

การคำนวณเชิงขนานสำหรับสี่นามิ



นางสาวนัฏฐิตา โพธิ์เพชร

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต


สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2550

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

PARALLEL COMPUTATION FOR TSUNAMI



Miss Nuttita Pophet

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Science Program in Computational Science

Department of Mathematics

Faculty of Science

Chulalongkorn University


Academic Year 2007

Copyright of Chulalongkorn University

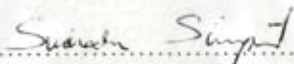
Thesis Title                      PARALLEL COMPUTATION FOR TSUNAMI  
By                                      Miss Nuttita Pophet  
Field of Study                      Computational Science  
Thesis Advisor                      Associate Professor Jack Asavanant, Ph.D.  
Thesis Co-advisor                      Mansour Ioualalen, Ph.D.

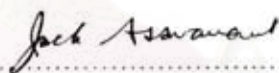
---

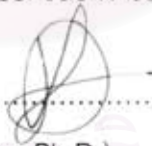
Accepted by the Faculty of Science, Chulalongkorn University in Partial  
Fulfillment of the Requirements for the Master's Degree


 ..... Dean of the Faculty of Science  
(Professor Supot Hannongbua, Ph.D.)

THESIS COMMITTEE

 ..... Chairman  
(Associate Professor Suchada Siripant)

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

 ..... Thesis Co-advisor  
(Mansour Ioualalen, Ph.D.)

 ..... Member  
(Assistant Professor Krung Sinapiromsaran, Ph.D.)

นัญฐิตา โพธิ์เพชร : การคำนวณเชิงขนานสำหรับสึนามิ. (PARALLEL COMPUTATION FOR TSUNAMI) อ.ที่ปรึกษา: รองศาสตราจารย์ ดร.จักษ์ อัครวานันท์, อ.ที่ปรึกษาร่วม : ดร.มองซูร์ อีวาลาเลน, จำนวนหน้า 64 หน้า.

การจำลองการเดินทางของคลื่นในโดเมนขนาดใหญ่เช่นในกรณีของสึนามิ เป็นปัญหาที่มีความท้าทาย เนื่องจากขนาดของโดเมนและจำนวนของตัวแปรไม่ทราบค่าหลายตัว การจำลองคลื่นจึงจำเป็นต้องใช้เวลาในการปฏิบัติการคำนวณและใช้หน่วยความจำมาก ในกรณีดังกล่าวการใช้คอมพิวเตอร์เพียงเครื่องเดียวซึ่งมีหน่วยความจำที่จำกัด ไม่เพียงพอที่จะทำการคำนวณ เพื่อที่จะตอบสนองความต้องการในการคำนวณ จึงได้พัฒนาการคำนวณเชิงขนานโดยใช้เทคนิคการแบ่งโดเมน และใช้ message passing interface (MPI) แบบจำลองที่ใช้ในการจำลองการเดินทางของสึนามิ คือ Boussinesq Model และใช้ระเบียบวิธีผลต่างสี่เหลี่ยม (Finite difference method) ใช้ Parallel tridiagonal solver และวิธีการรับส่งข้อมูลที่มีประสิทธิภาพเพื่อแก้ปัญหาระบบไตรไดอะโกนอลเมทริกซ์ (tridiagonal matrix systems) ทำการทดสอบโปรแกรมการคำนวณเชิงขนานด้วยเหตุการณ์สึนามิที่เกิดขึ้นในปี 1999 ที่วานูอาตู และปี 2004 ในมหาสมุทรอินเดีย ผลที่ได้พบว่า การคำนวณเชิงขนานมีประสิทธิภาพที่ดีและเหมาะสมกับปัญหาการจำลองคลื่นสึนามิ

## สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา คณิตศาสตร์  
สาขาวิชา วิทยาการคนนา  
ปีการศึกษา 2550

ลายมือชื่อผู้คิด..... นัญฐิตา โพธิ์เพชร  
ลายมือชื่ออาจารย์ที่ปรึกษา..... Jack Stewart  
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม..... H. Inalalen

## 4872334323 : MAJOR COMPUTATIONAL SCIENCE

KEY WORD: PARALLEL COMPUTATION / TSUNAMI SIMULATION / MPI / TRIDIAGONAL SOLVER / BOUSSINESQ MODEL

NUTTITA POPHET : PARALLEL COMPUTATION FOR TSUNAMI. THESIS

ADVISOR: ASSOC. PROF. JACK ASAVANANT, Ph.D., THESIS CO-ADVISOR :

MANSOUR IOUALALEN, Ph.D., 64 pp.

In a simulation of wave propagation in a large fluid domain like the case of tsunami, the implementation of wave model for such large computational domain has been a challenging task. Because of the domain size with several unknown variables, wave simulation requires large amounts of computational time and memory. In such case a single computer with limited memory size is inadequate to carry out the computations. To make such computational demand possible, parallel code is developed with the use of domain decomposition technique and the message passing interface (MPI). The Boussinesq wave model for the numerical simulation of tsunami propagation is implemented based on a finite difference scheme. Parallel tridiagonal solver and efficient data transfer scheme are used to improve the performance of solving tridiagonal systems. Two tsunami events, the 1999 Vanuatu tsunami and the 2004 Indian Ocean tsunami are used to test the parallelized program. Experimental results indicate that parallel code running on more than one processor gives much better performance and is found to fit the tsunami model very satisfactorily in comparison with running on a single processor.

Department Mathematics  
Field of study Computational Science  
Academic year 2007

Student's signature.....*ນັກສຶກສາ*.....  
Advisor's signature.....*Jack Asavanant*.....  
Co-advisor's signature.....*M. Ioualalen*.....

## ACKNOWLEDGEMENTS

I am deeply grateful to my supervisor, Associate Professor Dr. Jack Asavanant, who always gives generous advice, encouragement and kindly support me throughout this program. I am particularly grateful to my co-supervisor, Dr. Mansour Ioualalen from Geosciences Azur (IRD, CNRS, UPMC, UNSA), Villefranche-sur-Mer, France, to whom with his kindness, understanding and valuable advice enabled me to make this thesis possible.

I would like to gratefully acknowledge Faculty of Science, Chulalongkorn University, the National Electronics and Computer Technology Center (NECTEC) and Geosciences Azur, Villefranche-sur-Mer, France, for the use of their computer clusters. I would also like to thank the Advanced Virtual and Intelligence Computing Research Center (AVIC) for supporting in enabling me to accomplish research. Also thank you very much Mr. Supakit Prueksaaron and Mr. Chinawat Vilasineewan, thank you for supporting and advising me in using parallel computer.

I sincerely thank to the chairman and member of the committee of this thesis, Associate Professor Suchada Siripant and Assistant Professor Dr. Krung Sinapiromsaran, for their valuable comments and personal efforts in reviewing my work.

I am indebted to National Research Council of Thailand and CU. Graduate School Thesis Grant for financial support and Franco-Thai Cooperation Program for supporting me 1 month to visit Geosciences Azur, Villefranche-sur-Mer, France.

I would like to give a special thank to Ioualalen's family for providing accommodation during my stay in France including moral support and thanks especially to all of my friends in the Department of Mathematics for providing me constant encouragement. Finally, I would like express my sincere gratitude to my beloved parents for warmest support that becomes an invaluable source of strength along the duration of this study.

## TABLE OF CONTENTS

	Page
ABSTRACT(THAI) .....	iv
ABSTRACT(ENGLISH) .....	v
ACKNOWLEDGEMENTS .....	vi
TABLE OF CONTENTS .....	vii
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
CHAPTER	
I INTRODUCTION. ....	1
1.1 Introduction to Tsunamis. ....	1
1.2 Introduction to FUNWAVE Model. ....	5
1.3 Objective and Scope of the Thesis. ....	7
1.4 Outline of the Thesis. ....	8
II Descriptions of the Tsunami Cases Studies. ....	9
2.1 The November 26, 1999 Vanuatu Earthquake/Tsunami. ....	9
2.2 The December 26, 2004 Indian Ocean Earthquake/Tsunami. ....	12
III PARALLEL COMPUTATION .....	16
3.1 Parallel Computer Memory Architectures. ....	16
3.1.1 Shared-Memory Architecture. ....	16
3.1.2 Distributed-Memory Architecture. ....	17
3.1.3 Distributed-Shared Memory Architecture. ....	18
3.2 Message Passing Interface (MPI) .....	18
3.3 Domain Decomposition. ....	19
3.4 Overlapping Communication. ....	20
3.5 Parallel Tridiagonal Solver. ....	21
3.6 Parallel Performance Measurements. ....	23
IV PARALLEL IMPLEMENTATION. ....	25
4.1 Parallel Implementation of Tsunami Source. ....	29
4.2 Parallel Implementation of Wave Propagation. ....	30
V EXPERIMENTAL RESULTS AND DISCUSSION. ....	32
5.1 Parallel Computation for Boussinesq Equations with Gaussian Source Function. ....	32

## TABLE OF CONTENTS (Continued)

	Page
5.2 Parallel Computation for Tsunami Events. . . . .	34
5.2.1 The November 26, 1999 Vanuatu Tsunami. . . . .	35
5.2.2 The December 26, 2004 Indian Ocean Tsunami. . . . .	36
5.2.3 Performance Results for Different Domain Sizes. . . . .	38
5.2.4 Parallel Performance based on Different Domain Decompositions. . . . .	40
5.2.5 Performance and Parallel I/O. . . . .	41
5.2.6 Block Sizes and Pipelining Tridiagonal Solver. . . . .	42
5.2.7 Overall Performance of Parallelization on Computer Clusters. . . . .	42
VI CONCLUSION. . . . .	44
References. . . . .	47
APPENDICES	
A Boussinesq wave Equations, Bottom friction, Wave Breaking, Moving Shorelines, and Subgrid Turbulence . . . . .	53
B Numerical Scheme . . . . .	58
VITAE . . . . .	64



## LIST OF TABLES

Table	Page
2.1 1999 Vanuatu fault tsunami source parameters . . . . .	10
2.2 2004 Indian Ocean tsunami source parameters. . . . .	15
4.1 Text of parallel code . . . . .	29
5.1 Performance results of the tsunami source. . . . .	35
5.2 Performance results of the wave propagation. . . . .	35
5.3 Performance results of the tsunami source and the wave propagation. .	35
5.4 Performance results of the tsunami source. . . . .	36
5.5 Performance results of the wave propagation. . . . .	36
5.6 Performance results of the tsunami source and the wave propagation. .	37
5.7 Performance results of the tsunami source. . . . .	37
5.8 Performance results of the wave propagation. . . . .	37
5.9 Performance results of the tsunami source and the wave propagation. .	38
5.10 Performance results of the tsunami source and the wave propagation between difference clusters. . . . .	43

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## LIST OF FIGURES

Figures	Page
2.1 Central Vanuatu Area. . . . .	9
2.2 Simulated vertical displacements. . . . .	11
2.3 Northern Sumatra area. . . . .	13
2.4 Initial surface elevation for the 2004 Indian Ocean tsunami. . . . .	14
3.1 Shared-Memory Architecture. . . . .	17
3.2 Distributed-Memory Architecture. . . . .	17
3.3 Distributed-Shared Memory Architecture. . . . .	18
3.4 Domain decomposition topologies for 1D and 2D decompositions of 4 processors. . . . .	19
3.5 The computational domain with overlapping areas. . . . .	20
3.6 The discretized domain with unknown variables in X direction. . . . .	22
3.7 The pipelining scheme for 4 processors. . . . .	23
4.1 The flowchart of the program. . . . .	25
4.2 The 2D Cartesian topology. . . . .	27
4.3 The algorithm for sending output information to the master processor. . . . .	28
5.1 Gaussian initial free surface profile. . . . .	32
5.2 The wave profile calculated by using 4-processors at $t=15, 30$ and 45s. . . . .	33
5.3 Relationship between speedup, efficiency and the number of processor for tsunami source. . . . .	38
5.4 Relationship between speedup, efficiency and the number of processor for wave propagation. . . . .	39
5.5 Relationship between speedup, efficiency and the number of processor for tsunami source and wave propagation. . . . .	39
5.6 Speedup for different decompositions. . . . .	40
5.7 Performance for parallel output. . . . .	41
5.8 Speedup for different block size of messages in pipelining scheme. . . . .	42

# CHAPTER I

## INTRODUCTION

### 1.1 Introduction to Tsunamis

Tsunami is a Japanese word represented by two characters as "tsu" and "nami". The character "tsu" means harbor, while the character "nami" means wave. The phenomenon of tsunami (*soo-NAH-mee*) is a series of waves of extremely long wavelength and long period caused by earthquakes, submarine slides, volcanic eruptions and asteroids (Bondevik *et al.*, 1997). They have a long wavelength (e.g. up to 150 km for the December 26, 2004 Indian Ocean tsunami) and travel across the oceans at great velocity (Dawson and Smith, 2000). Tsunamis are shallow-water waves and are different from wind-generated waves seen on a beach. A wave is characterized as a shallow-water wave if the ratio between the water depth and its wavelength is small while wind-generated waves usually have period of five to twenty seconds and a wavelength of about 100 to 200 meters.

The devastating impacts of tsunami waves have been reported for hundreds of years, and yet only a few modern tsunami deposits have been described in detail (e.g. Wright and Mella, 1963; Nishimura and Miyaji, 1995; Sato *et al.*, 1995; Shi *et al.*, 1995; Minoura *et al.*, 1997; Dawson *et al.*, 1996; Gelfenbaum and Jaffe, 2003). In several cases, depending on the tsunami source characteristics and on its location, when the tsunami arrives, the water level in the sea may first drop significantly and the waters receding several hundred metres comes back with great speed faster than any person can run (Bryant, 2001). Then the first wave crest follows.

The initial condition of an earthquake-generated tsunami is the water surface displacement caused by ocean bottom deformation due to faulting. The fault motion of the earthquake can be described by the fault parameters proposed by Okada (1985): the location of the fault, geometry (strike, dip, and rake angles), the fault size (length and width), and the slip amount.

Earthquakes generate tsunamis when the sea floor suddenly deforms and vertically displaces the overlying water from its equilibrium position. An earthquake that causes a tsunami with magnitude greater than expected from routine analysis of

its seismic waves is called a tsunami earthquake (Kanamori, 1972). Waves are formed as the displaced water mass, which acts under the influence of gravity, attempts to regain its equilibrium. The main factor which determines the initial size of a tsunami is the amount of vertical sea floor displacement estimated by the earthquake's magnitude, depth, fault plane mechanism, the velocity of the sea floor deformation and the water depth near the earthquake source.

The speed of a shallow-water wave is equal to the square root of the product of the acceleration of gravity ( $9.81 \text{ m.s}^{-2}$ ) and the depth of the water. In deep oceans where the depth is  $\sim 6 \text{ km}$ , the wave speed can be as high as  $970 \text{ km.hr}^{-1}$ . The rate at which a wave loses its energy is inversely related to its period because viscous damping time scale decreases with the wave frequency. Since a tsunami has a very large wavelength, it will lose little energy as it propagates. Hence in very deep water, a tsunami will travel at high speeds especially across the great transoceanic distances with limited energy loss.

The amplification of a tsunami is largely determined by three processes: shoaling, focusing and resonance. Shoaling occurs when a wave encounters a negative bathymetric gradient (water depth decreasing in the direction of propagation). The leading wave component has a decreasing velocity and the trailing water builds up behind, thereby increasing the wave height. This process takes place mainly in coastal areas or at the continental slope. When a tsunami encounters a concave-shaped (convergent) pattern in the bathymetry, the process of focusing occurs at which refraction causes the wave front to concentrate. This phenomenon commonly arises near a cape and is enhanced when the region is surrounded by open bays of convex (divergent) shape. Bay resonance generally takes place in closed bays where a trapped wave oscillates in a fashion similar to that of a standing wave. Resonance may then occur when a phase-locking begins between one particular mode of the tsunami and the bay frequency mode. The phase-locking is strictly linked to the time scales of the tsunami and the size of the bay. The existence of a particular resonance mode associated with the phase-locking is linked to the nature of the dispersion of the incoming wave.

All three main modes of amplification can be explained by linear theory. Contrary to wind-generated short wave, the nonlinear interactions of a tsunami are weak offshore and are not crucial in coastal areas. This is because in the open ocean a tsunami does not encounter other waves with the same order wave spectrum. Local

amplification effects do not solely determine the final wave size of a tsunami. Fairly, short-wave nonlinear interactions do not directly contribute to wave amplification but allow an increase of the maximum possible load of the wave when the wave increases its period through Benjamin and Feir (1967) sub-harmonic instability. The instability generates a wave modulation and a subsequent frequency downshift controlled by viscous damping of the initial higher frequency wave. This instability has been described by McLean (1982a,b) for two-dimensional progressive waves (Class I instability) and later by Ioualalen and Kharif (1994) and Ioualalen *et al.* (1999) for three-dimensional waves (Classes Ia and Ib instabilities).

Better knowledge of the mode of triggering and propagation of a tsunami may help in a better characterization of the seismic source. The relationship is however not straightforward because some transfer functions are not accurately known. In particular, it is difficult to estimate the localized seafloor deformation resulting from an earthquake. Furthermore the representation of Okada (1985) frequently used to characterize seafloor deformation, at times, inappropriately constrains the medium involved to be static and homogeneous. Estimating the manner in which the sea floor deformation is restituted on the water column is also problematic. We generally consider the two deformations as identical. It is fair to say that a more complex formulation or simulation of these transfer functions would require much more elaborated and extended observational networks than the existing ones used to constrain them. Currently tide gauges are employed to measure for the hydrographic behavior and GPS positioning and seismic stations are used to observe earthquake conditions. In other words, the degree of accuracy, the variety and the extent of the observational networks are always dependent on our capability to simulate and represent the prominent physical processes. This ability is continually developing as numerical and theoretical representations and computational power increase. This relationship between instrumentation and modeling is in constant improvement. We may always, at a certain degree of accuracy, perform simulations of tsunami to complement the direct observations of any tsunamigenic source. As an example, Ioualalen *et al.* (2007) have calibrated a tsunami source of the December 26, 2004 Sumatra event with the aid of available tide gauges records, sea level anomalies obtained from JASON altimeter and numerical simulations. Another example showing how a tsunami information (observations and modeling) may help in better constraining its source characteristics has been described by Ioualalen (2008): With

the aid of a single tide gauge he shows that the sequence of waves (the crests/troughs chronology, the ratio between the first successive crest or trough amplitudes) may determine some aspects of the fault mode of rupture, e.g., dipping orientation and its amount (normal or reverse faulting).

