

การคำนวณเชิงขนานของปัญหาการไหลในน้ำตื้น



นายไพรินทร์ สุวรรณศรี

สถาบันวิทยบริการ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์

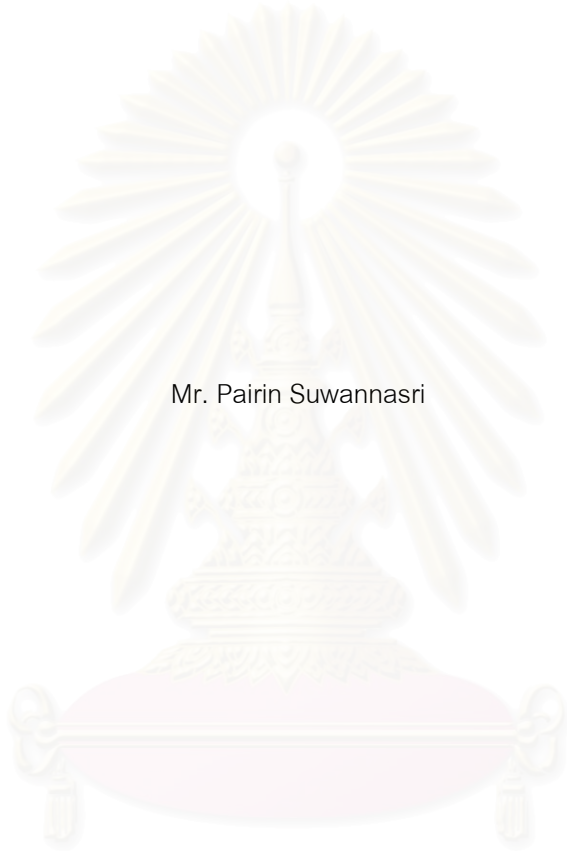
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2547

ISBN 974-17-6768-4

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

PARALLEL COMPUTATION OF SHALLOW WATER FLOW PROBLEMS



Mr. Pairin Suwannasri

สถาบันวิทยบริการ

A Thesis Submitted in Partial Fulfillment of the Requirements

จุฬาลงกรณ์มหาวิทยาลัย

for the Degree of Master of Science in Computational Science

Department of Mathematics

Faculty of Science

Chulalongkorn University

Academic Year 2004

ISBN 974-17-6768-4

Thesis Title                      PARALLEL COMPUTATION OF SHALLOW WATER FLOW  
PROBLEMS  
By                                      Mr. Pairin Suwannasri  
Field of study                      Mathematics  
Thesis Advisor                      Associate Professor Jack Asavanant, Ph.D.  
Thesis Co-advisor                      Professor Chidchanok Lursinsap, Ph.D.

---

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

..... Dean of the Faculty of Science  
(Professor Dr. Piamsak Menasveta)

THESIS COMMITTEE

..... Chairman  
(Assistant Professor Pronchai Satravaha, Ph.D. )

..... Thesis Advisor  
(Associate Professor Jack Asavanant, Ph.D.)

..... Thesis Co-advisor  
(Professor Chidchanok Lursinsap, Ph.D.)

..... Member  
(Assistant Professor Vimolrat Ngamaramvaranggul, Ph.D.)

..... Member  
(Assistant Professor Anusorn Chonwerayuth, Ph.D.)

ไพรินทร์ สุวรรณศรี : การคำนวณเชิงขนานของปัญหาการไหลในน้ำตื้น. (PARALLEL COMPUTATION OF SHALLOW WATER FLOW PROBLEMS) อาจารย์ที่ปรึกษา : รศ.ดร. จักร์ อัสวานันท์, อาจารย์ที่ปรึกษาร่วม : ศ.ดร. ชิดชนก เหลือสินทรัพย์ ; 93 หน้า. ISBN 974-17-6768-4

วิทยานิพนธ์นี้นำเสนอระเบียบวิธีการเชิงตัวเลขแบบขนานสำหรับแก้ปัญหาการไหลในน้ำตื้นแบบสองมิติ โดยการใช้แบบจำลองทางคณิตศาสตร์ วิธีการความละเอียดสูงของ Godunov ซึ่งอยู่บนพื้นฐานของตัวแก้ปัญหาค่า Riemann ความถูกต้องระดับสองจะถูกใช้ในการแก้สมการ shallow water ปัญหา Riemann จะถูกประมาณค่าโดยใช้วิธีการของ Harten Lax VanLeer(HLL) และวิธีการของ Roe โปรแกรมแบบขนานถูกสร้างขึ้นบนระบบแบบการกระจายที่ใช้หน่วยความจำร่วมกัน โดยใช้เทคนิคการแบ่งโดเมน Message Passing Interface(MPI) จะถูกนำมาช่วยในการสื่อสารระหว่างเครื่อง งานวิจัยนี้ยังได้มีการเปรียบเทียบประสิทธิภาพของการคำนวณแบบขนานด้วย

## สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.....คณิตศาสตร์.....  
สาขาวิชา.....วิทยาการคอมพิวเตอร์.....  
ปีการศึกษา.....2547.....

ลายมือชื่อนิสิต.....  
ลายมือชื่ออาจารย์ที่ปรึกษา.....  
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....

# # 4572429023 : MAJOR Computational Science

KEY WORD: Parallel computation/ Finite volume methods / Dam break problems.

PAIRIN SUWANNASRI : PARALLEL COMPUTATION OF SHALLOW WATER FLOW PROBLEMS. THESIS ADVISOR : ASSOC.PROF JACK ASAVANANT, Ph.D., THESIS COADVISOR : PROF. CHICHANOK LURSINSAP, Ph.D., 93 pp. ISBN 974-17-6768-4.

In this thesis, a parallel numerical method for solving two-dimensional shallow water flow problems is presented. A mathematical model is described. A high-resolution Godunov's method which is based on a second-order approximate Riemann solver is used to solve the 2-D shallow water equations. The local Riemann problem is solved by using the Harten, Lax and Van Leer approach (HLL) and by the Roe method. The parallel code program has been implemented on distributed-shared memory system, by using domain decomposition techniques. A message passing interface (MPI) is incorporated for inter-processor data communication. In addition, numerical solutions and performance results are also presented.

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

Department.....Mathematics..... Student's signature.....  
Field of study.....Computational Science..... Advisor's signature.....  
Academic year ...2004..... Co-advisor's signature.....

## Acknowledgements

First of all, I would like to express my deep gratitude to my supervisor, Assoc.Dr. Jack Asavanant, who is very kindly read and edited the thesis and offered many useful suggestion. I would also like to thank my co-supervisor, Prof.Dr. Chidchanok Lursinsap, who gave me wonderful suggestions.

My thanks also go to the dissertation committee, Assist.Prof.Dr. Pronchai Satravaha, Assist.Prof.Dr Vimolrat Ngamaramvaranggul and Assist.Prof.Dr. Anusorn Chonwerayuth, who made helpful comments on the suitability of thesis content.

I want to express my gratitude to Ubon Rajathanee University for their scholarship in supporting me during my education.

I would also like to thank Dr. Tawun Remsungnen who supported and advised me for using parallel computer. I would like to thank the Advanced Virtual and Intelligence Computing Research Center(AVIC) for support in enabling me to accomplish research and thanks especially to all of my friends in the Department of Mathematics for providing me constant encouragement .

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

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

# TABLE OF CONTENTS

	Page
ABSTRACT (THAI).....	iv
ABSTRACT (ENGLISH).....	v
ACKNOWLEDGEMENTS.....	vi
LIST OF TABLES.....	x
LIST OF FIGURES.....	xi
 <b>CHAPTER</b>	
<b>I INTRODUCTION.....</b>	<b>1</b>
<b>II DERIVATION OF SHALOW WATER EQUATIONS.....</b>	<b>3</b>
2.1 Conservation of Mass.....	3
2.2 Conservation of Momentum.....	5
<b>III NUMERICAL METHODS.....</b>	<b>10</b>
3.1 Conservative Finite Volume Method.....	10
3.2 Riemann Problem.....	13
3.3 Godunov's Method.....	15
3.3.1 The Wave-Propagation form of Godunov's Method.....	16
3.3.2 The CFL Condition.....	16
3.4 Approximate Riemann Solver.....	20
3.4.1 Roe Linearization.....	20
3.4.2 Roe Riemann Solver.....	22
3.4.2.1 Sweep in the $x$ -direction.....	22
3.4.2.2 Sweep in the $y$ -direction.....	23
3.4.3 HLL Riemann Solver.....	25
3.5 High-resolution Godunov's Method.....	26
3.5.1 Slope-Limiter.....	28
3.5.1.1 Minmod Slope Limiter.....	28
3.5.1.2 MC Slope Limiter.....	28

## TABLE OF CONTENTS (Continued)

	Page
3.5.2 Wave Limiter.....	29
3.5.3 High-resolution Method with Roe Approximation Riemann Solver...	29
3.5.4 High-resolution Method with HLL Approximation Riemann Solver...	37
Flowchart of the Numerical Method.....	40
<b>IV PARALLEL COMPUTATION.....</b>	<b>42</b>
4.1 Parallel Performance Measurements.....	42
4.2 Domain Decomposition.....	43
4.3 Message Passing Interface (MPI).....	46
Flowchart of the parallel solver.....	47
<b>V RESULTS.....</b>	<b>49</b>
5.1 Circular Dam Break Problem.....	49
5.1.1 Wet bed.....	49
5.1.2 Dry bed.....	51
5.2 Rectangular Dam Break Problem.....	62
5.2.1 Wet bed.....	64
5.2.2 Dry bed.....	66
5.3 Comparison of the Performance Results Between Roe and HLL Solvers..	77
<b>VI CONCLUSION.....</b>	<b>79</b>
REFERENCES.....	80
VITAE.....	82



## LIST OF TABLES

Table	Page
4.1 Text of serial code for 2D decomposition.....	45
4.2 Text of parallel code for 2D decomposition.....	45
5.1 Performance results of circular dam break with wet bed.....	50
5.2 Performance results of circular dam break with dry bed.....	51
5.3 Performance results of rectangular dam break with wet bed.....	64
5.4 Comparison of performance results between 2 differencing domain decomposition.....	65
5.5 Performance results of rectangular dam break with dry bed.....	67
5.6 Comparison of performance results between the Roe and the HLL solver for solving the circular dam beak .....	77
5.7 Comparison of performance results between the Roe and the HLL solver for solving the rectangular dam break.....	77

## LIST OF FIGURES

Figure	Page
2.1 (a) Control volume (b) Normal velocity.....	4
2.2 The directions of all forces and momentum fluxes in $x$ -direction.....	5
3.1 Finite volume grid cells in two space dimensions, where $\bar{Q}_{ij}$ represents a cell average .....	10
3.2 Solution of the linear Riemann problem in the $x-t$ plane.....	14
3.3 Fluctuations for the Godunov's method .....	19
3.4 Sketch of the linear variable extrapolation at the cell interface.....	27
3.5 (a) Waves moving normal to the cell interface. (b) The true velocity $(u, v)$ is at an angle to the grid, and information from cell $(i-1, j-1)$ should also affect the new value in cell $(i, j)$ .....	30
3.6 Transverse propagations at the interface $(i-1/2, j)$ .....	32
3.7 Flowchart of numerical method.....	41
4.1 Plot of speedup $S$ versus number of processors $P$ .....	43
4.2 Domain decomposition topologies for 1D (a) 2D (b) decompositions.....	44
4.3 Flow chart of parallel solver.....	48
5.1 Geometry of circular dam break problem.....	49
5.2 Relationship between the speedup and the number of processors in the cluster .....	50
5.3 Relationship between the efficiency and the number of processors in the cluster .....	51
5.4 Relationship between the speedup and the number of processors in the cluster .....	52
5.5 Relationship between the efficiency and the number of processors in the cluster .....	53
5.6 (a1)-(a11) show the circular dam break results in the case of wet bed, the left of figure shows water profile and the right of figure shows a contour plot of the depth.....	57
5.7 (b1)-(b11) show the circular dam break results in the case of dry bed, the left	

## LIST OF FIGURES (Continued)

Figure	Page
of figure shows water profile and the right of figure shows a contour plot of the depth.....	61
5.8 Geometry of rectangular dam break problem.....	62
5.9 Appropriate domain decomposition of the rectangular dam break problem, (a) for using 2 processors, (b) for using 4 processors, (c) for using 6 processors and (d) for using 8 processors.....	63
5.10 Relationship between the speedup and the number of processors in the cluster.....	65
5.11 Relationship between the efficiency and the number of processors in the cluster.....	66
5.12 Relationship between the speedup and the number of processors in the cluster.....	67
5.13 Relationship between the efficiency and the number of processors in the cluster.....	68
5.14 (c1)-(c11) show the rectangular dam break results in the case of wet bed, the left of figure shows water profile and the right of figure shows a contour plot of the depth.....	72
5.15 (d1)-(d11) show the rectangular dam break results in the case of dry bed, the left of figure shows water profile and the right of figure shows a contour plot of the depth.....	76
5.16 Comparison of the performance results between the Roe and the HLL solvers for solving the dam break problems (a)Circular dam break (b) Rectangular dam break .....	78

## CHAPTER I

### INTRODUCTION

Open-channel flows, flood mitigation, dam break analysis and some other related free surface flows are generally considered as shallow water problems. These solutions may include the propagation of shock waves or rarefaction waves. Mathematical models based on the shallow water equations (SWE) are widely used to simulate of this shallow water problems (see example [1] - [9]). The SWE can be derived from mass and momentum conservations (de Saint-Venant 1871). For two dimensions problem, there are three unknowns: water depth and velocities in  $x$ -and  $y$  - directions to be found. The distinguishing feature of the shallow water equations is that they admit both discontinuities and smooth solutions. Even the case in which the initial data is smooth can lead to discontinuous solutions at finite time. The nonlinear character of the equations suggests that solution to these equations may be limited to only some special cases. As an alternative, appropriate numerical methods can be used to obtain solution including discontinuities.

Of all numerical techniques, finite volume methods (FVM) have several advantages compared to other approaches. It combines the simplicity of finite difference methods with the geometric flexibility of finite element methods. Since FVM are based on the integral form of the conservation laws, a numerical scheme in conservation form can easily be constructed to capture the discontinuities. The fluid domain is subdivided into grid cells. Each cell average is modified in each time step by the normal flux through the edges of the grid cells. To approximate the numerical fluxes we must use the solution of local Riemann problem at cell interfaces. This approach was proposed by Godunov, and the schemes derived from this principle are generally called Godunov-type-scheme [3,6,10]. In the well known work of Godunov, the exact solution of the Riemann problem was used. For the linear system, the exact Riemann problems are mathematically too difficult to solve and the computation is too expensive. Today the exact solution of the Riemann problem is replaced with the approximate solutions that

are obtained by using approximate Riemann solvers. There have been quite a few efficient approximate Riemann solvers developed by various researchers, such as Roe (1981), VanLeer (1982), Harten (1983), and many others (see example [2],[3],[11]). Since the original Godunov method is first order-accurate, there are serious numerical oscillations occurred near the discontinuities. In order to avoid such oscillations, second order schemes of the high-resolution Godunov's methods are developed. In this work the high-resolution Godunov's methods are obtained by using the VanLeer's Monotonic Upstream Schemes for Conservation Laws (MUSCL) approach associated with the minmod and the monotonized-central-difference slope limiters.

In order to reduce the computational time, a parallel computation of the numerical method is used to solve the problems. The domain decomposition technique is used to divide the physical domain of the shallow water problems into sub-domain [4]. The MPI (Message Passing Interface) is implemented to define each sub-domain associated with each processor and is incorporated for inter-processor data communication. They also support process topology that is a mapping of processes in a communicator to an addressing scheme. The parallel program has been tested on the cluster that consists of eight PCs with 1.7 GHz Pentium 8 processors, 256 Mbytes RAM, 40 Gbytes Hard disk, and Fast Ethernet Switch Interconnection between nodes.

This thesis is organized as follows. Chapter 2 presents the derivation of the shallow water equations. Details of finite volume method and the sequential numerical algorithms are given in chapter 3. Chapter 4 introduces the concept of the parallel system. Numerical results and performance results of the parallel model are discussed in chapter 5. The conclusion of this thesis is in Chapter 6.

## CHAPTER II

### DERIVATION OF SHALLOW WATER EQUATIONS

This chapter concerns the derivation of equations governed the flows in 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 1871). The SWE is mathematical representation of water movement subject to the following assumptions:

- i) The fluid is assumed to be incompressible and inviscid.
- ii) The pressure distribution is hydrostatic.
- iii) Turbulence effect is negligible.

See [12] for a more complete description of the shallow water assumptions.

#### 2.1 Conservation of Mass

Consider a small rectangular element ( $dx \times dy \times h$ ) of water when  $h(x, y, t)$  is water depth as show in figure 2.1. Conservation of mass for this box or “control volume” states that:

*The rate of volume increase in the column is equal to the net volume flux into the column from all 4 sides.*

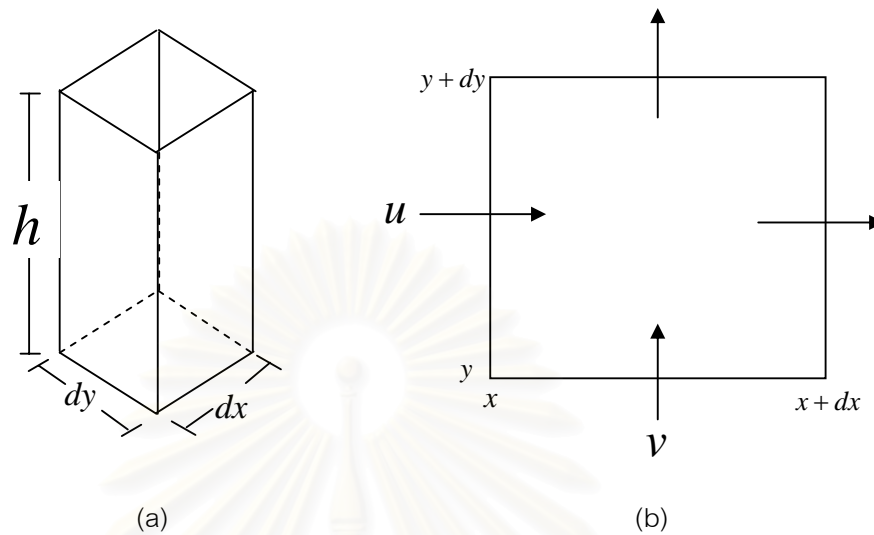


Figure 2.1(a) Control volume (b) Normal velocity.

Since  $dx$  and  $dy$  are fixed, the volume of water in the box can change only if the depth changes. The corresponding the rate of volume in the column is  $dx dy \frac{\partial h}{\partial t}$ .

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

$$(hu|_x - hu|_{x+dx}) dy \quad (2.1)$$

Using Taylor series expansion and omitting terms of higher orders in  $dx$ , equation (2.1) becomes

$$-dx dy \frac{\partial(hu)}{\partial x}$$

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

$$-dx dy \frac{\partial(hv)}{\partial y}$$



The conservation of mass can then be expressed quantitatively as

$$dxdy \frac{\partial h}{\partial t} = -dxdy \frac{\partial hu}{\partial x} - dxdy \frac{\partial hv}{\partial y} \quad (2.2)$$

Equation (2.2) can be rewritten

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

## 2.2 Conservation of Momentum

Conservation of momentum in the  $x$ -direction can be 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 the net force acting on the control volume in the  $x$ -direction.*

Figure 2.2 depicts the directions of all forces and momentum fluxes on the control volume.

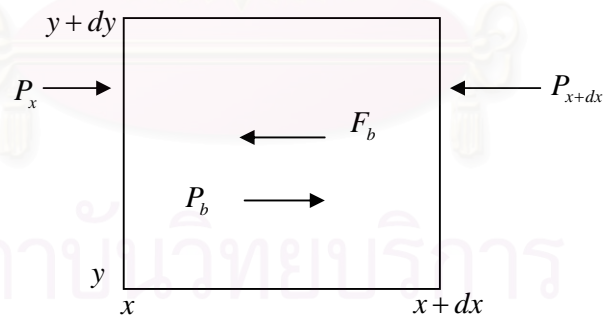


Figure 2.2 The directions of all forces and momentum fluxes in  $x$ -direction.

Here

- $P_x$  and  $P_{x+dx}$  are pressure forces on the sides of the box.
- $P_b$  is the pressure force due to a sloping bed.
- $F_b$  is the friction force at the channel bottom.



The rate of change of momentum is

$$\rho dx dy \frac{\partial(hu)}{\partial t} \quad (2.4)$$

The net influx of momentum through four vertical sides is

$$-\rho dx dy \frac{\partial hu^2}{\partial x} - \rho dx dy \frac{\partial hv}{\partial y}$$

The net of pressure force on two vertical sides normal to the  $x$ -direction is  $\left\{ \int P dA \Big|_x - \int P dA \Big|_{x+dx} \right\}$ . From the hydrostatic assumption, the pressure  $P$  is  $\rho gh$  and the net hydrostatic pressure forces becomes

$$-\frac{\rho g dy dx}{2} \frac{\partial h^2}{\partial x}$$

The pressure force due to the sloping bed is

$$P_b = \rho g \Delta x \Delta y h S_{0x}$$

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  $F_b = \tau_{bx} \Delta x \Delta y$ .

Equating these terms, the conservation of momentum is obtained

$$\begin{aligned} \rho dx dy \frac{\partial(hu)}{\partial t} = & -\frac{\rho g dx dy}{2} \frac{\partial h^2}{\partial x} - \rho dx dy \frac{\partial hu^2}{\partial x} \\ & - \rho dx dy \frac{\partial huv}{\partial y} + \rho g dx dy h S_{0x} - \tau_{bx} dx dy \end{aligned} \quad (2.5)$$

Dividing equation (2.5) by  $\rho \Delta x \Delta y$ , we have

$$\frac{\partial(hu)}{\partial t} = -\frac{g}{2} \frac{\partial h^2}{\partial x} - \frac{\partial(hu^2)}{\partial x} - \frac{\partial(huv)}{\partial y} + gh S_{0x} - gh S_{fx} \quad (2.6)$$

which can be written as

$$\frac{\partial(hu)}{\partial t} + \frac{\partial(hu^2)}{\partial x} + \frac{g}{2} \frac{\partial h^2}{\partial x} + \frac{\partial(huv)}{\partial y} = gh(S_{0x} - S_{fx}) \quad (2.7)$$

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

Similarly, the  $y$ -momentum gives

$$\frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial x} + \frac{g}{2} \frac{\partial(h^2)}{\partial y} + \frac{\partial(hv^2)}{\partial y} = gh(S_{0y} - S_{fy}) \quad (2.8)$$

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

Equations (2.3), (2.7) and (2.8) are called the two-dimensional shallow water equations. For future references, the SWE are

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} &= 0 \\ \frac{\partial(hu)}{\partial t} + \frac{\partial(hu^2)}{\partial x} + \frac{g}{2} \frac{\partial(h^2)}{\partial x} + \frac{\partial(huv)}{\partial y} &= gh(S_{0x} - S_{fx}) \\ \frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial x} + \frac{g}{2} \frac{\partial(h^2)}{\partial y} + \frac{\partial(hv^2)}{\partial y} &= gh(S_{0y} - S_{fy}). \end{aligned} \quad (2.9)$$

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 using the Manning resistance law

$$S_{fx} = \frac{un^2 \sqrt{u^2 + v^2}}{h^{4/3}} \quad \text{and} \quad S_{fy} = \frac{vn^2 \sqrt{u^2 + v^2}}{h^{4/3}}$$

in which  $n$  is the Manning roughness coefficient.

We can rewrite (2.9) in the matrix form as

$$\bar{q}_t + \bar{f}'(\bar{q})_x + \bar{g}'(\bar{q})_y = \bar{S} \quad (2.10)$$

where

$$\bar{q} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \bar{f}'(\bar{q}) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}, \quad \bar{g}'(\bar{q}) = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix} \quad (2.11)$$

and

$$\bar{S} = \begin{bmatrix} 0 \\ g(S_{0x} - S_{fx}) \\ g(S_{0y} - S_{fy}) \end{bmatrix}$$

These equations can be written in quasilinear form as

$$\bar{q}_t + f'(\bar{q})\bar{q}_x + g'(\bar{q})\bar{q}_y = \bar{S} \quad (2.12)$$

with the of Jacobian matrices

$$f'(\bar{q}) = \begin{bmatrix} 0 & 1 & 0 \\ -u^2 + gh & 2u & 0 \\ -uv & v & u \end{bmatrix} \quad (2.13)$$

$$g'(\bar{q}) = \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ -v^2 + gh & 0 & 2v \end{bmatrix}$$

Let  $c = \sqrt{gh}$  be the speed of gravity waves. Then the matrix  $f'(\bar{q})$  has eigenvalues and eigenvectors

$$\lambda^{x1} = u - c, \quad \lambda^{x2} = u, \quad \lambda^{x3} = u + c$$

$$\bar{r}^{x1} = \begin{bmatrix} 1 \\ u-c \\ v \end{bmatrix}, \bar{r}^{x2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \bar{r}^{x3} = \begin{bmatrix} 1 \\ u+c \\ v \end{bmatrix} \quad (2.14)$$

The Jacobian  $g'(\bar{q})$  has a similar set of eigenvalues and eigenvectors,

$$\lambda^{y1} = v-c, \quad \lambda^{y2} = u, \quad \lambda^{y3} = v+c$$

$$\bar{r}^{y1} = \begin{bmatrix} 1 \\ u \\ v-c \end{bmatrix}, \bar{r}^{y2} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \bar{r}^{y3} = \begin{bmatrix} 1 \\ u \\ v+c \end{bmatrix} \quad (2.15)$$



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## CHAPTER III

### NUMERICAL METHODS

#### 3.1 Conservative Finite Volume Method

Finite volume methods can be obtained on the basis of the integral form of conservation laws [11]. The fluid domain is subdivided into grid cells. The calculations are based on an approximation to the integral of certain quantities over each of these volumes or specifically, the cell average (i.e., this integral divided by the volume of each grid cell). These values are modified in each time step by the normal flux through the edges of the grid cells. The fluxes are determined by solving the Riemann problem for the two constant states at each side of the boundary edges. The Godunov's method for hyperbolic systems is generalized to nonlinear systems. The important step now is how to find solutions to the nonlinear Riemann problem at each cell interface.

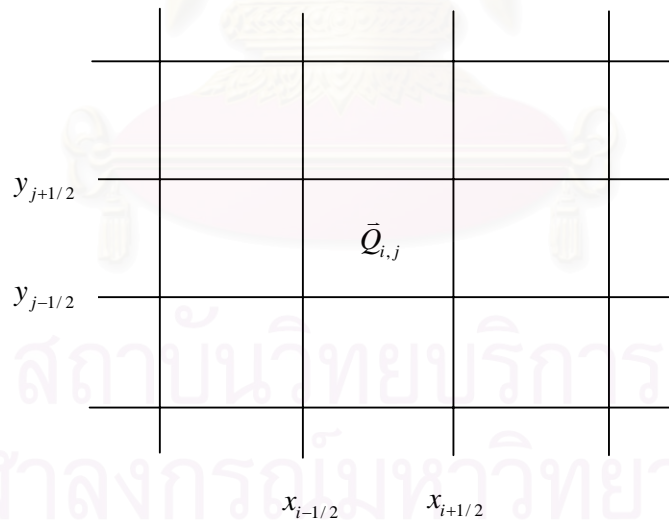


Figure 3.1 Finite volume grid cells in two space dimensions, where  $\bar{Q}_{ij}$  represents a cell average.

In deriving the two-dimensional conservation law  $\bar{q}_t + \bar{f}(\bar{q})_x + \bar{g}(\bar{q})_y = 0$ , the numerical domain is subdivided into rectangular grid cells of the form  $C_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$  as shows in Figure 3.1. Let  $\Delta x = x_{i+1/2} - x_{i-1/2}$  and  $\Delta y = y_{j+1/2} - y_{j-1/2}$ . The normal fluxes  $\bar{f}(\bar{q})$  and  $\bar{g}(\bar{q})$  represent the fluxes along the left and right edges and along the top and the bottom respectively. Integrating the conservation law over each grid cell, we obtain

$$\begin{aligned} \frac{d}{dt} \iint_{C_{ij}} \bar{q} dx dy + \iint_{C_{ij}} \bar{f}(\bar{q})_x dx dy + \iint_{C_{ij}} \bar{g}(\bar{q})_y dx dy &= 0 \\ \frac{d}{dt} \iint_{C_{ij}} \bar{q} dx dy + \int_{y_{j-1/2}}^{y_{j+1/2}} \bar{f}(\bar{q}(x_{i+1/2}, y, t)) dy - \int_{y_{j-1/2}}^{y_{j+1/2}} \bar{f}(\bar{q}(x_{i-1/2}, y, t)) dy \\ + \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{g}(\bar{q}(x, y_{j+1/2}, t)) dx - \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{g}(\bar{q}(x, y_{j-1/2}, t)) dx &= 0 \end{aligned} \quad (3.1)$$

Integrating (3.1) from  $t_n$  to  $t_{n+1}$ , yields

$$\begin{aligned} \iint_{C_{ij}} \bar{q}(x, y, t_{n+1}) dx dy - \iint_{C_{ij}} \bar{q}(x, y, t_n) dx dy &= - \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \bar{f}(\bar{q}(x_{i+1/2}, y, t)) dy dt \\ &+ \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \bar{f}(\bar{q}(x_{i-1/2}, y, t)) dy dt \\ &- \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{g}(\bar{q}(x, y_{j+1/2}, t)) dx dt \\ &+ \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{g}(\bar{q}(x, y_{j-1/2}, t)) dx dt \end{aligned} \quad (3.2)$$

Dividing (3.2) by the cell area  $\Delta x \Delta y$ , we obtain

$$\begin{aligned}
\frac{1}{\Delta x \Delta y} \iint_{C_{ij}} \bar{q}(x, y, t_{n+1}) dx dy &= \frac{1}{\Delta x \Delta y} \iint_{C_{ij}} \bar{q}(x, y, t_n) dx dy \\
&- \frac{1}{\Delta x \Delta y} \left[ \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \bar{f}(\bar{q}(x_{i+1/2}, y, t)) dy dt - \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \bar{f}(\bar{q}(x_{i-1/2}, y, t)) dy dt \right] \\
&- \frac{1}{\Delta x \Delta y} \left[ \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{g}(\bar{q}(x, y_{j+1/2}, t)) dx dt - \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{g}(\bar{q}(x, y_{j-1/2}, t)) dx dt \right]
\end{aligned}
\tag{3.3}$$

Equation (3.3) can be written as

$$\begin{aligned}
\frac{1}{\Delta x \Delta y} \iint_{C_{ij}} \bar{q}(x, y, t_{n+1}) dx dy &= \frac{1}{\Delta x \Delta y} \iint_{C_{ij}} \bar{q}(x, y, t_n) dx dy \\
&- \frac{\Delta t}{\Delta x} \left[ \frac{1}{\Delta t \Delta y} \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \bar{f}(\bar{q}(x_{i+1/2}, y, t)) dy dt - \frac{1}{\Delta t \Delta y} \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \bar{f}(\bar{q}(x_{i-1/2}, y, t)) dy dt \right] \\
&- \frac{\Delta t}{\Delta y} \left[ \frac{1}{\Delta t \Delta x} \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{g}(\bar{q}(x, y_{j+1/2}, t)) dx dt - \frac{1}{\Delta t \Delta x} \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{g}(\bar{q}(x, y_{j-1/2}, t)) dx dt \right]
\end{aligned}
\tag{3.4}$$

or, symbolically,

$$\bar{Q}_{ij}^{n+1} = \bar{Q}_{ij}^n - \frac{\Delta t}{\Delta x} [\bar{F}_{i+1/2,j}^n - \bar{F}_{i-1/2,j}^n] - \frac{\Delta t}{\Delta y} [\bar{G}_{i,j+1/2}^n - \bar{G}_{i,j-1/2}^n]
\tag{3.5}$$

where  $\bar{Q}_{i,j}^n$  represents a cell average over the  $(i, j)$  grid cell at time  $t_n$ ,

$$\bar{Q}_{ij}^n \approx \iint_{C_{ij}} \bar{q}(x, y, t_n) dx dy$$

with  $\bar{F}_{i-1/2,j}^n$  is some approximation to the average flux along  $x = x_{i-1/2,j}$ ,

$$\bar{F}_{i-1/2,j}^n \approx \frac{1}{\Delta t \Delta y} \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \bar{f}(\bar{q}(x_{i-1/2}, y, t)) dy dt$$

and  $\bar{G}_{i,j-1/2}^n$  is some approximation to the average flux along  $y = y_{i,j-1/2}$

$$\bar{G}_{i,j-1/2}^n \approx \frac{1}{\Delta t \Delta x} \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{g}(\bar{q}(x, y_{j-1/2}, t)) dx dt$$

A numerical flux at the edge of each grid cell is determined, based on the data at the beginning of each time step. These fluxes are used to update the cell average over a time step. This brings us to the discussion of solutions to the Riemann problem.

### 3.2 Riemann Problem

The Riemann problem in our context consists of the hyperbolic equations together with initial data which is piecewise constant with a single jump discontinuity at some point, say  $x = 0$ ,

$$\bar{q}(x, 0) = \begin{cases} \bar{q}_l & \text{if } x < 0 \\ \bar{q}_r & \text{if } x > 0 \end{cases} \quad (3.6)$$

At each cell interface  $x_{i-1/2,j}$ , there is a discontinuity with  $\bar{q}_l = \bar{Q}_{i-1,j}$  to the left and  $\bar{q}_r = \bar{Q}_{i,j}$  to the right. We can obtain information that can be used to compute the numerical flux. Generally, systems solution to the Riemann problem is written in terms of similarity variable  $x/t$  and consists of a finite set of waves that propagate away from the origin with constant wave speeds. For linear hyperbolic systems the Riemann problem is easily solved in terms of the eigenvalues and eigenvectors.

For 2D shallow water equations, a Riemann solution has three states: the original state to the left  $\bar{q}_l$ , the original state to the right  $\bar{q}_r$ , and a middle state between the two discontinuities. This middle state will be denoted as  $\bar{q}^*$ . (see Figure 3.2)



In solving the 2D shallow water equations we must sweep in each direction. For example, in the  $x$ -direction the Riemann problem consists of  $\bar{q}_t + \bar{f}(\bar{q})_x = 0$  together with the initial condition (3.6). Then we solve this Riemann problem as follows:

1. Determine whether each of the two waves is a shock or a rarefaction wave.
2. Determine the intermediate state  $\bar{q}^*$  between the two waves.
3. Determine the structure of the solution through any rarefaction waves.

For finite volume methods, this process is often simplified by using the approximate Riemann solver discussed in Section 3.4. Computing the exact Riemann solution can be expensive.

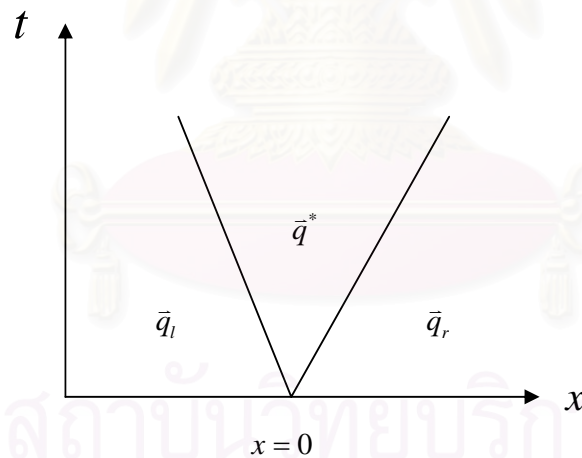


Figure 3.2 Solution of the linear Riemann problem in the  $x-t$  plane.

### 3.3 Godunov's Method

Godunov (1959) suggested the use of characteristic information of a hyperbolic system within the framework of a conservative method. He proposed that the numerical flux could be obtained by solving a local Riemann problem at each interface.

Godunov's method can be implemented for a general system of conservation laws described in the following procedure :

#### **Algorithm 1**

1. Reconstruct a piecewise polynomial function  $\tilde{q}^n(x, y, t_n)$  defined for all  $x$  and  $y$ , from the cell averages  $\bar{Q}_{ij}^n$ .
2. Compute the numerical fluxes.
3. Apply the flux-differencing formula (3.5) then repeat (1) in the next time step.

*In step1* : We reconstruct a function  $\tilde{q}^n(x, y, t_n)$  from the discrete cell averages. In the simplest case this is a piecewise constant function that takes the value  $\bar{Q}_{ij}^n$  in the  $ij^{th}$  grid cell, i.e.,

$$\tilde{q}(x, y, t_n) = \bar{Q}_{ij}^n \text{ for all } x \in C_{ij} \quad (3.7)$$

This reconstruction gives only a first-order accurate method. To obtain better accuracy one might consider using a better reconstruction, such as a piecewise linear function. This idea forms the basis for the high-resolution methods considered in Section 3.5.

*In step2* : We want to evaluate the flux function using solution from the Riemann problem at each cell interface. Let us denote the Riemann solution at the interface  $(i - 1/2, j)$  by  $\bar{Q}_{i-1/2, j}^\downarrow = \bar{q}^\downarrow(\bar{Q}_{i-1, j}^n, \bar{Q}_{ij}^n)$ . The numerical flux at this edge is determined by

$$\begin{aligned}\bar{F}_{i-1/2,j} &= \bar{f}(\bar{Q}_{i-1/2,j}^\downarrow), \\ \bar{G}_{i,j-1/2} &= \bar{g}(\bar{Q}_{i,j-1/2}^\downarrow),\end{aligned}\tag{3.8}$$

where  $\bar{Q}_{i-1/2,j}^\downarrow$  is obtained from the local Riemann problem for  $\bar{q}_t + \bar{f}(\bar{q})_x = 0$  with data  $\bar{Q}_{i-1,j}$  and  $\bar{Q}_{i,j}$ , while  $\bar{Q}_{i,j-1/2}^\downarrow$  is obtained from the local Riemann problem for  $\bar{q}_t + \bar{g}(\bar{q})_y = 0$  with data  $\bar{Q}_{i,j-1}$  and  $\bar{Q}_{i,j}$ .

In this thesis, the numerical fluxes are calculated using the approximate Riemann solvers (e.g., Roe and HLL solvers) described in Section 3.4.

*In step3* : We use the numerical value  $\bar{Q}_{i-1/2,j}^\downarrow$ , etc. from step2 to update the cell average by substituting into the Godunov's scheme.

### 3.3.1 The Wave-Propagation form of Godunov's Method

In step 2 we can determine the solution of the local Riemann problem which consists of a set of waves. For a general  $m \times m$  linear system,  $\bar{q}_t + A\bar{q}_x = 0$  ( $\bar{f}(\bar{q}) = A\bar{q}$ ), the solution of Riemann with arbitrary initial data  $\bar{q}_l$  and  $\bar{q}_r$ , consists of  $m$  discontinuities traveling with speed  $\lambda^1, \lambda^2, \dots, \lambda^m$  which are the eigenvalue of the  $m \times m$  matrix  $A$ . To solve the Riemann problem we take the initial data  $(\bar{Q}_l, \bar{Q}_r)$  and decompose the jump  $\bar{Q}_r - \bar{Q}_l$  into eigenvectors of  $A$ .

$$\bar{Q}_r - \bar{Q}_l = \alpha^1 \bar{r}^1 + \alpha^2 \bar{r}^2 + \dots + \alpha^m \bar{r}^m\tag{3.9}$$

This requires solving the linear system of equations

$$R\bar{\alpha} = \bar{Q}_r - \bar{Q}_l\tag{3.10}$$

for the vector  $\bar{\alpha} = R^{-1}(\bar{Q}_r - \bar{Q}_l)$  and  $R$  is the matrix of eigenvectors. Since  $\alpha^p \bar{r}^p$  is the jump in  $\bar{Q}$  across the  $p^{\text{th}}$  wave in the solution to the Riemann problem, we introduce the notation

$$\bar{W}^p = \alpha^p \bar{r}^p \quad (3.11)$$

for these waves.

The solution of this Riemann problem at the cell interface is

$$\begin{aligned} \bar{Q}^\downarrow &= \bar{Q}_l + \sum_{p:\lambda^p < 0} \bar{W}^p \\ \text{or} \quad \bar{Q}^\downarrow &= \bar{Q}_r - \sum_{p:\lambda^p > 0} \bar{W}^p. \end{aligned} \quad (3.12)$$

We can compute the numerical flux as

$$\begin{aligned} \bar{F}_{i-1/2,j} &= \bar{f}(\bar{Q}_{i-1/2,j}^\downarrow) = A \bar{Q}_{i-1/2,j}^\downarrow \\ &= A(\bar{Q}_{ij} - \sum_{p:\lambda^p > 0} \bar{W}_{i-1/2,j}^p) \\ &= A \bar{Q}_{ij} - \sum_{p:\lambda^p > 0} A \bar{W}_{i-1/2,j}^p \\ &= A \bar{Q}_{ij} - \sum_{p=1}^m (\lambda_{i-1/2,j}^p)^+ \bar{W}_{i-1/2,j}^p \end{aligned}$$

$$\text{Similarly, we obtain } \bar{F}_{i+1/2,j} = A \bar{Q}_{ij} + \sum_{p=1}^m (\lambda_{i+1/2,j}^p)^- \bar{W}_{i+1/2,j}^p.$$

The solution of the Riemann problem  $\bar{q}_t + B \bar{q}_x = 0$  ( $\bar{g}(\bar{q}) = B \bar{q}$ ) is of the same structure as (3.12). Hence the numerical fluxes in  $y$ -direction are obtained by

$$\bar{G}_{i,j-1/2} = A \bar{Q}_{ij} - \sum_{p=1}^m (\lambda_{i,j-1/2}^p)^+ \bar{W}_{i,j-1/2}^p$$

$$\text{and } \bar{G}_{i,j+1/2} = A \bar{Q}_{ij} + \sum_{p=1}^m (\lambda_{i,j+1/2}^p)^- \bar{W}_{i,j+1/2}^p.$$

Substituting those numerical fluxes into the flux differencing (3.5), we obtain

$$\begin{aligned} \bar{Q}_{ij}^{n+1} = \bar{Q}_{ij}^n - \frac{\Delta t}{\Delta x} & \left[ \sum_{p=1}^m (\lambda_{i-1/2,j}^p)^+ \bar{W}_{i-1/2,j}^p + \sum_{p=1}^m (\lambda_{i+1/2,j}^p)^- \bar{W}_{i+1/2,j}^p \right] \\ & - \frac{\Delta t}{\Delta y} \left[ \sum_{p=1}^m (\lambda_{i,j-1/2}^p)^+ \bar{W}_{i,j-1/2}^p + \sum_{p=1}^m (\lambda_{i,j+1/2}^p)^- \bar{W}_{i,j+1/2}^p \right] \end{aligned} \quad (3.13)$$

The cell average is affected by the right-going waves from  $x_{i-1/2,j}$ , the left-going waves from  $x_{i+1/2,j}$ , the up-going waves from  $y_{i,j-1/2}$ , and the down-going waves from  $y_{i,j+1/2}$ . As a shorthand notation, we introduce the following symbols:

$$\begin{aligned} \hat{\lambda}^- \Delta \bar{Q}_{i-1/2,j} &= \sum_{p=1}^m (\lambda_{i-1/2,j}^p)^- \bar{W}_{i-1/2,j}^p \\ \hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j} &= \sum_{p=1}^m (\lambda_{i-1/2,j}^p)^+ \bar{W}_{i-1/2,j}^p \end{aligned}$$

and

$$\begin{aligned} B^- \Delta \bar{Q}_{i,j-1/2} &= \sum_{p=1}^m (\lambda_{i,j-1/2}^p)^- \bar{W}_{i,j-1/2}^p \\ B^+ \Delta \bar{Q}_{i,j-1/2} &= \sum_{p=1}^m (\lambda_{i,j-1/2}^p)^+ \bar{W}_{i,j-1/2}^p \end{aligned}$$

Hence (3.13) can be rewritten in the fluctuation form as

$$\begin{aligned} \bar{Q}_{ij}^{n+1} = \bar{Q}_{ij}^n - \frac{\Delta t}{\Delta x} & (\hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j} + \hat{\lambda}^- \Delta \bar{Q}_{i+1/2,j}) \\ & - \frac{\Delta t}{\Delta y} (B^+ \Delta \bar{Q}_{i,j-1/2} + B^- \Delta \bar{Q}_{i,j+1/2}) \end{aligned} \quad (3.14)$$

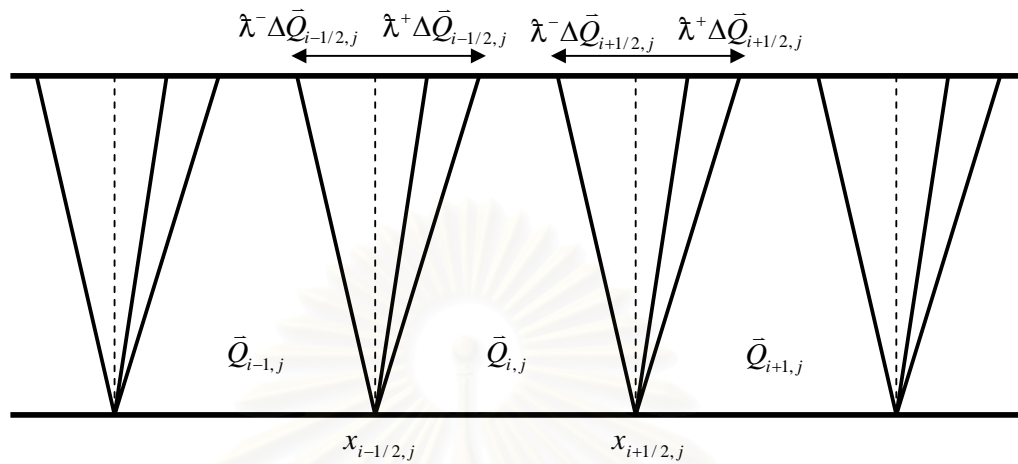


Figure 3.3 Fluctuations for the Godunov's method.

### 3.3.2 The CFL Condition

The CFL condition was proposed by Courant, Friedrichs and Lewy in 1928. They recognized the following necessary stability condition for any numerical method:

**CFL condition** : *A numerical method can be convergent only if its numerical domain of dependence contains the true domain of dependence of the PDE, at least in the limit as  $\Delta t$  and  $\Delta x$  go to zero.*

It is important to bear in mind that the CFL condition is only a necessary condition for stability (and hence convergence). It is, of course, not sufficient to guarantee stability.

For a hyperbolic system of equations there is generally a set of  $m$  wave speeds  $\lambda^1, \dots, \lambda^m$ . Then we define the Courant number by

$$\nu = \frac{\Delta t}{\Delta x} \max_{p=1,2,\dots,m} |\lambda^p|$$

For a scheme (3.5) and (3.14) the CFL condition leads to a necessary condition  $\nu \leq 1$ .

### 3.4 Approximate Riemann Solver

In Godunov's method we only need to determine  $\bar{Q}^\downarrow$ , the state along  $x/t = 0$  based on the Riemann data  $\bar{Q}_l$  and  $\bar{Q}_r$ . The process of solving the Riemann problem is thus often quite expensive for the nonlinear systems of conservation law and we must use more information to extend to the high-resolution methods. Even so, it is often true that it is not necessary to compute the exact solution to the Riemann problem in order to the numerical calculations.

A wide variety of approximate Riemann solvers have been proposed. Most of them can be applied with much less computational cost than the exact Riemann solver and yet almost equally good results in many cases when used in the Godunov or high-resolution methods.

#### 3.4.1 Roe Linearization

To avoid difficulty in solving the nonlinear Riemann problem, we use a linearized approximate Riemann solver normal to each cell interface. Then this linear approach is easily extended to the nonlinear case.

From the quasilinear form of shallow water equations (2.12), after neglecting the source term temporarily, we apply the Roe linearization to obtain the Roe matrix in form of

$$q_t + \hat{A}\bar{q}_x + \hat{B}\bar{q}_y = 0 \quad (3.15)$$

where the matrix  $\hat{A}$  (approximation of  $\bar{f}'(\bar{q})$  near the interface) and the Roe averages  $\bar{h}, \hat{u}$  and  $\hat{v}$  are

$$\hat{A} = \begin{bmatrix} 0 & 1 & 0 \\ -\hat{u}^2 + g\bar{h} & 2\hat{u} & 0 \\ -\hat{u}\hat{v} & \hat{v} & \hat{u} \end{bmatrix} \quad (3.16)$$

$$\bar{h} = \frac{1}{2}(h_{i-1,j} + h_{i,j}) \quad (3.17)$$

$$\hat{u} = \frac{\sqrt{h_{i-1,j}}u_{i-1,j} + \sqrt{h_{i,j}}u_{i,j}}{\sqrt{h_{i-1,j}} + \sqrt{h_{i,j}}} \quad (3.18)$$

$$\hat{v} = \frac{\sqrt{h_{i-1,j}}v_{i-1,j} + \sqrt{h_{i,j}}v_{i,j}}{\sqrt{h_{i-1,j}} + \sqrt{h_{i,j}}} \quad (3.19)$$

The matrix  $\hat{A}$  has eigenvalues and eigenvectors

$$\hat{\lambda}^{x1} = \hat{u} - \hat{c}, \quad \hat{\lambda}^{x2} = \hat{u}, \quad \hat{\lambda}^{x3} = \hat{u} + \hat{c}$$

$$\hat{r}^{x1} = \begin{bmatrix} 1 \\ \hat{u} - \hat{c} \\ \hat{v} \end{bmatrix}, \quad \hat{r}^{x2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \hat{r}^{x3} = \begin{bmatrix} 1 \\ \hat{u} + \hat{c} \\ \hat{v} \end{bmatrix}$$

respectively, where  $\hat{c} = \sqrt{g\bar{h}}$  is the speed of gravity waves.

Similarly, the matrix  $\hat{B}$  (approximation of  $\bar{g}'(\bar{q})$  near the interface) and the Roe averages  $\bar{h}, \hat{u}$  and  $\hat{v}$  can be written as

$$\hat{B} = \begin{bmatrix} 0 & 0 & 1 \\ -\hat{u}\hat{v} & \hat{u} & \hat{v} \\ -\hat{v}^2 + g\bar{h} & \hat{v} & 2\hat{v} \end{bmatrix} \quad (3.20)$$

$$\bar{h} = \frac{1}{2}(h_{i,j-1} + h_{i,j}) \quad (3.21)$$



$$\hat{u} = \frac{\sqrt{h_{i,j-1}}u_{i,j-1} + \sqrt{h_{i,j}}u_{i,j}}{\sqrt{h_{i,j-1}} + \sqrt{h_{i,j}}} \quad (3.22)$$

$$\hat{v} = \frac{\sqrt{h_{i,j-1}}v_{i,j-1} + \sqrt{h_{i,j}}v_{i,j}}{\sqrt{h_{i,j-1}} + \sqrt{h_{i,j}}} \quad (3.23)$$

The matrix  $\hat{B}$  has eigenvalues and eigenvectors

$$\begin{aligned} \hat{\lambda}^{y1} &= \hat{v} - \hat{c}, & \hat{\lambda}^{y2} &= \hat{u}, & \hat{\lambda}^{y3} &= \hat{v} + \hat{c} \\ \hat{r}^{y1} &= \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} - \hat{c} \end{bmatrix}, & \hat{r}^{y2} &= \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, & \hat{r}^{y3} &= \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} + \hat{c} \end{bmatrix} \end{aligned} \quad (3.24)$$

respectively, where  $\hat{c} = \sqrt{gh}$  is the speed of gravity waves.

### 3.4.2 Roe Riemann Solvers

Recalling the wave-propagation form of Godunov's method, to update the cell average in next time step we need to compute the wave and the wave speed. They can be achieved by solving the Riemann problem. In the process to solve the Riemann problem by Roe solver we start by sweeping in the  $x$ -direction along each row of the grid cell. Then we sweep in the  $y$ -direction along each column.

#### 3.4.2.1 Sweep in the $x$ -direction

First we apply the Roe linearization to  $\bar{q}_r + \bar{f}'(\bar{q})\bar{q}_x = 0$ , and obtain the Roe matrix  $\hat{A}$  as described in Section 3.4.1. At the interface we decompose the jump  $\bar{q}_r - \bar{q}_l$  into eigenvectors of  $\hat{A}$ :

$$\bar{q}_r - \bar{q}_l = \alpha^1 \hat{r}^{x1} + \alpha^2 \hat{r}^{x2} + \alpha^3 \hat{r}^{x3}$$

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \alpha^1 \begin{bmatrix} 1 \\ \hat{u} - \hat{c} \\ \hat{v} \end{bmatrix} + \alpha^2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \alpha^3 \begin{bmatrix} 1 \\ \hat{u} + \hat{c} \\ \hat{v} \end{bmatrix} \quad (3.25)$$

where

$$\begin{aligned} d_1 &= \bar{h}_{ij} - \bar{h}_{i-1,j}, \\ d_2 &= \hat{u}_{ij} - \hat{u}_{i-1,j}, \\ d_3 &= \hat{v}_{ij} - \hat{v}_{i-1,j}. \end{aligned}$$

Solving the linear system (3.25) for  $\bar{\alpha}$ , we obtain

$$\begin{aligned} \alpha^1 &= \frac{(\hat{u} + \hat{c})d_1 - d_2}{2\hat{c}}, \\ \alpha^2 &= \frac{-(\hat{u} - \hat{c})d_1 + d_2}{2\hat{c}}, \end{aligned}$$

and

$$\alpha^3 = -\hat{v}d_1 + d_3.$$

Consequently, the waves  $\bar{W}^p$  are given by

$$\bar{W}_{i-1/2,j}^1 = \alpha^1 \begin{bmatrix} 1 \\ \hat{u} - \hat{c} \\ \hat{v} \end{bmatrix}, \bar{W}_{i-1/2,j}^2 = \alpha^2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \bar{W}_{i-1/2,j}^3 = \alpha^3 \begin{bmatrix} 1 \\ \hat{u} + \hat{c} \\ \hat{v} \end{bmatrix}$$

and the corresponding wave speeds are  $s_{i-1/2,j}^1 = \hat{u} - \hat{c}$ ,  $s_{i-1/2,j}^2 = \hat{c}$ , and  $s_{i-1/2,j}^3 = \hat{u} + \hat{c}$ .

#### 3.4.2.2 Sweep in the y-direction

At the interface  $(i, j-1/2)$ , we decompose the jump  $\bar{q}_r - \bar{q}_l$  into eigenvectors of  $\hat{B}$ :

$$\bar{q}_r - \bar{q}_l = \alpha^1 \hat{r}^{y1} + \alpha^2 \hat{r}^{y2} + \alpha^3 \hat{r}^{y3}$$

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \alpha^1 \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} - \hat{c} \end{bmatrix} + \alpha^2 \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} + \alpha^3 \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} + \hat{c} \end{bmatrix} \quad (3.26)$$

where

$$\begin{aligned} d_1 &= \bar{h}_{ij} - \bar{h}_{i,j-1}, \\ d_2 &= \hat{u}_{ij} - \hat{u}_{i,j-1}, \\ d_3 &= \hat{v}_{ij} - \hat{v}_{i,j-1}. \end{aligned}$$

Solving the linear system (3.26) for  $\bar{\alpha}$ , we obtain

$$\begin{aligned} \alpha^1 &= \frac{(\hat{u} + \hat{c})d_1 - d_2}{2\hat{c}}, \\ \alpha^2 &= \frac{-(\hat{u} - \hat{c})d_1 + d_2}{2\hat{c}}, \end{aligned}$$

and

$$\alpha^3 = -\hat{v}d_1 + d_3.$$

The waves  $\bar{W}^p$  are given by

$$\bar{W}_{i-1/2,j}^1 = \alpha^1 \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} - \hat{c} \end{bmatrix}, \quad \bar{W}_{i-1/2,j}^2 = \alpha^2 \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \quad \bar{W}_{i-1/2,j}^3 = \alpha^3 \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} + \hat{c} \end{bmatrix}$$

and the corresponding wave speeds are  $s_{i,j-1/2}^1 = \hat{v} - \hat{c}$ ,  $s_{i,j-1/2}^2 = \hat{c}$ , and  $s_{i,j-1/2}^3 = \hat{v} + \hat{c}$ .

It should be noted that the Roe Riemann solver works well for many shallow water flow problems except when there is a dry region. In that case, the Roe Riemann solver may produce negative depth. In this thesis, the HLL Riemann solver is used as an alternative to approximate the flux in the dry case.

### 3.4.3 HLL Riemann Solver

The HLL Riemann solver was proposed by Harten, Lax and Leer in 1983 [2]. Information at the interface must be determined subject to the wave speed of propagation.

$$\bar{Q}^\downarrow \cdot \bar{n} \equiv \bar{q}^\downarrow(\bar{q}_l, \bar{q}_r) \cdot \bar{n} = \begin{cases} \bar{q}_l \cdot \bar{n} & \text{for } x/t < s_l \\ \bar{q}^* \cdot \bar{n} & \text{for } s_l < x/t < s_r \\ \bar{q}_r \cdot \bar{n} & \text{for } s_r < x/t \end{cases} \quad (3.27)$$

where  $\bar{n}$  is the outward normal unit vector; the subscripts  $r$  and  $l$  referred to the right and left cell interfaces,  $s_l$  and  $s_r$  are the smallest and largest wave speeds of propagation, respectively, and the state  $\bar{q}^*$  is determined from

$$\bar{q}^* \cdot \bar{n} = \frac{(s_r \bar{q}_r - s_l \bar{q}_l) - (\bar{F}_r \cdot \bar{n} - \bar{F}_l \cdot \bar{n})}{s_r - s_l} \quad (3.28)$$

We now turn to the determination of the associated numerical flux. We set

$$\bar{F}^\downarrow \cdot \bar{n} = \begin{cases} \bar{F}_l \cdot \bar{n} & \text{when } 0 < s_l \\ \bar{F}^* \cdot \bar{n} & \text{when } s_l < 0 < s_r \\ \bar{F}_r \cdot \bar{n} & \text{when } s_r < 0. \end{cases} \quad (3.29)$$

Here the numerical flux at the star region is determined by

$$\bar{F}^* \cdot \bar{n} = \frac{s_r \bar{F}_l \cdot \bar{n} - s_l \bar{F}_r \cdot \bar{n} + s_l s_r (\bar{q}_r - \bar{q}_l)}{s_r - s_l} \quad (3.30)$$

where  $\bar{F}_r = \bar{f}(\bar{q}_r)$  or  $\bar{g}(\bar{q}_r)$  and  $\bar{F}_l = \bar{f}(\bar{q}_l)$  or  $\bar{g}(\bar{q}_l)$ .

The wave speeds  $s_l$  and  $s_r$  can be estimated via the two expansion approaches due to Toro:

$$\begin{aligned} s_l &= \min(w_l \cdot \bar{n} - \sqrt{gh_l}, u^* - \sqrt{gh^*}) \\ s_r &= \max(w_r \cdot \bar{n} + \sqrt{gh_r}, u^* + \sqrt{gh^*}) \end{aligned} \quad (3.31)$$

where  $w = (u, v)$  and

$$\begin{aligned} u^* &= \frac{1}{2}(w_l + w_r) \cdot \bar{n} + \sqrt{gh_l} - \sqrt{gh_r} \\ \sqrt{gh^*} &= \frac{1}{2}(\sqrt{gh_l} + \sqrt{gh_r}) + \frac{1}{4}(w_l - w_r) \cdot \bar{n} \end{aligned} \quad (3.32)$$

If the cell on the right of the interface is dry, then

$$s_l = w_l \cdot \bar{n} - \sqrt{gh_l} \quad \text{and} \quad s_r = w_l \cdot \bar{n} + 2\sqrt{gh_l} \quad (3.33)$$

On the contrary, if the cell on the right of the interface is dry, we have

$$s_l = w_r \cdot \bar{n} - 2\sqrt{gh_r} \quad \text{and} \quad s_r = w_r \cdot \bar{n} + \sqrt{gh_r} \quad (3.34)$$

For the homogeneous shallow water equations, if the middle stage  $\bar{q}^*$  is determined from (3.28), using (3.31) for the speeds  $s_l$  and  $s_r$ , the depth  $h^*$  in the middle state is always non-negative [2].

### 3.5 High-resolution Godunov's Method

In Section 3.3, we introduced the original Godunov's method that is first order accurate, giving poor accuracy in smooth region of the flow. Moreover, shocks tend to be heavily smeared and poorly resolved on the grid. In this section a high-resolution Godonov's method is introduced. It is at least second order accurate on smooth solution and yet give well resolved result, avoiding the nonphysical oscillations.

There are various techniques to achieve the high-resolution scheme, for example MUSCL (Monotonic Upstream-centered Scheme for Conservation Laws). The MUSCL procedure applied to the Godunov's method consists of a linear extrapolation of the corresponding variables at the cell interfaces. In doing this, we introduce a piecewise linear function of the form.

$$\tilde{q}^n(x, y, t_n) = \bar{Q}_{ij}^n + \frac{1}{2} \bar{\sigma}_{ij}^n \Delta x.$$

where  $\bar{\sigma}_{ij}^n$  is the slope in  $ij^{th}$  grid cell. This replaces the first step of the original Godunov's method (see Figure 3.4).

In using the slope  $\bar{\sigma}_{ij}^n$  on a piecewise linear function, we must consider how to limit the slope for second-order accuracy while guaranteeing that no nonphysical oscillations will arise.

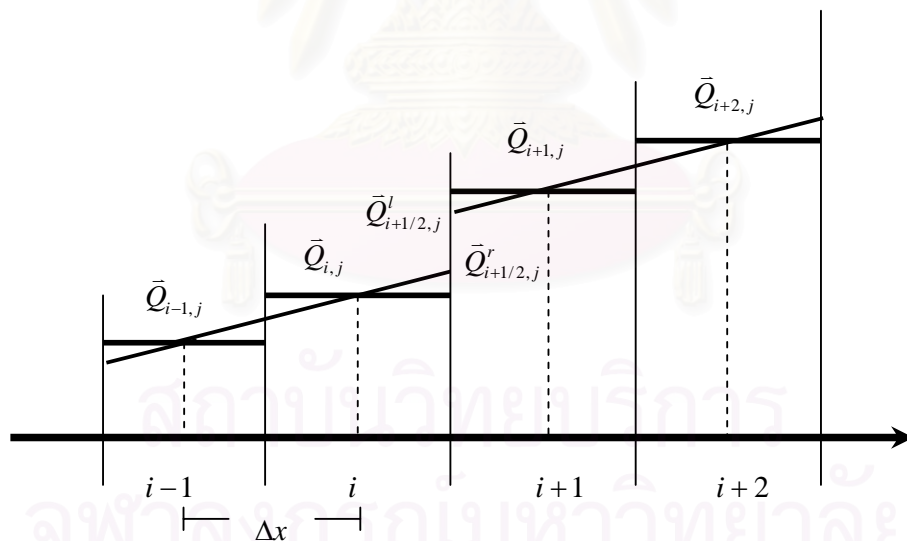


Figure 3.4 Sketch of the linear variable extrapolation at the cell interface.

### 3.5.1 Slope -Limiter

There are many choices of slope that give second-order accuracy for smooth solutions while still satisfying the TVD property. For examples, the minmod slope limiter, the superbee slope limiter, the van leer slope limiter, the monotized-central-difference limiter (MC limiter), etc. In this work we use the minmod slope limiter and the MC slope limiter.

#### 3.5.1.1 Minmod Slope Limiter

$$\bar{\sigma}_{ij} = \min \text{mod} \left( \frac{\bar{Q}_{i-1,j} - \bar{Q}_{i-2,j}}{\Delta x}, \frac{\bar{Q}_{i,j} - \bar{Q}_{i-1,j}}{\Delta x} \right) \quad (3.35)$$

where the minmod function of two arguments is defined by

$$\min \text{mod}(a,b) = \begin{cases} a & \text{if } |a| \leq |b| \text{ and } ab > 0 \\ b & \text{if } |b| \leq |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases} \quad (3.36)$$

The minmod technique compares the two slopes with the choice of a smaller one. If the two slopes have different sign then  $\bar{Q}_{i,j}$  must take on a local maximum or minimum value. It is easy to check in this case and we must set  $\bar{\sigma}_{ij} = 0$  in order to satisfy TVD property. In this work, the minmod method is used to reduce the oscillation of dry bed case.

#### 3.5.1.2 MC Limiter

$$\bar{\sigma}_{ij} = \min \text{mod} \left( \left( \frac{\bar{Q}_{i+1,j} - \bar{Q}_{i-1,j}}{2\Delta x} \right), 2 \left( \frac{\bar{Q}_{i,j} - \bar{Q}_{i-1,j}}{\Delta x} \right), 2 \left( \frac{\bar{Q}_{i+1,j} - \bar{Q}_{i,j}}{\Delta x} \right) \right) \quad (3.37)$$

The MC limiter increases the slopes in these two cells to twice the value of the minmod slopes and still have TVD property satisfied. This reconstruction will lead to sharper resolution of the discontinuity in the next time step than we would obtain with the minmod slopes.

### 3.5.2 Wave limiter

The wave limiter version of  $\bar{W}_{i-1/2,j}^p$  is determined by

$$\tilde{W}_{i-1/2,j}^p = \phi(\theta_{i-1/2,j}^p) \bar{W}_{i-1/2,j}^p \quad (3.38)$$

where  $\theta_{i-1/2,j}^p$  should be some measure of the smoothness of the  $p^{\text{th}}$  characteristic component of solution defined by

$$\theta_{i-1/2,j}^p = \frac{\|\bar{W}_{I-1/2,j}^p\|}{\|\bar{W}_{i-1/2,j}^p\|} \quad (3.39)$$

The index  $I$  here is used to represent the interface on the upwind side of  $x_{i-1/2,j}$ , that is

$$I = \begin{cases} i-1 & \text{if } s_{i-1/2,j}^p > 0 \\ i+1 & \text{if } s_{i-1/2,j}^p < 0 \end{cases} \quad (3.40)$$

The function  $\phi(\theta)$  is the flux-limiter function.

$$\begin{aligned} \text{min mod} &: \phi(\theta) = \text{min mod}(1, \theta) \\ \text{MC} &: \max(0, \min(\frac{1+\theta}{2}, 2, 2\theta)) \end{aligned}$$

### 3.5.3 High-resolution Method with Roe Approximate Riemann Solver

Recall the fluctuation form of Godunov's method (3.14)



$$\begin{aligned}\bar{Q}_{ij}^{n+1} = \bar{Q}_{ij}^n &- \frac{\Delta t}{\Delta x} (\lambda^+ \Delta \bar{Q}_{i-1/2,j} + \lambda^- \Delta \bar{Q}_{i+1/2,j}) \\ &- \frac{\Delta t}{\Delta y} (\mathbf{B}^+ \Delta \bar{Q}_{i,j-1/2} + \mathbf{B}^- \Delta \bar{Q}_{i,j+1/2})\end{aligned}\quad (3.41)$$

To improve the Godonov's method we introduce the correction terms  $\tilde{\mathbf{F}}$  and  $\tilde{\mathbf{G}}$  into (3.41). This gives

$$\begin{aligned}\bar{Q}_{ij}^{n+1} = \bar{Q}_{ij}^n &- \frac{\Delta t}{\Delta x} (\lambda^+ \Delta \bar{Q}_{i-1/2,j} + \lambda^- \Delta \bar{Q}_{i+1/2,j}) \\ &- \frac{\Delta t}{\Delta y} (\mathbf{B}^+ \Delta \bar{Q}_{i,j-1/2} + \mathbf{B}^- \Delta \bar{Q}_{i,j+1/2}) \\ &- \frac{\Delta t}{\Delta x} (\tilde{\mathbf{F}}_{i+1/2,j} + \tilde{\mathbf{F}}_{i-1/2,j}) - \frac{\Delta t}{\Delta y} (\tilde{\mathbf{G}}_{i,j+1/2} + \tilde{\mathbf{G}}_{i,j-1/2})\end{aligned}\quad (3.42)$$

In the first-order accurate method (3.41) the update value of  $\bar{Q}_{ij}^{n+1}$  depends on only the three values  $\bar{Q}_{ij}$ ,  $\bar{Q}_{i-1,j}$  and  $\bar{Q}_{i,j-1}$  shown in Figure 3.5(a). This is not quite a good representation, since the flow of information is at an angle to the grid, (see Figure 3.5(b)). The value  $\bar{Q}_{i-1,j-1}$  should also affect  $\bar{Q}_{ij}^{n+1}$ . This is the principle idea of the Corner-Transport Upwind method (CTU method) [11]

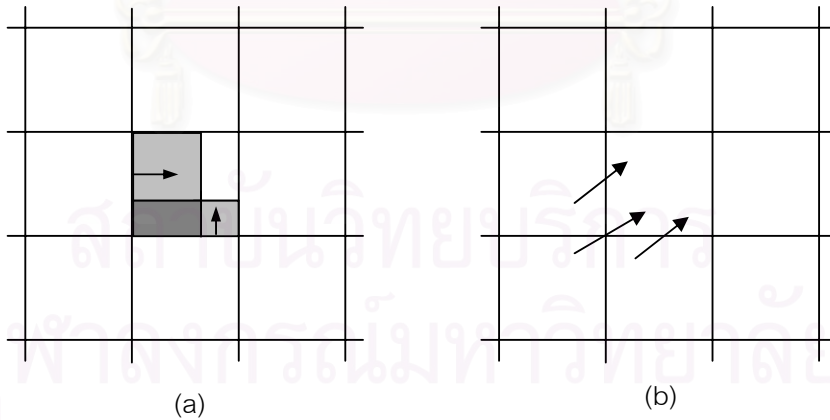


Figure 3.5 (a) Waves moving normal to the cell interfaces. (b) The true velocity  $(u, v)$  is at an angle to the grid, and information from cell  $(i-1, j-1)$  should also affect the new value in cell  $(i, j)$ .

To compute the correction fluxes needed for the CTU method, we view the flow of information as a transversely propagating wave into each of the neighboring cells (see Figure (3.6)). Instead of giving a single expression for each correction flux, we will build up these fluxes by adding in any transverse terms arising from each Riemann problem.

At the beginning of each time step we set  $\tilde{F}_{i-1/2,j} := 0$  and  $\tilde{G}_{i,j-1/2} := 0$  for all  $i$  and  $j$ . After solving each Riemann problem in the  $x$ -direction, we set  $\tilde{\lambda}^\pm \Delta \bar{Q}_{i-1/2,j}$  at the interface as in (3.41) and then update the nearby correction fluxes by

$$\begin{aligned}
 \tilde{G}_{i-1,j-1/2} &:= \tilde{G}_{i-1,j-1/2} - \frac{\Delta t}{2\Delta x} B^- \tilde{\lambda}^- \Delta \bar{Q}_{i-1/2,j} \\
 \tilde{G}_{i-1,j+1/2} &:= \tilde{G}_{i-1,j+1/2} - \frac{\Delta t}{2\Delta x} B^+ \tilde{\lambda}^- \Delta \bar{Q}_{i-1/2,j} \\
 \tilde{G}_{i,j-1/2} &:= \tilde{G}_{i,j-1/2} - \frac{\Delta t}{2\Delta x} B^- \tilde{\lambda}^+ \Delta \bar{Q}_{i-1/2,j} \\
 \tilde{G}_{i,j+1/2} &:= \tilde{G}_{i,j+1/2} - \frac{\Delta t}{2\Delta x} B^+ \tilde{\lambda}^+ \Delta \bar{Q}_{i-1/2,j}.
 \end{aligned} \tag{3.43}$$

We then sweep in the  $y$ -direction, after solving the Riemann problem at interface  $(i, j-1/2)$ . The fluxes are updated by

$$\begin{aligned}
 \tilde{F}_{i-1/2,j-1} &:= \tilde{F}_{i-1/2,j-1} - \frac{\Delta t}{2\Delta x} \tilde{\lambda}^- B^- \Delta \bar{Q}_{i,j-1/2} \\
 \tilde{F}_{i+1/2,j-1} &:= \tilde{F}_{i+1/2,j-1} - \frac{\Delta t}{2\Delta x} \tilde{\lambda}^+ B^- \Delta \bar{Q}_{i,j-1/2} \\
 \tilde{F}_{i-1/2,j} &:= \tilde{F}_{i-1/2,j} - \frac{\Delta t}{2\Delta x} \tilde{\lambda}^- B^+ \Delta \bar{Q}_{i,j-1/2} \\
 \tilde{F}_{i+1/2,j} &:= \tilde{F}_{i+1/2,j} - \frac{\Delta t}{2\Delta x} \tilde{\lambda}^+ B^+ \Delta \bar{Q}_{i,j-1/2}.
 \end{aligned} \tag{3.44}$$

This is the process of CTU method for computing the transverse propagations to update the correction fluxes. Although CTU method is first-order accurate it has better stability than those that exclude the correction fluxes. To achieve second-order accuracy, we use the correction fluxes as follows:

$$\begin{aligned}
\tilde{F}_{i-1/2,j} &:= \tilde{F}_{i-1/2,j} + \frac{1}{2} \sum_{p=1}^m |s_{i-1/2,j}^p| \left( 1 - \frac{\Delta t}{\Delta x} |s_{i-1/2,j}^p| \right) \tilde{W}_{i-1/2,j}^p \\
\tilde{G}_{i,j-1/2} &:= \tilde{G}_{i,j-1/2} + \frac{1}{2} \sum_{p=1}^m |s_{i,j-1/2}^p| \left( 1 - \frac{\Delta t}{\Delta y} |s_{i,j-1/2}^p| \right) \tilde{W}_{i,j-1/2}^p \\
\tilde{F}_{i+1/2,j} &:= \tilde{F}_{i+1/2,j} + \frac{1}{2} \sum_{p=1}^m |s_{i+1/2,j}^p| \left( 1 - \frac{\Delta t}{\Delta x} |s_{i+1/2,j}^p| \right) \tilde{W}_{i+1/2,j}^p \\
\tilde{G}_{i,j+1/2} &:= \tilde{G}_{i,j+1/2} + \frac{1}{2} \sum_{p=1}^m |s_{i,j+1/2}^p| \left( 1 - \frac{\Delta t}{\Delta y} |s_{i,j+1/2}^p| \right) \tilde{W}_{i,j+1/2}^p
\end{aligned} \tag{3.45}$$

where  $\tilde{W}_{i-1/2,j}^p$  represents a limited version of the wave  $\tilde{W}_{i-1/2,j}^p$

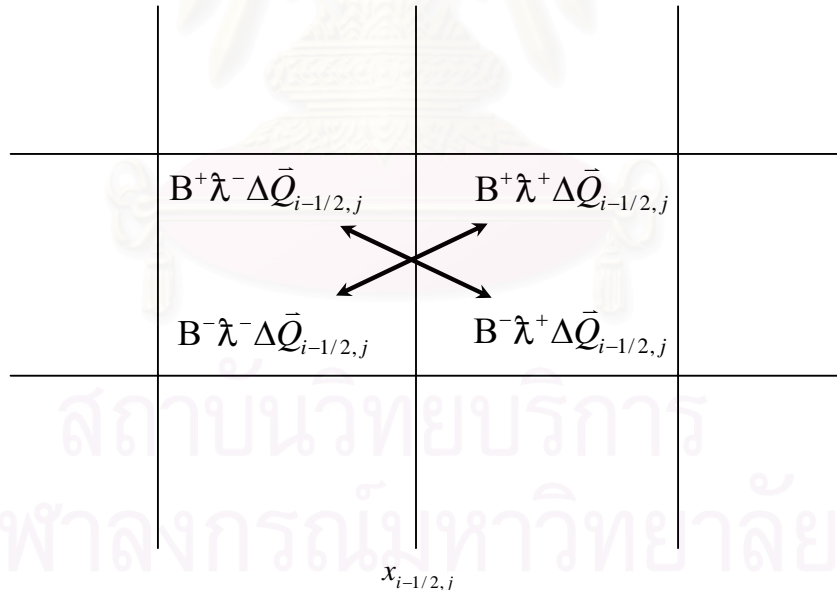


Figure 3.6 Transverse propagations at the interface  $(i-1/2, j)$ .

To calculate the transverse propagations in  $x$ -direction, we split the fluctuation  $\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j}$  into up-going and down-going fluctuations  $\mathbf{B}^+ \tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j}$  and  $\mathbf{B}^- \tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j}$  that modify the fluxes  $\tilde{G}_{i,j+1/2}$  and  $\tilde{G}_{i,j-1/2}$  above and below the cell  $(i, j)$ , respectively. To compute  $\mathbf{B}^\pm \tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j}$ , we decompose the vector  $\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j}$  into eigenvectors of  $\hat{\mathbf{B}}$ ,

$$\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j} = \beta^1 \hat{r}^{y1} + \beta^2 \hat{r}^{y2} + \beta^3 \hat{r}^{y3}$$

$$\begin{bmatrix} (\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j})^1 \\ (\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j})^2 \\ (\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j})^3 \end{bmatrix} = \beta^1 \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} - \hat{c} \end{bmatrix} + \beta^2 \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} + \beta^3 \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} + \hat{c} \end{bmatrix} \quad (3.46)$$

Solving the linear system (3.46) for  $\bar{\beta}$ , we obtain

$$\beta^1 = \frac{-(\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j})^3 + (\hat{v} + \hat{c})(\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j})^1}{2\hat{c}}$$

$$\beta^2 = -(\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j})^2 + \hat{u}(\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j})^1$$

$$\beta^3 = \frac{-(\hat{v} - \hat{c})(\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j})^1 + (\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j})^3}{2\hat{c}}$$

The transverse waves are given by

$$\bar{W}b_{i-1/2,j}^1 = \beta^1 \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} - \hat{c} \end{bmatrix}, \bar{W}b_{i-1/2,j}^2 = \beta^2 \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \bar{W}b_{i-1/2,j}^3 = \beta^3 \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} + \hat{c} \end{bmatrix}$$

and the transverse waves speeds are  $sb_{i-1/2,j}^1 = \hat{v} - \hat{c}$ ,  $sb_{i-1/2,j}^2 = \hat{v}$ , and  $sb_{i-1/2,j}^3 = \hat{v} + \hat{c}$ . The transverse fluctuations are given by

$$\mathbf{B}^- \tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j} = \sum_{p=1}^3 (sb_{i-1/2,j}^p)^- \overline{Wb}_{i-1/2,j}^p$$

$$\mathbf{B}^+ \tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j} = \sum_{p=1}^3 (sb_{i-1/2,j}^p)^+ \overline{Wb}_{i-1/2,j}^p$$

where  $\tilde{\lambda}^* \Delta \bar{Q}_{i-1/2,j}$  represent  $\tilde{\lambda}^+ \Delta \bar{Q}_{i-1/2,j}$  or  $\tilde{\lambda}^- \Delta \bar{Q}_{i-1/2,j}$ .

To calculate the transverse propagations in y-direction, we split the fluctuation  $\mathbf{B}^* \Delta \bar{Q}_{i,j-1/2}$  into right-going and left-going fluctuations  $\tilde{\lambda}^+ \mathbf{B}^* \Delta \bar{Q}_{i,j-1/2}$  and  $\tilde{\lambda}^- \mathbf{B}^* \Delta \bar{Q}_{i,j-1/2}$  that modify the fluxes  $\tilde{F}_{i+1/2,j}$  and  $\tilde{F}_{i-1/2,j}$  right and left the cell  $(i, j)$ , respectively. To compute  $\tilde{\lambda}^\pm \mathbf{B}^* \Delta \bar{Q}_{i,j-1/2}$ , we decompose the vector  $\mathbf{B}^* \Delta \bar{Q}_{i,j-1/2}$  into eigenvectors of  $\hat{A}$ ,

$$\mathbf{B}^* \Delta \bar{Q}_{i,j-1/2} = \beta^1 \hat{r}^{x1} + \beta^2 \hat{r}^{x2} + \beta^3 \tilde{r}^{x3}$$

$$\begin{bmatrix} (\mathbf{B}^* \Delta Q_{i,j-1/2})^1 \\ (\mathbf{B}^* \Delta Q_{i,j-1/2})^2 \\ (\mathbf{B}^* \Delta Q_{i,j-1/2})^3 \end{bmatrix} = \beta^1 \begin{bmatrix} 1 \\ \hat{u} - \hat{c} \\ \hat{v} \end{bmatrix} + \beta^2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \beta^3 \begin{bmatrix} 1 \\ \hat{u} + \hat{c} \\ \hat{v} \end{bmatrix} \quad (3.47)$$

Solving the linear system (3.47) for  $\vec{\beta}$ , we obtain

$$\beta^1 = \frac{-(\mathbf{B}^* \Delta Q_{i,j-1/2})^3 + (\hat{u} + \hat{c})(\mathbf{B}^* \Delta Q_{i,j-1/2})^1}{2\hat{c}}$$

$$\beta^2 = -(\mathbf{B}^* \Delta Q_{i,j-1/2})^2 + \hat{v}(\mathbf{B}^* \Delta Q_{i,j-1/2})^1$$

$$\beta^3 = \frac{-(\hat{u} - \hat{c})(\mathbf{B}^* \Delta Q_{i,j-1/2})^1 + (\mathbf{B}^* \Delta Q_{i,j-1/2})^3}{2\hat{c}}$$

The transverse waves are given by

$$\overline{Wb}_{i,j-1/2}^1 = \beta^1 \begin{bmatrix} 1 \\ \hat{u} - \hat{c} \\ \hat{v} \end{bmatrix}, \overline{Wb}_{i,j-1/2}^2 = \beta^2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \overline{Wb}_{i,j-1/2}^3 = \beta^3 \begin{bmatrix} 1 \\ \hat{u} + \hat{c} \\ \hat{v} \end{bmatrix}$$

and the transverse waves speeds are  $sb_{i,j-1/2}^1 = \hat{u} - \hat{c}$ ,  $sb_{i,j-1/2}^2 = \hat{u}$ , and  $sb_{i,j-1/2}^3 = \hat{u} + \hat{c}$ . The transverse fluctuations are given by

$$\begin{aligned} \lambda^- \mathbf{B}^* \Delta \bar{Q}_{i,j-1/2} &= \sum_{p=1}^3 (sb_{i,j-1/2}^p)^- \overline{Wb}_{i,j-1/2}^p \\ \lambda^+ \mathbf{B}^* \Delta \bar{Q}_{i,j-1/2} &= \sum_{p=1}^3 (sb_{i,j-1/2}^p)^+ \overline{Wb}_{i,j-1/2}^p \end{aligned}$$

where  $\mathbf{B}^* \Delta \bar{Q}_{i-1/2,j}$  represents  $\mathbf{B}^+ \Delta \bar{Q}_{i-1/2,j}$  or  $\mathbf{B}^- \Delta \bar{Q}_{i-1/2,j}$ .

Implementation of the above method is described in an algorithm as follows.

#### Algorithm 2

1. Initialize  $\tilde{\mathbf{F}}_{i-1/2,j} := 0$  and  $\tilde{\mathbf{G}}_{i,j-1/2} := 0$  at each interface.
2. Solve each Riemann problem in the x-direction by sweeping through the grid. At interface  $(i-1/2, j)$  between  $C_{i-1,j}$  and  $C_{i,j-1}$  we use  $\bar{Q}_{i-1,j}$  and  $\bar{Q}_{i,j}$  to compute the waves  $\bar{W}_{i-1/2,j}^p$  and the speeds  $s_{i-1/2,j}^p$ . Calculate fluctuations  $\lambda^+ \Delta \bar{Q}_{i-1/2,j}$  and  $\lambda^- \Delta \bar{Q}_{i-1/2,j}$ .

$$\begin{aligned} \lambda^- \Delta \bar{Q}_{i-1/2,j} &= \sum_{p=1}^3 (s_{i-1/2,j}^p)^- \bar{W}_{i-1/2,j}^p \\ \lambda^+ \Delta \bar{Q}_{i-1/2,j} &= \sum_{p=1}^3 (s_{i-1/2,j}^p)^+ \bar{W}_{i-1/2,j}^p \end{aligned}$$

3. Determine  $\tilde{W}_{i-1/2,j}^p$  and use this to update the correction fluxes at the interface (The Godunov's method is improved to second-order accurate)

$$\tilde{F}_{i-1/2,j} := \tilde{F}_{i-1/2,j} + \frac{1}{2} \sum_{p=1}^3 |s_{i-1/2,j}^p| \left( 1 - \frac{\Delta t}{\Delta x} |s_{i-1/2,j}^p| \right) \tilde{W}_{i-1/2,j}^p$$

4. Use right-going fluctuation  $\hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j}$  to compute an up-going transverse fluctuation  $B^+ \hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j}$  and a down-going transverse fluctuation  $B^- \hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j}$  by solving the transverse Riemann problem. These are computed by decomposing the fluctuation  $\hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j}$  into eigenvectors of B,

$$\hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j} = \sum_{p=1}^3 \beta^p \bar{r}^{yp}$$

and then setting

$$B^\pm \hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j} = \sum_{p=1}^3 (\lambda^{yp})^\pm \beta^p \bar{r}^{yp}$$

5. Update the correction fluxes above and below the cell  $C_{i,j}$  by using the fluctuations  $B^\pm \hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j}$ :

$$\tilde{G}_{i,j+1/2} := \tilde{G}_{i,j+1/2} - \frac{\Delta t}{2\Delta x} B^+ \hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j}$$

$$\tilde{G}_{i,j-1/2} := \tilde{G}_{i,j-1/2} - \frac{\Delta t}{2\Delta x} B^- \hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j}$$

6. In a similar manner, the left-going fluctuation  $\hat{\lambda}^- \Delta \bar{Q}_{i-1/2,j}$  is separated into transverse fluctuations  $\mathbf{B}^\pm \hat{\lambda}^- \Delta \bar{Q}_{i-1/2,j}$ , which are then used to update the fluxes above and below cell  $C_{i-1,j}$  :

$$\begin{aligned}\tilde{\bar{G}}_{i-1,j+1/2} &:= \tilde{\bar{G}}_{i-1,j+1/2} - \frac{\Delta t}{2\Delta x} \mathbf{B}^+ \hat{\lambda}^- \Delta \bar{Q}_{i-1/2,j} \\ \tilde{\bar{G}}_{i-1,j-1/2} &:= \tilde{\bar{G}}_{i-1,j-1/2} - \frac{\Delta t}{2\Delta x} \mathbf{B}^- \hat{\lambda}^- \Delta \bar{Q}_{i-1/2,j}\end{aligned}$$

7. Steps 2-6 are repeated for each Riemann problem in the y-direction, at interfaces between cell  $C_{i,j-1}$  and  $C_{i,j}$ . The resulting waves  $\bar{W}_{i,j-1/2}^p$  are limited by comparisons in the y-direction and used to update  $\tilde{\bar{G}}_{i,j-1/2}$ . In solving these Riemann problems we also compute fluctuations  $\mathbf{B}^\pm \Delta \bar{Q}_{i,j-1/2}$  which are then separated transversely into  $\hat{\lambda}^\pm \mathbf{B}^+ \Delta \bar{Q}_{i,j-1/2}$  and  $\hat{\lambda}^\pm \mathbf{B}^- \Delta \bar{Q}_{i,j-1/2}$ . These four transverse fluctuations are used to modify the nearby  $\tilde{\bar{F}}$  fluxes.

8. Finally, the updating formula (3.42).

$$\begin{aligned}\bar{Q}_{ij}^{n+1} &= \bar{Q}_{ij}^n - \frac{\Delta t}{\Delta x} (\hat{\lambda}^+ \Delta \bar{Q}_{i-1/2,j} + \hat{\lambda}^- \Delta \bar{Q}_{i+1/2,j}) \\ &\quad - \frac{\Delta t}{\Delta y} (\mathbf{B}^+ \Delta \bar{Q}_{i,j-1/2} + \mathbf{B}^- \Delta \bar{Q}_{i,j+1/2}) \\ &\quad - \frac{\Delta t}{\Delta x} (\tilde{\bar{F}}_{i+1/2,j} - \tilde{\bar{F}}_{i-1/2,j}) - \frac{\Delta t}{\Delta y} (\tilde{\bar{G}}_{i,j+1/2} - \tilde{\bar{G}}_{i,j-1/2})\end{aligned}$$

#### 3.5.4 High-resolution Method with HLL Approximate Riemann Solver

The HLL method can easily be improved to the second-order accuracy by using the MUSCL procedure. In this thesis, the reconstruction of  $\bar{Q}_l$  and  $\bar{Q}_r$  are given by



At the interface  $(i-1/2, j)$

$$\begin{aligned}\bar{Q}_l &\equiv \bar{Q}'_{i-1/2,j} = \bar{Q}_{i-1,j} + \frac{1}{2} \bar{\delta}_{i-1,j} \Delta x \\ \bar{Q}_r &\equiv \bar{Q}^r_{i-1/2,j} = \bar{Q}_{i,j} - \frac{1}{2} \bar{\delta}_{i,j} \Delta x\end{aligned}\tag{3.48}$$

At the interface  $(i, j-1/2)$

$$\begin{aligned}\bar{Q}_l &\equiv \bar{Q}'_{i,j-1/2} = \bar{Q}_{i,j-1} + \frac{1}{2} \bar{\delta}_{i,j-1} \Delta y \\ \bar{Q}_r &\equiv \bar{Q}^r_{i,j-1/2} = \bar{Q}_{i,j} - \frac{1}{2} \bar{\delta}_{i,j} \Delta y\end{aligned}\tag{3.49}$$

After the reconstruction of  $\bar{Q}_l$  and  $\bar{Q}_r$ , these values are used to update the numerical fluxes in (3.8). The high-resolution HLL approximate Riemann solver can be described by the following algorithm.

**Algorithm 3**

1. Sweep through grids in the x-direction. At the cell interface  $(i-1/2, j)$ , we compute a piecewise linear function

$$\begin{aligned}\bar{Q}_l &\equiv \bar{Q}'_{i-1/2,j} = \bar{Q}_{i-1,j} + \frac{1}{2} \bar{\delta}_{i-1,j} \Delta x \\ \bar{Q}_r &\equiv \bar{Q}^r_{i-1/2,j} = \bar{Q}_{i,j} - \frac{1}{2} \bar{\delta}_{i,j} \Delta x\end{aligned}$$

2. Compute the numerical fluxes

$$\bar{F}^n_{i-1/2,j}(\bar{Q}_l, \bar{Q}_r) = \begin{cases} \bar{F}(\bar{Q}_l) & \text{when } 0 < s_l \\ \bar{F}^*(\bar{Q}_l, \bar{Q}_r) & \text{when } s_l < 0 < s_r \\ \bar{F}(\bar{Q}_r) & \text{when } s_r < 0. \end{cases}$$

$$\text{Here } \bar{F}^*(\bar{Q}_l, \bar{Q}_r) = \frac{s_r \bar{F}(\bar{Q}_l) - s_l \bar{F}(\bar{Q}_r) \cdot n + s_l s_r (\bar{Q}_r - \bar{Q}_l)}{s_r - s_l}$$

The wave speeds  $s_l$  and  $s_r$  can be determined by

$$s_l = \min(u_{i-1/2,j} - \sqrt{gh_l}, u^* - \sqrt{gh^*})$$

$$s_r = \max(u_{i-1/2,j} + \sqrt{gh_r}, u^* + \sqrt{gh^*})$$

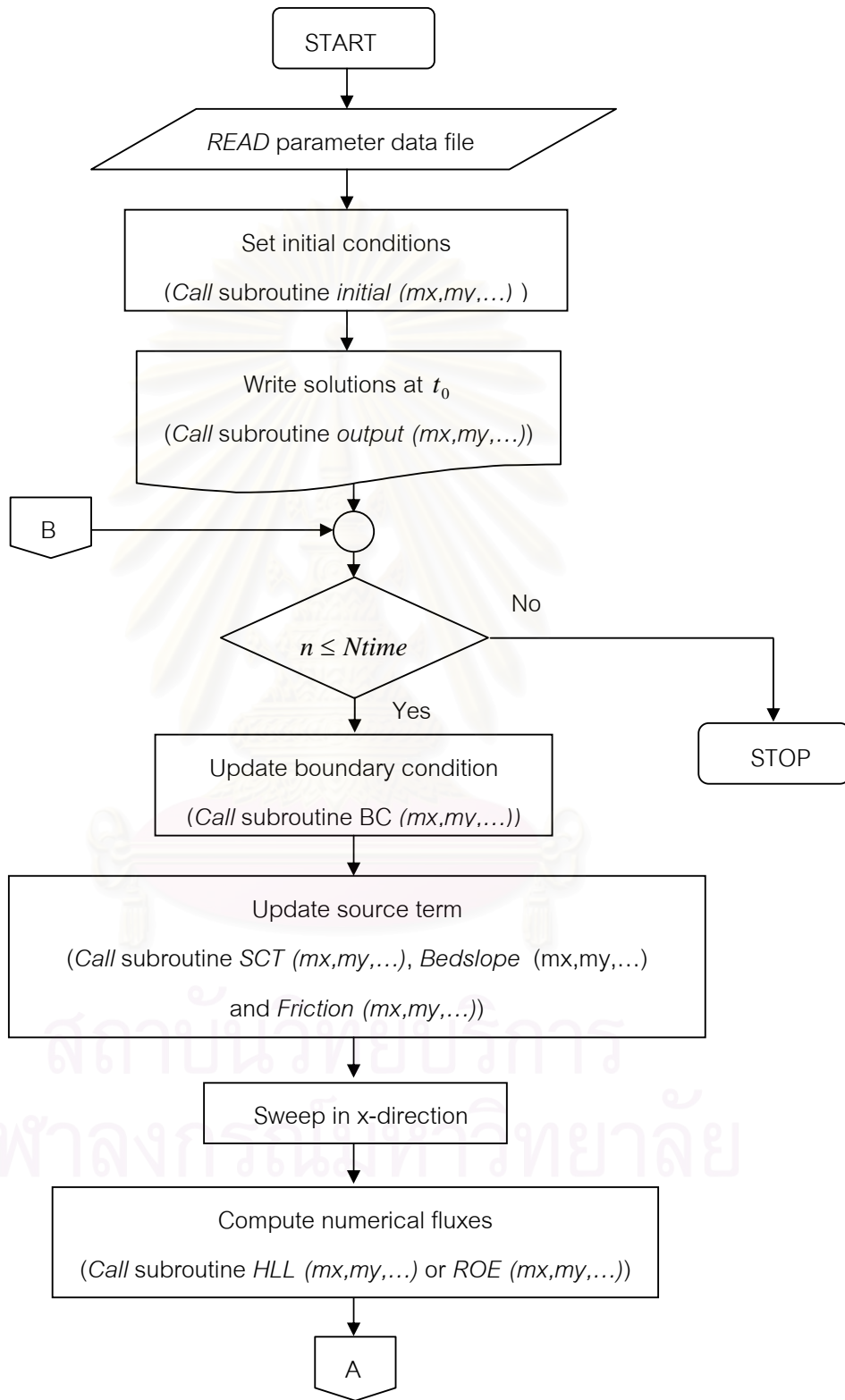
where

$$u^* = \frac{1}{2}(w_l + w_r) \cdot \bar{n} + \sqrt{gh_l} - \sqrt{gh_r}$$

$$\sqrt{gh^*} = \frac{1}{2}(\sqrt{gh_l} + \sqrt{gh_r}) + \frac{1}{4}(w_l - w_r) \cdot \bar{n}$$

3. Use similar procedure to compute the numerical fluxes at the interface  $(i+1/2, j)$ .
4. Sweep through grids in the y-direction and use similar processes 1-2 to compute the numerical fluxes at the interfaces  $(i, j-1/2)$  and  $(i, j+1/2)$ .
5. Finally, update the cell average at  $(i, j)$

$$\begin{aligned} \bar{Q}_{i,j}^{n+1} = \bar{Q}_{i,j}^n &- \frac{\Delta t}{\Delta x} \left[ \bar{F}_{i+1/2,j}^n(\bar{Q}_l, \bar{Q}_r) - \bar{F}_{i-1/2,j}^n(\bar{Q}_l, \bar{Q}_r) \right] \\ &- \frac{\Delta t}{\Delta y} \left[ \bar{G}_{i,j+1/2}^n(\bar{Q}_l, \bar{Q}_r) - \bar{G}_{i,j-1/2}^n(\bar{Q}_l, \bar{Q}_r) \right] \end{aligned}$$

**Flowchart of the Numerical Method**

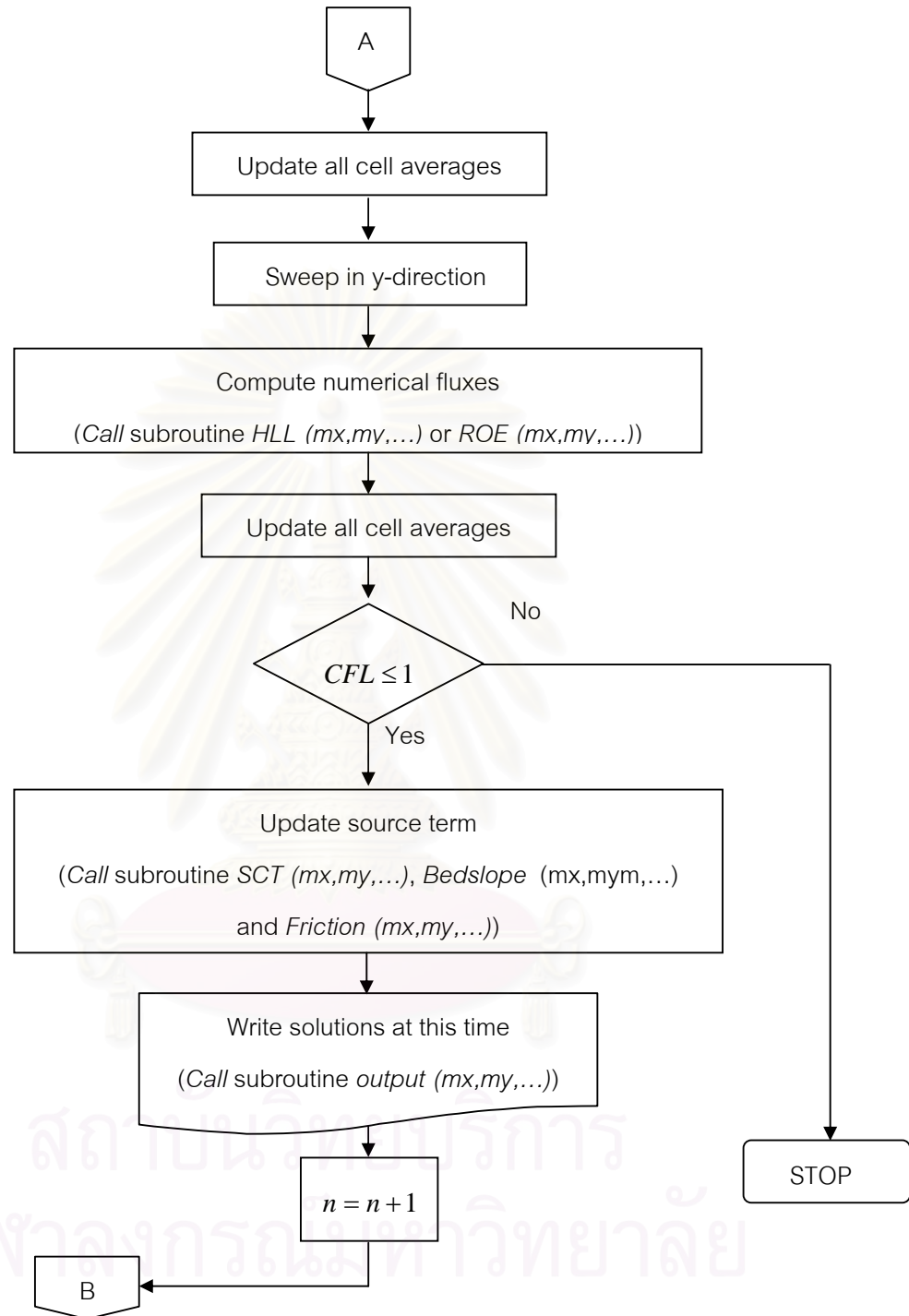


Figure 3.7 Flowchart of numerical method.

## CHAPTER IV

### PARALLEL COMPUTATION

The computational domain of the shallow water problem tends to be quite large, with complex geometries. On a single processor the calculation of its solution would take several hundreds of computational time and would have not enough a buffer zone for stored data variables. The development of parallel computer can overcome the limitations of the problem size and space resolution for shallow water problem simulation. To implement the parallel computation, the domain decomposition techniques are used to divide the physical domain into a number of smaller domains, each of which corresponds to a processor. A message passing interface (MPI) is incorporated for inter-processor data communication.

#### 4.1 Parallel Performance Measurements

We define the execution time of a parallel program as the time that elapses from when the first processor starts execution. On each processor, the execution time is spent on computing, communication and idling.

The performance parameters that are commonly used for measuring the gain of using a parallel code are the speedup ( $S$ ) and efficiency ( $E$ ). The speedup  $S$  of a parallel system is defined as the ratio of the execution time  $T_1$  of a given application run on a single processor to the time  $T_p$  taken to solve the same problem on  $P$  processors.

$$S = \frac{T_1}{T_p} \quad (4.1)$$

The efficiency is a measure of the time the processor spent in the computational phase. It is defined as the ratio of speedup  $S$  to the number of processors:

$$E = \frac{S}{P} \quad (4.2)$$

In principle, we would expect  $S = P$  and  $E = 1$ . In practice, a parallel system containing  $P$  processors does not usually achieve a speedup of  $P$ , because part of time required by the processors is spent on communication or idling. That is usually  $0 < S < P$  and  $0 < E < 1$ . Figure 4.1 shows a typical relationship between speedup and the number of processors used.

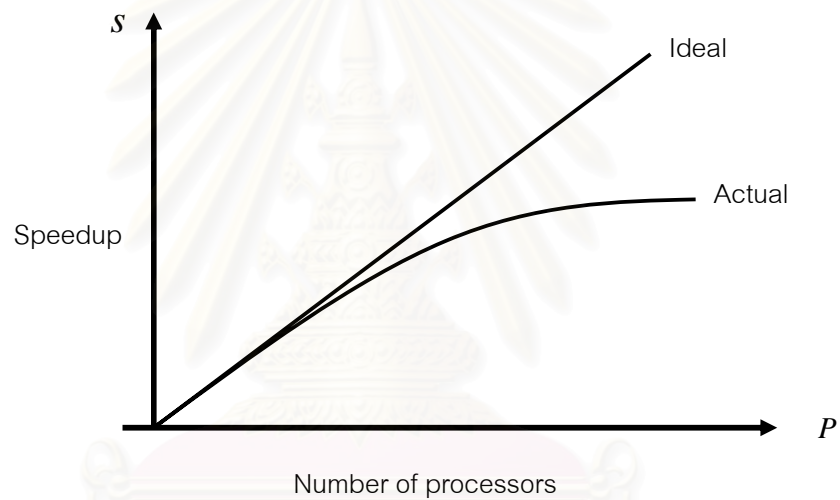


Figure 4.1 Plot of speedup  $S$  versus number of processors  $P$  for a fixed problem size.

Figure 4.1 shows that the speedup does not increase linearly with the number of processors, instead, it tends to saturate. In other words, the efficiency drops as the number of processors increase. This is often referred to as Amdahl's law.

#### 4.2 Domain Decomposition

A domain decomposition scheme is applied for the shallow water domain that is divided into smaller sub-domains. Each sub-domain is defined as the local domain of an individual processor which is addressed to as domain

decomposition. Several strategies exist within the domain decomposition paradigm for dividing the physical domain into sub-domains.

Figure 4.2 shows a 1D and 2D decompositions for a two-dimensional domain. The 1D decomposition can be achieved by dividing the computational domain in one direction only, while in the 2D decomposition, the computational domain is decomposed both in  $x$  and  $y$  coordinate directions.

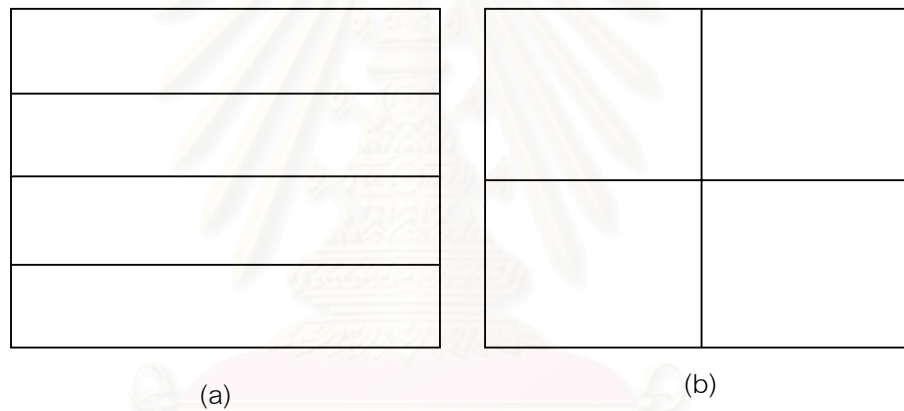


Figure 4.2 Domain decomposition topologies for 1D (a) 2D (b) decompositions.

In this thesis, the 2D decomposition is chosen in the parallel solver for the shallow water equations. In the 2D decomposition, the physical domain is divided into rectangular sub-domains that are distributed across the available processors. Let  $m_x$  and  $m_y$  be the number of grid cells in the  $x$  and  $y$  directions of the physical domain. The total number of processors is  $P = P_x \times P_y$ , where  $P_x$  and  $P_y$  are the numbers of processors assigned to the  $x$  and  $y$  directions respectively. So the number of grid cells in each sub-domain is  $\frac{m_x}{P_x} \cdot \frac{m_y}{P_y}$ . In this work  $\frac{m_x}{P_x}$  and  $\frac{m_y}{P_y}$  are integers.

Table 4.1 Text of serial code for 2D decomposition

Serial code
<pre> do <math>i = 1, m_x</math>   do <math>j = 1, m_y</math>     -- code segment --   end do end do </pre>

Table 4.2 Text of parallel code for 2D decomposition

Parallel code
<pre> do <math>i = 1, \frac{m_x}{P_x}</math>   do <math>j = 1, \frac{m_y}{P_y}</math>     -- code segment --   end do end do </pre>

Consider a loop in the serial code that performs a chunk of computations for all the interior nodes as shown in Table 4.1. In the parallel code, this loop is decomposed into blocks divided by the number of processors (see Table 4.2), with one block assigned to each processor. Hence the loops in the parallel code are smaller than the loops in the serial code and the computational in the parallel code is less than in the serial code.

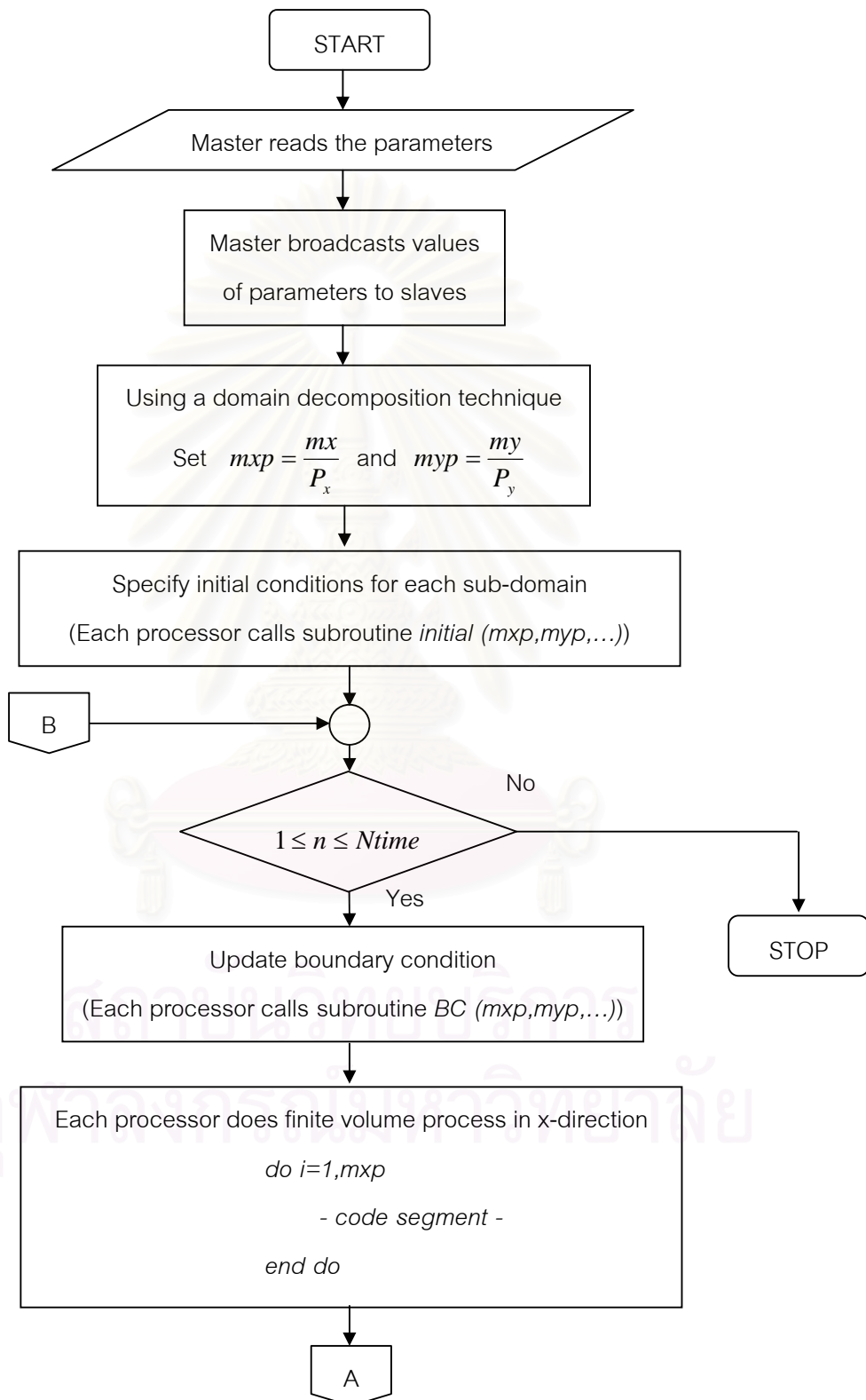


Thus at the end of each time step, each processor must update the boundaries of the sub-domains because of the spatial discretization by copying the appropriate values from adjacent processors. The copying procedure will be in the form of messages exchanged among processors by using MPI.

#### 4.3 Message Passing Interface (MPI)

MPI was developed in 1993-1994 by a group of researchers from industry, government, and academia. As such, it is one of the first standards for programming parallel processors, and it is the first that is based on a message passing. MPI's goal is to provide a standard for writing message-passing programs. This standard defines the syntax and semantics of a core of library routines useful for writing parallel programs. The standard includes descriptions of various types of communication including processor to processor or point to point message passing and global calls for collective communication. The standard also allows process groups and communication contexts to be defined by the user. This enables a single program to control several groups of processors, each working on different tasks. There is also supported for process topologies that is a mapping of processes in a communicator to an addressing scheme. The addressing scheme is usually chosen for convenience. More details of MPI can be found in [13].

Flow chart of the parallel solver



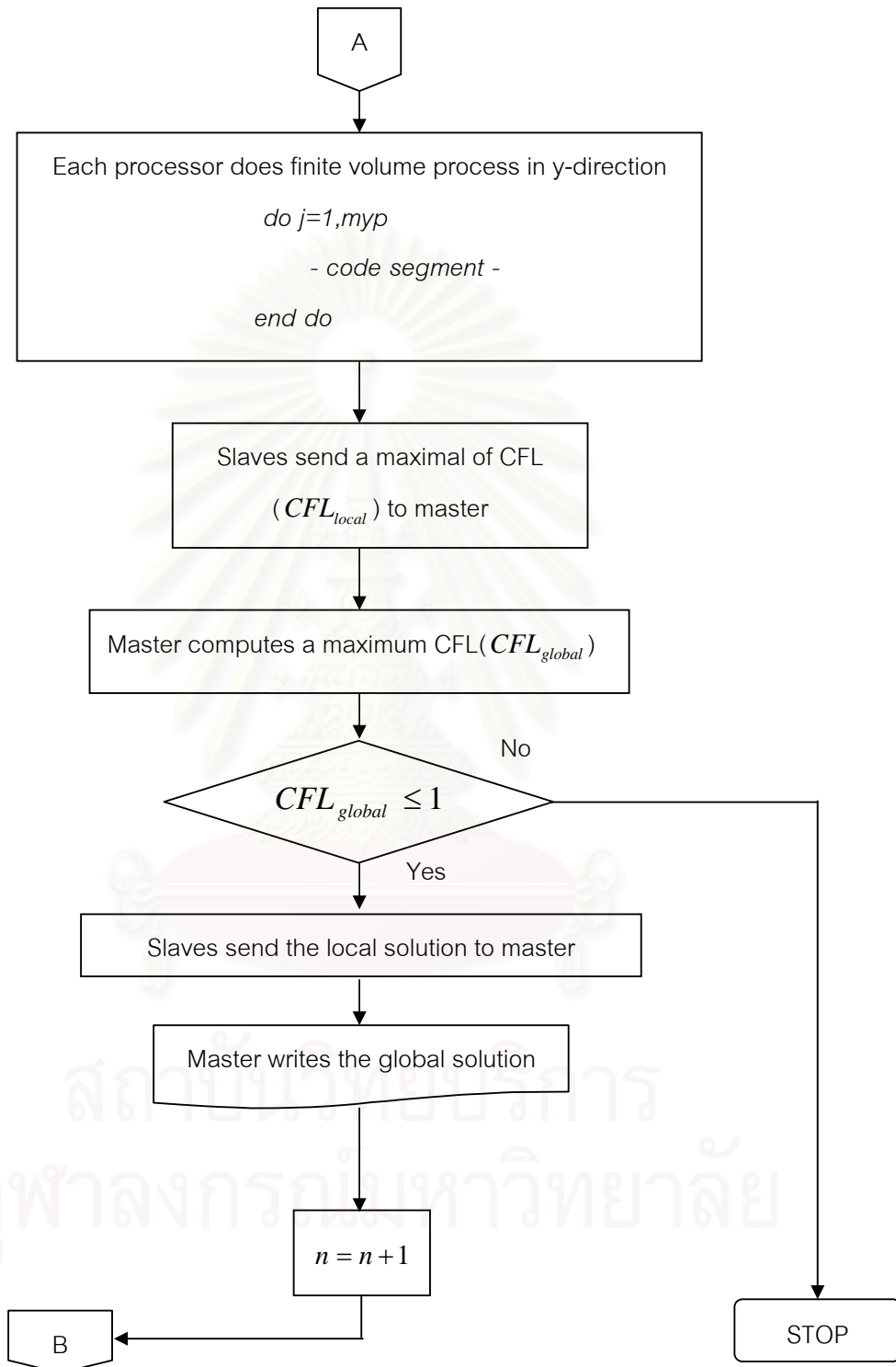


Figure 4.3 Flowchart of parallel solver

## CHAPTER V

### RESULTS

In this chapter, the numerical solution and the performance results of the parallel computation of the circular dam break and the rectangular dam break are reported in the following sections.

#### 5.1 Circular Dam Break

The space of the circular dam break is a  $200m$  long square with a cylindrical dam with radius  $50m$  and centered in the square, as shown in Figure 5.1. In this thesis, we consider two cases: the wet bed and the dry bed. For both cases, the computational domain is divided into  $200 \times 200$  square cells and the numerical solution is computed at time  $t = 5s$  using  $\Delta t = 0.025s$ .

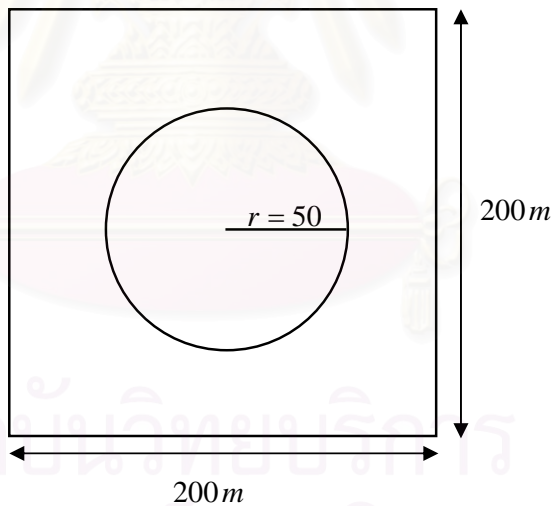


Figure 5.1 Geometry of circular dam break problem.

##### 5.1.1 Wet bed

In wet bed case, the initial water height is  $10m$  inside the dam and  $5m$  outside the dam. Both components of the velocities  $u$  and  $v$  are set to zero everywhere.

Typical profiles of surface elevation of Roe solver are shown in Figure 5.6 and the performances of the parallel computation are shown in Table 5.1.

Table 5.1 Performance results of circular dam break with wet bed.

Number of processors	Unknowns per processor	Elapsed time(s)	Speedup	Efficiency(%)
1	120000	120.502	1.000	100.000
2	60000	64.883	1.857	92.861
4	30000	37.952	3.175	79.378
8	15000	24.594	4.900	61.246

Figure 5.2 shows a relationship between the speedup and the number of processors. Figure 5.3 shows a relationship between the efficiency and the number of processors. We can see that they drop when increasing the number of processors.

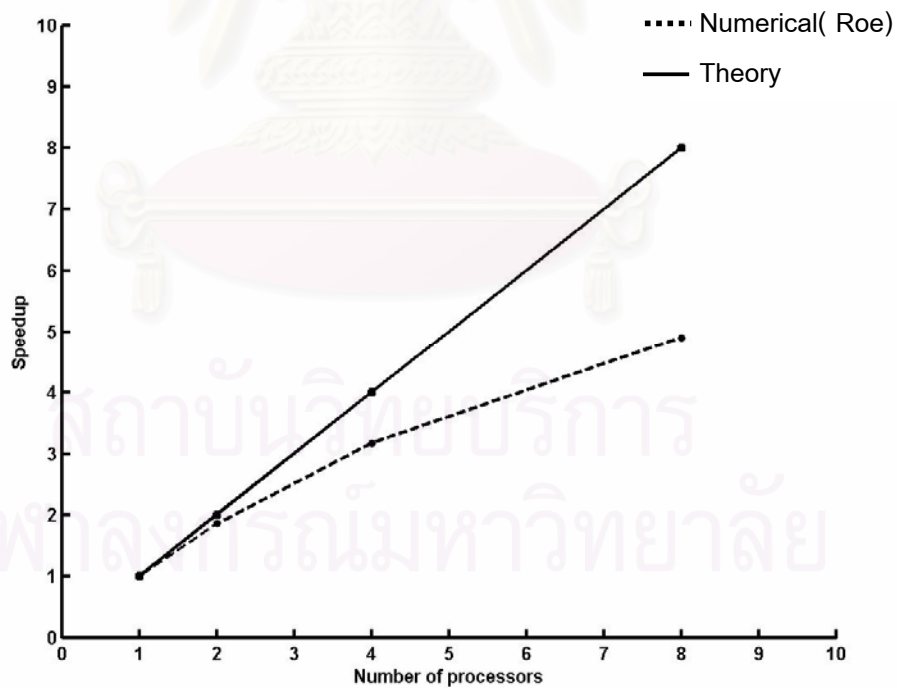


Figure 5.2 Relationship between the speedup and the number of processors in the cluster.

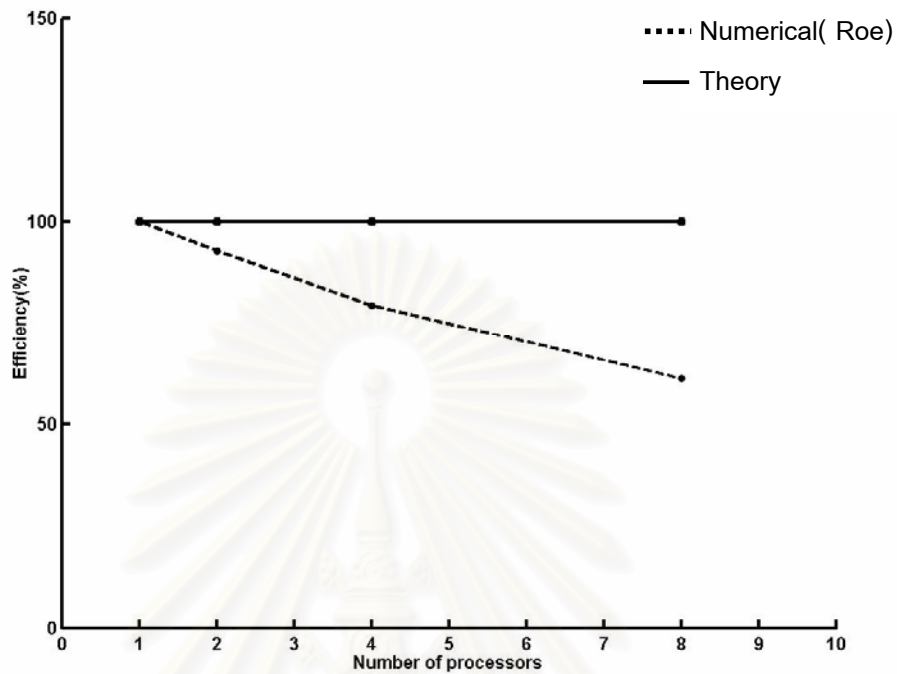


Figure 5.3 Relationship between the efficiency and the number of processors in the cluster.

### 5.1.2 Dry bed

In dry bed case, the initial water height is  $10m$  inside the dam and  $0m$  outside the dam. Both components of the velocities  $u$  and  $v$  are set to zero everywhere. We use only the HLL solver to solve this case. Typical profiles of surface elevation of HLL solver are shown in Figure 5.7 and the performances of the parallel computation are shown in Table 5.2.

Table 5.2 Performance results of circular dam break with dry bed.

Number of processors	Unknowns per processor	Elapsed time(s)	Speedup	Efficiency(%)
1	120000	49.070	1.000	100.000
2	60000	29.565	1.660	82.987
4	30000	20.388	2.407	60.170
8	15000	16.472	2.979	37.237

Figure 5.4 shows a relationship between the speedup and the number of processors. Figure 5.5 shows a relationship between the efficiency and the number of processors. We can see that they drop when increasing the number of processors.

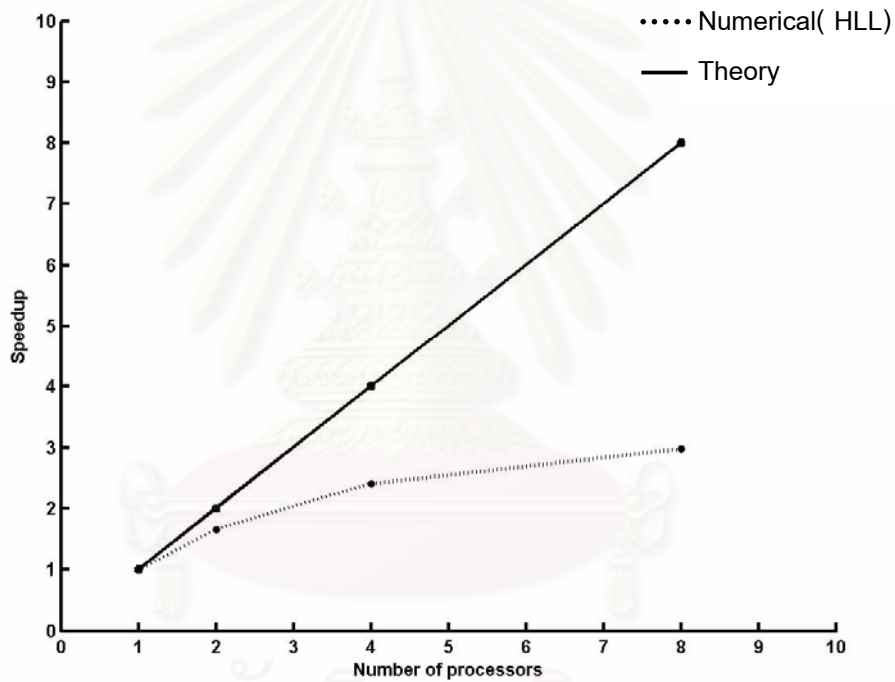


Figure 5.4 Relationship between the speedup and the number of processors in the cluster.

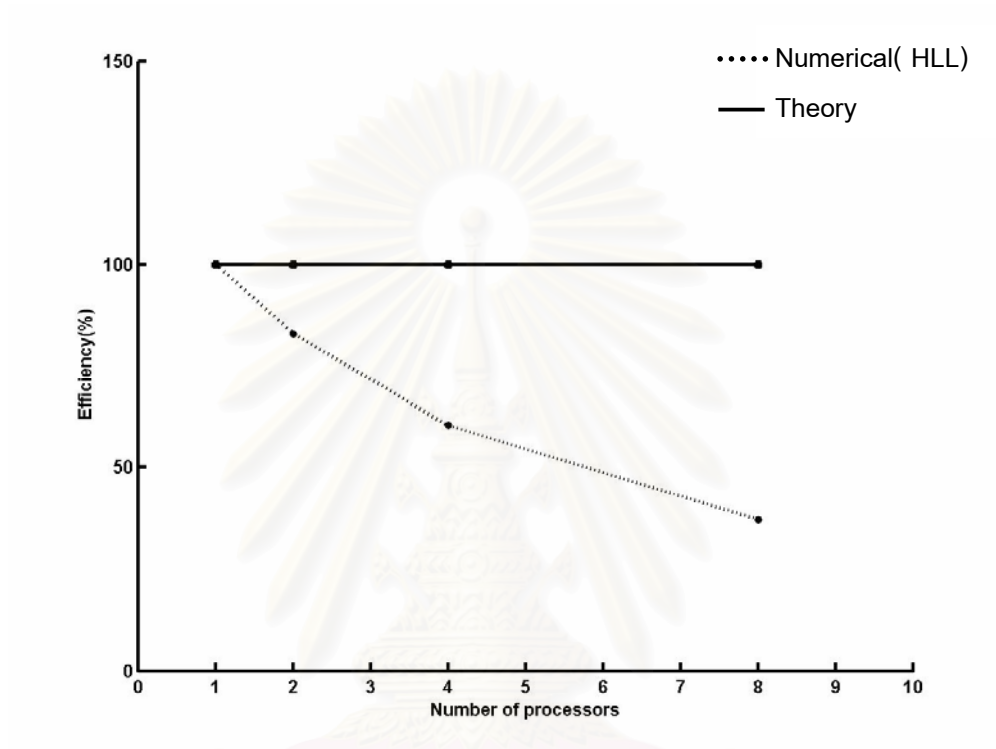
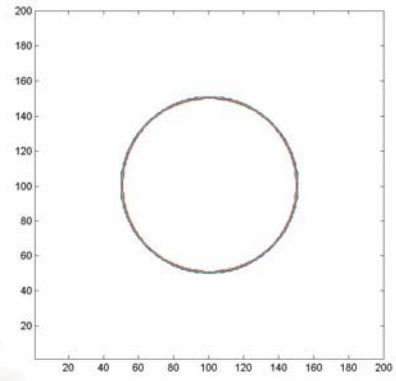
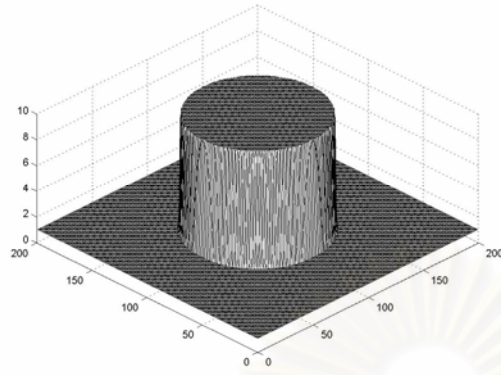
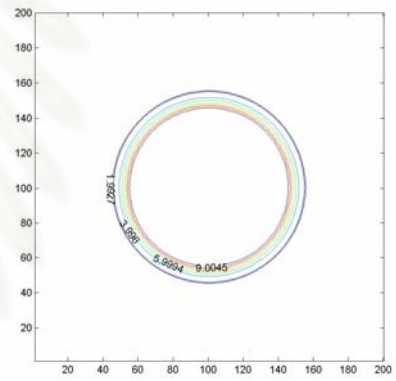
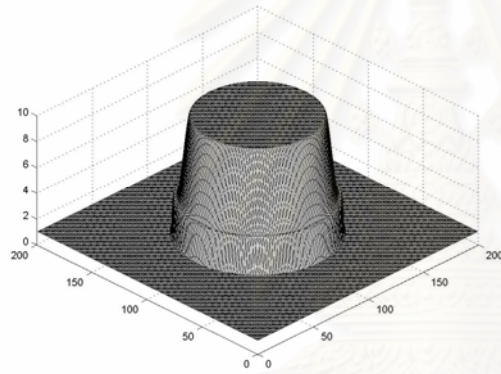
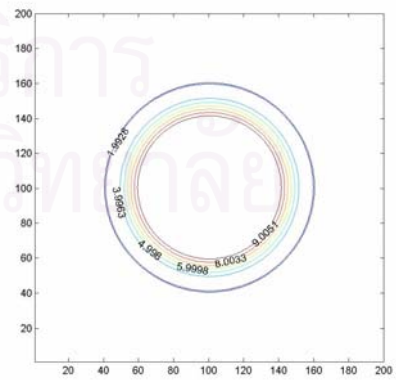
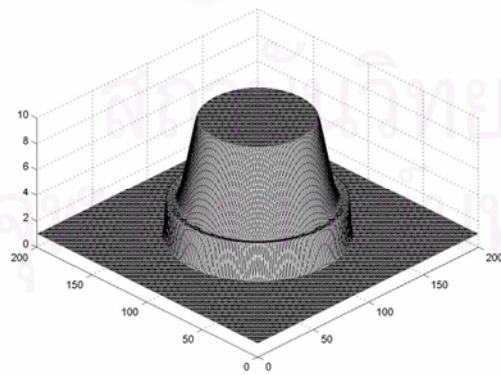
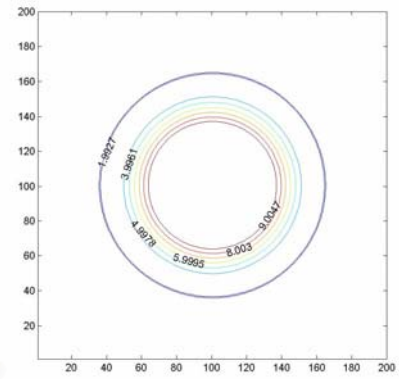
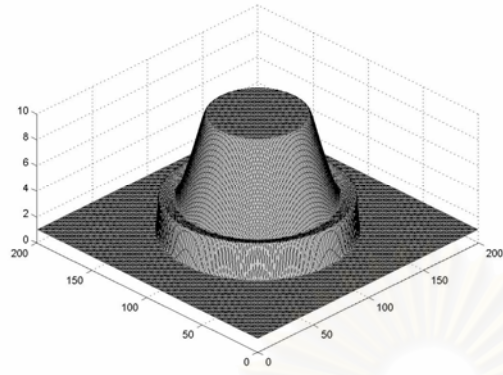
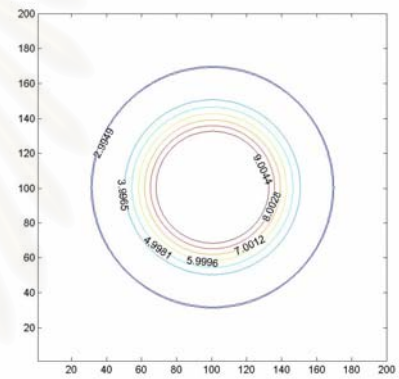
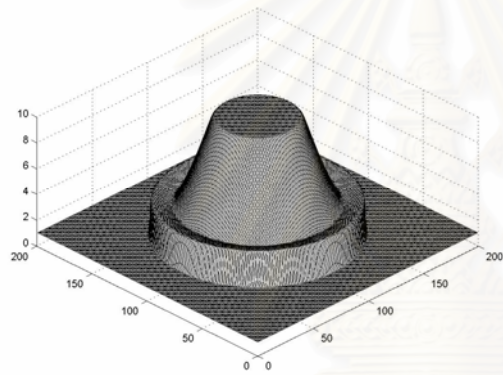
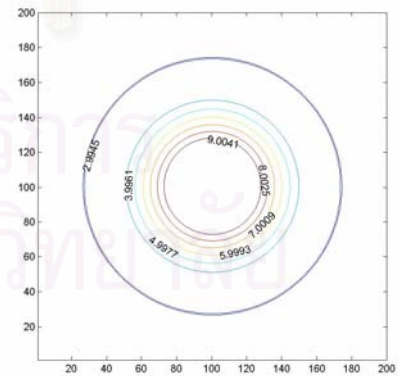
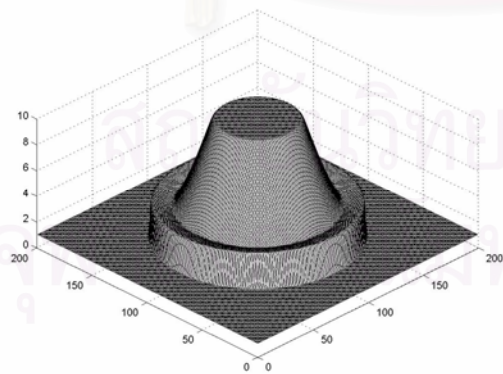


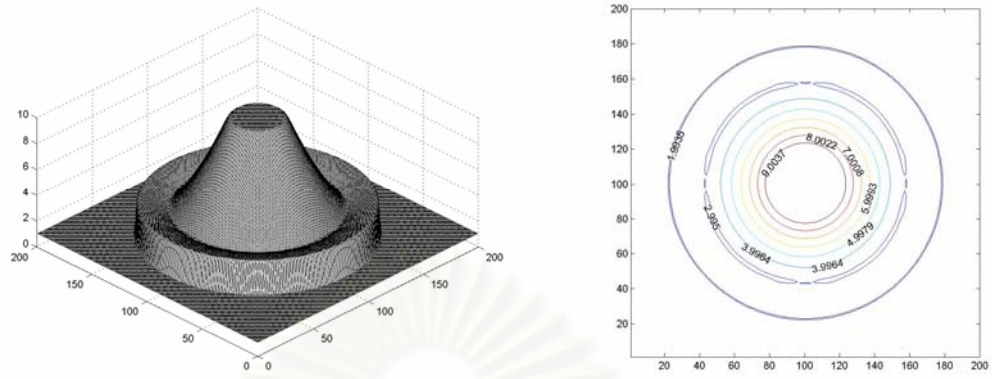
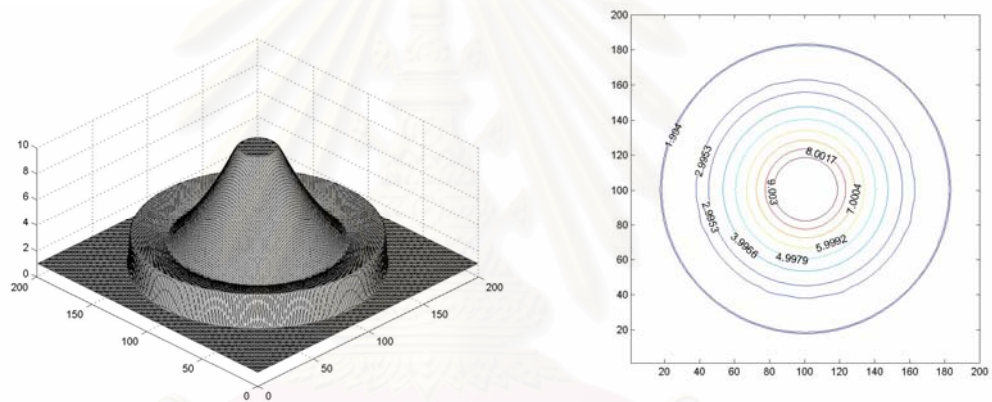
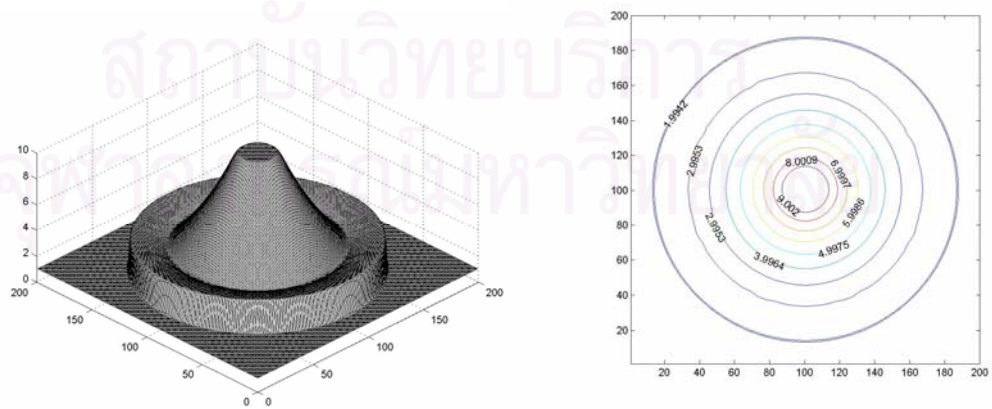
Figure 5.5 Relationship between the efficiency and the number of processors in the cluster.

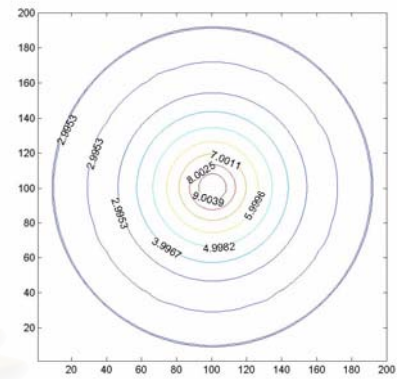
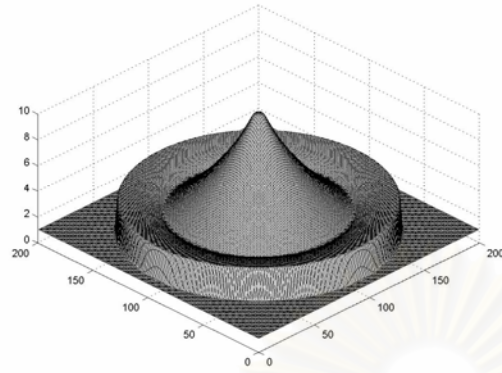
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



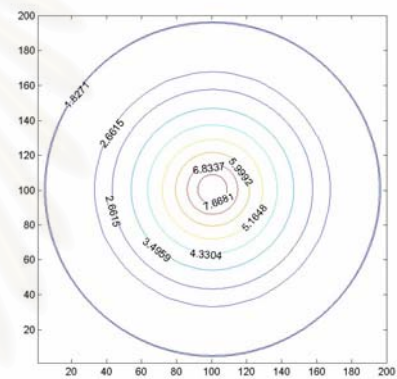
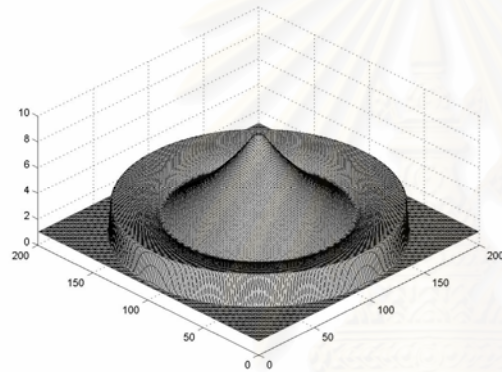
(a1)  $t=0$ (a2)  $t=0.5$ (a3)  $t=1$

(a4)  $t=1.5$ (a5)  $t=2$ (a6)  $t=2.5$

(a7)  $t=3$ (a8)  $t=3.5$ (a9)  $t=4$



(a10) t=4.5

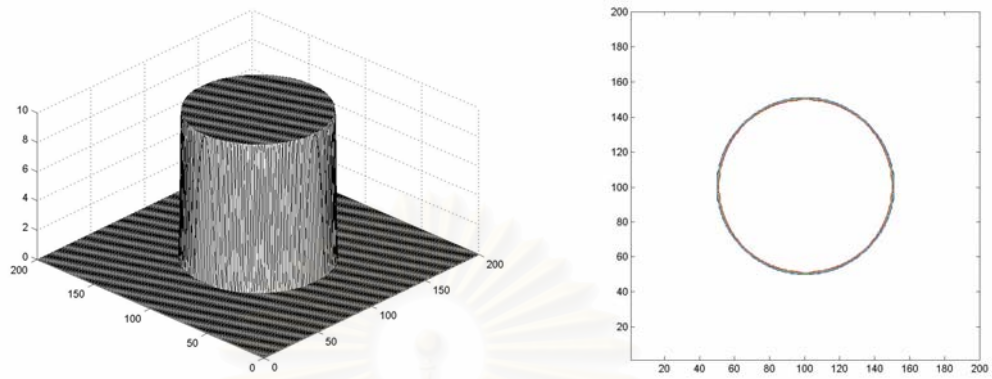
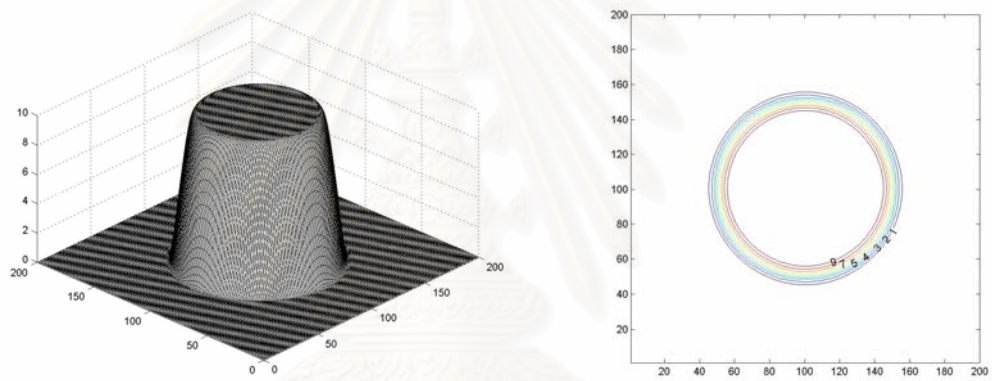
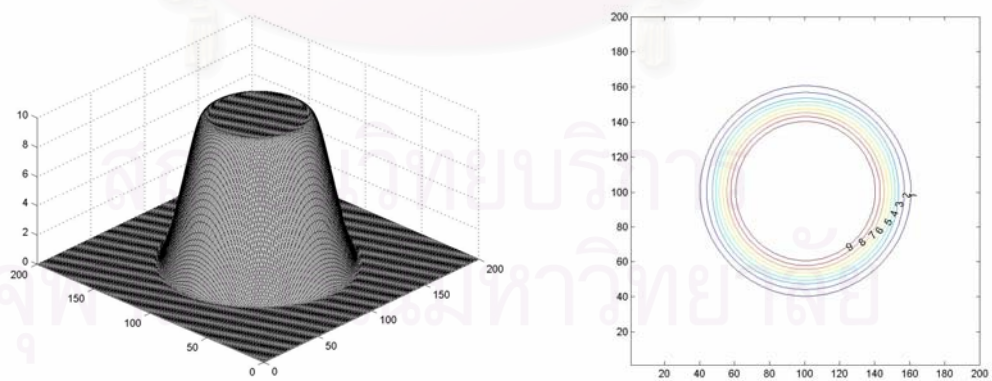


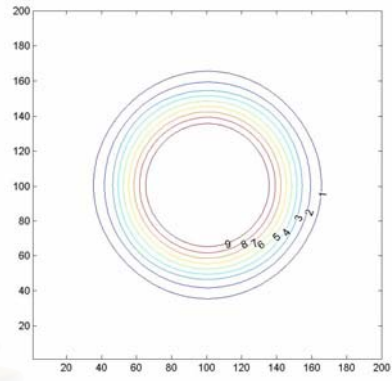
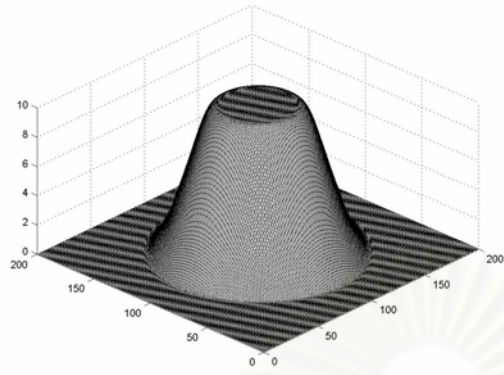
(a11) t=5

Figure 5.6 (a1)-(a11) show the circular dam break results in the case of wet bed, the left of figure shows water profile and the right of figure shows a contour plot of the depth.

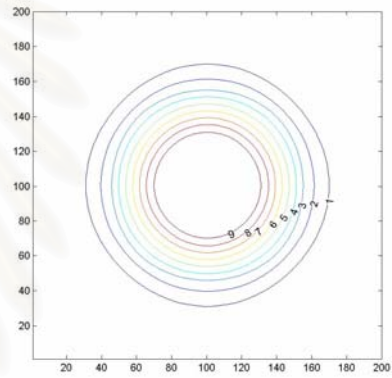
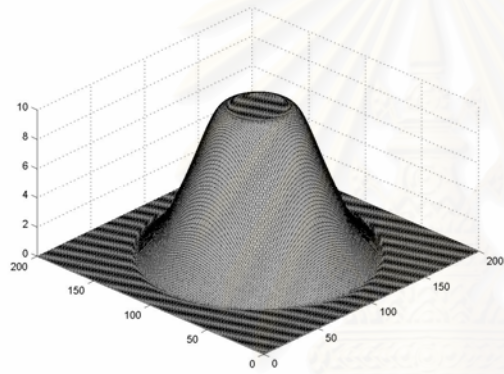
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



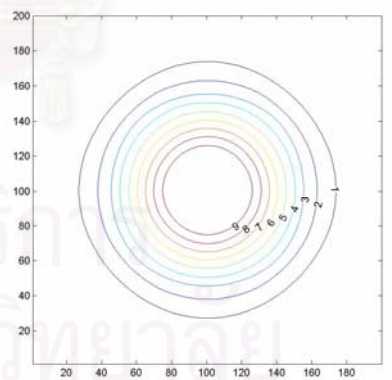
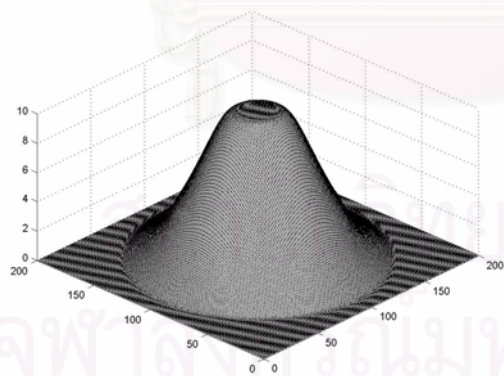
(b1)  $t=0$ (b2)  $t=0.5$ (b3)  $t=1$



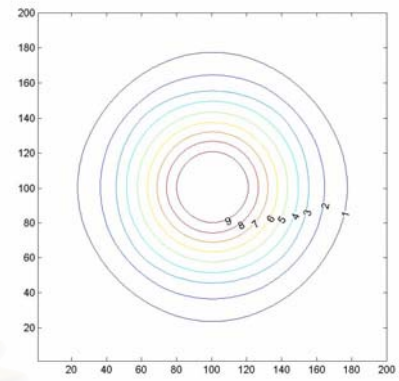
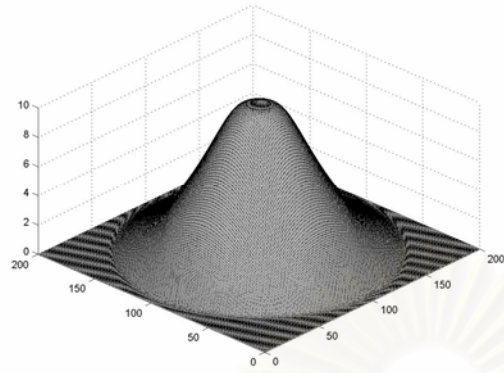
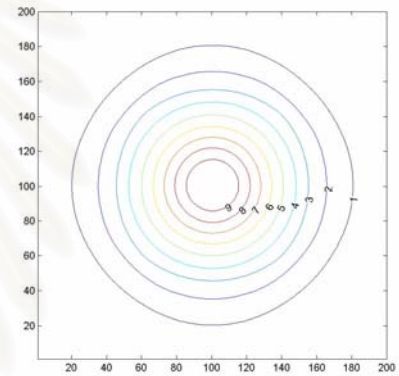
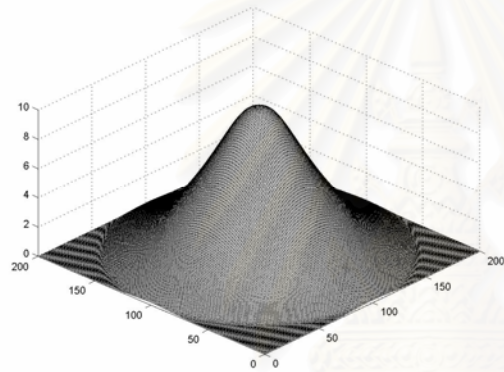
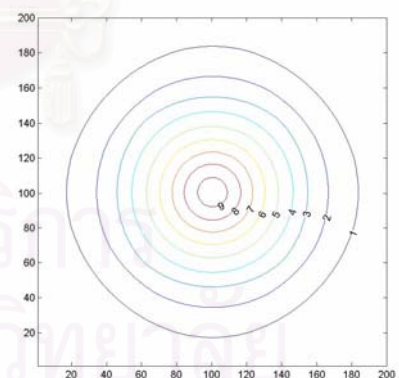
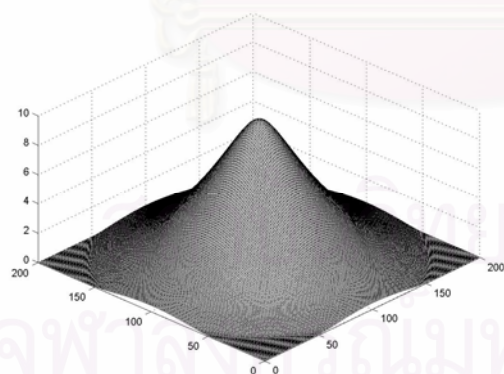
(b4)  $t=1.5$



(b5)  $t=2$



(b6)  $t=2.5$

(b7)  $t=3$ (b8)  $t=3.5$ (b9)  $t=4$

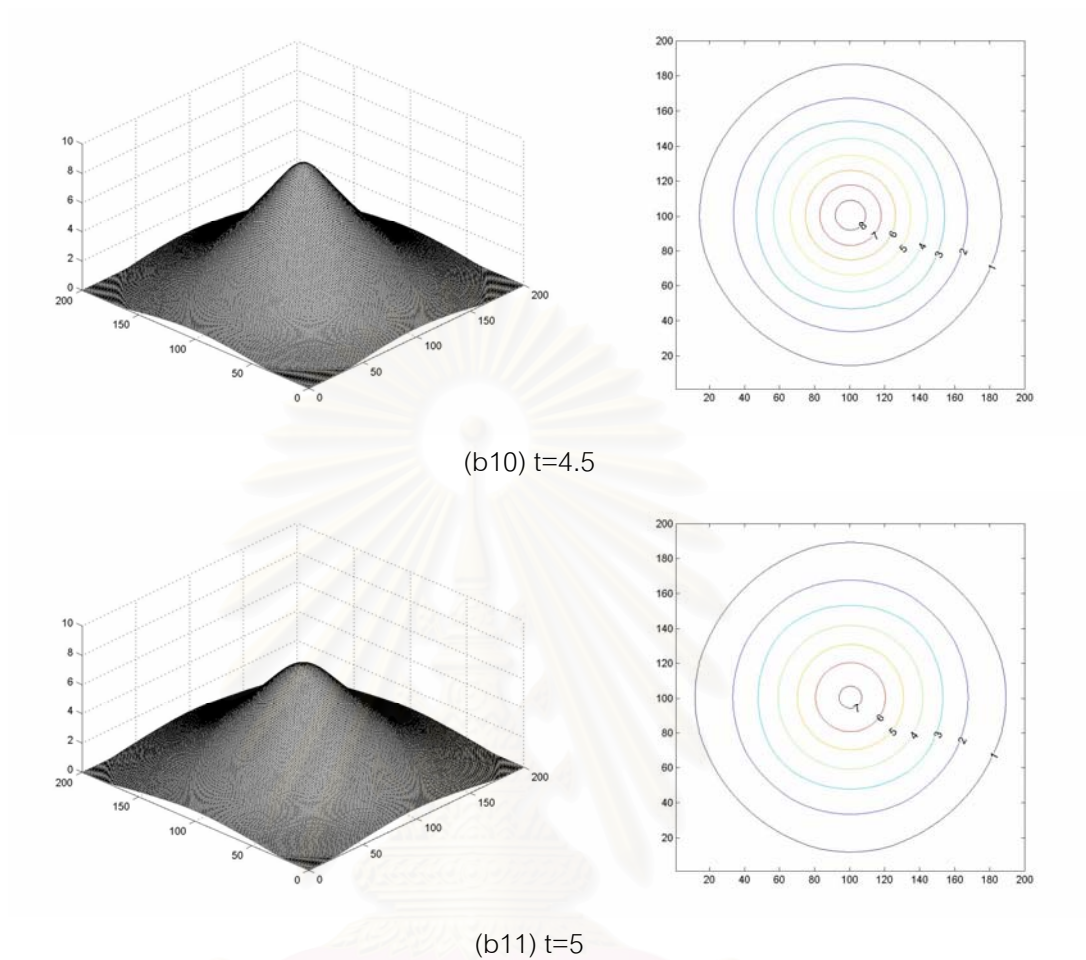


Figure 5.7 (b1)-(b11) show the circular dam break results in the case of dry bed, the left of figure shows water profile and the right of figure shows a contour plot of the depth.

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



## 5.2 Rectangular Dam Break

We consider the dam-break problem in a square domain  $200\text{ m} \times 200\text{ m}$ . The domain is separated into two zones by an infinitesimally thin wall. The breach is  $75\text{ m}$  wide, as shown in Figure 5.8. Again we consider two cases: the wet bed and the dry bed. For both cases, the computational domain is divided into  $200 \times 200$  square cells and the numerical solution is computed at time  $t = 5\text{ s}$  using  $\Delta t = 0.025\text{ s}$ . The rectangular dam break is a special problem and it has more complex computation than a circular dam break or other problems. We present the appropriate domain decomposition technique for using 2,4,6 and 8 processors as shown in Figure 5.9.

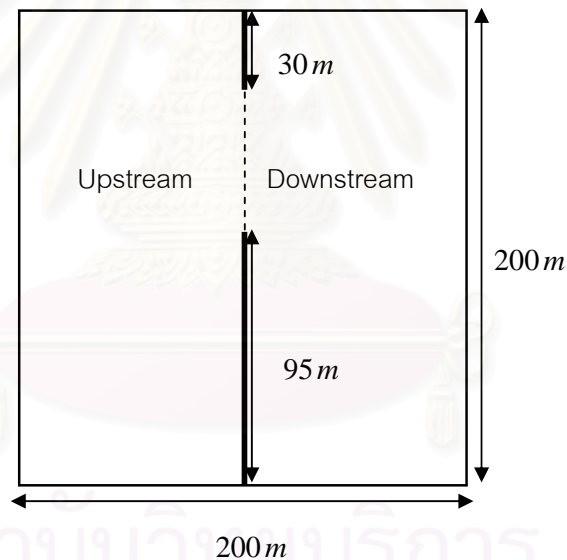


Figure 5.8 Geometry of rectangular dam break problem

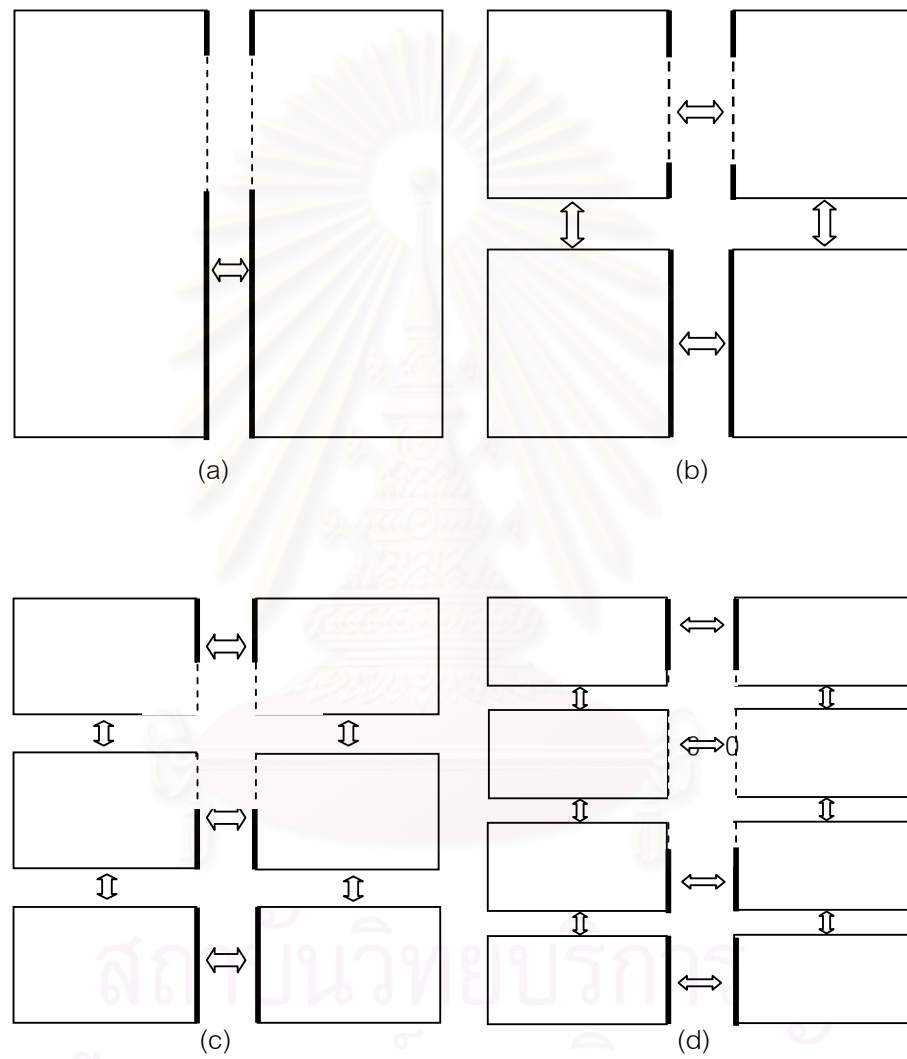


Figure 5.9 Appropriate domain decomposition technique of the rectangular dam break problem, (a) for using 2 processors, (b) for using 4 processors, (c) for using 6 processors and (d) for using 8 processors.

For this domain decomposition technique, we obtained a regular sub-domain. Hence, solving on a sub-domain can reduce a complexly computational. Moreover, there is no sending and no receive across the boundaries on dam region because of the dam wall is a boundary of sub-domain.

### 5.2.1 Wet bed

Initially, the water depth upstream of the dam is  $10m$  and the downstream water depth is assumed to be  $5m$ , both components of the velocity  $u$  and  $v$  are set to zero everywhere. Typical profiles of surface elevation of Roe solvers are shown in Figure 5.14. The performances of the parallel computation are shown in Table 5.3.

Table 5.3 Performance results of rectangular dam break with wet bed.

Number of processors	Unknowns per processor	Elapsed time(s)	Speedup	Efficiency(%)
1	120000	165.890	1.000	100.000
2	60000	82.085	2.021	101.048
4	30000	54.945	3.019	75.480
8	15000	36.567	4.537	56.708

We see that the speedup and the efficiency drop when increasing the number of processors, except when running on 2 processors due to using different code when running on 1 and 2 processors (see Figure 5.10 and Figure 5.11). The code that is used to run on 2 processors doesn't have complex computation and intercommunicating at dam wall because of the optimal domain decomposition. Table 5.4 shows the comparison of performance results between the optimal domain decomposition and the other. We can see that the first domain decomposition is more efficient than the other.

Table 5.4 Comparison of performance results between 2 differencing domain decomposition.

Number of processors	Unknowns per processor	Elapsed time(s)	Speedup	Efficiency(%)
1	120000	165.890	1.000	100.000
2 with appropriate domain decomposed	60000	82.085	2.021	101.048
2 with the other domain decomposed	60000	98.611	1.682	84.113

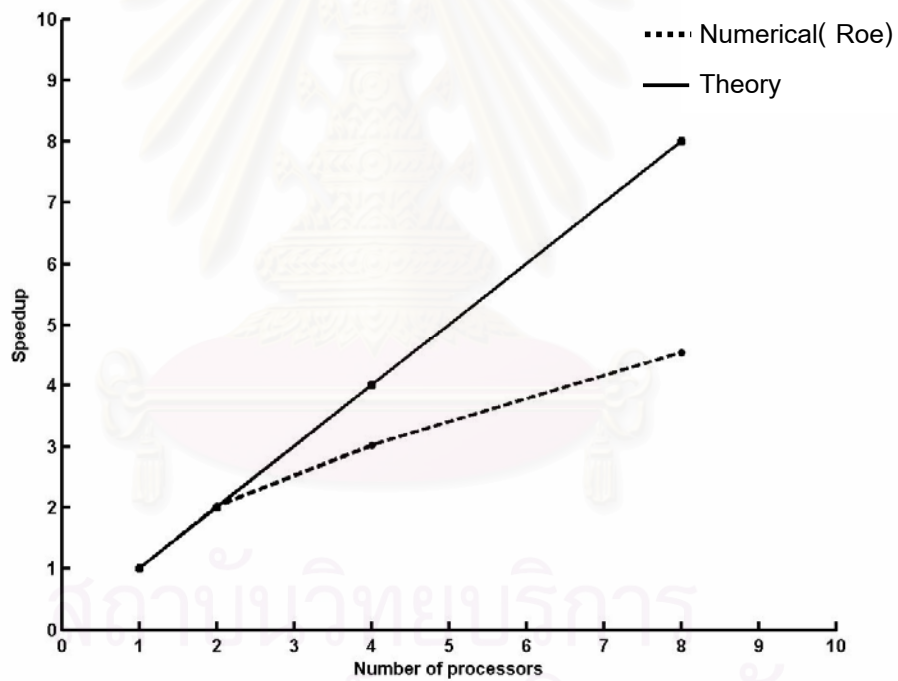


Figure 5.10 Relationship between the speedup and the number of processors in the cluster.

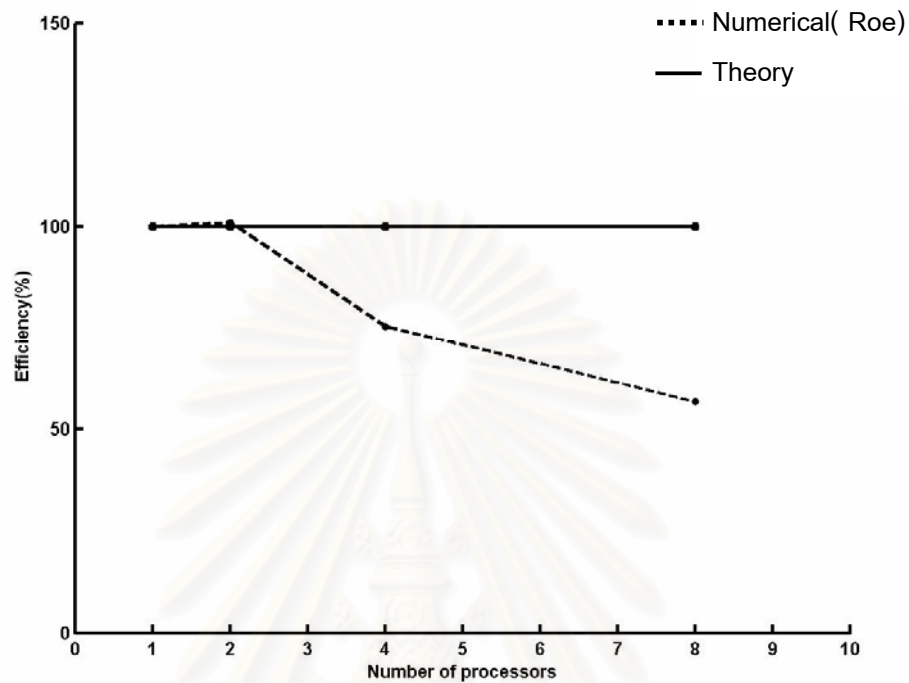


Figure 5.11 Relationship between the efficiency and the number of processors in the cluster.

### 5.2.2 Dry bed

In dry bed case, the initial water depth upstream of the dam is  $10m$  and the downstream water depth is assumed to be  $0m$  and both components of the velocity  $u$  and  $v$  are set to zero everywhere. Typical profiles of surface elevation of HLL solvers are shown in Figure 5.15 and the performances of the parallel computation are shown in Table 5.5. Figure 5.12 shows the relationship between the speedup and the number of processors and Figure 5.13 shows the relationship between the efficiency and the number of processors.

Table 5.5 Performance results of rectangular dam break with dry bed.

Number of processors	Unknowns per processor	Elapsed time(s)	Speedup	Efficiency(%)
1	120000	54.931	1.000	100.000
2	60000	29.633	1.854	92.686
4	30000	20.627	2.663	66.577
8	15000	16.589	3.311	41.391

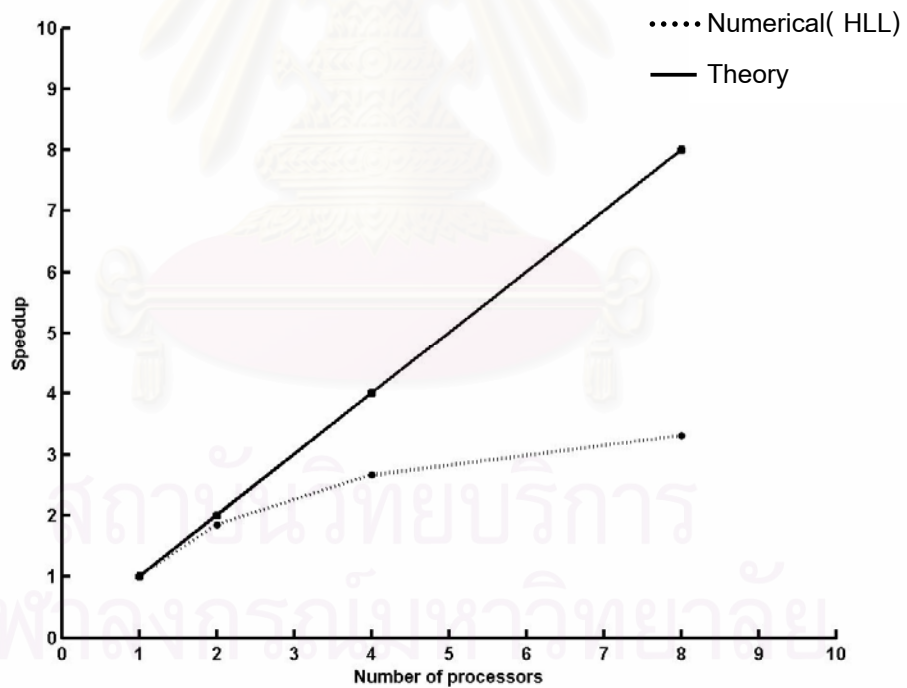


Figure 5.12 Relationship between the speedup and the number of processors in the cluster.

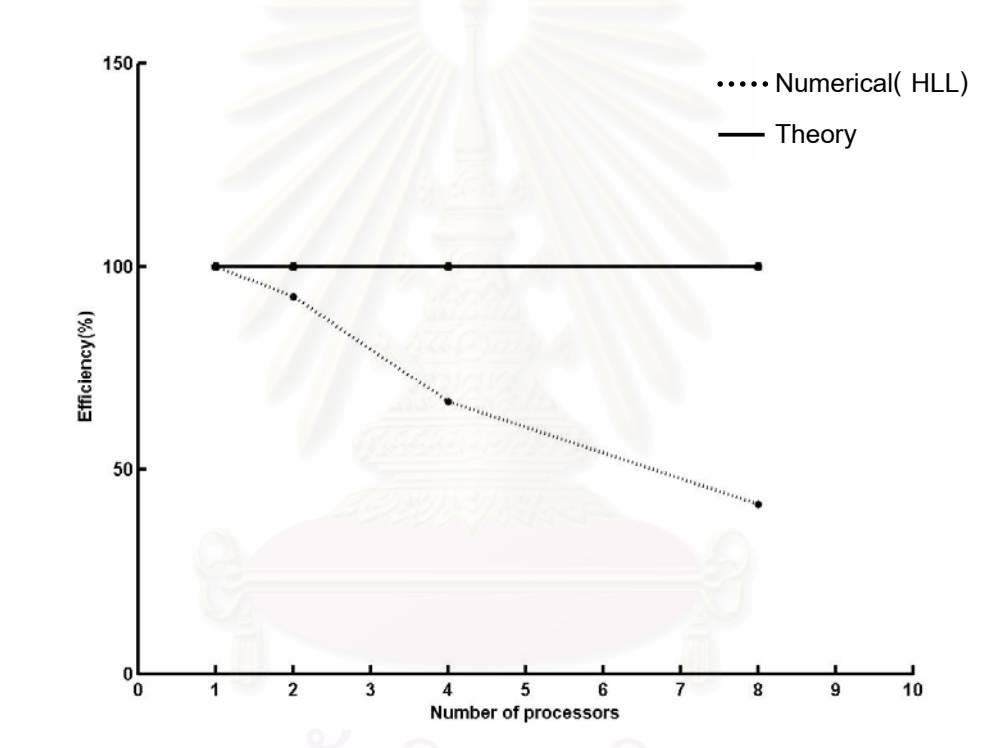
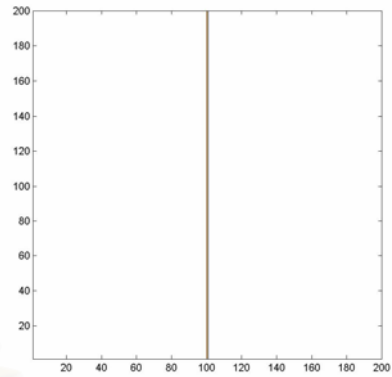
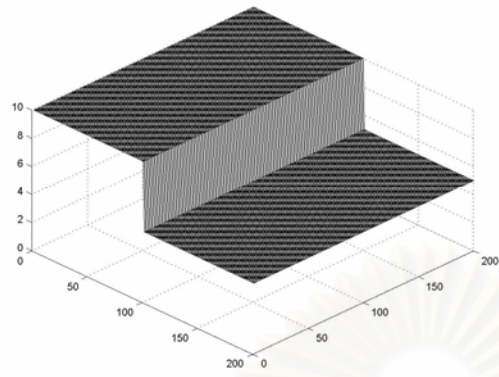
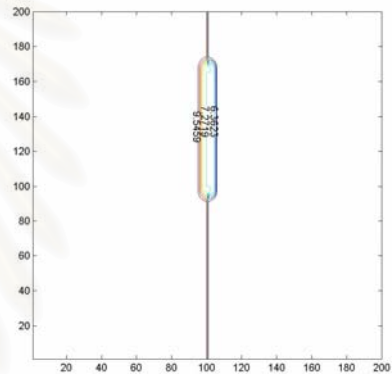
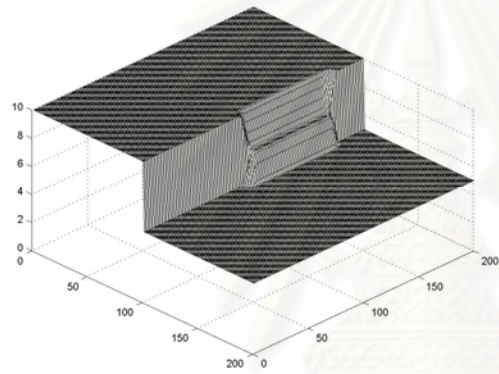


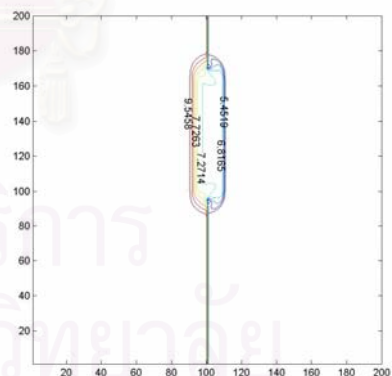
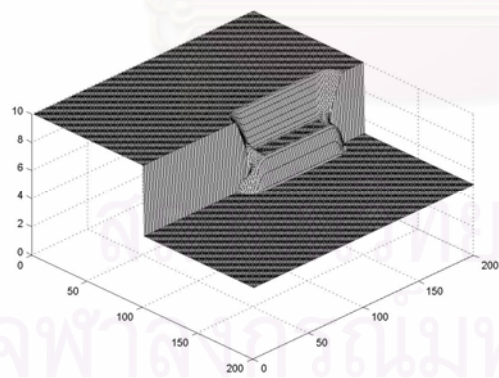
Figure 5.13 Relationship between the efficiency and the number of processors in the cluster.



(c1)  $t=0$

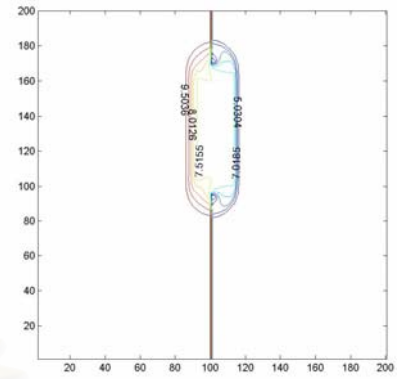
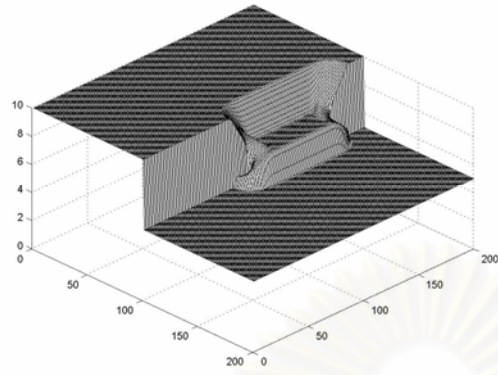
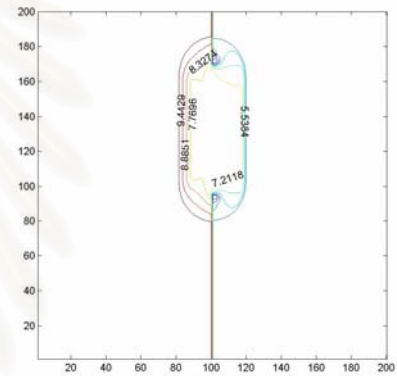
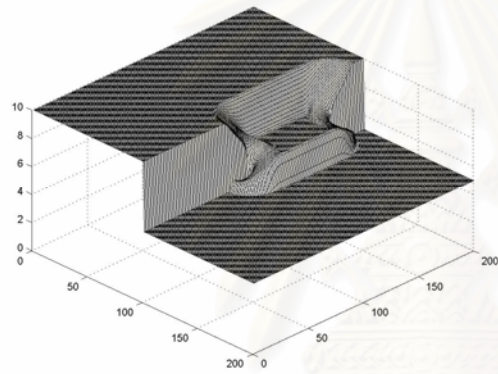
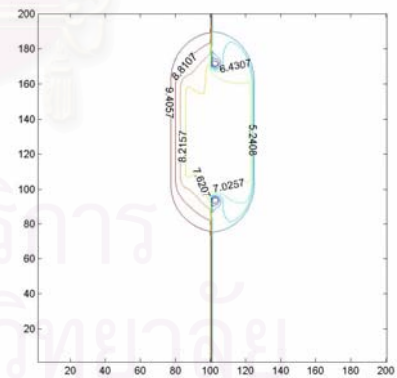
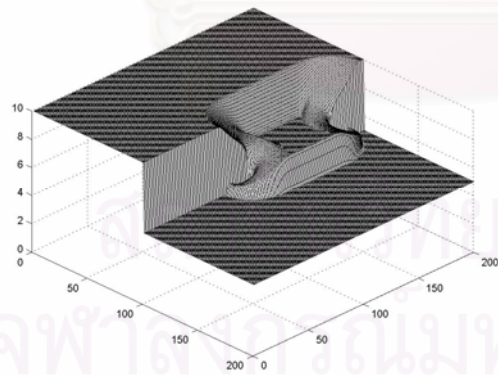


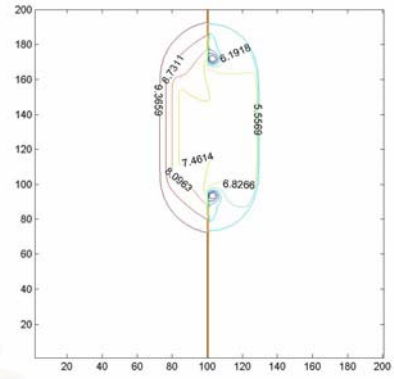
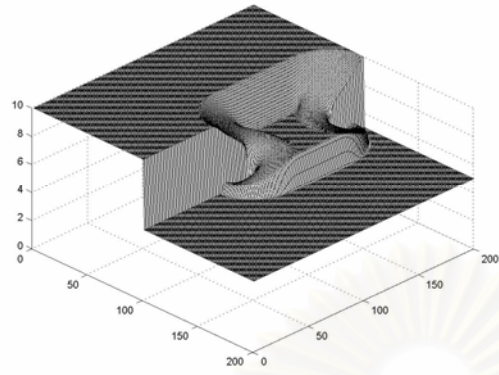
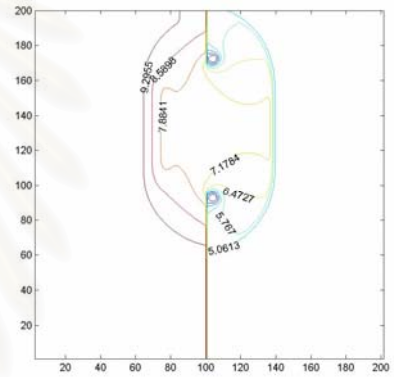
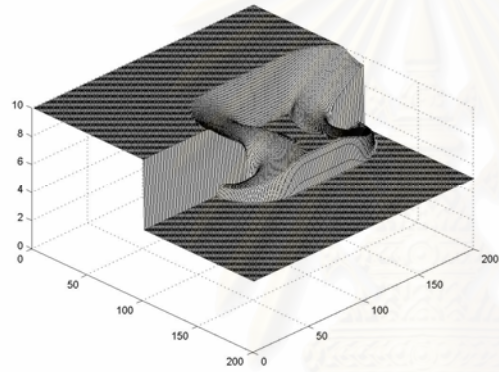
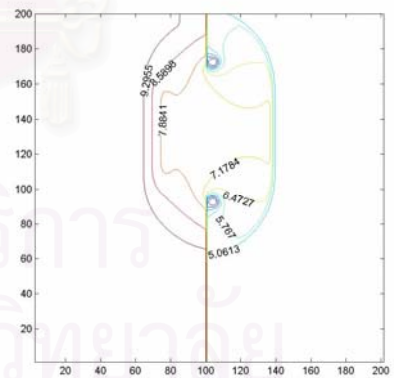
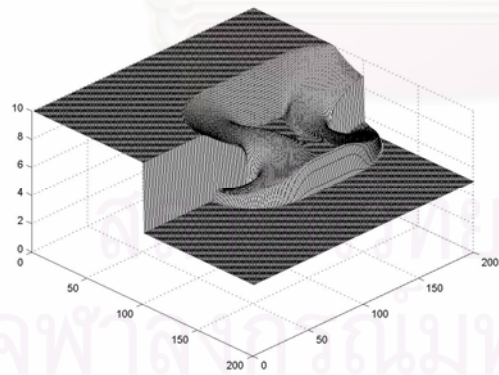
(c2)  $t=0.5$



(c3)  $t=1$



(c4)  $t=1.5$ (c5)  $t=2$ (c6)  $t=2.5$

(c7)  $t=3$ (c8)  $t=3.5$ (c9)  $t=4$

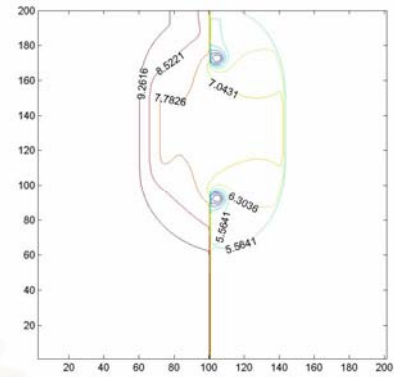
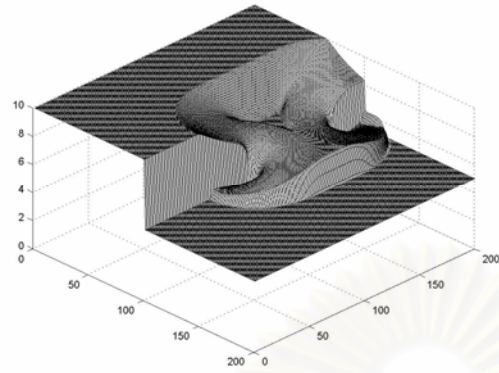
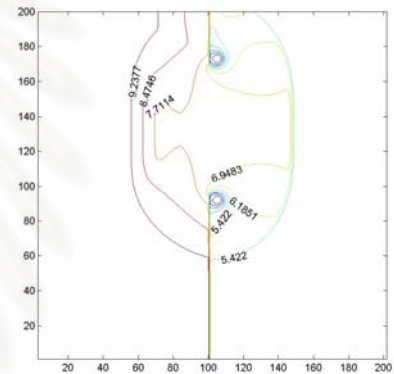
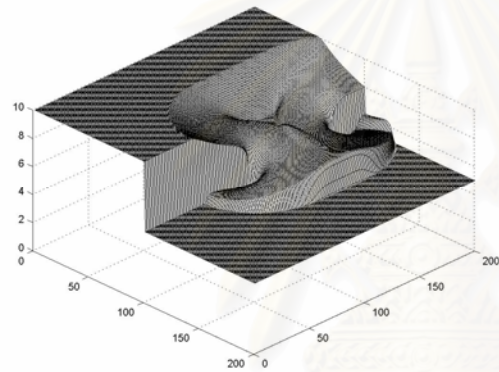
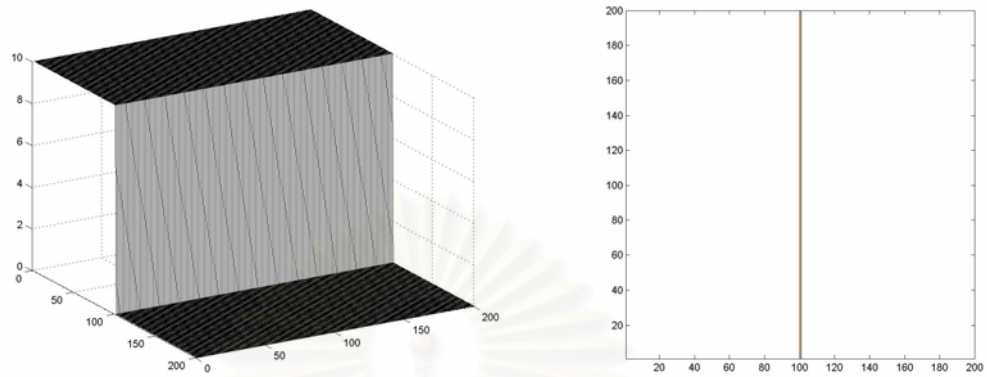
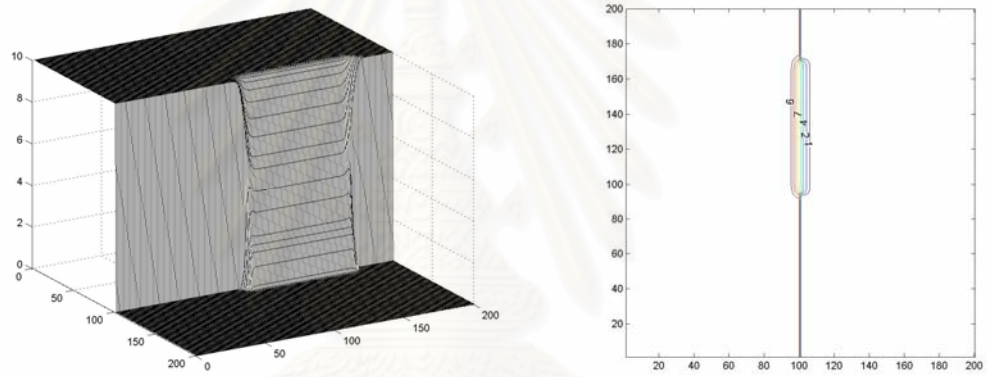
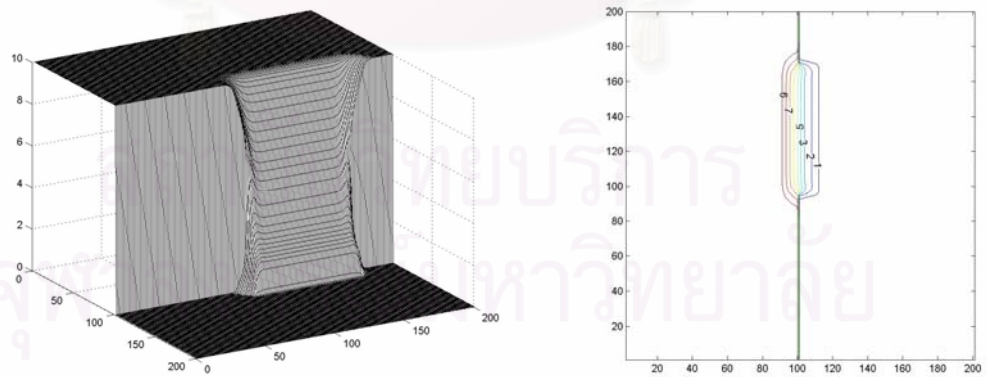
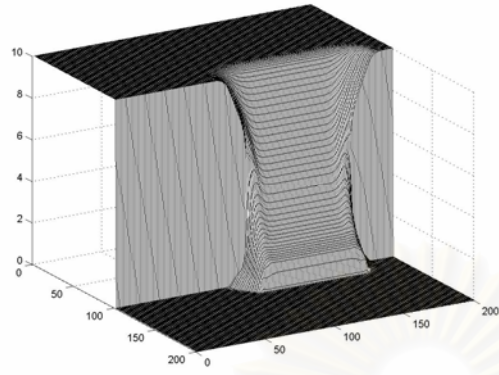
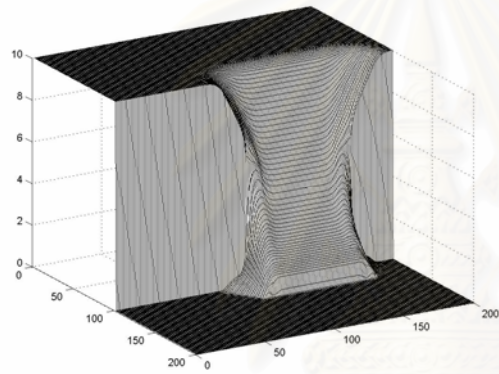
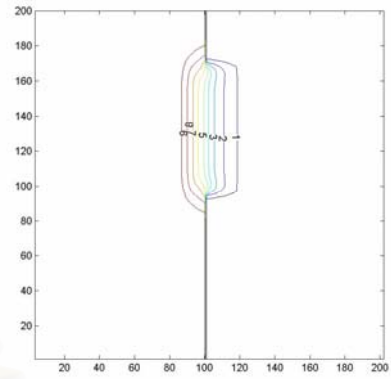
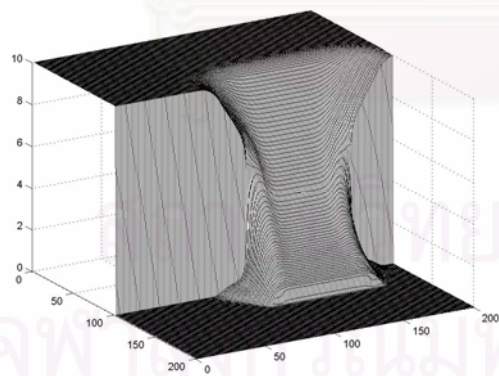
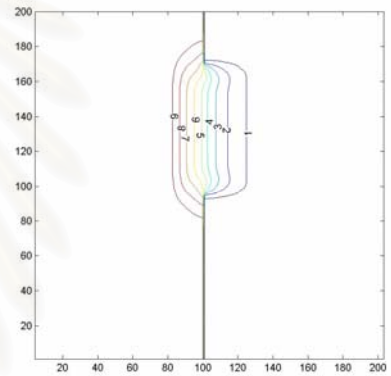
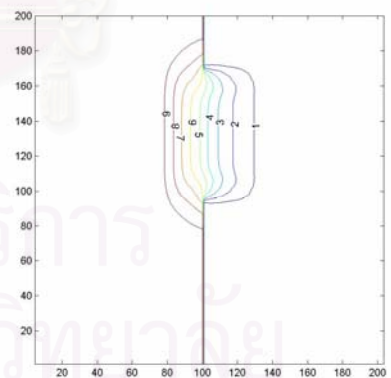
(c10)  $t=4.5$ (c11)  $t=5$ 

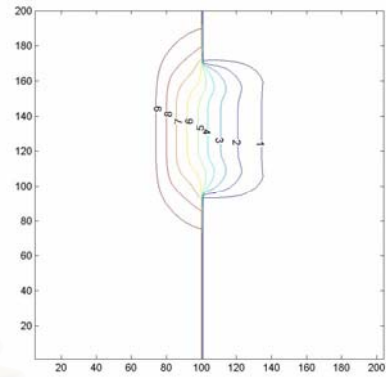
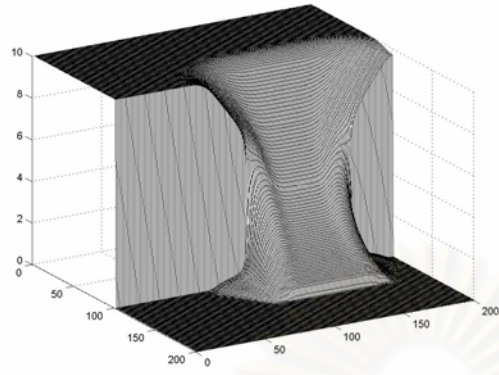
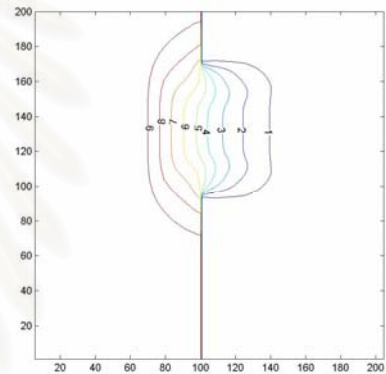
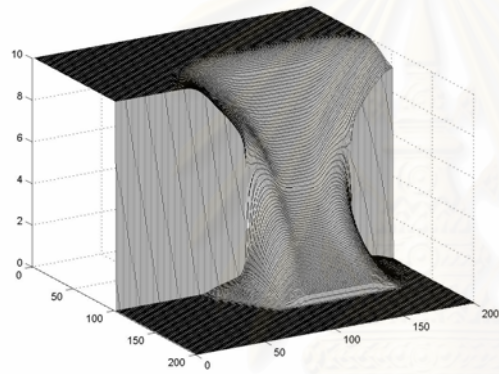
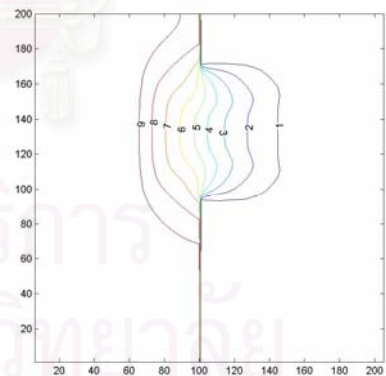
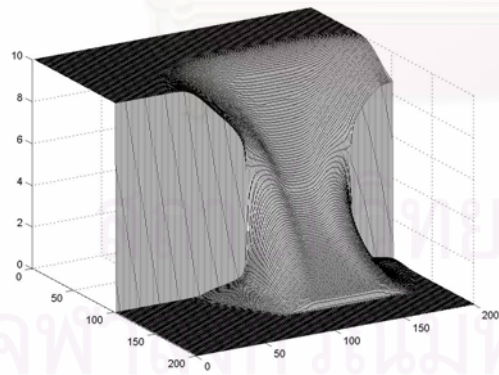
Figure 5.14 (c1)-(c11) show the rectangular dam break results in the case of wet bed, the left of figure shows water profile and the right of figure shows contour plot of the depth.

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

(d1)  $t=0$ (d2)  $t=0.5$ (d3)  $t=1$

(d4)  $t=1.5$ (d5)  $t=2$ (d6)  $t=2.5$ 



(d7)  $t=3$ (d8)  $t=3.5$ (d9)  $t=4$

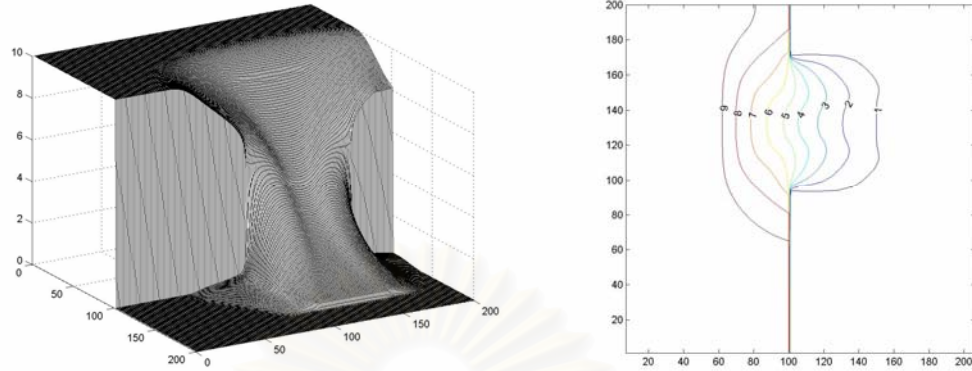
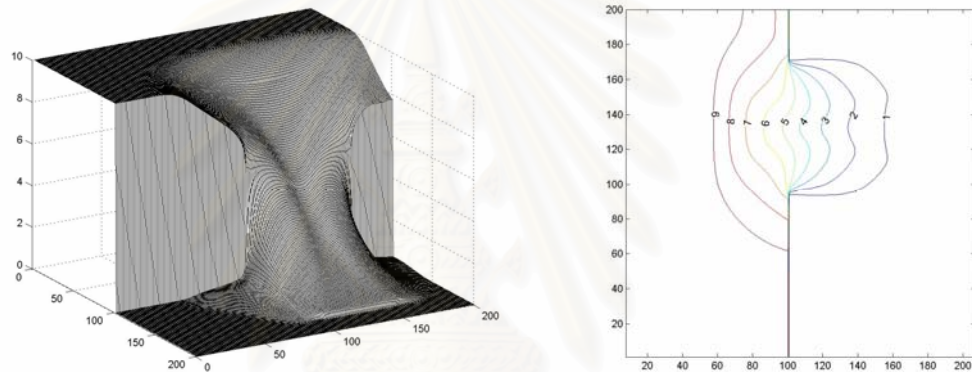
(d10)  $t=4.5$ (d11)  $t=5$ 

Figure 5.15 (d1)-(d11) show the rectangular dam break results in the case of dry bed, the left of figure shows water profile and the right of figure shows contour plot of the depth.

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

### 5.3 Comparison of the Performance Results Between Roe and HLL Solvers

From Tables 5.6 and 5.7 we see that the HLL solver requires less computational time than the Roe solver. This is mainly due to the fact that Roe solver has to first calculate the wave propagation in both directions before evaluating the fluxes. Hence the Roe solver has more operations as compared with the HLL solver. Surprisingly, the Roe solver can achieved better performance results than the HLL solver (see Figure 5.15).

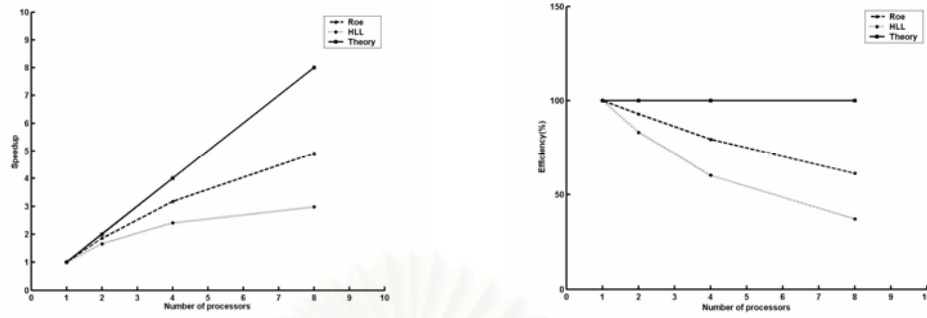
Table 5.6 Comparison of performance results between the Roe and the HLL solver for solving the circular dam break.

Number of processors	Unknowns per processor	Elapsed time(s)		Speedup		Efficiency(%)	
		HLL	Roe	HLL	Roe	HLL	Roe
1	120000	49.070	120.502	1.000	1.000	100.000	100.000
2	60000	29.565	64.883	1.660	1.857	82.987	92.861
4	30000	20.388	37.952	2.407	3.175	60.170	79.378
8	15000	16.472	24.594	2.979	4.900	37.237	61.246

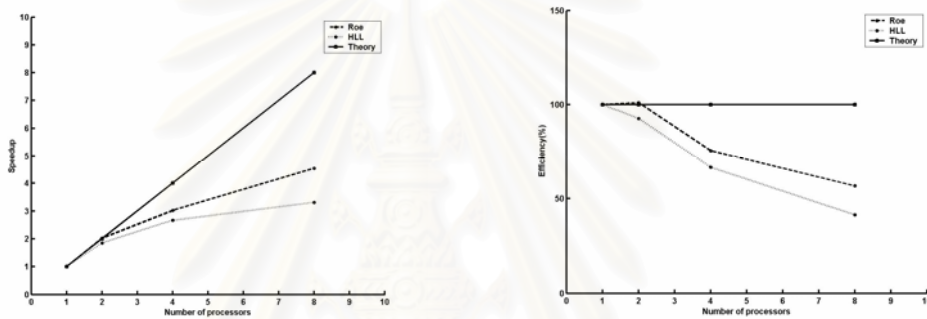
Table 5.7 Comparison of performance results between the Roe and the HLL solver for solving the rectangular dam break.

Number of processors	Unknowns per processor	Elapsed time(s)		Speedup		Efficiency(%)	
		HLL	Roe	HLL	Roe	HLL	Roe
1	120000	54.931	165.890	1.000	1.000	100.000	100.000
2	60000	29.633	82.085	1.854	2.021	92.686	101.048
4	30000	20.627	54.945	2.663	3.019	66.577	75.480
8	15000	16.589	36.567	3.311	4.537	41.391	56.708





(a)



(b)

Figure 5.16 Comparison of the performance results between the Roe and the HLL solvers for solving the dam break problems (a) Circular dam break (b) Rectangular dam break.

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## CHAPTER VI

### CONCLUSION

In this thesis, we use the shallow water equations to simulate the shallow water flow problems for the rectangular and the circular dam break problems. We use the finite volume methods to approximate the solution. A high-resolution Godunov's method is used to compute the numerical fluxes by using solution of the local Riemann problem at each cell interface. The Roe and the HLL approximate Riemann solvers are used to approximate the solution of this local Riemann problem. Our numerical results of dam break problems are in good agreement with the previous works [1,3,7,9].

We also present a parallel algorithm for solving the dam break problems. The domain decomposition technique is used to divide a physical domain into a number of smaller sub-domains. MPI is incorporated for inter-processor data communication. We find that a parallel computer can reduce the computational time. However, the efficiency decreases when there are too many processors in the cluster due to the imbalanced usage of resources. We analyze the performance results between Roe and HLL solvers and present the appropriate domain decomposed for the rectangular dam break.

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## REFERENCES

- [1] C. Zoppou and S. Roberts. Numerical solution of two-dimensional unsteady dam break. *Applied Mathematical Modelling*. 24(2000) : 457-475.
- [2] David L. George. Numerical Approximation of the Nonlinear Shallow Water Equation with Topography and Dry Beds: A Godunov-Type Scheme. Master's thesis Applied Mathematics Science University of Washington, 2004.
- [3] Jin-Wen Wang and Ru-Xun Liu. A comparative study of finite volume methods on unstructured meshes for simulation of 2D shallow water wave problems. *Mathematics and Computers in Simulation*. 53(2000) : 171-184.
- [4] P. Rao. A parallel hydrodynamic model for shallow water equations. *Applied Mathematics and Computation*. 150(2004) : 291-302.
- [5] R.J. LeVeque. Balancing Source Terms and Flux Gradients in High-Resolution Godunov Methods: The Quasi-Steady Wave-Propagation Algorithm. *Journal of Computational Physic*. 146(1998) : 346-365
- [6] S. Chippada, C. N. Dawson, M. L. Martinez and M. F. Wheeler. A Godunov-type finite volume method for the system of Shallow water equations. *Computer Methods in Applied Mechanics and Engineering*. 151(1998) : 105-129
- [7] S. Roberts and C. Zoppou. Robust and efficient solution of the 2D shallow water equation with domain containing dry bed. *ANZIAM Journal*. 4(2000) : c1260-c1282.
- [8] V. Guinot. An approximate two-dimensional Riemann solver for hyperbolic system of conservation laws. *Journal of Computational Physics*. (2004)
- [9] V. Caleffi, A. Valiani and A. Zanni. Finite volume method for simulating extreme flood events in natural channels. *Jornal of Hudraulic Research*. 2(2003) : 167-177
- [10] A. Harten, P.D. Lax and B. Van Leer. On upstream differencing and Godunov-type scheme for hyperbolic conservation laws. *SIAM Review*. 25(1983) : 235-261.
- [11] R.J. LeVeque. Finite volume methods for hyperbolic problems. Cambridge Texts in Applied Mathematics: Cambridge University Press, Cambridge, United Kingdom 2002.

- [12] Tan Weiyen. Shallow water hydrodynamics : mathematical theory and numerical solution for a two-dimensional system of shallow water equations. 1992.
- [13] P.S. Pacheco and Woo Chat Ming. MPI User Guide in FORTRAN.  
[http://www.hku.hk/cc/sp2/ftp/mpi/MPI\\_ug\\_in\\_FORTRAN.doc](http://www.hku.hk/cc/sp2/ftp/mpi/MPI_ug_in_FORTRAN.doc)



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## VITAE

Pairin Suwannasri was born in February 22, 1980. He received a bachelor's degree in Science (Mathematics) from Faculty of Science, KhonKhan University in 2001.



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย