

ผลเฉลยเชิงตัวเลขสำหรับสมการน้ำเต้าในหนึ่งและสองมิติบนพื้นฐานของระเบียบวิธี
ปริพันธ์อันดับร่วมกับการกระจายเซบีเซฟ



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2565

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

NUMERICAL SOLUTIONS FOR ONE- AND TWO-DIMENSIONAL SHALLOW
WATER EQUATIONS BASED ON FINITE INTEGRATION METHOD WITH
CHEBYSHEV EXPANSION

Miss Lalita Apisornpanich



จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Applied Mathematics and

Computational Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2022

Copyright of Chulalongkorn University

Thesis Title NUMERICAL SOLUTIONS FOR ONE- AND TWO-DIMENSIONAL
 SHALLOW WATER EQUATIONS BASED ON FINITE INTEGRA-
 TION METHOD WITH CHEBYSHEV EXPANSION

By Miss Lalita Apisornpanich

Field of Study Applied Mathematics and Computational Science

Thesis Advisor Associate Professor Ratinan Boonklurb, Ph.D.

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

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

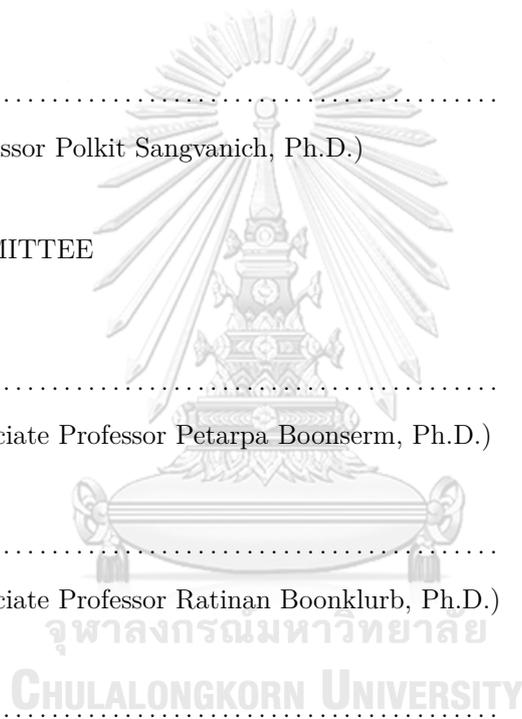
THESIS COMMITTEE

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

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

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

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



ลลิตา อภิสรพาณิชย์ : ผลเฉลยเชิงตัวเลขสำหรับสมการน้ำตื้นในหนึ่งและสองมิติบนพื้นฐานของระเบียบวิธีปริพันธ์อันตะร่วมกับการกระจายเซบีเชฟ. (NUMERICAL SOLUTIONS FOR ONE- AND TWO-DIMENSIONAL SHALLOW WATER EQUATIONS BASED ON FINITE INTEGRATION METHOD WITH CHEBYSHEV EXPANSION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร.รตินันท์ บุญเคลือบ, 40 หน้า.

มีปัญหาในโลกจริงหลายอย่างที่สามารถอธิบายโดยสมการน้ำตื้น เช่น พฤติกรรมของน้ำบริเวณชายหาด พฤติกรรมของน้ำเมื่อเขื่อนแตก หรือการเกิดสึนามิ สมการน้ำตื้นเป็นสมการเชิงอนุพันธ์ย่อยไม่เชิงเส้นที่ยากจะหาผลเฉลยแม่นยำตรง ในงานนี้จะใช้วิธีปริพันธ์อันตะร่วมกับการกระจายเซบีเชฟมาหาค่าประมาณผลเฉลยของสมการน้ำตื้น โดยวิธีการนี้มีพื้นฐานมาจากการประมาณค่าปริพันธ์ของพจน์นั้นและแทนค่าไปในสมการอินทิกรัลที่เทียบเท่ากับสมการที่กำลังพิจารณา เนื่องจากสมการน้ำตื้นไม่ได้มีแค่ตัวแปรพื้นที่แต่ยังมีตัวแปรเวลาอีกด้วย ดังนั้นเราจึงใช้ผลต่างข้างหน้าเพื่อประมาณค่าของพจน์ที่มีอนุพันธ์เทียบเวลา และใช้วิธีปรับให้เป็นเชิงเส้นเพื่อจัดการกับพจน์ที่ไม่เชิงเส้น เพื่อที่จะตรวจสอบประสิทธิภาพและความแม่นยำของระเบียบวิธีการของเราจึงได้มีการนำตัวอย่างที่หลากหลายมาทดสอบ กล่าวคือ ลำธารนิ่ง เขื่อนแตก และพื่นนูนแบบเกาส์ในปัญหา 1 มิติ และลำธารนิ่ง และพื่นนูนแบบเกาส์ในปัญหา 2 มิติ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

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

6370023023 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE
 KEYWORDS : FINITE INTEGRATION METHOD / CHEBYSHEV EXPANSION / DAM
 BREAK / SHALLOW WATER EQUATION

LALITA APISORNPANICH : NUMERICAL SOLUTIONS FOR ONE- AND TWO-
 DIMENSIONAL SHALLOW WATER EQUATIONS BASED ON FINITE INTEGRA-
 TION METHOD WITH CHEBYSHEV EXPANSION. ADVISOR : ASSOC. PROF. RATI-
 NAN BOONKLURB, PH.D., 40 pp.

There are several real-world problems that can be described by their behavior by the shallow water equations (SWEs) such as shallow beaches, dam break, tsunami, etc. The SWE is one of the nonlinear partial differential equations (PDEs) which is difficult to solve analytically. In this work, we apply the finite integration method via Chebyshev polynomial expansion to find the approximate solution to SWEs. This method is based on approximating the integral term of the equivalent integral equation which is transformed from the given PDEs. The SWEs do not only contain the space variables but also consist of the time variables. We should present the forward difference quotient to estimate the temporal derivative term and employ the linearization method for manipulating the nonlinear terms. To examine the effectiveness and accuracy of our obtained algorithm, we provide various examples, including a stationary lake, a dam break and a Gaussian pulse in one-dimensional problems and a stationary lake and a Gaussian pulse in two-dimensional problems.

Department : .. Mathematics and Student's Signature

..... Computer Science Advisor's Signature

Field of Study : .. Applied Mathematics and ..

..... Computational Science

Academic Year : .. 2022

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor, Associate Professor Ratinan Boonklurb, Ph.D. for his invaluable guidance, encouragement and support throughout the entire research process. His insightful feedback and expertise was instrumental in shaping this thesis and helping me to achieve my academic goals. I am also grateful for the trust and freedom they gave me to pursue my research interests without any undue pressure.

I would also like to thank my senior, Ampol Duangpan, Ph.D. for his unwavering support and generosity. His vast knowledge and experience in the field was an invaluable resource for me and his kindness and willingness to share his time and expertise have been crucial in making this thesis possible.

I would like to express my sincere gratitude to the thesis exam committee members, Associate Professor Dr. Petarpa Boonserm, Associate Professor Dr. Khamron Mekchay and Associate Professor Dr. Tawikan Treeyaprasert, for their time and valuable feedback in evaluating my thesis.

Finally, I want to thank my family for their love, patience, and understanding. Their unwavering belief in me, their constant encouragement, and their positive attitude have been an immense source of strength and motivation throughout my academic journey.

CONTENTS

	Page
ABSTRACT IN THAI	iv
ABSTRACT IN ENGLISH	v
ACKNOWLEDGEMENTS	vi
CONTENTS	vii
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER	
1 INTRODUCTION	1
2 BACKGROUND KNOWLEDGE	3
2.1 Shallow Water Equation	3
2.1.1 The One-Dimensional Shallow Water Equation	3
2.1.2 The Two-Dimensional Shallow Water Equation	4
2.2 Chebyshev Expansion	4
2.3 Kronecker Product	6
2.4 Developed FIM-CPE	7
2.4.1 One-Dimensional Chebyshev Integration Matrices	7
2.4.2 Two-Dimensional Chebyshev Integration Matrices	8
3 NUMERICAL ALGORITHM FOR ONE-DIMENSIONAL SWES	11
3.1 Numerical Algorithm	11
3.2 Numerical Simulations	16
4 NUMERICAL ALGORITHM FOR TWO-DIMENSIONAL SWES	23
4.1 Numerical Algorithm	23
4.2 Numerical simulations	32
5 CONCLUSIONS AND FUTURE WORK	36
5.1 Conclusions	36
5.2 Future Work	37
REFERENCES	38
BIOGRAPHY	40

LIST OF TABLES

Table	Page
3.1 Comparisons of exact and numerical solutions at $T = 0.1$	19



LIST OF FIGURES

Figure	Page
2.1 Physical variables for one-dimensional shallow water model	4
2.2 Physical variables for two-dimensional shallow water model	5
2.3 The indices of the grid points globally and locally	8
3.1 The flowchart for solving one-dimension SWEs	15
3.2 Water height at $T = 10$ s.	17
3.3 Graphical solutions with $M = 200$ at time $T = 0.1$	19
3.4 Graphical solutions with $M = 100$ at different times T	21
3.5 Water height $h(x, t)$ at various times(s).	22
4.1 The flowchart for solving two-dimension SWEs	32
4.2 Analytical solutions	34
4.3 Water height $h(x, t)$ at various times(s).	35

CHAPTER I

INTRODUCTION

The shallow water equations (SWEs), also known as Saint-Venant's equations, are a mathematical model that helps us understand how water moves in shallow areas, like rivers and beaches. They are used to simulate the behavior of water during events like dam breaks, tsunamis, floods and so on, see more details in [4] and [8]. The flows are also depending on the bottom topography. The study of SWEs allows us to analyze how the bottom topography affects the behavior of water flow.

The shallow water model is based on the assumption that the horizontal distance over which water flows is much greater than the depth of the water. This allows us to simplify the equations that describe the flow of water by averaging the mass and momentum conservation equations over the depth and disposing of one of the vertical dimensions. The SWEs are a set of partial differential equations (PDEs) that describe the behavior of shallow water in terms of its height and velocity. They are derived from the laws of conservation of mass and momentum.

Since the SWEs are a nonlinear hyperbolic system of conservation equations with a source term due to topography. The resulting partial differential equations are challenging to solve analytically, and therefore, numerical methods are commonly used to simulate water flow. There are many numerical schemes to obtain an approximate solution of the SWEs such as the adaptive finite difference method (FDM) by Hudson [5], summation-by-parts operators with simultaneous approximation terms (SBP-SAT) by Lundgren [8], the forward time centered space by Crowhurst and Li [2], A well-balanced finite volume method with weighted average flux (WAF) by [6], etc. It can be seen that most methods actually employ a differential approximation. We have known that it is quite sensitive to round-off errors with a very small step size. To deal with this issue, in this paper, we

present the finite integration method with Chebyshev polynomial expansion (FIM-CPE) in order to design a genuinely high-order accurate algorithm for finding numerical solutions instead. The FIM-CPE approach is a technique used to approximate the solutions of partial differential equations (PDEs). In this approach, the PDEs are transformed into equivalent integral equations. Then, the integral term is approximated using Chebyshev expansion, which is a method of representing a function as a sum of Chebyshev polynomials.

The goal of developing a numerical algorithm for solving the one- and two-dimensional SWEs using the FIM-CPE is to accurately simulate the behavior of shallow water in different situations. The proposed algorithm has been validated by comparing several examples over different rigid bottoms. These tests have demonstrated that the proposed algorithm is effective at capturing the resolution for both smooth and discontinuous solutions.



CHAPTER II

BACKGROUND KNOWLEDGE

2.1 Shallow Water Equation

The shallow water equation has two parts. The first part, called the continuity equation which is derived from the conservation of mass, describes how the water height changes over time. It takes into account the flow of water in and out of a particular area and how it affects the water height. The second part, called the momentum equation which is derived from the conservation of momentum principles, describes how the water moves horizontally. It takes into account the forces acting on the water, such as gravity and pressure gradients, that affect its velocity. We assume that the equation is frictionless when all examples are conducted only on the wet bottom topography. Therefore, similar to [6] both one- and two-dimensional SWEs are obtained which are expressed in the form without friction term.

2.1.1 The One-Dimensional Shallow Water Equation

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) = 0, \quad (2.1)$$

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x} \left(hu^2 + \frac{gh^2}{2} \right) + gh \frac{\partial z}{\partial x} = 0, \quad (2.2)$$

where x denotes the space variable, t denotes the time variable, h is the water height, u is the horizontal velocity, g is the gravitational constant and z is the topography of the bottom, see Figure 2.1 for visualizing of the physical variables.

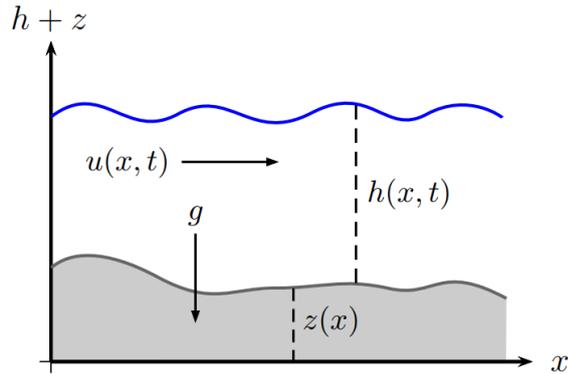


Figure 2.1: Physical variables for one-dimensional shallow water model

2.1.2 The Two-Dimensional Shallow Water Equation

The two-dimensional SWEs are expanded from one-dimensional by adding y which is a space variable along the y -direction and the involved variables.

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0, \quad (2.3)$$

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}\left(hu^2 + \frac{gh^2}{2}\right) + \frac{\partial}{\partial y}(huv) + gh\left(\frac{\partial z}{\partial x} + S_{fx}\right) = 0, \quad (2.4)$$

$$\frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial x}(huv) + \left(hv^2 + \frac{gh^2}{2}\right) + gh\left(\frac{\partial z}{\partial y} + S_{fy}\right) = 0, \quad (2.5)$$

where x and y denote the space variables in the x - and y -directions, respectively, u and v are the horizontal velocity in the x - and y -directions, respectively, see Figure 2.2 for visualizing of the physical variables.

2.2 Chebyshev Expansion

Definition 2.2.1 ([3]). The Chebyshev polynomial of degree $n \geq 0$ is defined by

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

The followings are properties of $R_n(x)$ which were proved in [3] and we will use these

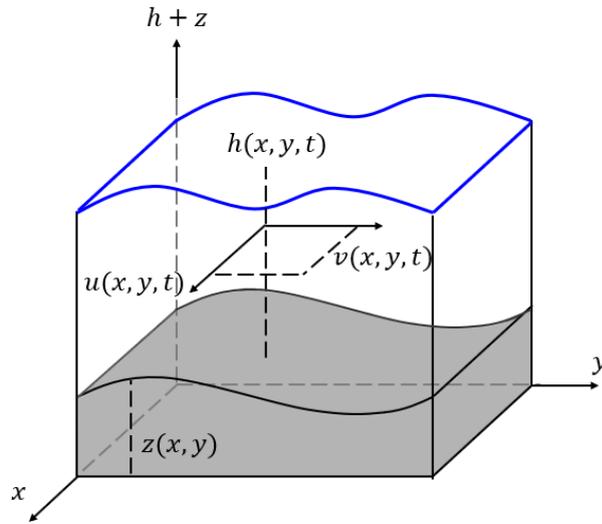


Figure 2.2: Physical variables for two-dimensional shallow water model

properties in this thesis.

Lemma 2.2.1. The followings are properties of the Chebyshev polynomials:

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

$$x_k = \frac{1}{2} \left((b-a) \cos \left(\frac{2k-1}{2n} \pi \right) + a+b \right), \quad k \in \{1, 2, 3, \dots, n\}. \quad (2.6)$$

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

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

(iii) The Chebyshev matrix \mathbf{R} at each zero x_k defined in (2.6) is defined 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 $\mathbf{R}^{-1} = \frac{1}{n} \text{diag}\{1, 2, 2, \dots, 2\} \mathbf{R}^\top$.

In order to define the Chebyshev integration matrices in two-dimensional space, we need to introduce the Kronecker product.

2.3 Kronecker Product

Definition 2.3.1 ([13]). Let $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{m \times n}$ and $\mathbf{B} = [b_{ij}] \in \mathbb{R}^{p \times q}$. Then, $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{mp \times nq}$ is the Kronecker product defined by a block matrix as follows:

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

We state the fact of the Kronecker product without proof as follows.

Theorem 2.3.1 ([13]). The Kronecker product has the following characteristics.

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

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

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

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

3. Let $\mathbf{A} \in \mathbb{R}^{m \times m}$, $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $\mathbf{P} := [\mathbf{I}_n \otimes \mathbf{e}_1, \mathbf{I}_n \otimes \mathbf{e}_2, \dots, \mathbf{I}_n \otimes \mathbf{e}_m]$ be an $mn \times mn$ permutation matrix, where \mathbf{I}_n is an $n \times n$ identity matrix and $\mathbf{e}_i := [0, \dots, 0, 1, 0, \dots, 0]^T$ is an m -dimensional column vector which has 1 in the i^{th} positions and 0's elsewhere. Then, $\mathbf{P}(\mathbf{A} \otimes \mathbf{B})\mathbf{P}^T = \mathbf{B} \otimes \mathbf{A}$.

2.4 Developed FIM-CPE

2.4.1 One-Dimensional Chebyshev Integration Matrices

Next, we construct the Chebyshev integration matrix in one-dimensional which is an instrument for dealing with the integral term. First, let $M \in \mathbb{N}$ and u be any function that can be approximated by the Chebyshev polynomial expansion as follows

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

where c_n are the unknown coefficients to be determined later and $a, b \in \mathbb{R}$. Let $x_1 < x_2 < \dots < x_M$ be nodal points which are discretized by the zeros of Chebyshev polynomial $R_M(x)$ defined in (2.6). Substituting each x_k into (2.7), it can be expressed as

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

which is denoted by $\mathbf{u} = \mathbf{R}\mathbf{c}$. Thus, $\mathbf{c} = \mathbf{R}^{-1}\mathbf{u}$, where \mathbf{R}^{-1} is defined in Lemma 2.2.1. Next, let us consider the single layer integration of u from a to x_k which is denoted by $U^{(1)}(x_k)$, we obtain

$$U^{(1)}(x_k) = \int_a^{x_k} u(\xi) d\xi = \sum_{n=0}^{M-1} c_n \int_a^{x_k} R_n(\xi) d\xi = \sum_{n=0}^{M-1} c_n \bar{R}_n(x_k), \quad (2.8)$$

for $k \in \{1, 2, 3, \dots, M\}$ where each $\bar{R}_n(x_k)$ is defined in Lemma 2.2.1. Or in 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 is denoted by $\mathbf{U}^{(1)} = \bar{\mathbf{R}}\mathbf{c} = \bar{\mathbf{R}}\bar{\mathbf{R}}^{-1}\mathbf{u} := \mathbf{A}\mathbf{u}$, where $\mathbf{A} = \bar{\mathbf{R}}\bar{\mathbf{R}}^{-1} := [a_{ki}]_{M \times M}$ is called the Chebyshev integration matrix. For $k \in \{1, 2, 3, \dots, M\}$, it has another form,

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

2.4.2 Two-Dimensional Chebyshev Integration Matrices

For two-dimensional Chebyshev integration matrices, let $M, N \in \mathbb{N}$. Let x_k and y_k be computation nodes over $[a, b] \times [a, b]$ along the horizontal and vertical directions meshed by the zeros of Chebyshev polynomials $R_M(x)$ and $R_N(y)$, respectively. Hence, the total grid numbers in the system are $M \times N$ points. For convenience, we index the numbering of grid points along the x -direction by the global numbering system (Figure 2.3a) and y -direction by the local numbering system (Figure 2.3b).

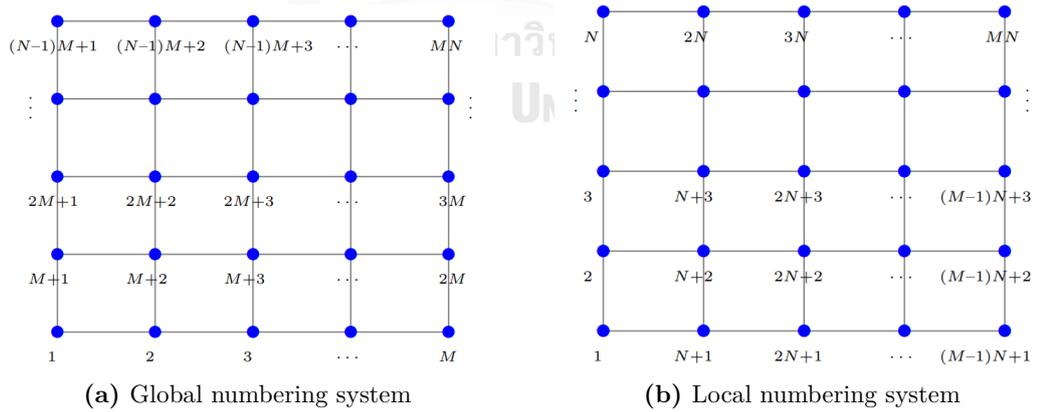


Figure 2.3: The indices of the grid points globally and locally

First, let us consider the single-layer integration with respect to the variables x and y that are denoted by $U_x^{(1)}(x, y)$ and $U_y^{(1)}(x, y)$, respectively. For each fixed y , we have $U_x^{(1)}(x_k, y)$ in the global numbering system as

$$U_x^{(1)}(x_k, y) = \int_a^{x_k} u(\xi, y) d\xi = \sum_{i=1}^M a_{ki} u(x_i, y). \quad (2.9)$$

For $k \in \{1, 2, 3, \dots, M\}$, (2.9) can be expressed by $\mathbf{U}_x^{(1)}(\cdot, y) = \mathbf{A}_M \mathbf{u}(\cdot, y)$, where $\mathbf{A}_M = \overline{\mathbf{R}}\mathbf{R}^{-1}$ is an $M \times M$ matrix. Thus, for each $y \in \{y_1, y_2, y_3, \dots, y_N\}$,

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

which is represented by $\mathbf{U}_x^{(1)} = \mathbf{A}_x \mathbf{u}$, where

$$\mathbf{A}_x = \mathbf{I}_N \otimes \mathbf{A}_M \quad (2.10)$$

is the Chebyshev integration matrix with respect to x -axis. Similarly, for each fixed x , $U_y^{(1)}(x, y_s)$ can be expressed in the local numbering system as

$$U_y^{(1)}(x, y_s) = \int_a^{y_s} u(x, \eta) d\eta = \sum_{j=1}^N a_{sj} u(x, y_j). \quad (2.11)$$

For $s \in \{1, 2, 3, \dots, N\}$, (2.11) can be expressed as $\mathbf{U}_y^{(1)}(x, \cdot) = \mathbf{A}_N \mathbf{u}(x, \cdot)$, where $\mathbf{A}_N =$

$\overline{\mathbf{R}}\mathbf{R}^{-1}$ is an $N \times N$ matrix. So, for each $x \in \{x_1, x_2, x_3, \dots, x_M\}$,

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

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

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

for all $k \in \{1, 2, 3, \dots, M\}$ and $s \in \{1, 2, 3, \dots, N\}$. We obtain that $\mathbf{U}_y^{(1)} = \mathbf{P}\tilde{\mathbf{U}}_y^{(1)}$ and $\mathbf{u} = \mathbf{P}\tilde{\mathbf{u}}$. Therefore, $\mathbf{U}_y^{(1)} = \mathbf{A}_y \mathbf{u}$, where

$$\mathbf{A}_y = \mathbf{P}\tilde{\mathbf{A}}_y\mathbf{P}^{-1} = \mathbf{P}(\mathbf{I}_M \otimes \mathbf{A}_N)\mathbf{P}^T \quad (2.12)$$

is the Chebyshev integration matrix with respect to y -axis in the global numbering system. We noted that the integration with respect to both x and y is $\mathbf{U}_{xy}^{(1)} = \mathbf{A}_x \mathbf{A}_y \mathbf{u} = \mathbf{A}_y \mathbf{A}_x \mathbf{u}$. Now, we have one- and two-dimensional Chebyshev integration matrices which is an important tool to develop schemes for solving both one- and two-Dimensional shallow water equations.

CHAPTER III

NUMERICAL ALGORITHM FOR ONE-DIMENSIONAL SWES

In this chapter, we propose a numerical algorithm for approximating the solutions to the one-dimensional SWEs (2.1) and (2.2) for various types of initial heights and bottom topographies, with reflecting boundaries (walls), using the suggested FIM-CPE.

3.1 Numerical Algorithm

Before deriving the algorithm, let the quantity hu be expressed by the discharge q . Then, (2.1) and (2.2) can be written as

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = 0, \quad (3.1)$$

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x} \left(\frac{q^2}{h} + \frac{gh^2}{2} \right) + gh \frac{dz}{dx} = 0, \quad (3.2)$$

for all $(x, t) \in (a, b) \times (0, T]$, where $a, b, T \in \mathbb{R}$ in which h is the water depth, $q = hu$ is the discharge, u is the flow velocity in the x -direction, g is the acceleration due to gravity and z is the bottom elevation. Assume that h and u are smooth real-valued functions of the temporal coordinate. This system subjects to the initial conditions

$$h(x, 0) = \phi_0(x) \quad \text{and} \quad u(x, 0) = \phi_1(x) \quad \text{for } x \in [a, b], \quad (3.3)$$

and the reflecting boundary conditions [8],

$$u(a, t) = 0 \quad \text{and} \quad u(b, t) = 0 \quad \text{for } t \in (0, T]. \quad (3.4)$$

The FIM-CPE for one-dimensional SWEs is started by discretizing the computational spatial domain (a, b) into M nodes generated by the zeros of Chebyshev polynomial $R_M(x)$ as defined in (2.6) in ascending order, that is x_k for $k \in \{1, 2, 3, \dots, M\}$. Then, we divide the temporal domain $(0, T]$ by the step-size of time τ which will be defined later and $t_m = t_{m-1} + \tau$ for all $m \in \mathbb{N}$, where $t_0 = 0$.

Next, we handle the temporal variable t in both (3.1) and (3.2) by specifying the time step t_m into them which are notated by a superscript of $\langle m \rangle$. By assumption, since h and u are smooth functions in time variable t , then $q = hu$ is also smooth. Hence, we obtain that the functions h and q at any two-consecutive times provide the values most closely. In other words, let arbitrary two-consecutive times $0 \leq t_{m-1} < t_m$ for all $m \in \mathbb{N}$. If $|t_{m-1} - t_m| \rightarrow 0$, then $|h^{\langle m \rangle} - h^{\langle m-1 \rangle}| \rightarrow 0$ and $|q^{\langle m \rangle} - q^{\langle m-1 \rangle}| \rightarrow 0$. Therefore, this assumption is sufficient to employ the linearization for nonlinear terms under the time variable t and also the approximation of derivatives with respect to time t . Afterward, we apply the first-order forward difference quotient to deal with the time derivatives in (3.1) and (3.2) and utilize the linearization method to manipulate the nonlinear terms in (3.2). Thus, (3.1) and (3.2) become

$$\frac{h^{\langle m \rangle} - h^{\langle m-1 \rangle}}{\tau} + \frac{dq^{\langle m \rangle}}{dx} = 0, \quad (3.5)$$

$$\frac{q^{\langle m \rangle} - q^{\langle m-1 \rangle}}{\tau} + \frac{d}{dx} \left(\frac{q^{\langle m-1 \rangle} q^{\langle m \rangle}}{h^{\langle m-1 \rangle}} + \frac{gh^{\langle m-1 \rangle} h^{\langle m \rangle}}{2} \right) + gh^{\langle m \rangle} \frac{dz}{dx} = 0, \quad (3.6)$$

where $h^{\langle m \rangle} = h^{\langle m \rangle}(x) = h(x, t_m)$ and $q^{\langle m \rangle} = q^{\langle m \rangle}(x) = q(x, t_m)$ are the numerical values at the m^{th} time step. Next, we multiply (3.5) and (3.6) by τ to preserve the round-off error caused by division by a small step size. To apply the proposed FIM-CPE, we first eliminate all derivatives out of (3.5) and (3.6) by taking the single-layer integral on both

sides of them from a to the zero x_k . Then, we obtain

$$\int_a^{x_k} h^{(m)}(\xi) - h^{(m-1)}(\xi) d\xi + \tau q^{(m)}(x_k) + r_1 = 0, \quad (3.7)$$

$$\int_a^{x_k} q^{(m)}(\xi) - q^{(m-1)}(\xi) d\xi + \frac{\tau q^{(m-1)}(x_k) q^{(m)}(x_k)}{h^{(m-1)}(x_k)} + \frac{\tau g h^{(m-1)}(x_k) h^{(m)}(x_k)}{2} + \tau g \int_a^{x_k} \frac{dz(\xi)}{d\xi} h^{(m)}(\xi) d\xi + r_2 = 0, \quad (3.8)$$

where r_1 and r_2 are arbitrary constants that emerged from the process of integration for the functions $h^{(m)}$ and $q^{(m)}$. Next, we transform (3.7) and (3.8) into the matrix forms by hiring the Chebyshev integration matrix. When each zero x_k for $k \in \{1, 2, 3, \dots, M\}$ is plugged into (3.7) and (3.8), we have the following simplified matrix equations

$$\mathbf{A}\mathbf{h}^{(m)} + \tau\mathbf{q}^{(m)} + r_1\mathbf{e} = \mathbf{A}\mathbf{h}^{(m-1)}, \quad (3.9)$$

$$\mathbf{A}\mathbf{q}^{(m)} + \tau\mathbf{B}^{(m-1)}\mathbf{q}^{(m)} + \frac{\tau g}{2}\mathbf{D}^{(m-1)}\mathbf{h}^{(m)} + \tau g\mathbf{AZ}\mathbf{h}^{(m)} + r_2\mathbf{e} = \mathbf{A}\mathbf{q}^{(m-1)}, \quad (3.10)$$

where $\mathbf{A} = \bar{\mathbf{R}}\mathbf{R}^{-1}$ is the Chebyshev integration matrix and $\mathbf{e} = [1, 1, 1, \dots, 1]^\top$ has M entries. Other parameters contained in both (3.9) and (3.10) are defined by

$$\begin{aligned} \mathbf{h}^{(m)} &= [h^{(m)}(x_1), h^{(m)}(x_2), h^{(m)}(x_3), \dots, h^{(m)}(x_M)]^\top, \\ \mathbf{q}^{(m)} &= [q^{(m)}(x_1), q^{(m)}(x_2), q^{(m)}(x_3), \dots, q^{(m)}(x_M)]^\top, \\ \mathbf{Z} &= \text{diag}\{z'(x_1), z'(x_2), z'(x_3), \dots, z'(x_M)\}, \\ \mathbf{B}^{(m-1)} &= \text{diag}\left\{\frac{q^{(m-1)}(x_1)}{h^{(m-1)}(x_1)}, \frac{q^{(m-1)}(x_2)}{h^{(m-1)}(x_2)}, \frac{q^{(m-1)}(x_3)}{h^{(m-1)}(x_3)}, \dots, \frac{q^{(m-1)}(x_M)}{h^{(m-1)}(x_M)}\right\} \text{ and} \\ \mathbf{H}^{(m-1)} &= \text{diag}\{h^{(m-1)}(x_1), h^{(m-1)}(x_2), h^{(m-1)}(x_3), \dots, h^{(m-1)}(x_M)\}. \end{aligned}$$

Now, we note in (3.9) and (3.10) that their numbers of unknown variables have up to $2M + 2$ including $\mathbf{h}^{(m)}$, $\mathbf{q}^{(m)}$, r_1 and r_2 . However, they have only $2M$ numbers of equations. Thus, we need to construct 2 more equations. From the given reflecting boundary conditions (3.4), they can be written into the vector form by employing the Chebyshev

polynomial expansion (2.7) specifying at the m^{th} time step as the following

$$q^{(m)}(a) = \sum_{n=0}^{M-1} c_n^{(m)} R_n(a) := \mathbf{p}_l^\top \mathbf{c}^{(m)} = \mathbf{p}_l^\top \mathbf{R}^{-1} \mathbf{q}^{(m)} = 0, \quad (3.11)$$

$$q^{(m)}(b) = \sum_{n=0}^{M-1} c_n^{(m)} R_n(b) := \mathbf{p}_r^\top \mathbf{c}^{(m)} = \mathbf{p}_r^\top \mathbf{R}^{-1} \mathbf{q}^{(m)} = 0, \quad (3.12)$$

where $\mathbf{p}_l = [R_0(a), R_1(a), R_2(a), \dots, R_{M-1}(a)]^\top$ and $\mathbf{p}_r = [R_0(b), R_1(b), R_2(b), \dots, R_{M-1}(b)]^\top$. In fact, the Chebyshev polynomials at end points are $R_n(a) = (-1)^n$ and $R_n(b) = 1$, for $n \in \mathbb{N} \cup \{0\}$. Finally, we can construct the system of linear equations from (3.9)–(3.12) which has a total number of $2M + 2$ unknowns containing $\mathbf{h}^{(m)}$, $\mathbf{q}^{(m)}$, r_1 and r_2 as follows

$$\left[\begin{array}{cc|cc} \mathbf{A} & \tau \mathbf{I} & \mathbf{e} & \mathbf{0} \\ \frac{\tau g}{2} \mathbf{H}^{(m-1)} + \tau g \mathbf{A} \mathbf{Z} & \mathbf{A} + \tau \mathbf{B}^{(m-1)} & \mathbf{0} & \mathbf{e} \\ \hline \mathbf{0}^\top & \mathbf{p}_l^\top \mathbf{R}^{-1} & 0 & 0 \\ \mathbf{0}^\top & \mathbf{p}_r^\top \mathbf{R}^{-1} & 0 & 0 \end{array} \right] \begin{bmatrix} \mathbf{h}^{(m)} \\ \mathbf{q}^{(m)} \\ r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A} \mathbf{h}^{(m-1)} \\ \mathbf{A} \mathbf{q}^{(m-1)} \\ 0 \\ 0 \end{bmatrix}, \quad (3.13)$$

where \mathbf{I} is the $M \times M$ identity matrix and $\mathbf{0}$ is the M zero column vector. Therefore, we can solve (3.13) to obtain the approximate solutions $\mathbf{h}^{(m)}$ and $\mathbf{q}^{(m)}$ by starting with the initial conditions (3.3) written in the vector forms $\mathbf{h}^{(0)} = [\phi_0(x_1), \phi_0(x_2), \phi_0(x_3), \dots, \phi_0(x_M)]^\top$ and $\mathbf{u}^{(0)} = [\phi_1(x_1), \phi_1(x_2), \phi_1(x_3), \dots, \phi_1(x_M)]^\top$. Then, $\mathbf{q}^{(0)} = \mathbf{h}^{(0)} \odot \mathbf{u}^{(0)}$, where \odot is the Hadamard product [1] which is a product of two vectors element-wise. Also, the solution $\mathbf{u}^{(m)}$ is directly obtained by $\mathbf{u}^{(m)} = \mathbf{q}^{(m)} \oslash \mathbf{h}^{(m)}$, where \oslash is the Hadamard division [10] which means a division of two vectors element-wise. However, the stability of this scheme should be mentioned. The obtained approximations by the such scheme will only converge to their analytical solution as the refined grid if the Courant-Friedrichs-Lewy condition is satisfied based on [9], which we need to set

$$\tau = \text{CFL} \frac{\min_i (\Delta x_i)}{\max_k \left(|u^{(m-1)}(x_k)| + \sqrt{gh^{(m-1)}(x_k)} \right)}, \quad (3.14)$$

where CFL is the Courant number. To satisfy the stability, the CFL must be less than one because the distance traveled by the wave in each time step must not exceed the distance

between neighboring nodes or cells in the mesh. To ensure that the water will move in a controlled manner and prevents it from jumping over other cells. Since τ is varying throughout the approximation process, we cannot accurately determine the value at the desired time. Therefore, we choose the time that is closest and less than the desired time. For computational convenience, we provide the flowchart algorithm as seen in Figure 3.1.

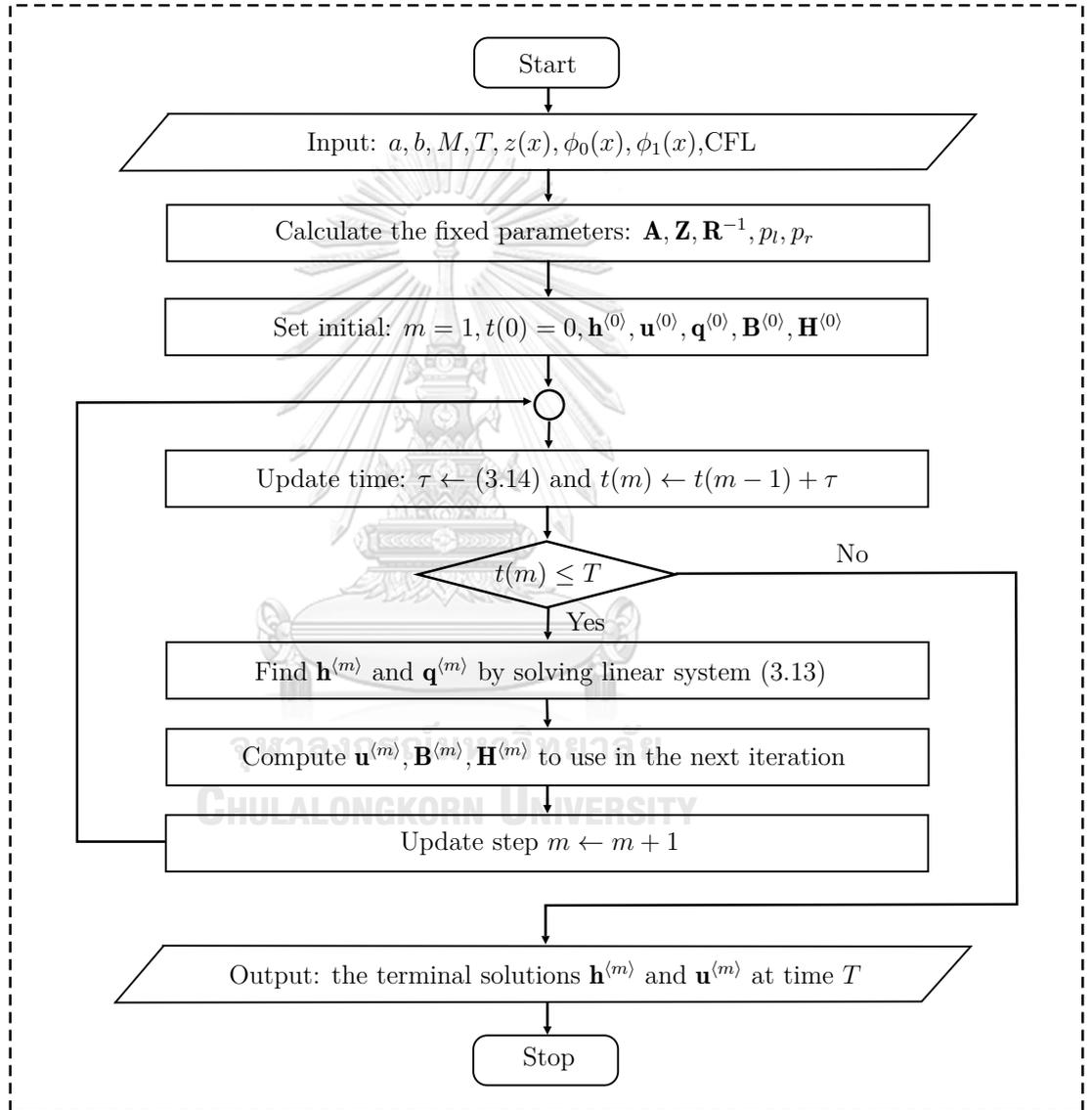


Figure 3.1: The flowchart for solving one-dimension SWEs

3.2 Numerical Simulations

In this section, we investigate the efficiency, accuracy and stability of our proposed numerical algorithm through four examples of one-dimensional SWEs with different types of bottom topography on a wet bed, including the lake at rest, the dam break flow which is an important type of disaster and Gaussian pulse. Extensive research efforts have been made to understand and defend against this disaster using methods ranging from small- and large-scale experiments to numerical modeling. Therefore, the experimental examples include dam break problems with flat and non-flat bottoms, as well as a lake at rest to verify the accuracy of the scheme. In this study, we used an AMD Ryzen 7 4800HS CPU with 16.0 GB of RAM and Matlab software for all data processing and analysis.

Example 3.2.1 (Lake at rest). The lake at rest problem is a simple and well-defined scenario where the water surface height and horizontal velocity are both constant over time and space [8]. Therefore, it serves as a good benchmark for testing the accuracy of numerical schemes for solving SWEs. The computational domain is $[0, 10]$. The initial velocity is zero. The bottom topography is defined by

$$z(x) = 5e^{-\left(\frac{x-5}{0.8}\right)^2} \quad (3.15)$$

and the initial water height is $h(x, 0) = 10 - z(x)$. In this computational, we run the simulation on $M = 100$ grids using $T = 10$ s for a long end period. Notice from implementing with values of $\text{CFL} = 0.1, 0.2, 0.3, \dots, 0.9$, we found that $\text{CLF} = 0.5$ provides the best accuracy for our algorithm. In this example, we employ our proposed algorithm in Figure 3.1 with $\text{CLF} = 0.5$ to solve the problem. In addition, we also use $\text{CLF} = 0.5$ for the other examples. We can see the result in Figure 3.2 that, it is stable in time with a run time of 30.0897s. The mean absolute error defined by

$$\text{MAE}_v = \frac{1}{M} \sum_{k=1}^M |v^*(x_k, T) - v(x_k, T)|, \quad (3.16)$$

where v^* and v are exact and numerical solutions, respectively. At the end time, the errors in h and u are 6.4485×10^{-12} and 8.491×10^{-13} , respectively. To evaluate the

conservative of mass, we compute the total volume by using

$$\frac{1}{2} \left[(x_2 - a)h_1^{<m>} + (b - x_{M-1})h_M^{<m>} + \sum_{i=1}^{M-2} (x_{i+2} - x_i)h_{i+1}^{<m>} \right]$$

where x_1, x_2, \dots, x_M are nodal points, M is the number of nodal points, $h_1^{<m>}, h_2^{<m>}, \dots, h_M^{<m>}$ are the height of water at each time $t(m)$ from the simulation. By the experiment, the total water varies minimally, not exceeding 10^{-5} throughout the simulation period. Similarly with Example 3.2.2-3.2.4.

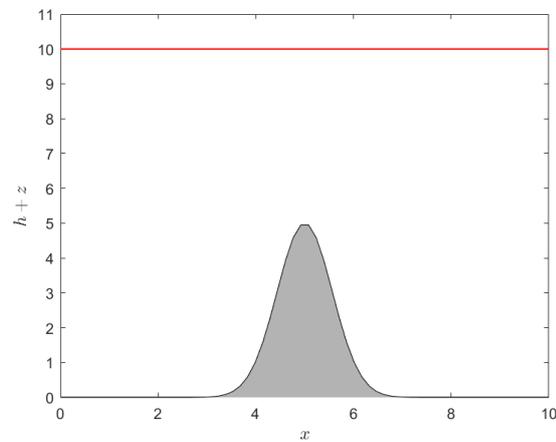


Figure 3.2: Water height at $T = 10$ s.

Example 3.2.2 (Dam break over flat bottom [12]). This dam break problem is one of the most basic studies based on the SWEs due to the flat topography or $z(x) = 0$. The computational domain is $[0, 1]$. The initial velocity is zero and the initial water height is

$$h(x, 0) = \begin{cases} 1 & \text{for } 0 \leq x < \frac{1}{2}, \\ \frac{1}{2} & \text{for } \frac{1}{2} \leq x \leq 1. \end{cases}$$

This means that the discontinuity at $x = \frac{1}{2}$ represents a barrier that separates the

two initial river heights. The exact solution to this problem is given by Stoker [12], i.e.,

$$h^*(x, t) = \begin{cases} 1 & \text{for } x < \frac{1}{2} - t\sqrt{g}, \\ \frac{1}{9g} \left(2\sqrt{g} - \frac{2x-1}{2t}\right)^2 & \text{for } \frac{1}{2} - t\sqrt{g} \leq x \leq \frac{1}{2} + (v-w)t, \\ \frac{w^2}{g} & \text{for } \frac{1}{2} + (v-w)t < x \leq \frac{1}{2} + St, \\ \frac{1}{2} & \text{for } x > \frac{1}{2} + St \end{cases}$$

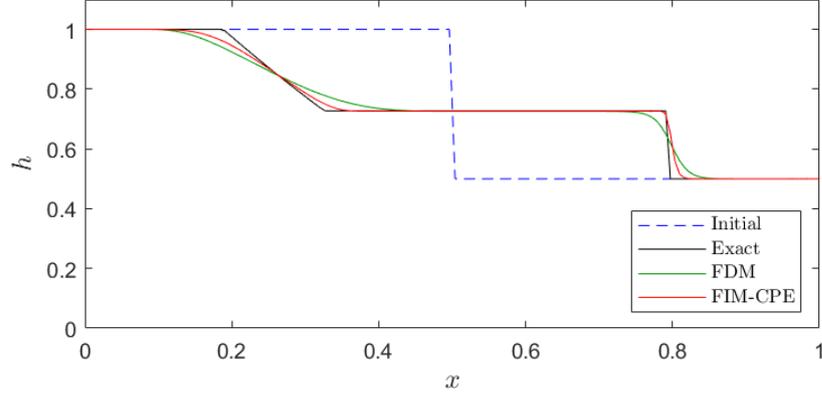
and

$$u^*(x, t) = \begin{cases} 0 & \text{for } x < \frac{1}{2} - t\sqrt{g}, \\ \frac{1}{3t} (2x - 1 + 2t\sqrt{g})^2 & \text{for } \frac{1}{2} - t\sqrt{g} \leq x \leq \frac{1}{2} + (v-w)t, \\ v & \text{for } \frac{1}{2} + (v-w)t < x \leq \frac{1}{2} + St, \\ 0 & \text{for } x > \frac{1}{2} + St, \end{cases}$$

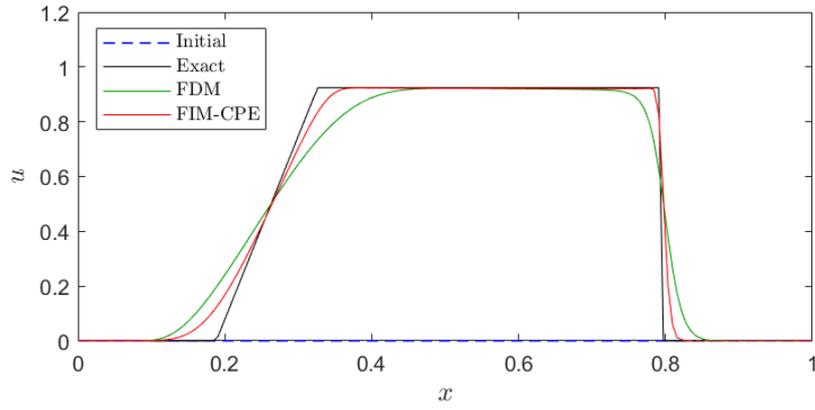
where $v = S - \frac{g}{8S} \left(1 + \sqrt{1 + \frac{16S^2}{g}}\right)$, $w = \sqrt{\frac{g}{4} \left(\sqrt{1 + \frac{16S^2}{g}} - 1\right)}$ and $S = 2.9579181201875$. For a more depth analysis of how the value S was obtained, see Stoker [12]. By testing, we found that the obtained results are well accurate when compared with its exact solution given in [12]. The comparisons are demonstrated in Table 3.1 measured at the final interested time T for different values of grid point M by using the mean absolute error (3.16). The error in Table 3.1 does not differ much as the grid becomes finer, because the initial height is not smooth (at the shock $x = \frac{1}{2}$). This causes the approximate solution to not capture the area around the shock accurately but we further compare approximate solutions obtained by our algorithm and by the FDM [5] for various nodal points M as shown in Table 3.1. We can see that at the same number M , our scheme provides solutions closer to the analytical solutions than the FDM. Additionally, when increasing the number of nodes M , the obtained solutions are likewise ever-increasing accurate. The consuming times of computation are also indicated in Table 3.1.

Moreover, we can visualize the solutions for both water height h and velocity u received by using our algorithm with $M = 200$ at $T = 0.1$ as depicted in Figure 3.3.

From Figure 3.3, the initial height is broken at $x = \frac{1}{2}$, when time passes. We see that our scheme can well capture at around the shock break and quite matches the exact solution.



(a) Water height $h(x, T)$



(b) Water velocity $u(x, T)$

Figure 3.3: Graphical solutions with $M = 200$ at time $T = 0.1$

Table 3.1: Comparisons of exact and numerical solutions at $T = 0.1$

M	FDM[5]		Our algorithm		Run time (s.)
	MAE_h	MAE_u	MAE_h	MAE_u	
100	2.02×10^{-2}	7.53×10^{-2}	4.73×10^{-3}	1.65×10^{-2}	0.4147
200	1.31×10^{-2}	4.82×10^{-2}	2.88×10^{-3}	1.07×10^{-2}	2.6712
300	9.96×10^{-3}	3.64×10^{-2}	2.05×10^{-3}	7.72×10^{-3}	8.4295
400	8.15×10^{-3}	2.98×10^{-2}	1.63×10^{-3}	6.19×10^{-3}	20.0024

Example 3.2.3 (Dam break over a bump). This presented problem discussed by LeVeque [7]. It is a nonflat bottom of the topography with smoothness. Therefore, $z'(x) \neq 0$ for

some values of x . This test problem is considered over the smooth bottom topography shaped as the bump defined by

$$z(x) = \begin{cases} \frac{1}{4} (\cos(10x - 5)\pi + 1) & \text{for } 0.4 \leq x \leq 0.6, \\ 0 & \text{otherwise.} \end{cases}$$

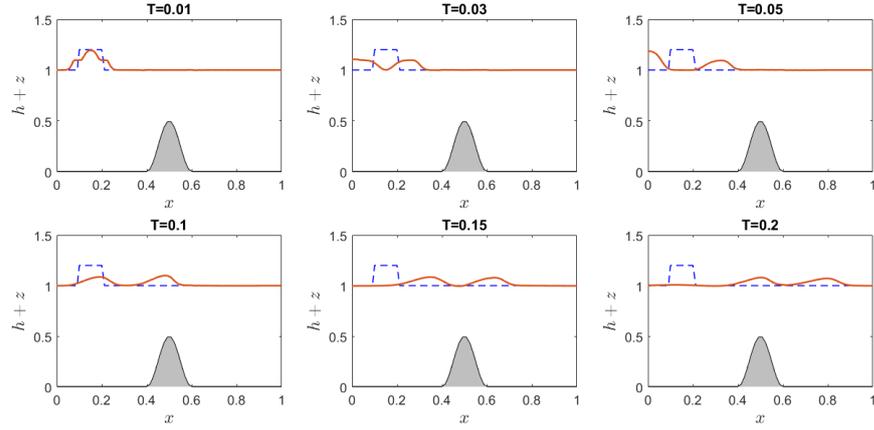
The computational domain of this problem is $[0, 1]$. The initial velocity is zero and the initial water height is defined as the following equations involved the bump $z(x)$,

$$h(x, 0) = \begin{cases} 1 - z(x) & \text{for } x < 0.1, \\ \frac{6}{5} - z(x) & \text{for } 0.1 \leq x \leq 0.2, \\ 1 - z(x) & \text{for } x > 0.2. \end{cases}$$

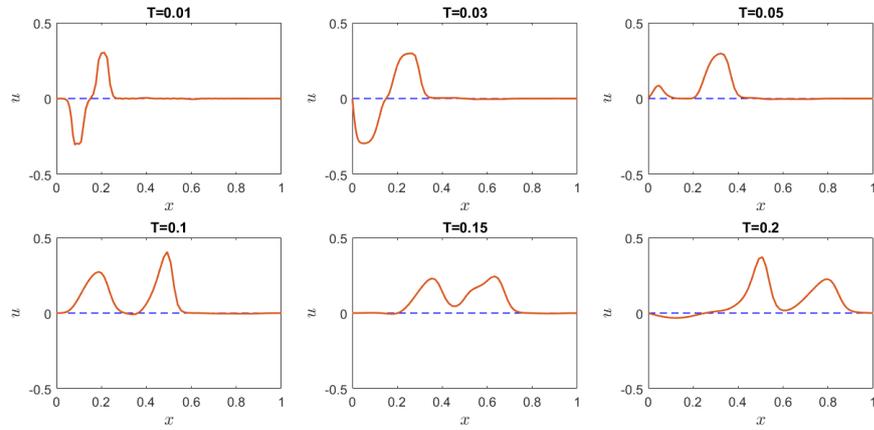
This problem presents an initial water height shaped like a pulse that breaks up into two waves moving in opposite directions. The square-wave pulse moving towards the right side passes through the bump in the riverbed, partially reflecting and causing a disturbance behind the bump. The other side reflects off the wall.

In the experiment, our presented numerical algorithm is applied to find the approximate solutions $h(x, t)$ and $u(x, t)$ of the problem. Using the grid number $M = 100$, we can simulate the behavior of wave propagation at different times $t \in \{0.01, 0.03, 0.05, 0.1, 0.15, 0.2\}$ depicted in Figure 3.4 for water height and velocity with run time 11.9876s. We can see that at starting time, the wave pulse moves right-toward with $h + z$ around 1.1. After the wave pulse moving to the bump, it is partially reflected and affects to decrease $h + z$ when moving afterward. Behind the bump, we see that the wave pulse is still ever-decreasing. The wave pulse also moves left towards $h + z$ around 1.1. After it meets the wall, it increases ($T = 0.05$) and then returns to the same level as before ($T = 0.1$). The behavior of the left wave is the same as the right wave when passing the bump. Furthermore, this problem is also studied with several methods given in [5]. Our algorithm produces the same right wave behavior as those methods because the left side meets the wall.

Example 3.2.4 (Gaussian Pulse [8]). By utilizing this example, we aim to demonstrate



(a) Water height $h(x, T) + z(x)$ at time $T = 0.01, 0.03, 0.05, 0.1, 0.15, 0.2$



(b) Water velocity $u(x, T)$ at time $T = 0.01, 0.03, 0.05, 0.1, 0.15, 0.2$

Figure 3.4: Graphical solutions with $M = 100$ at different times T

that our method offers an advantage in handling smooth solutions with reflecting boundary conditions due to flat bottom [8]. The initial Gaussian is defined by the following equation

$$h(x, 0) = 1 + 0.1e^{-\left(\frac{x-0.5}{0.1}\right)^2}.$$

We simulate this experiment on $M = 60$ grids. The result from our method, see in Figure 3.5. The wave consistently appears to be smooth. Initially, it separates into left and right sides, reflects off the walls, and turns into a hump. This cycle continues over time with the height decreasing until the velocity approaches zero, leading to a constant wave. This state can be referred to as a steady state. Therefore, we observe that at time $T = 9.5$,

the experiment has reached a steady state within a tolerance of 10^{-5} with a run time 8.6104mins.

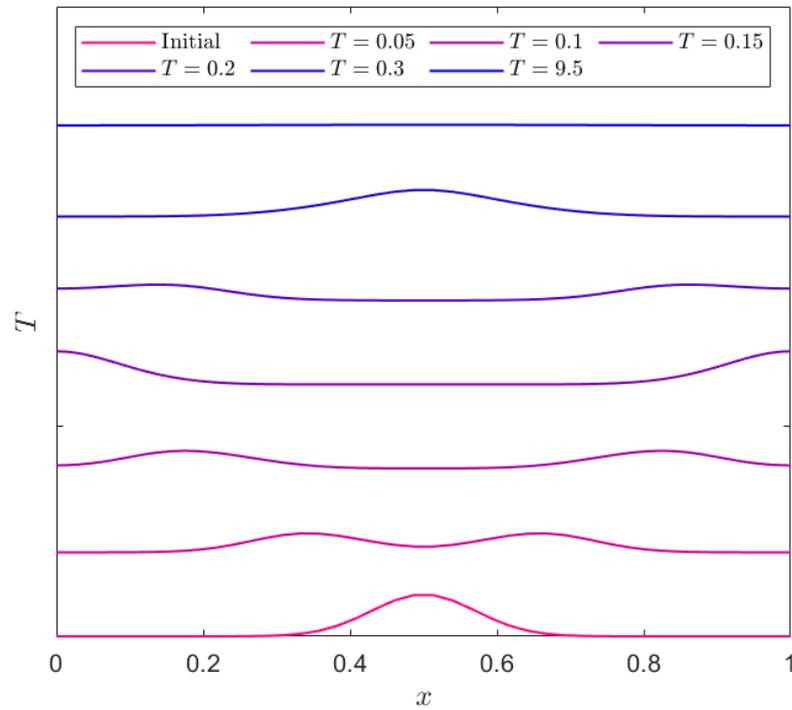


Figure 3.5: Water height $h(x, t)$ at various times(s).



CHAPTER IV

NUMERICAL ALGORITHM FOR TWO-DIMENSIONAL SWES

This chapter presents a numerical method that uses the suggested FIM-CPE to find rough solutions for the two-dimensional SWEs (2.3), (2.4) and (2.5) with the reflecting boundary conditions (wall).

4.1 Numerical Algorithm

Before deriving the algorithm, let the quantity hu, hv be expressed by the discharge q_1, q_2 , respectively. Then, (2.3), (2.4) and (2.5) can be written as

$$\frac{\partial h}{\partial t} + \frac{\partial q_1}{\partial x} + \frac{\partial q_2}{\partial y} = 0, \quad (4.1)$$

$$\frac{\partial q_1}{\partial t} + \frac{\partial}{\partial x} \left(\frac{q_1^2}{h} + \frac{gh^2}{2} \right) + \frac{\partial}{\partial y} \left(\frac{q_1 q_2}{h} \right) + gh \frac{\partial z}{\partial x} = 0, \quad (4.2)$$

$$\frac{\partial q_2}{\partial t} + \frac{\partial}{\partial x} \left(\frac{q_1 q_2}{h} \right) + \frac{\partial}{\partial y} \left(\frac{q_2^2}{h} + \frac{gh^2}{2} \right) + gh \frac{\partial z}{\partial y} = 0, \quad (4.3)$$

for all $(x, y, t) \in (a, b) \times (c, d) \times (0, T]$, where $a, b, c, d, T \in \mathbb{R}$, $q_1 = hu$ and $q_2 = hv$ are the discharges. Assume that h, u and v are smooth real-valued functions of the temporal coordinate. This system subjects to the initial conditions

$$h(x, y, 0) = \phi_0(x, y), \quad q_1(x, y, 0) = \phi_1(x, y) \quad \text{and} \quad q_2(x, y, 0) = \phi_2(x, y) \quad (4.4)$$

for $(x, y) \in [a, b] \times [c, d]$. The reflecting boundary conditions, described in [8],

$$q_1(x, y, t) = 0 \quad \text{and} \quad q_2(x, y, t) = 0, \quad (4.5)$$

for $(x, y, t) \in \partial([a, b] \times [c, d]) \times (0, T]$.

The FIM-CPE for two-dimensional SWEs uses similar procedures to those in the one-dimensional SWEs as follows :

1. Discretize the computational domain:

- To partition the domain $(a, b) \times (c, d)$ into $H = M \times N$ nodes, we use the zeros of the Chebyshev polynomial $R_M(x)$ and $R_N(y)$ as defined in (2.6) within the global numbering system. The nodes are denoted by (x_k, y_k) where $k \in \{1, 2, \dots, H\}$.
- We divide the temporal domain $(0, T]$ by the time step τ , which will be defined later and set $t_m = t_{m-1} + \tau$ for all $m \in \mathbb{N}$, where $t_0 = 0$.

2. Approximate the time derivatives using the first-order forward difference quotient and use the linearization method to manipulate the nonlinear terms:

Thus, (4.1) – (4.3) become

$$0 = \frac{h^{(m)} - h^{(m-1)}}{\tau} + \frac{\partial q_1^{(m)}}{\partial x} + \frac{\partial q_2^{(m)}}{\partial y}, \quad (4.6)$$

$$0 = \frac{q_1^{(m)} - q_1^{(m-1)}}{\tau} + \frac{\partial}{\partial x} \left(\frac{q_1^{(m-1)} q_1^{(m)}}{h^{(m-1)}} + \frac{gh^{(m-1)}h^{(m)}}{2} \right) + \frac{\partial}{\partial y} \left(\frac{q_1^{(m-1)} q_2^{(m)}}{h^{(m-1)}} \right) + gh^{(m)} \frac{\partial z}{\partial x}, \quad (4.7)$$

$$0 = \frac{q_2^{(m)} - q_2^{(m-1)}}{\tau} + \frac{\partial}{\partial x} \left(\frac{q_2^{(m-1)} q_1^{(m)}}{h^{(m-1)}} \right) + \frac{\partial}{\partial y} \left(\frac{q_2^{(m-1)} q_2^{(m)}}{h^{(m-1)}} + \frac{gh^{(m-1)}h^{(m)}}{2} \right) + gh^{(m)} \frac{\partial z}{\partial y}, \quad (4.8)$$

where $h^{(m)} = h^{(m)}(x, y) = h(x, y, t_m)$, $q_1^{(m)} = q_1^{(m)}(x, y) = q_1(x, y, t_m)$ and $q_2^{(m)} = q_2^{(m)}(x, y) = q_2(x, y, t_m)$ are the numerical values at the m^{th} time step.

3. Eliminate all derivatives:

We take the double-layer integral on both sides of (4.6) – (4.8) from a to the zero

x_k and c to the zero y_k . Then, we obtain

$$0 = \int_c^{y_k} \int_a^{x_k} h^{(m)}(\xi, \eta) - h^{(m-1)}(\xi, \eta) d\xi d\eta + \tau \int_c^{y_k} q_1^{(m)}(x_k, \eta) d\eta + \tau \int_a^{x_k} q_2^{(m)}(\xi, y_k) d\xi + r_1(x) + s_1(y) + r_2(x) + s_2(y), \quad (4.9)$$

$$0 = \int_c^{y_k} \int_a^{x_k} q_1^{(m)}(\xi, \eta) - q_1^{(m-1)}(\xi, \eta) d\xi d\eta + \tau \int_c^{y_k} \frac{q_1^{(m-1)}(x_k, \eta) q_1^{(m)}(x_k, \eta)}{h^{(m-1)}(x_k, \eta)} + \frac{gh^{(m)}(x_k, \eta) h^{(m-1)}(x_k, \eta)}{2} d\eta + \tau \int_a^{x_k} \frac{q_1^{(m-1)}(\xi, y_k) q_2^{(m)}(\xi, y_k)}{h^{(m-1)}(\xi, y_k)} d\xi + \tau g \int_c^{y_k} \int_a^{x_k} \frac{\partial z(\xi, \eta)}{\partial \xi} h^{(m)}(\xi, \eta) d\xi d\eta + r_3(x) + s_3(y), \quad (4.10)$$

$$0 = \int_c^{y_k} \int_a^{x_k} q_2^{(m)}(\xi, \eta) - q_2^{(m-1)}(\xi, \eta) d\xi d\eta + \tau \int_c^{y_k} \frac{q_2^{(m-1)}(x_k, \eta) q_1^{(m)}(x_k, \eta)}{h^{(m-1)}(x_k, \eta)} d\eta + \tau \int_a^{x_k} \frac{q_2^{(m-1)}(\xi, y_k) q_2^{(m)}(\xi, y_k)}{h^{(m-1)}(\xi, y_k)} + \frac{gh^{(m-1)}(\xi, y_k) h^{(m)}(\xi, y_k)}{2} d\xi + \tau g \int_c^{y_k} \int_a^{x_k} \frac{\partial z(\xi, \eta)}{\partial \eta} h^{(m)}(\xi, \eta) d\xi d\eta + r_4(x) + s_4(y), \quad (4.11)$$

where

$$r_j(x) = \sum_{n=0}^{M-1} r_{j,n} R_n(x) \text{ and } s_j(y) = \sum_{n=0}^{N-1} s_{j,n} R_n(y), \quad (4.12)$$

where $j \in \{1, 2, 3, 4\}$ are arbitrary functions that emerged in the process of integration.

4. Rearrange each of the equations ((4.9), (4.10), (4.11)) into matrix form using the Chebyshev integration matrix:

We substitute all zero $(x_k, y_k), k \in \{1, 2, 3, \dots, H\}$ into (4.9) – (4.11), we have the

following simplified matrix equations

$$\mathbf{A}_x \mathbf{A}_y \mathbf{h}^{(m-1)} = \mathbf{A}_x \mathbf{A}_y \mathbf{h}^{(m)} + \tau \mathbf{A}_y \mathbf{q}_1^{(m)} + \tau \mathbf{A}_x \mathbf{q}_2^{(m)} + \Phi_x r_1 + \Phi_y s_1 + \Phi_x r_2 + \Phi_y s_2, \quad (4.13)$$

$$\begin{aligned} \mathbf{A}_x \mathbf{A}_y \mathbf{q}_1^{(m-1)} &= \mathbf{A}_x \mathbf{A}_y \mathbf{q}_1^{(m)} + \tau \mathbf{A}_y \mathbf{B}_1^{(m-1)} \mathbf{q}_1^{(m)} + \frac{\tau g}{2} \mathbf{A}_y \mathbf{H}^{(m-1)} \mathbf{h}^{(m)} \\ &\quad + \tau \mathbf{A}_x \mathbf{B}_1^{(m-1)} \mathbf{q}_2^{(m)} + \tau g \mathbf{A}_x \mathbf{A}_y \mathbf{Z}_x \mathbf{h}^{(m)} + \Phi_x r_3 + \Phi_y s_3, \end{aligned} \quad (4.14)$$

$$\begin{aligned} \mathbf{A}_x \mathbf{A}_y \mathbf{q}_2^{(m-1)} &= \mathbf{A}_x \mathbf{A}_y \mathbf{q}_2^{(m)} + \tau \mathbf{A}_y \mathbf{B}_2^{(m-1)} \mathbf{q}_1^{(m)} + \tau \mathbf{A}_x \mathbf{B}_2^{(m-1)} \mathbf{q}_2^{(m)} \\ &\quad + \frac{\tau g}{2} \mathbf{A}_x \mathbf{H}^{(m-1)} \mathbf{h}^{(m)} + \tau g \mathbf{A}_x \mathbf{A}_y \mathbf{Z}_y \mathbf{h}^{(m)} + \Phi_x r_4 + \Phi_y s_4, \end{aligned} \quad (4.15)$$

where \mathbf{A}_x and \mathbf{A}_y are the Chebyshev integration matrices defined in (2.10) and (2.12), respectively. Other parameters contained in (4.13), (4.14) and (4.15) are defined by

$$\begin{aligned} \mathbf{h}^{(m)} &= [h_1^{(m)}, h_2^{(m)}, h_3^{(m)}, \dots, h_H^{(m)}]^\top \text{ for } h_i^{(\cdot)} = h^{(\cdot)}(x_i, y_i), \\ \mathbf{q}_1^{(m)} &= [q_{11}^{(m)}, q_{12}^{(m)}, q_{13}^{(m)}, \dots, q_{1H}^{(m)}]^\top \text{ for } q_{1i}^{(\cdot)} = q_1^{(\cdot)}(x_i, y_i), \\ \mathbf{q}_2^{(m)} &= [q_{21}^{(m)}, q_{22}^{(m)}, q_{23}^{(m)}, \dots, q_{2H}^{(m)}]^\top \text{ for } q_{2i}^{(\cdot)} = q_2^{(\cdot)}(x_i, y_i), \\ \mathbf{Z}_x &= \text{diag}\{z_{x,1}, z_{x,2}, z_{x,3}, \dots, z_{x,H}\} \text{ for } z_{x,i} = z_x(x_i, y_i), \\ \mathbf{Z}_y &= \text{diag}\{z_{y,1}, z_{y,2}, z_{y,3}, \dots, z_{y,H}\} \text{ for } z_{y,i} = z_y(x_i, y_i), \\ \mathbf{B}_1^{(m-1)} &= \text{diag}\left\{\frac{q_{11}^{(m-1)}}{h_1^{(m-1)}}, \frac{q_{12}^{(m-1)}}{h_2^{(m-1)}}, \frac{q_{13}^{(m-1)}}{h_3^{(m-1)}}, \dots, \frac{q_{1H}^{(m-1)}}{h_H^{(m-1)}}\right\}, \\ \mathbf{B}_2^{(m-1)} &= \text{diag}\left\{\frac{q_{21}^{(m-1)}}{h_1^{(m-1)}}, \frac{q_{22}^{(m-1)}}{h_2^{(m-1)}}, \frac{q_{23}^{(m-1)}}{h_3^{(m-1)}}, \dots, \frac{q_{2H}^{(m-1)}}{h_H^{(m-1)}}\right\}, \\ \mathbf{H}^{(m-1)} &= \text{diag}\{h_1^{(m-1)}, h_2^{(m-1)}, h_3^{(m-1)}, \dots, h_H^{(m-1)}\}, \\ \mathbf{r}_j &= [r_{j,0}, r_{j,1}, r_{j,2}, \dots, r_{j,M-1}]^\top \text{ for } j \in \{1, 2, 3, 4\}, \\ \mathbf{s}_j &= [s_{j,0}, s_{j,1}, s_{j,2}, \dots, s_{j,N-1}]^\top \text{ for } j \in \{1, 2, 3, 4\}. \end{aligned}$$

From (4.12), we have

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

and

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

5. Transform the given boundary conditions into the matrix form:

We use the linear combination of Chebyshev polynomials (2.7) to convert the boundary conditions (3.4) into the matrix form, which can be divided into 2 cases:

(a) For the left and right boundary conditions, we fixed the value of x and varied the value of y , we have

$$q^{(m)}(x, y) = \sum_{n=0}^{M-1} c_n^{(m)} R_n(x) := \mathbf{p}_M(x) \mathbf{R}_M^{-1} \mathbf{q}^{(m)}(\cdot, y) = 0, \quad (4.16)$$

where $\mathbf{p}_M(x) = [R_0(x), R_1(x), R_2(x), \dots, R_{M-1}(x)]^\top$, \mathbf{R}_M is an $M \times M$ Chebyshev matrix. We substitute $y \in \{y_1, y_2, y_3, \dots, y_N\}$ which are the zeros of the Chebyshev polynomial $R_N(x)$ into (4.16) as follows:

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

It can be simplified to

$$(\mathbf{I}_N \otimes \mathbf{p}_M(x) \mathbf{R}_M^{-1}) \mathbf{q}^{(m)} = \mathbf{0},$$

for $x \in \{a, b\}$. Then, the left boundary is

$$\mathbf{P}_l \mathbf{q}^{(m)} := (\mathbf{I}_N \otimes \mathbf{p}_M(a) \mathbf{R}_M^{-1}) \mathbf{q}^{(m)} = \mathbf{0} \quad (4.17)$$

and the right boundary is

$$\mathbf{P}_r \mathbf{q}^{(m)} := (\mathbf{I}_N \otimes \mathbf{p}_M(b) \mathbf{R}_M^{-1}) \mathbf{q}^{(m)} = \mathbf{0}. \quad (4.18)$$

Remark that the left and the right boundaries are quite similar except that we evaluate the $\mathbf{p}_M(x)$ by a and b , respectively.

- (b) For the bottom and top boundary conditions, we fixed the value of y and varied the value of x , we have

$$\mathbf{q}^{(m)}(x, y) = \sum_{n=0}^{N-1} c_n^{(m)} R_n(x) := \mathbf{p}_N(x) \mathbf{R}_N^{-1} \mathbf{q}^{(m)}(x, \cdot) = 0, \quad (4.19)$$

where $\mathbf{p}_N(y) = [R_0(y), R_1(y), R_2(y), \dots, R_{N-1}(y)]^\top$, \mathbf{R}_N is an $N \times N$ Chebyshev matrix. We substitute $x \in \{x_1, x_2, x_3, \dots, x_M\}$ which are the zeros of the Chebyshev polynomial $R_M(x)$ into (4.19) as follows:

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

It can be simplified to

$$(\mathbf{I}_M \otimes \mathbf{p}_N(y) \mathbf{R}_N^{-1}) \mathbf{P}^{-1} \mathbf{q}^{(m)} = \mathbf{0},$$

for $y \in \{c, d\}$. Then, the bottom boundary is

$$\mathbf{P}_b \mathbf{q}^{(m)} := (\mathbf{I}_M \otimes \mathbf{p}_N(c) \mathbf{R}_N^{-1}) \mathbf{P}^{-1} \mathbf{q}^{(m)} = \mathbf{0} \quad (4.20)$$

and the top boundary is

$$\mathbf{P}_t \mathbf{q}^{(m)} := (\mathbf{I}_M \otimes \mathbf{p}_N(d) \mathbf{R}_N^{-1}) \mathbf{P}^{-1} \mathbf{q}^{(m)} = \mathbf{0}. \quad (4.21)$$

Remark that the bottom and the top boundaries are quite similar except that we evaluate the $\mathbf{p}_N(y)$ by c and d , respectively.

6. Construct the system of linear equations:

From (4.13) – (4.15), (4.17), (4.18), (4.20) and (4.21) which have a total number of $3H + 4M + 4N$ unknowns containing $\mathbf{h}^{(m)}$, $\mathbf{q}_1^{(m)}$, $\mathbf{q}_2^{(m)}$, r_1 , s_1 , r_2 , s_2 , r_3 , s_3 , r_4 and s_4 , we have

$$\left[\begin{array}{c|c} \Lambda_1 & \Lambda_2 \\ \hline \Lambda_3 & \mathbf{0}_{4M+4N} \end{array} \right] \begin{bmatrix} \mathbf{h}^{(m)} \\ \mathbf{q}_1^{(m)} \\ \mathbf{q}_2^{(m)} \\ \mathbf{r}_1 \\ \mathbf{s}_1 \\ \mathbf{r}_2 \\ \mathbf{s}_2 \\ \mathbf{r}_3 \\ \mathbf{s}_3 \\ \mathbf{r}_4 \\ \mathbf{s}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{A}\mathbf{h}^{(m-1)} \\ \mathbf{A}\mathbf{q}_1^{(m-1)} \\ \mathbf{A}\mathbf{q}_2^{(m-1)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \quad (4.22)$$



where

$$\Lambda_1 = \begin{bmatrix} \mathbf{A}_x \mathbf{A}_y & \tau \mathbf{A}_y & \tau \mathbf{A}_x \\ \frac{\tau g}{2} \mathbf{A}_y \mathbf{H}^{(m-1)} + \tau g \mathbf{A}_x \mathbf{A}_y \mathbf{Z}_x & \mathbf{A}_x \mathbf{A}_y + \tau \mathbf{A}_y \mathbf{B}_1^{(m-1)} & \tau \mathbf{A}_x \mathbf{B}_1^{(m-1)} \\ \frac{\tau g}{2} \mathbf{A}_x \mathbf{H}^{(m-1)} + \tau g \mathbf{A}_x \mathbf{A}_y \mathbf{Z}_y & \tau \mathbf{A}_y \mathbf{B}_2^{(m-1)} & \mathbf{A}_x \mathbf{A}_y + \tau \mathbf{A}_x \mathbf{B}_2^{(m-1)} \end{bmatrix}_{3H \times 3H},$$

$$\Lambda_2 = \begin{bmatrix} \Phi_x & \Phi_y & \Phi_x & \Phi_y & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \Phi_x & \Phi_y & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \Phi_x & \Phi_y & \mathbf{0} \end{bmatrix}_{3H \times (4M+4N)} \quad \text{and}$$

$$\Lambda_3 = \begin{bmatrix} \mathbf{0} & \mathbf{P}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_r & \mathbf{0} \\ \mathbf{P}_b & \vdots & \\ \mathbf{P}_t & \mathbf{0} & \\ \vdots & \mathbf{0} & \mathbf{P}_l \\ \mathbf{0} & \mathbf{P}_r & \\ \vdots & \mathbf{P}_b & \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_t \end{bmatrix}_{(4M+4N) \times 3H}$$

Finally, we have the system of linear equations (4.22), which can be solved to yield the approximate solutions $\mathbf{h}^{(m)}$, $\mathbf{q}_1^{(m)}$ and $\mathbf{q}_2^{(m)}$ by starting with the initial conditions (4.4) written in the vector forms $\mathbf{h}^{(0)} = [\phi_0(x_1), \phi_0(x_2), \phi_0(x_3), \dots, \phi_0(x_M)]^\top$, $\mathbf{u}^{(0)} = [\phi_1(x_1), \phi_1(x_2), \phi_1(x_3), \dots, \phi_1(x_M)]^\top$ and $\mathbf{v}^{(0)} = [\phi_2(x_1), \phi_2(x_2), \phi_2(x_3), \dots, \phi_2(x_M)]^\top$. Then, $\mathbf{q}_1^{(0)} = \mathbf{h}^{(0)} \odot \mathbf{u}^{(0)}$ and $\mathbf{q}_2^{(0)} = \mathbf{h}^{(0)} \odot \mathbf{v}^{(0)}$. Also, the solution $\mathbf{u}^{(m)}$ and $\mathbf{v}^{(m)}$ are directly obtained by $\mathbf{u}^{(m)} = \mathbf{q}_1^{(m)} \oslash \mathbf{h}^{(m)}$ and $\mathbf{v}^{(m)} = \mathbf{q}_2^{(m)} \oslash \mathbf{h}^{(m)}$, respectively.

Additionally, it is important to note the stability of this scheme similar to Chapter 3. The approximations obtained through this scheme will only converge to their analytical solution on a refined grid if the Courant-Friedrichs-Lewy condition in [9],

$$\tau = \text{CFL} \frac{\min_{i,j} (\Delta x_i, \Delta y_j)}{\max_k \left(|u_k^{(m-1)}| + \sqrt{gh_k^{(m-1)}}, |v_k^{(m-1)}| + \sqrt{gh_k^{(m-1)}} \right)} \quad (4.23)$$

is satisfied where CFL is the Courant number. To satisfy the stability, it is the same as

in Chapter 3 that the CFL must be less than one. For computational convenience, we provide the flowchart algorithm in Figure 4.1.

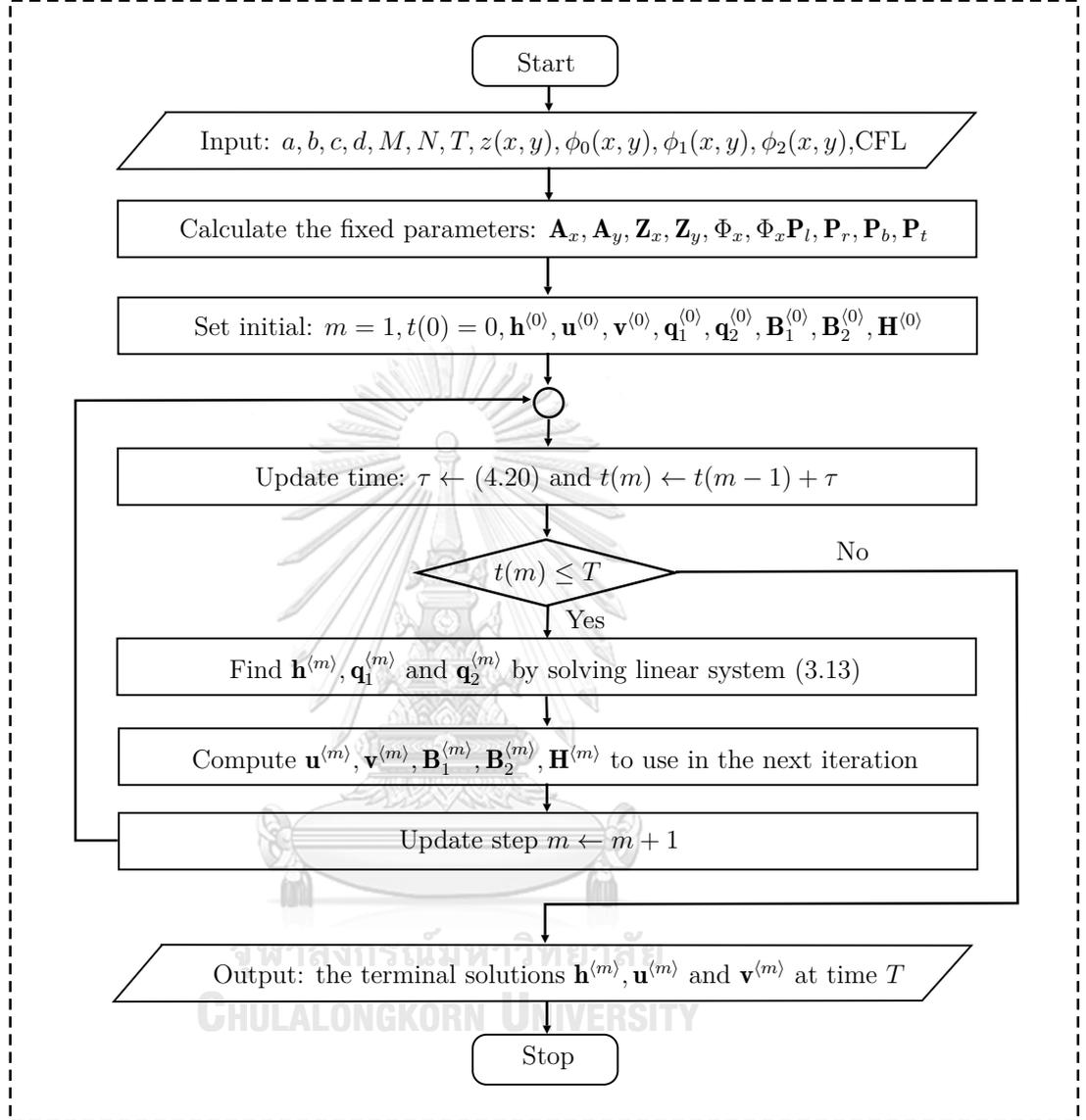


Figure 4.1: The flowchart for solving two-dimension SWEs

4.2 Numerical simulations

In this section, our proposed numerical algorithm has been investigated the efficiency and accuracy via two examples of two-dimensional SWEs with different types of bottom topography only on a wet bed. Thus, the experimental examples are Lake at

rest problems with nonflat bottoms and the Gaussian-shaped peak with flat bottoms as follows. In this study, we used an AMD Ryzen 7 4800HS CPU with 16.0 GB of RAM and Matlab software for all data processing and analysis.

Example 4.2.1 (Lake at rest [6]). The lake at rest is also usually used as a benchmark to verify numerical schemes for solving shallow water equations in two dimensions, see Figure 4.2 where the analytic solution is presented. The computational domain is $[0, 1] \times [0, 1]$. The bottom topography is defined by

$$z(x, y) = 0.8e^{-50((x-0.5)^2+(y-0.5)^2)}$$

and the initial water height is $h(x, y) = 1 - z(x, y)$ with initial zero velocity. We conduct the simulation on 30×30 grids using $T = 5$ s as the end time. The simulation results are stable over time, and at the end time, the errors in h, u and v are 2.4977×10^{-6} , 7.7669×10^{-4} and 7.7669×10^{-4} , respectively with a run time about 11.4308hrs. To evaluate the conservative of mass, we compute the total volume by using

$$\begin{aligned} & \frac{1}{4} \left[(y_2 - c) \left((x_2 - a) h_1^{<m>} + \sum_{i=1}^{M-2} (x_{i+2} - x_i) h_{i+1}^{<m>} \right) \right. \\ & + (d - y_{N-1}) \left((b - x_{M-1}) h_{MN}^{<m>} + \sum_{i=1}^{M-2} (x_{i+2} - x_i) h_{MN-M+i+1}^{<m>} \right) \\ & + (b - x_{M-1}) \left((y_2 - c) h_M^{<m>} + \sum_{j=1}^{N-2} (y_{j+2} - y_j) h_{jM+M}^{<m>} \right) \\ & + (x_2 - a) \left((d - y_{N-1}) h_{MN-M+1}^{<m>} + \sum_{j=1}^{N-2} (y_{j+2} - y_j) h_{jM+1}^{<m>} \right) \\ & \left. + \sum_{j=1}^{N-2} \sum_{i=1}^{M-2} (x_{i+2} - x_i) (y_{j+2} - y_j) h_{jM+i+1}^{<m>} \right] \end{aligned}$$

where x_1, x_2, \dots, x_M and y_1, y_2, \dots, y_N are nodal points, M, N are the number of nodal points along the x-axis and y-axis, respectively. $h_1^{<m>}, h_2^{<m>}, \dots, h_{MN}^{<m>}$ are the height of water at each time $t(m)$ form the simulation. By the experiment, the total water varies minimally, not exceeding 10^{-5} throughout the simulation period. Similarly

with Example 4.2.2.

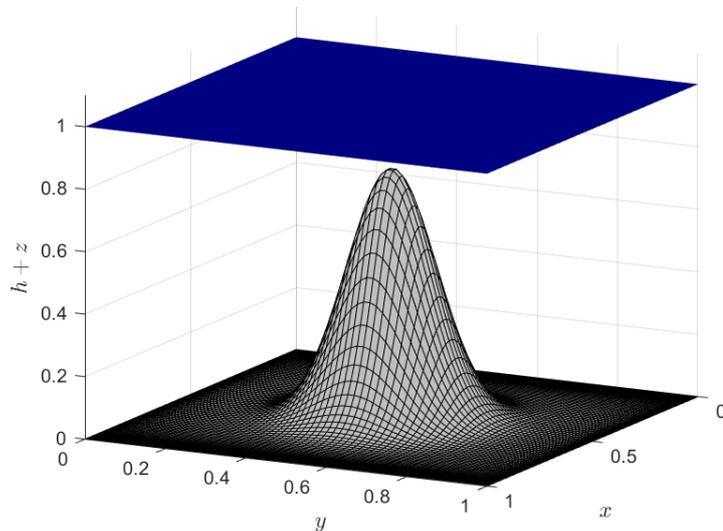


Figure 4.2: Analytical solutions

Example 4.2.2 (Gaussian pulse 2D [8]). We apply a 2D Gaussian-shaped peak as the initial condition for the water depth with zero velocity due to the flat topography or $z(x, y) = 0$ [11] which is defined by

$$h(x, y) = 1 + 0.1e^{-100((x-0.5)^2+(y-0.5)^2)}.$$

The computational domain is $[0, 1] \times [0, 1]$. The simulations for the time interval $[0, 0.25]$ are shown in Figure 4.3 with a run time about 32.9167mins. The wave moves out from the center, and all edges first reflect at the wall, followed by all corners. This causes the top view to change from a circle to a symmetrical shape with four corners. Furthermore, the water height decreases on average over time. The behaviors of water depth, denoted by $h(x, y, t)$, remain consistent with those observed in other schemes, see [8].

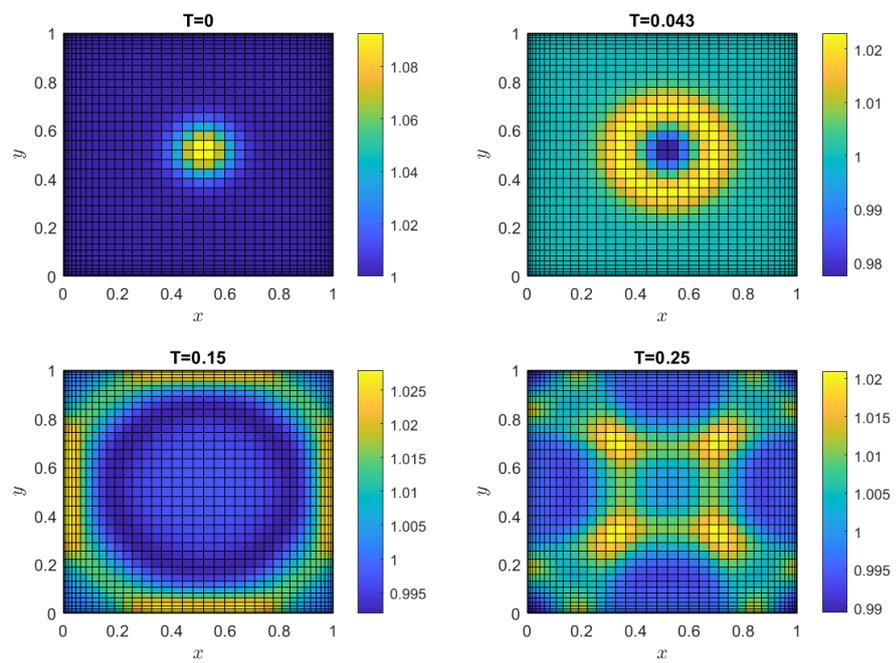


Figure 4.3: Water height $h(x,t)$ at various times(s).

CHAPTER V

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

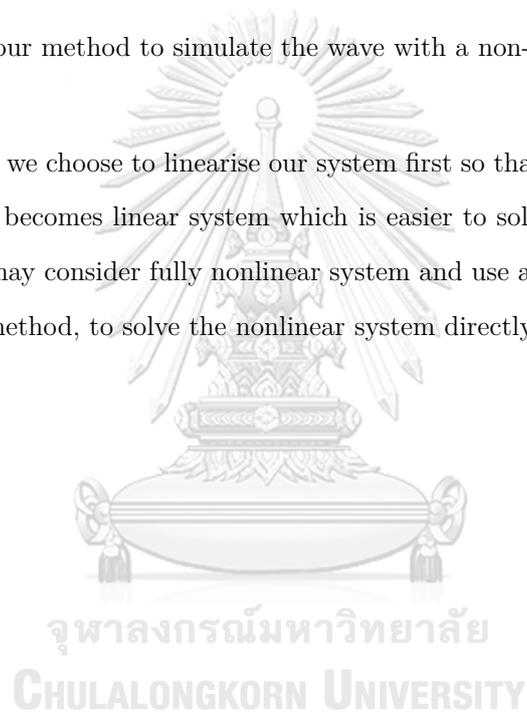
This study developed a numerical algorithm using the FIM-CPE for the spatial variable and the forward difference quotient for the temporal variable. The algorithm was employed to obtain precise solutions for the one- and two-dimensional SWEs over two bottom types of topography with reflecting boundary conditions (wall). Our proposed scheme was designed to ensure stability by incorporating the Courant-Friedrichs-Lewy condition. Additionally, flowcharts of the algorithms are presented in Figures 3.1 and 4.1 for one- and two-dimensional, respectively. In Chapter 3, we illustrate four experimental examples of the one-dimensional SWEs. Example 3.2.1 provides a measure of the accuracy of our method by evaluating how well it approximates the known exact solution for a static water body. Example 3.2.2 with a flat bottom shows that our algorithm provides more accurate results than FDM when using the same number of nodes M , as displayed in Table 3.1. Example 3.2.3 presents nonflat bottoms that have no analytical solutions. Our algorithm accurately approximates these solutions at different times T , as compared to other methods. Example 3.2.4 demonstrates that our algorithm effectively deals with smooth solutions. Furthermore, in Chapter 4, we present two experimental examples of the two-dimensional SWEs. Example 4.2.1 also assesses the accuracy of our approach by comparing the computed results with the exact solution for the static water body. Example 4.2.2 provides smooth solutions with flat bottoms for which there are no analytical solutions. Applying our algorithm to these examples yields solutions at different times T that display the characteristic features of the wave movement, in comparison to other methods.

5.2 Future Work

It is worth noting that this algorithm has some limitations. For instance, it cannot be applied to problems with an initial water height of zero due to the parameter $\mathbf{B}^{(m-1)}$ and cannot simulate large scales for two-dimensional problems. Hence, in future work, we hopefully expect that:

- Improve our method to simulate on the dry bed.
- Extend the two-dimensional scheme for the large-scale experiment.
- Improve our method to simulate the wave with a non-reflecting boundary.

Note that we choose to linearise our system first so that the nonlinearity disappear and the system becomes linear system which is easier to solve numerically. However, in the future, we may consider fully nonlinear system and use another numerical technique, e.g. Newton's method, to solve the nonlinear system directly.



REFERENCES

- [1] F. J. Caro-Lopera, V. Leiva, and N. Balakrishnan. Connection between the hadamard and matrix products with an application to matrix-variate birnbaum-saunders distributions. *J. Multivar. Anal.*, 104(1):126–139, 2012.
- [2] P. Crowhurst and Z. Li. Numerical solutions of one-dimensional shallow water equations. *15th International Conference on Modelling and simulation*, 55-60, 2013.
- [3] A. Duangpan and R. Boonklurb. Finite integration method using chebyshev expansion for solving nonlinear poisson equations on irregular domains. *J. Numer. Ind. Appl. Math.*, 14(1–2):7–24, 2020.
- [4] H. P. Gunawan. Numerical simulation of shallow water equations and related models. *Doctoral dissertation, Paris Est*, 2015.
- [5] J. Hudson. Numerical techniques for the shallow water equations. Master’s thesis, *University of Reading*, 1999.
- [6] K. Mekchay, T. Pongsanguansin, and M. Maleewong. Numerical methods based on discontinuous galerkin and finite volume methods for shallow water model and applications. Master’s thesis, *Chulalongkorn University*, 2016.
- [7] R. J. LeVeque. Balancing source terms and flux gradients in high-resolution godunov methods: the quasi-steady wave-propagation algorithm. *J. Comput. Phys.*, 146(1):346–365, 1998.
- [8] L. Lundgren. Efficient numerical methods for the shallow water equations. Master’s thesis, *Uppsala University*, 2018.
- [9] S. H. Peng. 1D and 2D numerical modeling for solving dam-break flow problems using finite volume method. *J. Appl. Math.*, 2012:1–14, 2012.

- [10] R. Reams. Hadamard inverses, square roots and products of almost semidefinite matrices. *Linear Algebra Appl.*, 288:35–43, 1999.
- [11] O. San and K. Kara. High-order accurate spectral difference method for shallow water equations. *Int. j. appl. sci.*, 6, 01 2011.
- [12] J. J. Stoker. *The Water Waves: The Mathematical Theory with Applications Book*. John Wiley & Sons, 2011.
- [13] H. Zhang and F. Ding. On the kronecker products and their applications. *J. Appl. Math.*, 2013:1–8, 2013.



BIOGRAPHY

Name Miss Lalita Apisornpanich

Date of Birth 4 February 1998

Place of Birth Yala, Thailand

Education B.Sc. (Mathematics)(First Class Honours),
Prince of Songkla University, 2020

Publication

- L. Apisornpanich and R. Boonklurb, Finite integration method with Chebyshev polynomial for solving one-dimensional shallow water equations, *Proceeding of Annual Meeting in Mathematics and Annual Pure and Applied Mathematics Conference* (2021), 53-67.