Tsunami modeling can also help in determining which possible source has generated an observed tsunami when several candidates are in competition. Such configuration may happen when an earthquake has generated submarine landslides, both being possibly tsunamigenic. This is a crucial issue in terms of coastal risk assessment because the identification of the actual tsunamigenic source determines the type of future potential tsunamis, their recurrence and associated impact. For example, Ioualalen et al. (2006) have studied the tsunami that occurred in Vanuatu on November 26, 1999 which hit Pentecost island (Martelli Bay) with waves as high as 6-7 m. The candidates were the  $M_w=7.5$  earthquake itself and two possible (but not dated) submarine landslides. The two landslides have been identified through floor scars. Thanks to multibeam bathymetric surveys that have been operated before the event and one of them has been located in the immediate vicinity of the earthquake hypocenter. Despite the uncertainty on the landslides information, the authors had no choice than taking into consideration all possible sources. They found that the co-seismic deformations were more likely to have generated the tsunami according to available tsunami observations.

Another key issue in tsunami modeling is the ability of a robust numerical simulation to provide a synoptic picture of a specific event and, in particular, the runup distribution along a coastline. A robust simulation relies on accurate bathymetric and topographic data sets (an accurate computational domain), a best-fitted geophysical source and a reliable numerical model simulating the tsunami propagation and runup. The best-fitted solution may be obtained through tsunami modeling by using both available hydrographic and geophysical data sets. Thus, ideally, the runup distribution could be obtained in a prognostic mode (or predictive), or at least partially. As a result the simulation would be validated using available runup observations and, elsewhere, the simulation would predict runup. Such methodology allows runup to be predicted in areas where no observations were made. Poorly sampled data, generally arises because a location is too sparsely populated, difficult to access, or simply because the coastline was too long to be fully sampled as in the case of the December 26, 2004 Indian Ocean tsunami. Moreover, the picture

may reveal vulnerable areas as well as sheltered ones. Such tsunami risk assessment may help in future development plans of a particular coastal area in the case of an eventual recurrent geophysical event. This aspect is essential for building a reliable and optimal instrumental sampling for a tsunami warning system. Such methodology has been applied successfully by Grilli *et al.* (2007) and Ioualalen *et al.* (2007).

## 1.2 Introduction to FUNWAVE Model

For an earthquake-derived tsunami, the initial tsunami elevation is based on the half-plane solution of an elastic dislocation problem (Okada, 1985). A planar rectangular fault is discretized into many small trapezoids and the point source solution of Okada (1985) is used to sum the contributions made by each trapezoid to vertical coseismic displacement, based on the actual depth of the trapezoid.

The computed initial condition is then transferred directly to FUNWAVE tsunami propagation model. FUNWAVE tsunami Boussinesq propagation and runup model is fully nonlinear and dispersive, retaining information to leading order in frequency dispersion  $O[(kh)^2]$  and to all orders in nonlinearity  $a/h$  (where  $k$  denotes an inverse wavelength scale,  $a$  denotes a wave amplitude, and  $h$  denotes a water depth) (Wei and Kirby, 1995; Wei *et al.*, 1995) (Appendix A for details). These equations have been used for several decades to model nonlinear waves with weak frequency dispersion. The advantage of Boussinesq system over the Navier-Stokes equation is the assumption no momentum transfer in the vertical direction. This assumption is equivalent to hydrostatic pressure distribution. With this useful feature, Boussinesq model provides an alternative for researchers and scientists in the fields of ocean modeling and tsunami simulation. Instead of tracking the moving boundary during wave runup/rundown on the beach or coastlines, Funwave model treats the entire computational domain as an active fluid domain by employing an improved version of the slot or permeable-seabed technique, i.e. the moving shoreline algorithm proposed by Chen *et al.* (2000) and Kennedy *et al.* (2000) for simulation of runup. Basic idea behind this technique is to replace the solid bottom where there is little or no water covering the land by a porous seabed or to assume that the solid bottom contains narrow slots. This is incorporated in terms of mass flux and free surface elevation in order to conserve mass in the presence of slots. The model includes bottom friction, energy dissipation to account for the wave breaking and a subgrid

turbulence scheme. The bottom friction is modeled by the use of the quadratic law with bottom friction coefficient between  $1.0 \times 10^{-3}$  to  $5.0 \times 10^{-3}$ . The subgrid turbulence is modeled in terms of Smagorinsky-subgrid turbulent mixing type to account for the effect of underlying current field. The energy dissipation due to wave breaking in shallow water is treated by the use of momentum mixing terms. The associated eddy viscosity is essentially proportional to the gradient of the horizontal velocity which is strongly localized on the front face of the breaking wave. The validation and verification of this breaking wave formulation for short wave shoaling and runup can be found in Chen *et al.* (2000) and Kennedy *et al.* (2000). Without the bottom dissipation and wave breaking terms, the energy flow into the shallow water region may remain large and would artificially amplify at the coast. FUNWAVE has been validated based on case studies of a pyroclastic flow generated tsunami (Waythomas and Watts, 2003) and several underwater landslide generated tsunamis (Watts *et al.*, 2003). It has also been applied to a debris flow generated tsunami (Walder and Watts, 2003). For co-seismic sources it has been successfully used for simulating the December 26, 2004 Indian Ocean mega-thrust earthquake-triggered tsunami (Grilli *et al.*, 2007; Ioualalen *et al.*, 2007) and the November 26, 1999 Vanuatu tsunami (Ioualalen *et al.*, 2006).

In the numerical simulation, the wave is calculated by discretizing the computational domain into a set of grid points (nodes) and then by solving the system of Boussinesq equations on each node (see Appendix B for details of the numerical procedure). Meanwhile the time integration is performed. For this technique, the more grid points are used in the problem, the finer is the spatial resolution and the more accurate is the simulation. For example, Ioualalen *et al.* (2007) used a 1/4' grid spacing (~450 m) in order to obtain a reasonably accurate simulation of the impact of the December 26, 2004 Indian Ocean tsunami on the Andaman coast of Thailand. Such grid spacing may appear to be large, however, considering the size of the initial tsunami wavelength (~150 km) Ioualalen *et al.* (2007) showed that it is acceptable for this case study. For this resolution, a 2,383 x 2,017 - nodes computational domain has been constructed and a best-fit 0.5 sec. time step has been chosen to avoid numerical instabilities. This simulation takes about 120 hours to cover only 5 hours of actual propagation. Their results, compared to observations, are coherent but the simulation still requires a smaller grid size. Such computational characteristics are relatively large (especially for the Boussinesq model used here). The statement also applies for



the numerical simulations performed by Ioualalen *et al.* (2006) for the November 26, 1999 Vanuatu earthquake/tsunami that occurred in Central Vanuatu, South-West Pacific. These simulations were less recent and the authors used an even smaller grid (~155 m with a subsequent 0.25 sec. time step). Considering the relatively small size of the initial tsunami wavelength (~20 km), the authors have chosen a finer grid-spacing compared to the Sumatra case study. They showed also that their grid spacing is acceptable, however, they had to limit drastically the size of their computational domain. As a consequence, they did not include available data that were necessary to constrain the model, in particular a tide gauge record that was located far from the earthquake hypocenter.

Finally, a reliable model cannot be efficient if computational facilities do not allow a sufficiently small grid spacing (and thus time step) for any specific event. Considering the fast development of clusters that are mounted locally, there is a real need to parallelize numerical codes. This is the main objective of the present work.

### **1.3 Objective and Scope of the Thesis**

The goal of this thesis was to parallelize the original FUNWAVE sequential model for efficient simulation of long wave propagation, coastal inundation and runup of the November 26, 1999 Vanuatu and the December 26, 2004 Indian Ocean tsunamis. The gain in computational time and memory usage would be straightforward compared to a simulation operated in a sequential mode.

In this thesis, we use domain decomposition technique which is common in parallelizing scientific problems. The strategy subdivides the data domain of a problem into multiple regions and assigns different processors to compute the results for each region. This type of decomposition typically leads to a single-program, multiple-data (SPMD) structure, with each processor executing the same code on different data points. Each processor exchanges the data via message passing interface (MPI). Overlapping communication is used to minimize idle time within processes. Tridiagonal systems are solved by using parallel pipelining tridiagonal solver. We examine accuracy of the parallel code by comparing runups (point by point) with the original sequential code and test performance of the parallel implementation with two measurements: speedup and efficiency.

#### 1.4 Outline of the Thesis

This thesis is organized as follows. Chapter II describes the descriptions of the two tsunami case studies: the November 26, 1999 Vanuatu and the December 26, 2004 Indian Ocean tsunamis. The key aspects of parallel computation are given in Chapter III. Implementation of the parallel program is described in Chapter IV. In Chapter V, experimental results are presented and discussed. Conclusion is given in Chapter VI.



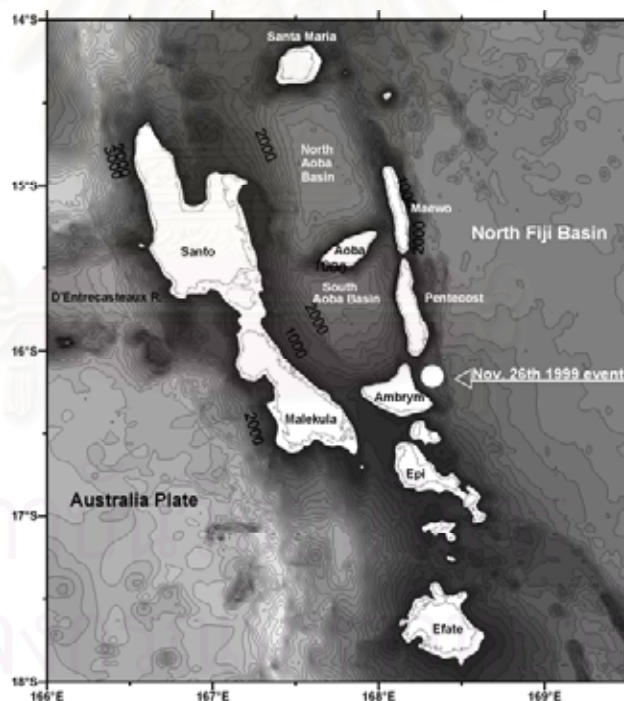
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## CHAPTER II

### Descriptions of the Tsunami Case Studies

Numerical simulations in sequential mode for the Vanuatu and Indian Ocean tsunamis were performed by Ioualalen *et al.* (2006) and Ioualalen *et al.* (2007), respectively, with FUNWAVE. Their simulated runups were in good agreement with observations and eyewitness reports. In the present work, these two cases are tested for parallel simulations to evaluate performance of the parallelized code. The parallel model is validated by comparing the runups with those obtained from sequential mode. We describe below the characteristics of these tsunami events with brief descriptions of the related earthquakes.

#### 2.1 The November 26, 1999 Vanuatu Earthquake/Tsunami



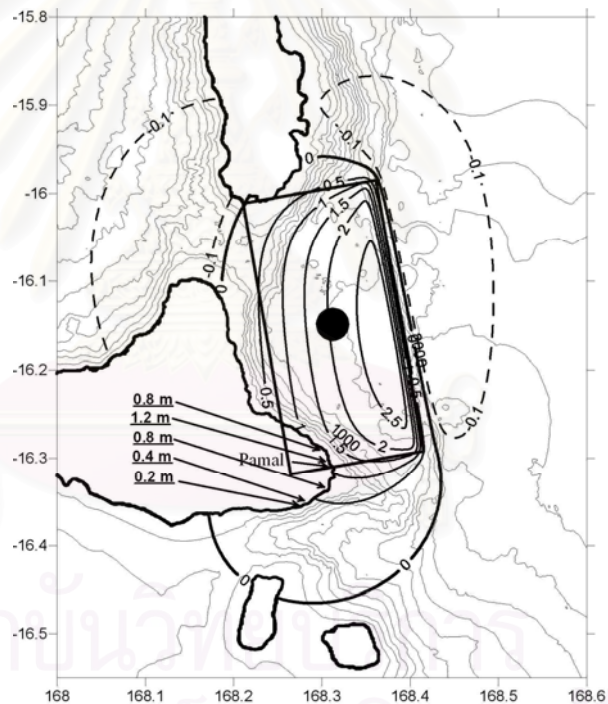
**Figure 2.1: Central Vanuatu Area.** Bathymetry of the area (200 m iso-intervals) along with the position of the November 26, 1999 earthquake

On 26 November 1999, central Vanuatu was struck by a large earthquake ( $M_w = 7.5$ ) followed by a tsunami that killed five people and caused damages in nearshore structures. From the observations of the post-tsunami surveys, the maximum runup was approximately 6 - 7 m at Martelli Bay, south Pentecost Island. Even if the tsunami arrived within only 10 min after the earthquake and occurred during the night, no serious damages had been reported. This was mainly because of a wedding party that kept the residents close to the nearshore area and the receding water that warned them to run away from the shore line. Here are some details on this earthquake and tsunami (Ioualalen *et al.*, 2006).

Quantities	Fault
$x_o$ (longitude)	168.31°
$y_o$ (latitude)	-16.15°
$d$ (km)	7.5
$\varphi$ (degrees)	170°
$\lambda$ (degrees)	64°
$\delta$ (degrees)	140°
$\Delta$ (m)	6.5
$L$ (km)	35
$W$ (km)	20
$\mu$ (Pa)	$3.5 \times 10^{10}$
$M_o$ (J)	$1.35 \times 10^{20}$
$\lambda_o$ (km)	19.352
$\eta_o$ (m)	2.783

**Table 2.1: 1999 Vanuatu fault tsunami source parameters.** Okada (1985) rupture parameters. The inputs are (in descending order): the longitude of the earthquake centroid  $x_o$ , the latitude of the earthquake centroid  $y_o$ , the centroid depth  $d$ , the fault strike clockwise from north  $\varphi$ , the fault rake counter-clockwise from strike  $\lambda$ , the fault dip  $\delta$  (positive from the horizontal plane) and dip counted clockwise, the maximum slip  $\Delta$ , the fault length along rupture  $L$ , the fault width across rupture  $W$ , and the shear modulus  $\mu$ . The outputs are the seismic moment  $M_o$ , the characteristic wavelength  $\lambda_o$ , and the characteristic tsunami amplitude  $\eta_o$ .

Across Central Vanuatu between 14°S and 17°S, the New Hebrides subduction zone was composed of the fore-arc belt (Espiritu Santo and Malekula islands) which overthrust westward the subducting Australian plate, the intra-arc area which was mainly composed of the deep Aoba Basin and three active volcanoes (Santa Maria, Aoba and Ambrym islands), and finally the back-arc belt (Maewo and Pentecost islands) which backthrust eastward onto the North Fiji Basin oceanic crust (Figure 2.1). This peculiar morphology of the central part of the Vanuatu arc (termed also the New Hebrides arc) was considered to be a direct consequence of the subduction/collision of the D'Entrecasteaux Ridge on the Australian plate (Collot *et al.*, 1985; Taylor *et al.*, 1995; Louat and Pelletier, 1989; Pelletier *et al.*, 1994, 1998). Both the western and eastern belts had been uplifted and were still subjected to present days positive vertical motion (Taylor *et al.*, 1995; Lagabrielle *et al.*, 2003).

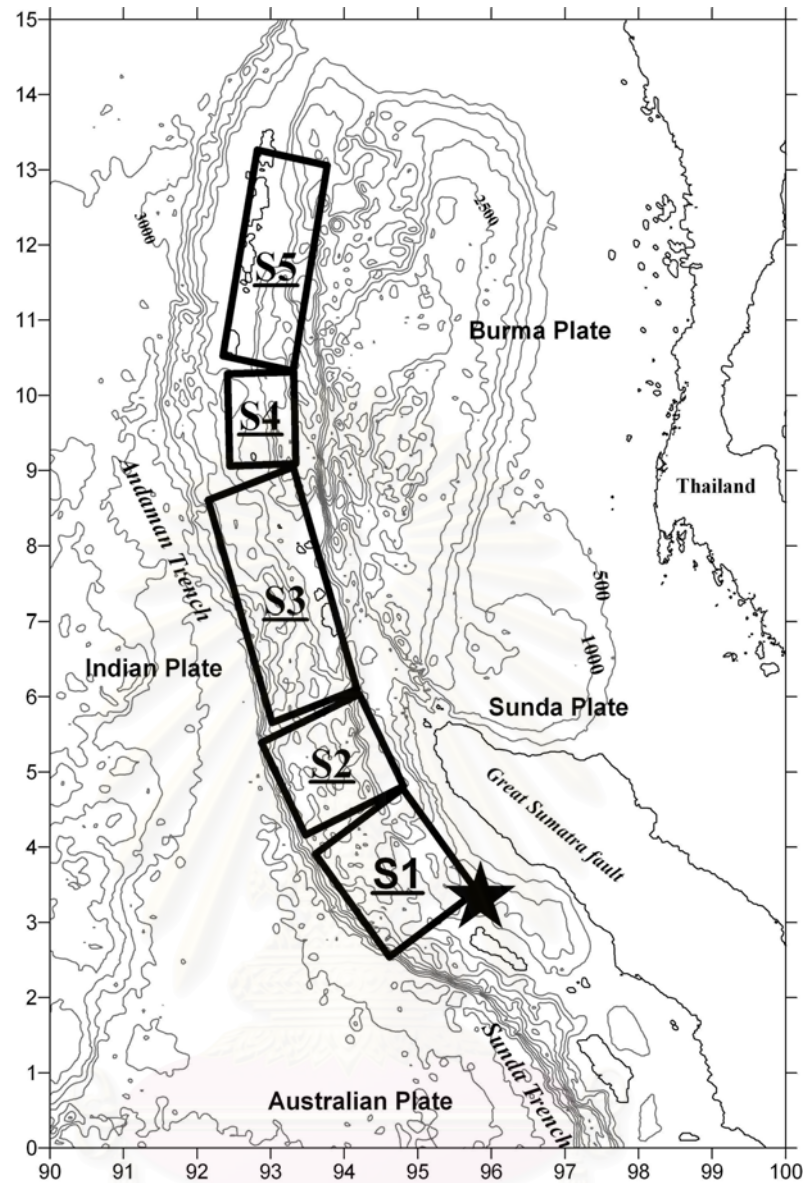


**Figure 2.2: Simulated vertical displacements.** The displacements are plotted in meters with Okada's (1985) dislocation model in the vicinity of the earthquake epicenter vertical along with the bathymetry (Regnier, *et al.*, 2003). The rupture parameters are displayed in Table 2.1. Positives values represent uplift (positive initial sea surface elevation) and negative values are set for subsidence (initial sea surface depression). The vertical motions estimated by Pelletier *et al.* (2000) though field surveys are reported (underlined).

The November 26, 1999 earthquake was located in the southern part of the back-arc belt east of Ambrym island (Regnier *et al.*, 2003). Co-seismic vertical motion (uplift as high as 1.2 m -and even a little higher at 1.5 m) and subsidence had been measured along the eastern shore of Ambrym and observed later around surrounding islands (Pelletier *et al.*, 2000; Lagabrielle *et al.*, 2003). The best fit solution using a uniform rectangular fault was obtained with a 35×20 km fault located at 168°31'E, 16.15°S, 7.5 km with the set of parameters displayed in table 2.1 (Regnier *et al.*, 2003). The solution showed good agreement as compared with the vertical motions (see Figure 2.2).

## 2.2 The December 26, 2004 Indian Ocean Earthquake/Tsunami

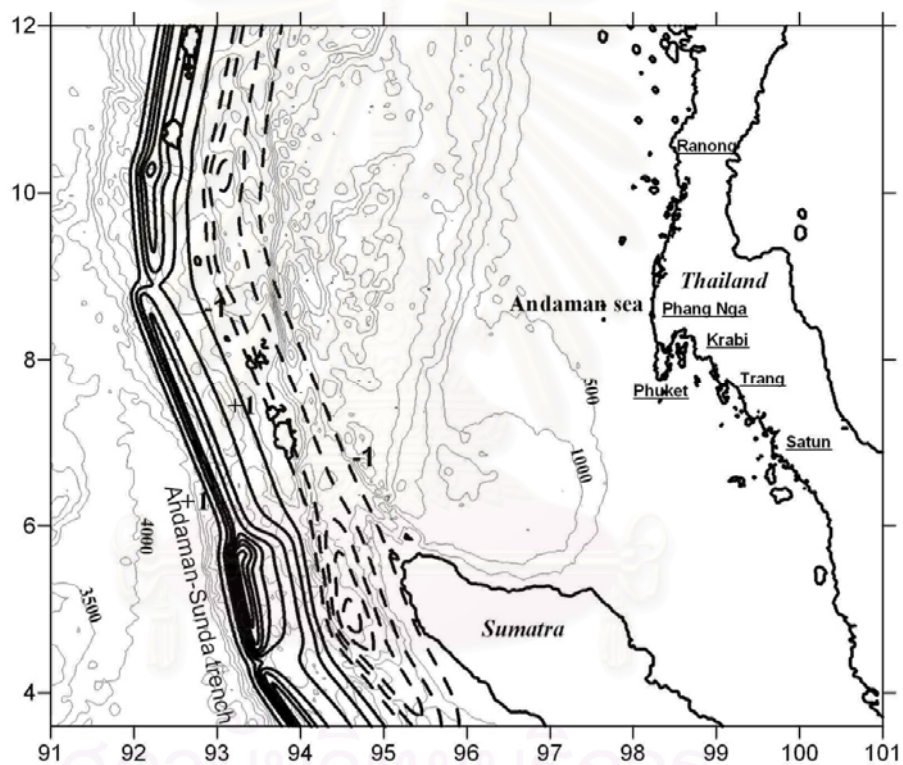
On 26 December 2004, the mega earthquake of magnitude  $M_w = 9.2$  occurred at a hypocentral depth of around 25-30 km from the surface. The epicenter was located at 3.32° Latitude N and 95.58° Longitude E. The total rupture length was around 1,200-1,300 km requiring less than 10 min for the rupture to propagate end to end. The earthquake triggered a tsunami that caused so much damage in more than 10 countries across the entire Indian Ocean basin. Over 200,000 people were killed with tens of thousands reported missing. Below, we briefly present some details of this tsunami event (Ioualalen *et al.*, 2007).



**Figure 2.3: Northern Sumatra area.** ETOPO2 bathymetry around the December 26, 2004 earthquake hypocenter (★) contoured at 500 m intervals. Rectangles S1 - S5 represent the Okada (1985) dislocation model fault segments calibrated by Grilli *et al.* (2007) and Ioualalen *et al.* (2007) see Table 2.2.

The relative motion between the Indian and Sunda Plates is on the order of 4 cm per year in direction  $20^{\circ}\text{N}$  while, between the Australian and Sunda plates, it is on the order of 5 cm per year in direction  $8^{\circ}\text{N}$  (Socquet *et al.*, 2006) (Fig. 2.3). The December 26, 2004  $M_w \sim 9.3$  megathrust earthquake (Stein and Okal, 2005) was a consequence of strain accumulated in the Indian/Sunda junction, some of which had not experienced a large earthquake for the past 150 years or so. Recent large events in the region include  $M_w \sim 8.4$  in 1797,  $M_w \sim 9$  in 1833, and  $M_w \sim 8.5$  in 1861, for the

Australian/Sunda boundary, and weaker  $M_w \sim 7.9$  events for the Indian/Sunda boundary in 1881 and 1941 (Lay *et al.*, 2005). This unbalanced partition of past earthquake magnitudes and recurrence times between the two plate boundaries indicated that larger strains had accumulated in the Indian/Sunda boundary prior to the December 26, 2004 event, and explained both the epicenter location at the junction between the subducting Indian and Australian plates and the overriding Eurasian plate (Burma and Sunda subplates) and the northward rupture propagation, where most of the aftershocks were recorded along a  $\sim 1300$  km arc of the Andaman trench (Lay *et al.*, 2005).



**Figure 2.4: Initial surface elevation for the 2004 Indian Ocean tsunami.** The tsunami source developed by Grilli *et al.* (2007) and Ioualalen *et al.* (2007) is based on five Okada's (1985) dislocation segments (S1-S5; Table 2.2). Continuous lines represent sea floor uplift (initial tsunami wave crest) and dashed lines represent subsidence (initial wave depression), both at 1 m contour intervals in the range -5 to +8 m. The background bathymetry is plotted in grey at 500 m contour intervals. The 6 exposed provinces of the Andaman coast of Thailand are underlined (Ioualalen *et al.*, 2007).



The focal mechanisms used here for the earthquake are those proposed by Ioualalen *et al.* (2007) and Grilli *et al.* (2007). Their 5-segment solution had been calibrated using available tide gauge records in the Indian Ocean and JASON-1 anomaly of sea level (Table 2.2 for the tsunami source characteristics and Figure 2.4 for the initial wave, e.g., the sea floor deformation).

Parameters	S1	S2	S3	S4	S5
$\tau(s)$	60	272	588	913	1273
$x_o, y_o$	94.57°E, 3.83°N	93.90°E, 5.22°N	93.21°E, 7.41°N	92.60°E, 9.70°N	92.87°E, 11.70°N
$\phi$	323°	348°	338°	356°	10°
$\Delta$ (m)	18	23	12	12	12
$L, W$ (km)	220, 130	150, 130	390, 120	150, 95	35°, 95
$M_o$ (J)	$1.85 \times 10^{22}$	$1.58 \times 10^{22}$	$2.05 \times 10^{22}$	$0.61 \times 10^{22}$	$1.46 \times 10^{22}$
$\lambda_o$ (km)	130	130	120	95	95
$\tau_o$ (min)	24.77	17.46	23.30	18.72	18.72
$\eta_o$ (m)	-3.27, +7.02	-3.84, +8.59	-2.33, +4.72	-2.08, +4.49	-2.31, +4.60

**Table 2.2: 2004 Indian Ocean tsunami source parameters.** Pure thrust reverse fault solution dipping eastward (Grilli *et al.*, 2007 ; Ioualalen *et al.*, 2007). Input parameters for Okada (1985) dislocation method (first 5 lines) and outputs (last 4 lines) for 5 tsunami source segments (Figure 2.3): time delay of segment rupture from earthquake time  $t$  (a 60 s rising time is added); longitude and latitude of segment centroid ( $x_o, y_o$ ); the centroid depth is  $d = 25\text{km}$  for all segments, the fault strike angle  $f$  (clockwise from North); the fault rake angle is  $l = l_o = 90^\circ$  for all segments (counterclock-wise from strike); the fault dip angle is  $d = 168^\circ$  (positive from the horizontal plane); the maximum fault slip  $\Delta$ ; the segment length along and width across ( $L, W$ ); and the medium shear modulus taken  $m = 4 \times 10^{10}$  Pa for all segments; the seismic moment  $M_o$ ; the characteristic initial tsunami wavelength  $\lambda_o$  and period  $t_o$ ; and the characteristic tsunami trough and peak amplitudes  $h_o$ .

## CHAPTER III

### PARALLEL COMPUTATION

Parallel computing has increasingly become an important component of the computing technology. In large-scale problems such as climate forecasting and ocean modeling, the computation on a single-processor computer is forbidden due to the limitation on the processing speed and the size of local memory and storage. To overcome this obstacle, parallel computation has been widely used and developed on both parallel hardware and software including techniques to implement parallel program. We provide below the issues involved in parallel computer architectures and introduction of message passing interface (MPI). In addition, we present techniques and algorithms for domain decompositions, overlapping communications and parallel tridiagonal solver.

#### 3.1 Parallel Computer Architectures

A parallel computer is a collection of processing elements/processors that cooperate to solve large scale problems with improved performance. Hence, parallel computing refers to the use of multiple processors to reduce the computing time, and it requires computer hardware that supports multiple processors. Several architectural approaches to parallelism have been developed over the years, such as shared-memory, distributed-memory and distributed-shared memory architectures (Dongarra *et al.*, 2003).

##### 3.1.1 Shared-Memory Architecture

In the shared-memory architecture (Figure 3.1), there is a large block of random access memory that can be accessed by several different CPUs in a multiple-processor computer system. It is based on a method of inter-process communication, which is a way of exchanging data between programs running at the same time. The advantage of a shared memory system is that it is relatively easy to program since there are no explicit communications between processors. The communications between processors can be as fast as memory accesses to a same location.

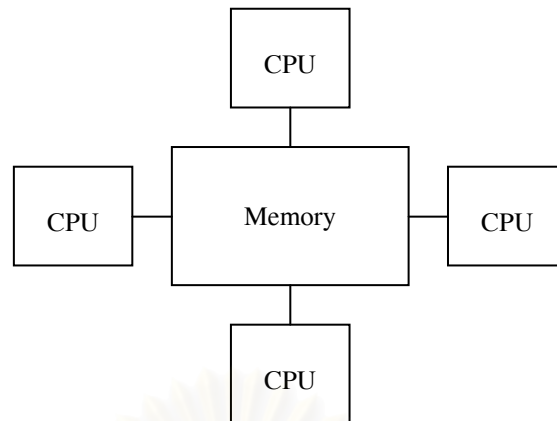


Figure 3.1: Shared-Memory Architecture

However, the shared memory architecture does not scale well. The main problem occurs when a number of processors attempt to access the global memory store at the same time. One method of avoiding this memory access complication is to divide the memory into multiple memory modules in a way that each module is connected to the processors via a high performance switching network.

### 3.1.2 Distributed-Memory Architecture

For distributed-memory architecture (shown in Figure 3.2), each processor has only access to the memory which is directly attached to it. If one processor requires data exchanging in the memory of a remote processor, it must send a message to the remote processor to establish the communication network. Thus, this system is often referred to as a message passing multiprocessor.

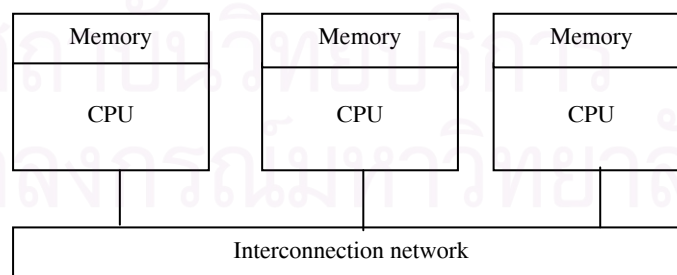


Figure 3.2: Distributed-Memory Architecture

The advantages of this type of architecture are that each processor can rapidly access its own memory, and the memory is scalable with the number of processors. If the number of processors increases, the size of memory increases proportionally.

### 3.1.3 Distributed-Shared Memory Architecture

The computer architecture widely used today employs both shared and distributed memory architectures for keeping the advantage of both. It is referred to a wide class of hardware and software implementations, in which each node of a system has access to a large shared memory in addition to each node's limited non-shared memory store.

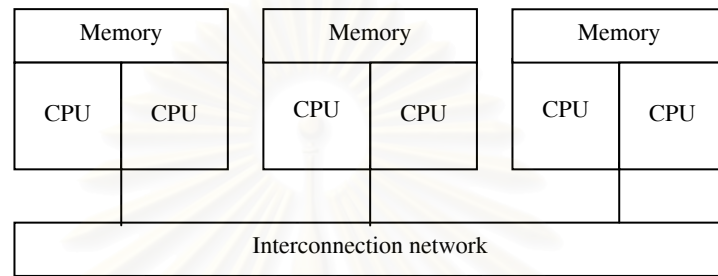


Figure 3.3: Distributed-Shared Memory Architecture

Performance of the interconnection network is important in this parallel architecture. The performance is measured in two dimensions: bandwidth and latency. Bandwidth is the rate at which the data can be moved between nodes and is measured in megabytes (MB) per second. Latency is defined as the time spent in setting up access to a remote node. If the processor does one thing at a time, the bandwidth is proportional to inverse of the latency. When this system is configured with a large number of nodes, the limitations of the network can degrade the performance. The bandwidth and latency of the cluster interconnection is the key to improving the scalability of computer cluster.

### 3.2 Message Passing Interface (MPI)

MPI is the library of subroutines inserted into the parallel source code to perform data communication between processors. The goals addressed by MPI are to provide source code portability and allow efficient implementations across a range of parallel computing architectures. Moreover, it includes a number of different types of communication, special routines for common collective operations, and the ability to handle user-defined data type and topologies.

### 3.3 Domain Decomposition

Domain decomposition appears to be a more common approach. It has been used even before parallel processing came into use. The domain decomposition was introduced to allow for applications of large scale models on a computer with limited memory (Hendrik, 2002). In parallel computation, a simple way to implement domain decomposition is to divide the domain into several smaller sub-domains for which the number of sub-domains is equal to the number of processors. To decompose the domain, there are several possible ways. The domain can be decomposed in vertical or horizontal direction, which is called as 1D decomposition, and if both two directions are decomposed, it is called as 2D decomposition. For example, if the 4 processors are used, the domain can be decomposed into three possible ways as shown in Figure 3.4.



Figure 3.4: Domain decomposition topologies for 1D and 2D decompositions of 4 processors

The advantage of domain decomposition is that it is relatively simple to implement, particularly when the standard MPI routines can be used. The main issue in decomposing a domain is the minimization of communication and the load balancing task in each processor. Load balancing for a parallel system is one of the most important problems which have to be solved in order to enable the efficient use of parallel computer systems. The goal of load balancing strategies is to balance the load of all processors in the system by exchanging work only between directly connected processors. Thus, in a parallel environment, the work must be assigned between processors in order to get optimal resource utilization, throughput, and response time.

### 3.4 Overlapping Communication

In the parallel computation, the original domain is decomposed into smaller sub-domains. For problems which are solved by numerical techniques, such as finite difference schemes, calculations at each grid point require information from its neighbors. So, at the boundaries of sub-domain, data are interchanged between contiguous processors. To efficiently exchange the data between processors, the data, which consist of arrays of horizontal and/or vertical grid points, are first stored in a contiguous memory prior to execute the sending processes (Sitanggang and Lynett, 2005). At the same time contiguous memories of the same size as used in the sending processes are created to receive the data from the sending processes. At this point, the data are ready for sending and receiving processes.

Figure 3.5 shows an example of the computational sub-domains for which each is composed of two areas: the gray-shade and the white-shade areas. The gray-shade area is the computational domain, whereas the white-shade area is the overlapping zone. The latter is the zone at which the overlapping data are stored and interchanged with the neighboring processors. After obtaining new values, at each time step, the overlapping areas of the sub-domain must be updated.

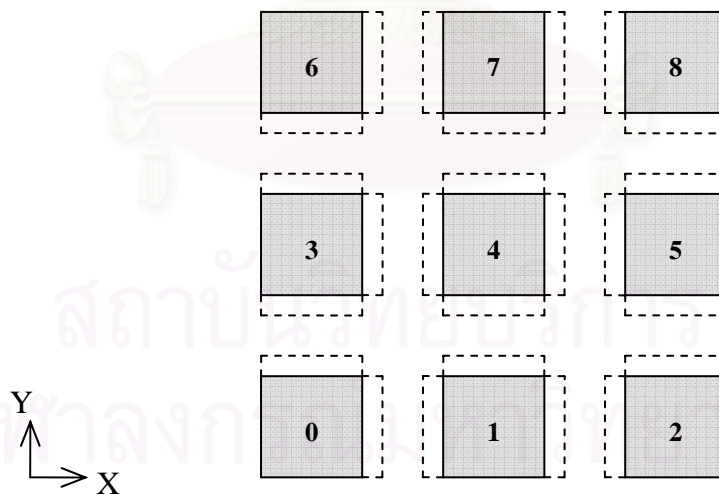


Figure 3.5: The computational domain with overlapping areas

In the updating procedure, number of communication is generally a function of the number of processors in each direction. Let  $np_x$  be the number of column processors in X direction and  $np_y$  be the number of row processors in Y direction. The number of communication is defined as

$$N_{com} = np_y \times (np_x - 1) + np_x \times (np_y - 1) \quad (3.1)$$

For example, the number of communication for a 3×3-decomposition with 9 processors is 12. Number of grid points in the overlapping zone depends on number of processors and grid points in the global domain in each direction. Thus, the number of grid points in overlapping area can be defined as

$$N_{gp} = a \times ny \times (np_x - 1) + a \times mx \times (np_y - 1) \quad (3.2)$$

where  $a$  is the number of rows or columns in overlapping zone,  $mx$  is the global number of grid points in X direction and  $ny$  is the global number of grid points in Y direction.

### 3.5 Parallel Tridiagonal Solver

For sequential algorithm, the tridiagonal matrix systems are usually solved by Thomas algorithm on a single computer with three-step processes: LU decomposition, forward substitution and backward substitution. For LU decomposition and forward substitution, each process starts the calculation from the first element to the last and then in the reverse direction for backward substitution. During this process, each element has to wait for the previous calculated data. Consequently, the matrix system must be carried out in sequential order according to dependencies of the data among processes so it is difficult to efficiently parallelize. For this reason, parallel algorithms for solving tridiagonal systems have been proposed and developed (e.g. Wang, 1981; Mattor *et al.*, 1995; Xian-He *et al.*, 2004) so that it can be used for an efficient implementation on parallel computer.

For the tridiagonal system, there are many independent matrices which must be solved at the same time. Figure 3.6 shows the discretized domain with unknown variables of one system that requires the tridiagonal solver in X direction. Each line is replaced by a matrix as shown for example in equation (3.3). Each matrix in the system is independent from the others.

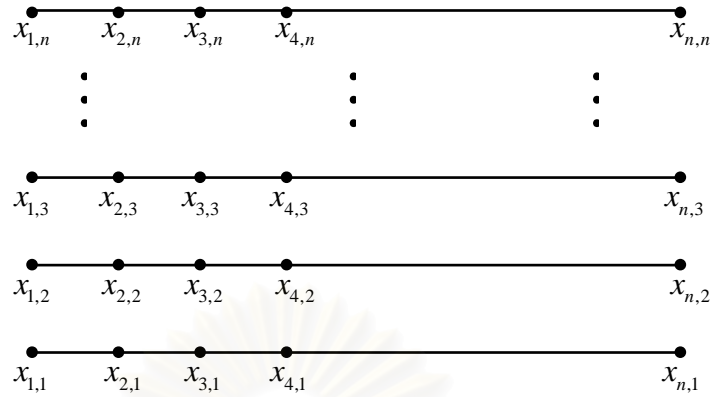


Figure 3.6: The discretized domain with unknown variables in X direction

$$\begin{bmatrix} a_1 & c_1 & 0 & 0 & 0 \\ b_1 & a_2 & c_2 & \ddots & 0 \\ 0 & b_2 & a_3 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & c_{n-1} \\ 0 & 0 & 0 & b_{n-1} & a_n \end{bmatrix} \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ x_{3,1} \\ \vdots \\ x_{n,1} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix} \quad (3.3)$$

To solve a tridiagonal matrix system by pipelining (Garg and Sharapov, 2001), the system is divided into subsets, for which each subset is solved by using the Thomas algorithm. The partial solutions of a subset of the system are sent to the next processor before solving another.

As an example, suppose we want to use the tridiagonal solver with 4 processors in X direction as shown in Figure 3.7. In the first stage of pipelining scheme, first part of the first L system is solved with Thomas algorithm by processor 0 while processors 1 to 3 are idle. When the first stage finishes, processor 0 sends the requiring message to processor 1. In the second stage, processor 1 works on the second part of the first L system, while processor 0 works on the first part of the second L system and processors 2 and 3 are idle. The process continues until finishing the work.



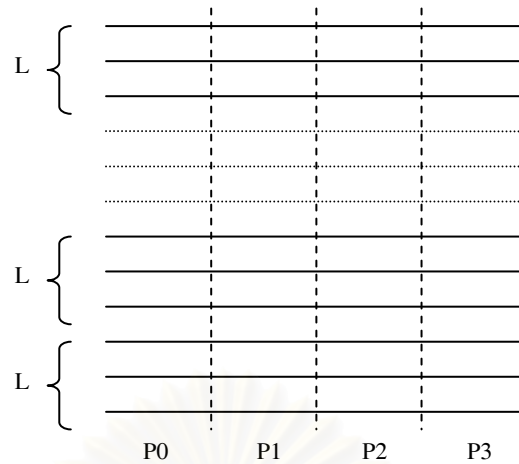


Figure 3.7: The pipelining scheme for 4 processors

To obtain good performance in solving the tridiagonal matrix system, two important sources must be considered. The first source is the processor idle time associated with the pipeline being filled or emptied. This problem can be resolved by reducing the block size of  $L$ . The second source is the communication between processors which is dominated by the message latency. This problem can be avoided by setting the block size of  $L$  to be sufficiently large. The balance of both sources depends on the hardware of the parallel computer.

### 3.6 Parallel Performance Measurements

The most common measurements of the performance of parallel implementations are speedup ( $S$ ) and efficiency ( $E$ ). Here, the speedup achieved by the parallel algorithm running on  $P$  processors is defined as the ratio of the execution time ( $T_s$ ) of the sequential algorithm on a single processor and the execution time ( $T_p$ ) of the parallel algorithm on  $P$  processors. The speedup can be expressed as

$$S = \frac{T_s}{T_p} \quad (3.4)$$

The efficiency is a measure of the time that each processor spends in the computational phase. It is defined as the ratio of speedup to the number of processors, which can be expressed as

$$E = \frac{S}{P} \quad (3.5)$$

Ideally if a program is running on  $P$  processors, the goal is  $P$  times faster than on a single processor. That is  $S = P$  and  $E = 1$ . In practice, this is difficult to achieve due to overheads such as communication time, idle time and time to define local variables for running the program with more than one processor.



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## CHAPTER IV

### PARALLEL IMPLEMENTATION

In this chapter, we describe the parallel implementation of tsunami simulation. We begin with the flowchart of main program with details and then dive into two main components of the program: tsunami source and wave propagation.

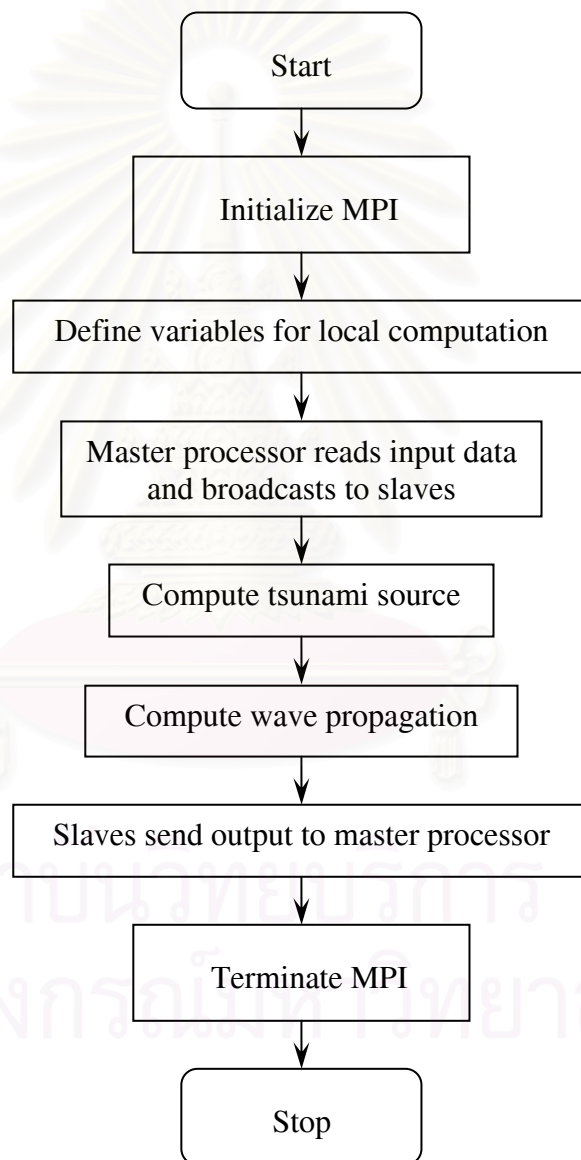


Figure 4.1: The flowchart of the program

First step in parallel implementation is the initialization of MPI to set the identification (ID) for each processor by using three following functions.

`MPI_INIT(ierr)`

*Initiate the computation*

Output *ierr*            error flag

`MPI_COMM_RANK(comm, rank, ierr)`

*Determine the identification of current processor*

Input *comm*            communicator handle

Output *rank*            processor ID in the group of *comm*

`MPI_COMM_SIZE(comm, size, ierr)`

*Determine the number of processors in the computation*

Input *comm*            communicator handle

Output *size*            number of processors in the group of  
*comm*

In this program, some computations or communications perform on only row processors or column processors or boundary of the domain. Here we use the idea of virtual mapping to transform the processors into 2D Cartesian topology with following functions.

`MPI_COMM_CREATE(old_comm, ndims, dim_size, periods, reorder, new_com, ierr)`

*Create a new communicator using the Cartesian topology*

Input *old\_comm*        communicator handle

Input *ndims*            number of dimensions

Output *dim\_size*        array of size *ndims* providing length in  
each dimension

Input *periods*         array of size *ndims* specifying periodicity  
status of each dimension

Input *reorder*         whether process rank reordering by MPI  
is permitted

Output *new\_com*        communicator handle

Then the following function is used to return the corresponding Cartesian coordinate of a linear rank in the Cartesian communicator.

`MPI_COMM_COORDS(comm, rank, maxdims, coords, ierr)`

*return the corresponding Cartesian coordinate of a linear rank*

Input	<i>comm</i>	communicator handle
Input	<i>rank</i>	calling process rank
Input	<i>maxdims</i>	number of dimensions in Cartesian topology
Output	<i>coords</i>	corresponding Cartesian coordinates of <i>rank</i>

After setting the Cartesian topology, we obtain new ID of communicator and it is now ready to start the parallel computation. Figure 4.2 shows an example of the 9 processors mapped into 2D Cartesian topology.

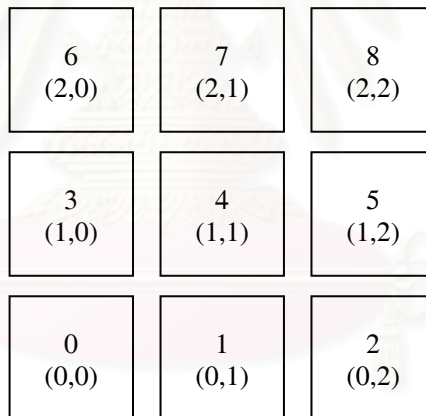


Figure 4.2: the 2D Cartesian topology

Next, an index of arrays, in which the variables are stored, is defined for local computation by dividing the number of grid points with the number of processors in each direction. If the number of grid points is not divisible by the number of processors, the remainder will be distributed to the first  $m$  processors where  $m$  is the remainder.

Prior to the simulation, the master processor (processor 0) has to send portions of input data to its slaves. Then, all processors perform the computation of tsunami source and wave propagation which is described in sections 4.1 and 4.2 respectively. Output data from each processor is sent back to the master processor by using the algorithm shown in Figure 4.3 (for the case of 9 processors).

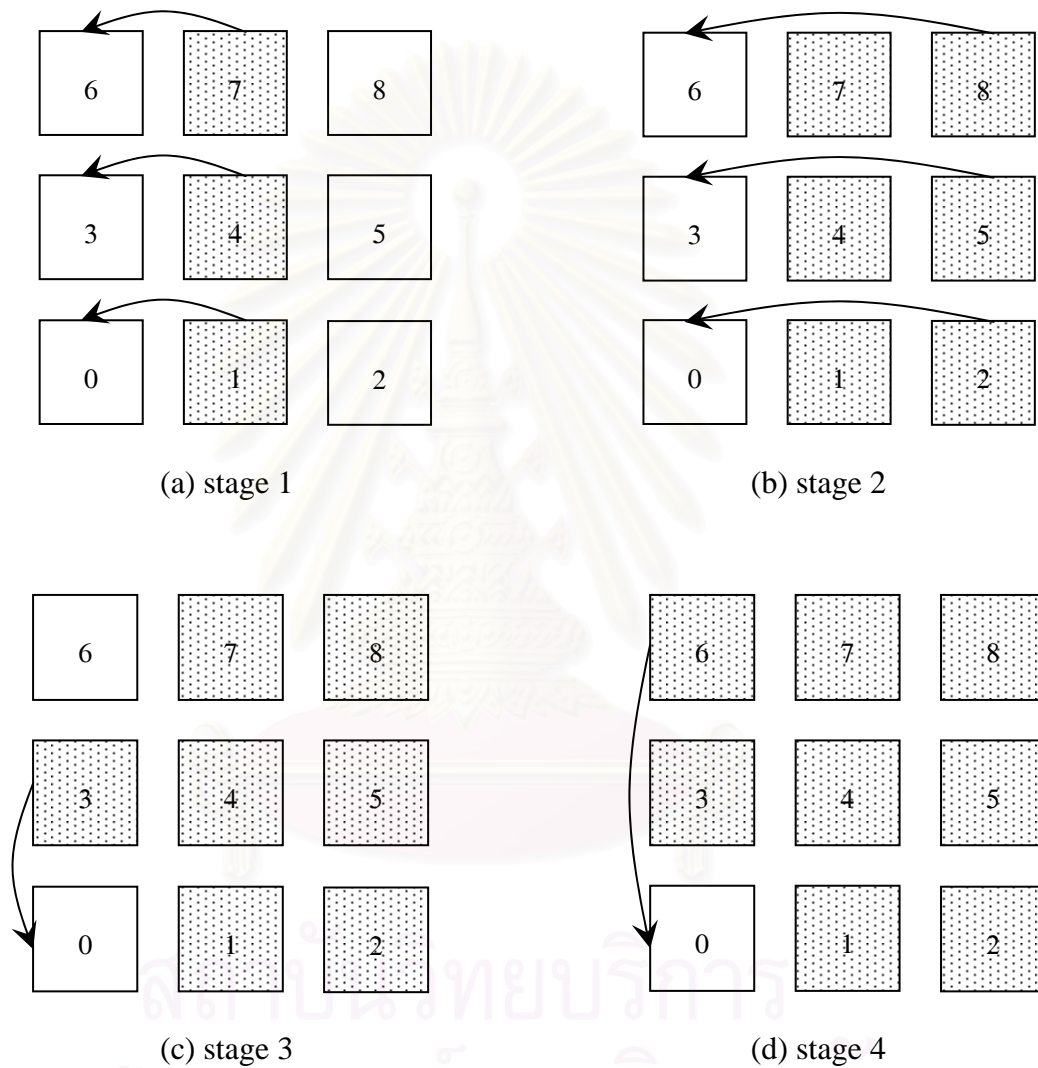


Figure 4.3: The algorithm for sending output information to the master processor

#### 4.1 Parallel Implementation of Tsunami Source

To generate the earthquake tsunami source, Okada's solution (Okada, 1985) has been implemented and parallelized. This solution provides an initial surface elevation for the wave propagation model, which is based on the half-plane solution of an elastic dislocation problem. A planar rectangular fault is discretized into many small trapezoids. The point source of Okada is used to summing the contributions made by each trapezoid to vertical co-seismic displacement, based on the actual depth of the trapezoid. Thus, at every grid point, the initial surface elevation is calculated with the contribution from the discretized rectangular plate depending on their mutual distance. Consequently, the grid points of the domain are independent to each other: they depend only on their distance and orientation from the rectangular plate and on the faulting parameters. The calculations are composed of 4 Fortran *do*-loops. The two outer loops are for computation in the domain and the two inner loops are for rectangular plate as shown in Table 4.1, where  $mx$  and  $ny$  are the number of grid points in X and Y directions of the domain,  $mmx$  and  $nny$  are the number of grid points in X and Y directions of the rectangular plate,  $np_x$  and  $np_y$  are the number of processors in X and Y directions, respectively.

Source code
<pre> do i = 1, <math>\frac{mx}{np_x}</math> do j = 1, <math>\frac{ny}{np_y}</math> do k = 1, <math>mmx</math> do l = 1, <math>nny</math> -- Instruction -- enddo enddo enddo enddo </pre>

Table 4.1: Text of parallel code

Since data at the grid points are independent of each other and require no communication between processors, implementation of the tsunami source can be efficiently parallelized. The parallelization is basically to separate a loop of global computation into local computation for each processor. The output of tsunami source component is then transferred directly to tsunami propagation component as initial condition for the model.

#### 4.2 Parallel Implementation of Wave Propagation

The original FUNWAVE model, developed at the University of Delaware, is parallelized for parallel implementation of tsunami propagation. Governing equations for this wave model are based on the Boussinesq system (see Appendix A), which is modified with the extensions to include bottom friction, wave breaking, moving shoreline and subgrid turbulence effects. It is discretized and solved by finite difference schemes (see Appendix B). Algorithm for parallel implementation of tsunami propagation is given as follows.

For time-step = 1, the initial data of  $u, v$  and  $\eta$  which are obtained as the output from the tsunami source component are used to compute  $\{E', F', G', F_1, G_1\}$  for the first three time step ( $it = -1, 0, 1$ ) at each processor. The computation starts from time-step = 2 with the following processes. All processors have the same task but for different data.

##### 1. Predictor step:

1.1 Predict  $\{U, V, \eta\}^{n+1}$  from  $\{\eta, U, V, E', F', G', F_1, G_1\}$  at time-step  $n, n-1$  and  $n-2$

1.2 Compute  $\{u, v\}^{n+1}$  from  $\{U, V\}^{n+1}$  by solving tridiagonal matrix system with parallel pipelining tridiagonal solver

1.3 Update the new values of  $X = \{u, v, \eta\}^{n+1}$  at overlapping areas

##### 2. Corrector step:

2.1 Compute  $\{E', F', G', F_1, G_1\}^{n+1}$  from  $X = \{u, v, \eta\}^{n+1}$  of predictor step

2.2 Correct  $\{U, V, \eta\}^{n+1}$  from  $\{\eta, U, V, E', F', G', F_1, G_1\}$  at time-step  $n+1, n, n-1$  and  $n-2$



2.3 Compute  $\{u, v\}^{n+1}$  from  $\{U, V\}^{n+1}$  by solving tridiagonal matrix system with parallel pipelining tridiagonal solver

2.4 Update the new values of  $X = \{u, v, \eta\}^{n+1}$  at overlapping areas

2.5 Iterate 2.1 – 2.4 until  $\frac{\|X^* - X\|}{\|X\|} < 10^{-5}$ , where  $X^* = \{u, v, \eta\}^*$  and

$X = \{u, v, \eta\}$  denote the current and previous results, respectively.

3. Compute  $\{E', F', G', F_1, G_1\}^{n+1}$  from  $X = \{u, v, \eta\}^{n+1}$

4. Filter short wave every 20 time steps or when wave breaking occurs to determine a new value at each grid point by using the original value at 9 adjacent points

$$Z_i^* = \frac{1}{256} \left[ 186Z_i + 56(Z_{i+1} + Z_{i-1}) - 28(Z_{i+2} + Z_{i-2}) + 8(Z_{i+3} + Z_{i-3}) - (Z_{i+4} + Z_{i-4}) \right]$$

and update the new values of  $X = \{u, v, \eta\}^{n+1}$  at overlapping areas

5. Reorder the variables by step down one index in time (i.e., from  $n+1$  to  $n$ )

6. Repeat 1-5 until finish computing tsunami propagation component

## CHAPTER V

### EXPERIMENTAL RESULTS AND DISCUSSION

#### 5.1 Parallel Computation for Boussinesq Equations with Gaussian Source Function

In this section, the idealized case is used to test the model on a distributed-memory computer system. The earthquake/tsunami source is replaced with the source term represented by Gaussian function. The initial free surface  $\eta$  is given by  $\eta^0 = 0.5e^{0.05[(x-150)^2+(y-150)^2]}$  as shown in Figure 5.1.

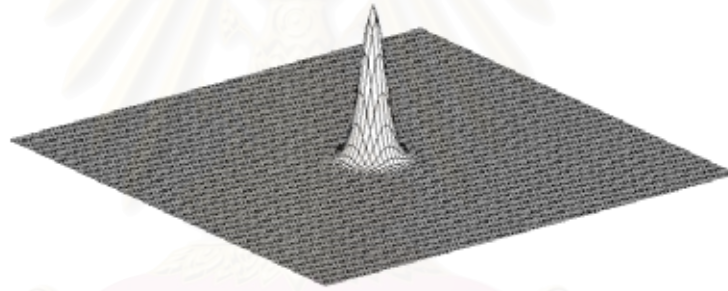


Figure 5.1: Gaussian initial free surface profile

The parallel Boussinesq model is simulated on wave propagation problem with Gaussian initial wave profile in the computational domain of  $300 \text{ m} \times 300 \text{ m}$  with a constant depth of 5 m. Fluid domain is discretized into  $100 \times 100$  numerical grids. The simulation is performed by using  $\Delta t = 0.05$  for 800 steps in time. For this case, the 4-processor PC cluster consisting of PC Pentium III with the 100-Mpbs network speed at the Advanced Virtual and Intelligent Computing Center (AVIC) is set up for this parallel simulation. Simulation results show that waves propagate radially and symmetrically away from the initial source function. The numerical results agree in both cases: sequential and parallel computations. Typical free surface profiles at different times are shown in Figure 5.2. This constitutes a check on the parallel code for the Boussinesq wave model.

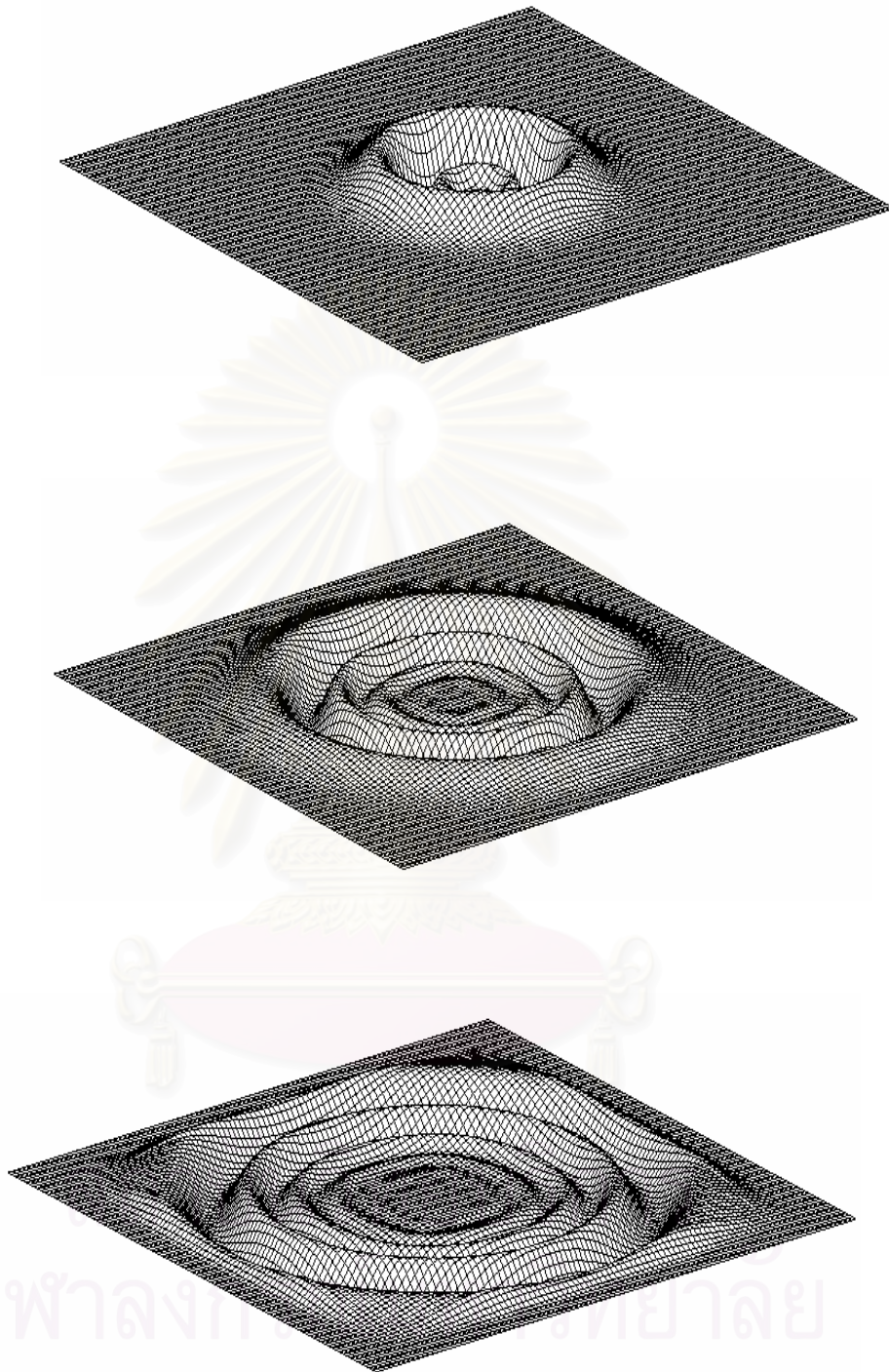


Figure 5.2: The wave profile calculated by using 4-processors  
at time  $t = 15$ , 30 and 45s

## 5.2 Parallel Computation for Tsunami Events

In this section, the developed parallel program is tested on the two referenced cases of tsunamis: the November 26, 1999 Vanuatu and the December 26, 2004 Indian Ocean tsunamis. These two cases are different by size of the affected areas and the magnitude of the coastal impact. The Indian Ocean tsunami was a truly global event with significant coastal impact whereas the Vanuatu tsunami was rather a local event with smaller magnitude of impact. To test parallel performance with the sequential codes from previous studies, the computational domains are kept unchanged from the original sizes.

To investigate and test the performance of the parallel program, numerical simulations of the tsunami events are performed on three clusters with different architectures. First distributed-shared memory cluster (“Xeon”), at Faculty of Science, Chulalongkorn University, has 8 computing nodes with Dual Xeon 3.0 processors. Each node has 2 GBytes of main memory and is connected with 1 Gigabit Ethernet ports. The second cluster (“Itanium”), distributed-shared memory architecture, is located at National Electronics and Computer Technology Center (NECTEC) in Thailand. This “Itanium” cluster has 32 computing nodes with Dual Itanium2 processors. Each node has 4 GBytes of main memory and is connected with 2 Gigabit Ethernet ports. For the shared-memory cluster (“Opteron”), it is located in Geosciences Azur, Villefranche-sur-Mer, France. This cluster is Sun V40z with 4 dual core Opteron processors and 32 GBytes of shared memory. The operating system of each cluster is Linux with MPI version MPICH.

The parallel program is run on computer systems with different number of processors (1, 2, 4, 6, 8, 12 and 16 processors). From the numerical experiments, simulation results from the parallel program are compared with those from the sequential program and are found to be in very good agreement. The performances of the parallel program are presented in the next sections.

### 5.2.1 The November 26, 1999 Vanuatu Tsunami

In this event, the computational domain includes the central Vanuatu ranging from 167.1° to 168.9° E Longitude and 18° to 15.25° S Latitude. Parallel simulations are performed on a 0.5' grid spacing (referred as "VT05"), yielding 208 by 331 numerical grids, with a time step of  $\Delta t = 1.079s$ . The performance results of the tsunami source, tsunami propagation components and the combination of these two components are shown in Table 5.1, 5.2 and 5.3 respectively.

Table 5.1: Performance results of the tsunami source

Number of processors	Elapsed time (s)	Speedup	Efficiency (%)
1	43.80	1.00	100.00
2	22.70	1.93	96.48
4	11.95	3.66	91.61
6	8.40	5.22	86.95
8	6.66	6.58	82.24
12	4.92	8.90	74.20
16	4.02	10.90	68.14

Table 5.2: Performance results of the wave propagation

Number of processors	Elapsed time (s)	Speedup	Efficiency (%)
1	1606.88	1.00	100.00
2	966.80	1.66	83.10
4	585.44	2.74	68.62
6	412.06	3.90	64.99
8	385.44	4.17	52.11
12	333.55	4.82	40.15
16	352.81	4.55	28.47

Table 5.3: Performance results of the tsunami source and the wave propagation

Number of processors	Elapsed time (s)	Speedup	Efficiency (%)
1	1650.68	1.00	100.00
2	989.50	1.67	83.41
4	597.39	2.76	69.08
6	420.46	3.93	65.43
8	392.10	4.21	52.62
12	338.47	4.88	40.64
16	356.83	4.63	28.91

## 5.2.2 The December 26, 2004 Indian Ocean Tsunami

In this case, the computational domain is selected to cover the Andaman coast of Thailand starting from 91° to 101° E in Longitude and 3.6° to 12° N in Latitude. This gigantic tsunami events is performed on both 1' and 0.5' grids (referred to as "IO10" and "IO05"), yielding 596 by 505 points and 1192 by 1009 points, respectively, using a time step of  $\Delta t = 0.5s$ . The performance results for this tsunami event are shown in the following subsections.

### 5.2.2.1 Performance of the 1' grid computational domain

Performance results of the tsunami source component, tsunami propagation component and of the program are shown in Table 5.4, 5.5 and 5.6 respectively.

Table 5.4: Performance results of the tsunami source

Number of processors	Elapsed time (s)	Speedup	Efficiency (%)
1	2816.31	1.00	100.00
2	1434.04	1.96	98.20
4	748.88	3.76	94.02
6	513.77	5.48	91.36
8	416.65	6.76	84.49
12	279.69	10.07	83.91
16	209.79	13.42	83.90

Table 5.5: Performance results of the wave propagation

Number of processors	Elapsed time (s)	Speedup	Efficiency (%)
1	28908.60	1.00	100.00
2	16860.74	1.71	85.73
4	8491.19	3.40	85.11
6	5945.77	4.86	81.03
8	4478.47	6.46	80.69
12	3067.70	9.42	78.53
16	2680.29	10.79	67.41

Table 5.6: Performance results of the tsunami source and the wave propagation

Number of processors	Elapsed time (s)	Speedup	Efficiency (%)
1	31724.91	1.00	100.00
2	18294.78	1.73	86.70
4	9240.07	3.43	85.84
6	6459.54	4.91	81.86
8	4895.11	6.48	81.01
12	3347.39	9.48	78.98
16	2890.08	10.98	68.61

### 5.2.2.2 Performance of the 0.5' grid computational domain

Performance results of the tsunami source component, tsunami propagation component and of the program are shown in Table 5.7, 5.8 and 5.9 respectively.

Table 5.7: Performance results of the tsunami source

Number of processors	Elapsed time (s)	Speedup	Efficiency (%)
1	11257.93	1.00	100.00
2	5679.23	1.98	99.11
4	2904.88	3.88	96.89
6	1966.07	5.73	95.44
8	1498.75	7.51	93.89
12	1030.03	10.93	91.08
16	756.90	14.87	92.96

Table 5.8: Performance results of the wave propagation

Number of processors	Elapsed time (s)	Speedup	Efficiency (%)
1	151183.50	1.00	100.00
2	77217.93	1.96	97.89
4	39175.33	3.86	96.48
6	26180.85	5.77	96.24
8	20076.94	7.53	94.13
12	14678.89	10.30	85.83
16	11965.76	12.63	78.97

Table 5.9: Performance results of the tsunami source and the wave propagation

Number of processors	Elapsed time (s)	Speedup	Efficiency (%)
1	162441.44	1.00	100.00
2	82897.17	1.96	97.98
4	42080.21	3.86	96.51
6	28146.92	5.77	96.19
8	21575.69	7.53	94.11
12	15708.92	10.34	86.17
16	12722.66	12.77	79.80

### 5.2.3 Performance Results for Different Domain Sizes

The scalability of the parallel computation on different domain sizes and number of grid points can be seen from the performance results in the previous subsections. Vanuatu tsunami (VT05) domain's is discretized into the small number of grid points (68,848 numerical grids) and the Indian Ocean tsunami's domain is discretized in a way that the problem can be categorized as the medium (IO10, with 300,980 numerical grids) and the large (IO05, with 1,202,728 numerical grids) scales.

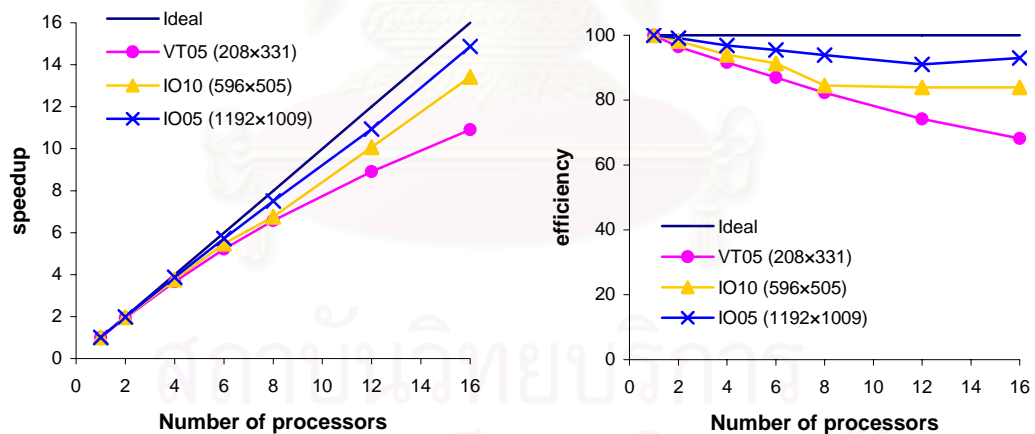


Figure 5.3: Relationship between speedup, efficiency, and the number of processors for the tsunami source



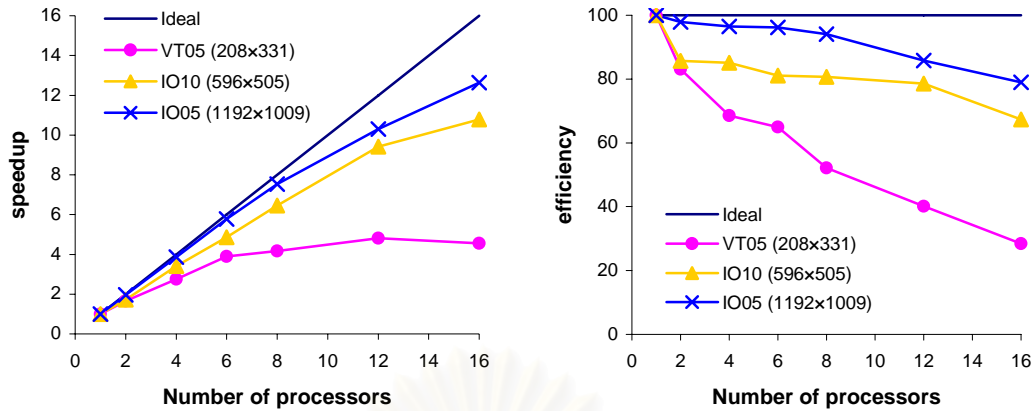


Figure 5.4: Relationship between speedup, efficiency, and number of processors for the wave propagation

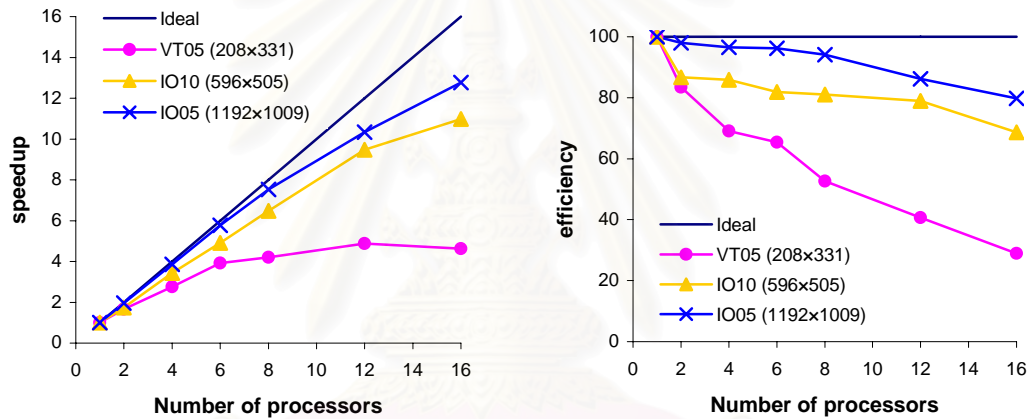


Figure 5.5: Relationship between speedup, efficiency and number of processors for the tsunami source and wave propagation

It can be seen that the performance of parallel computations improves as the number of grid point increases. This is mainly due to the ratio of communication to computation. For large-scale problem, the communication time is small compared with the computation time. However, for small-scale problem, the ratio increases significantly. This ratio can be found by comparing number of grid points in the overlapping zone (see equation (3.2) in Chapter 3) with the computing zone. For example, with 16 processors, the 1,192×1,009 case has 37,412 communicated points and 1,202,728 computed points while the 596×505 case has 18,708 communicate points and 300,980 computed points and the 208×331 case has 6,468 communicated points and 68,484 computed points for which the ratios of communicated points to computed points are 3.11%, 6.21% and 9.44% respectively.

When the number of processors increases, the performance decreases quite significantly and the performance curve is far from the ideal (or linear) response. This is due to the increasing of the overlapping area with the number of processors. To be specific, the communication time increases while the computation time remains unchanged. Generally, performance of a parallel code can be improved if the ratio of communication to computation is kept small.

Tables 5.1, 5.2, 5.4, 5.5, 5.7 and 5.8 show that the performance of all three cases is much better for tsunami source than for the wave propagation part, mainly because tsunami source computation requires no communication. The communication usually takes place when input/output data has to be distributed between computing nodes. Overall computation of earthquake/tsunami simulation is dominated by the wave propagation part, hence the performance of tsunami simulation is resemble those of the wave propagation part as shown in the previous subsections.

#### 5.2.4 Parallel Performance based on Different Domain Decompositions

The effect of domain decomposition can be seen from the comparison of speedup for different decompositions. In this study, we experiment domain decomposition for the medium-scale problem (IO10) on the 16-processor computer cluster.

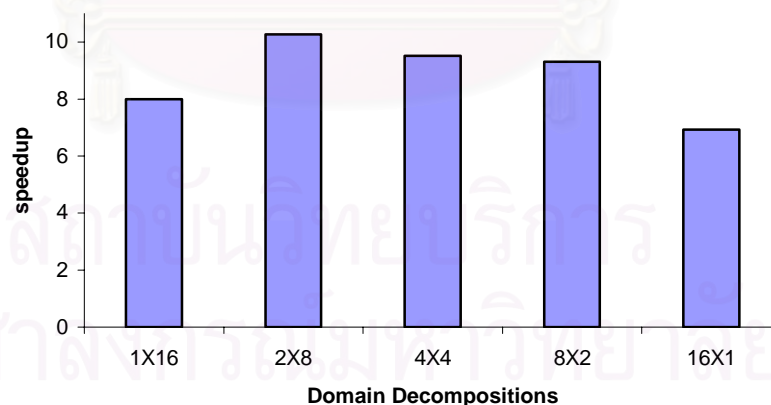


Figure 5.6: Speedup for different decompositions

From Figure 5.8, it can be seen that the 2×8, 4×4 and 8×2 decompositions give better performance (speedup > 8) than the others. This is because the communication in overlapping zone between neighboring sub-domains for these three decompositions

is small. By comparing the number of grid points in overlapping zone (see equation (3.2) in Chapter 3) with computing zone, the ratios of the communicated points to the computed points are found to be 11.88%, 6.21%, 4.39%, 5.49% and 10.07% for the  $1 \times 16$ ,  $2 \times 8$ ,  $4 \times 4$ ,  $8 \times 2$  and  $16 \times 1$  decompositions respectively.

For this particular event, the result indicates that the  $2 \times 8$  decomposition is the appropriate choice for the parallel computation. It should be noted that the communication to computation ratio is not the only factor affecting the parallel speedup and efficiency. In tsunami simulation, the load balancing task in terms of land and water distribution is another important issue for the improvement of speedup and efficiency.

### 5.2.5 Performance and Parallel I/O

In this section, the effect of parallel I/O to the performance of the parallel computation is investigated for the medium-scale problem (IO10). Numerical experiments are performed for 15,000 time steps, and outputs are obtained every 100 time steps. The performance results are shown in Figure 5.7.

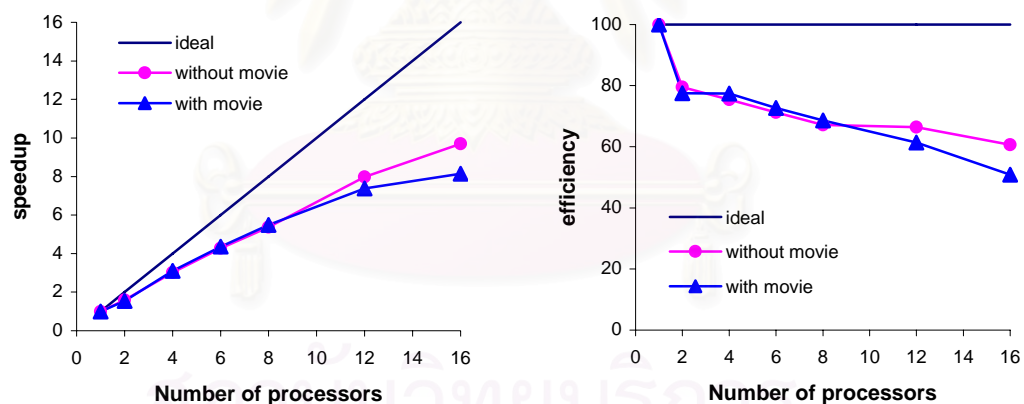


Figure 5.7: Performance for parallel output

Figure 5.7 shows that the performance of the parallel program with and without writing output files (movie) show significant different when the number of processors increases. This is caused by the idle time that each processor has to wait for transferring data to the master processor.

### 5.2.6 Block Sizes and Pipelining Tridiagonal Solver

To investigate the effect of block size, the performance of parallel computation between different block sizes of message in solving tridiagonal matrix by using the pipelining approach is tested on the medium-scale problem (IO10) on the 4-processor computer system. Since solving tridiagonal matrix is performed only on wave computation, the experiments are based only on the wave propagation part.

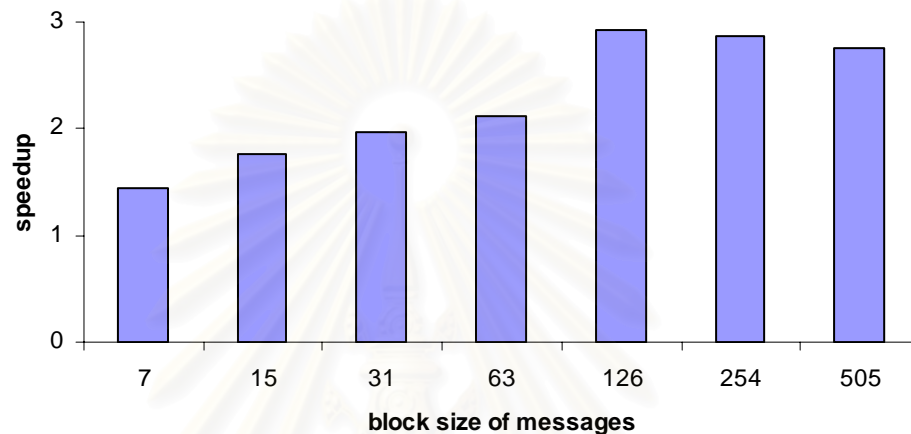


Figure 5.8: Speedup for different block size of messages in pipelining scheme

From figure 5.8, it can be seen that the longer size the messages, the better performance can be expected. This is due to the communication overhead in term of latency and bandwidth. However, the extra long size messages do not give the best performance due to the idle time associated with the pipeline being filled or emptied.

### 5.2.7 Overall Performance of Parallelization on Computer Clusters

In this section, the performance of tsunami parallel computation on different clusters with the medium-scale problem is presented. Based on the characteristic of each cluster, total CPU time may be inconsistent and vary as the number of processors increases depending on the computing characteristics. In order to better estimate the parallel performance from different clusters, new performance indices must be used for comparing cluster performances. Since the speedup of parallel performance can be affected by the bandwidth and interface problems, we define the effective speedup (Kim, 1997) as follows:

$$S_{eff}(n) = \frac{\text{Total CPU time using } n \text{ processors}}{\text{Elapsed time using } n \text{ processors}}$$

This definition is the identical to the parallel speedup if the elapsed time on one processor is equal to the total CPU time on  $n$  processors. Consequently, the CPU efficiency (Kim, 1997) can be defined as follows:

$$E_{CPU}(n) = \frac{\text{Total CPU time using } n \text{ processors}}{\text{Total elapsed time using } n \text{ processors}}$$

Table 5.10: Performance results of the tsunami source and the wave propagation between different clusters

Number of processors	Speedup			Efficiency (%)		
	Xeon	Itanium	Opteron	Xeon	Itanium	Opteron
2	1.90	1.98	1.87	94.93	99.05	93.74
4	3.68	3.91	3.53	91.94	97.71	88.22
6	5.25	5.81	5.16	87.43	96.84	86.07
8	6.84	7.68	6.56	85.55	96.04	82.06

Using these new performance indices, it is found that the developed parallel program can be run on both distributed-shared memory and shared-memory computer architectures with satisfactory results as shown in Table 5.10. Among these clusters, the NECTEC computer system gives best performance results. This is due to the high-speed interconnection network between nodes of cluster.

## CHAPTER VI

### CONCLUSION

In this study, parallel codes for the simulation of tsunami problem are developed and validated. The model is based on FUNWAVE propagation and runup model that has been written in sequential program. This model is widely used in the tsunami modeling community. Two tsunami events, the 1999 Vanuatu tsunami and the 2004 Indian Ocean tsunami, are used as numerical tests on the distributed-shared memory and shared-memory computer systems. This is a further step towards a goal that remains elusive for tsunami simulation: the ability to model this complex problem in a quick manner.

The two earthquake/tsunami events have recently been studied in details. Information on tsunami simulations, such as bathymetric data, surveyed runups and eyewitness reports, and etc., are based on previous studies of Ioualalen *et al.* (2006) and Ioualalen *et al.* (2007). Objectives of the present work are (i) to develop the FORTRAN/MPI programming for parallel computer based on the sequential numerical code, which is difficult due to complexity of the extended Boussinesq equations and the numerical scheme (spatial and temporal variations), (ii) to validate the parallel model by using field observations, and finally (iii) to test the capabilities of the new parallel model for two tsunami events, which have different characteristics.

The November 26, 1999 Vanuatu earthquake/tsunami is a relatively small scale problem with significant impact confined in limited areas as compared with the December 26, 2004 Indian Ocean earthquake/tsunami event. Accuracy of the parallel model is tested satisfactorily with the sequential model, which provides robust simulation and accurately predicts the tsunami impact in the affected areas. Overall performances of the parallel model for both large and small scale tsunami problems are very promising and tend to linear speedup (the ideal response).

Performance of the parallel model depends on several important factors. In this study we are concerned with the domain decomposition in term of load balancing task and minimization of the communication, the parallel tridiagonal solver, the parallel I/O and the characteristic of the cluster. It is found that parallel computation is more favorable to larger problem size. This is mainly due to the ratio of communication to

computation. For large-scale problem, the ratio of computing zone is much greater than the overlapping zone. On the contrary, for small-scale problem, such ratio is smaller and thus requires more communication time. In the domain decomposition technique, we are concerned with the balance tasks and minimization of the communication between processors. Physical characteristic of the domain is very important to load balancing. If the sub-domain contains more land area, it will require lower computation time than the others. Also, the imbalance tasks can occur by the increasing number of operations in some processors due to extra terms in the computation, such as sponge layer boundary condition, wave breaking, moving shoreline and short wave filtering subroutine. Another factor that can affect the communication time is the numbers of communications and grid points in the overlapping areas due to the interchanging of data between the contiguous processors. The optimal decomposition can be achieved based on the control of these factors to get the best performance of parallel computation. For the issue of parallel tridiagonal solver, we are concerned with the size of block messages. It is found that the block size should be sufficiently large to gain better performance. For parallel I/O, major problem is the bottleneck communication during the send/receive processes. To improve the parallel performance when transferring input and output between processors, MPI version MPI2, which provides libraries for parallel I/O and high-speed interconnection network, may be an alternative to parallel programming.

The numerical code has been implemented and tested in three computer systems with different architectures, e.g., Chulalongkorn University, NECTEC in Bangkok, and Géosciences Azur in France (University of Nice, University of Paris VI, CNRS and IRD). Performance results and discussions are presented in Chapter 5. These provide a benchmark for which the performance of the tsunami simulation algorithm on a parallel computer can be estimated. It is found that parallel algorithm gains better performance with large-scale simulation, in particular domain with complex coastal areas for which finer grid-spacing is required. Mesh adaptation has not been considered in this work, primarily because the emphasis here has been on designing the basic parallel algorithm for tsunami problem. The mesh resolution is generally predetermined by the expected wavelength and at present the mesh is designed to resolve the wavelength across the whole spatial domain. To speed up the calculation with better accuracy in numerical solutions, the combination of the adaptative process with the restrictions on the mesh uniformity required by this model

may be used but it could complicate the process of parallelism. A nontrivial extension, but one that would greatly benefit from this parallel code, would be the refinement of tsunami model for the past events. One obvious example is the case study of the 1999 Vanuatu tsunami. The work of Ioualalen *et al.* (2006) based on the sequential code of FUNWAVE was limited to the coastal area within distances away from the tsunami source and unable to use the only available tide gauge record at Prot Vila, Efate Island in South Vanuatu. This parallel code would indeed be useful for the further study on this tsunami event and others as well.



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



## References

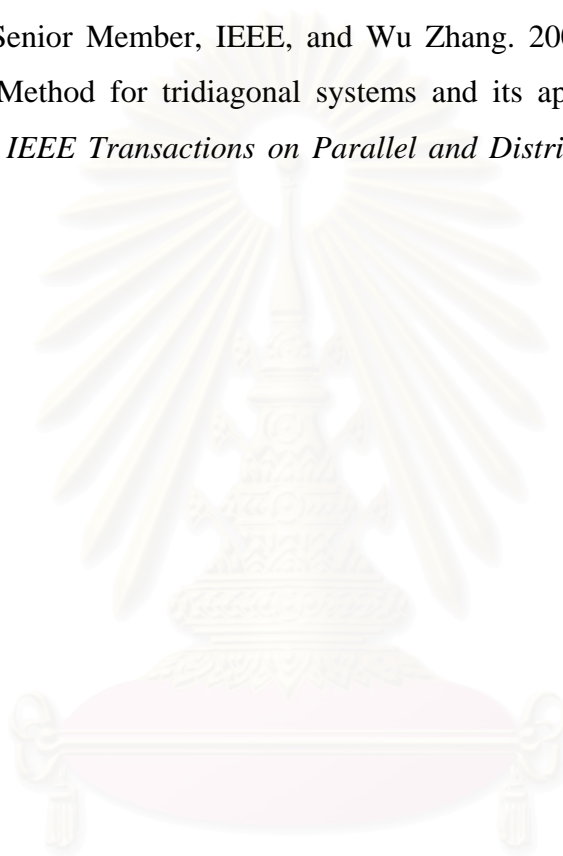
- Benjamin, T.B., and Feir, J.E. 1967. The disintegration of wave trains on deep water. *Journal of Fluid Mechanics*. 27: 417-430.
- Bondevik, S., Svendsen, J.I., Johnsen, G., Mangerud, J., and Kaland, P.E. 1997. The Storegga tsunami along the Norwegian coast, its age and runup. *Boreas*. 26: 29-53.
- Bryant, E. 2001. *The underrated hazard*. Cambridge University Press.
- Chen, Q., Kirby, J.T., Dalrymple, R.A., Kennedy, A.B., and Chawla, A. 2000. Boussinesq modeling of wave transformation, breaking, and run-up. II: 2D. *Journal of Waterway Port Coastal and Ocean Engineering*. 126(1): 48-56.
- Collot, J.-Y., Daniel, J., and Burne, R.V. 1985. Recent tectonics associated with the subduction/collision of the D'Entrecasteaux zone in the central New Hebrides. *Tectonophysics*. 112: 325-356.
- Dawson, A.G., Shi, S., Dawson, S., Takahashi, T., and Shuto, N. 1996. Coastal sedimentation associated with the June 2nd and 3rd, tsunami in Rajegwesi, Java. *Quaternary Science Reviews*. 15: 901-912.
- Dawson, S., and Smith, D.E. 2000. The sedimentology of middle Holocene tsunami facies in northern Sutherland, Scotland. *Marine Geology*. 170: 69-79.
- Day, S.J., Watts, P., Grilli, S.T., and Kirby, J.T. 2005. Mechanical models of the 1975 Kalapana, Hawaii earthquake and tsunami. *Marine Geology*. 215(1-2): 59-92.
- Dongarra, J., Foster, I., Fox, G., Gropp, W., Kennedy, K., Torczon, L., and White, A. 2003. *Sourcebook of Parallel Computing*. Morgan Kaufmann Publishers.
- Garg, R.P., and Sharapov, I. 2001. *Techniques for Optimizing Applications: High Performance Computing*. Sun Microsystems Press Blueprints.
- Gelfenbaum, G., and Jaffe, B. 2003. Erosion and sedimentation from the 17 July, 1998 Papua New Guinea Tsunami. *Pure Applied Geophysics*. 160: 1969-1999.
- Grilli, S.T., Ioualalen, M., Asavanant, J., Shi, F., Kirby, J., and Watts, P. 2007. Source constraints and model simulation of the December 26, 2004 Indian Ocean Tsunami. *Journal of Waterway Port Coastal and Ocean Engineering*. 133(6): 414-428.
- Hendrik, L. 2002. Distributed-memory concepts in the wave model WAVEWATCH III. *Parallel computing*. 28(1): 35-52.

- Ioualalen, M., Asavanant, J., Kaewbanjak, N., Grilli, S.T., Kirby, J.T. and Watts, P. 2007. Modeling the 26th December 2004 Indian Ocean tsunami: Case study of impact in Thailand. *Journal of Geophysical Research*. 112: C07024, doi: 10.1029/2006JC003850.
- Ioualalen, M. 2008. Sensitivity tests on relations between the tsunami signal and seismic rupture characteristics; the 26 December 2004 Indian Ocean event case study. *Environmental Modeling and Software*. doi :10.1016/j.envsoft.2007.07.007, In press.
- Ioualalen, M., and Kharif, C. 1994. On the subharmonic instabilities of steady three-dimensional deep water waves. *Journal of Fluid Mechanics*. 262: 265-291.
- Ioualalen, M., Pelletier, B., Watts, P., and Regnier, M. 2006. Numerical modeling of the 26th November 1999 Vanuatu tsunami. *Journal of Geophysical Research*. 111: C06030, doi:10.1029/2005JC003249.
- Ioualalen, M., Kharif, C., and Roberts, A.J. 1999. Stability regimes of finite depth short-crested water waves. *Journal of Physical Oceanography*. 29(9): 2318-2331.
- Kanamori, H. 1972. Tectonic implications of the 1944 Tonankai and the 1946 Nankaido earthquakes. *Physics of the Earth and Planetary Interiors*. 5: 129-139.
- Kennedy, A.B., Chen, Q., Kirby, J.T., and Dalrymple, R.A. 2000. Boussinesq modeling of wave transformation, breaking, and runup. I:1D. *Journal of Waterway Port Coastal and Ocean Engineering*. 126: 39-47.
- Kim, S.J., and Kim, J.H. 1997. Large-scale structural analysis using domain decomposition method on distributed parallel computing environment. *HPC Asia'97*. Seoul. 573-578.
- Kirby, J.T., Wei, G., Chen, Q., Kennedy, A.B., and Dalrymple, R.A. 1998. Fully Nonlinear Boussinesq wave model Documentation and User's Manual., *Research Report NO.CACR-98-06*.
- Lagabrielle, Y., Pelletier, B., Cabioch, G., Regnier, M., and Calmant, S. 2003. Coseismic and long-term vertical displacement due to back arc shortening, central Vanuatu: Offshore and onshore data following the  $M_w$  7.5 26 November 1999 Ambrym earthquake. *Journal of Geophysical Research*. 108(B11): 2519, doi:10.1029/2002JB002083.

- Lay, T., Kanamori, H., Ammon, C.J., Nettles, M., Ward, S.N., Aster, R., Beck, S.L., Bilek, B.L., Brudzinski, M.R., Butler, R., DeShon, H.R., Ekstrom, G., Satake, K., and Sipkin, S. 2005. The great Sumatra-Andaman earthquake of 26 December 2004. *Science*. 308: 1127-1132.
- Louat, R., and Pelletier, B. 1989. Seismotectonics and present day relative plate motions in the New Hebrides-North Fiji Basin region. *Tectonophysics*. 167: 41-55.
- Mattor, N., Williams, T.J., and Hewett, D.W. 1995. Algorithm for solving tridiagonal matrix problems in parallel. *Parallel Computing*. 21(11): 1769-1782.
- McLean, J.W. 1982a. Instabilities of finite amplitude water waves. *Journal of Fluid Mechanics*. 114: 315-330.
- McLean, J.W. 1982b. Instabilities of finite amplitude gravity waves on water of finite depth. *Journal of Fluid Mechanics*. 114: 331-341.
- Minoura, K., Imamura, F., Takahashi, T., and Shuto, N. 1997. Sequence of sedimentation processes caused by the 1992 Flores tsunami: evidence from Babi Island. *Geology*. 25: 523-526.
- Nishimura, Y., and Miyaji, N. 1995. Tsunami deposits from the 1993 Southwest Hokkaido Earthquake and the 1640 Hokkaido Komagatake Eruption, Northern Japan. *Pure Applied Geophysics*. 144: 719-733.
- Okada, Y. 1985. Surface deformation due to shear and tensile faults in a half-space. *Bulletin of the Seismological Society of America*. 75(4): 1135-1154.
- Pelletier, B., Calmant, S., and Pillet, R. 1998. Current tectonics of the Tonga-New Hebrides region. *Earth and Planetary Science Letters*. 164: 263-276.
- Pelletier, B., Meschede, M., Chabernaud, T., Roperch, R., and Zhao, X. 1994. Tectonics of the Central New Hebrides arc, North Aoba basin. *Proceedings of the Ocean Drilling: Science Results*. 134: 431-444.
- Pelletier, B., Regnier, M., Calmant, S., Pillet, R., Cabioch, G., Lagabrielle, Y., Bore, J.-M., Caminade, J.-P., Lebellegard, P., Christopher, I., and Temakon, S. 2000. Le seisme d'Ambrym-Pentecôte (Vanuatu) du 26 novembre 1999 (M<sub>w</sub> : 7.5) : Données préliminaires sur la séismicité, le tsunami et les déplacements associés. *C. R. Acad. Sci. ser. II*. 331: 21-28.
- Regnier, M., Calmant, S., Pelletier, B., Lagabrielle, Y., and Cabioch, G. 2003. The M<sub>w</sub> 7.5 1999 Ambrym earthquake, Vanuatu: A back arc intraplate thrust event. *Tectonics*. 22(4): 1034, doi:10.1929/2002TC001422.

- Sato, H., Shimamoto, T., Tsutsumi, A., and Kawamoto, E. 1995. Onshore tsunami deposits caused by the 1993 Southwest Hokkaido and 1983 Japan Sea Earthquakes. *Pure Applied Geophysics*. 144: 693-717.
- Shi, S., Dawson, A.G., and Smith, E.D. 1995. Coastal sedimentation associated with the December 12th, 1992 tsunami in Flores, Indonesia. *Pure Applied Geophysics*. 144: 525-536.
- Sitanggang, K. and Lynett, P. 2005. Parallel Computation of a Highly Nonlinear Boussinesq Equation Model through Domain Decomposition. *International Journal for Numerical Methods in Fluids*. 49: 57-74.
- Socquet, A., Vigny, C., Simons, W., Chamot-Rooke, N., Rangin, C., and Ambrosius, B. 2006. GPS determination of the relative motion between India and Sunda, and its accommodation in Myanmar. *Journal of Geophysical Research*. 111: B05406, doi:10.1029/2005JB003877.
- Stein, S., and Okal, E. 2005. Speed and size of the Sumatra earthquake. *Nature*. 434: 581-582.
- Taylor, F.W., Bevis, M., Schutz, B., Kuang, D., Recy, J., Calmant, S., Charley, D., Regnier, M., Perin, B., Jackson, M., and Reichenfeld, C. 1995. Geodetic measurements of convergence at New Hebrides island arc indicate arc fragmentation caused by an impinging aseismic ridge. *Geology*. 23: 1011-1014.
- Walder, J.S., and Watts, P. 2003. Evaluating tsunami hazards from debris flows. *Submarine Mass Movements and Their Consequences*, J. Locat and J. Mienert (Eds.), Kluwer Academic Publishers, Dordrecht: 155-162.
- Wang, H.H. 1981. A parallel method for Tridiagonal Equations. *ACM Transactions on Mathematical Software*. 7(2): 170-183.
- Watts, P., Grilli, S.T., Kirby, J.T., Fryer, G.J., and Tappin, D. 2003. Landslide tsunami case studies using a Boussinesq model and a fully nonlinear tsunami generation model. *Natural Hazards and Earth System Sciences*. 3(5): 391-402.
- Waythomas, C.F., and Watts, P. 2003. Numerical simulation of tsunami generation by pyroclastic flow at Aniakchak Volcano, Alaska. *Geophysical Research Letters*. 30(14): 1751-1755.
- Wei, G. and Kirby, J.T. 1995. Time-dependent numerical code for extended Boussinesq equations. *Journal of Waterway Port Coastal and Ocean Engineering*. 121(5): 251-261.

- Wei, G., Kirby, J.T., Gilli, S.T., and Subramanya, R. 1995. A fully nonlinear Boussinesq model for surface waves, Part 1: Highly nonlinear unsteady waves. *Journal of Fluid Mechanics*. 294(5): 71-92.
- Wright, C., and Mella, A. 1963. Modifications to the soil pattern of South-Central Chile resulting from seismic and associated phenomena during the period May to August 1960. *Bulletin of the Seismological Society of America*. 53: 1367-1402.
- Xian-He Sun, Senior Member, IEEE, and Wu Zhang. 2004. A parallel Two-Level Hybrid Method for tridiagonal systems and its application to Fast Poisson Solvers. *IEEE Transactions on Parallel and Distributed Systems*. 15(2): 97-106.



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



## **Appendices**

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## Appendix A

### Boussinesq Equations, Bottom Friction, Wave Breaking, Moving Shorelines, and Subgrid Turbulence

In this study, we consider a three-dimensional wave field with water surface elevation  $\eta(x, y, t)$  at arbitrary time  $t$  propagating over a variable water depth  $h(x, y)$ . A cartesian coordinate system  $(x, y, z)$  is adopted, with  $z$  measured upwards from the still-water level. The fluid is assumed to be inviscid and incompressible, and the flow is assumed to be irrotational. Following Wei *et al.* (1995), the fully nonlinear Boussinesq equations are given by:

$$\begin{aligned} \eta_t + \nabla \cdot \left\{ (h + \eta) \left[ \mathbf{u}_\alpha + \left( z_\alpha + \frac{1}{2}(h - \eta) \right) \nabla (\nabla \cdot (h\mathbf{u}_\alpha)) \right. \right. \\ \left. \left. + \left( \frac{1}{2} z_\alpha^2 - \frac{1}{6} (h^2 - h\eta + \eta^2) \right) \nabla (\nabla \cdot \mathbf{u}_\alpha) \right] \right\} = 0 \end{aligned} \quad (\text{A.1})$$

$$\begin{aligned} \mathbf{u}_{\alpha t} + (\mathbf{u}_\alpha \cdot \nabla) \mathbf{u}_\alpha + g \nabla \eta + z_\alpha \left\{ \frac{1}{2} z_\alpha \nabla (\nabla \cdot \mathbf{u}_{\alpha t}) + \nabla (\nabla \cdot (h\mathbf{u}_{\alpha t})) \right\} \\ + \nabla \left\{ \frac{1}{2} (z_\alpha^2 - \eta^2) (\mathbf{u}_\alpha \cdot \nabla) (\nabla \cdot \mathbf{u}_\alpha) + \frac{1}{2} [\nabla \cdot (h\mathbf{u}_\alpha) + \eta \nabla \cdot \mathbf{u}_\alpha]^2 \right\} \\ + \nabla \left\{ (z_\alpha - \eta) (\mathbf{u}_\alpha \cdot \nabla) (\nabla \cdot (h\mathbf{u}_\alpha)) - \eta \left[ \frac{1}{2} \eta \nabla \cdot \mathbf{u}_{\alpha t} + \nabla \cdot (h\mathbf{u}_{\alpha t}) \right] \right\} = 0 \end{aligned} \quad (\text{A.2})$$

where  $\eta$  is the surface elevation,  $h$  is the still water depth,  $\mathbf{u}_\alpha$  is the horizontal velocity vector at the water depth  $z = z_\alpha = -0.531h$ ,  $\nabla = (\partial/\partial x, \partial/\partial y)$  is the horizontal gradient operator,  $g$  is the gravitational acceleration, and subscript  $t$  denotes the partial derivative with respect to time.

In order to develop the model for practical application, the physical effects of frictional damping and wave breaking, boundary absorption, and moving shoreline are incorporated into the model scheme. Equations (A.1) and (A.2) are modified by including these extensions as

$$\eta_t = E(\eta, u, v) + \gamma E_2(\eta, u, v) \quad (\text{A.3})$$

$$\begin{aligned} [U(u)]_t &= F(\eta, u, v) + [F_1(v)]_t + \gamma [F_2(\eta, u, v) + F^t(\eta, u_t, v_t)] \\ &\quad + F_b + F_{br} + F_{bs} + F_{sp} \end{aligned} \quad (\text{A.4})$$

$$\begin{aligned} [V(v)]_t &= G(\eta, u, v) + [G_1(u)]_t + \gamma [G_2(\eta, u, v) + G^t(\eta, u_t, v_t)] \\ &\quad + G_b + G_{br} + G_{bs} + G_{sp} \end{aligned} \quad (\text{A.5})$$

Here  $u$  and  $v$  are the horizontal velocities in horizontal directions  $x$  and  $y$  at depth  $z = z_\alpha = -0.531h$ , i.e.,  $(u, v) = u_\alpha$  and  $\gamma$  is a control parameter allowing us to choose between fully nonlinear ( $\gamma = 1$ ) or weakly nonlinear ( $\gamma = 0$ ) Boussinesq cases. The quantities  $U, V, E, F, G, F_1, G_1, F^t, G^t, E_2, F_2$  and  $G_2$  are functions of  $\eta, u, v, u_t$  or  $v_t$  which are defined as

$$U = u + h [b_1 h u_{xx} + b_2 (hu)_{xx}] \quad (\text{A.6})$$

$$V = v + h [b_1 h v_{yy} + b_2 (hv)_{yy}] \quad (\text{A.7})$$

$$\begin{aligned} E &= -\frac{1}{\kappa} [(\Lambda u)_x + (\Lambda v)_y] \\ &\quad - \left\{ a_1 h^3 (u_{xx} + v_{xy}) + a_2 h^2 [(hu)_{xx} + (hv)_{xy}] \right\}_x \\ &\quad - \left\{ a_1 h^3 (u_{xy} + v_{yy}) + a_2 h^2 [(hu)_{xy} + (hv)_{yy}] \right\}_y \end{aligned} \quad (\text{A.8})$$

$$F = -g\eta_x - (uu_x + vv_y) \quad (\text{A.9})$$

$$G = -g\eta_y - (uv_x + vv_y) \quad (\text{A.10})$$

$$F_1 = -h [b_1 h v_{xy} + b_2 (hv)_{xy}] \quad (\text{A.11})$$

$$G_1 = -h [b_1 h u_{xy} + b_2 (hu)_{xy}] \quad (\text{A.12})$$

$$F^t = \left\{ \frac{1}{2} \eta^2 [(u_t)_x + (v_t)_y] + \eta [ [h(u_t)]_x + [h(v_t)]_y ] \right\}_x \quad (\text{A.13})$$

$$G^t = \left\{ \frac{1}{2} \eta^2 [(u_t)_x + (v_t)_y] + \eta [ [h(u_t)]_x + [h(v_t)]_y ] \right\}_y \quad (\text{A.14})$$



$$\begin{aligned}
E_2 = & -\left\{ \left[ a_1 h^2 \eta + \frac{1}{6} \eta (h^2 - \eta^2) \right] (u_{xx} + v_{yy}) \right\}_x \\
& - \left\{ \left[ a_2 h \eta - \frac{1}{2} \eta (h + \eta) \right] [(hu)_{xx} + (hv)_{yy}] \right\}_x \\
& - \left\{ \left[ a_1 h^2 \eta + \frac{1}{6} \eta (h^2 - \eta^2) \right] (u_{xy} + v_{yx}) \right\}_y \\
& - \left\{ \left[ a_2 h \eta - \frac{1}{2} \eta (h + \eta) \right] [(hu)_{xy} + (hv)_{yx}] \right\}_y
\end{aligned} \tag{A.15}$$

$$\begin{aligned}
F_2 = & -\left\{ \frac{1}{2} (z_\alpha^2 - \eta^2) \left[ u (u_x + v_y)_x + v (u_x + v_y)_y \right] \right\}_x \\
& - \left\{ (z_\alpha - \eta) \left[ u [(hu)_x + (hv)_y]_x + v [(hu)_x + (hv)_y]_y \right] \right\}_x \\
& - \frac{1}{2} \left\{ [(hu)_x + (hv)_y + \eta (u_x + v_y)]^2 \right\}_x
\end{aligned} \tag{A.16}$$

$$\begin{aligned}
G_2 = & -\left\{ \frac{1}{2} (z_\alpha^2 - \eta^2) \left[ u (u_x + v_y)_x + v (u_x + v_y)_y \right] \right\}_y \\
& - \left\{ (z_\alpha - \eta) \left[ u [(hu)_x + (hv)_y]_x + v [(hu)_x + (hv)_y]_y \right] \right\}_y \\
& - \frac{1}{2} \left\{ [(hu)_x + (hv)_y + \eta (u_x + v_y)]^2 \right\}_y
\end{aligned} \tag{A.17}$$

where  $a_1, a_2, b_1, b_2$  are constants which are related to the dimensionless referenced water depth  $\beta = z_\alpha/h = -0.531$  by

$$a_1 = \frac{1}{2} \beta^2 - \frac{1}{6}, \quad a_2 = \beta + \frac{1}{2}, \quad b_1 = \frac{1}{2} \beta^2, \quad b_2 = \beta \tag{A.18}$$

The factors  $\Lambda$  and  $\kappa$  in (A.8) are provided for a treatment of moving shoreline.

In this study, the Boussinesq model is modified with extensions to cover bottom friction, wave breaking, moving shorelines, and subgrid turbulence effects developed by Chen *et al.* (2000) and Kennedy *et al.* (2000). The bottom friction is given by

$$(F_b, G_b) = \frac{K}{h + \eta} \mathbf{u}_\alpha |\mathbf{u}_\alpha| \tag{A.19}$$

where  $K = 1 \times 10^{-5}$  is the friction coefficient.

Wave breaking in shallow water based on Kennedy *et al.* (2000) is defined by

$$F_{br} = \frac{1}{h+\eta} \left[ \left( \nu \left( (h+\eta) u_{\alpha} \right)_x \right)_x + \frac{1}{2} \left( \nu \left( \left( (h+\eta) u_{\alpha} \right)_y + \left( (h+\eta) v_{\alpha} \right)_x \right) \right)_y \right] \quad (\text{A.20})$$

$$G_{br} = \frac{1}{h+\eta} \left[ \frac{1}{2} \left( \nu \left( \left( (h+\eta) v_{\alpha} \right)_x + \left( (h+\eta) u_{\alpha} \right)_y \right) \right)_x + \left( \nu \left( (h+\eta) v_{\alpha} \right)_y \right)_y \right] \quad (\text{A.21})$$

where subscripts  $x$  and  $y$  denote spatial derivatives, and  $\nu$  is the eddy viscosity localized on the front face of the breaking wave, which is define as

$$\nu = B\delta^2 \left| (h+\eta) \nabla \cdot \mathbf{M} \right| \quad (\text{A.22})$$

where  $\delta$  is a mixing length coefficient with an empirical of  $\delta = 1.2$ . The quantity  $B$  that controls the occurrence of energy dissipation is given by

$$B = \begin{cases} 1, & \eta_t \geq 2\eta_t^* \\ \frac{\eta_t}{\eta_t^*} - 1, & \eta_t^* < \eta_t < 2\eta_t^* \\ 0, & \eta_t \leq \eta_t^* \end{cases} \quad (\text{A.23})$$

and the onset and cessation of wave breaking using the parameter,  $\eta_t^*$ , which is defined as

$$\eta_t^* = \begin{cases} \eta_t^{(F)}, & t \geq T^* \\ \eta_t^{(I)} + \frac{t-t_0}{T^*} (\eta_t^{(F)} - \eta_t^{(I)}), & 0 \leq t-t_0 < T^* \end{cases} \quad (\text{A.24})$$

where  $T^*$  is the transition time,  $t_0$  is the time when the wave breaking occurs, and  $t-t_0$  is the age of the breaking event. The value  $\eta_t^{(I)}$  varies between  $0.35\sqrt{gh}$  and  $0.65\sqrt{gh}$ , while the value of  $\eta_t^{(F)}$  and  $T^*$  are  $0.15\sqrt{gh}$  and  $5\sqrt{h/g}$ , respectively. The construction and verification of the breaking model was detailed by Kennedy *et al.* (2000).

The factor  $\Lambda$  and  $\kappa$  in (A.8) were introduced by Kennedy *et al.* (2000) and Chen *et al.* (2000) to implement a porous (i.e., absorbing) beach method, used to keep the subaerial portion of the model grid computationally active and to simplify the calculation of runup on dry shorelines. These are given by

$$\kappa = \begin{cases} \delta + (1-\delta) e^{\lambda \left( \frac{\eta-z}{h_0} \right)}, & \eta \leq z^* \\ 1, & \eta > z^* \end{cases} \quad (\text{A.25})$$

and

$$\Lambda = \begin{cases} \delta(\eta + h_0) + \frac{(1-\delta)h_0}{\lambda} \left( e^{\lambda \frac{(\eta-z^*)}{h_0}} - e^{-\lambda \frac{(h_0+z^*)}{h_0}} \right), & \eta \leq z^* \\ (\eta - z^*) + \delta(z^* + h_0) + \frac{(1-\delta)h_0}{\lambda} \left( 1 - e^{-\lambda \frac{(h_0+z^*)}{h_0}} \right), & \eta > z^* \end{cases} \quad (\text{A.26})$$

where  $h_0$  represents the porous beach depth, which must be deeper than the depth of maximum wave rundown during a calculation. The choice of  $z^*$  is discussed by Kennedy *et al.* (2000) and given by

$$z^* = \frac{z^s}{1-\delta} + h_0 \left( \frac{\delta}{1-\delta} + \frac{1}{\lambda} \right) \quad (\text{A.27})$$

where  $z^s$  is the elevation of the solid seabed. Here  $\delta = 0.08$  and  $\lambda = 25$ , based on studies of a number of tsunami runup events followed by Watts *et al.* (2003) and Day *et al.* (2005).

The Smagorinsky type subgrid model (Smagorinsky 1963) to account for the effect of the resultant eddy viscosity on the underlying flow.

$$F_{bs} = \frac{1}{h+\eta} \left[ \left( v_s \left( (h+\eta) u_\alpha \right)_x \right)_x + \frac{1}{2} \left( v_s \left( \left( (h+\eta) u_\alpha \right)_y + \left( (h+\eta) v_\alpha \right)_x \right) \right)_y \right] \quad (\text{A.28})$$

$$G_{br} = \frac{1}{h+\eta} \left[ \frac{1}{2} \left( v_s \left( \left( (h+\eta) v_\alpha \right)_x + \left( (h+\eta) u_\alpha \right)_y \right) \right)_x + \left( v_s \left( (h+\eta) v_\alpha \right)_y \right)_y \right] \quad (\text{A.29})$$

where  $v_s$  is the eddy viscosity due to the subgrid turbulence.

$$v_s = c_m \Delta x \Delta y \left[ (U_x)^2 + (V_y)^2 + \frac{1}{2} (U_y + V_x)^2 \right]^{\frac{1}{2}} \quad (\text{A.30})$$

in which  $U$  and  $V$  are the velocity components of the time averaged underlying current field,  $\Delta x$  and  $\Delta y$  are the grid spacing in the  $x$  and  $y$  directions, respectively, and  $c_m$  is the mixing coefficient with the default value of 0.2. In the course of simulation, the underlying current field is obtained by averaging the instantaneous velocity over two peak wave periods and  $v_\alpha$  is updated accordingly.

## Appendix B

### Numerical Scheme

#### B.1 Finite difference Scheme

Following Wei *et al.* (1995) and Kirby *et al.* (1998), a composite fourth-order Adams Bashforth-Moulton scheme (utilizing a third-order Adams-Bashfort predictor step and forth-order Adams-Moulton corrector step) is used to step the model forward in time. Terms involving first-order spatial derivatives are differenced to  $O(\Delta x^4)$  accuracy by utilizing a five-point formula. All errors involved in solving the underlying nonlinear shallow water equations are thus reduced to 4th order in grid spacing and time step size. Spatial and temporal differencing of the higher-order dispersion terms is done to the second-order accuracy, which again reduces the truncation errors to a size smaller than those terms themselves. No further back-substitution of apparent truncation error terms is performed.

#### Time-differencing

The arrangement of cross-differentiated and nonlinear time derivative terms on the right hand side of equation (A.6)-(A.7) marks the resulting set of left-hand sides purely tridiagonal. The governing equations are finite-different on a centered grid in  $x = i\Delta x, y = j\Delta y, t = n\Delta t$ . Level  $n$  refers to information at the present, known time level. The predictor step is the third-order explicit Adams-Bashforth scheme, given by

$$\eta_{i,j}^{n+1} = \eta_{i,j}^n + \frac{\Delta t}{12} \left[ 23(E')_{i,j}^n - 16(E')_{i,j}^{n-1} + 5(E')_{i,j}^{n-2} \right] \quad (\text{B.1})$$

$$U_{i,j}^{n+1} = U_{i,j}^n + \frac{\Delta t}{12} \left[ 23(F')_{i,j}^n - 16(F')_{i,j}^{n-1} + 5(F')_{i,j}^{n-2} \right] + \frac{\Delta t}{12} \left[ 23((F_1)_t)_{i,j}^n - 16((F_1)_t)_{i,j}^{n-1} + 5((F_1)_t)_{i,j}^{n-2} \right] \quad (\text{B.2})$$

$$V_{i,j}^{n+1} = V_{i,j}^n + \frac{\Delta t}{12} \left[ 23(G')_{i,j}^n - 16(G')_{i,j}^{n-1} + 5(G')_{i,j}^{n-2} \right] + \frac{\Delta t}{12} \left[ 23((G_1)_t)_{i,j}^n - 16((G_1)_t)_{i,j}^{n-1} + 5((G_1)_t)_{i,j}^{n-2} \right] \quad (\text{B.3})$$

where

$$E' = E + \gamma E_2 \quad (\text{B.4})$$

$$F' = F + \gamma(F_2 + F^t) + F_{br} + F_b + F_{sp} \quad (\text{B.5})$$

$$G' = G + \gamma(G_2 + G^t) + G_{br} + G_b + G_{sp} \quad (\text{B.6})$$

All information on the right hand sides of (B.1)-(B.3) is known from previous calculations. The value of  $\eta_{i,j}^{n+1}$  are thus straightforward to obtain. The elevation of horizontal velocities at the new time level, however, requires simultaneous solution of tridiagonal matrix systems which are linear in the unknowns at level  $n+1$ . Specifically, for a given  $j$ ,  $u_{i,j}^{n+1}$  ( $i=1,2,\dots,M$ ) are obtained through tridiagonal matrix solution. Similarly,  $v_{i,j}^{n+1}$  ( $j=1,2,\dots,M$ ) are solved by a system of tridiagonal matrix equation for given  $i$ . The matrices involved are constant in time and may be pre-factored, inverted and stored for use at each time step.

After the predicted value of  $\{\eta, u, v\}_{i,j}^{n+1}$  are evaluated, we obtain the corresponding quantities of  $\{E', F', G'\}_{i,j}$  at time levels  $(n+1), (n), (n-1), (n-2)$ , and apply the fourth-order Adams-Moulton corrector method

$$\eta_{i,j}^{n+1} = \eta_{i,j}^n + \frac{\Delta t}{24} \left[ 9(E')_{i,j}^{n+1} + 19(E')_{i,j}^n - 5(E')_{i,j}^{n-1} + (E')_{i,j}^{n-2} \right] \quad (\text{B.7})$$

$$U_{i,j}^{n+1} = U_{i,j}^n + \frac{\Delta t}{24} \left[ 9(F')_{i,j}^{n+1} + 19(F')_{i,j}^n - 5(F')_{i,j}^{n-1} + (F')_{i,j}^{n-2} \right] \\ + \frac{\Delta t}{24} \left[ 9((F_1)_t)_{i,j}^{n+1} + 19((F_1)_t)_{i,j}^n - 5((F_1)_t)_{i,j}^{n-1} + ((F_1)_t)_{i,j}^{n-2} \right] \quad (\text{B.8})$$

$$V_{i,j}^{n+1} = V_{i,j}^n + \frac{\Delta t}{24} \left[ 9(G')_{i,j}^{n+1} + 19(G')_{i,j}^n - 5(G')_{i,j}^{n-1} + (G')_{i,j}^{n-2} \right] \\ + \frac{\Delta t}{24} \left[ 9((G_1)_t)_{i,j}^{n+1} + 19((G_1)_t)_{i,j}^n - 5((G_1)_t)_{i,j}^{n-1} + ((G_1)_t)_{i,j}^{n-2} \right] \quad (\text{B.9})$$

From the definition, we see that calculation of  $F^t$  and  $G^t$  at certain time level requires the corresponding values of  $u_t$  and  $v_t$ . Also, the terms  $(F_1)_t$  and  $(G_1)_t$  involves time derivatives. Defining quantity  $w$  as

$$w = \{u, v, F_1, G_1\} \quad (\text{B.10})$$

Then its time derivatives for predictor stage are

$$(w_t)_{i,j}^n = \frac{1}{2\Delta t} \left[ 3w_{i,j}^n - 4w_{i,j}^{n-1} + w_{i,j}^{n-2} \right] + O(\Delta t^2) \quad (\text{B.11})$$

$$(w_t)_{i,j}^{n-1} = \frac{1}{2\Delta t} \left[ w_{i,j}^n - w_{i,j}^{n-2} \right] + O(\Delta t^2) \quad (\text{B.12})$$

$$(w_t)_{i,j}^{n-2} = \frac{1}{2\Delta t} \left[ 3w_{i,j}^{n-2} - 4w_{i,j}^{n-1} + w_{i,j}^n \right] + O(\Delta t^2) \quad (\text{B.13})$$

For the corrector stage, we evaluate  $w_t$  according to

$$(w_t)_{i,j}^{n+1} = \frac{1}{6\Delta t} [11w_{i,j}^{n+1} - 18w_{i,j}^n + 9w_{i,j}^{n-1} - 2w_{i,j}^{n-2}] + O(\Delta t^3) \quad (\text{B.14})$$

$$(w_t)_{i,j}^n = \frac{1}{6\Delta t} [2w_{i,j}^{n+1} + 3w_{i,j}^n - 6w_{i,j}^{n-1} + w_{i,j}^{n-2}] + O(\Delta t^3) \quad (\text{B.15})$$

$$(w_t)_{i,j}^{n-1} = -\frac{1}{6\Delta t} [2w_{i,j}^{n-2} + 3w_{i,j}^{n-1} - 6w_{i,j}^n + w_{i,j}^{n+1}] + O(\Delta t^3) \quad (\text{B.16})$$

$$(w_t)_{i,j}^{n-2} = -\frac{1}{6\Delta t} [11w_{i,j}^{n-2} - 18w_{i,j}^{n-1} + 9w_{i,j}^n - 2w_{i,j}^{n+1}] + O(\Delta t^3) \quad (\text{B.17})$$

By substituting  $(F_1)_t$  and  $(G_1)_t$  into the equation (B.2), (B.3), (B.8), and (B.9), the last terms in these equations reduce to

$$\frac{\Delta t}{12} [23(w_t)_{i,j}^n - 16(w_t)_{i,j}^{n-1} + 5(w_t)_{i,j}^{n-2}] = 2w_{i,j}^n - 3w_{i,j}^{n-1} + w_{i,j}^{n-2} \quad (\text{B.18})$$

$$\frac{\Delta t}{24} [9(w_t)_{i,j}^{n+1} + 19(w_t)_{i,j}^n - 5(w_t)_{i,j}^{n-1} + (w_t)_{i,j}^{n-2}] = w_{i,j}^{n+1} - w_{i,j}^n \quad (\text{B.19})$$

where  $w = \{F_1, G_1\}$ . The corrector step is iterated until the error between two successive results a required limit. The error is computed for each of the three dependent variables  $\eta, u$  and  $v$  and is defined as

$$\Delta f = \frac{\sum_{i=1, j=1}^{i=M, j=N} |f_{i,j}^{n+1} - f_{i,j}^*|}{\sum_{i=1, j=1}^{i=M, j=N} |f_{i,j}^{n+1}|} \quad (\text{B.20})$$

Where  $f = \{\eta, u, v\}$ ,  $f^{n+1}$  and  $f^*$  denote the current and previous results, respectively. The corrector step is iterated if any of  $\Delta f$ 's exceeds  $10^{-4}$  or  $10^{-3}$ . For ‘‘cold start’’ running of the model, the denominator in (B.20) is zero initially, which will result in infinite value of  $\Delta f$ . To eliminate this problem, we first compute the corresponding denominator. If value of the denominator is smaller than a small value (say,  $10^{-3}$ ), then only numerator from (B.20) is used for iteration errors.

To increase the convergence rate, an over-relaxation technique is applied to the iteration stage. Writing the previous and current iterated values as  $f_{i,j}^*$  and  $f_{i,j}$  then the adjusted valued  $f_{i,j}^r$  for over-relaxation is given by

$$f_{i,j}^r = (1-r)f_{i,j}^* + Rf_{i,j} \quad (\text{B.21})$$

where  $R$  is a coefficient in the range of  $(0,1)$ . In all computations, it is found that  $R = 0.2$  gives quite satisfactory results.

## Spatial differencing

For first order spatial derivatives, the following five-point difference scheme are used

$$(w_x)_{1,j} = \frac{1}{12\Delta x} (-25w_{1,j} + 48w_{2,j} - 36w_{3,j} + 16w_{4,j} - 3w_{5,j}) \quad (\text{B.22})$$

$$(w_x)_{2,j} = \frac{1}{12\Delta x} (-3w_{1,j} - 10w_{2,j} + 18w_{3,j} - 6w_{4,j} + w_{5,j}) \quad (\text{B.23})$$

$$(w_x)_{i,j} = \frac{1}{12\Delta x} (8(w_{i+1,j} - w_{i-1,j}) - (w_{i+2,j} - w_{i-2,j})) \quad (\text{B.24})$$

$$(i = 3, 4, \dots, M - 2)$$

$$(w_x)_{M-1,j} = \frac{1}{12\Delta x} (3w_{M,j} + 10w_{M-1,j} - 18w_{M-2,j} + 6w_{M-3,j} - w_{M-4,j}) \quad (\text{B.25})$$

$$(w_x)_{M,j} = -\frac{1}{12\Delta x} (25w_{M,j} - 48w_{M-1,j} + 36w_{M-2,j} - 16w_{M-3,j} + 3w_{M-4,j}) \quad (\text{B.26})$$

where  $w = \{\eta, u, v\}$ ,  $M_k = M - k$  ( $k = 1, 2, 3, 4$ ), and  $M$  is the total number of grid point in  $x$  direction.

For second order derivatives, we use three-point difference scheme

$$(w_{xx})_{i,j} = \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta x)^2} \quad (i = 1, 2, \dots, M - 1) \quad (\text{B.27})$$

Similar expressions can be obtained for derivatives with respect to  $y$ . For mixed derivatives, we use

$$(w_{xy})_{i,j} = \frac{w_{i+1,j+1} + w_{i-1,j-1} - w_{i-1,j+1} - w_{i+1,j-1}}{4\Delta x\Delta y} \quad (\text{B.28})$$

$$(i = 2, 3, \dots, M - 1; j = 2, 3, \dots, N - 1)$$

## B.2 Boundary Conditions

### Reflective Boundaries

For a perfect reflecting vertical wall, the horizontal velocity normal to the wall is always zero, i.e.

$$\mathbf{u} \cdot \mathbf{n} = 0; \quad (x, y) \in \partial\Omega \quad (\text{B.29})$$

The corresponding values of surface elevation and tangential velocity, the normal derivative as zero, i.e.,

$$\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0; \quad (x, y) \in \partial\Omega \quad (\text{B.30})$$

$$\frac{\partial \eta}{\partial \mathbf{n}} = 0; \quad (x, y) \in \partial \Omega \quad (\text{B.31})$$

where  $\Omega$  = the fluid domain,  $\partial \Omega$  = the boundary and  $(x, y)$  = a position in the domain.

### Absorbing Boundaries

There are several types of absorbing boundary condition which allows waves to propagate out of domain with minimum reflection. A sponge layer boundary condition is used here since it is able to damp wave energy for a wide range of frequencies and directions. Although extra grid points are needed, it is justified to apply sponge layer due to the decreasing cost of computer storage and the stability of the numerical model.

To absorb wave energy, artificial damping terms  $F_{sp}$  and  $G_{sp}$  are added to the right hand side of the momentum equation (A.4) and (A.5), respectively. The damping terms are defined as

$$F_{sp} = -w_1(x, y)u + w_2(x, y)(u_{xx} + u_{yy}) + w_3(x, y)\sqrt{\frac{g}{h}}\eta \quad (\text{B.32})$$

$$G_{sp} = -w_1(x, y)v + w_2(x, y)(v_{xx} + v_{yy}) + w_3(x, y)\sqrt{\frac{g}{h}}\eta \quad (\text{B.33})$$

Where  $w_1, w_2$  and  $w_3$  are function for three different kinds of damping mechanism, which were referred to as Newtonian cooling, viscous damping, and sponge filter, respectively. Assuming that there is only one sponge layer on the right end of domain,  $w_i, i = 1, 2, 3$  defined as

$$w_i(x, y) = \begin{cases} c_i \omega f(x), & x_s < x < x_l \\ 0, & \text{elsewhere} \end{cases} \quad (\text{B.34})$$

where  $i = 1, 2, 3$ ,  $c_i$  are constant coefficients corresponding to the three damping functions,  $\omega$  is frequency of wave to be damped,  $x_s$  is starting coordinate of damping layer (the computing domain is from  $x = 0$  to  $x = x_l$ ) and  $f(x)$  is a smooth monotonically increasing function varying from 0 to 1 when  $x$  varies from  $x_s$  to  $x_l$ .

Function  $f(x)$  is defined as



$$f(x) = \frac{e^{\left[\frac{(x-x_s)}{(x_l-x_s)}\right]^2} - 1}{e - 1} \quad x_s < x < x_l \quad (\text{B.35})$$

The width of the damping layer (i.e.  $x_l - x_s$ ) is usually taken to be two or three wave lengths.



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## VITAE

Miss Nuttita Pophet was born in July 21, 1980, in Suratthani, Thailand. She received a bachelor degree in Secondary Education: General Science and Mathematics, Department of Secondary Education, Faculty of Education, Chulalongkorn University, Bangkok, Thailand in 2001. She received a master degree in Mathematics Education, Department of Secondary Education, Faculty of Graduate School, Chulalongkorn University, Bangkok, Thailand in 2003.



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย