

เทคนิคการคิดค้นไม้แบบปรับตัวได้อย่างพลวัตสำหรับการจำลองน้ำท่วม

นายอนุรักษ์ บุญระมัญญ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต

สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2557

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR) are the thesis authors' files submitted through the Graduate School.

DYNAMICALLY ADAPTIVE TREE GRID TECHNIQUE  
FOR FLOOD SIMULATION

Mr. Anurak Busamun

A Dissertation Submitted in Partial Fulfillment of the Requirements  
for the Degree of Doctor of Science Program in  
Applied Mathematics and Computational Science  
Department of Mathematics and Computer Science  
Faculty of Science  
Chulalongkorn University  
Academic Year 2014  
Copyright of Chulalongkorn University

Thesis Title            DYNAMICALLY ADAPTIVE TREE GRID TECHNIQUE  
                                 FOR FLOOD SIMULATION  
By                            Mr. Anurak Busamun  
Field of Study            Applied Mathematics and Computational Science  
Thesis Advisor            Assistant Professor Khamron Mekchay, Ph.D.  
Thesis Co-Advisor       Associate Professor Suchada Siripant

---

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 Supot Hannongbua, Dr.rer.nat.)

THESIS COMMITTEE

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

.....Thesis Advisor  
(Assistant Professor Khamron Mekchay, Ph.D. )

.....Thesis Co-Advisor  
(Associate Professor Suchada Siripant )

.....Examiner  
(Associate Professor Pornchai Satravaha, Ph.D.)

.....Examiner  
(Kitiporn Plaimas, Dr.rer.nat.)

.....External Examiner  
(Somporn Chuai-Aree, Dr.rer.nat.)

อนุรักษ์ บุษะมัญญ : เทคนิคกริดต้นไม้แบบปรับตัวได้อย่างพลวัตสำหรับการจำลองน้ำท่วม  
(DYNAMICALLY ADAPTIVE TREE GRID TECHNIQUE FOR FLOOD SIMULATION)

อ. ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ.ดร. คำรณ เมฆฉาย, อ. ที่ปรึกษาวิทยานิพนธ์ร่วม: รศ.  
สุชาดา ศิริพันธุ์, 112 หน้า.

วัตถุประสงค์ของงานวิจัยนี้คือเพื่อพัฒนาขั้นตอนวิธีการคำนวณโดยใช้เทคนิคกริดต้นไม้แบบปรับตัวได้อย่างพลวัตสำหรับการจำลองน้ำท่วม ขั้นตอนวิธีดังกล่าวกระทำโดยการประยุกต์วิธีปริมาตรจำกัดสำหรับการหาผลเฉลยของสมการน้ำขึ้นและออกแบบวิธีการปรับกริดบนพื้นฐานของกริดต้นไม้สี่เหลี่ยมทั่วไป การทดสอบประสิทธิภาพเชิงการคำนวณของขั้นตอนวิธีดังกล่าวกระทำโดยการทดสอบและวัดผลแบบจำลองที่ประยุกต์ใช้ในการจำลองน้ำท่วมในประเทศไทย ผลการเปรียบเทียบผลลัพธ์กับข้อมูลการทดลองและผลลัพธ์จากการจำลองโดยใช้กริดละเอียดแสดงให้เห็นว่าแบบจำลองสามารถลดจำนวนของข้อมูลและเวลาในการคำนวณโดยสูญเสียความถูกต้องน้อยมากในผลลัพธ์ นอกจากนี้งานวิจัยนี้ยังทำการประยุกต์ขั้นตอนวิธีดังกล่าวไปสู่การหาผลเฉลยของแบบจำลองอย่างง่ายบนพื้นฐานของสมการการแพร่สำหรับการจำลองพื้นที่ความเสี่ยงของน้ำท่วมที่เกิดจากฝนตก ผลลัพธ์สามารถแสดงความถูกต้องของแบบจำลองอย่างง่ายที่เข้าสู่ผลลัพธ์ของแบบจำลองน้ำขึ้นและแสดงประสิทธิภาพของเทคนิคกริดต้นไม้แบบปรับตัวได้อย่างพลวัตสำหรับการจำลองพื้นที่ความเสี่ยงของน้ำท่วม

ภาควิชา ..คณิตศาสตร์..... ลายมือชื่อนิสิต.....  
..และวิทยาการคอมพิวเตอร์..

สาขาวิชา ..คณิตศาสตร์ประยุกต์..... ลายมือชื่อ อ.ที่ปรึกษาหลัก.....  
...และวิทยาการคณนา.....

ปีการศึกษา .....2557..... ลายมือชื่อ อ.ที่ปรึกษาร่วม.....

# # 5472903323 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORDS : SHALLOW WATER EQUATIONS/ DIFFUSION EQUATION/ HYDROSTATIC RECONSTRUCTION/ ADAPTIVE GRIDS/DYNAMIC DOMAIN DEFINING METHOD

ANURAK BUSAMUN : DYNAMICALLY ADAPTIVE TREE GRID TECHNIQUE FOR FLOOD SIMULATION. ADVISOR : ASST.PROF. KHAMRON MEKCHAY, Ph.D., CO-ADVISOR : ASSOC.PROF. SUCHADA SIRIPANT, 112 pp.

The purpose of this research is to develop a computational algorithm with using adaptive tree grid technique for flood simulation. For the algorithm, we applied the finite volume method to solve the shallow water equations, and we designed the grid adaptivity method based on general rectangular tree grids. In order to test a potential of the algorithm in terms of computational costs, the model is tested and applied to simulate flood in Thailand. The results are compared with experimental data and the simulation on uniform grids. The comparison shows that the model can reduce number of grid cells and computational times without losing much accuracy in the results. Moreover, the algorithm is applied to simulate a simple model based on a diffusion equation for flood risk area assessment occurred by rainfall. The results show the accuracy of the simple model that converges to the shallow water model, and show the efficiency of the adaptive grid technique for the flood risk area assessment.

Department : ..Mathematics..... Student's Signature : .....  
 ..and Computer Science.....

Field of Study : ..Applied Mathematics.. Advisor's Signature : .....  
 ..and Computational Science..

Academic Year : ..2014..... Co-Advisor's Signature : .....

## ACKNOWLEDGEMENTS

First of all, I would like to express my deep gratitude to my advisor, Assist. Prof. Dr. Khamron Mekchay, who has very kindly read and edited the dissertation and offered many useful suggestions. I would also like to thank my co-advisor, Assoc. Prof. Suchada Siripant, who has given me wonderful suggestions.

My thanks also go to the dissertation committees, Assoc. Prof. Dr. Anusorn Chonwerayuth, Assoc. Prof. Dr. Pornchai Satravaha and Dr. Kitiporn Plaimas, for their helpful comments on the suitability of dissertation contents.

I want to express my gratitude to Chulalongkorn University and Research Professional Development Project under the Science Achievement Scholarship of Thailand for the scholarship and supports during my study.

I would also like to thank Dr. Somporn Chuai-aree who has supported and advised me for modeling, simulation and visualization of flood. I would like to thank the Advanced Virtual and Intelligence Computing Research Center (AVIC) for supporting me to accomplish research, and thanks especially to all of my friends in the program of Applied Mathematics and Computational Science for providing me constant encouragement.

Finally, my deepest gratitude goes to my family for their encouragement, love and care that inspire the research.

# CONTENTS

	Page
ABSTRACT (THAI) . . . . .	iv
ABSTRACT (ENGLISH) . . . . .	v
ACKNOWLEDGEMENTS . . . . .	vi
LIST OF FIGURES . . . . .	x
CHAPTER	
1 INTRODUCTION . . . . .	1
1.1 Ideas of Research . . . . .	2
1.2 Objectives of Research . . . . .	4
1.3 Operation of Research . . . . .	5
1.4 Organization of Dissertation . . . . .	5
2 DERIVATION OF THE MODEL BASED ON SHALLOW WATER EQUATIONS . . . . .	6
2.1 Conservation of Mass . . . . .	6
2.2 Conservation of Momentum . . . . .	8
3 DYNAMICALLY ADAPTIVE GRID ALGORITHM FOR FLOOD SIMULATION . . . . .	13
3.1 Tree Grids . . . . .	13
3.2 Numerical Algorithm . . . . .	14
3.2.1 Finite volume formulation . . . . .	15
3.2.2 Numerical flux . . . . .	16
3.2.3 Reconstruction procedure . . . . .	17
3.2.4 Total variation diminishing . . . . .	18
3.2.5 Source term computation . . . . .	19
3.2.6 Stability condition . . . . .	19
3.2.7 Boundary condition . . . . .	20
3.2.8 Grid adaptivity method . . . . .	20

CHAPTER	Page
3.2.9 Topography interpolation . . . . .	22
3.2.10 Dynamic domain defining method . . . . .	23
3.2.11 Algorithm overview . . . . .	23
3.3 Experimental Results . . . . .	25
3.3.1 2D Dam break simulation . . . . .	26
3.3.2 2D Parabolic bowl simulation . . . . .	27
3.3.3 Rainfall simulation on sloped plane . . . . .	30
3.4 Conclusion . . . . .	32
4 FLOOD SIMULATIONS USING THE DYNAMICALLY ADAPTIVE TREE GRID TECHNIQUE . . . . .	33
4.1 Rain Water Overland Flow . . . . .	33
4.2 Dam-Break Flood . . . . .	39
4.3 Breaching Flood Inundation . . . . .	45
4.4 Conclusion . . . . .	52
5 SIMPLIFICATION OF THE DYNAMICALLY ADAPTIVE TREE GRID ALGORITHM . . . . .	53
5.1 Numerical Algorithm . . . . .	53
5.1.1 Finite volume formulation . . . . .	54
5.1.2 Boundary condition . . . . .	55
5.1.3 Grid adaptivity method . . . . .	55
5.1.4 Dynamic domain defining method . . . . .	56
5.1.5 Algorithm overview . . . . .	56
5.2 Results . . . . .	58
5.2.1 Dam break experiment simulation . . . . .	58
5.2.2 Dam break flow on natural topography . . . . .	62
5.3 Conclusion . . . . .	68



CHAPTER	Page
6 DIFFUSION MODELING ON DYNAMICALLY ADAPTIVE TREE GRID FOR RAINFALL OVERLAND FLOW SIMULATION . . . . .	69
6.1 The Mathematical Model . . . . .	69
6.2 Numerical Algorithm . . . . .	70
6.2.1 Finite volume formulation . . . . .	70
6.2.2 Numerical flux . . . . .	71
6.2.3 Stability condition . . . . .	72
6.2.4 Boundary condition . . . . .	72
6.2.5 Grid adaptivity method . . . . .	73
6.2.6 Topography interpolation . . . . .	73
6.2.7 Dynamic domain defining method . . . . .	74
6.2.8 Algorithm overview . . . . .	75
6.3 Results . . . . .	76
6.3.1 Accuracy and efficiency of the diffusion model . . . . .	77
6.3.2 Accuracy and efficiency of the adaptive scheme . . . . .	80
6.4 Conclusion . . . . .	83
7 CONCLUSION . . . . .	85
7.1 Dynamically Adaptive Tree Grid Algorithm for Shallow Water Model	85
7.2 Simplification of The Dynamically Adaptive Tree Grid Algorithm .	86
7.3 Diffusion Modeling on Dynamically Adaptive Tree Grid . . . . .	87
7.4 Future Works and Recommendation . . . . .	87
REFERENCES . . . . .	89
APPENDIX. . . . .	93
VITA. . . . .	98

## LIST OF FIGURES

FIGURE	Page
2.1 The control volume (left); the directions of all fluxes and the added rain-water (right). . . . .	7
2.2 The directions of all forces in x-direction. . . . .	8
3.1 Regularized tree grids. . . . .	13
3.2 Hierarchical data structure. . . . .	13
3.3 The main control volume and descriptions of numerical fluxes. . . .	15
3.4 The algorithm overview. . . . .	24
3.5 The initial grid cells of the rectangular dam break. . . . .	27
3.6 The 3D representation (left) and the adaptive grid (right) of the dam break simulation at $t = 5s$ . . . . .	27
3.7 Water surface profiles along the center-line in the x-direction of the simulations (solid blue line) compared with the analytic solution (dotted red line) for the 2D parabolic bed topography (solid green line) at $t = 30.3532$ s, $671.7932$ s, $1358.1300$ s, and $2503.5160$ s (from left to right and top to bottom).The small pictures show some zooms at the wet/dry transition for the simulations. . . . .	29
3.8 The evolution of the velocities of the model compared with analytic solutions at $x = 6000$ m. . . . .	29
3.9 The water profiles along the center-line in the x-direction of the simulations at 10 s (red), 20 s (green) and 30 s (blue) compared with Iwagaki's experimental and calculated (theoretical) data. . . .	30
3.10 The evolutions of the discharges at the downstream with rainfall durations: 10 s (red), 20 s (green) and 30 s (blue), compared with Iwagaki's experimental and calculated (theoretical) data. . . . .	31
3.11 The evolutions of the water depth at the downstream with rainfall durations: 10 s (red), 20 s (green) and 30 s (blue), compared with Iwagaki's experimental and calculated (theoretical) data. . . . .	31

FIGURE	Page
4.1 The 3D simulations of the behavior of the rain-water overland flow at different times $t$ . . . . .	35
4.2 The 3D representations of the adapted grids of the rain-water overland flow simulations at different times $t$ . . . . .	36
4.3 The 3D simulations of the rain-water overland flow based on the adaptive grid (a), uniform level 4 grid (small cells) (b), and uniform level 1 grid (large cells) (c) at $t = 1800$ s. . . . .	37
4.4 The comparisons of water depths (left) and discharges (right) at (16710 m, 21060 m) for the adaptive grid (solid), uniform level 4 grid, small cells (dashed), and uniform level 1 grid, large cells (dotted). . . . .	37
4.5 The comparisons of number of computational cells (left) and times (right) for adaptive grid (solid), uniform level 4 grid, small cells (dashed). . . . .	38
4.6 The comparisons of number of computational cells (left) and times (right) using adaptivity with dynamic DDM (solid) and without dynamic DDM (dashed). . . . .	38
4.7 The dam location and the measure points of the Mae-suai dam-break simulation. . . . .	39
4.8 The initial grid of the Mae-suai dam-break simulation. . . . .	39
4.9 The 3D representation of the adapted grids of the dam-break overland flow simulation, at different times. . . . .	40
4.10 The 3D representation of the comparison for the adaptive grid (left) and the uniform finest grid (right) of the Mae-suai dam-break simulation at time=3600 s. . . . .	41
4.11 The comparison of the evolutions of the water depth at the measure points P1, P2, P3, P4, P5, P6 (from left to right and top to bottom). The solid, dotted and dashed lines refer to the simulations of the adaptive grid with $\epsilon_1 = 1.0$ and $\epsilon_2 = 1.0$ , the adaptive grid with $\epsilon_1 = 1.0$ and $\epsilon_2 = 0.5$ , and with the finest grid, respectively. . . . .	42

FIGURE	Page
4.12 The comparisons of the arrival times (left) and the maximum water depth (right) at each locations between the adaptive grid and the uniform finest grid. . . . .	43
4.13 The comparisons of the number of computational cells (left) and the computational times (right). The solid, dotted and dashed lines refer the simulations of the adaptive grid with $\epsilon_1 = 1.0$ and $\epsilon_2 = 1.0$ , the adaptive grid with $\epsilon_1 = 1.0$ and $\epsilon_2 = 0.5$ , and the finest grid, respectively. . . . .	43
4.14 The breach location and the measured points of the breaching flood inundation simulation. . . . .	47
4.15 The initial grid of the breaching flood inundation simulation. . . .	47
4.16 The idealist hydro graph defined at the breach to drive the flood event. . . . .	48
4.17 Flood inundation at t=2 h: (a) the flood simulation by the adaptive grid, (b) the flood simulation by the finest grid, and (c) the adapted tree grid. . . . .	49
4.18 Flood inundation at t=12 h: (a) the flood simulation by the adaptive grid, (b) the flood simulation by the finest grid, and (c) the adapted tree grid. . . . .	50
4.19 The comparisons of the total volume of water inside the computational domain (model predictions) and the total volume of mass through the breach (inflow volumes). . . . .	51
4.20 The comparisons of arrival times (left) and maximum water depths (right) of each the locations for the adaptive grid and the uniform finest grid. . . . .	51
4.21 The comparisons of number of computational cells (left) and computational times (right) for the adaptive grid and the uniform finest grid. . . . .	51
5.1 The main control volume and descriptions for each cell. . . . .	54

FIGURE	Page
5.2 The algorithm overview. . . . .	57
5.3 The geometry of the physical experiment of Kleefsman et al. . . . .	59
5.4 The initial grid of the dam-break simulation. . . . .	59
5.5 The 3D representations and 2D adaptive grids of the dam break simulation compared with experiment at different times. . . . .	60
5.6 The comparisons of the evolutions of the water depths at the reservoir H4 (top-left), H3 (top-right), H2 (bottom-left), and H1 (bottom-right), for the experimental data (dotted), the adaptive grid second order scheme (dashed), and the adaptive grid first order scheme (solid). . . . .	61
5.7 The comparisons of the number of computational cells (left) and the computational times (right) for the adaptive grid second order scheme (dashed) and the adaptive grid first order scheme (solid). . . . .	61
5.8 The evaluated locations of the Malpasset dam-break simulation. . . . .	63
5.9 The initial grid of the Malpasset dam-break simulation. . . . .	63
5.10 The 3D adaptive grid representation of the dam-break simulations at different times: 10 s, 100 s, 150 s, 300 s, 600 s, and 3000 s, respectively. . . . .	64
5.11 The comparisons of the adaptive grid first order scheme simulations (left) with the finest uniform grid simulations (right) at different times $t$ . . . . .	65
5.12 The comparison of the evolutions of the water depth in the measured points P1, P2, P3, P4, P5, P6 (from left to right and top to bottom). The solid, dotted and dashed lines refer to the simulations of the adaptive grid first order scheme, the adaptive grid second order scheme, and the finest grid second order scheme, respectively. . . . .	66

FIGURE	Page
5.13 The comparisons of the arrival times (left) and the maximum water depths (right) at each location for the adaptive grid and the uniform finest grid. . . . .	67
5.14 The comparisons of the number of computational cells (left) and the computational times (right) for the adaptive grid second order scheme and the adaptive grid first order scheme. . . . .	67
6.1 The main control volume and descriptions for each cell. . . . .	70
6.2 The algorithm overview. . . . .	75
6.3 The 3D representation of the rainfall overland flow simulation using the shallow water equations (left) and the diffusion model (right). . . . .	78
6.4 The left picture is the errors of the diffusion model compared with the shallow water equations, and the right picture is the comparison of the computational times for the simulation of the diffusion model (solid) and the shallow water equations (dashed). . . . .	79
6.5 The 3D representation of the adapted grids of the rain water overland flow simulation for each time. . . . .	81
6.6 The 3D representation of the adapted grids of the rain water overland flow simulation for each time. . . . .	82
6.7 The 3D representation of the comparison for adaptive grids (a), uniform coarsen grids (b), and uniform finest grids (c) of the rain water overland flow simulation at time=53 h. . . . .	83
6.8 The comparisons of the number of computational cells (left) and the computational times (right) of the rain water overland flow simulation for the adaptive grid (solid), the uniform finest grid (dashed), and the uniform coarsen grid (dotted). . . . .	84

# CHAPTER I

## INTRODUCTION

Water flooding has often occurred in the world. They caused many related problems such as flooding, landslide, disease, and other problems. The situations have occurred by continuously heavy rain, dam-break, breaching of flood defence etc., and it damaged humans and economies. The problems have necessitated researchers to do numerical modeling of flood inundation which is an important tool for a risk assessment, preparation, evacuation planning, and real-time forecasting of flood warning. Moreover, the simulation could also be used to analyze water management, since it can reproduce information on water depth, water speed, and flow directions in any regions.

In general, simulation of floods model involves numerical methods with high computational demand for solving shallow water equations. The shallow water model has been first introduced in 1D by Saint-Venant [13], and Zhang and Cundy [46] are the first to use shallow water equations for rain-water overland flow equations. Fiedler and Ramirez [14] are the first to use finite volume methods, and Delestre and team [9, 10] are the first to use well-balanced schemes for rain-water overland flow modeling. The other related applications for hyperbolic problems are dam-break [15, 30], tsunami [16, 34], river flow [17], and tides [37].

The well-balanced property, where the flux gradients is balanced with the source terms for steady state solutions, is very important for numerical schemes to avoid oscillated steady state solution. Bermúdez and Vázquez [6] are the first to mention the problem of balance between the pressure term and the topography source term, and Greenberg and LeRoux [19] are the first that developed well-balanced schemes to avoid this problem.

However, several flood models (see [12, 38, 44]) have been developed and successfully applied using simple equations and reduced complexity approaches. The

motivation of the flood models is that solving the simple equations should reduce the computational burden and the simulation run times (Dottori [12]).

Since topography has an important influence on the water overland flow model, using of FVM requires a large amount of digital terrain elevation data grid cells to improve accuracy of a simulation, and in practice, the computational cost is very expensive. This needs attention for designing algorithms and schemes for reducing the number of computed cells without losing much accuracy in the results. One solution is to use adaptive grid methods where the methods continuously adjust grid resolutions to follow features in the flow. By adaptive techniques, Liang [30] solved 2D shallow water equations with dynamically adaptive quad tree grids using FVM for dam and dyke break simulations, Popinet [34] developed a quad tree adaptive solver for tsunami modeling with 2D shallow water equations using a generalized well-balanced positivity-preserving scheme of Audusse et al [3], and An and Yu [2] used hydrostatic reconstruction and quad tree grid to model urban flooding. George [15] solved 2D shallow water equations with well-balanced high-resolution finite volume methods and block-structured dynamic adaptive mesh refinement (AMR) for modeling floods in rugged terrain, and applied to the Malpasset dam-break flood (France, 1959). Moreover, Schreiber [40] presented a method for solving hyperbolic partial differential equations based on dynamically adaptive triangular grid.

An alternative technique is the dynamic domain defining method (dynamic DDM) described by Yamaguchi [45] for the development of GIS-Based flood simulation software for flood-risk assessment, where during the simulation the calculation area is adjusted (expanded or shrunk) to exclude dry grid cells.

## 1.1 Ideas of Research

The basic issues for developing algorithms for flood simulations can be stated as in the following ideas:

1. Since most of the floods models have been developed and successfully applied using shallow water equations which is often solved by the finite volume methods,



we would like to develop floods model based on the shallow water equations and the finite volume methods. The well-balanced scheme with hydrostatic reconstruction of Audusse will be adopted in this work to preserve the non-negative of the water depth and conserve the total water depth, including the ability to compute dry states.

2. In general, the dynamic DDM can reduce the computational time, however, it can still increase, if the calculation area is continuously expanded. This needs attention to utilize an adaptive grid method with the dynamic DDM; an adaptive grid method can reduce the number of grid cells by merging the grid cells in some calculation area, while the dynamic DDM can automatically define the calculation area to exclude unnecessary cells.
3. For the adaptive grid method, two adaptive grid techniques were found in literature, the block adaptivity and the tree grid adaptivity. The block adaptivity [5] uses dynamic embedding of Cartesian meshes of different resolutions, which works well when relatively large blocks of uniform resolutions can be combined to track medium- to fine-scale features in the flow. Based on a tree hierarchical structure, the tree grid adaptivity keeps Cartesian cells as basic discretization volumes and allows neighboring cells with different spatial resolutions. Through a generalization of the basic discrete operators (gradient, divergence, etc.), solvers can be constructed to work across resolution boundaries. The tree grid adaptivity allows a more flexible tracking of flow features than the block adaptivity (Popinet [34]). Therefore, this work is developed based on the adaptive grid method using the tree hierarchical structure.
4. For the adaptive tree grid method, most of recent researches as have performed modeling using adaptivity on regularized quad tree grids, where each parent cell has zero or four children and is balanced in the sense that the levels of adjacent cells cannot differ more than one. In this work, we attempt to design the algorithm and numerical schemes for the adaptivity based on general rectangular tree grids.

5. The previous adaptive tree grid models used second order high-resolution finite volume schemes in terms of time and space. However, in the simulation of the flood flow over natural topography, the second order high-resolution schemes may not be significant when compared with uncertainties in topographic data and information available for calibration of friction parameterization. If first order schemes are implemented, the adaptive tree grid method should provide further significant improvements in computational efficiencies (Liang [31]). This becomes attention for the development of the adaptive tree grid method with first order finite volume schemes. The main motivation of the attention is simple schemes should reduce the computational resources and the simulation run times. Moreover, since validations of the first order schemes with the adaptive tree grid method have been ignored by other researchers, it becomes attention to validate the scheme for the flood simulation.
6. Since flood simulation using simple equations and with reduced complexity approaches are efficiency in terms of computational costs, it becomes attention to develop the simple model for floods risk assessment. The diffusion and wave equations can also be to describe water or wave propagations, (see [1, 8, 38]), however, they have not been used to describe the water flow and wave propagations in rugged terrains. In this work, we will develop a model using the diffusion equation for flood risk areas assessment occurred after continuous heavy rainfall.

## 1.2 Objectives of Research

The objectives of this research can be described as follows:

1. To develop algorithms to simulate and visualize floods on natural topography based on finite volume methods for shallow water equations, by modifying the adaptive grid technique developed in [2, 30, 34] for rectangular quad tree grids, and combining with the dynamic DDM. Here, the idea of quad tree grids is extended for general tree grids to handle natural topography data in general.

2. To apply the developed algorithms to simulate and visualize flood risk areas based on the diffusion model for water flow.

### **1.3 Operation of Research**

The details of operation of the research are the following:

1. Study previous works about finite volume methods for shallow water equations, numerical schemes and algorithms based on adaptive tree grid methods, the dynamic DDM, and other models for dynamic system.
2. Develop computational algorithms for shallow water equations based on the idea of the dynamically adaptive general tree grid and the dynamic DDM.
3. Develop the software to simulate and visualize floods on natural topography. The software is programmed using Delphi Version 7 with OpenGL library.
4. Check the accuracy and efficiency of the developed algorithms and softwares.
5. Develop a simple algorithm based on the diffusion equation.
6. Write papers and dissertation of the research.

### **1.4 Organization of Dissertation**

The rest of the dissertation is organized as follows. Chapter 2 presents the derivation of the flood model based on shallow water equations. The designing of the computational algorithm of FVM based on the adaptive grids and dynamic DDM are explained in Chapter 3. Also presented in Chapter 3 are some experiments to check the accuracy of the algorithm. The numerical simulations and results for floods on natural topography are shown in Chapter 4. Chapter 5 describes simplification of the dynamically adaptive grid algorithm using a first order finite volume scheme, while Chapter 6 presents the developed algorithm to simulate and visualize floods based on the diffusion model. The conclusion of the dissertation is in Chapter 7.

## CHAPTER II

### DERIVATION OF THE MODEL BASED ON SHALLOW WATER EQUATIONS

In this research, the algorithm for simulation and visualization of floods are developed for rain-water overland flows. A process of the rain-water overland flow in general is very complex. The process occurs when the rainfall intensity exceeds soil infiltration, and water begins to accumulate on the surface, and then flows down slopes under gravity as overland flow. In order to reduce the complexity process of simulation, we consider only a model that consists of rainfall intensity, where the infiltration is very little compared to the volume of the flow and can be ignored in addition to other processes such as saturation excess, erosion, evaporation, subsurface flow. The model is developed based on shallow water equations for determining the behavior of rain-water flow above bottom elevation without infiltration.

The derivation of equations governed the flow is considered as shallow water problem. The usual approach that simplifies the mathematical description of the problem is to use a depth averaging procedure of the Navier-Stokes equations. This leads to the Shallow Water Equations (SWE) model. Alternatively, the SWE can be derived from the basic principles of conservations of mass and momentum together with a set of constitutive laws related to the driving and resisting forces of fluid properties and motion (De Saint-Venant [13]). The derivation of the model system is presented as follow.

#### 2.1 Conservation of Mass

Consider a small rectangular element ( $\Delta x \times \Delta y \times h$ ) of water when  $h(x, y, t)$  is water depth as shown in Figure 2.1. Conservation of mass for this control volume

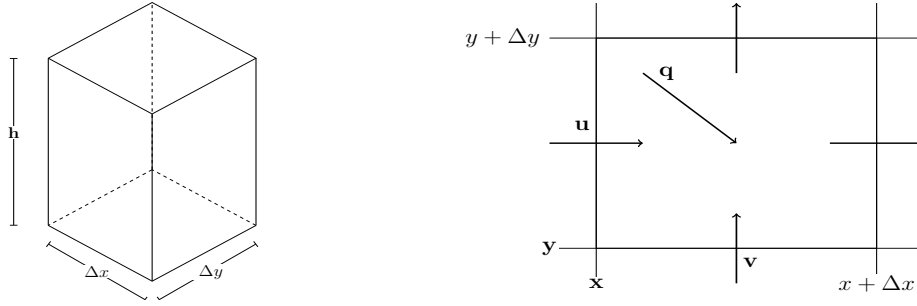


Figure 2.1: The control volume (left); the directions of all fluxes and the added rain-water (right).

or box states that:

The rate of volume increase in the control volume is equal to the net volume flux into the control volume from all 4 sides plus a net rain-water volume that is vertically added to the control volume per unit time.

To formulate the equations for the model, let  $u$  and  $v$  be velocity in the  $x$  and  $y$  directions, and  $q = q(x, y, t)$  be the rainfall rate added to control volume at time  $t$ .

Since  $\Delta x$  and  $\Delta y$  are fixed, the volume of water in the box can change only if the depth changes. The corresponding rate of change of volume in the box is

$$\Delta x \Delta y \frac{\partial h}{\partial t}. \quad (2.1)$$

Along the boundary, the net volume flux along  $x$ -direction is

$$(hu|_x - hu|_{x+\Delta x})\Delta y. \quad (2.2)$$

Using Taylor's series expansion and omitting terms of higher orders in  $\Delta x$ , equation (2.2) becomes

$$- \Delta x \Delta y \frac{\partial uh}{\partial x}. \quad (2.3)$$

Similarly, the net volume flux long  $y$ -direction is

$$- \Delta x \Delta y \frac{\partial vh}{\partial y}. \quad (2.4)$$

The net rain-water volume is

$$\Delta x \Delta y q. \quad (2.5)$$

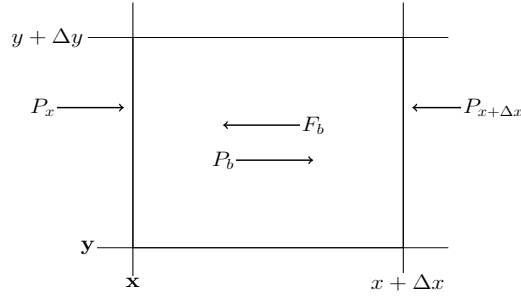


Figure 2.2: The directions of all forces in x-direction.

Equating these terms, the conservation of mass can then be expressed quantitatively as

$$\Delta x \Delta y \frac{\partial h}{\partial t} = -\Delta x \Delta y \frac{\partial uh}{\partial x} - \Delta x \Delta y \frac{\partial vh}{\partial y} + \Delta x \Delta y q. \quad (2.6)$$

Dividing Equation (2.6) with  $\Delta x \Delta y$ , the conservation of mass can be described as

$$\frac{\partial h}{\partial t} + \frac{\partial uh}{\partial x} + \frac{\partial vh}{\partial y} = q. \quad (2.7)$$

## 2.2 Conservation of Momentum

Conservation of momentum in the x-direction is stated as follow:

The rate of change of momentum in the x-direction in the control volume is equal to the net influx of momentum through vertical wall plus net forces acting on the control volume in the x-direction.

The rate of change of momentum is

$$\rho \Delta x \Delta y \frac{\partial uh}{\partial t}. \quad (2.8)$$

The net influx of momentum through four vertical sides is

$$-\rho \Delta x \Delta y \frac{\partial u^2 h}{\partial x} - \rho \Delta x \Delta y \frac{\partial vuh}{\partial y}. \quad (2.9)$$

Figure 2.2 depicts the directions of all forces, which are:

1.  $P_x$  and  $P_{x+\Delta x}$  are pressure forces on the sides of the box;
2.  $P_b$  is the pressure force due to a sloping bed;
3.  $F_b$  is the friction force at the channel bottom.

The net of pressure force on two vertical sides normal to x-direction is

$$\int P\Delta A|_x - \int P\Delta A|_{x+\Delta x}. \quad (2.10)$$

From the pressure  $P = \rho gh$ , the net hydrostatic pressure force becomes

$$-\frac{\rho g \Delta x \Delta y}{2} \frac{\partial h^2}{\partial x}. \quad (2.11)$$

The pressure force due to the sloping bed (or the gravity force) is

$$P_b = -\rho g \Delta x \Delta y h S_{0x}, \quad (2.12)$$

where  $S_{0x}$  is the bed slope in the x-direction.

The resisting force on the bottom can be expressed as shear stresses multiplied by surface area as

$$F_b = \tau_{bx} \Delta x \Delta y. \quad (2.13)$$

Equating these terms, the conservation of momentum in the x-direction becomes

$$\begin{aligned} \rho \Delta x \Delta y \frac{\partial uh}{\partial t} &= -\rho \Delta x \Delta y \frac{\partial u^2 h}{\partial x} - \rho \Delta x \Delta y \frac{\partial vuh}{\partial y} \\ &\quad - \frac{\rho g \Delta x \Delta y}{2} \frac{\partial h^2}{\partial x} - \rho g \Delta x \Delta y h S_{0x} - \tau_{bx} \Delta x \Delta y. \end{aligned} \quad (2.14)$$

Dividing equation (2.14) by  $\rho \Delta x \Delta y$ , we obtain

$$\frac{\partial uh}{\partial t} = -\frac{\partial u^2 h}{\partial x} - \frac{\partial vuh}{\partial y} - \frac{g}{2} \frac{\partial h^2}{\partial x} - gh S_{0x} - \frac{\tau_{bx}}{\rho}, \quad (2.15)$$

which can be written as

$$\frac{\partial uh}{\partial t} + \frac{\partial u^2 h}{\partial x} + \frac{g}{2} \frac{\partial h^2}{\partial x} + \frac{\partial uvh}{\partial y} = -gh S_{0x} - gh S_{fx}, \quad (2.16)$$

where  $S_{fx} = \frac{\tau_{bx}}{\rho gh}$  is the friction slope in the x-direction.

Similarly, the conservation of momentum in the y-direction gives

$$\frac{\partial vh}{\partial t} + \frac{\partial uvh}{\partial x} + \frac{\partial v^2 h}{\partial y} + \frac{g}{2} \frac{\partial h^2}{\partial y} = -gh S_{0y} - gh S_{fy}, \quad (2.17)$$

where  $S_{fy} = \frac{\tau_{by}}{\rho gh}$  is the friction slope in the y-direction and  $S_{0y}$  is the bed slope in the y-direction.

Equations (2.7), (2.16) and (2.17) are called the two-dimensional shallow water equations. For future references, the SWE are

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial uh}{\partial x} + \frac{\partial vh}{\partial y} &= q, \\ \frac{\partial uh}{\partial t} + \frac{\partial u^2 h}{\partial x} + \frac{g}{2} \frac{\partial h^2}{\partial x} + \frac{\partial uvh}{\partial y} &= -ghS_{0x} - ghS_{fx}, \\ \frac{\partial vh}{\partial t} + \frac{\partial uvh}{\partial x} + \frac{\partial v^2 h}{\partial y} + \frac{g}{2} \frac{\partial h^2}{\partial y} &= -ghS_{0y} - ghS_{fy}. \end{aligned} \quad (2.18)$$

Here  $g$  is the acceleration due to gravity,  $h$  is the water depth,  $u$  and  $v$  are the flow velocities in the  $x$ - and  $y$ - directions, respectively,  $S_{0x}$  and  $S_{0y}$  are the bed slopes in  $x$ - and  $y$ -directions, respectively. The bed frictions  $S_{fx}$  and  $S_{fy}$  can be estimated by the Manning resistance law

$$S_{fx} = \frac{n^2 \sqrt{u^2 + v^2} u}{h^{4/3}}, \quad S_{fy} = \frac{n^2 \sqrt{u^2 + v^2} v}{h^{4/3}}, \quad (2.19)$$

where  $n$  is the Manning's roughness coefficient.

The equation (2.18) can be written in the vector form as

$$\frac{\partial \vec{w}}{\partial t} + \frac{\partial \vec{f}(\vec{w})}{\partial x} + \frac{\partial \vec{g}(\vec{w})}{\partial y} = \vec{z}(\vec{w}) + \vec{s}(\vec{w}), \quad (2.20)$$

where

$$\vec{w} = \begin{bmatrix} h \\ uh \\ vh \end{bmatrix} \quad (2.21)$$

is the vector of dependent variables consisting of the water depth  $h$ , the discharges per unit width  $uh$  and  $vh$  along the velocity components  $u$  and  $v$  in the  $x$  and  $y$  directions. The flux vectors in the  $x$  and  $y$  directions are

$$\vec{f}(\vec{w}) = \begin{bmatrix} uh \\ u^2 h + g \frac{h^2}{2} \\ uvh \end{bmatrix}, \quad \vec{g}(\vec{w}) = \begin{bmatrix} vh \\ uvh \\ v^2 h + g \frac{h^2}{2} \end{bmatrix}, \quad (2.22)$$

respectively. The right hand side of (2.20) represents the gravity force vector

$$\vec{z}(\vec{w}) = \begin{bmatrix} 0 \\ -ghS_{0x} \\ -ghS_{0y} \end{bmatrix}, \quad (2.23)$$



comprised of the bottom slopes  $S_{0x} = \frac{\partial z}{\partial x}$  and  $S_{0y} = \frac{\partial z}{\partial y}$  of the height of topography,  $z$ , and the vector of source terms

$$\vec{s}(\vec{w}) = \begin{bmatrix} q \\ -ghS_{fx} \\ -ghS_{fy} \end{bmatrix}, \quad (2.24)$$

where  $q$  is the rainfall rate that is vertically added to control volume and  $S_{fx}$  and  $S_{fy}$  are the friction slopes representing the effect of bed roughness. In these equations,  $g$  is the gravity constant and  $t$  denotes the time.

Equation (2.20) can be written in quasilinear form as

$$\frac{\partial \vec{w}}{\partial t} + \frac{\partial \vec{f}(\vec{w})}{\partial \vec{w}} \frac{\partial \vec{w}}{\partial x} + \frac{\partial \vec{g}(\vec{w})}{\partial \vec{w}} \frac{\partial \vec{w}}{\partial y} = \vec{z}(\vec{w}) + \vec{s}(\vec{w}) \quad (2.25)$$

with the Jacobian matrices

$$\frac{\partial \vec{f}(\vec{w})}{\partial \vec{w}} = - \begin{bmatrix} 0 & 1 & 0 \\ -u^2 + gh & 2u & 0 \\ -uv & v & u \end{bmatrix}, \quad (2.26)$$

$$\frac{\partial \vec{g}(\vec{w})}{\partial \vec{w}} = - \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ -v^2 + gh & 0 & 2v \end{bmatrix}. \quad (2.27)$$

The Jacobian matrix  $\frac{\partial \vec{f}(\vec{w})}{\partial \vec{w}}$  has eigenvalues,

$$\lambda^{x1} = u - \sqrt{gh}, \quad \lambda^{x2} = u, \quad \lambda^{x3} = u + \sqrt{gh} \quad (2.28)$$

and eigenvectors,

$$\vec{r}^{x1} = \begin{bmatrix} 1 \\ u - \sqrt{gh} \\ v \end{bmatrix}, \quad \vec{r}^{x2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \vec{r}^{x3} = \begin{bmatrix} 1 \\ u + \sqrt{gh} \\ v \end{bmatrix}. \quad (2.29)$$

Similarly, the Jacobian matrix  $\frac{\partial \vec{g}(\vec{w})}{\partial \vec{w}}$  has eigenvalues,

$$\lambda^{y1} = v - \sqrt{gh}, \quad \lambda^{y2} = v, \quad \lambda^{y3} = v + \sqrt{gh} \quad (2.30)$$

and eigenvectors,

$$\vec{r}^{y1} = \begin{bmatrix} 1 \\ u \\ v - \sqrt{gh} \end{bmatrix}, \vec{r}^{y2} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \vec{r}^{y3} = \begin{bmatrix} 1 \\ u \\ v + \sqrt{gh} \end{bmatrix}. \quad (2.31)$$

# CHAPTER III

## DYNAMICALLY ADAPTIVE GRID ALGORITHM FOR FLOOD SIMULATION

In this chapter we present numerical scheme and algorithm to simulate and visualize floods models on natural topography based on finite volume methods for shallow water equations by modifying the adaptive grid technique, developed in [30, 34, 2] for rectangular quad tree grids, and combining with the dynamic DDM. Here, the idea of quad tree grids is extended for general tree grids to handle natural topography data in general.

The rest of the chapter is organized as follows. Section 3.1 describes the designing of the data structure for general rectangular tree grids. The numerical algorithm of FVM based on the adaptive grids and dynamic DDM are explained in Section 3.2, and the numerical simulations and experimental results, to check the accuracy of the algorithm, are shown in Section 3.3.

### 3.1 Tree Grids

The data structure for tree grids is designed for adaptivity based on rectangular quad tree grids, appeared in [30, 34, 2], where each parent cell has four children and

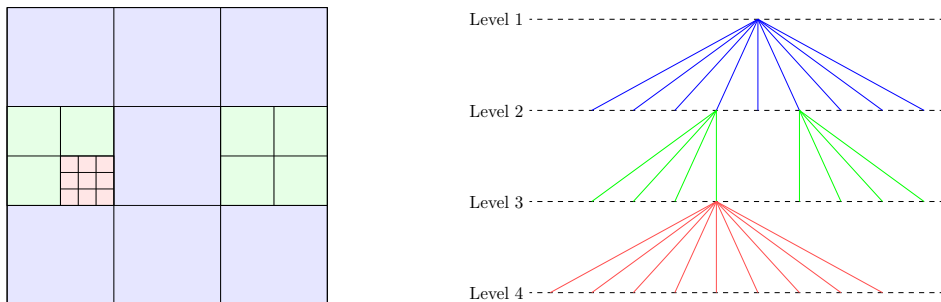


Figure 3.1: Regularized tree grids. Figure 3.2: Hierarchical data structure.

the levels of adjacent cells cannot differ more than one. In order to handle general natural topography data, we design in the algorithm that parent cells can have any number of children cells with square dimension and the levels of adjacent cells can differ more than one. In addition, all parent cells in the same tree level must have the same number of children. For example, Figure 3.1 shows a regularized grid with 4 subdivision levels having 1 big cell in level 1, and number of children for parent cells in levels 1, 2 and 3 are  $3 \times 3$ ,  $2 \times 2$  and  $3 \times 3$ , respectively, while level 4 cells have no child. The corresponding hierarchical tree structure is illustrated in Figure 3.2.

### 3.2 Numerical Algorithm

We present in this section the developed numerical algorithm for the system (2.20) using conservative finite volume methods based on well-balanced schemes with hydrostatic reconstruction, proposed by Audusse [3] and Delestre [9], to simulate and visualize the floods models over non flat bottom on adaptive grids. We utilize the scheme so that the system preserves the non-negativity of the water depth (negative water depths are physically incorrect and undesirable in numeric) and conserves the total water depth, including the ability to compute dry states. Furthermore, the scheme also satisfies the natural properties for still water, where  $(u, v) = 0$  and  $h + z = \text{constant}$  is the solution of the system, reflecting the fact that a lake at rest remains so, and satisfies the discrete entropy inequality to guarantee that the total energy behaves physically correct (see [3]).

However, the hydrostatic reconstruction can fail when the topography slope becomes too big compared to the water height as demonstrated in [11], but most of the time in those cases, the shallow water equations are no longer valid. Some improvements have been performed in order to overcome those limits in [24, 32]. Incorporation of these improvements are beyond the adopted scheme of this research. The details of the numerical scheme and algorithm of this research are described as follows.

### 3.2.1 Finite volume formulation

The finite volume formulation was performed on physical domain that is made of regularized cells. Each cell is considered as a main control volume whose boundary is formed by the four direct walls surrounding it. An example of a single main control volume is shown in Figure 3.3.

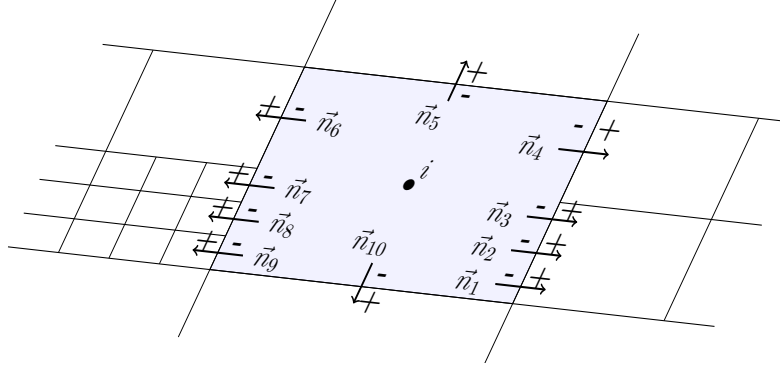


Figure 3.3: The main control volume and descriptions of numerical fluxes.

By integrating equation (2.20) over an arbitrary cell domain  $\Omega_i$  and using the Green's theorem, we have

$$\int_{\Omega_i} \frac{\partial \vec{w}}{\partial t} d\Omega + \oint_{\tau} \vec{F} \cdot \vec{n} d\tau = \int_{\Omega_i} \vec{z}(\vec{w}) d\Omega + \int_{\Omega_i} \vec{s}(\vec{w}) d\Omega, \quad (3.1)$$

where  $\tau$  is the boundary of  $\Omega_i$ ,  $\vec{n} = [\cos \theta \ \sin \theta]^T$  is the unit outward normal vector to the boundary, and  $\theta$  is the angle between  $\mathbf{n}$  and the positive x-axis.  $\vec{F} = [\vec{f}(\vec{w}), \vec{g}(\vec{w})]^T$  is the vector that consists of flux vectors in equations (2.20) at each interface of the cell boundary.

Dividing equation (3.1) with the cell area  $\Delta_i = \Delta x_i \times \Delta y_i$ , and denoting  $\vec{W}_i$ ,  $\vec{Z}_i$  and  $\vec{S}_i$  as the averages of  $\vec{w}_i$ ,  $\vec{z}(\vec{w}_i)$  and  $\vec{s}(\vec{w}_i)$  over the cell, respectively, the finite volume formulation becomes

$$\frac{\partial \vec{W}_i}{\partial t} = -\frac{1}{\Delta_i} \oint_{\tau} \vec{F} \cdot \vec{n} d\tau + \vec{Z}_i + \vec{S}_i, \quad (3.2)$$

where subscripts  $i$  denote spatial index of the cell.

Since our computing model is performed on dynamically adaptive grids considered as non-uniform grids (by the non-uniform grids, each cell can possibly have

many adjacent cells and with different size for each wall), the discretization of the line integral in equation (3.2) is computed as the sum of numerical fluxes of sub-interfaces generated by a reconstruction of the cell boundary, as illustrated in Figure 3.3. The reconstruction detail is described in §3.2.3. Equation (3.2) can be written as

$$\frac{\partial \vec{W}_i}{\partial t} = -\frac{1}{\Delta_i} \sum_k \vec{F}_k \cdot \vec{n}_k \Delta\tau_k + \vec{Z}_i + \vec{S}_i, \quad (3.3)$$

where  $k$  is index of sub-interface  $k$ , and  $\Delta\tau_k$  is size of the sub-interface  $k$ .

However, by using the Audusse's scheme [3], the gravity force  $\vec{Z}_i$  can be distributed to the numerical fluxes for each sub-interfaces, i.e.

$$-\frac{1}{\Delta_i} \oint_{\tau} \vec{F} \cdot \vec{n} d\tau + \vec{Z}_i \approx -\frac{1}{\Delta_i} \sum_k \hat{\vec{F}}_k \cdot \vec{n}_k \Delta\tau_k. \quad (3.4)$$

Equation (3.3) can now be written as

$$\frac{\partial \vec{W}_i}{\partial t} = -\frac{1}{\Delta_i} \sum_k \hat{\vec{F}}_k \cdot \vec{n}_k \Delta\tau_k + \vec{S}_i, \quad (3.5)$$

where  $\hat{\vec{F}}$  is the numerical flux, depending upon the developed scheme.

### 3.2.2 Numerical flux

In the developed algorithm, a computational scheme based on the work of Audusse *et al* [3] has been employed for numerical fluxes calculation for each sub-interface. The numerical fluxes with the gravity force are computed by the HLL Riemann solver, which was developed by Harten, Lax and Van Leer [20], as

$$\hat{\vec{F}}_k \cdot \vec{n}_k = \frac{\alpha_+ \hat{\vec{F}}(\vec{U}_k^-) \cdot \vec{n}_k - \alpha_- \hat{\vec{F}}(\vec{U}_k^+) \cdot \vec{n}_k + \alpha_+ \alpha_- (\vec{U}_k^+ - \vec{U}_k^-)}{\alpha_+ - \alpha_-} + \hat{\vec{Z}}(\vec{U}_k^-) \cdot \vec{n}_k, \quad (3.6)$$

where

$$\alpha_{\pm} = \pm \max \left\{ \pm \vec{v}_k^- \cdot \vec{n} + \sqrt{g \hat{h}_k^-}, \pm \vec{v}_k^+ \cdot \vec{n} + \sqrt{g \hat{h}_k^+}, 0 \right\} \quad (3.7)$$

are the wave speeds, based on the work of Kurganov [27], which are the smallest and the largest eigenvalues of the system (2.25).  $\vec{v} = [u \ v]^T$  is velocity vector.

$\hat{F}(U) = [\vec{f}(\vec{U}) \vec{g}(\vec{U})]^T$  is the flux function defined in (2.22).  $\hat{Z}(\vec{U}) = [\vec{z}_x \vec{z}_y]^T$  is the term to satisfy the balance of momentum flux and momentum gravity forces, as introduced in the second order Audusse's scheme [3] where

$$\vec{z}_x = \left[ 0, \frac{g}{2}(h_k^-)^2 - \frac{g}{2}(\hat{h}_k^-)^2 + gh_i z_k^-, 0 \right]^T, \quad (3.8)$$

$$\vec{z}_y = \left[ 0, 0, \frac{g}{2}(h_k^-)^2 - \frac{g}{2}(\hat{h}_k^-)^2 + gh_i z_k^- \right]^T. \quad (3.9)$$

In these equations,  $\vec{U}_k^\pm = [\hat{h}_k^\pm (u\hat{h})_k^\pm (v\hat{h})_k^\pm]^T$ , where  $u$  and  $v$  are the reconstructed values of the velocities in  $x$  and  $y$  directions,  $\hat{h}_k^\pm$  are the hydrostatic reconstruction of water depths for preserving the lake-at-rest condition to guarantee that the water depth is nonnegative. The evaluations of  $\hat{h}_k^\pm$  are given by

$$\hat{h}_k^\pm = \max\{0, h_k^\pm + z_k^\pm - \max\{z_k^-, z_k^+\}\}, \quad (3.10)$$

where the superscripts - and + are indices for the reconstructed values within the cell and the adjacent cell at the sub interface  $k$ , respectively, see Figure 3.3.

### 3.2.3 Reconstruction procedure

In this section, the reconstruction procedure is presented. The common goals are to interpolate the profiles of the various quantities within each cell; to recover the information about the behavior of the quantities inside cell that is lost; and to improve the accuracy of numerical result to have a better estimation for solving the Riemann problem at the interface between two adjacent cells. To ensure stable solutions, the reconstruction is performed using a slope limited extrapolation, presented for 2D non-uniform grid, see [16] and [34]. In this research, the formulation can be written as

$$\vec{Q}_k^- = \vec{Q}_i + \sigma_k(x_k - x_i) + \delta_k(y_k - y_i), \quad (3.11)$$

where  $\vec{Q} = [h, u, v, h+z]^T$  is the interpolated variable at each interface  $k$ ,  $(x_i, y_i)$  is position of the cell, while  $(x_k, y_k)$  is position of each interface  $k$ , and  $\sigma_k$  and  $\delta_k$  are the slopes at the position  $(x_k, y_k)$  in  $x$  and  $y$  directions, respectively.

Since the interpolation might generate oscillation near steep gradients in the solution, the minmod slopes limiter are used for  $\sigma_k$  and  $\delta_k$  as

$$\sigma_k = MM\left(\frac{\vec{Q}_k^E - \vec{Q}_i}{\Delta x_i}, \frac{\vec{Q}_i - \vec{Q}_k^W}{\Delta x_i}\right), \quad (3.12)$$

$$\delta_k = MM\left(\frac{\vec{Q}_k^N - \vec{Q}_i}{\Delta y_i}, \frac{\vec{Q}_i - \vec{Q}_k^S}{\Delta y_i}\right), \quad (3.13)$$

where the superscripts E, W, N and S refer to the four neighbor cells (at east, west, north and south positions of the center cell  $i$ ) that correspond to the position of the sub-interface  $k$ .  $MM$  is minmod function which given by

$$MM(a, b) = \frac{\text{sign}(a) + \text{sign}(b)}{2} \min(|a|, |b|) \quad (3.14)$$

with the function  $\text{sign}$  returns the value 1 or  $-1$  follow the sign of parameter.

Notice that the variable  $z$  is instead performed via the quantity  $h + z$ , so the reconstruction of  $z$  is obtained by doing the difference between reconstructions of  $h + z$  and  $h$ . This choice of reconstruction of  $z$  is done following Audusse *et al* [3] to preserve the steady state and non-negativity of water depth.

### 3.2.4 Total variation diminishing

The concept of total variation diminishing (TVD) technique is to have the solution that is both second or higher order accuracy in time and does not produce spurious oscillations. The second order TVD Runge-Kutta method (also called Heun's method) is employed to solve the ODE without source term  $\vec{S}_i$  in the equation (3.5). The scheme has iteration steps as follows:

$$\vec{W}^{(1)} = \vec{W}^t + \Delta t_1 \Phi(\vec{W}^t), \quad (3.15)$$

$$\vec{W}^{(2)} = \vec{W}^{(1)} + \Delta t_2 \Phi(\vec{W}^{(1)}), \quad (3.16)$$

$$\vec{W}^{t+\Delta t} = \frac{\vec{W}^t + \vec{W}^{(2)}}{2}, \quad (3.17)$$

where  $\Delta t = \frac{\Delta t_1 + \Delta t_2}{2}$  is the time step size, while the superscripts (1) and (2) are iteration steps. In this scheme,  $\Phi$  denotes the right hand side function in equation (3.5) without the source term  $\vec{S}_i$ .



### 3.2.5 Source term computation

The source term vector  $\vec{S}_i$  in equation (3.5) consists of the rainfall rate and the friction forces. For the rainfall rate, since this one is the water depth that is vertically added to control volume per unit time, we update a new value of  $h_i^{t+\Delta t}$  in equation (3.17) by adding  $q_i^t \Delta t$ , namely,

$$h_i^{t+\Delta t} \leftarrow h_i^{t+\Delta t} + q_i^t \Delta t, \quad (3.18)$$

where  $q_i^t$  is rainfall rate average in each cell  $i$  in the range of time step from  $t$  to  $t + \Delta t$ .

For the friction force, we used the semi-implicit method (proposed in [9]) to ensure a stability criteria, and in order to preserve steady states at rest. It updates a new value of  $uh_i^{t+\Delta t}$  in equation (3.17) as

$$uh_i^{t+\Delta t} \leftarrow \frac{uh_i^{t+\Delta t}}{1 + \Delta t \frac{gn^2 \sqrt{(u_i^t)^2 + (v_i^t)^2}}{(h_i^{t+\Delta t})^{4/3}}}. \quad (3.19)$$

Similarly,  $vh_i^{t+\Delta t}$  can be computed by a formula in the form of the equation (3.19) using  $vh_i^{t+\Delta t}$  instead of  $uh_i^{t+\Delta t}$ .

### 3.2.6 Stability condition

In order to have stable scheme, the maximum of time step size was limited by the Courant Friedrich Lewy (CFL) condition ( presented by Hargen [21]) as

$$\Delta t_m \leq 0.5 \frac{\Delta A_{\min}}{\lambda_{\max}}, \quad m = 1, 2, \quad (3.20)$$

where  $\Delta t_m$  are the time step sizes at iteration step  $m = 1, 2$  for equation (3.15) and (3.16) of the second order TVD Runge-Kutta process, and  $\Delta t$  in the equation (3.18) and (3.19) is computed from average value of  $\Delta t_1$  and  $\Delta t_2$ . In the above equation,  $\Delta A_{\min} = \min_{\forall i} \{\Delta x_i, \Delta y_i\}$  is the minimum dimension for all cells, while  $\lambda_{\max}$  is the maximum absolute value of all the wave speeds in the computational domain, which is the maximum of  $\max_{k,i} \{\alpha_+, -\alpha_-\}$  for all the sub interfaces and for all cells.

### 3.2.7 Boundary condition

In this research, we proposed two types of boundary conditions, the open (transmissive) and the closed (solid wall). The open boundary conditions can be defined as

$$h_m = h_n, u_m = u_n, v_m = v_n, z_m = z_n. \quad (3.21)$$

The closed boundary conditions are imposed as

$$h_m = 0, u_m = 0, v_m = 0, z_m = z_{large}. \quad (3.22)$$

In these equations, the subscript  $m$  refers to the spatial index of the boundary cells, and the subscript  $n$  refers to the spatial index of the neighbors of the boundary cells, while  $z_{large}$  is some big constant value for the topography height, for example, we assign  $z_{large} = 9999$ .

### 3.2.8 Grid adaptivity method

The accuracy of the numerical simulation of water overland flow can be increased by grid refinement to have smaller cell size on high resolution of the digital terrain data grids. However, computation is very expensive for very fine grids, in terms of time and resources, if it is performed for all cells in the computational domain. To overcome this, a grid adaptivity technique is applied so that cells are adjusted according to some conditions.

The refinement criterion is based on two conditions, the gradient of the free surface,  $H = h + z$ , and the water depth,  $h$ . A grid cell is marked for refinement if the following two conditions hold:

*Condition 1:* The cell has at least one neighbor cell that

$$|\nabla H| \Delta_l > \epsilon_1, \quad (3.23)$$

where  $\nabla H$  is the free surface gradient,  $\Delta_l$  is the distance between the checked cell and the neighbor cell, and  $\epsilon_1$  is an adjustable parameter. This condition was first proposed by Popinet[34] for tsunami simulation which expected that the error was covered by the discretization of sharp fronts.

*Condition 2:* The cell itself or at least one of neighbor cells has water depth

$$h > \xi, \quad (3.24)$$

where  $\xi$  is an adjustable parameter. The condition 2 is important for simulation of water flow over natural topography, because it is possible that the cell and its neighbors are dry or have very small depths that satisfies the condition 1. In this case, the cell should not be marked for refinement because it has no water flow.

### *Refinement Procedure*

A marked cell is refined by creating children cells in the next level. The newly assigned dependent value for each children cells is computed based on the slope limited extrapolation. For non-uniform grids, each children dependent value is computed using the average of the minmod slopes of the values of its neighbors, namely,

$$D_j^{l+1} = D_i^l + \bar{\sigma}_j(x_j - x_i) + \bar{\delta}_j(y_j - y_i), \quad (3.25)$$

where  $i$  and  $j$  refer to the spatial indices of the parent cell and the children cells, respectively.  $D_j^{l+1}$  is the dependent values (which are  $h$ ,  $uh$  and  $vh$ ) of the child cell at level  $l+1$ , while  $D_i^l$  is the values of its parent cell. The position of the child cell is  $(x_j, y_j)$ , while the position of its parent cell is  $(x_i, y_i)$ . Here

$$\bar{\sigma}_j = \frac{1}{N_j} \sum_{n \in \Omega_j} \sigma_n, \quad \bar{\delta}_j = \frac{1}{M_j} \sum_{m \in \Omega_j} \delta_m \quad (3.26)$$

are the averages minmod slope limiters from its neighbor cells in  $x$  and  $y$  directions, respectively, where  $\Omega_j$  is the domain of the children cells,  $M_j$  and  $N_j$  are the number of columns and rows formed by its neighbor cells, respectively, and  $\sigma_n$  and  $\delta_m$  are the minmod slope limiters defined in §3.2.3.

### *Coarsening procedure*

The coarsening conditions are the same as the refining conditions where the parameter  $\epsilon_1$  in the first condition (3.23) is replaced by another parameter  $\epsilon_2$ . All

children cells with the same parent cell are combined or merged into their parent cell, if the computed value for the parent

$$D_i^l = \frac{1}{n_i^l \times n_i^l} \sum_{j \in \Omega_i} D_j^{l+1} \quad (3.27)$$

does not satisfy one of the coarsening conditions. In this case, the newly merged cell will have the dependent values defined by (3.27). In this equation (3.27),  $n_i^l \times n_i^l$  is the number of children.

In the computation, the heights of topography on computational grid cells,  $z$ , is obtained via provided digital terrain data on each level, otherwise, in the case that the digital terrain data is not given, the values are obtained similar to the other dependent values as defined in (3.25) or (3.27).

### 3.2.9 Topography interpolation

In general modeling and simulation based on natural topography, the heights of topography on computational grid cells,  $z$ , is usually obtained using the digital terrain data grids. Since in computation using adaptive grids, it is possible that the grid cell resolution is higher than that of the data; therefore, the topography interpolation is required. In this work, we used the bilinear interpolation technique to obtain the heights of topography. The approximated value of the cell at  $i$  is obtained by

$$\begin{aligned} z_i = & z_{x',y'}(x' + 1 - m)(y' + 1 - n) + z_{x'+1,y'}(m - x')(y' + 1 - n) \\ & + z_{x',y'+1}(x' + 1 - m)(n - y') + z_{x'+1,y'+1}(m - x')(n - y'), \end{aligned} \quad (3.28)$$

where  $(x', y')$  is the position of the topographic data grid cell used for approximation with  $x' = \lfloor m \rfloor$  and  $y' = \lfloor n \rfloor$ ,

$$m = x_i \times n^{x'} / n^x, \quad n = y_i \times n^{y'} / n^y, \quad (3.29)$$

are the mapped indices of the computational grid cell position  $(x_i, y_i)$  to the topographic data grid,  $n^{x'}$  and  $n^{y'}$  are numbers of columns and rows of the topographic data grid, and  $n^x$  and  $n^y$  are numbers of columns and rows of the finest computational grids.

### 3.2.10 Dynamic domain defining method

In order to improve the computational efficiency, the dynamic DDM is applied in this modeling and simulation. The method automatically expands or shrinks the calculation area during the simulation to exclude unnecessary cells for the computation. To apply this method to the water overland flow simulation, the cells are checked to identify whether the cell is inside the computational domain or not. The checking is based on three conditions of indicators:

1. the cell is wet, *i.e.*  $h_i^t > 0$ ;
2. the cell has one or more of neighbor cells that are wet, *i.e.*  $h_k^t > 0$  for some neighbor cell,  $h_k^t$ ;
3. the cell has rainfall, *i.e.*  $q_i^t > 0$ .

If one of these conditions is satisfied, the calculation area is expanded where the cell and its neighbor cells are included to the domain whenever these cells are not in the domain. The cell is checked (that is in the domain), before including the cell to the domain to protect against errors, and to avoid computation from repeated work. On the other hand, the calculation area is shrunk, where the cell is excluded from the domain. Only grid cells within the computational domain are adapted and computed in order to minimize the total number of the computational cells.

We used this checking because when the cell and its neighbor cells are all dry with no rainfall, the fluxes of flowing is zero, therefore the cell dependent value is zero, so the numerical computation is unnecessary for the cell.

In the algorithm, the dynamic DDM is employed to obtain a new computational domain for each iteration after the solution of the (3.15) and (3.16) is obtained via TVD Runge-Kutta method.

### 3.2.11 Algorithm overview

We present here an overview of the algorithm for the computational scheme presented above. The developed algorithm consists of several steps describing the

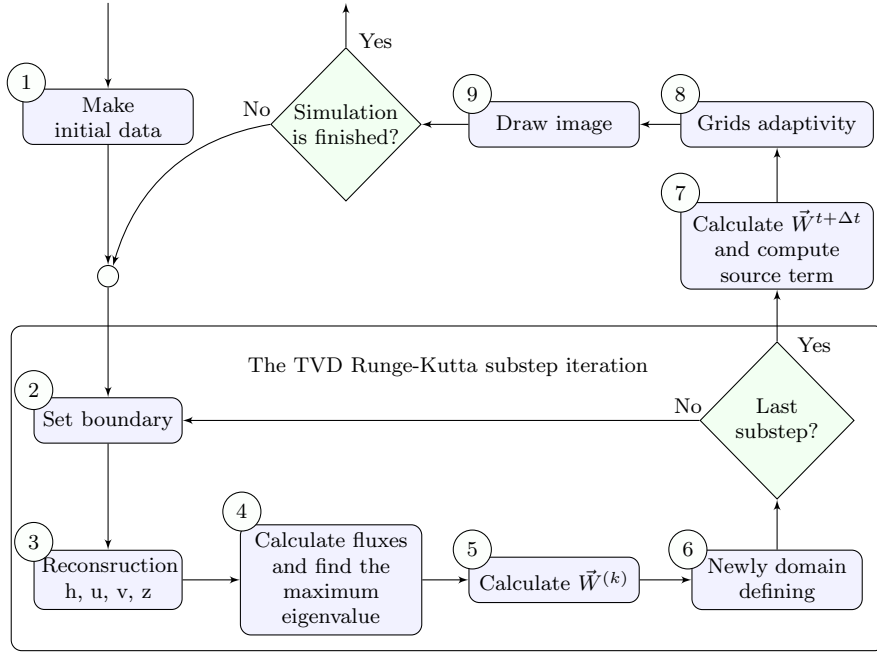


Figure 3.4: The algorithm overview.

calculation procedures, as illustrated in Figure 3.4. The detail for each step is described as follows.

- Step ① : Set the initial values, the cell sizes, the number of children in each level, and the maximum of tree grid levels.
- Step ② : Set the computational grid boundaries, see §3.2.7.
- Step ③ : Conduct the reconstruction procedure, see §3.2.3.
- Step ④ : Calculate the numerical fluxes and the maximum wave speed.
- Step ⑤ : Calculate the solutions of  $\vec{W}^{(1)}$ , see (3.15).
- Step ⑥ : Define a new domain based on the dynamic DDM, see §3.2.11.
- Repeat Step ② to ⑥, for the solutions of  $\vec{W}^{(2)}$ , see (3.16), and go to Step ⑦.
- Step ⑦ : Calculate the solutions of  $\vec{W}^{t+\Delta t}$ , see (3.17), and compute the source terms, see §3.2.5.

- Step ⑧ : Conduct the grid adaptivity technique, see §3.2.8.

#### I Refinement procedure

- i For each cells whose level is not maximum level
  - Mark cells that satisfy the refinement conditions.
- ii For each marked cells
  - Create new children cells, and assign the dependent values as equation (3.25).

#### II Coarsening procedure

- i For each parent cells whose children have no child
  - Compute the dependent values defined in (3.26);
  - Check the coarsening conditions;
  - If one of the conditions is false, then merge all children cells into a parent cell, and assign the dependent values.

- Step ⑨ : Visualize the solutions in 2D and 3D by OpenGL.
- Repeat Step ② to Step ⑨ until the simulation is finished.

Note that, in Step ③, cells average values  $u$  and  $v$  are used for the reconstruction procedure, which are calculated by dividing  $uh$  and  $vh$  with  $h$ . However, since the values of  $u$  and  $v$  are not defined in case that  $h = 0$ , the function presented by Kurganov [29] is used for finding the cells average values of  $u$  and  $v$  as

$$\text{div}(a, h) = \frac{\sqrt{2}ah}{\sqrt{h^4 + \max(h^4, \eta)}}, \quad (3.30)$$

where  $\eta$  is a small positive constant value, and  $a$  is substituted by  $uh$  and  $vh$ , respectively.

### 3.3 Experimental Results

We illustrate here some numerical simulations based on adaptive grids and dynamic DDM to illustrate the effectiveness of the developed model. The experiments

consist of three tests as follows: the dam break simulation to show the adapted grid and water profile of the model; the simulation of a 2D parabolic bowl to compare with the analytical solution; and the rainfall run-off simulation on sloped plane compared with real experimental data. The numerical simulations are for accuracy tests of the adaptive tree grid technique and the numerical algorithm. The developed algorithm is programmed using Delphi Version 7 on 2 GHz Intel Core i7 machine with RAM 6 GB. The details of the numerical simulations and experimental results are shown in the following subsections.

### 3.3.1 2D Dam break simulation

This experiment simulates the aspect of the water flow for a dam break behavior on  $200 \text{ m} \times 200 \text{ m}$  rectangular area with a flat horizontal frictionless bottom. The square domain is divided into two halves by infinitesimally thin dam. The breaking part of the dam is 75 m long from 100 m to 175 m in the y-direction, see Figure 3.5. The experiment is performed with the initial water depths at 10 m upstream, while the downstream is at 5 m, and with frictionless solid walls. The adaptivity of rectangular tree grid is employed with 3 levels: the initial level 1 has  $50 \times 50$  grid cells; the dimensions of children for levels 1 and 2 are  $2 \times 2$  and  $3 \times 3$ , respectively; the level 3 cells have no children. In addition, the parameters used in the grid adaptivity, in (3.23)-(3.24), are  $\epsilon_1 = \epsilon_2 = 0.05$ ,  $\xi = 0$ .

The initial tree grid is shown in Figure 3.5 and the simulation of the behavior of water flow together with the grid cells at time  $t = 5 \text{ s}$ , based on the dynamically DDM with adaptive tree grids, are shown in Figure 3.6. This experiment showed that the high-resolution regions, where the gradients of the free surface are steep, are obtained automatically via the adaptive technique. The results agree closely with other numerical simulation results in literature, see [30, 35, 36, 41, 43].



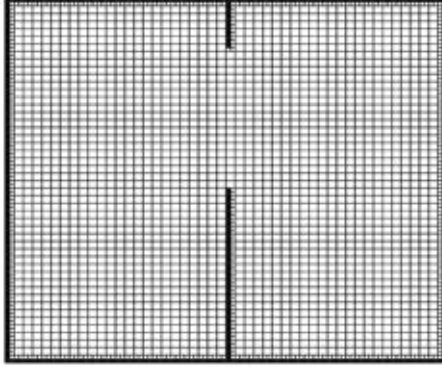


Figure 3.5: The initial grid cells of the rectangular dam break.

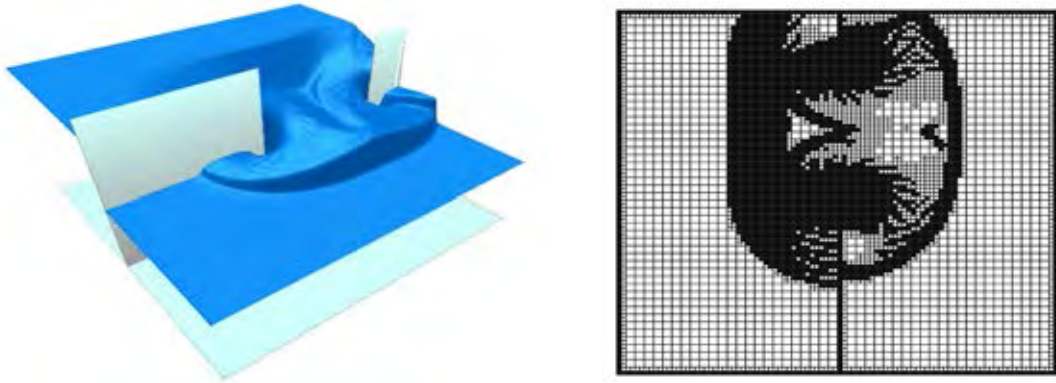


Figure 3.6: The 3D representation (left) and the adaptive grid (right) of the dam break simulation at  $t = 5s$ .

### 3.3.2 2D Parabolic bowl simulation

To validate our developed algorithm, the numerical experiment of a water flow on 2D parabolic bed slope is performed and compared with analytic solution, presented by Wang [44]. Note that the 2D parabolic bowl test cases is a variant of Thackers solution [42] with Mannings friction term.

For the domain  $L \text{ m} \times L \text{ m}$ , the 2D parabolic bed topography is defined by

$$z(x, y) = h_0 \left( (x - L/2)^2 + (y - L/2)^2 \right) / a^2, \quad (3.31)$$

where  $h_0$  and  $a$  are positive constants. It is known that the analytic solution depends on the relationship between the bed friction parameter  $\tau$  and a peak

amplitude parameter  $p = \sqrt{8gh_0}/a$ . For  $\tau < p$ , the analytic solution for the water depth is

$$h(x, y, t) = h_0 - \frac{1}{2g}B^2e^{-\tau t} - \frac{1}{g}Be^{-\tau t/2}\left(\frac{\tau}{2}\sin st + s\cos st\right)\left(x - \frac{L}{2}\right), \\ - \frac{1}{g}Be^{-\tau t/2}\left(\frac{\tau}{2}\cos st - s\sin st\right)\left(y - \frac{L}{2}\right), \quad (3.32)$$

and the velocities are

$$u(t) = Be^{-\tau t/2}\sin st, \quad (3.33)$$

$$v(t) = -Be^{-\tau t/2}\cos st, \quad (3.34)$$

where  $s = \sqrt{p^2 - \tau^2}/2$  and  $B$  is a constant.

In this experiment, we assume that  $L = 10000$  m,  $h_0 = 10$  m,  $a = 3000$  m, and  $B = 5$  m/s. The bed friction parameter,  $\tau = 0.002$  s<sup>-1</sup>, is related to the Manning's coefficient  $n^2 = \frac{\tau h^{4/3}}{g\sqrt{u^2+v^2}}$ . The initial grid level 1 has 50 x 50 cells, and the maximum tree grid level is 3 with 2 x 2 and 3 x 3 children for levels 1 and 2, respectively, while level 3 cells have no children. The initial water depth, velocities and bed topography are defined using the analytic solution at  $t = 0$ . The duration of the simulation is 6000 s and performed with parameters of the grid adaptivity  $\epsilon_1 = \epsilon_2 = 0.3$  and  $\xi = 0.3$ , and with opened boundaries.

The water surface profiles along the center-line in the x-direction obtained by the simulations are compared with the analytic solutions at different time steps. As shown in Figure 3.7, the profile of the water surfaces and the wet-dry fronts agree closely with the analytical solution. The evolutions of the velocities  $u$  and  $v$  compared with analytic results are shown in Figure 3.8. The effect of friction on the flow is clear, because the amplitudes of the velocities oscillate and decrease throughout the simulation. This experiment showed that the developed numerical scheme is capable of simulating a water flow on non-uniform topography with moving wet-dry fronts and bed roughness.

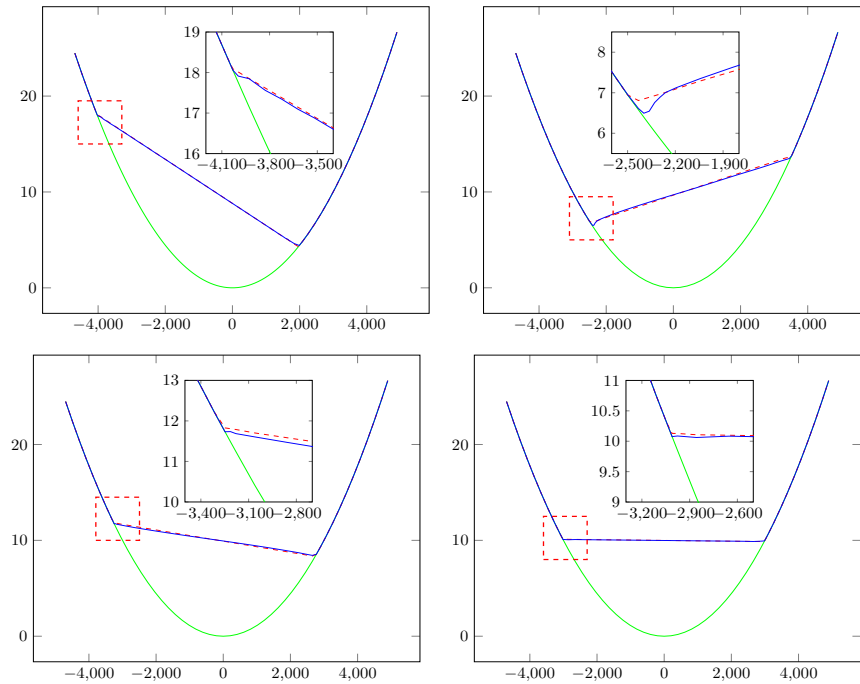


Figure 3.7: Water surface profiles along the center-line in the x-direction of the simulations (solid blue line) compared with the analytic solution (dotted red line) for the 2D parabolic bed topography (solid green line) at  $t = 30.3532$  s,  $671.7932$  s,  $1358.1300$  s, and  $2503.5160$  s (from left to right and top to bottom). The small pictures show some zooms at the wet/dry transition for the simulations.

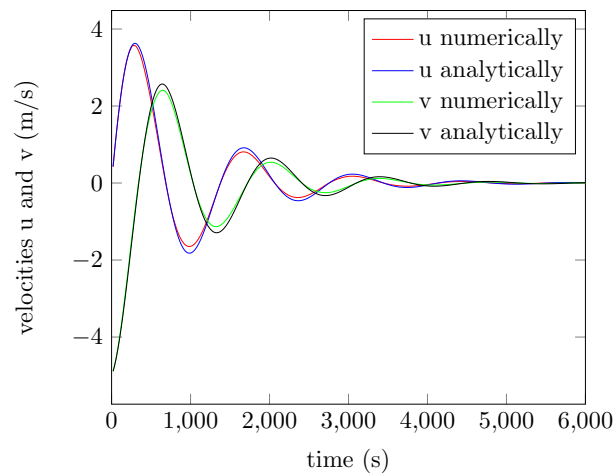


Figure 3.8: The evolution of the velocities of the model compared with analytic solutions at  $x = 6000$  m.

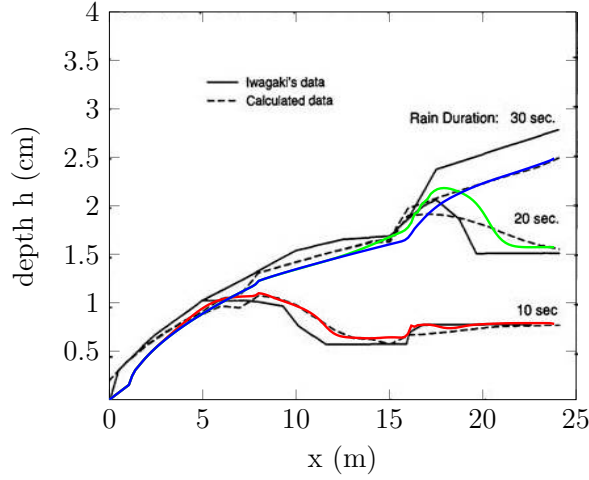


Figure 3.9: The water profiles along the center-line in the  $x$ -direction of the simulations at 10 s (red), 20 s (green) and 30 s (blue) compared with Iwagaki's experimental and calculated (theoretical) data.

### 3.3.3 Rainfall simulation on sloped plane

In this test, the simulation of rainfall run-off on sloped plane was performed and compared with experimental data of Iwagaki [25]. In the physical experiment by Iwagaki, the flow was performed on 24 m long and 0.196 m wide, cascaded by three (8-m long) aluminium planes with slopes of 0.02, 0.015, and 0.01 in the downstream direction, which received constants rainfall intensities of 389, 230 and 288  $\text{cm}\cdot\text{h}^{-1}$ , respectively. The experiment was performed in three cases of different durations of rainfall: 10 s (case 1), 20 s (case 2), and 30 s (case 3).

In this numerical simulation, the initial grid level 1 has  $24 \times 7$  cells, and the maximum tree grid level is 3 with  $3 \times 3$  and  $2 \times 2$  children for levels 1 and 2, respectively, while level 3 cells have no children. The initial water depth and velocities are defined as zero. The boundary conditions are defined as closed for both lateral walls and opened at the upstream and downstream. The simulation is performed with the Manning's coefficient 0.009 for the duration of 80 s, and with parameters of the grid adaptivity  $\epsilon_1 = \epsilon_2 = 0.002$  and  $\xi = 0.003$ .

The numerical simulations of water profiles compared with Iwagaki's experimental and calculated data are shown in Figure 3.9. The comparisons of the

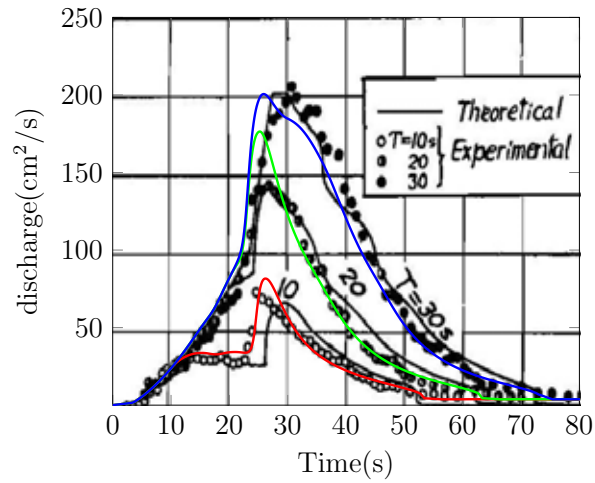


Figure 3.10: The evolutions of the discharges at the downstream with rainfall durations: 10 s (red), 20 s (green) and 30 s (blue), compared with Iwagaki's experimental and calculated (theoretical) data.

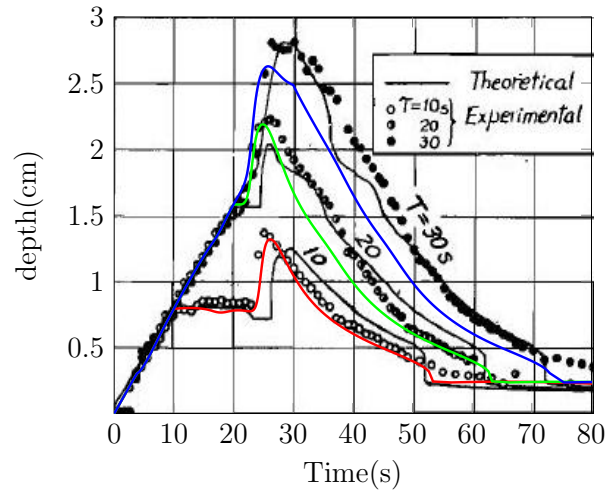


Figure 3.11: The evolutions of the water depth at the downstream with rainfall durations: 10 s (red), 20 s (green) and 30 s (blue), compared with Iwagaki's experimental and calculated (theoretical) data.

evolutions of the discharge and water depth downstream are shown in Figures 3.10 and 3.11. The results agree closely with the rainfall run-off experiment data.

### **3.4 Conclusion**

In this chapter, we describe the dynamically adaptive tree grid algorithm for simulation and visualization of water overland flow based on adaptive finite volume methods for shallow water equations, and in combination with the dynamic DDM. The numerical simulations show that the developed algorithm is suitable for simulating and visualizing the flow of water on natural topography, which agree closely with other results obtained in literature, and for both experimental and theoretical results. In the next chapter, we will present the numerical simulations and results of the algorithm to check the efficiency of the developed algorithm.

# CHAPTER IV

## FLOOD SIMULATIONS USING THE DYNAMICALLY ADAPTIVE TREE GRID TECHNIQUE

We illustrate here the numerical simulations based on adaptive grids and dynamic DDM to show the effectiveness of the developed model for the simulation of floods on natural topography. The experiments consist of three tests, which are the rain-water overland flow simulation, the simulation of a dam-break on natural topography, and the breaching flood simulation. The developed algorithm is programmed using Delphi Version 7 on 2 GHz Intel Core i7 machine with RAM 6 GB.

### 4.1 Rain Water Overland Flow

In this experiment, the developed dynamic DDM with adaptive tree grid numerical algorithm is employed to simulate the rain-water flow on natural topography at Krungching, Nakornsritamarat, Thailand. The accuracy and efficiency of the dynamically adaptive technique are tested and compared with simulations on the uniform grids. The simulations are performed on the domain of digital terrain data  $36000 \text{ m} \times 36000 \text{ m}$ , generated from Shuttle Radar Topography Mission (SRTM) data source. The maximum tree grid level is 4 with resolutions  $900 \text{ m} \times 900 \text{ m}$ ,  $180 \text{ m} \times 180 \text{ m}$ ,  $90 \text{ m} \times 90 \text{ m}$ , and  $30 \text{ m} \times 30 \text{ m}$ , for levels 1, 2, 3, and 4, respectively with the initial grid level 1 has  $40 \times 40$  cells. The topography data is provided for levels 1, 2 and 3, except the level 4 obtained automatically via the bilinear interpolation. The number of grid cells for children are designed as  $5 \times 5$ ,  $2 \times 2$ , and  $3 \times 3$  for levels 1, 2, and 3, respectively, while level 4 cells have no children. The numerical experiment is simulated for 3600 s with the Manning's coefficient 0.001 and parameters of the grid adaptivity  $\epsilon_1 = \epsilon_2 = 1.5$  and  $\xi = 0.1$ . The rainfall rate is set at 0.001 m/s within the rectangular region, where  $15750 \leq x \leq 20250$  and

$13500 \leq y \leq 18000$ , for the duration of 1800 s. Moreover, the boundary conditions are defined as opened boundaries.

The 3D simulations of rain-water overland flow are presented at different times in Figure 4.1, and the corresponding automatically adapted grids are shown in Figure 4.2. The results show that the high-resolution areas are automatically obtained by the adaptivity technique in corresponding to the regions where the free surface gradients are steep and the water depths are larger than 0.1 m, based on adaptivity parameters.

To check the accuracy and efficiency of the dynamically adaptive algorithm, we performed the following test. We compared the dynamically adaptive simulation with non-adaptive simulations computed using only the dynamic DDM on two uniform grids, level 1 grid (largest cells with  $40 \times 40$  cells) and level 4 grid (smallest cells with  $1200 \times 1200$  cells). The 3D representations of the rain-water overland flow simulations at 1800 s for this test are shown in Figure 4.3, and the profiles of water depths and discharges at the location (16710 m, 21060 m) are shown in Figure 4.4. The results show that the simulation and the flow profiles obtained by the dynamically adaptive scheme are nearly the same as that obtained from the finest grid (level 4 grid), while the result from the largest cells (level 1 grid) is not acceptable. This shows the accuracy of the adaptive scheme. The efficiency of the adaptive scheme is presented by comparing the number of computational cells and the computational times for the simulations as shown in Figure 4.5. The results show that the adaptive scheme is more efficient because it helps reducing the computational cells, and therefore, reduces the computational time, while keeping the same accuracy as that of the finest grid.

Moreover, to compare the effect of the dynamic DDM, the computational times and number of computational cells are compared for the adaptive schemes with or without the dynamic DDM, as shown in Figure 4.6. The results show that the dynamic DDM can help improving the simulation, but not as much as that by adaptivity.



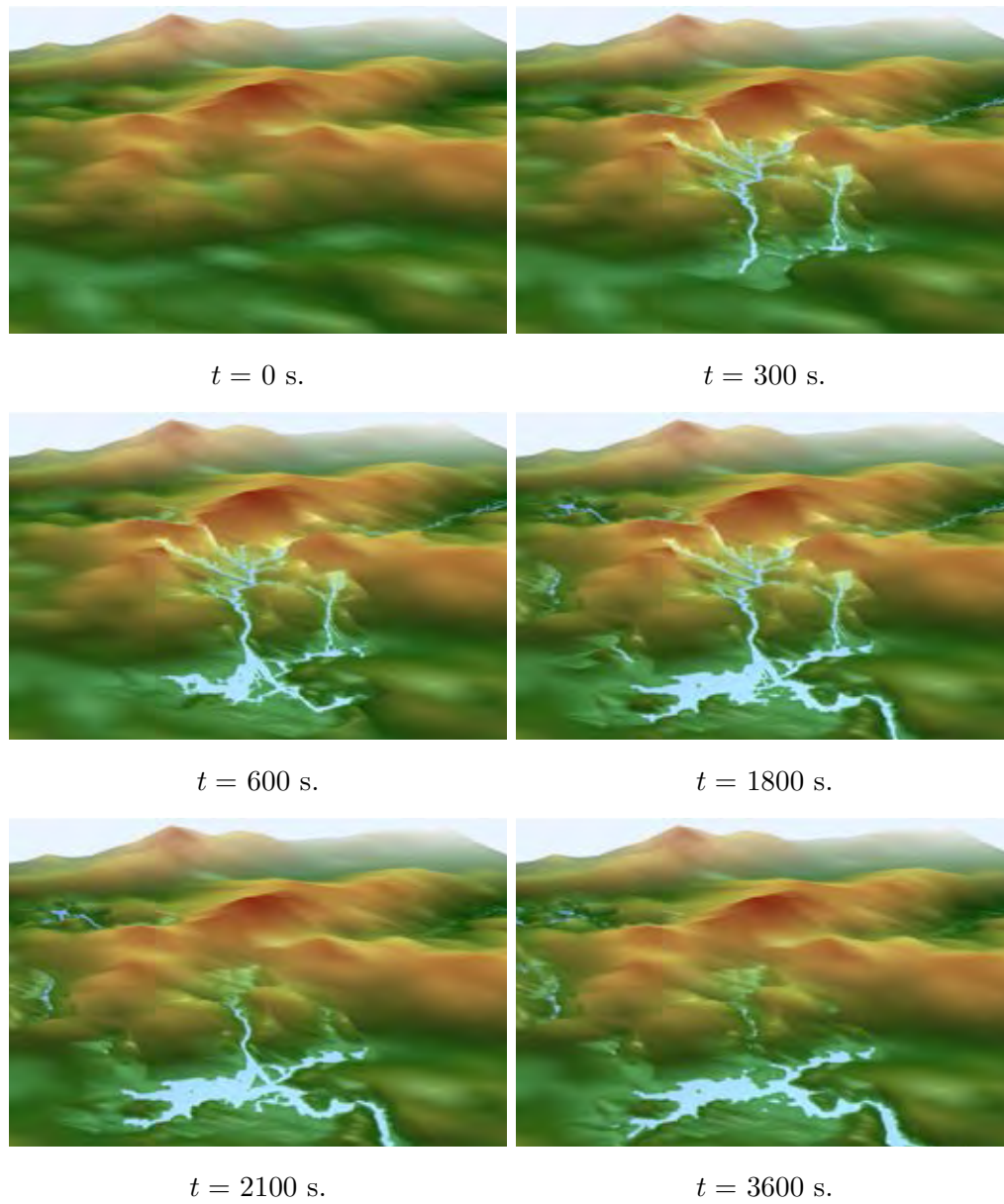


Figure 4.1: The 3D simulations of the behavior of the rain-water overland flow at different times  $t$ .

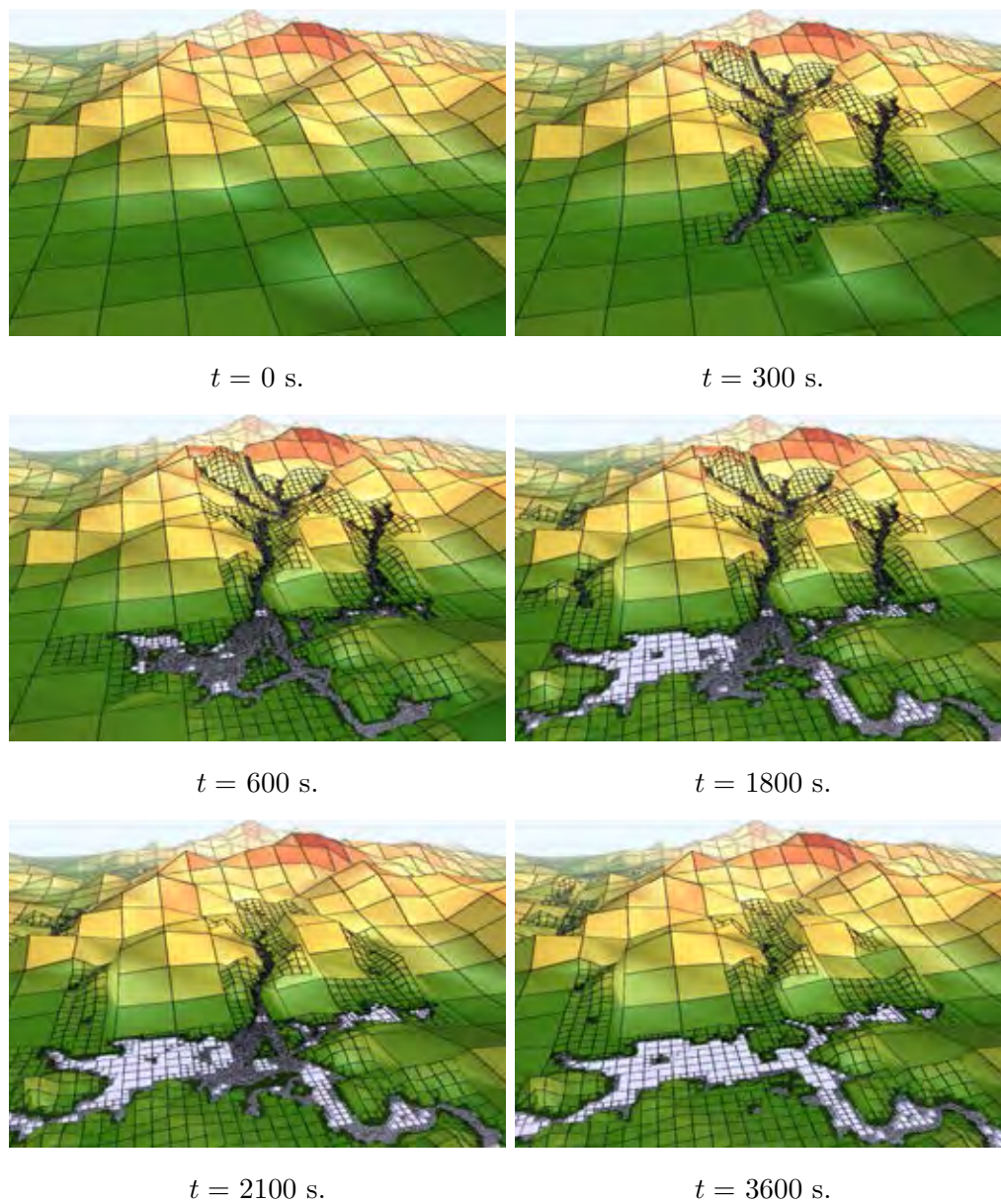


Figure 4.2: The 3D representations of the adapted grids of the rain-water overland flow simulations at different times  $t$ .

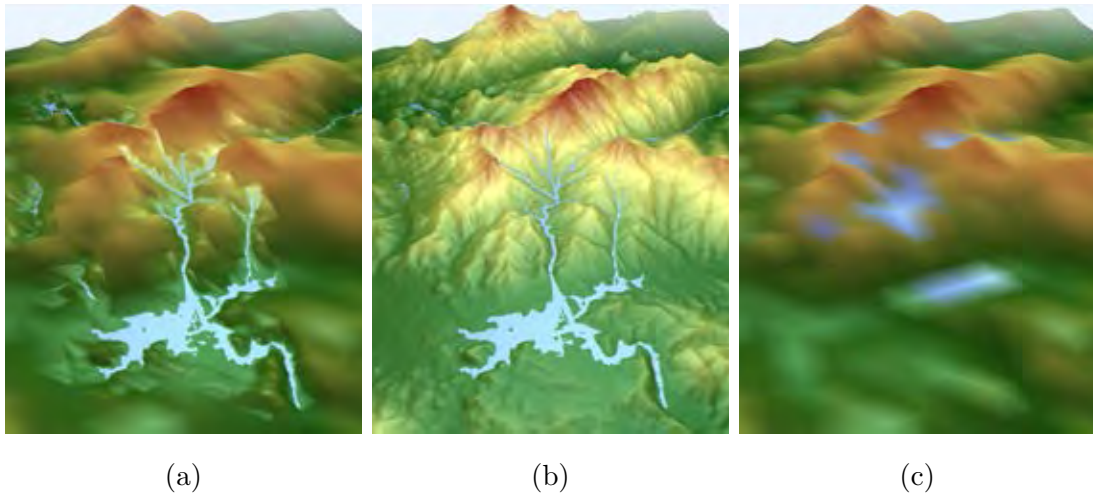


Figure 4.3: The 3D simulations of the rain-water overland flow based on the adaptive grid (a), uniform level 4 grid (small cells) (b), and uniform level 1 grid (large cells) (c) at  $t = 1800$  s.

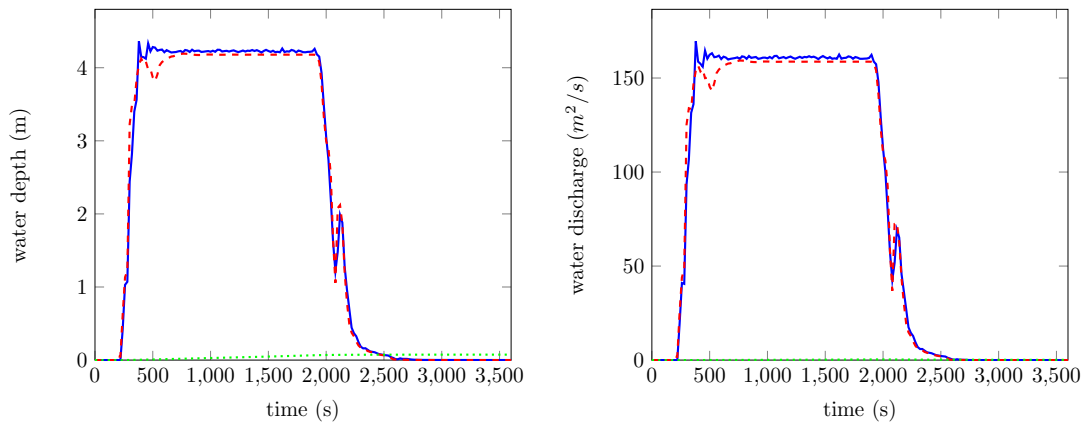


Figure 4.4: The comparisons of water depths (left) and discharges (right) at (16710 m, 21060 m) for the adaptive grid (solid), uniform level 4 grid, small cells (dashed), and uniform level 1 grid, large cells (dotted).

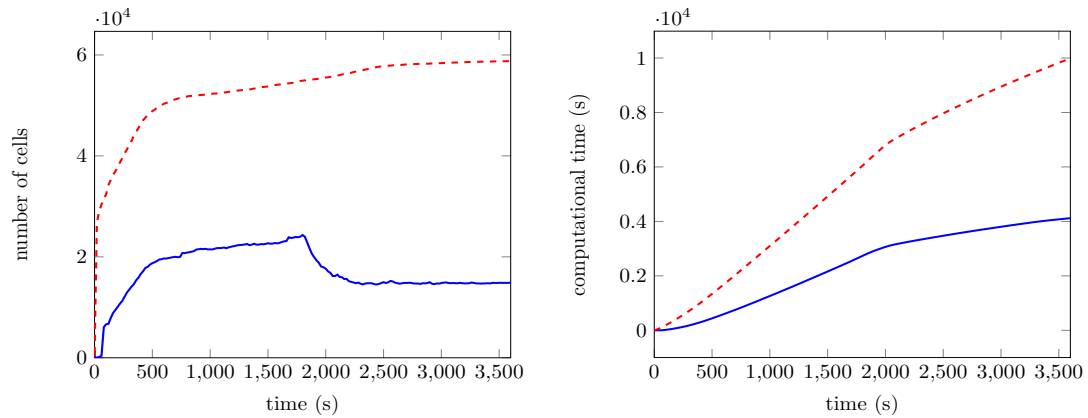


Figure 4.5: The comparisons of number of computational cells (left) and times (right) for adaptive grid (solid), uniform level 4 grid, small cells (dashed).

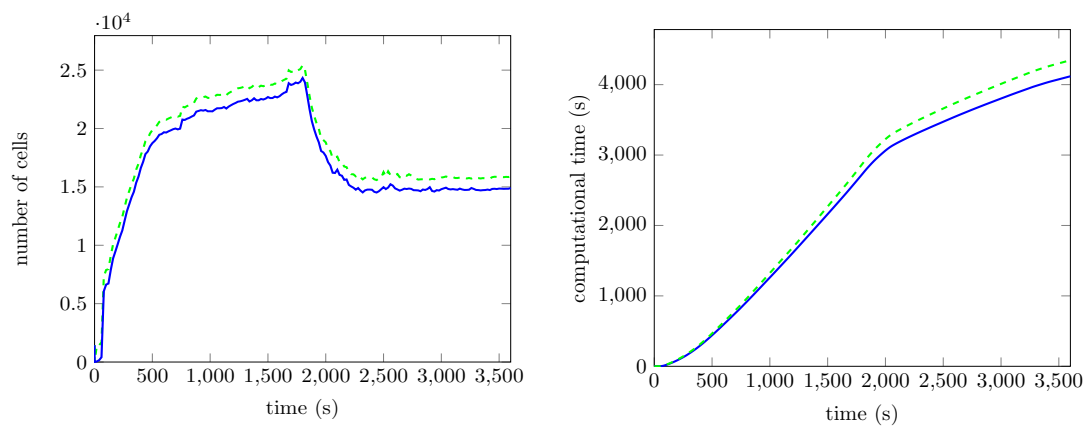


Figure 4.6: The comparisons of number of computational cells (left) and times (right) using adaptivity with dynamic DDM (solid) and without dynamic DDM (dashed).

## 4.2 Dam-Break Flood

In this experiment, the adaptive tree grid algorithm is employed to simulate the dam break flow on natural topography in case of the Mae-suai dam-break at Mae-suai, Chiangrai, Thailand. The earthquake, on 5 May 2014 with 6.3 magnitude on the Payao breakage, requires the simulation of the assumed Mae-suai dam-break to estimate risky flooding area that may occur by dam-break. We simulate the assumed event for creation informations on the flood problem, and to check the accuracy and efficiency of the adaptive tree grid algorithm.



Figure 4.7: The dam location and the measure points of the Mae-suai dam-break simulation.

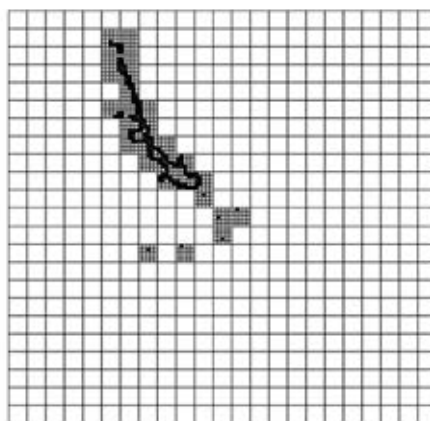


Figure 4.8: The initial grid of the Mae-suai dam-break simulation.



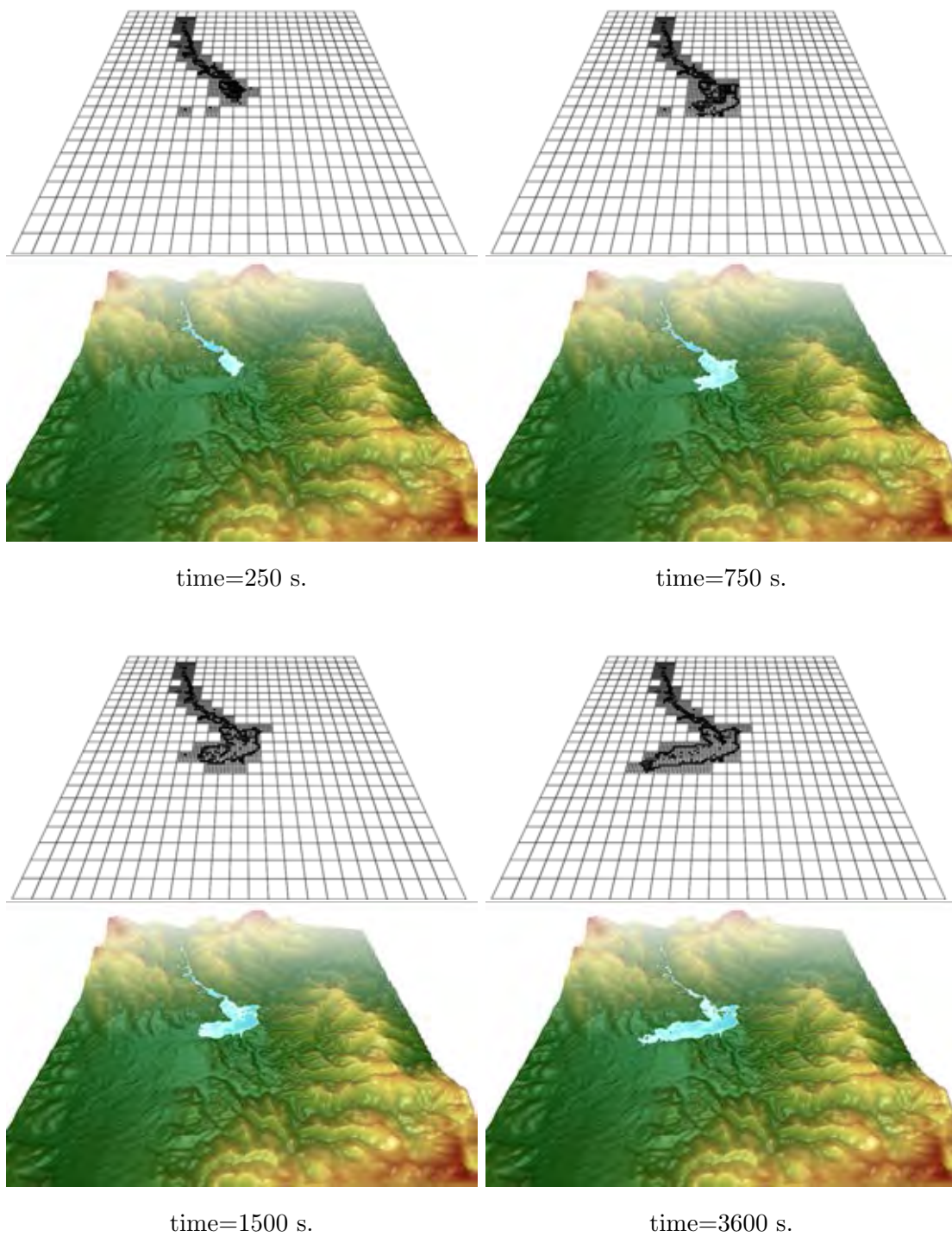


Figure 4.9: The 3D representation of the adapted grids of the dam-break overland flow simulation, at different times.

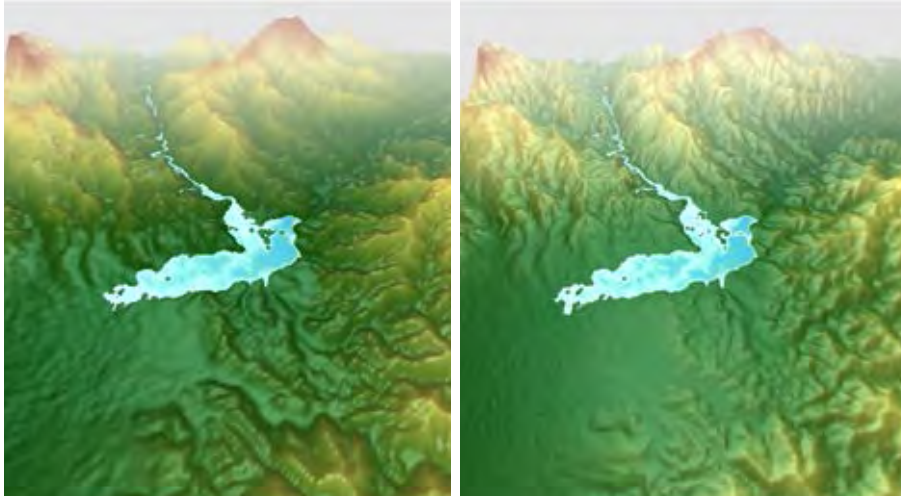


Figure 4.10: The 3D representation of the comparison for the adaptive grid (left) and the uniform finest grid (right) of the Mae-suai dam-break simulation at time=3600 s.

The simulation is performed on the domain of digital terrain data  $21,960 \text{ m} \times 21,960 \text{ m}$ , generated from Shuttle Radar Topography Mission (SRTM) data source shown in Figure 4.7. The approximated free surface elevation of the reservoir behind the dam is 507 m, follows the normal water level of Mae-suai dam. Also indicated in Figure 4.7 are six measure points for recording time histories of the model results during the simulation. The initial grid level 1 has  $25 \times 25$  cells with the maximum tree grid level is 5. The tree grid cells for children are designed as  $5 \times 5$  for level 1, and  $2 \times 2$  for levels 2, 3 and 4, while level 5 cells have no children. Therefore, the tree grid have resolution  $878.4 \text{ m} \times 878.4 \text{ m}$ ,  $175.68 \text{ m} \times 175.68 \text{ m}$ ,  $87.84 \text{ m} \times 87.84 \text{ m}$ ,  $43.92 \text{ m} \times 43.92 \text{ m}$ , and  $21.96 \text{ m} \times 21.96 \text{ m}$  for levels 1, 2, 3, 4 and 5, respectively. In doing this, For each levels, the topography values are obtained automatically via the bilinear interpolation, interpolate the topography data resolution 90 m of SRTM. The numerical experiment is simulated for 3600 s with the Manning's coefficient 0.033, and with parameters of the grid adaptivity  $\epsilon_1 = 1.0$ ,  $\epsilon_2 = 1.0$  and  $\xi = 0.01$ . Moreover, the boundary condition is defined as opened boundaries. The initial grid of the simulation is shown in Figure 4.8, where the finest grid cells are generated at the boundary of the reservoir. Moreover, the

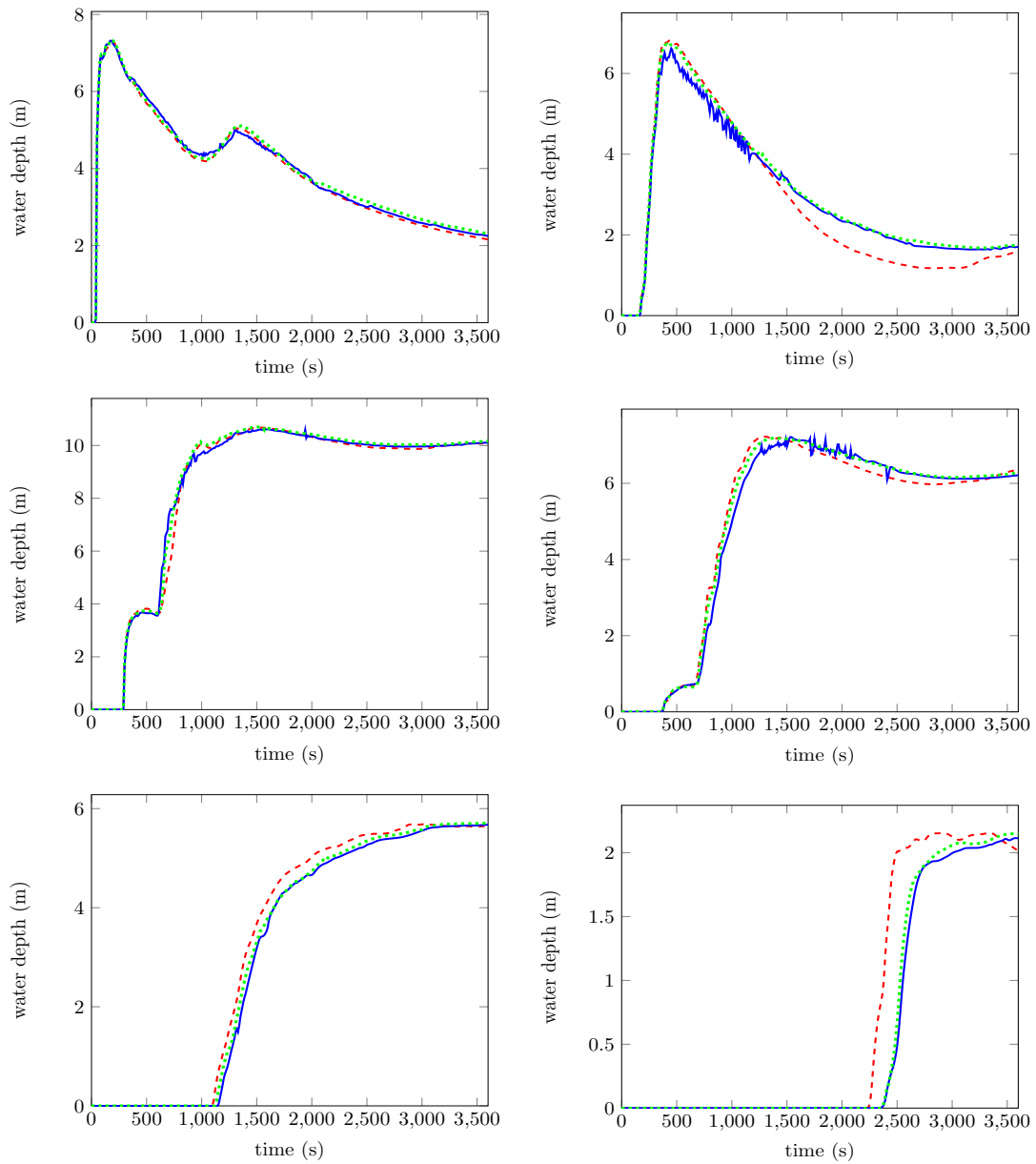


Figure 4.11: The comparison of the evolutions of the water depth at the measure points P1, P2, P3, P4, P5, P6 (from left to right and top to bottom). The solid, dotted and dashed lines refer to the simulations of the adaptive grid with  $\epsilon_1 = 1.0$  and  $\epsilon_2 = 1.0$ , the adaptive grid with  $\epsilon_1 = 1.0$  and  $\epsilon_2 = 0.5$ , and with the finest grid, respectively.



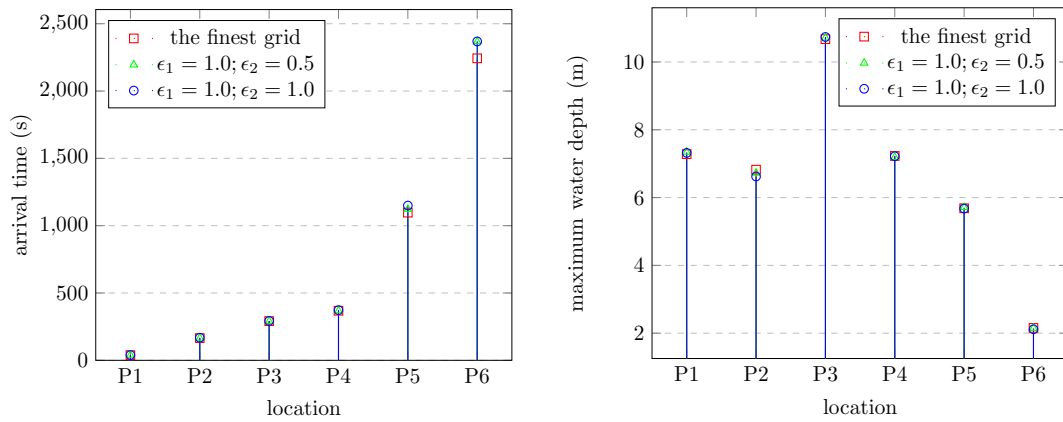


Figure 4.12: The comparisons of the arrival times (left) and the maximum water depth (right) at each locations between the adaptive grid and the uniform finest grid.

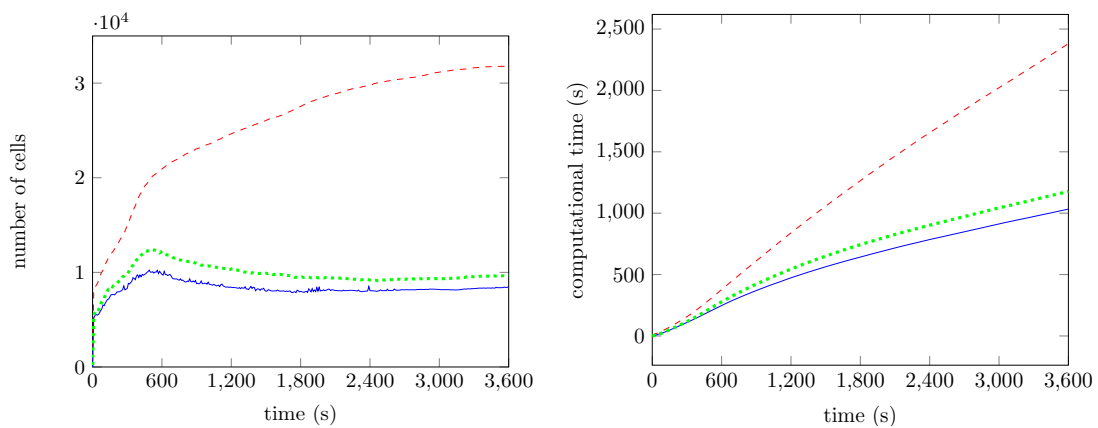


Figure 4.13: The comparisons of the number of computational cells (left) and the computational times (right). The solid, dotted and dashed lines refer the simulations of the adaptive grid with  $\epsilon_1 = 1.0$  and  $\epsilon_2 = 1.0$ , the adaptive grid with  $\epsilon_1 = 1.0$  and  $\epsilon_2 = 0.5$ , and the finest grid, respectively.

finest grid cells are also generated at the measured points through the simulation.

The 3D adaptive grid simulations of dam break flow for this experiment are presented at different times in Figure 4.9. The results illustrate that the high-resolution areas are automatically obtained by the adaptivity method in the corresponding regions where the free surface gradients are steep and the water depths are larger than 0.01 m, based on adaptivity parameters.

To check accuracy and efficiency of the dynamically adaptive technique, we performed the following test. We compared the dynamically adaptive simulation with non-adaptive simulations computed using only the dynamic DDM on the finest uniform grid, i.e., the level 4 grid as shown in Figure 4.10. The results show that the flooding region obtained by the dynamically adaptive scheme is nearly the same as that obtained from the finest grid.

Other comparisons are the accuracy and efficiency of the dynamical adaptive tree grid simulations with different adaptivity parameters compared with the finest grid simulations. Figures 4.11 and Figure 4.12 show the comparisons of the adaptive grid simulations and the finest grid simulation for the water depth, the arrival times, and the maximum water depth at each measured points, respectively. Figure 4.13 displays the number of computational cells and the computational times of the simulations. As illustrated in Figures 4.12 and 4.2, the results of adaptive tree grid simulations agree closely with the results of the finest grid simulation. However, the number of computational cells and the computational times of all the adaptive tree grid simulations are much less than that of the finest grid simulations as shown in Figure 4.13. This concludes that the adaptive grid technique is very efficient because it helps reducing the computational cells, and therefore, reduces the computational time while keeping the same accuracy as that of the finest grid simulation.

Also as seen in Figure 4.11, some results obtained from the adaptivity parameters,  $\epsilon_1 = 1.0$  and  $\epsilon_2 = 1.0$  are not smooth. These results may come from cycles of refining/coarsening between successive time-steps. However, the cycles are not found for the simulation using the adaptivity parameters,  $\epsilon_1 = 1.0$  and  $\epsilon_2 = 0.5$ . This indicates that we can prevent the cycles by using  $\epsilon_2 < \epsilon_1$ .

In this section, the simulations present the information on the flood problem for the assumed event. The flood risk maps animations in Figure 4.9 are generated to show the flood extension, together with water depths, flood arrival times, and maximum water depths, as shown in Figure 4.11 and 4.12. As indicated in Figure 4.12, the arrival time of the measured point, P6, the farthest point of Mae-suai place, about 2367.6729 s with the maximum water depth 2.1482 m. Therefore, if the Mae-suai dam break had occurred, the flood wave would rushed along the valley and down to the plain in 20 minutes, where damages could be predicted by the maximum water depths for each area. The informations for the assumed event are very useful for planning, preparation, prevention, and evacuation of the possibly serious disaster.

### 4.3 Breaching Flood Inundation

We illustrated in this section the numerical simulations using the developed model. The model with adaptive tree grid numerical scheme is applied to predict an idealist inundation event at Don-Muang place located at southern bank of Rangsit-Prayurasak canal, Bangkok, Thailand. The assumed flood event is a breaching flood of the flood defence at the Rangsit-Prayurasak canal which is a part of the Bangkok flood defence. We simulate the assumed event for obtaining informations on the flood problem, and checking the accuracy and efficiency of the adaptive tree grid algorithm.

The simulation is performed on the domain of digital terrain data  $28,800 \text{ m} \times 18,000 \text{ m}$ , generated from 90 m Shuttle Radar Topography Mission (SRTM) data source shown in Figure 4.14. As shown in Figure 4.14, a breach of the flood defence is defined at the bottom left corner (16,245 m, 14,535 m) and the top right corner (17,895 m, 14,850 m). Also indicated in Figure 4.14 are six measured points for recording time histories of the model during the simulation. The idealist hydrograph shown in Figure 4.16 is defined at the breach to drive the flood event. The

hydro-graph values are given by the inflow volumes,

$$D(t) = \begin{cases} \frac{m(U-L)e^{m(T-t)}}{(1+e^{m(T-t)})^2}, & t \leq T; \\ \frac{m(U-L)}{4}, & T < t \leq T + \Delta T_s; \\ \frac{m(U-L)e^{m(T+\Delta T_s-t)}}{(1+e^{m(T+\Delta T_s-t)})^2}, & T + \Delta T_s < t, \end{cases} \quad (4.1)$$

where  $D(t)$  is the inflow volumes per unit time ( $m^3/s$ ),  $L$  is the minimum value of the discharges,  $U$  is the maximum value of the discharges,  $m$  is the slope of the hydro-graph,  $T$  is the time at the total volume is  $\frac{U+L}{2}$ ,  $\Delta T_s$  is the duration of the study flood event, and  $t$  is the independent time variable.

The inflow volumes function is corresponded with the total volume of inflow function,

$$S(t) = \begin{cases} L + \frac{U-L}{1+e^{m(T-t)}}, & t \leq T; \\ L + \frac{U-L}{2} + (t-T)D(T), & T < t \leq T + \Delta T_s; \\ \Delta T_s D(T) + L + \frac{U-L}{1+e^{m(T+\Delta T_s-t)}}, & T + \Delta T_s < t. \end{cases} \quad (4.2)$$

In this simulation, we set the parameters as:  $U=1000000 \text{ m}^3$ ,  $L=0 \text{ m}^3$ ,  $m = 0.008$ ,  $T=7200 \text{ s}$ , and  $\Delta T_s=14400 \text{ s}$ .

The maximum tree grid level is 4 with resolution  $900 \text{ m} \times 900 \text{ m}$ ,  $180 \text{ m} \times 180 \text{ m}$ ,  $90 \text{ m} \times 90 \text{ m}$ , and  $45 \text{ m} \times 45 \text{ m}$  for levels 1, 2, 3, and 4, respectively; therefore, the initial grid level 1 has  $32 \times 20$  cells. The topography values are provided for levels 1-3 while the levels 4 is obtained automatically via the bilinear interpolation. The tree grid cells for children are designed as  $5 \times 5$  for level 1, and  $2 \times 2$  for levels 2 and 3, while level 4 cells have no children. The numerical experiment is simulated for 44,000 s with the Manning's coefficient 0.001, and with parameters of the grid adaptivity  $\epsilon_1 = 0.5$ ,  $\epsilon_2 = 0.25$ , and  $\xi = 0.01$ . Moreover, the boundary condition is defined as opened boundaries while at the Bangkok flood defence (red line in Figure 4.14) is defined as the closed boundaries. The initial grid of the simulation is shown in Figure 4.15. The finest grid cells are generated at the breach of the flood defence and at those measured points through the simulation.

To check accuracy and efficiency of the dynamically adaptive technique, we perform the following test. We compare the dynamically adaptive simulations with



Figure 4.14: The breach location and the measured points of the breaching flood inundation simulation.

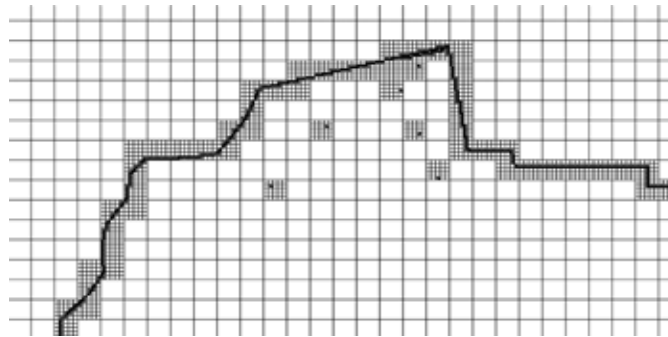


Figure 4.15: The initial grid of the breaching flood inundation simulation.

non-adaptive simulations computed using only the dynamic DDM on the finest uniform grid, the level 4 grid with  $640 \times 400$  smallest cells. The 2D simulations of flood inundation for this experiment are presented at different times, shown in Figures 4.17 and 4.18. Figures 4.17a and 4.18a present the inundation extent and water depth at  $t=2$  h and  $t=12$  h. The corresponding predictions on the finest grid are shown in Figures 4.17b and 4.18b. The simulation results show that the flood plain obtained by the dynamically adaptive scheme are nearly the same as simulations that obtained from the finest grid. Adapted tree grids are presented in Figures 4.17c and 4.18c. The adapted tree grids show that the high resolution areas are automatically obtained by the adaptivity in corresponding to the regions where the free surface gradients are steep and have water depths more than 0.01 m, according to adaptivity parameters.

In order to confirm the mass conservation of the results, the total volume of

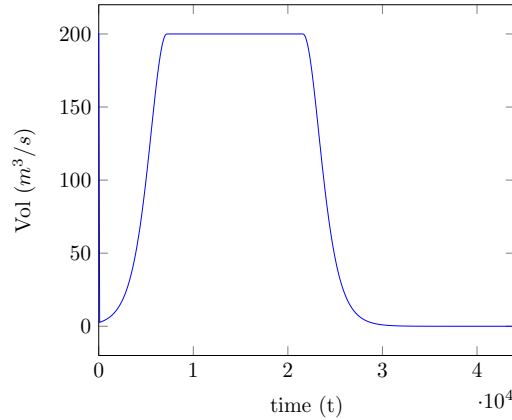


Figure 4.16: The idealist hydro graph defined at the breach to drive the flood event.

water inside the computational domain is measured and compared with the total volume of mass through the breach in Figure 4.19. The total volume of mass values are given by equation (4.2). For this case, since no outflow is induced, the flow mass should be the same as those accumulated inside the domain. Evidently, excellent mass conservation property of the adaptive tree grid model is confirmed.

Further comparisons are the accuracy and efficiency of the dynamical adaptive tree grid simulation undertaken with half and double the adaptivity parameters  $\epsilon_1$  and  $\epsilon_2$ , i.e.  $\epsilon_1 = 0.25; \epsilon_2 = 0.125$  and  $\epsilon_1 = 1.0; \epsilon_2 = 0.5$ , respectively. Figure 4.20 presents the comparisons of the adaptive grid simulations and the finest grid simulation for the arrival times and the maximum water depth at each measured points. Figure 4.21 shows the number of computational cells and the computational times of the simulations. As indicated in Figure 4.20, The results of all adaptive tree grid simulations agree closely with the finest grid simulations. However, the number of computational cells and the computational times of all adaptive tree grid simulations are less than that of the finest grid simulations, as seen in Figure 4.21. These show that the adaptive grid technique is more efficient because it helps reducing the computational cells, and therefore, reduces the computational times while keeping the same accuracy as that of the finest grid simulation.

Also indicated in Figure 4.21, the number of computational cells and the computational times tend to reduce as the adaptivity parameters for  $\epsilon_1$  and  $\epsilon_2$  increase.

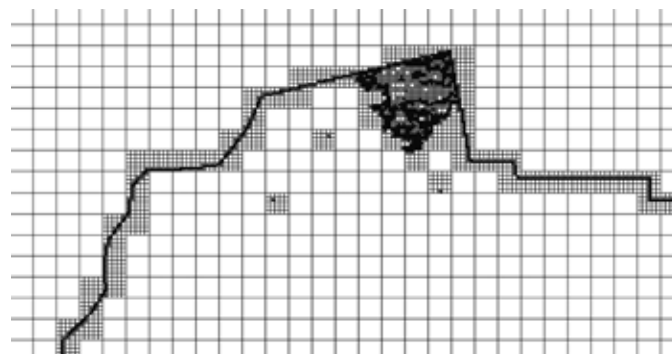
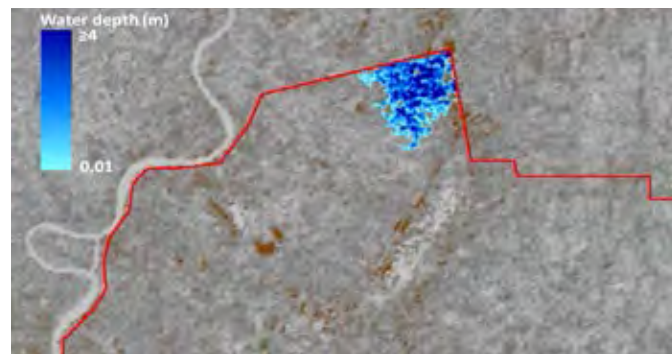
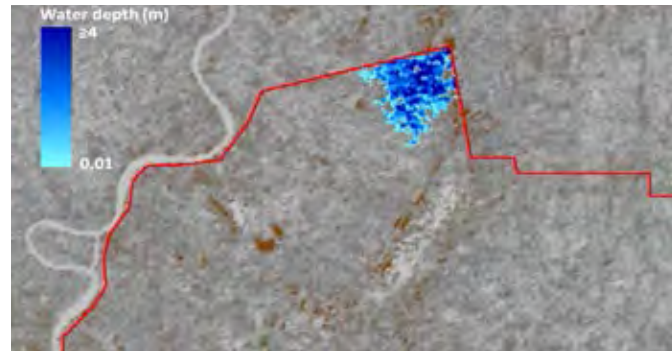
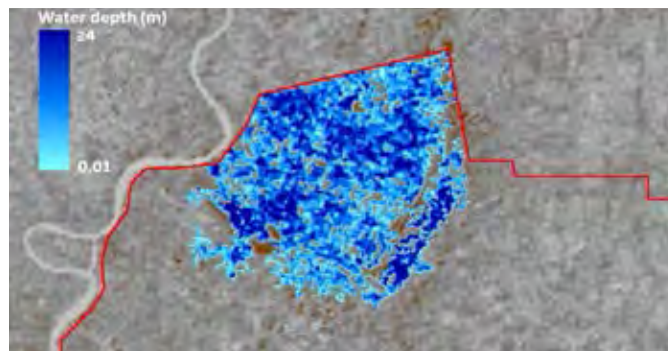
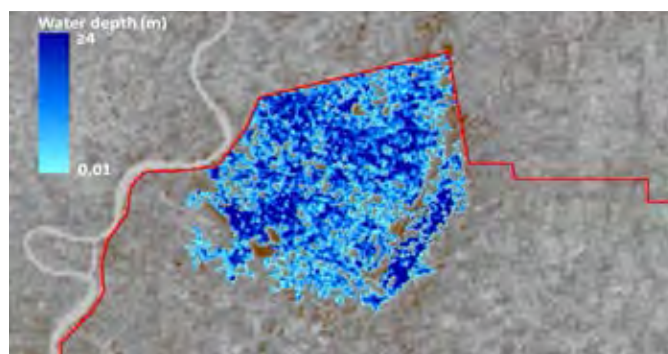


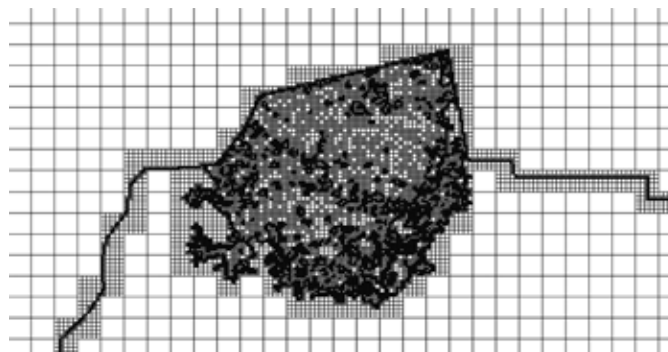
Figure 4.17: Flood inundation at  $t=2$  h: (a) the flood simulation by the adaptive grid, (b) the flood simulation by the finest grid, and (c) the adapted tree grid.



(a)



(b)



(c)

Figure 4.18: Flood inundation at  $t=12$  h: (a) the flood simulation by the adaptive grid, (b) the flood simulation by the finest grid, and (c) the adapted tree grid.



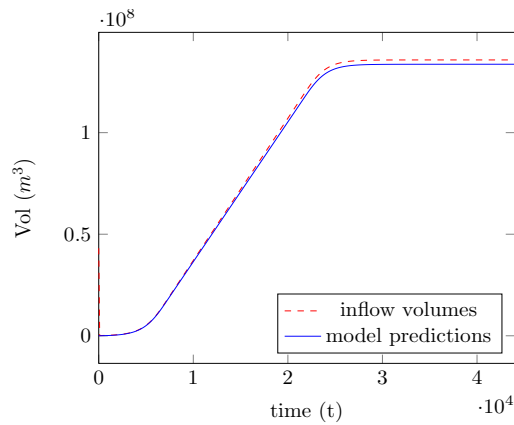


Figure 4.19: The comparisons of the total volume of water inside the computational domain (model predictions) and the total volume of mass through the breach (inflow volumes).

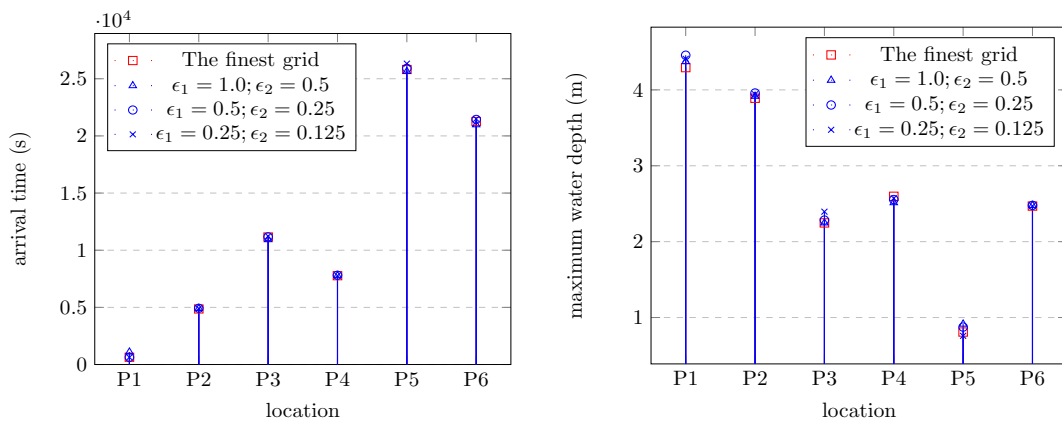


Figure 4.20: The comparisons of arrival times (left) and maximum water depths (right) of each the locations for the adaptive grid and the uniform finest grid.

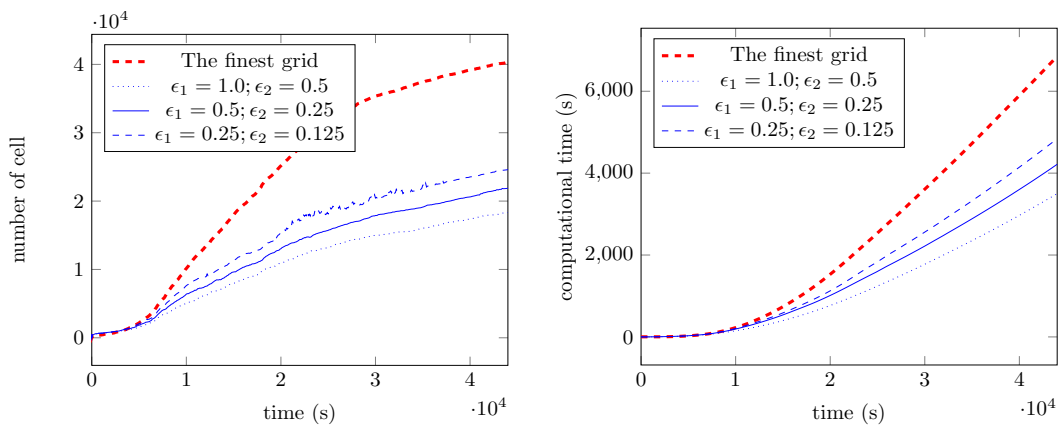


Figure 4.21: The comparisons of number of computational cells (left) and computational times (right) for the adaptive grid and the uniform finest grid.

These indicate that we can utilize the adaptivity parameters to reduce the number of computational cells and the computational times, whenever the results are still accurate.

Also presented in these results are information on flood problem for the assumed event. Flood risk maps animation in Figures 4.17 and 4.18 are generated to show flood extension, and together with flood arrival times and maximum water depths in Figure 4.20. As indicated in Figure 4.20, the arrival time of the measured point P5 at Don-Muang air port is about 25914.4838 s with maximum water depth 0.87 m. Therefore, if the assumed breaching flood occurs, the flood wave will arrive Don-Muang air port in about 7 h, where damages can be estimated by the maximum water depths for each area. The informations for the assumed event are useful for planning, preparation, prevention, and evacuation of the possibly serious disaster.

#### **4.4 Conclusion**

In this section, we applied the dynamically adaptive tree grid algorithm for simulation and visualization of water overland flow, which are the rain-water overland flow simulation, the simulation of a dam-break on natural topography, and the breaching flood simulation. Numerical simulations show that the developed algorithm is suitable for simulating and visualizing flow of water on natural topography and creating informations on floods problem. The adaptivity and dynamic DDM help making the computation efficient while keeping the same accuracy as that of the computation on the finest grid. Therefore, it has a potential for practical usages, and can be applied to other flooding simulations on natural topography such as river flood, including utilization for the water management system.

# CHAPTER V

## SIMPLIFICATION OF THE DYNAMICALLY ADAPTIVE TREE GRID ALGORITHM

The presented adaptive tree grid algorithm in Chapter 3 is developed based on second order high-resolution finite volume methods, in terms of time and space. However, for a simulation of the flood flow on natural topography, the second order high-resolution scheme may not be significant when compared with uncertainties in topographic data and information available for calibration of friction parameterization. Therefore, if first order scheme is implemented, the adaptive algorithm should provide further significant improvements in computational efficiencies (Liang [31]) as motivation that simple schemes should reduce the computational resources and simulation run times. Moreover, since validations of the first order schemes with adaptive tree grid techniques have often been ignored by researchers, it becomes attention to validate the scheme for flood simulations.

The purpose of this chapter is to develop the dynamically adaptive grid algorithm based on the first order finite volume scheme for simulation and visualization of flood flow on natural topography. The developed algorithm is constructed based on a well-balanced, positivity-preserving first order scheme of Audusse [3], and equipped with the dynamically adaptive general rectangular tree grid method for shallow water equations.

### 5.1 Numerical Algorithm

We present in this section the first order finite volume algorithm based on Audusse's work [3] for solving of the system (2.20) with the dynamically adaptive tree grid technique. By the first order scheme, the slope limiter in space (or reconstruction procedure in §3.2.3) is not used, and therefore, TVD Runge-Kutta iteration, to

prevent spurious oscillations is unnecessary, two procedures which are the slope limiter in space and the TVD Runge-Kutta iteration are excluded for this algorithm. The reasons for adopting this algorithm are as follows:

1. The algorithm has a potential for the water flow simulation over natural topography;
2. Based on rectangular tree grids, the development of algorithm is much simpler and requires less computational costs than the higher order schemes.

The computational details are provided as follows.

### 5.1.1 Finite volume formulation

In this chapter, the finite volume scheme for shallow water equations performs on discretized domain that consists of rectangular cells as shown in Figure 5.1, and can be formulated as described in section 3.2.1.

However, in order to keep the scheme as simple as first order and to reduce computational cost, the time discretization of equation (3.5) is done using the Euler's method, namely,

$$\vec{W}_i^{t+\Delta t} = \vec{W}_i^t - \frac{\Delta t}{\Delta_i} \sum_k \hat{F}_k^t \cdot \vec{n}_k \Delta \tau_k + \Delta t \vec{S}_i^t, \quad (5.1)$$

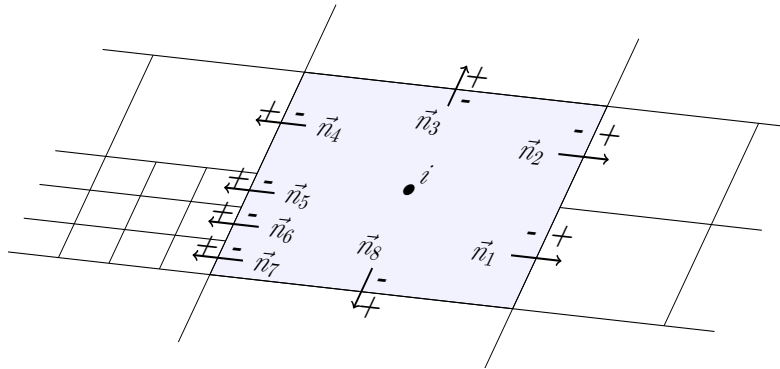


Figure 5.1: The main control volume and descriptions for each cell.

where  $k$  is index of the sub interface between the cell  $i$  and the neighbor cell  $k$ ,  $\Delta\tau_k$  is size of the sub interface  $k$ . Here,  $\Delta t$  is time step size, limited by the CFL condition as section 3.2.6.

The equation (5.1) is constituting the first order finite volume formulation over each rectangular cell of the grid, and  $\hat{F}_k \cdot \vec{n}_k$  can be calculated using the formulation based on the HLL schemes as described in section 3.2.2. However, by using the first order Audusse's scheme [3], the terms that gives well-balanced scheme proposed in equations (3.8) and (3.9) are modified as

$$\vec{z}_x = [0 \quad -\frac{g}{2}(\hat{h}_k^-)^2 \quad 0]^T, \quad \vec{z}_y = [0 \quad 0 \quad -\frac{g}{2}(\hat{h}_k^-)^2]^T. \quad (5.2)$$

For this algorithm, because the reconstruction procedure in section 3.2.3 is not used, the superscripts - and + refer to the non-reconstructed values within the considered cell  $i$  and the neighbor cell  $k$ , respectively.

For the source term  $\vec{S}_i^t$  in equation (5.1) consists of the rainfall rate and the friction forces, they can be computed by the scheme described in section 3.2.5.

### 5.1.2 Boundary condition

Two types of boundary condition, proposed in this chapter are the opened and closed boundary conditions. The closed boundary condition is the same as declared in section 3.2.7. However, the opened boundary condition is given by

$$[h_m \ u_m \ v_m \ z_m]^T = [0 \ 0 \ 0 \ 0]^T, \quad (5.3)$$

where the subscript  $m$  refers to the spatial index of the boundary cells.

### 5.1.3 Grid adaptivity method

The designing grid adaptivity technique consists of three parts, the designing of the tree grid, the refining and coarsening criteria, and the defining of cell-dependent values. For the tree grid, the designed scheme is constructed based on the general rectangular tree grid described in Section 3.1.

For refining and coarsening criteria, we use a simple criterion based on two conditions, the gradient of the free surface,  $H = h + z$ , and the water depth. The checking of refinement and coarsening are performed as already described in section 3.2.8. However, when a cell is refined, the children cells are created in next level that is not maximum level, and they are defined by the same values as their parent cell, i.e.

$$\vec{W}_j^{l+1} = \vec{W}_i^l, \quad (5.4)$$

where  $i$  and  $j$  refer the spatial indices of the parent cell and the child cell, respectively. For the coarsening, the values of the merged parent cell is defined by the averages as:

$$\vec{W}_i^l = \frac{1}{n_i^l \times n_i^l} \sum_{j \in \Omega_i} \vec{W}_j^{l+1}, \quad (5.5)$$

where  $n_i^l \times n_i^l$  is the number of children.

As described in section 3.2.9, the height of topography,  $z$ , can be defined by the bilinear interpolation technique for cases of the simulation using topographic data grids.

#### 5.1.4 Dynamic domain defining method

As described in section 3.2.10, in order to improve the computational efficiency, the cells are checked to identify whether the cell is inside the computational domain or not. By the checking, the cell is excluded from the computational domain if the cell and its neighbor cells are all dry. On the other hand, the cell and its neighbor cells are included to the computational domain whenever these cells are not in the domain. Only grid cells within the computational domain are adapted and computed in order to minimize the total number of the computational cells.

#### 5.1.5 Algorithm overview

We present here an overview of the algorithm for the computational scheme presented above. The algorithm is simplification of the algorithm in §3.2.11 to exclude

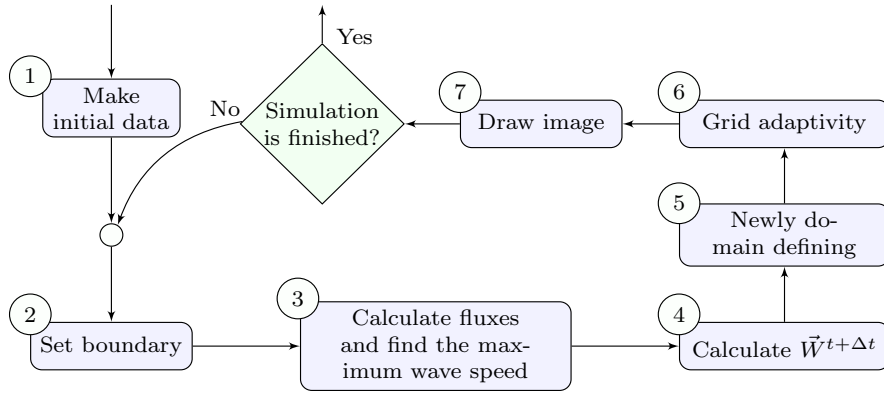


Figure 5.2: The algorithm overview.

the reconstruction step and TVD Runge-Kutta iteration. The simplified algorithm consists of several steps describing the calculations for each grid cell, as illustrated in Figure 5.2. The details for each steps can be described as follows:

- Step ① : Set the initial values, the cell sizes, the number of children in each level, and the maximum of tree grid levels.
- Step ② : Set the computational grids boundaries, see §5.1.2.
- Step ③ : Calculate the numerical fluxes and the maximum wave speed.
- Step ④: Calculate the solutions of  $\vec{W}^{t+\Delta t}$  as equation (5.1).
- Step ⑤ : Define a new domain based on the dynamic DDM.
- Step ⑥ : Conduct the grids adaptivity technique, see §5.1.3.

#### I Refinement procedure

- i For each cells whose level is not maximum level
  - Mark cells that satisfy the refinement conditions.
- ii For each marked cells
  - Create new children cells, and assign the dependent values as equation (5.4).

## II Coarsening procedure

- i For each parent cells whose children have no child
  - Average the children values defined in (5.5);
  - Check the average with the conditions of the coarsening;
  - If one of the conditions is false, then merge all children cells into a parent cell, and assign the dependent values.
- Step ⑦ : Visualize the solutions in 2D and 3D by OpenGL.
- Repeat Step ② to ⑦ until the simulation is finished.

## 5.2 Results

We illustrate in this section the numerical simulations using the developed algorithm. The simulations consist of two tests as follows: the dam break simulation compared with real experimental data, and the simulation of dam break overland flow compared the performance with the simulation on the finest grid using the second order finite volume scheme. The coding and simulations are performed using 2 GHz Intel Core i7 with RAM 6 GB.

### 5.2.1 Dam break experiment simulation

In this test, the adaptive grid first order scheme simulation of dam break with an obstacle was performed and compared with the real experimental data of Kleefsman et al [26]. Figure 5.3 shows geometry of the physical experiment, where  $H_1$ ,  $H_2$ ,  $H_3$  and  $H_4$  are corresponding to the vertical wave probes. In the figure, all dimensions are in meters.

In this simulation, the adaptivity of the tree grid is employed with 3 levels, the level 1 has 25x75 grid cells with the dimension of children for levels 1 and 2 are 3x3 and 2x2, respectively, and the level 3 cells have no children. The duration of the simulation is 6 s and performed with the Manning's coefficient 0.032, and with parameters of the grid adaptivity  $\epsilon_1 = \epsilon_2 = 0.007$  and  $\xi = 0.01$ . The boundary



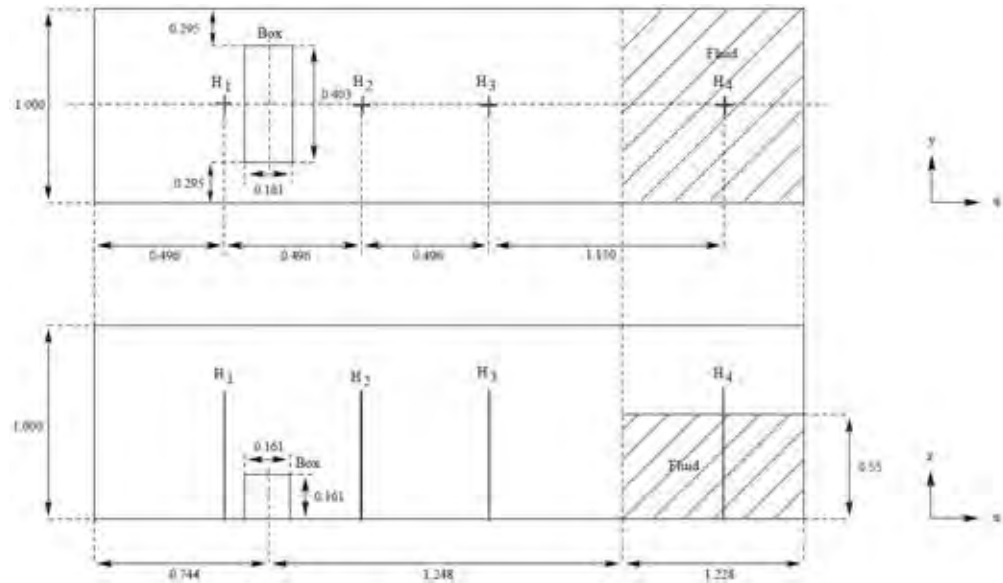


Figure 5.3: The geometry of the physical experiment of Kleefsman et al.

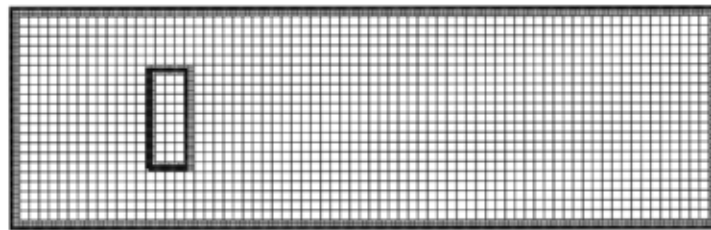


Figure 5.4: The initial grid of the dam-break simulation.

condition is considered as closed boundaries. The initial grid is shown in Figure 5.4.

The simulation results of the behavior of water flow at time  $t=0.4$  s and 0.56 s are shown in Figures 5.5a and 5.5d. The 3D representation of the results agree with the experiment in Figures 5.5b and 5.5e. The corresponding automatically adapted grids are shown in Figures 5.5c and 5.5f. The adapted grid of the results show that the high resolution regions with high slope free surface are obtained automatically via the adaptive technique. Furthermore, the comparison of the evolutions of the water depths at each the probes are shown in Figure 5.6. The results agree closely with the experimental data. In order to check the efficiency of the adaptive grid first

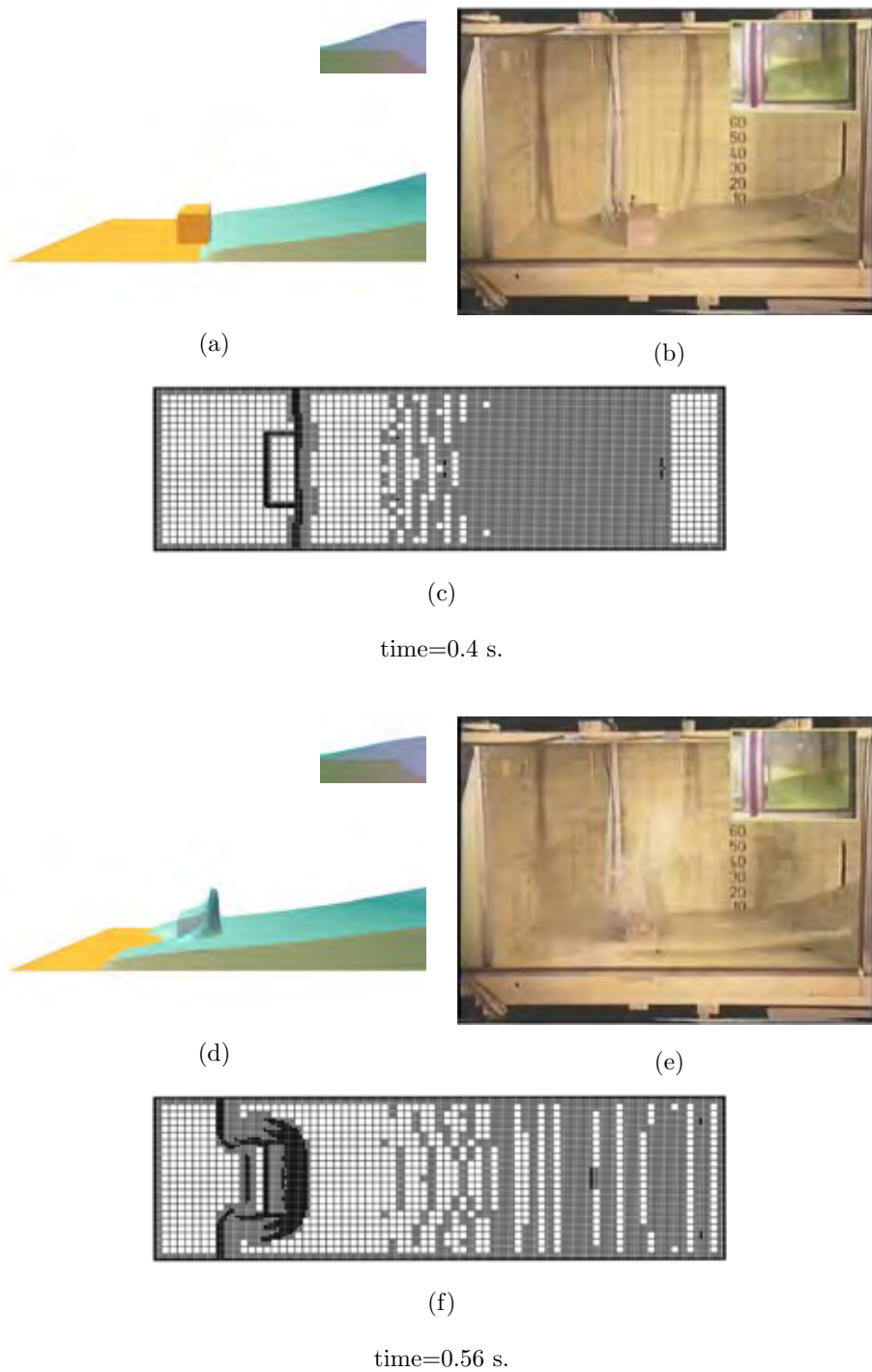


Figure 5.5: The 3D representations and 2D adaptive grids of the dam break simulation compared with experiment at different times.

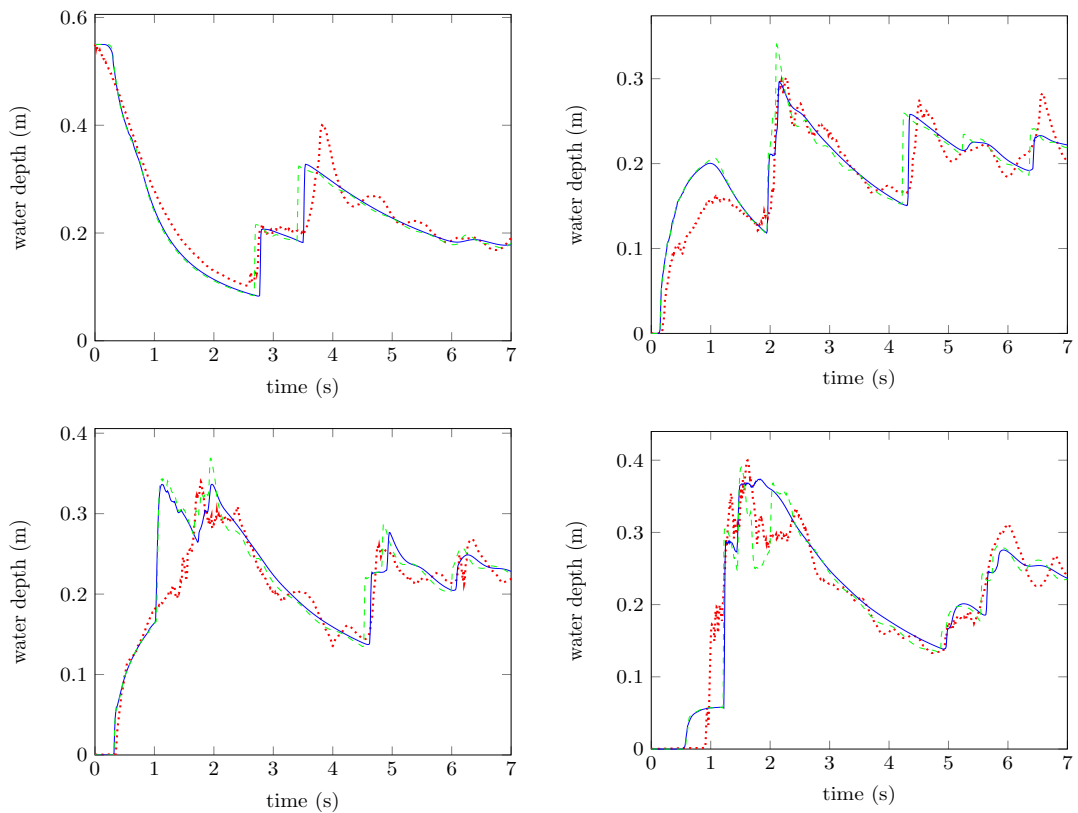


Figure 5.6: The comparisons of the evolutions of the water depths at the reservoir H4 (top-left), H3 (top-right), H2 (bottom-left), and H1 (bottom-right), for the experimental data (dotted), the adaptive grid second order scheme (dashed), and the adaptive grid first order scheme (solid).

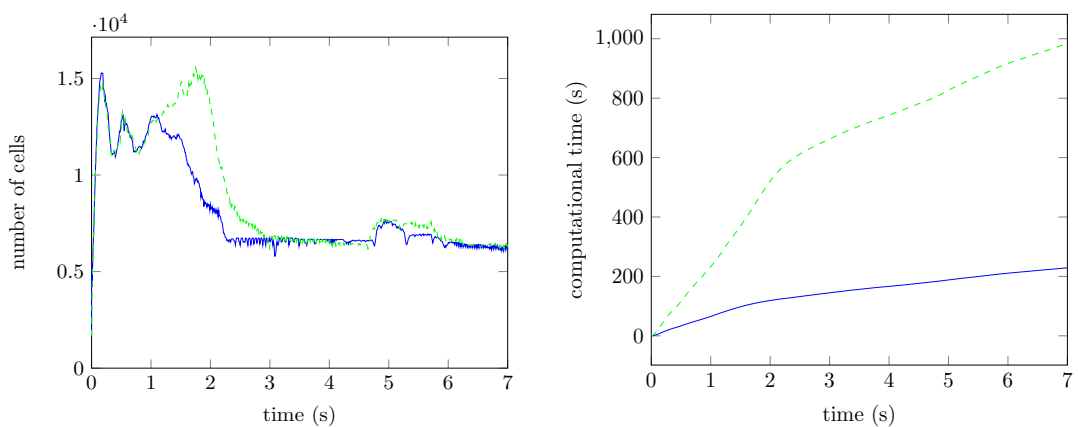


Figure 5.7: The comparisons of the number of computational cells (left) and the computational times (right) for the adaptive grid second order scheme (dashed) and the adaptive grid first order scheme (solid).

order scheme, we compared the results with a dynamically adaptive grid second order scheme simulation (the adaptive grid second order scheme was described in Chapter 3). The results of the both adaptive grid schemes, compared between first and second order schemes, are similarly as shown in Figure 5.6. The number of computational cells of the adaptive grid first order scheme are nearly the same as that obtained from the adaptive grid of second order scheme, as shown in Figure 5.7 (left). However, as shown in Figure 5.7 (right), the computational times of the adaptive grid first order scheme are much less than the computational times of the adaptive grid second order scheme with speedup about 4 times faster. These indicate that the adaptive grid first order scheme is efficient for the dam-break experimental simulations in general.

## 5.2.2 Dam break flow on natural topography

In this experiment, the developed model with adaptive tree grid first order scheme is employed to simulate the dam break flow on natural topography in case of the Malpasset dam break that occurred on the Reyan river valley in southern France, on the 2<sup>nd</sup> of December 1959. The flood wave rushed along the valley and down to the plain area in 20 minutes, where the city of Frejus is located. This flood event caused about \$68 million economic losses and 421 casualties (Wang [44]).

The simulation is performed on the domain of digital terrain data  $16200 \text{ m} \times 19800 \text{ m}$ , generated from Shuttle Radar Topography Mission (SRTM) data source shown in Figure 5.8. The approximate free surface elevation of the reservoir behind the dam is 100 m. Also indicated in Figure 5.8 are six measured points for recording time histories of the model results during the simulation. The maximum tree grid level is 5 with resolution  $900 \text{ m} \times 900 \text{ m}$ ,  $180 \text{ m} \times 180 \text{ m}$ ,  $90 \text{ m} \times 90 \text{ m}$ ,  $45 \text{ m} \times 45 \text{ m}$ , and  $15 \text{ m} \times 15 \text{ m}$  for levels 1, 2, 3, 4, and 5, respectively; therefore, the initial grid level 1 has  $18 \times 22$  cells. The topography values are provided for levels 1-3 while the levels 4-5 are obtained automatically via the bilinear interpolation. The tree grid cells for children are designed as  $5 \times 5$  for level 1,  $2 \times 2$  for levels 2 and 3, and  $3 \times 3$  for level 4, while level 5 cells have no children. The numerical



Figure 5.8: The evaluated locations of the Malpasset dam-break simulation.

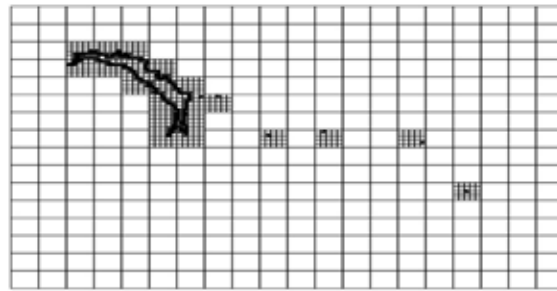


Figure 5.9: The initial grid of the Malpasset dam-break simulation.

experiment is simulated for 3000 s with the Manning's coefficient 0.033, and with parameters of the grid adaptivity  $\epsilon_1 = \epsilon_2 = 0.1$  and  $\xi = 0.01$ . Moreover, the boundary condition is considered as opened boundaries. The initial grid of the simulation is shown in Figure 5.9, where the finest grid cells are generated at the boundary of the reservoir. Moreover, the finest grid cells are also generated at the measured points throughout the simulation.

The 3D adaptive grid first order simulations of dam break flow for this experiment are presented at different times in Figure 5.10. The results show that the high-resolution areas are automatically obtained by the adaptivity method in corresponding to the regions where the free surface gradients are steep and have water depths more than 0.01 m, according to the adaptivity parameters.

To check the accuracy and efficiency of the dynamically adaptive scheme we performed the following test. We compared the dynamic ally adaptive grid first order scheme simulation with the two second order scheme simulations computed by adaptive and non-adaptive grid schemes. The non adaptive grid simulation

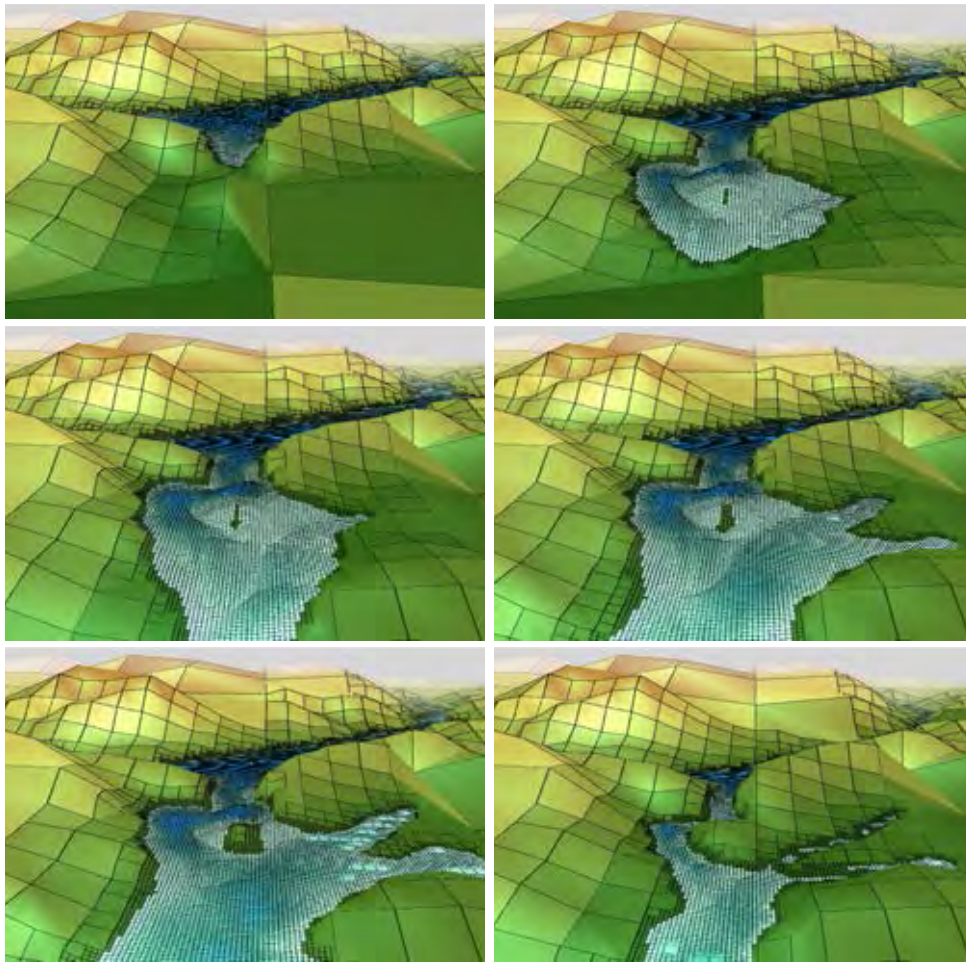


Figure 5.10: The 3D adaptive grid representation of the dam-break simulations at different times: 10 s, 100 s, 150 s, 300 s, 600 s, and 3000 s, respectively.

is simulated on finest uniform grid, level 5 grid (smallest cells with  $1080 \times 1320$  cells) using only the dynamic DDM. The comparisons of 3D representations for the flood simulations of the adaptive grid first order scheme at different times are shown in Figure 5.11. The flood area simulations obtained by the adaptive grid first order scheme are nearly the same as that obtained from the finest uniform grid. The comparison of the evolutions of the water depth at each measured points are shown in Figure 5.12. The maximum of water depths and the arrival times at each measured points are shown in Figure 5.13. The results obtained by the dynamically adaptive grid schemes agree closely the finest grid (level 5 grid). These show the accuracy of the adaptive grid schemes. The efficiency of the adaptive



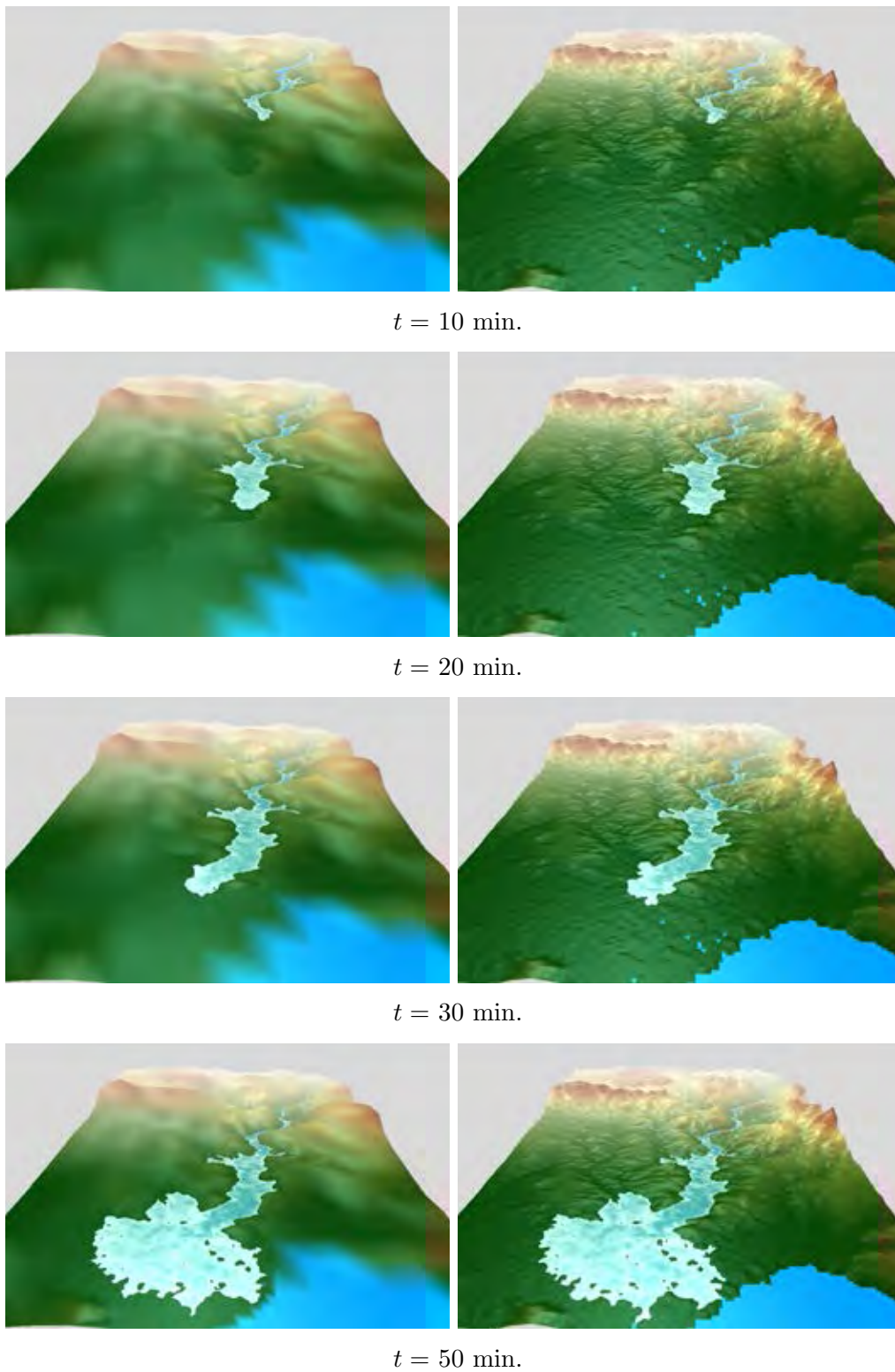


Figure 5.11: The comparisons of the adaptive grid first order scheme simulations (left) with the finest uniform grid simulations (right) at different times  $t$ .

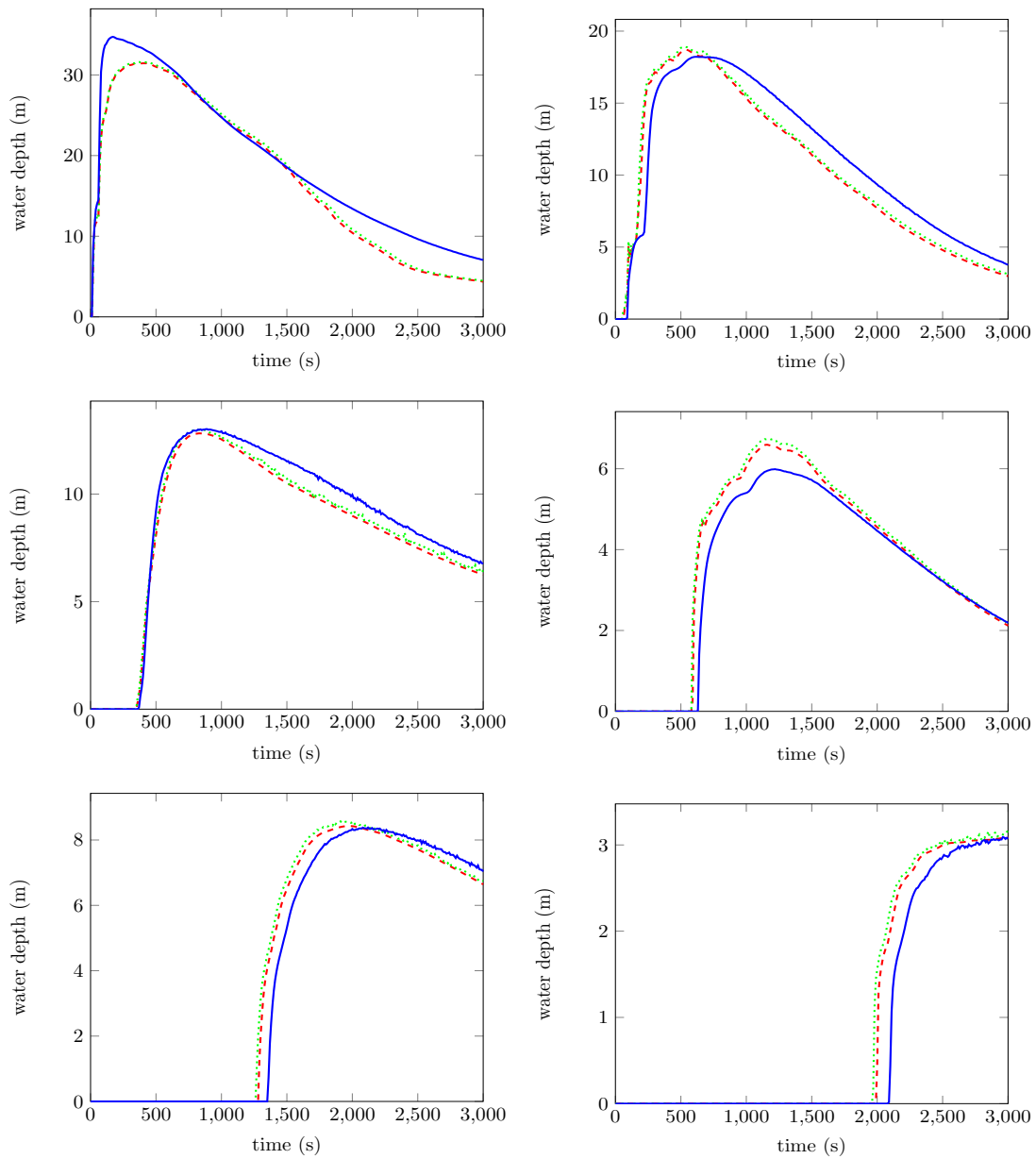


Figure 5.12: The comparison of the evolutions of the water depth in the measured points P1, P2, P3, P4, P5, P6 (from left to right and top to bottom). The solid, dotted and dashed lines refer to the simulations of the adaptive grid first order scheme, the adaptive grid second order scheme, and the finest grid second order scheme, respectively.



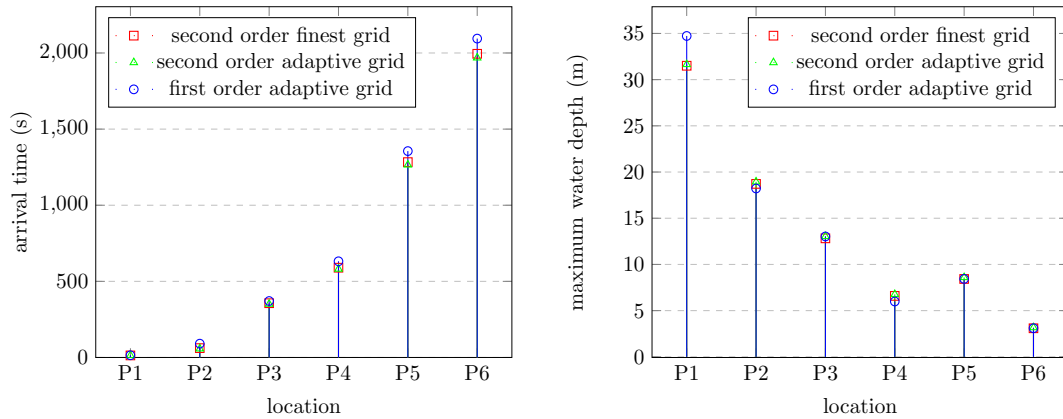


Figure 5.13: The comparisons of the arrival times (left) and the maximum water depths (right) at each location for the adaptive grid and the uniform finest grid.

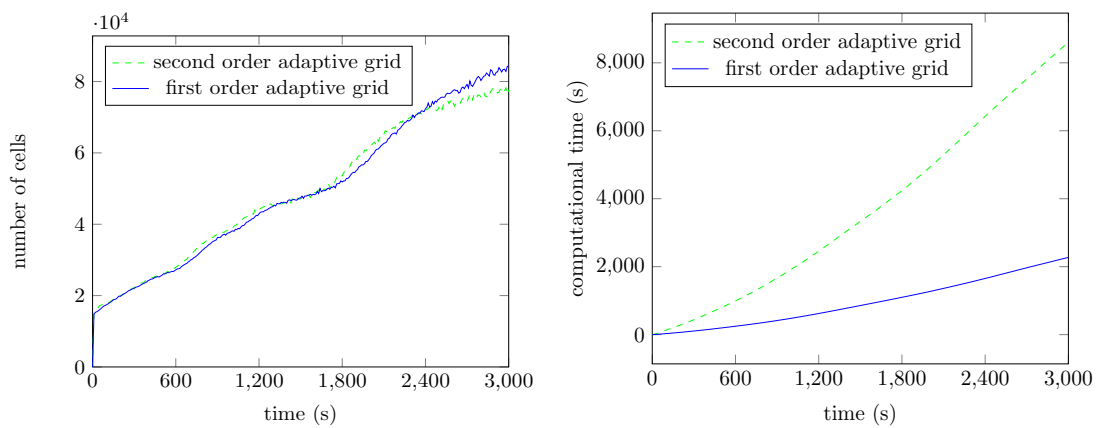


Figure 5.14: The comparisons of the number of computational cells (left) and the computational times (right) for the adaptive grid second order scheme and the adaptive grid first order scheme.

grid schemes is presented by comparing the number of computational cells and the computational times for the simulations as shown in Figure 5.14. The results show that the adaptive schemes are more efficient because they help reducing the computational cells (from 120,672 cells by the finest uniform grid to less than 90,000 cells by the adaptive grid first order scheme, and less than 80,000 cells by the adaptive grid second order scheme), and therefore, reduces the computational time (from 9337.3630 s by the finest uniform grid to 2272.5610 s by the adaptive grid first order scheme, and 8602.6709 s by the adaptive grid second order scheme), while keeping the same accuracy as that of the finest uniform level 5 grid. The number of the computational cells of the adaptive first grid schemes are nearly the same as that of the adaptive grid second order scheme as shown in Figure 5.14 (left). Moreover, the adaptive grid first order scheme results are almost the same as that obtained from the adaptive grid second order scheme. Nevertheless, since the first order scheme is easier than the second order scheme for the computation on the adaptive grids, the first order scheme reduces computational times when compared with the second order scheme, by speeding up 3.7 times faster.

### 5.3 Conclusion

In this chapter, the dam-break overland flow simulations are developed by applying the first order Audusse's finite volume scheme for the shallow water equations to solved with the dynamically adaptive general rectangular tree grid method. The simulation results are compared with the real experimental data and also with simulations of the second order scheme on the finest grid. The comparisons show that the developed algorithm for simulation is very efficient in terms of computational costs, by reducing the number of the computational cells and the computational time without losing much accuracy in the results. Therefore, it has potential in practical usages in order to product flooding areas. The scheme can be applied to simulate the other situations on flood problems.

# CHAPTER VI

## DIFFUSION MODELING ON DYNAMICALLY ADAPTIVE TREE GRID FOR RAINFALL OVERLAND FLOW SIMULATION

Most of researchers have used the shallow water equations for the flood simulation. However, several flood models (such as [12], [38], and [44]) have been developed and successfully applied using simple equations and reduced complexity approaches. The motivation of the flood models is that solving the simple equations should reduce the computational burden and the simulation run times (Dottori [12]). Diffusion and wave equations were used for description of water and wave propagations (see in [1, 8, 38]). However, both equations were not used to describe water and wave propagations in rugged terrain.

The purpose of this section is to develop an algorithm based on adaptive grid finite volume method in solving diffusion equations for water flows in order to simulate and visualize rainfall overland flows on natural topography. The adaptivity is developed based on general rectangular tree grids and in combination with the dynamic DDM.

### 6.1 The Mathematical Model

In this work, we develop a two-dimensional model for water flow based on diffusion equations for determining the behavior of rainfall flow above bottom elevation without infiltration. The model is written as

$$\frac{\partial H}{\partial t} = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + R, \quad (6.1)$$

where  $H = z + h$  is the free surface elevation,  $h$  is the water depth,  $z$  is the height of topography,  $R$  is the rainfall rate, while

$$f = \alpha \frac{\partial H}{\partial x}, \quad g = \alpha \frac{\partial H}{\partial y} \quad (6.2)$$

are fluxes in the directions of the  $x$  and  $y$  coordinate axis, respectively. In these equations,  $\alpha$  is the water diffusion coefficient and  $t$  is time.

## 6.2 Numerical Algorithm

We present in this section the numerical solution to the model equations described in the previous section. The solution can be generated using conservative finite volume methods. The details are described as follows.

### 6.2.1 Finite volume formulation

In this chapter, the finite volume formulation performs on physical domain that comprises of regularized cells. Each cell is considered as a main control volume whose boundary is formed by the direct walls surrounding it. An example of a single main control volume is shown in Figure 6.1.

By integrating equation (6.1) over an arbitrary cell domain  $\Omega_i$  and using the Green's theorem, we have

$$\int_{\Omega_i} \frac{\partial H}{\partial t} d\Omega = \oint_{\tau} \vec{F} \cdot \vec{n} d\tau + \int_{\Omega_i} R d\Omega, \quad (6.3)$$

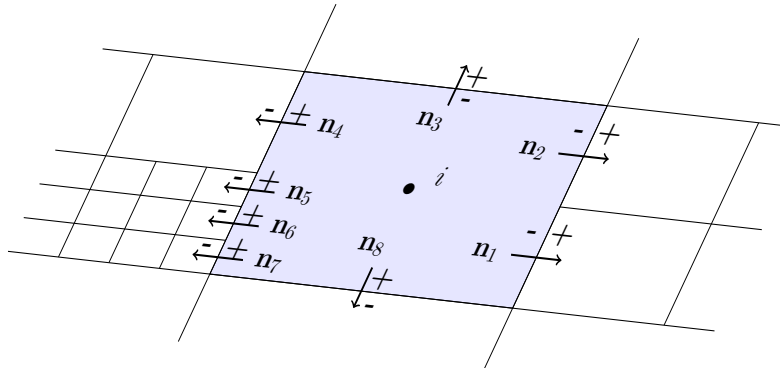


Figure 6.1: The main control volume and descriptions for each cell.

where  $\tau$  is boundary of  $\Omega_i$ ,  $\vec{n}$  is unit outward normal vector to the boundary, and  $\vec{F} = (f, g)$  is the vector that consists of the fluxes in the equations (6.2) at each interface of the cell boundary.

Dividing the equation (6.3) with the cell area, and calling  $H_i$  and  $R_i$  as the average of the free surface elevation and rainfall rate interior of the cell, respectively, the finite volume formulation is generated as

$$\frac{\partial H_i}{\partial t} = \frac{1}{\Delta x_i \Delta y_i} \oint_{\tau} \vec{F} \cdot \vec{n} d\tau + R_i, \quad (6.4)$$

where  $\Delta x_i$  and  $\Delta y_i$  are widths of the cell wall in the directions of the coordinate  $x$  and  $y$  axis, respectively. The subscript  $i$  is the spatial index of the cell.

Since our model is performed on dynamically adaptive grid considered as non-uniform grid (by non-uniform grid, each cell can possibly have many adjacent cells and with different size for each wall), the line integral in equation (6.4) can be discretized by substituting with the sum over sub interfaces around the control volume of  $\vec{F} \cdot \vec{n}$  as

$$\frac{\partial H_i}{\partial t} = \frac{1}{\Delta x_i \Delta y_i} \sum_k \vec{F}_k \cdot \vec{n}_k \Delta \tau_k + R_i, \quad (6.5)$$

where the subscript  $k$  is index for each of sub interfaces over boundary of  $\Omega$  between the cell and its neighbor, and  $\Delta \tau_k$  is the width of the sub interface.

By application of the Euler's method to the equation (6.5), the finite volume formulation over each rectangular cell of the grid is given by

$$H_i^{t+\Delta t} = H_i^t + \frac{\Delta t}{\Delta x_i \Delta y_i} \sum_k \vec{F}_k \cdot \vec{n}_k \Delta \tau_k + \Delta t R_i^t, \quad (6.6)$$

where  $\Delta t$  is time step size.

## 6.2.2 Numerical flux

In the developed algorithm, an approximation of fluxes between adjacent cells through each sub interface for simulation of the rainfall flow in rugged terrain is

given by

$$f_k = \alpha \frac{\partial H}{\partial x} \approx \alpha \frac{\hat{H}_+ - \hat{H}_-}{\Delta x}, \quad g_k = \alpha \frac{\partial H}{\partial y} \approx \alpha \frac{\hat{H}_+ - \hat{H}_-}{\Delta y} \quad (6.7)$$

with the Riemann states of water depth

$$\hat{H}_\pm = \max \left\{ 0, H_\pm^t - \max\{z_+, z_-\} \right\}. \quad (6.8)$$

This was also presented by Audusse et al [3] in the context of hydrostatic reconstruction. The marker  $(-)$  and  $(+)$  are indices for the left and right cells at each sub interfaces in the  $x$  direction while the  $y$  direction, the signs  $(-)$  and  $(+)$  indicate the bottom and top cells at each sub interfaces.

### 6.2.3 Stability condition

In order to have stable scheme for diffusion model as proposed in [33], the time step size is limited by the condition

$$\Delta t \leq 0.25 \frac{(\Delta A_{\min})^2}{\alpha}, \quad (6.9)$$

where  $\Delta A_{\min} = \min_{\forall i} \{\Delta x_i, \Delta y_i\}$  is a minimum dimension of all cells in the computational domain.

### 6.2.4 Boundary condition

The opened and closed boundary conditions used in the algorithm are defined as follows. The opened boundary condition can be given by

$$h_m = 0, z_m = 0, \quad (6.10)$$

while the closed boundary condition are given by

$$h_m = 0, z_m = z_{large}. \quad (6.11)$$

In these equations, the subscript  $m$  refers to the spatial index of the computational domain boundary cells, and  $z_{large}$  is some large constant value for the topography height, for example, we assign  $z_{large} = 9999$ .

### 6.2.5 Grid adaptivity method

The grid adaptivity method consists of designing of the tree grid, and the grid adaptivity conditions, together with the new values defining of the adapted cells variables. For the tree grid, we designed in the numerical scheme based on the general rectangular tree grid as described in section 3.1.

For the grid adaptivity conditions, we have used a simple criterion based on the two conditions, the gradient of the free surface  $H = h + z$  and the water depth. The checking of refinement and coarsening are performed as described in section 3.2.8. However, when a cell is refined, the children cells are created in next level that is not maximum level, and they are defined by the same values as their parent cell, i.e.

$$h_j^{l+1} = h_i^l, \quad (6.12)$$

where  $h_j^{l+1}$  is the water depth of the child of level  $l + 1$  cell at position  $j$ , and  $h_i^l$  is the water depth of the parent cell at  $i$ .

In addition, the values of height of topography,  $z$ , are defined by the digital terrain data for each levels as described in the next section.

For the coarsening, the values of the merged parent cell is defined by the averages as:

$$h_i^l = \frac{1}{n_i^l \times n_i^l} \sum_{j \in \Omega_i} h_j^{l+1}, \quad (6.13)$$

where  $n_i^l \times n_i^l$  is number of the children of the parent cell.

### 6.2.6 Topography interpolation

In real application of the simulation and visualization of rainfall overland flows, the values of heights of topography are obtained by digital terrain data grids. Since the computational grid resolution is possibly different (finer or coarser) than the digital terrain data grid resolution, topography interpolation is required. We present here a method for the topography interpolation using triangular planes approximation.

For the approximation of the value of the cell in the domain at  $i$ :

$$z_i = \begin{cases} z_{x',y'} + (z_{x'+1,y'} - z_{x',y'})(m - x') \\ \quad + (z_{x',y'+1} - z_{x',y'})(n - y') & , m \leq x' - n + y' + 1; \\ z_{x'+1,y'+1} + (z_{x'+1,y'} - z_{x'+1,y'+1})(m - (x' + 1)) \\ \quad + (z_{x'+1,y'} - z_{x'+1,y'+1})(n - (y' + 1)) & , m > x' - n + y' + 1, \end{cases} \quad (6.14)$$

where  $(x', y')$  is the position of the topographic data grid cell used for approximation with  $x' = \lfloor m \rfloor$  and  $y' = \lfloor n \rfloor$ ,

$$m = x_i \times n^{x'} / n^x, \quad n = y_i \times n^{y'} / n^y \quad (6.15)$$

are the mapped indices of the computational grid cell position  $(x_i, y_i)$  to the topographic data grid,  $n^{x'}$  and  $n^{y'}$  are numbers of columns and rows of the topographic data grid, and  $n^x$  and  $n^y$  are numbers of columns and rows of the finest computational grids.

### 6.2.7 Dynamic domain defining method

As described in section 3.2.10, in order to improve the computational efficiency, the dynamic DDM is applied in this modeling and simulation. The cells are checked for identify whether the cell is inside the computational domain or not. The checking is based on three conditions of indicators as:

1. the cell is wet, *i.e.*  $h_i^t > 0$ ;
2. the cell has one or more of neighbor cells that are wet, *i.e.*  $h_k^t > 0$  for some  $h_k^t \in N_i$ , the set of neighbors of the  $i$  cell;
3. the cell has rainfall, *i.e.*  $R_i^t > 0$ .

If one of these conditions is true, the cell and its neighbor cells are included to the domain whenever these cells are not in the domain. On the other hand, the cell is excluded from the domain. Only grid cells within the computational domain are



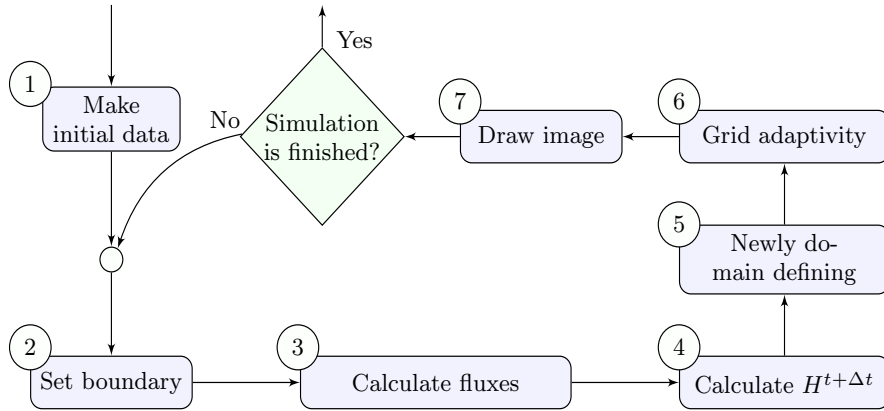


Figure 6.2: The algorithm overview.

adapted and computed in order to minimize the total number of the computational cells.

In this algorithm, the dynamic DDM is used for defining the newly computation domain in each iteration steps after the solution of the equation (6.6).

### 6.2.8 Algorithm overview

We present here an overview of the algorithm for the computational scheme presented above. the developed algorithm consists of several steps describing the calculations for each grid cell, as illustrated in Figure 6.2. The details for each steps can be described as follows:

- Step ① : Set the initial values, the cell sizes, the number of children in each level, and the maximum of tree grid levels.
- Step ② : Set the computational grids boundaries.
- Step ③ : Calculate the numerical fluxes and  $\Delta t$ .
- Step ④ : Calculate the solutions of  $H^{t+\Delta t}$  as equation (6.6).
- Step ⑤ : Define a new domain based on the dynamic DDM.
- Step ⑥ : Conduct the grids adaptivity technique, see §6.2.5.

### I Refinement procedure

- i For each cells whose level is not maximum level
  - Mark cells that satisfy the refinement conditions.
- ii For each marked cells
  - Create new children cells, and assign the dependent values as equation (6.12).

### II Coarsening procedure

- i For each parent cells whose children have no child
  - Average the children values defined in (6.13);
  - Check the average with the coarsening conditions;
  - If either of the conditions is false, then merge all children cells into a parent cell, and assign the dependent values.

- Step ⑦ : Visualize the solutions in 2D and 3D by OpenGL.
- Repeat Step ② to ⑦ until the simulation is finished.

## 6.3 Results

In this chapter, the developed diffusion model is employed to simulate the rainfall flow on natural topography. The simulations consist of two tests as follows: the rainfall overland flow simulation compared with the simulations using the shallow water equation, and the simulation of rainfall overland flow compared the performance with the simulation on finest grid. The first numerical simulation is for accuracy and efficiency tests of the developed diffusion model, while the second simulation is for accuracy and efficiency tests of the adaptive tree grid technique. The coding and simulations are performed using 2 GHz Intel Core i7 with RAM 6 GB.

### 6.3.1 Accuracy and efficiency of the diffusion model

We illustrated in this section the accuracy and efficiency tests of the diffusion model, which are tested by comparing with the simulation using the shallow water equations. The shallow water equation is simulated using the second order finite volume scheme in [3] and [9] with the Manning's coefficient 0.001. The simulation of the diffusion model is performed on the domain of digital terrain data 7200 m  $\times$  7200 m, generated from Shuttle Radar Topography Mission (SRTM) data source with resolution 45 m  $\times$  45 m. Therefore, the initial grid has 160  $\times$  160 cells. The numerical experiment is simulated for 10000 s with the water diffusion coefficient is 500 m/s. The rainfall rate is set at 0.001 m/s within the rectangular regions where 2700 m  $\leq x \leq 4500$  m and 2700 m  $\leq y \leq 4500$  m for the duration of 3600 s. Moreover, the domain boundaries are defined as closed boundary condition.

The 3D representations of the rainfall overland flow simulation for this experiment are presented at different times, see Figure 6.3. The small pictures in Figure 6.3 show the corresponding 2D images. The results show that the diffusion model with the developed numerical scheme has a potential for the simulation of the water flow in rugged terrain.

To check the accuracy and efficiency of the diffusion model with the developed numerical scheme, we found errors of the numerical flooded area of the diffusion model compared with the numerical flooded area using the shallow water equations.

A Fit-Statistic method,  $F$ , introduced by [23], is used to estimate the ratio between the numerical flooded area and the observed inundated area. Here,  $F$  is used to estimate the percentage of the coincided flood extent between the current prediction and the benchmark, which can be described by,

$$F = \frac{W_c \cap W_b}{W_c \cup W_b}, \quad (6.16)$$

where  $W_c$  and  $W_b$  stand for the number of inundated cells given by the present simulation and the benchmark solution, respectively.

$F$  can be specifically used to calculate an error percentage of the model results

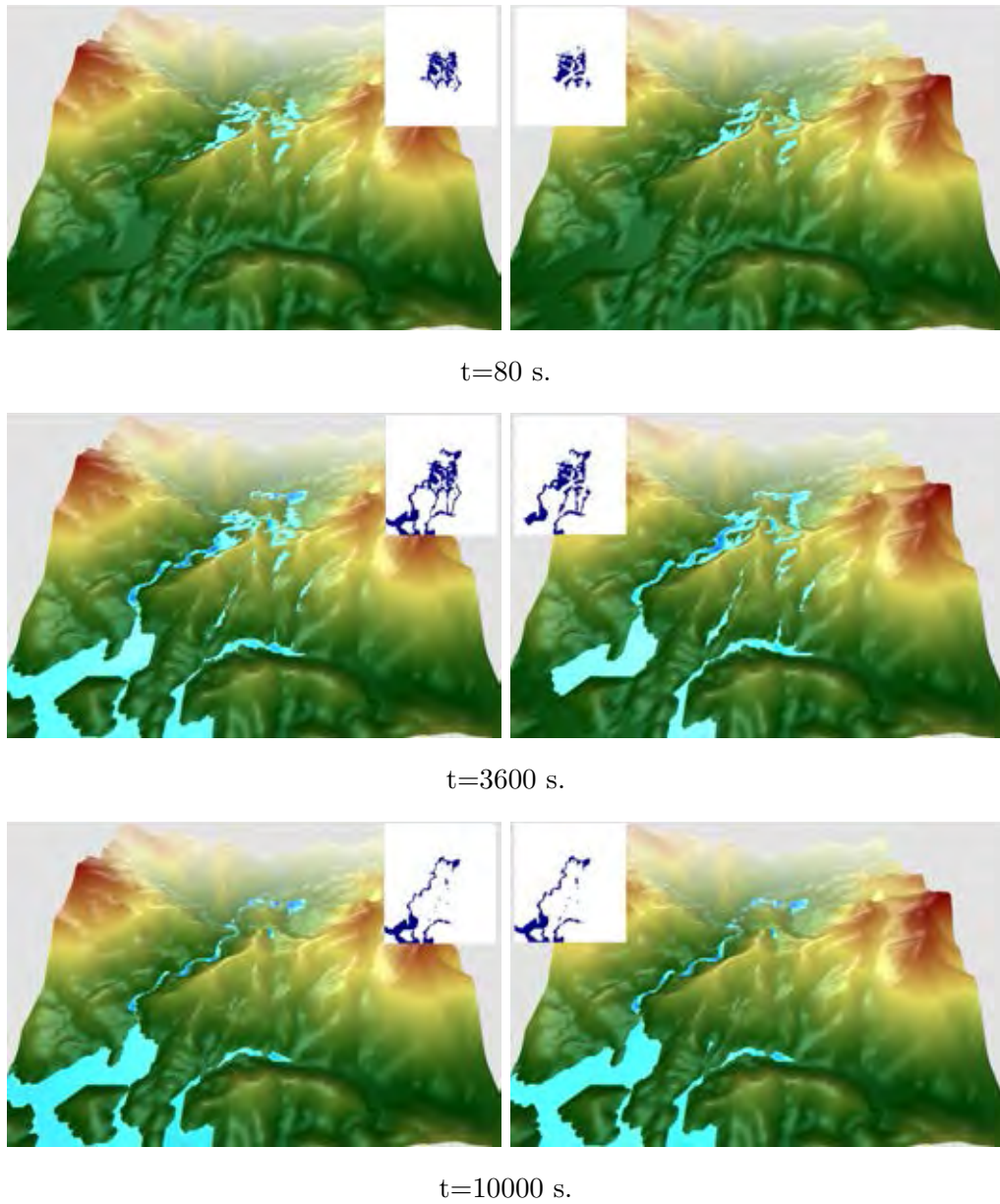


Figure 6.3: The 3D representation of the rainfall overland flow simulation using the shallow water equations (left) and the diffusion model (right).

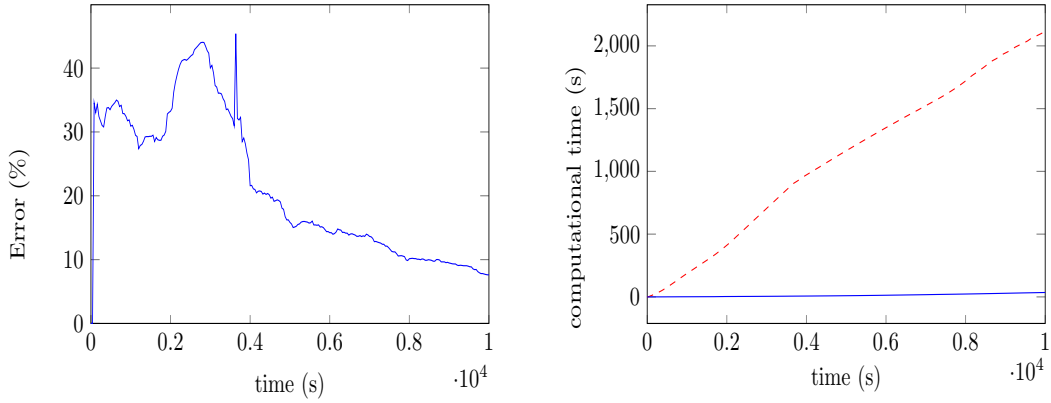


Figure 6.4: The left picture is the errors of the diffusion model compared with the shallow water equations, and the right picture is the comparison of the computational times for the simulation of the diffusion model (solid) and the shallow water equations (dashed).

using the following formula,

$$E = \left( 1 - \frac{\sum_i P_i^{B_1 C_1}}{\sum_i P_i^{B_1 C_1} + \sum_i P_i^{B_1 C_0} + \sum_i P_i^{B_0 C_1}} \right) \times 100, \quad (6.17)$$

where  $P_i^{B_1 C_1}$  is 1 when the benchmark cell is wet and the present compared cell is wet; otherwise its value is 0.  $P_i^{B_1 C_0}$  is 1 when the benchmark cell is wet and the present compared cell is dry; otherwise it is 0.  $P_i^{B_0 C_1}$  is 1 when the benchmark cell is dry and the compared cell is wet; otherwise its value is 0. In equation (6.17), the numerator accounts for the coincided inundated area predicted by both sets of solutions. The denominator gives the sum of the flooding area defined by either the compared simulation or the benchmark. Therefore,  $E$  is a variable between 0 and 100. 100 represents the wet-dry state in the compared solution is totally different from the benchmark lattices while 0 represents that the predicted flooding extent is absolutely the same as that in the benchmark lattices.

Here, the computed results of the shallow water flow model are taken as the benchmark. As shown in Figure 6.4 (left),  $E$  decrease from  $t = 4000$  s. At  $t = 9000$  s,  $E$  lower 10%, and the results predicted by the diffusion model presents the same matched flood extent state as the simulation of the shallow water flow model. It

also proves that the accuracy of the diffusion model converges to the shallow water flow model.

The efficiency of the diffusion model is presented by comparing the computational times for the simulations as shown in Figure 6.4 (right). The results show that the diffusion model is more efficient because it helps reducing the computational run times (from 2054.97397 s by the shallow water equation to 15.0300 s by the diffusion model) while the accuracy of the diffusion model converges to the shallow water model.

### 6.3.2 Accuracy and efficiency of the adaptive scheme

In this section, the accuracy and efficiency of the dynamically adaptive scheme are tested by comparing with the simulations on the uniform grids. The simulation is performed on the domain of digital terrain data 36000 m  $\times$  36000 m of Banglang place which is in Yala, southern Thailand. The digital terrain data is generated from Shuttle Radar Topography Mission (SRTM) data source. The maximum tree grid level is 4 with resolutions 360 m  $\times$  360 m, 180 m  $\times$  180 m, 90 m  $\times$  90 m, and 45 m  $\times$  45 m, for levels 1, 2, 3, and 4, respectively; therefore, the initial grid level 1 has 100  $\times$  100 cells. The topography values are provided for levels 1, 2 and 3, except the level 4 which are obtained automatically via the triangular planes approximation. The tree grid cells for children are designed as 2  $\times$  2 for levels 1, 2, and 3, while level 4 cells have no children.

The numerical experiment is simulated for 53 hours with the water diffusion coefficient is 50.625 m/s, and the parameters of the grid adaptivity method are  $\epsilon_1 = \epsilon_2 = 0.1$  and  $\xi = 0.001$ . The rainfall rate is set at 0.001 m/s within the two rectangular regions which are the first region: 13500 m  $\leq x \leq 18000$  m and 21150 m  $\leq y \leq 25650$  m, and the second region: 18900 m  $\leq x \leq 23400$  m and 19800 m  $\leq y \leq 24300$  m for the duration of 7200 s. Moreover, the domain boundaries are defined as the opened boundary condition.

The 3D representation of the rainfall overland flow simulation for this experiment are presented at different times in Figure 6.5, and the corresponding au-

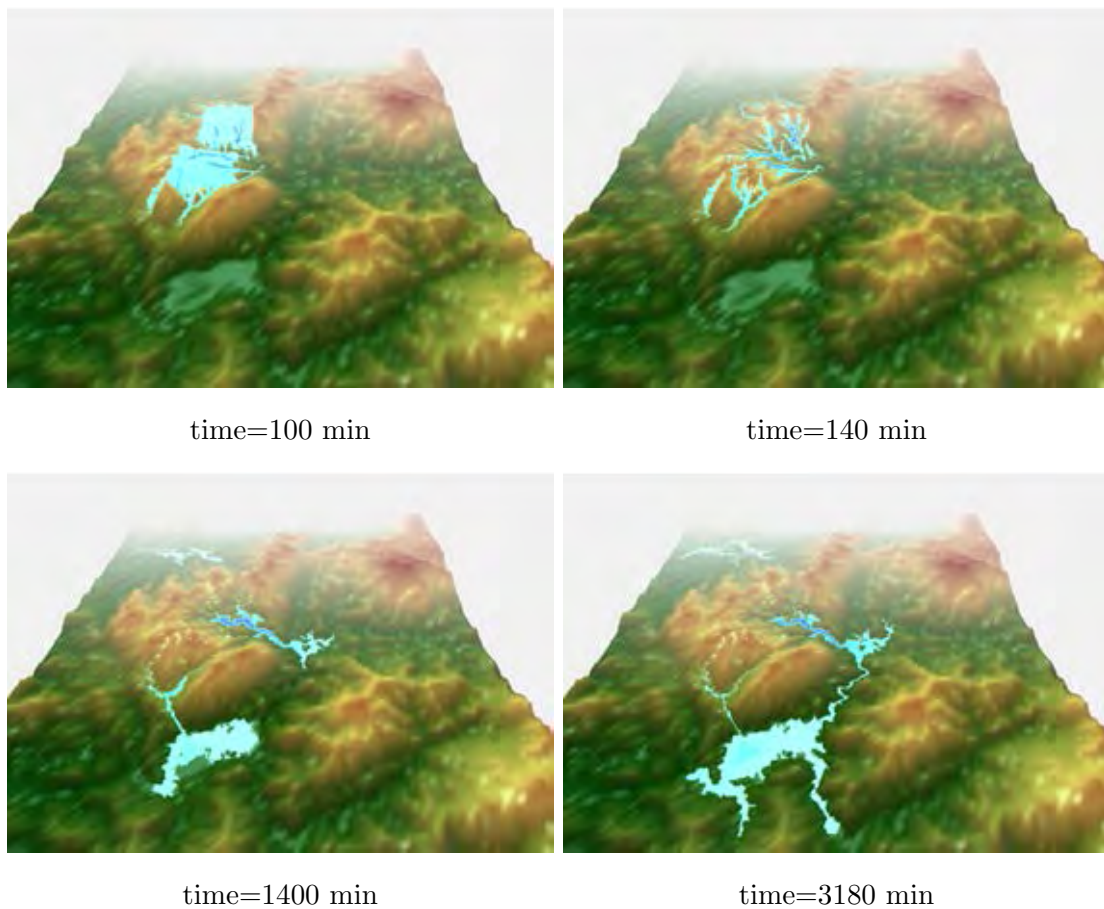


Figure 6.5: The 3D representation of the adapted grids of the rain water overland flow simulation for each time.

tomatically adapted grids are shown in Figure 6.6. The results show that the high-resolution areas are automatically obtained by the adaptivity technique in corresponding to the regions where the free surface gradients are steep and have water depths more than 0.001 m, based on adaptivity parameters. The results also indicate that the diffusion model with the developed numerical scheme has a potential for the simulation of the water flow in rugged terrain.

To check the accuracy and efficiency of the dynamically adaptive scheme we performed the following test. We compared the dynamically adaptive simulation with non-adaptive simulations using only the dynamic DDM on two uniform grids, level 1 grid (largest cells with 100 x 100) and level 4 grid (smallest cells with 800



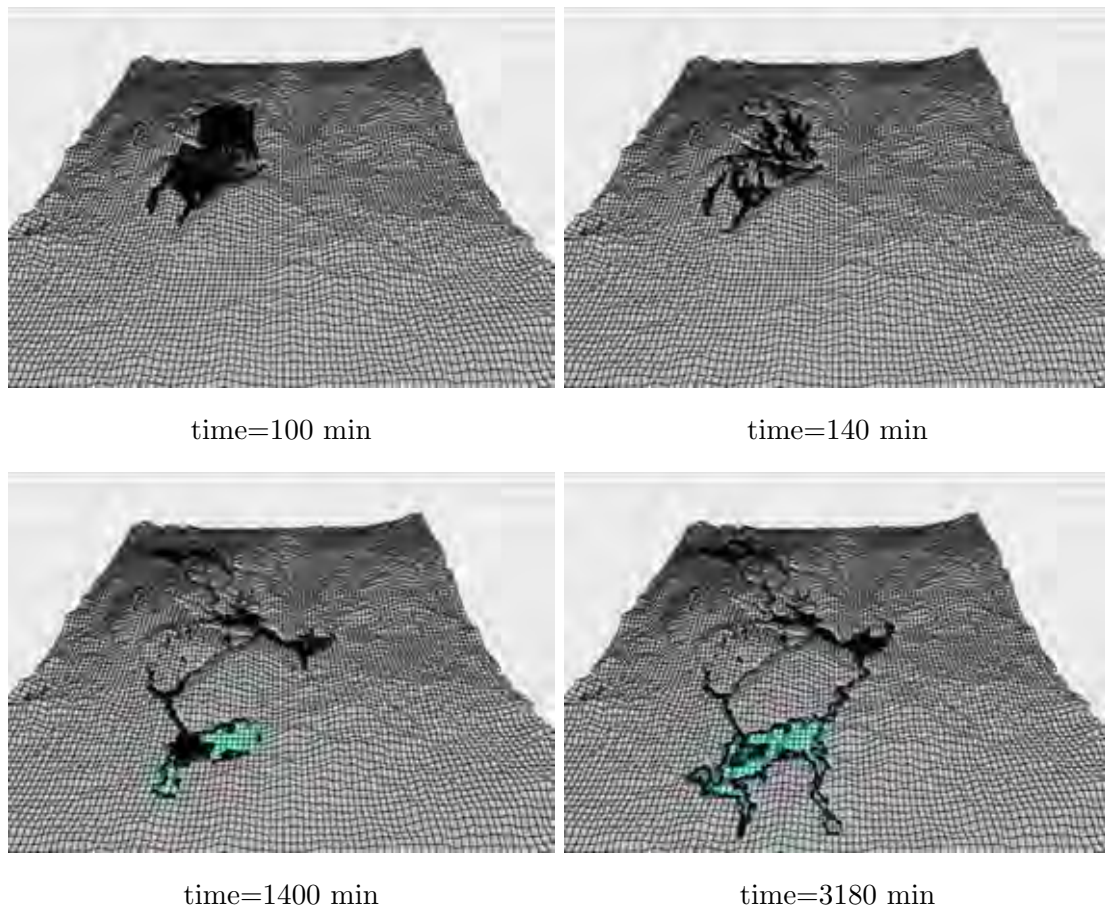


Figure 6.6: The 3D representation of the adapted grids of the rain water overland flow simulation for each time.

x 800 cells).

The 3D representations of the rainfall overland flow simulations at 53 hours for this test are shown in Figure 6.7. The results show that the simulation and the flow profiles obtained by the dynamically adaptive scheme are nearly the same as that obtained from the finest grid (level 4 grid), while the result from the coarsen grid (level 1 grid) is not acceptable. This shows the accuracy of the adaptive scheme. The efficiency of the adaptive scheme is presented by comparing the number of computational cells and the computational times for the simulations as shown in Figure 6.8. The results show that the adaptive scheme is more efficient because it helps reducing the computational cells (from 69,708 cells by the finest grid to less



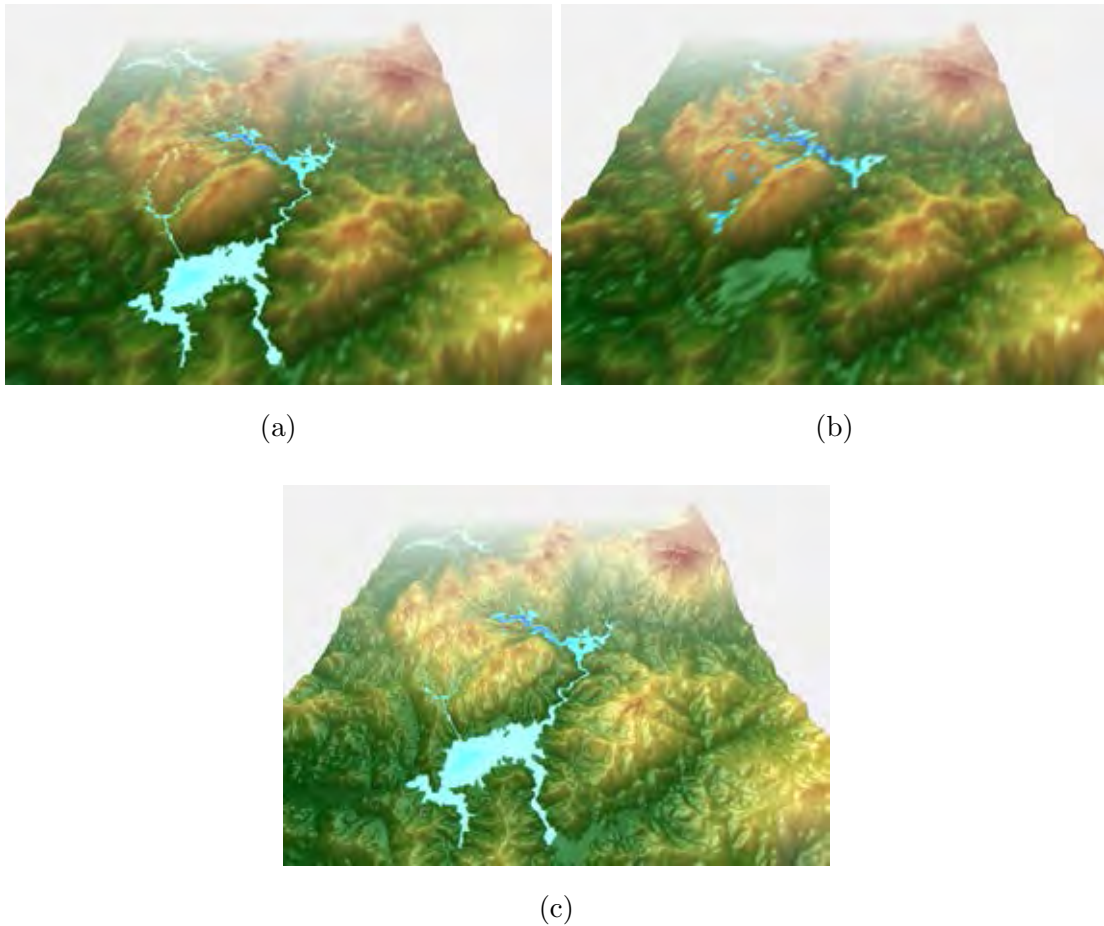


Figure 6.7: The 3D representation of the comparison for adaptive grids (a), uniform coarsen grids (b), and uniform finest grids (c) of the rain water overland flow simulation at time=53 h.

than 35,000 cells by the adaptive grid), and therefore, reduces the computational run times (from 1247.8610 s by the finest grid to 703.4000 s by the adaptive grid) while keeping the same accuracy as that of the finest grid.

## 6.4 Conclusion

The simulations of rainfall overland flow in general have been developed based on shallow water equations. In this chapter, the algorithm has been constructed based on the diffusion model to simulate the inundation of rainfall in rugged terrain

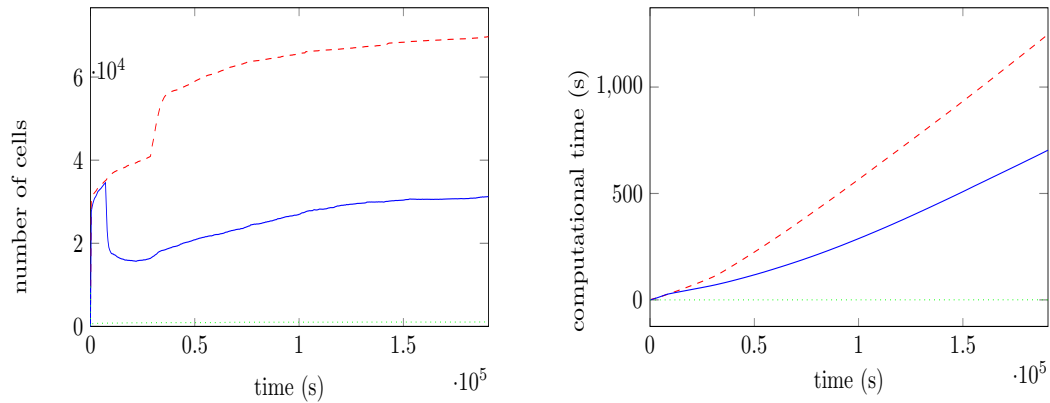


Figure 6.8: The comparisons of the number of computational cells (left) and the computational times (right) of the rain water overland flow simulation for the adaptive grid (solid), the uniform finest grid (dashed), and the uniform coarsen grid (dotted).

using the adaptive finite volume scheme. The algorithm applied the dynamically adaptive rectangular tree grid technique and together with the dynamic DDM. The results predicted by the diffusion model presents the accuracy converges to the shallow water flow model. The comparisons of the developed simulation results with the uniform small and large cell size grids results reveal the advantages that it can reduce the number of the computational cells and the simulation run times without losing much accuracy in the results. This is an enormous efficiency advantage in terms of computational costs which can be very helpful for the flood risk assessments and water managements in the real-world applications.

## CHAPTER VII

### CONCLUSION

This dissertation has presented algorithms and software to simulate and visualize flood models on natural topography based on finite volume methods for shallow water equations by modifying the adaptive grid technique, developed in [30, 34, 2] for rectangular quad tree grids, and combining with the dynamic DDM. Here, the idea of quad tree grids is extended for general tree grids to handle natural topography data in general. The algorithm applied the second order finite volume scheme of Audusse, and was validated for idealist flooding inundation cases on natural topography. The simplification of the algorithm using the first order finite volume scheme of Audusse was validated for a dam-break flooding inundation case. A simple model based on a diffusion equation, has been solved by using the adaptive general tree grids technique, incorporated with the finite volume scheme and non-negative water depth reconstructed method, and validated for an idealist rainfall overland flow case. In this chapter, the conclusions of this research are presented as the following details.

#### **7.1 Dynamically Adaptive Tree Grid Algorithm for Shallow Water Model**

The flood model based on 2D shallow water equations has been solved by using a finite volume scheme. The HLL Riemann solver is adopted to solve the interface fluxes, incorporated with the gravity force calculation. The second order scheme is developed by using the TVD Runge-Kutta for time integration and the slope limiter in space. A non-negative water depth reconstruction (or hydrostatic reconstruction) approach is implemented to deal with the wetting and drying interfaces. A semi implicit scheme is implemented to calculate the friction source term to avoid

spurious oscillation. For the explicit numerical scheme, the Courant-Friedrichs-Lewy (CFL) criterion is adopted to limit the time step in order to maintain the computational stability. Two types of boundary conditions, the opened and closed boundaries are used for defining the domain boundary values.

These finite volume schemes are applied for the dynamically adaptive tree grid algorithm, developed in this work for general rectangular tree grids, and combined with the dynamic DDM. The algorithm is tested for the accuracy for the numerical simulations. The numerical simulations show that the developed algorithm is suitable for simulating and visualizing the flow of water on natural topography, which agree closely with other results obtained in literature, and for both experimental and theoretical results. Moreover, the algorithm is validated for simulation and visualization of idealist floods in Thailand, such as the rain-water overland flow simulation, the simulation of a dam-break on natural topography, and the breaching flood simulation. The adaptivity and dynamic DDM helps making the computation efficient while keeping the same accuracy as that of the computation on the finest grids. Therefore, it has a potential for practical usages, and can be applied to other flooding simulations on natural topography such as river flood, including utilization for the water management system.

## **7.2 Simplification of The Dynamically Adaptive Tree Grid Algorithm**

The dynamically adaptive grid algorithm was simplified using the first order Audusse's finite volume scheme for the shallow water equations solved by the dynamically adaptive general rectangular tree grid method. By the simplification, the slope limiter in space (or reconstruction procedure in §3.2.3) is not used, and therefore, TVD Runge-Kutta iteration, to prevent spurious oscillations is unnecessary, and the Euler's method is used for time integration, two procedures which are the slope limiter in space and the TVD Runge-Kutta iteration are excluded from the algorithm. The simplification reduces the complexity of the dynamically adaptive general rectangular tree grid algorithm. The simulation results are compared with

the real experimental data and with simulations from the second order scheme on the finest grid. The comparisons show that the developed scheme is very efficient in terms of computational cost by reducing the number of the computational cells and the computational time without losing much accuracy in the results. Therefore, it has a potential for practical usages, the scheme can be applied to simulate the other situations on flood problems.

### **7.3 Diffusion Modeling on Dynamically Adaptive Tree Grid**

In this work, the algorithm has been constructed based on the diffusion model to simulate the inundation of rainfall in rugged terrain using the adaptive finite volume scheme. The algorithm applied the dynamically adaptive rectangular tree grid technique and together with the dynamic DDM. Herein, the diffusion equation is simply solved by an explicit finite volume method. The interface flux is approximated by the difference equation of the fluxes. A non-negative reconstruction method (in terms of water depth) is implemented to capture the wet-dry front and avoid the negative water depth. The results predicted by the diffusion model presents the accuracy converges to the shallow water flow model. The comparisons of the developed simulation results with the uniform small and large cell size grids results reveal the advantages that it can reduce the number of the computational cells and the simulation run times without losing much accuracy in the results. This is an enormous efficiency advantage in terms of computational costs which can be very helpful for the flood risk assessments and water managements in the real-world applications.

### **7.4 Future Works and Recommendation**

In this research, the algorithms for simulation and visualization of floods are developed for rain-water overland flows. The models are validated, where the rainfall rate and manning's coefficient are assumed for idealist events. In real world applications, data from weather radar, land cover maps, soil maps and others can

be used for defining the net rainfall rate and Manning's coefficient, and simulating other processes of rain-water overland flows, such as infiltration, sediment transport, erosion, etc. For example, infiltration can be considered, where Green-Ampt equations [18] will be used in our model. Moreover, since real flood events can wash bridges, coastal defences and dams causing landslide, and also sweep the sediment along river bed and floodplain, the flood model should be combine with the sediment transport and erosion formulas. The improvement of the adaptive grid technique can be modified by using the idea of adaptive local time step method described in [16], which also has advantages in terms of computational costs.

## REFERENCES

- [1] Alsdorf D, Dunne T, Melack J, Smith L, Hess L. Diffusion modeling of recessional flow on central Amazonian floodplains. *Geophysical Research Letters*. 32 (2005).
- [2] An H, Yu S. Well-balanced shallow water flow simulation on quadtree cut cell grids. *Advances in Water Resources*. 39 (2012) : 60-70.
- [3] Audusse E, Bouchut F, Bristeau MO, Klein R, Perthame B. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows, *SIAM J.*25(6) (2004) : 2050-2065.
- [4] Audusse E, Bristeau MO. A well-balanced positivity preserving second-order scheme for shallow water flows on unstructured meshes. *Journal of Computational Physics*.206 (2005) : 311-333.
- [5] Berger MJ, Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of computational Physics*. 53(3) (1984) : 484-512.
- [6] Bermudez A, Vazquez ME. Upwind methods for hyperbolic conservation laws with source terms, *Computers & Fluids*. 23(8) (1994) : 1049-1071.
- [7] Busaman A. *Mathematical model for simulation and visualization of flood*. Master's Thesis : Applied Mathematics and Computational Science, Chulalongkorn University, 2010.
- [8] Chuai-Aree S, Kanbua W. SiTProS Fast and Real-Time Simulation of Tsunami Propagation. *Asia Modeling Symposium international conference*. (2007).
- [9] Delestre O, Cordier S, James F, Darboux F. Simulation of rain-water overland-flow, *In Proceedings of the 12th International Conference on Hyperbolic Problems, University of Maryland, College Park (USA)*. (2008).
- [10] Delestre O. Simulation du ruissellement d'eau de pluie sur des surfaces agricoles/ rain water overland flow on agricultural fields simulation, *PhD thesis, Université d'Orléans (in French)*, available from TEL: tel.archives-ouvertes.fr/INSMI/tel-00531377/fr. (2010).
- [11] Delestre O, Cordier S, Darboux F, James F. A limitation of the hydrostatic reconstruction technique for shallow water equations. *Comptes Rendus de l'Académie des Sciences*. 350 (2012) : 677-681.
- [12] Dottori F, Todini E. Developments of a flood inundation model based on the cellular automata approach: Testing different method to improve model performance. *Physics and Chemistry of the Earth*. 36 (2011) : 266-280.

- [13] De Saint Venant AJC. Théorie du mouvement non-permanent des eaux, avec application aux crues des rivières et à l'introduction des marées dans leur lit. *Comptes Rendus de l'Académie des Sciences*. 73 (1871) : 147-154.
- [14] Fiedler FR, Ramirez JA. A numerical method for simulating discontinuous shallow flow over an infiltrating surface. *International Journal for Numerical Methods in Fluids*. 32 (2000) : 219-240.
- [15] George DL. Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: application to the Malpasset dam-break flood (France, 1959). *International Journal for Numerical Methods in Fluids*. (2010).
- [16] George DL. Finite volume methods and adaptive refinement for tsunami propagation and inundation, *Doctor's Dissertation : Applied Mathematics, Washington University*. (2006).
- [17] Goutal N, Maurel F. A finite volume solver for 1D shallow-water equations applied to an actual river. *International Journal for Numerical Methods in Fluids*. 38 (2002) : 1-19.
- [18] Green WH, Ampt GA, Studies on soil physics, part I-the flow of air and water through soils. *J. Agric. Sci.* 4 (1911) : 1-24.
- [19] Greenberg JM, LeRoux AY. A well-balanced scheme for the numerical processing of source terms in hyperbolic equation. *SIAM Journal on Numerical Analysis*. 33 (1996) : 1-16.
- [20] Harten A, Lax PD, Leer BV. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review*. 25(1) (1983) : 35-61.
- [21] Hagen TR, Hjelmervik JM, Lie KA, Natvig JR, Henriksen MO. Visual simulation of shallow water waves. *Simulation Modelling Practice and Theory: Special Issue on Programmable Graphics Hardware*. 13 (2005) : 716-726.
- [22] Hieu PD. Finite volume method for long wave run-up: 1D model. *VNU J.* 24 (2008) : 10-15.
- [23] Horritt MS, Bates PD. Effects of spatial resolution on a raster based model of flood flow. *Journal of Hydrology*. 253(1-4) (2001) : 239-249.
- [24] Hou J, Simons F, Liang Q, Hinkelmann R. An improved hydrostatic reconstruction method for shallow water model. *Journal of Hydraulic Research*. 52(3) (2014) : 432-439.



- [25] Iwagaki Y. Fundamental studies on the run-off analysis by characteristics. *Disaster Presentation Research Institute Bulletin, Kyoto University, Kyoto, Japan.* 10 (1955) : 1-25.
- [26] Kleefsman KMT, Fekken G, Veldman AEP, Iwanowski B, Buchner B. A volume-of-fluid based simulation method for wave impact problems. *Computational Physics J.* 206(1) (2005) : 363-393.
- [27] Kurganov A, Noelle S, Petrova G. Semi-discrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations. *SIAM J.* 23(3) (2001) : 707-740.
- [28] Kurganov A, Levy D. Central-upwind schemes for the Saint-Venant system. *ESAIM: M2AN.* 26(3) (2002) : 397-425.
- [29] Kurganov A, Petrova G. Central-upwind schemes for two-layer shallow water equations. *SIAM J.* 31(3) (2009) : 1742-1773.
- [30] Liang Q, Borthwick AGL, Stelling G. Simulation of dam- and dyke-break hydrodynamics on dynamically adaptive quad-tree grids. *International Journal for Numerical Methods in Fluids.* 46 (2004) : 127-162.
- [31] Liang Q, Du G, Hall JW, Borthwick AGL. Flood inundation modeling with an adaptive quadtree grid shallow water equation solver. *International Journal for Hydraulic Engineering.* 132(11) (2008) : 1603-1610.
- [32] Morales de Luna T, Castro Daz MJ, Pares C. Reliability of first order numerical schemes for solving shallow water system over abrupt topography. *Applied Mathematics and Computation.* 219 (2013) : 9012-9032.
- [33] Murthy JY. Numerical methods in heat, Mass and momentum transfer, *Lecture Notes for School of Mechanical Engineering, Purdue University.* (2002).
- [34] Popinet S. Quad-tree adaptive tsunami modelling. *Ocean Dynamics.* 61(9) (2011) : 1261-1285.
- [35] Richardson SR. *Numerical simulation of impulsive wave over-topping events resulting from landslides.* Doctor's Thesis : Computing and Mathematics, Manchester Metropolitan University, 2010.
- [36] Roshandel A, Hedayat N, Kiamanesh H. Simulation of dam break using finite volume method. *World Academy of Science, Engineering and Technology.* 4(11) (2010):103-106.
- [37] Sampson J, Easton A, Singh M. Modelling the effect of proposed channel deepening on the tides in Port Phillip Bay. *ANZIAM Journal.* 46 (2005) : C888-C901.

- [38] Santillana M. *Analysis and Numerical Simulation of the Diffusive Wave Approximation of the Shallow Water Equations*. Doctor's thesis, University of Texas at Austin, 2008.
- [39] Sattamasakul S. *Simulation of shallow water flow with source terms*. Master's Thesis : Computational Science, Chulalongkorn University, 2006.
- [40] Schreiber M, Bungartz HJ, Bader M. Shared memory parallelization of fully-adaptive simulations using a dynamic tree-split and join approach. *High Performance Computing (HiPC), 19th International Conference on IEEE*. (2012).
- [41] Suwannasri P. *Parallel computation of shallow water flow problems*. Master's Thesis : Computational Science, Chulalongkorn University, 2004.
- [42] Thacker WC. Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics Digital Archive*. 107(1) (1981) : 499-508.
- [43] Valiani A, Caleffi V, Zanni A. Finite volume scheme for 2D shallow water equations application to the Malpasset dam-break. *The 4th CADAM Workshop*. (1999).
- [44] Wang Y. *Numerical improvements for large-scale flood simulation*. Doctor's Dissertation : School of Civil Engineering and Geosciences, Faculty of Science, Agriculture and Engineering, Newcastle University, 2011.
- [45] Yamaguchi S, Ikeda T, Iwamura K, Naono K, Ninomiya A, Tanaka K, Takahashi H. Development of GIS-based flood-simulation software and application to flood-risk assessment. *2nd IMA International Conference on Flood Risk Assessment*. (2007).
- [46] Zhang W, Cundy TW. Modeling of two-dimensional overland flow. *Water Resources Research*. 25(9) (1989) : 2019-2035.
- [47] Zanotti O, Manca GM. A very short introduction to Godunov methods. *Lecture Notes for the COMPSTAR School on Computational Astrophysics, 8-13/02/10, Caen, France*. (2010).

## APPENDIX

## INTEGRATION OF THE FLOOD MODEL WITH GOOGLE EARTH SOFTWARE

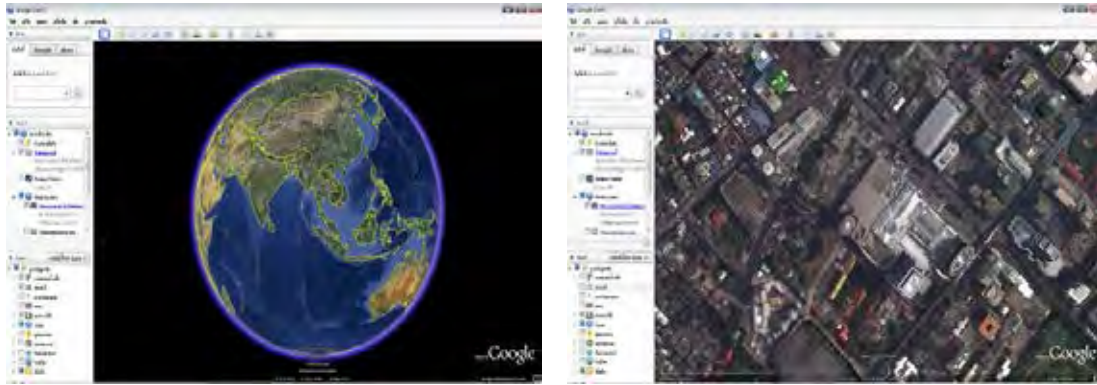


Figure 1: Snaps short of Google Earth software.

Google Earth software is a program for displaying satellite images and geographic informations such as topography height, position of locality, building, road etc. Therefore, the visualization of the flood model obtained via the Google Earth software is then usefulness for flood warning and water managements.

The visualization of the flood model equipped with the Google Earth software can be performed by generating series of command strings for a kml file to link each frame of the result images. Figure 1 shows a example of the command string for one frame, consist of times of the simulated event, a path of the result image and a position of the simulated locality. The command string of each frames are combined and arranged in order of the simulated times for generating the kml file. The kml file can be used to visualize an animation of flood via Google Earth software.

Figures 3, 4 and 5 show the visualization of the flood model results via Google Earth software in these works. Figure 3 shows the visualization of rain-water overland flow at Krungching place, Nakornsritamarat, Southern Thailand, while Figure 4 shows the visualization of Mae-suai dam break flood simulation at Chian-grai, Northern Thailand, and Figures 5 and 6 show the visualization of Breaching flood simulation at Bangkok, Thailand.

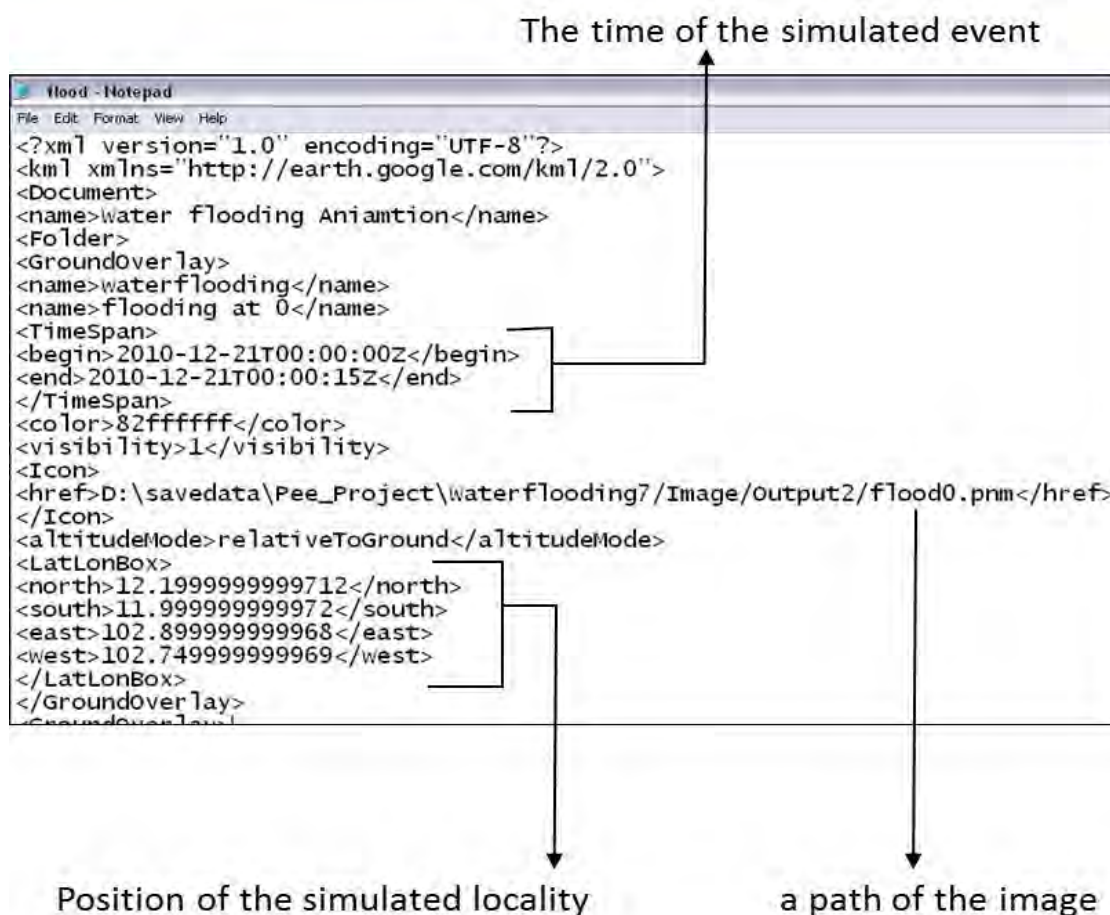


Figure 2: An example of the command strings for one frame, consist of times of the simulated event, a path of the result image, and a position of the simulated locality.

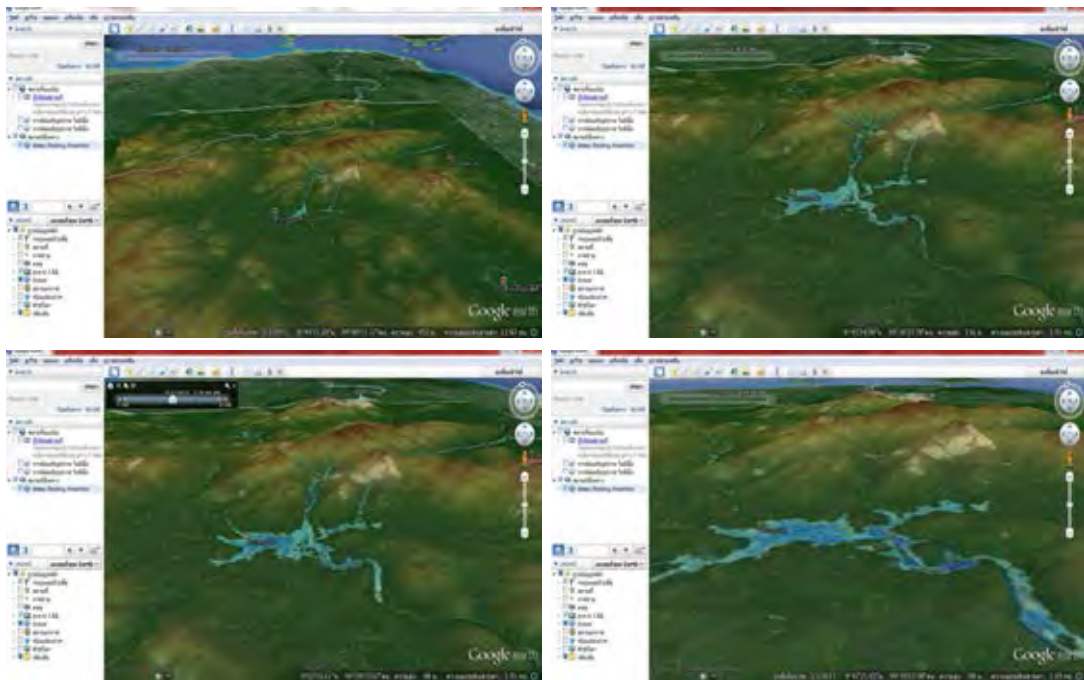


Figure 3: The visualizations via Google Earth software of rain-water overland flow at Krungching place, Nakornsritamarat, Southern Thailand, at different times  $t$ .

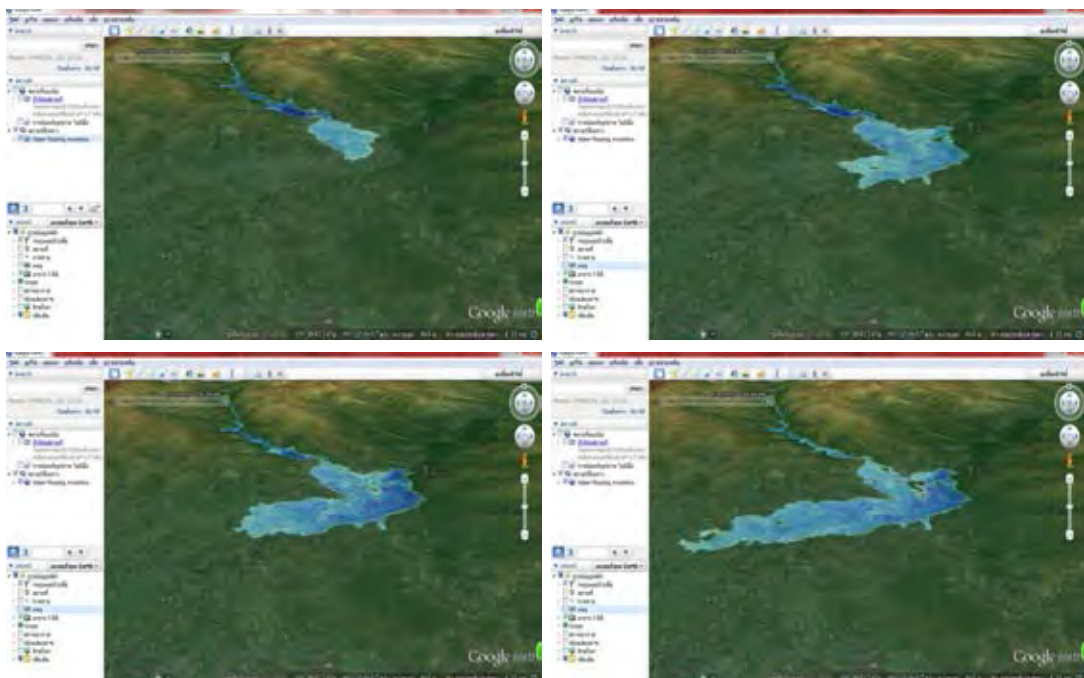


Figure 4: The visualizations via Google Earth software of Mae-suai dam break flood simulation at Chiangrai, Northern Thailand, at different times  $t$ .





Figure 5: The visualizations via Google Earth software of Breaching flood simulation at Bangkok, Thailand, at different times  $t$ .

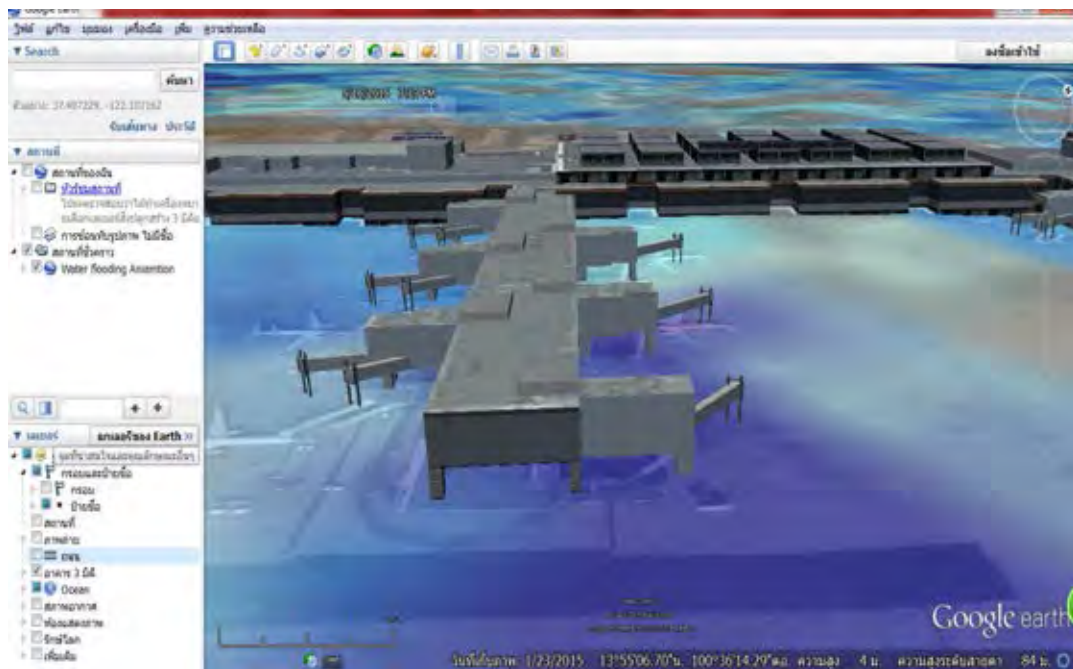


Figure 6: The zoom representation via Google Earth software of the breaching flood simulation that shows building at Donmuang air port.

## VITA

**Name** Mr. Anurak Busamun

**Date of Birth** 26 July 1986

**Place of Birth** Pattani, Thailand

**Education** B.Sc. (Applied Mathematics, Second Class Honors),  
Prince of Songkla University, Pattani Campus, 2009  
M.Sc. (Applied Mathematics and Computational Science),  
Chulalongkorn University, 2011