under the same parameters and conditions. Finally, we find the convergence order with respect to the time step τ of Algorithm 1 via Examples 3.1 - 3.4. As a result, the convergence order is linearly order or $\mathcal{O}(\tau)$. Afterward, we accelerate the convergence speed of Algorithm 1 by applying the Crank-Nicolson method which provides quadratically convergence order or $\mathcal{O}(\tau^2)$. We also show the convergence order of the accelerated procedure through the numerical experiments and plotting graphs which they indeed provide the quadratically convergence order.

In Chapter 4, we extend the idea of solving the Burgers' equation, that contains the first-order derivative with respect to time, in order to study the BBMB equation (4.1), that contains the fractional-order derivative with respect to time. Then, the numerical Algorithm 2 was created for finding an approximate solution of the time-fractional BBMB equation as displayed in Section 4.3. The definition of fractional-order derivative is used in Caputo sense. The numerical examples demonstrate that our proposed Algorithm 2 produces a much higher accuracy than the CNLDS under the same parameters and conditions for varying the discretized numbers of nodal points M, see Example 4.1. We also notice from Examples 4.1, 4.2 and 4.3 that they provide more accuracy even when we use a small number of nodal points M. Evidently, when we decrease the time step Δt, they furnish together more accurate results. In addition, our Algorithm 2 gives a good performance on the fractional-order derivative $\alpha \in (0, 1)$ and actually it can be easily applied to other nonlinear fractional PDEs. Finally, we seek the order of convergence with respect to the time step τ of the proposed Algorithm 2 via all numerical Examples 4.1 - 4.3 together with the plotting graphs. As a consequence, the obtained convergence order is linearly order or $\mathcal{O}(\tau)$ and it is independent of the fractional order α. Also, we can accelerate this Algorithm 2 to obtain the time complexity $\mathcal{O}(\tau^2)$ by using the Crank-Nicolson method in the same way with Chapter 3.

In Chapter 5, we apply our two-dimensional developed FIM-CPE to construct the numerical Algorithm 3 for seeking an approximate solution of two-dimensional nonlinear Poisson-type equation (5.1) over various irregular domains, including the pentagonal, circular, L-shaped, butterfly, peanut-shaped and elliptic domains. We demonstrate an efficiency and effectiveness of our presented Algorithm 3 via several numerical examples in Section 5.3. Moreover, we also express the order and rate of convergences for each example. We can conclude that Algorithm 3 provide the linearly convergent order. The rate of convergence obtained from each example is unequal which is the effect of nonlinear forcing term f. From Theorem 5.2, we obviously know that λ is the convergence rate that corresponds to $\|\nabla\mathcal{G}(\mathbf{u})\| < \lambda$, but the iterative function $\mathcal{G}(\mathbf{u}) := \mathbf{H}^{-1}\mathcal{F}(\mathbf{u})$, where $\mathcal{F}$ is constructed from the nonlinear forcing term f in each example. Thus, we can imply that the rate of convergence depends on the nonlinear forcing term f. Especially, in the case of the forcing term f is very complicated and fully nonlinear, it slowly converges to the analytical solution. However, our Algorithm 3 provides the solution that converges to the analytical solution for all above examples. Examples 5.1 and 5.2 contain only one term of Laplace operator on the left-hand side. Examples 5.3 and 5.4 are in full form of (5.1) with constant and variable coefficients, respectively. After implementing each example by Algorithm 3, we can see that it gives a high accuracy of approximate result compared to the analytical solution, although our Algorithm 3 uses a large convergent tolerance TOL and it also rapidly converges to the solution by using a small number of iteration. Moreover, we implement Examples 5.5 and 5.6 which are considered over the complicated domains in polar coordinate form. They provide a much higher accuracy than other methods. We further show that a fully nonlinear Poisson-type equation can be performed by Algorithm 3 and it still produces an accurate result. In fact, our procedure of Algorithm 3 can be performed not only the presented Poisson-type equation (5.1) with nonlinear forcing term, but it is also can carry out in the more general nonlinear Poisson-type equation as shown in Example 5.7. However, we notice from these examples that sometimes we cannot choose the same number of horizontal and vertical discretizations, simultaneously, like Examples 5.2 and 5.4. This is because there may be some computational points on the boundary that happen to be the same.

6.2 Future works

In this section, we provide the plan of future works that is the improvement and modification of our developed FIM by using Chebyshev polynomial expansion to apply with numerous interesting problems to other linear and nonlinear differential equations. The lists of our future plan include the followings:

- Extend our developed FIM-CPE to the multi-dimensional domains and find its general forms of Chebyshev integration matrices.
- Find the theoretical analysis of our proposed algorithms such as error estimation, stability, rate of convergence and order of convergence.
- Adapt our proposed FIM-CPE to handle the problem that has a random variable such as stochastic differential equations.
- Apply these presented algorithms for solving other nonlinear PDEs with some other boundary conditions such as Neumann and Robin.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

REFERENCES

[1] S. Abbasbandy and A. Shirzadi. The first integral method for modified Benjamin-Bona-Mahony equation. *Commun. Nonlinear. Sci. Numer. Simul.*, 15(7):1759–1764, 2010.

[2] A. Agila, D. Baleanu, R. Eid, and B. Irfanoglu. Applications of the extended fractional Euler-Lagrange equations model to freely oscillating dynamical systems. *Rom. J. Phys.*, 61(3):350–359, 2016.

[3] H. Arzani and M. H. Afshar. Solving Poisson's equations by the discrete least square meshless method. *WIT Trans. Model. Sim.*, 42:23–32, 2006.

[4] A. Asaithambi. Numerical solution of the Burgers' equation by automatic differentiation. *Appl. Math. Comput.*, 216(9):2700–2708, 2010.

[5] E. Ashpazzadeh, B. Han, and M. Lakestani. Biorthogonal multiwavelets on the interval for numerical solutions of Burgers' equation. *J. Comput. Appl. Math.*, 317:510–534, 2017.

[6] K. Atkinson and O. Hansen. Solving the nonlinear Poisson equation on the unit disk. *J. Integral Equ. Appl.*, 17(3):223–241, 2005.

[7] H. Bateman. Some recent researches on the motion of fluids. *Mon. Weather Rev.*, 43:163–170, 1915.

[8] O. A. Bauchau. *Dymore User's Manual Chebyshev Polynomials.* Georgia Institute of Technology, Atlanta, USA, 2007.

[9] T. B. Benjamin, J. L. Bona, and J. J. Mahony. Model equation for long waves in nonlinear dispersive systems. *Philos. Trans. R. Soc. Lond. A*, 272(1220):47–78, 1972.

[10] E. R. Benton and G. W. Platzman. A table of solutions of the one-dimensional Burgers' equations. *Q. Appl. Math.*, 30(2):195–212, 1972.

[11] M. P. Bonkile, A. Awasthi, C. Lakshmi, V. Mukundan, and V. S. Aswin. A systematic literature review of Burgers' equation with recent advances. *Pramana J. Phys.*, 90(69): 1–21, 2018.

[12] 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.

Chulalongkorn University

[13] R. Boonklurb, A. Duangpan, and A. Saengsiritongchai. Finite integration method via Chebyshev polynomial expansion for solving 2-D linear time-dependent and linear space-fractional differential equations. *Thai J. Math. (AMM 2019)*, pages 103–131, 2020.

[14] 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.

[15] J. M. Burgers. A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.*, 1:171–199, 1948.

[16] J. D. Cole. On a quasi-linear parabolic equations occurring in aerodynamics. *Q. Appl. Math.*, 9(3):225–236, 1951.

[17] A. Dogan. A Galerkin finite element approach to Burgers' equation. *Appl. Math. Comput.*, 157(2):331–346, 2004.

[18] A. Esen and O. Tasbozan. Numerical solution of time fractional Burgers equation. *Acta Univ. Sapientiae Math.*, 7(2):167–185, 2015.

[19] C.-E. Fröberg. *Numerical Mathematics, Theory and Computer Applications.* Benjamin & Cummings, Inc., California, USA, 1985.

[20] C. G. Gal, C. Gal, and M. Warma. Fractional-in-time semilinear parabolic equations and applications, 2019. `https://hal.archives-ouvertes.fr/hal-02061144`.

[21] I. Ganaie and V. Kukreja. Numerical solution of Burgers' equation by cubic Hermite collocation method. *Appl. Math. Comput.*, 237(15):571–581, 2014.

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

[23] S. Harris. Sonic shocks governed by the modified Burgers' equation. *Eur. J. Appl. Math.*, 7(2):201–222, 1996.

[24] W. L. Hosch. *Navier-Stokes Equation.* Encyclopædia Britannica, Inc., Scotland, USA, 2018.

[25] X. Hu, L. Mu, and X. Ye. A simple finite element method of the Cauchy problem for Poisson equation. *Int. J. Numer. Anal. Mod.*, 14(4–5):591–603, 2017.

[26] J. D. Jackson. *Classical Electrodynamics.* John Wiley & Sons, Inc., New York, USA, 1999.

[27] K. Kakuda and T. Nobuyoshi. The generalized boundary element approach to Burgers' equation. *Int. J. Numer. Method Eng.*, 29(2):245–261, 1990.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

[28] J. J. Kasab, S. R. Karur, and P. A. Ramachandran. Quasilinear boundary element method for nonlinear Poisson type problems. *Eng. Anal. Bound. Elem.*, 15(3):277–282, 1995.

[29] N. A. Khan, A. Ara, and A. Mahmood. Numerical solutions of time-fractional Burger equations: a comparison between generalized differential transformation technique and homotopy perturbation method. *Int. J. Numer. Method H.*, 22(2):175–193, 2015.

[30] C. I. Kondo and C. M. Webler. The generalized BBMB equations: convergence results for conservation law with discontinuous flux function. *Appl. Anal.*, 95(3):503–523, 2016.

[31] W. Kong and X. Wu. Chebyshev tau matrix method for Poisson-type equations in irregular domain. *J. Comput. Appl. Math.*, 228(1):158–167, 2009.

[32] D. Kumar, J. Singh, and D. Baleanu. A fractional model of convective radial fins with temperature-dependent thermal conductivity. *Rom. J. Phys.*, 69(103):1–13, 2017.

[33] S. Kumar and D. Kumar. Fractional modelling for BBMB equation by using new homotopy analysis transform method. *J. Assoc. Arab Univ. Basic Appl. Sci.*, 16(1):16–20, 2014.

[34] S. Kutluay, A. Esen, and I. Dag. Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline FEM. *J. Comput. Appl. Math.*, 167(1):21–33, 2004.

[35] N. Laskin. *Fractional Dynamics: Principles of Fractional Quantum Mechanics.* TopQuark Inc., Toronto, Canada, 2011.

[36] C. P. Li and Y. H. Wang. Numerical algorithm based on Adomian decomposition for fractional differential equations. *Comput. Math. Appl.*, 57(10):1672–1681, 2009.

[37] M. Li, C. S. Chen, Y. C. Hon, and P. H. Wen. Finite integration method for solving multi-dimensional partial differential equations. *Appl. Math. Model.*, 39(17):4979–4994, 2015.

[38] M. Li, Y. C. Hon, T. Korakianitis, and P. H. Wen. Finite integration method for nonlocal elastic bar under static and dynamic loads. *Eng. Anal. Bound. Elem.*, 37(5):842–849, 2013.

[39] M. Li, Z. L. Tian, Y. C. Hon, C. S. Chen, and P. H. Wen. Improved finite integration method for partial differential equations. *Eng. Anal. Bound. Elem.*, 64:230–236, 2016.

[40] Y. Li, M. Li, and Y. C. Hon. Improved finite integration method for multi-dimensional nonlinear Burgers' equation with shock wave. *Neur. Par. Sci. Comput.*, 23:63–86, 2015.

[41] S. Liu, J. Li, L. Chen, Y. Guan, C. Zhang, F. Gao, and J. Lin. Solving 2D Poisson-type equations using meshless SPH method. *Results Phys.*, 13:1–8, 2019.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

[42] J. D. Logan. *An Introduction to Nonlinear Partial Differential Equations, Second Edition.* Wiley & Sons, Inc., New York, USA, 2008.

[43] J. C. Mason and D. C. Handscomb. *Chebyshev Polynomials.* Chapman and Hall/CRC, New York, USA, 2002.

[44] R. Metzler and J. Klafter. The random walk's guide to anomalous diffusion: a fractional dynamics approach. *Phys. Rep.*, 339(1):1–77, 2000.

[45] E. L. Miller. *Predictor-Corrector Studies of Burgers' Model of Turbulent Flow.* University of Delaware, Newark, DE, USA, 1966.

[46] R. C. Mittal and R. K. Jain. Numerical solutions of nonlinear Burgers' equation with modified cubic b-splines collocation method. *Appl. Math. Comput.*, 218(15):7839–7855, 2012.

[47] R. C. Mittal and P. Singhal. Numerical solution of Burger's equation. *Commun. Numer. Method Eng.*, 9(5):397–406, 1993.

[48] J. R. Nagel. Numerical solutions to Poisson equations using the finite-difference method. *IEEE Antenn. Propag. Mag.*, 56(4):209–224, 2014.

[49] H. Nguyen and J. Reynen. A space-time finite element approach to Burgers' equation. *Numer. Method Nonlinear Probl.*, 2:718–728, 1982.

[50] K. B. Oldham and J. Spanier. *The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order.* Academic Press, Inc., New York, 1974.

[51] I. Podlubny. *Fractional Differential Equations.* Academic Press, San Diego, USA, 1998.

[52] E. Poisson and C. M. Will. *Gravity: Newtonian, Post-Newtonian, Relativistic.* Cambridge University Press, UK, 2014.

[53] K. Rahman, N. Helil, and R. Yimin. Some new semi-implicit finite difference schemes for numerical solution of Burgers' equation. *IEEE (ICCASM 2010)*, pages 451–455, 2010.

[54] S. Y. Reutskiy. Method of particular solutions for nonlinear Poisson-type equations in irregular domains. *Eng. Anal. Bound. Elem.*, 37(2):401–408, 2013.

[55] T. J. Rivlin. *Chebyshev Polynomials: from Approximation Theory to Algebra and Number Theory.* Dover Publications, Inc., New York, USA, 2020.

[56] A. P. S. Selvadurai. *Partial Differential Equations in Mechanics 2.* Springer, Berlin, Heidelberg, Germany, 2000.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

[57] X. Shen and A. Zhu. A Crank-Nicolson linear difference scheme for a BBM equation with a time fractional nonlocal viscous term. *Adv. Differ. Equ.*, 2018(351):1–12, 2018.

[58] N. Su, J. P. C. Watt, K. W. Vincent, M. E. Close, and R. Mao. Analysis of turbulent flow patterns of soil water under field conditions using Burgers equation and porous suction-cup samplers. *Aust. J. Soil Res.*, 42(1):9–16, 2004.

[59] A. Tri, H. Zahrouni, and M. P. Ferry. Perturbation technique and method of fundamental solution to solve nonlinear Poisson problem. *Eng. Anal. Bound. Elem.*, 35(3):273–278, 2011.

[60] C. C. Tsai. Homotopy method of fundamental solutions for solving certain nonlinear partial differential equations. *Eng. Anal. Bound. Elem.*, 36(8):1226–1234, 2012.

[61] C. C. Tsai, C. S. Liu, and W. C. Yeih. Fictitious, time integration method of fundamental solutions with Chebyshev polynomials for solving Poisson-type nonlinear PDEs. *Comput. Model. Eng. Sci.*, 56(2):131–151, 2010.

[62] E. Varoḡlu and W. D. L. Finn. Space-time finite elements incorporating characteristics for the Burgers' equation. *Int. J. Numer. Method Eng.*, 16(1):171–184, 1980.

[63] P. H. Wen, Y. C. Hon, M. Li, and T. Korakianitis. Finite integration method for partial differential equations. *Appl. Math. Model.*, 37(24):10092–10106, 2013.

[64] W. L. Wood. An exact solution for Burgers' equation. *Commun. Numer. Meth. Eng.*, 22(7): 797–798, 2006.

[65] M. Xu, R. H. Wang, J. H. Zhang, and Q. Fang. A novel numerical scheme for solving Burgers' equation. *Appl. Math. Comput.*, 217(9):4473–4482, 2011.

[66] X. J. Yang, F. Gao, and H. M. Srivastava. New rheological models within local fractional derivative. *Rom. J. Phys.*, 69(3):1–12, 2017.

[67] S. Yildirim. Exact and numerical solutions of Poisson equation for electrostatic potential problems. *Math. Probl. Eng.*, 2008(2):1–11, 2008.

[68] Y. Yu, D. Xu, and Y. C. Hon. Reconstruction of inaccessible boundary value in a sideways parabolic problem with variable coefficients-forward collocation with finite integration method. *Eng. Anal. Bound. Elem.*, 61:78–90, 2015.

[69] M. R. Yulita, M. S. M. Nooran, and I. Hashim. Variational iteration method for fractional heat- and wave-like equations. *Nonlinear Anal. Real World Appl.*, 10(3):1854–1869, 2009.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

[70] D. F. Yun, Z. H. Wen, and Y. C. Hon. Adaptive least squares finite integration method for higher-dimensional singular perturbation problems with multiple boundary layers. *Appl. Math. Comput.*, 271:232–250, 2015.

[71] M. Zarebnia and R. Parvaz. Numerical study of benjamin-bona-mahony-burgers equation. *Bol. Soc. Parana. Mat.*, 35(1):127–138, 2017.

[72] H. Zhang and F. Ding. On the kronecker products and their applications. *J. Appl. Math.*, 2013:1–8, 2013.

[73] J.-R. E. Zhang, Z. Wei, L. Yong, and Y. Xiao. Analytical solution for the time fractional BBMB equation using modified residual power series method. *Complexity*, 2018:1–11, 2018.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

APPENDICES

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

APPENDIX A : Some example of MatLab code for solving one-dimensional nonlinear Burgers' equation with shock wave. In this appendix, we demonstrate the MatLab code of Example 3.1 based on the proposed numerical Algorithm 1. The command for solving system of linear equations, we use the backslash command in MatLab solver.

```
%% -- Set parameters ------------------------------------------------
M = 80;                        % number of grid points
a = 0;                         % lower boundary
b = 1;                         % upper boundary
v = 0.01;                      % kinematic viscosity
x = 0.25;                      % a point that solution is sought
T = 0.4;                       % terminal time
dt = 10^-4;                    % time steps
u0 = @(x) sin(pi*x);           % initial condition
ua = @(t) 0;                   % left boundary condition
ub = @(t) 0;                   % right boundary condition
%% -- Chebyshev integration matrix A --------------------------------
xk = flip(1/2*((b-a)*cos((2*(1:M)'-1)/(2*M)*pi)+a+b));
R(:,1) = ones(M,1);
R(:,2) = (2*xk-a-b)/(b-a);
for n = 2:M
    R(:,n+1) = 2*(2*xk-a-b)/(b-a).*R(:,n)-R(:,n-1);
end
Rbar(:,1) = xk-a;
Rbar(:,2) = (xk-a).*(xk-b)/(b-a);
for n = 2:M-1
    Rbar(:,n+1) = (b-a)/4*(R(:,n+2)/(n+1)-R(:,n)/(n-1)-2*(-1)^n/(n^2-1))
end
Rinv = 1/M*diag([1,repmat(2,1,M-1)])*R(:,1:M)';
A = Rbar*Rinv;
%% -- Boundary Conditions -------------------------------------------
hl = (-1).^(0:M-1);
hr = ones(1,M);
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

```
%% -- Construct matrix R' ------------------------------------------
n = repmat(0:M-1,M,1);
y = repmat((2*xk-a-b)/(b-a),1,M);
Rdif = 2/(b-a)*n.*sin(n.*acos(y))./sqrt(1-y.^2);
%% -- Approximate solution um --------------------------------------
m = 1;                               % set initial iteration m
u = u0(xk); t(1) = dt;               % initial solution and time
while t(m) <= T
    Q = A*diag(u)-A^2*diag(Rdif*Rinv*u);
    B11 = A^2/dt+Q-v*eye(M);
    B12 = [-xk -ones(M,1)];
    B21 = [hl*Rinv; hr*Rinv];
    B22 = zeros(2);
    B = [B11 B12; B21 B22];
    f = [A^2*u/dt; ua(t(m)); ub(t(m))];
    U = pinv(B)*f;
    u = U(1:M);                      % solution u at each time t(m)
    m = m + 1;                       % update iteration m
    t(m) = m*dt;                     % compute consecutive time t(m)
end
Rx = cos((0:M-1)*acos((2*x-a-b)/(b-a)));
um = Rx*Rinv*u                       % approximate solution u(x,T)
%% -- Exact solution ue --------------------------------------------
N = 1:10^6;                          % number of terms in summation
F = @(x,n) exp((cos(pi*x)-1)/(2*pi*v)).*cos(n*pi*x);
a0 = integral(@(x)F(x,0),0,1);
for n = 1:length(N)
    an(n) = 2*integral(@(x)F(x,n),0,1);
end
s1 = sum(an.*exp(-N.^2*pi^2*v*T).*N.*sin(N*pi*x));
s2 = sum(an.*exp(-N.^2*pi^2*v*T).*cos(N*pi*x));
ue = 2*pi*v*s1/(a0+s2)               % exact solution u(x,T)
er = abs(ue-um)                      % absolute error
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

APPENDIX B : Some example of MatLab code for solving one-dimensional nonlinear BBMB equation. In this appendix, we illustrate the MatLab code of Example 4.2 based on the presented numerical Algorithm 2. The command that uses to solve system of linear equations, we easily choose the backslash command in MatLab software.

```
%% -- Set parameters ------------------------------------------------
M  = 40;                       % number of grid points
L  = 1;                        % upper boundary domain
T  = 1;                        % terminal time T
x = 0.5;                       % a point that solution is sought
a = 0.9;                       % order of fractional time alpha
dt = 0.01;                     % time step
u0 = @(x) 0*x;                 % initial condition
ul = @(t) t^2;                 % left boundary condition
ur = @(t) exp(1)*t^2;          % right boundary condition
f  = @(x,t) 2*exp(x)*t^(2-a)/gamma(3-a)+t*exp(x).*(exp(x)*t^3+t-2);
%% -- Chebyshev integration matrix A --------------------------------
xk  = flip(L/2*(cos((2*(1:M)'-1)/(2*M)*pi)+1));
R(:,1) = ones(M,1);
R(:,2) = 2*xk/L-1;
for n = 2:M
    R(:,n+1) = 2*(2*xk/L-1).*R(:,n)-R(:,n-1);
end
Rbar(:,1) = xk;
Rbar(:,2) = xk.^2/L-xk;
for n = 2:M-1
    Rbar(:,n+1) = L/4*(R(:,n+2)/(n+1)-R(:,n)/(n-1)-2*(-1)^n/(n^2-1));
end
Rinv = 1/M*diag([1,repmat(2,1,M-1)])*R(:,1:M)';
A = Rbar*Rinv;
%% -- Boundary Conditions -------------------------------------------
hl = (-1).^(0:M-1);
hr = ones(1,M);
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

```
%% -- Construct matrix R' -------------------------------------------
n = repmat(0:M-1,M,1);
y = repmat(2*xk/L-1,1,M);
Rdif = 2/L*n.*sin(n.*acos(y))./sqrt(1-y.^2);
%% -- Approximate solution um ---------------------------------------
m = 1;                             % set initial iteration m
t(1) = dt;                         % starting time t(1)
u(:,1) = u0(xk);                   % initial solution u0
w0 = dt^(-a)/gamma(2-a);
while t(m) <= T
    s = 0;
    for j = 1:m-1
        w = dt^-a/gamma(2-a)*((j+1)^(1-a)-j^(1-a));
        s = s + w*A^2*(u(:,m-j+1)-u(:,m-j));
    end
    K1 = w0*A^2-eye(M)/dt+A+A*diag(u(:,m))-A^2*diag(Rdif*Rinv*u(:,m));
    K2 = [xk ones(M,1)];
    K3 = [hl*Rinv; hr*Rinv];
    K4 = zeros(2);
    K = [K1 K2; K3 K4];
    F = [A^2*f(xk,t(m))-s+(w0*A^2-eye(M)/dt)*u(:,m);ul(t(m));ur(t(m))];
    U = pinv(K)*F;
    u(:,m+1) = U(1:M);             % solution u at each time t(m)
    m = m + 1;                     % update iteration m
    t(m) = m*dt;                   % compute consecutive time t(m)
end
Rx = cos((0:M-1)*acos(2*x/L-1));   % calculate vector R(x)
um = Rx*Rinv*u(:,end);             % approximate solution u(x,T)
ue = @(x,t) t^2*exp(x);            % analytical solution u(x,T)
er = abs(ue(T,x)-um)               % absolute error
plot(xk,u(:,end),'o')              % plot approximate solution
hold on
plot(xk,ue(xk,T))                  % plot analytical solution
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

APPENDIX C : Some example of MatLab code for solving two-dimensional nonlinear Poisson-type equation over irregular domain. In this appendix, we show the MatLab code of Example 5.1 based on the numerical Algorithm 3. The system of linear equations is easily solved by using the backslash command in MatLab software.

```
function Example_51
%% -- Initial inputs ------------------------------------------------
M = 14;                                 % grid numbers in x-direction
N = 14;                                 % grid numbers in y-direction
H = M*N;                                % total numbers of grid points
a = -1; b = 1;                          % left and right boundaries
c = -1; d = 1;                          % bottom and top boundaries
x = 0; y = 0;                           % x and y that result is sought
u0 = zeros(H,1);                        % initial guess solution
TOL = 10^-5;                            % convergent tolerance
e = @(x,y) exp(x).*cos(y);              % exact solution
f = @(x,y,u) exp(2*x).*cos(y).^2-u.^2;  % forcing term
%% -- Construct matrices A and Rinv ---------------------------------
function [A,Rinv] = CIM(a,b,x,n)
    R(:,1) = ones(n,1);
    R(:,2) = (2*x-a-b)/(b-a);
    for r = 2:n
        R(:,r+1)=2*(2*x-a-b)/(b-a).*R(:,r)-R(:,r-1);
    end
    Rb(:,1) = x-a;
    Rb(:,2) = (x-a).*(x-b)/(b-a);
    for r = 2:n-1
        Rb(:,r+1)=(b-a)/4*(R(:,r+2)/(r+1)-R(:,r)/(r-1)-2*(-1)^r/(r^2-1))
    end
    Rinv = 1/n*diag([1,repmat(2,1,n-1)])*R(:,1:n)';
    A = Rb*Rinv;
end
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

```
%% -- Set parameters -------------------------------------------------
xk = flip(1/2*((b-a)*cos((2*(1:M)'-1)/(2*M)*pi)+a+b));
yh = flip(1/2*((d-c)*cos((2*(1:N)'-1)/(2*N)*pi)+c+d));
xg = kron(ones(N,1),xk);              % all grid points in x-axis
yg = kron(yh,ones(M,1));              % all grid points in x-axis
[AM,RinvM] = CIM(a,b,xk,M);
[AN,RinvN] = CIM(c,d,yh,N);
Ax = kron(eye(N),AM);                 % Chebyshev integration matrix Ax
Ay = kron(AN,eye(M));                 % Chebyshev integration matrix Ax
X  = diag(xg);
Y  = diag(yg);
Dgx = repmat(0:M-1,H,1);              % degrees in matrix Phix
Dgy = repmat(0:N-1,H,1);              % degrees in matrix Phiy
Ndx = repmat((2*xg-a-b)/(b-a),1,M);   % nodal points in Phix
Ndy = repmat((2*yg-c-d)/(d-c),1,N);   % nodal points in Phiy
Phix = cos(Dgx.*acos(Ndx));           % matrix Phix
Phiy = cos(Dgy.*acos(Ndy));           % matrix Phiy
for k = 1:M
    for h = 1:N
        i = (h-1)*M+k;
        j = (k-1)*N+h;
        P(i,j) = 1;                   % permutation matrix P
    end
end
%% -- Boundary conditions --------------------------------------------
xl = [-yh(1:N/2)-1; yh(N/2+1:N)-1];   % nodes x on left boundary
xr = b*ones(N,1);                     % nodes x on right boundary
yb = [-xk(1:M/2)-1; c*ones(M/2,1)];   % nodes y on bottom boundary
yt = [xk(1:M/2)-1; d*ones(M/2,1)];    % nodes y on top boundary
gl = e(xl,yh);                        % left boundary condition
gr = e(xr,yh);                        % right boundary condition
gb = e(xk,yb);                        % bottom boundary condition
gt = e(xk,yt);                        % top boundary condition
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

```
Zl = []; Zr = []; Zb = []; Zt = [];
for i = 1:N
    Zl = blkdiag(Zl,cos((0:M-1)*acos((2*xl(i)-a-b)/(b-a)))*RinvM);
    Zr = blkdiag(Zr,cos((0:M-1)*acos((2*xr(i)-a-b)/(b-a)))*RinvM);
end
for i = 1:M
    Zb = blkdiag(Zb,cos((0:N-1)*acos((2*yb(i)-c-d)/(d-c)))*RinvN);
    Zt = blkdiag(Zt,cos((0:N-1)*acos((2*yt(i)-c-d)/(d-c)))*RinvN);
end
%% -- FIM-CPE -------------------------------------------------------
u(:,1) = u0;                          % initial guess solution
m = 1;                                % set initial iteration m
K1 = Ay^2+Ax^2;
K2 = [X*Phiy Phiy Y*Phix Phix];
K3 = [Zl; Zr; Zb*P'; Zt*P'];
K4 = zeros(2*(M+N));
K = [K1 K2; K3 K4];
F = [Ax^2*Ay^2*f(xg,yg,u(:,1)); gl; gr; gb; gt];
U = pinv(K)*F;
u(:,2) = U(1:H);                      % solution u at first iteration
while norm(u(:,m+1)-u(:,m))>TOL       % verify the error norm
    F = [Ax^2*Ay^2*f(xg,yg,u(:,m+1)); gl; gr; gb; gt];
    U = pinv(K)*F;
    m = m + 1;                        % update iteration m
    u(:,m+1) = U(1:H);                % solution u at mth iteration
end
zN = cos((0:N-1)*acos((2*y-c-d)/(d-c)))*RinvN;
zM = cos((0:M-1)*acos((2*x-a-b)/(b-a)))*RinvM;
ue = e(x,y)                           % analytical solution u(x,y)
ua = kron(zN,zM)*u(:,end);            % approximate solution u(x,y)
Er = abs(ue-ua)                       % absolute error
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

BIOGRAPHY

Name	Mr. Ampol Duangpan
Date of Birth	August 6, 1992
Place of Birth	Ranong, Thailand
Educations	B.Sc. (Applied Mathematics) (First Class Honours), King Mongkut Institute of Technology Ladkrabang, 2013 M.Sc. (Applied Mathematics and Computational Science), Chulalongkorn University, 2016
Scholarships	The 100th Anniversary Chulalongkorn University Fund for Doctoral Scholarship

Publications

- 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*, vol. 12, no. 3–4, pp. 1–19, 2018.
- A. Duangpan, R. Boonklurb and T. Treeyaprasert, Finite integration method with shifted Chebyshev polynomials for solving time-fractional Burgers' equations, *Mathematics*, vol. 7, no. 12, pp. 1–24, 2019.
- 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*, vol. 12, no. 4, pp. 1–19, 2020.
- R. Boonklurb, A. Duangpan and A. Saengsiritongchai, Finite integration method via Chebyshev polynomial expansion for solving 2-D linear time-dependent and linear space-fractional differential equations, *Thai Journal of Mathematics*, pp. 103–131, 2020.
- A. Duangpan and R. Boonklurb, Finite integration method using Chebyshev expansion for solving nonlinear Poisson equations on irregular domains, *Journal of Numerical Analysis, Industrial and Applied Mathematics*, vol. 14, no. 1–2, pp. 7–24, 2020.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY