論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Development of Rapid and High-resolution Tsunami Simulator with Unstructured Mesh System based on Finite Volume Method
著者(和文)	曽新翔
Author(English)	Xin Xiang Zeng
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第9514号, 授与年月日:2014年3月26日, 学位の種別:課程博士, 審査員:中村 恭志,石川 忠晴,山中 浩明,木内 豪,浅輪 貴史
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第9514号, Conferred date:2014/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Development of Rapid and High-resolution Tsunami Simulator with Unstructured Mesh System based on Finite Volume Method

Doctoral Dissertation

ZENG Xinxiang

January, 2014

Department of Environmental Science and Technology
Interdisciplinary Graduate School of Science and Engineering
Tokyo Institute of Technology

Abstract

A rapid and high-resolution tsunami simulator based on finite volume method with unstructured mesh system is developed in this research. Three kinds of idealized numerical experiment are conducted with several meshes of different spatial size to verify the effectiveness of the developed tsunami simulator, as well as to investigate necessary conditions for high-resolution tsunami simulation. Quantitative error analysis is conducted and suggesting that not only the numerical scheme, but also the mesh size as well as propagate distance will affect the resolution of tsunami simulation. The required mesh number "N_{req}" per wavelength are summarized for deep-ocean tsunami movement and run-up tsunami respectively. In addition, the simulation of the Tsunami in 2011 Tōhoku Earthquake is also conducted, and the summarized required mesh number "N_{req}" is applied in this simulation and is proved to be effective in achieving high-resolution results.

A new approach of adaptive mesh refinement (AMR) is proposed based on the unstructured triangular mesh system. Two kinds of idealized numerical experiment are conducted with several meshes of different spatial size to verify the abilities of the proposed new AMR method in solving wave propagate on deep-ocean and slopping beach respectively. The results indicate that the proposed AMR method works well in capturing wave propagation and the computational load (CPU time) is decreased to 1/10 at best. In addition, the proposed AMR strategy is also applied to the simulation of tsunami in 2011 Tōhoku earthquake, in which the computing wave height is improved indeed. However, in the real tsunami application, the computational load is just reduced to 1/3 at best. In addition, the effect of ΔH_C is also investigated which indicate that the appropriate value of ΔH_C plays an important role in achieving rapid and high-resolution tsunami simulation using AMR method.

A GPGPU based tsunami simulator is developed and is used to conduct the simulation of run-up tsunami (2011) in Tone River with several meshes of different spatial size. As a result of comparison with field observation data, it is found that the solver can restore run-up tsunami well. Furthermore, the acceleration of GPGPU is verified that 50 times speed-up is achieved by using single GPU (TESLA C1060). Meanwhile, the influence of output interval, subdomain size, load balance, edge-cut as well as the decomposition method are evaluated quantitatively. In detail, the bigger the subdomain is, the higher speed-up will be achieved. And the usage of MRB method will bring 20% additional speed-up comparing with RSB method. At last, the solver is extended to multi-GPUs, and 120 times speed-up is achieved by using three GPUs (Tsubame 2.0, TESLA C2050).

Contents

Chapter1 Introduction 1

1.1 Research background 1

1.2 Numerical simulation for tsunami prevention 2

1.2.1 Challenge I: Complex topography 3

1.2.2 Challenge II: Tremendous computational loads 5

1.3 Previous studies 5

1.3.1 Solution for Challenge I: Unstructured mesh system 5

1.3.2 Solution for Challenge II: Adaptive mesh refinement (AMR) 6

1.3.3 Solution for Challenge II: Parallel computation using GPGPU 7

1.4 Research objective 8

1.5 Outline of this study 9

Chapter2 FVM based tsunami simulator with unstructured mesh and its numerical features 18

2.1 Objective 18

2.2 Governing equations 18

2.3 Finite volume method (FVM) 19

2.3.1 First order scheme (Roe) 20

2.3.2 Second order scheme (SRNH) 21

2.4 Wet/Dry front 23

2.5 Numerical experiment 23

2.5.1 2D tsunami propagate on flat-bed ocean 23

2.5.2 Tsunami propagate on deep-ocean 25

2.5.3 Run-up Tsunami on sloping beach 27

2.6 Application to Tsunami in 2011 Tōhoku Earthquake 28

2.6.1 Initial conditions 28

2.6.2 Results and discussion 29

2.7 Conclusion 31

Chapter3 Rapid and high-resolution tsunami simulator by Adaptive Mesh Refinement (AMR) 62

3.1 Introduction 62

3.2 Methodology 63
3.2.1 Strategies of refining and coarsening 63
3.2.2 Criterions of refining and coarsening 65
3.3 Numerical experiment 66
3.3.1 Tsunami propagate on deep-ocean 66
3.3.2 Run-up tsunami on sloping beach 68
3.4Application to tsunami in 2011 Tōhoku Earthquake 69
3.4.1 Computational conditions 69
3.4.2 Results and discussion 69
3.5 Conclusions 71
Chapter4 Rapid and high-resolution tsunami simulator based on General-Purpose computing on GPU (GPGPU) 90
4.1 Motivation 90
4.2 Introduction of parallel computing using GPU 90
4.3 Basic concept of parallel computing using GPU 91
4.4 Program implementation 92
4.5 Domain decomposition 93
4.5.1 Recursive spectral bisection (RSB) 93
4.5.2 Multi-level spectral bisection (MSB) 95
4.6 Application to run-up tsunami in 2011 Tōhoku Earthquake 95
4.6.1 Computational conditions 96
4.6.2 Comparing with observational data 96
4.6.3 Validity of GPGPU acceleration 98
4.7 Extension to Multi-GPU 101
4.7.1 Multi-GPU implementation 101
4.7.2 Run-time comparison 102
4.8 Conclusions 102
Chapter5 Summary and future work 115
5.1 Summary 115
5.2 Conclusion 116
5.3 Future work 117

Chapter1 Introduction

1.1 Research background

The term tsunami is originated from the Japanese words “津波”, which means a series of water waves caused by the displacement of a large block of water in an ocean or a large lake. According to the formation sources, tsunami can be divided into earthquake tsunami, volcano eruptions tsunami, landslide tsunami or glacier calving tsunami and so on. Fig1.1 shows the world-wide historical tsunami sources from 1628 BC to present time, in which the size of circle is proportional to the event magnitude and the color represents the tsunami intensity on the basis of Soloviev-Imamura scale. It is very clear that most of tsunamis are originated from submarine earthquakes. Besides, Japan is one of the most tsunami prone countries. Table1.1 lists the top six historical tsunami disasters in Japan, which shows that the frequency for tsunami attacking Japan is so high that almost every 100 years there is definitely an enormous tsunami disaster.

Tsunami is seemed to be the most destructive natural disaster especially for the people who are living along the seacoast, not only for its power, but also for its rapid propagating velocity. In particular, the generation of tsunami can be divided into three steps: First, the initial wave generated by an offshore earthquake. The wavelength is up to hundreds kilometers long and it is so small that can’t be distinguished from normal waves. Meanwhile, the tsunami wave will propagate toward land at 800 kilometers per hour which is almost the same as an airplane. Second, the propagating speed of tsunami will decrease when approaches the coast. Therefore, the water will accumulate rapidly as the water becomes shallow, and the wave height will enlarged to even more than 10 meters. The huge water wall runs up to the land and will crush whatever it encounters. Fig1.2 shows the concept of wave height amplification when approaching land. Fig1.3 is an actual photograph of tsunami wave approaching land, and it is clear that the wave is so scaring that even much higher than two-storey house. And Fig1.4, Fig1.5 shows the observational data of wave height of 2011 Tōhoku earthquake tsunami. Third, the

large amount of water is also destructive even returning to the sea, which will sweep peoples and debris from land. Fig1.6 shows the satellite images of Ōfunato Rikuzentakata before and after the 2011 Tōhoku earthquake tsunami. Wherever the tsunami arrived, there will be nothing leave except mud.

Actually, in the 2011 Tōhoku earthquake (also known as the Great East Japan Earthquake), the damage caused by tsunami is far outstrips the earthquake itself. The tsunami caused damage ranging from Hokkaido to Kanagawa Prefecture, and the highest wave of 38.9 meters is also estimated at Omoe peninsula Miyako city, which affirmed that this tsunami achieves the maximum rank of tsunami ever attacked Japan. There are 15,854 people killed, 3,155 missing, 452,000 people were relocated to shelters, and the financial loss is more than $235 billions. In addition, some destructive tsunamis are anticipated to accompany with submarine earthquake such as Tokai and Tonankai earthquake. Therefore, a great diversity of need turned up in migrating or preventing the harmful effects of tsunamis to save lives and fortunes.

1.2 Numerical simulation for tsunami prevention

Given the above characteristics of tsunami, it is so powerful that far beyond human control. And the precise prediction of tsunami is likely an impossible mission with existing technology. There are, however, some technologies that can migrate the damage of tsunami. In which, numerical simulation is found to be effective in tsunami prevention. Take the tsunami warning system of Japan Meteorological Agency (JMA) as an example. Numerical simulation is introduced to the system in 1999 to give accurate tsunami warning. Although the most advanced computers are used, it is still impossible to give prompt tsunami warning. Alternatively, JMA carried out more than hundreds of thousands of trial simulations with various conditions (epicenter locations, earthquake magnitude etc.) in advance to establish a database. In case of an earthquake happened, the wave height, arriving time and the area of influence can be obtained immediately just query the database using the detected epicenter and magnitude. Thus, the tsunami

warning will be announced to peoples just 3 minutes after the earthquake. Furthermore, the results of numerical simulation can also be used to estimate the inundation area to make hazard map, which is important in familiarizing people with the escape routes.

The Method of Splitting Tsunami (MOST) numerical simulation model (Titov & Synolakis, 1995; Titov & Gonzalez, 1997; Titov & Synolakis, 1998) is widely used all over the world, which is developed by the Pacific Marine Environmental Laboratory (PMEL) of the National Oceanic and Atmospheric Administration (NOAA). Titov et al. (1998) have applied the MOST model to 1993 Okushiri, Japan, the 1994 Kuril Island, Russia, and the 1996 Chimbote, Peru tsunami, which verified the effectiveness of the MOST model in real tsunami simulation. The successful cases of the MOST model application also contain the 26 December 2004 Sumatra Tsunami (Titov et al., 2005), and the May 2006 Tonga tsunami (Tang et al., 2008), as well as the Solomon Islands tsunami on April 1, 2007 which widely propagated around the Pacific basin (Gisler, 2008). Furthermore, the MOST model is upgraded to enable real-time tsunami forecasting, which is also succeeded in application of the 2011 Tōhoku earthquake tsunami (Wei et al., 2013). According to the examples listed above, the numerical simulation plays a more and more important role in tsunami prevention. However, there still are some challenges should be tackled in practical application.

1.2.1 Challenge I: Complex topography

Fig1.4 shows the observational inundation height and run-up height of 2011 Tōhoku earthquake tsunami. In view of large scale, the faraway from the epicenter, the lower wave height it will be. But, as shown in Fig1.5, the local wave height distribution appears to be irregular. Actually, comparing with 4 meter's run-up height of Ishinomaki harbor, the run-up wave height in Onagawa harbor is up to 15 meters (The 2011 Tōhoku Earthquake Tsunami Joint Survey (TTJS) Group). It must be point out that both of these two harbors are located in Miyagi prefecture, and the distance is just about 10 kilometers. The reasonable excuse for this big difference of wave height is due to the different topography.

Actually, there already are some studies pointed out that accurate high-resolution topography is one of the key points of tsunami simulation (Wei et al., 2013; Mofjeld et al., 2001; Tang et al., 2008). Therefore, we may get a corollary that the ability of mesh system in discretizing complex topography is also very important for high-resolution tsunami simulation. There are two types of mesh system: one is structured mesh and the other is unstructured mesh. Structure mesh system is composed of square elements (2D) or hexahedral elements (3D) which is characterized by regular connectivity that have many coding advantages. While the unstructured mesh is characterized by irregular connectivity which is skilled in represent complex topography. Fig1.7 shows the brief example of structured mesh and unstructured mesh, indicating that unstructured mesh can represent complex topography much better than the structured mesh especially in river mouth or coastline.

Unfortunately, most of the widely used tsunami models are using structured mesh. The MOST model is using structured rectangular mesh which telescoped to 2 arcsec (~60 meters) based on finite difference method (FDM). In order to realize a real-time simulation, unidirectional nesting from outer mesh is used in this model (Wei et al., 2013). In detail, the outer big meshes provide the inner small meshes with calculated wave height and flow velocities at boundaries by linear interpolation. But this model cannot used to reproduce reflection wave, which is main source of the secondary wave that will extend the flooding area. Another wide used tsunami model is called GeoClaw (George et al., 2006; Berger et al., 2011; LeVeque et al., 2011), which is an open source program use structured mesh based on Godunov-type finite volume method (FVM). Adaptive mesh refinement (AMR) is used in this model to obtain results with sufficient spatial resolution but relatively lower computational load. A series of benchmark simulations are conducted using GeoClaw model and excellent results are obtained (Gonzalez et al., 2011). In both of the models introduced above, some mesh refinement strategies (mesh nesting for MOST model; AMR for GeoClaw model) are used to overcome the limited ability of structured mesh in representing complex topography.

1.2.2 Challenge II: Tremendous computational loads

The computational domain of tsunami simulation is always very huge. Especially, for the ocean-wide tsunami propagation, global simulation is required (Titov et al., 2005). While, as mentioned before, finer mesh is needed to realize high-resolution tsunami simulation. This dilemma causes the computational load became too large to afford. In addition, the tsunami simulation should be accomplished as much case as possible in a certain period of time to enhance the database. Here comes the problem, the rapid computation is hard to achieve in tsunami prevention.

General speaking, there are two choices to solve this problem. First, the adaptive mesh refinement (AMR, refer to Section1.3.2), which can be used to reduce the computational load without resolution decrease. Second, parallel computation using such as general-purpose computing on GPU (GPGPU, refer to Section1.3.3) could accelerate the computation.

1.3 Previous studies

1.3.1 Solution for Challenge I: Unstructured mesh system

Although the unstructured mesh is superior in discretizing complex topography compared with structured mesh. There are relatively few tsunami models using unstructured mesh except ELCIRC/SELFE (Zhang et al., 2004; Zhang et al., 2008a; Zhang et al., 2008b). The SELFE (Semi-implicit Eulerian-Lagrangian Finite Element) is an open source code that based on the 3-D nonlinear shallow-water wave (NSW) equations and solved by finite element method (FEM) using unstructured mesh. Zhang et al. (2008b) have conducted some benchmark test to verify the developed SELFE model that is effective in tsunami simulation. And the SELFE model is also used to reconstruct some historical tsunami such as 1993 Hokkaido-Nansei-Oki tsunami (Zhang et al., 2008b) and 1700 tsunami at Canon Beach, Oregon, USA (Witter et al., 2011). However, the SELFE model is a multi-purpose program that originally designed for the

simulation of baroclinic circulation in cross-scale "river-to-ocean". Though it was extended to tsunami simulation recently (Zhang et al., 2008b), it's not suitable for tsunami forecasting yet. Akoh et al., (2012) have also done valuable work that a tsunami model is constructed using unstructured mesh based on finite volume method (FVM). The developed model is used to simulate the run-up tsunami in Tone River which is triggered by the 2011 Tōhoku earthquake. The computational domain is divided into two parts: an ocean part with coarser mesh and the river part with finer mesh. The nesting technology is used to connect these two parts. The result shows good agreement with the observational data.

Furthermore, to the best knowledge of the author, there is merely quantitative analysis about features of numerical error in tsunami simulation based on finite volume method (FVM) using unstructured mesh, which is also very important in tsunami prevention. For example, there are some detailed researches about the estimation of mesh size's influence on tsunami simulation which is based on finite difference method (FDM) with structured mesh system (Hasegawa et al., 1987; Sayama et al., 1987; Goto et al., 1988; Kawata et al., 1998). By contrast, there barely is quantitative research about the numerical error of tsunami simulator using unstructured mesh system.

1.3.2 Solution for Challenge II: Adaptive mesh refinement (AMR)

Adaptive mesh refinement (AMR) is a method that can be used to refine the mesh locally in regions of greater interest during every time step or selected time steps of computation. Simultaneously, the rest meshes will retain the same size or be coarsened. In this way, the computational load is reduced as well as numerical error is suppressed, which enables high-resolution simulation. The AMR have shown its strong capacity for computational fluid dynamics (CFD) in various filed, such as global atmospheric modeling, numerical cosmology (O'Shea et al., 2004) and hyperbolic partial differential equations (Berger et al., 1989; Huang et al., 2010; MacNeice et al., 2000; Fryxell et al., 2000). However, in view of the computational efficiency, almost all of the popular AMR models are based on the structured mesh (Enzo model by O'Shea et al., 2004;

PARAMESH model by MacNeice et al., 2000; FLASH model by Fryxell et al., 2000).

Few tsunami models with adaptive mesh refinement (AMR) are developed in recent years (George & LeVeque, 2008; Popinet, 2012), most of which are based on structured mesh. George & LeVeque (2008) have introduced the AMR strategy to GeoClaw model, some necessary modifications for the AMR application in tsunami simulation is also described. Popinet (2012) have introduced his work on combining the quadtree-adaptive Saint-Venant solver with the Gerris Flow Solver. And the developed model is also used to simulate the 2011 Tōhoku tsunami. However, there is only one exception for the tsunami simulation with unstructured adaptive mesh refinement (Behrens et al. 2009). Unfortunately, the proposed methods have a fatal weakness in tsunami simulation that it just work well in simple geometry.

1.3.3 Solution for Challenge II: Parallel computation using GPGPU

General-purpose computing on GPU (GPGPU) is the technology using GPU to conduct general computation. GPGPU is still under developing, but consider its high cost-effective in parallel computation and the increasing computing power required by the simulation of complex system, the GPGPU developed very fast and it has already been applied to many fields of CFD. For example, Khajeh-Saeed et al. (2013) have applied the GPGPU for direct numerical simulation (DNS) of turbulence; and Li et al. (2013) have accelerated the simulation of multi-phase flow in porous media using GPGPU successfully. GPGPU is also used to accelerate the shallow water simulations. Brodtkorb et al. (2012) have presented the optimized implementation of Godunov-type central-upwind schemes for solving shallow water equations on GPU. Instead of the acceleration rate, they took much more attention on the efficiency between single and double precision. Lobeiras et al. (2013) have shown their implementation of pollutant transport model based on finite volume method (FVM) with structured mesh. An impressive 238.3x speed-up was obtained using a GPU with Brook+. The GPGPU implementation is also extended to multi-GPUs. Sætra M. L. & Brodtkorb A. R. (2012) have shown their implementation of multi-GPU for solving shallow water equations

based on finite volume method (FVM). Benchmark of dam break over a flat bathymetry is conducted by 4 GPUs and shows the multi-GPUs implementation is also capable of reproducing both analytical and real-world cases.

In addition, though meager, there still are some works about parallel computation of tsunami movement using GPGPU. Liang et al. (2009) have used GPGPU to conduct a simple numerical test of tsunami propagation and inundation based on finite difference method (FDM) with structured mesh. The ability of GPGPU in parallel computation for tsunami simulation is verified. Okamoto et al. (2010) have shown their works on GPU-based large-scale simulation of seismic wave propagation. The numerical model is also based on finite difference method (FDM) with structured mesh. A 45x speed-up is achieved using single GPU. Furthermore, they have extended the model to multi-GPU and obtained an acceptable result. Satria et al. (2012) have applied the GPGPU to accelerate tsunami simulation based on MacCormack scheme. A 223x speed-up is achieved using a single Fermi-generation NVIDIA GPU C2050. However, this implementation is still based on FDM with structured mesh. Table1.2 has listed all of the tsunami models introduced above as well as the two parts of present study.

1.4 Research objective

The main target of this research is to develop rapid and high-resolution tsunami simulator based on finite volume method (FVM) with unstructured triangular mesh. And then, verify the effectiveness of the adaptive mesh refinement (AMR) and the general-purpose computing on graphics processing unit (GPGPU) in achieving rapid and high-resolution tsunami simulation. At last, conduct quantitative evaluations of the numerical errors to investigate the necessary conditions for accurate and reliant tsunami simulation.

1.5 Outline of this study

Through the review of previous studies, the existing problems of tsunami simulation were made clear. First, the topography especially the coastline is very complicate and the reproduction of topography is the key point of tsunami simulation. Although the unstructured mesh system is effective for discretizing complex topography, there almost no Tsunami simulator is using unstructured mesh system. Furthermore, there barely is quantitative research about the numerical error of simulator using unstructured mesh system. Second, the computational domain of tsunami simulation is always very huge and finer mesh is required to get better simulation results. This dilemma causes the computational load became too large to afford. Third, the AMR and GPGPU are seemed to have the potential to reduce the computational load or accelerate the computation to achieve rapid and high-resolution tsunami simulation. These problems or challenges lead to the purposes of this research.

In chapter 2, a tsunami simulator based on finite volume method with unstructured mesh system is developed. Three kinds of idealized numerical experiment are conducted with several meshes of different spatial size to verify the abilities of the developed tsunami simulator as well as to investigate the necessary conditions in achieving high-resolution tsunami simulation. At last, a simulation of tsunami in 2011 Tōhoku Earthquake is also conducted.

In chapter 3, a new strategy of combining adaptive mesh refinement (AMR) with unstructured mesh system is proposed. Two kinds of numerical experiments as well as the simulation of tsunami in 2011 Tōhoku Earthquake are conducted to verify the effectiveness of the proposed AMR strategy.

In chapter 4, a GPGPU accelerated tsunami simulator is developed. The simulation of run-up tsunami (2011) in Tone River is conducted with several meshes of different spatial size. The necessary conditions of GPGPU implementation of run-up tsunami are investigated. In addition, the tsunami simulator is also extended to multi-GPU.

In chapter 5, the main results and conclusions of this research are summarized.

REFERENCE

Titov, V.V. and Synolakis, C.E.: Modeling of breaking and non-breaking long-wave evolution and runup using VTCS-2, J. Waterway, Ports, Coastal and Ocean Engin., Vol. 121(6), pp. 308-316, 1995.

Titov, V. V., and Gonzalez F. I.: Implementation and testing of theMethod of Splitting Tsunami (MOST) model, NOAA Tech. Memo., ERLPMEL-112, 11 pp., Pac. Mar. Environ. Lab., NOAA, Seattle, Wash., 1997.

Titov, V. V., and Synolakis, C. E.: Numerical modeling of tidal wave runup, J. Waterway. Port Coastal Ocean Eng., Vol. 124(4), pp. 157–171, 1998.

Titov, V. V., Rabinovich, A. B., Mofjeld, H. O., Thomson, R. E., and Gonzalez, F. I.: The Global Reach of the 26 December 2004 Sumatra Tsunami, Science, Vol. 309(5743), pp. 2045-2048, 2005.

Gisler, G. R.: Tsunami Simulation, Annu. Rev. Fluid Mech., Vol. 40, pp: 71-90, 2008.

Wei, Y., Chamberlin, C., Titov, V. V., Tang, L. J., and Bernard, E. N.: Modeling of the 2011 Japan Tsunami: Lessons for Near-Field Forecast, Pure Appl. Geophys., Vol. 170, pp. 1309-1331.

Mofjeld, H.O., Titov, V.V., Gonzalez, F.I., and Newman, J.C.: Tsunami scattering provinces in the Pacific Ocean, Geophys. Res. Lett., Vol. 28(2), pp. 335–337, 2001.

Tang, L., Titov, V.V., Wei, Y., Mofjeld, H.O., Spillane, M., Arcas, D., Bernard, E.N., Chamberlin, C., Gica, E., and Newman, J.: Tsunami forecast analysis for the May 2006 Tonga tsunami, J. Geophys. Res., Vol. 113, C12015.

青木尊之: 双曲型方程式の GPU による高速計算, 応用数理, Vol. 20, pp.6-18, 2010.

George, D. L., and LeVeque R. J.: Finite volume methods and adaptive refinement for global tsunami propagation and local inundation, Science of Tsunami Hazards, Vol. 24, pp. 319-328, 2006.

Berger, M. J., George, D. L., LeVeque, R. J., and Mandli, K. M.: The GeoClaw software for depth-averaged flows with adaptive refinement, Advances in Water Resources, Vol. 34, pp. 1195-1206, 2011.

LeVeque, R. J., George, D. L., and Berger, M. J.: Tsunami modeling with adaptively refined finite volume methods, Acta Numerica, Vol. 20, pp. 211-289, 2011.

Gonzalez, F. I., LeVeque, R. J., Chamberlain, P., Hirai, B., Varkovitzky, J., and George, D. L.: Validation of the GeoClaw model, NTHMP MMS Tsunami Inundation Model Validation Workshop, 2011.

Zhang, Y.-L., Baptista, A. M., and Myers, E.P.: A cross-scale model for 3D baroclinic circulation in estuary-plume-shelf systems: I. Formulation and skill assessment, Continental Shelf Res.,

Vol. 24, pp. 2187–2214, 2004.

Zhang, Y.-L., and Baptista, A. M.: SELFE: A semi-implicit Eulerian-Lagrangian finite-element model for cross-scale ocean circulation, Ocean Modeling, Vol. 21(3–4), pp. 71–96, 2008a.

Zhang, Y.-L., and Baptista, A. M.: An Efficient and Robust Tsunami Model on Unstructured Grids. Part I: Inundation Benchmarks, Pure Appl. Geophys., Vol. 165, pp. 1-20, 2008b.

Witter, R. C., Jaffe, B., Zhang, Y.-L., and Priest, G.: Reconstructing hydrodynamic flow parameters of the 1700 tsunami at Cannon Beach, Oregon, USA, Nat Hazards, Vol. 63, pp. 223-240, 2012.

長谷川賢一，鈴木孝夫，稲垣和男，首藤伸夫：津波の数値実験における格子間隔と時間積分間隔に関する研究，土木学会論文集，No.381/II-7，pp.111-120，1987.

佐山順二，今村文彦，後藤智明，首藤伸夫：外海域における津波の高精度計算法に関する検討，海岸工学講演会論文集， 第 34 巻，pp.177-181，1987.

後藤智明，今村文彦，首藤伸夫：遠地津波の数値計算に関する研究，地震，No.41, pp.515-526，1988.

河田恵昭，小池信昭，井上雅夫，嘉戸重仁：わが国沿岸部における遠地津波の伝播特性と数値予報の問題点について，京都大学防災研究所年報，No.41 B-2，pp.383-402，1998.

赤穂良輔，石川忠晴：平成 23 年東北地方太平洋沖地震津波における利根川下流の津波遡上再現計算，水工学論文集，第 56 巻, pp.I_1543-I_1548, 2012.

O'Shea, B. W., Bryan, G., Bordner, J., Norman, M. L., Abel, T., Harkness, R., and Kritsuk, A.: Introducing Enzo, an AMR Cosmology Application, "Adaptive Mesh Refinement - Theory and Applications", Eds. T. Plewa, T. Linde & V. G. Weirs, Springer Lecture Notes in Computational Science and Engineering, pp. 341-351, 2004.

Berger, M. J., and Colella, P.: Local Adaptive Mesh Refinement for Shock Hydrodynamics, J. Computational Physicis, Vol. 82, pp. 64-84, 1989.

Huang, W. Z., Russel, R. D.: Adaptive Moving Mesh Methods, Springer New York Dordrecht Heidelberg London, 2010.

MacNeice P., Olson, K. M., Mobarry, C., Rosalinda de Fainchtein and Packer, C.: PARAMESH: A parallel adaptive mesh refinement community toolkit, Computer Physics Communications, vol. 126, pp.330-354, 2000.

Fryxell, B., Olson, K.; Ricker, P., Timmes, F. X., Zingale, M., Lamb, D. Q., MacNeice, P., Rosner, R., Truran, J. W., and Tufo, H.: FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes, The Astrophysical Journal Supplement Series, Vol 131(1), pp. 273-334, 2000.

George, D. L., and LeVeque, R. J.: High Resolution Methods and Adaptive Refinement for Tsunami Propagation and Inundation, In: Hyperbolic Problems: Theory, Numerics,

Applications, Proceedings of the 11th International Conference on Hyperbolic Problems, Lyon, France, July 2006. Springer: Berlin, pp. 541-549, 2008.

Popinet, S.: Adaptive modeling of long-distance wave propagation and fine-scale flooding during the Tōhoku tsunami, Natural Hazards And Earth System Science, Vol.12, pp. 1213-1227, 2012.

Behrens, J., and Bader, M.: Efficiency considerations in triangular adaptive mesh refinement, Phil. Trans. R. Soc. A, Vol. 367, pp. 4577-4589, 2009.

Khajeh-Saeed, A., Perot, J. B.: Direct numerical simulation of turbulence using GPU accelerated supercomputers, J. Computational Physicis, Vol. 235, pp. 241-257, 2013.

Li, X., Zhang, Y., Wang, X. W., Ge, W.: GPU-based numerical simulation of multi-phase flow in porous media using multiple-relaxation-time lattice Boltzmann method, Chemical Engineering Science, Vol. 102, pp. 209-219, 2013.

Brodtkorb, A., Sætra, M., Altinakar, M.: Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation, Computers & Fluids, Vol.55, pp. 1-12, 2012.

Lobeiras, J., Viñas, M., Amor, M., Fraguela, B. B., García, A. J., Castro, M. J.: Parallelization of shallow water simulations on current multi-threaded systems, International Journal of High Performance Computing Applications, Vol. 27(4), pp. 493-512, 2013.

Sætra, M. L., Brodtkorb, A. R.: Shallow Water Simulations on Multiple GPUs, Applied Parallel and Scientific Computing Lecture Notes in Computer Science, Vol. 7134, pp. 56-66, 2012.

Liang,W.Y., Hsieh, T.J., Satria, M., Chang, Y.L., Fang, J.P., Chen, C.C., and Han, C.C.: A GPU-based simulation of tsunami propagation and inundation. In: Algorithms and Architectures for Parallel Processing, Lecture Notes in Computer Science, Springer Verlag, Berlin/Heidelberg, Vol. 5574, pp. 593–603, 2009.

Okamoto, T., Takenaka, H., Nakamura, T., and Aoki, T.: Accelerating large-scale simulation of seismic wave propagation by multi-GPUs and three-dimensional domain decomposition, Earth Planets Space, Vol. 62, pp. 939-942, 2010.

Satria, M. T., Huang, B., Hsieh, T-J., Chang, Y-L., Liang. W-Y.: Acceleration of Tsunami Propagation Model, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 5(3), pp. 1014-1023, 2012.

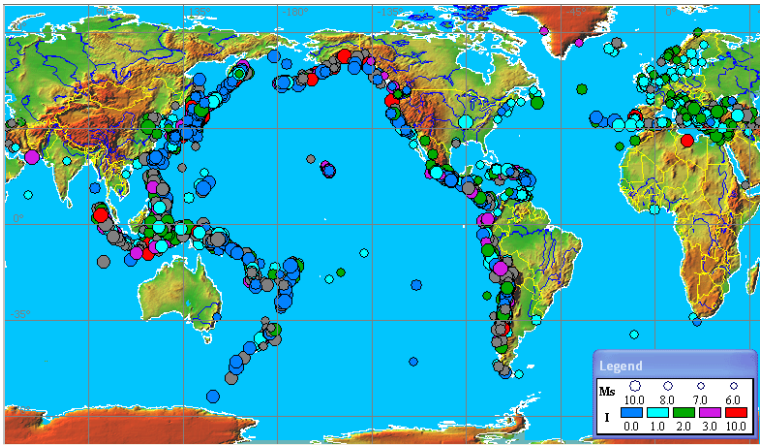


Fig1.1 Historical Tsunami Sources in the World

(Tsunami Lab, Siberian Division Russian Academy of Sciences, URL: http://tsun.sscc.ru/tgi_1.htm)

Table1.1 Historical Tsunami Disasters in Japan

Data(Y/m/d)	Affected Region	Source	Comments
1498/09/20	Kii, Mikawa, Surugu, Izu and Sagami	Enshunada Sea Earthquake(M8.3)	At least 31,000 people killed
1586/01/18	Ise bay	Ise bay Earthquake (M8.2)	8,000 people killed
1707/10/28	Kyushu, Shikoku, Kansai	Nankaido Earthquake (M8.4)	30,000 people killed, 30,000 building damaged
1771/04/24	Ryukyu Islands	Ryuku Islands earthquake (M7.4)	Nearly 12,000 people killed, 3,137 homes destroyed
1896/06/15	East coast of Japan East coast of China	Sanriku Earthquake	22,000 Japanese killed, 4000 Chinese killed
2011/03/11	East coast of Japan	Tōhoku earthquake (M9.0)	15,854 people killed, 3,155 missing, 452,000 people lost home, $235 billion damages

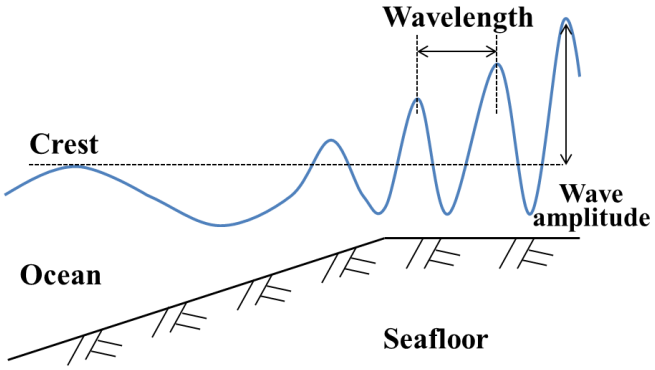


Fig1.2 Amplitude of tsunami wave when approaching land

Fig1.3 Actual photograph of tsunami wave approaching land

(http://yamatokamikaze.blog129.fc2.com/blog-entry-208.html)

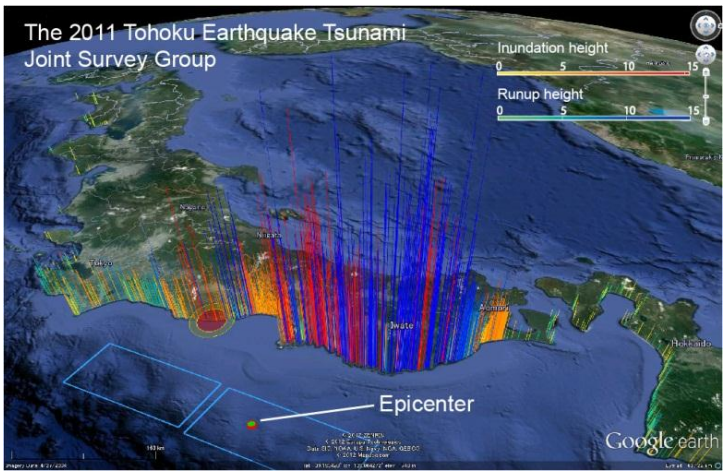


Fig1.4 Observational data of tsunami in 2011 Tōhoku earthquake

(The 2011 Tōhoku Earthquake Tsunami Joint Survey (TTJS) Group, URL: http://www.coastal.jp/ttjt/)

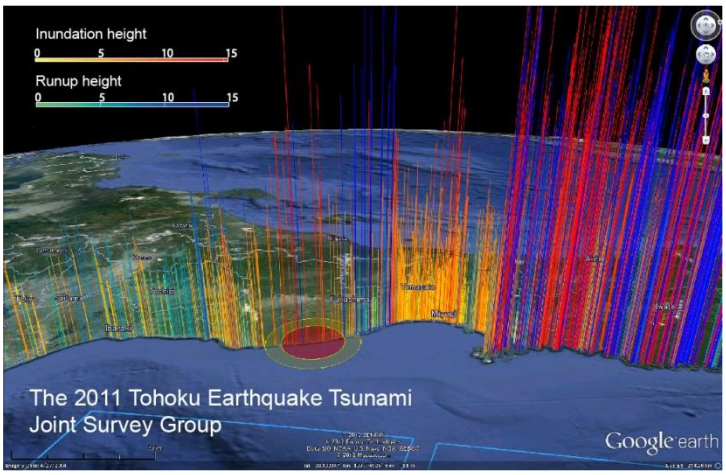


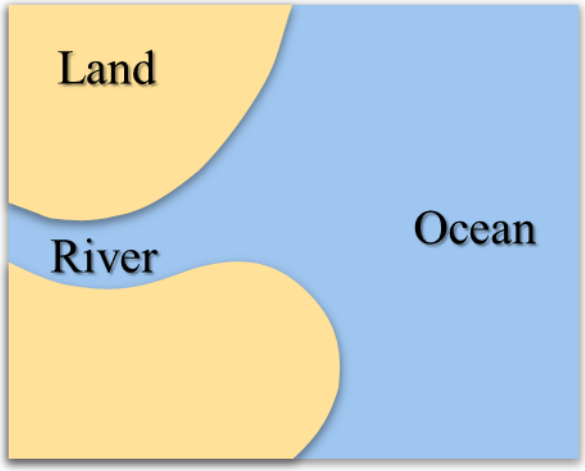
Fig1.5 Observational data of tsunami in 2011 Tōhoku earthquake, enlarged view around Miyagi prefecture and Fukushima prefecture (TTJS Group, URL: http://www.coastal.jp/ttjt/)

(a) Before tsunami attack

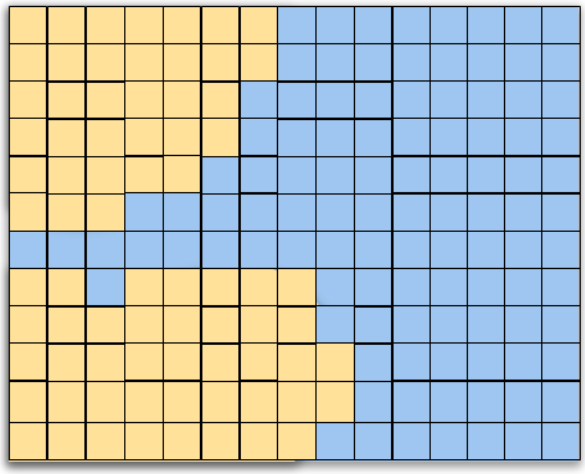
(b) After tsunami attack

Fig1.6 Satellite images of Ōfunato Rikuzentakata before and after the 2011 Tōhoku earthquake tsunami (http://news.mongabay.com/2011/0315-Ofunato.html)

(a)



(b)



(c)

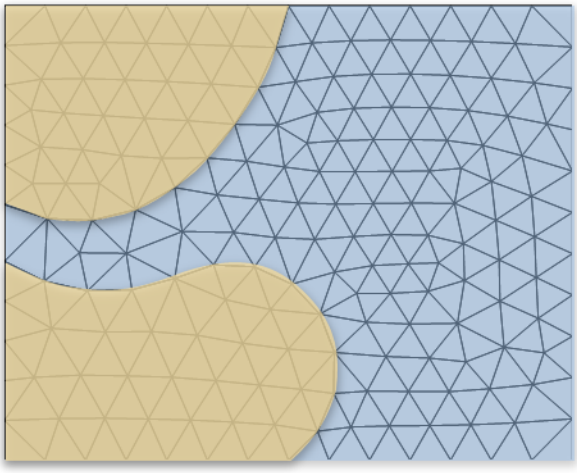


Fig1.7 Structure and unstructured mesh system

(a. Original topography; b. Structured rectangular mesh; c. Unstructured triangular mesh)

Table1.2 Review on tsunami models

Author	Year	Mesh system	Discretization	AMR	Parallel computation	Computing system	Quality of topography
Titov et al.	1997	Structured (with nesting)	FDM	No	Yes	CPU cluster	Medium
Zhang et al.	2004	Unstructured	FEM	No	Yes	CPU cluster	High
George et al.	2006	Structured	FVM	Yes	Yes	CPU cluster	Medium
Liang et al.	2009	Structured	FDM	No	Yes	GPU	Low
Okamoto et al.	2010	Structured	FDM	No	Yes	GPU	Low
Popinet	2012	Structured	FVM	Yes	No	CPU	Medium
Satria et al.	2012	Structured	FDM	No	Yes	GPU	Low
Akoh et al.	2012	Unstructured	FVM	No	No	CPU	High
Present research part I	2013	Unstructured	FVM	Yes	No	CPU	High
Present research part II	2013	Unstructured	FVM	No	Yes	GPU	High

Chapter2 FVM based tsunami simulator with unstructured mesh and its numerical features

2.1 Objective

As mentioned in chapter 1, there merely is tsunami simulator using unstructured mesh based on finite volume method (FVM). Therefore, in this chapter, a tsunami simulator based on FVM with unstructured mesh is developed. And several numerical experiments are conducted to verify the resolution of the developed simulator. And then, the necessary conditions for high-resolution tsunami simulation are also investigated.

2.2 Governing equations

The shallow water equations are depth-averaged equations that are widely used to model the gravity-induced movement of a fluid with free surface. In this research, two-dimensional shallow water equations are used to model the tsunami movements.

$$\frac{\partial}{\partial t}\mathbf{q}+\frac{\partial}{\partial x}\mathbf{F}+\frac{\partial}{\partial y}\mathbf{G}=\mathbf{S}_b+\mathbf{S} \tag{2.1}$$

$$\mathbf{q}=\begin{bmatrix} h \\ hu \\ hv \end{bmatrix},\quad \mathbf{F}=\begin{bmatrix} hu \\ hu^2+gh^2/2 \\ huv \end{bmatrix},\quad \mathbf{G}=\begin{bmatrix} hv \\ huv \\ hv^2+gh^2/2 \end{bmatrix}. \tag{2.2}$$

Where g is the gravitational acceleration, h is the water depth, u and v are the depth-averaged velocities along the x and y direction, respectively. Moreover, $\mathbf{S}_b$ represents the bed slope, and $\mathbf{S}$ is the source terms contain Reynolds stress and bottom friction. The schematic diagram of shallow water equations is shown in Fig2.1.

$$\mathbf{S}_b=\begin{bmatrix} 0 \\ -gh\partial_x z \\ -gh\partial_y z \end{bmatrix},\quad \mathbf{S}=\begin{bmatrix} 0 \\ -\tau_{bx}/\rho+\partial_x(h\tau_{uu})+\partial_y(h\tau_{uv})+f_{cx} \\ -\tau_{by}/\rho+\partial_x(h\tau_{uv})+\partial_y(h\tau_{vv})+f_{cy} \end{bmatrix} \tag{2.3}$$

In which $z(x,y)$ is the bottom elevation, ρ is the water density. And, τ_{bx}, τ_{by} are bottom shear stresses affected from bed surface, τ_{uu}, τ_{uv} and τ_{vv} represent the depth averaged Reynolds stress. Here, Manning formula (2.4) and 0-equation model (2.5) were introduced to calculate the bottom shear stress and Reynolds stress respectively. In addition, f_{cx} and f_{cy} represent the Coriolis force.

$$\tau_i = \left(\tau_{bx}, \tau_{by}\right) = \frac{g\rho n^2}{h^{\frac{1}{3}}} u_i \sqrt{u^2 + v^2} \tag{2.4}$$

$$\tau_{uu} = 2\varepsilon_h \partial_x u - \frac{2}{3}k, \quad \tau_{uv} = \varepsilon_h \left(\partial_y u + \partial_x v\right), \quad \tau_{vv} = 2\varepsilon_h \partial_y v - \frac{2}{3}k \tag{2.5.1}$$

$$k = 2.07 U_* \tag{2.5.2}$$

$$\varepsilon_h = \alpha \frac{\kappa h U_*}{6} \tag{2.5.3}$$

$$U_* = \sqrt{\frac{g n_*^2}{h^{1/3}} \left(u^2 + v^2\right)} \tag{2.5.4}$$

Here, n is the Manning's Roughness Coefficient, k is the with respect to turbulent energy, ε_h is the depth averaged coefficient of virtual viscosity, κ is von Karman's constant, and U_* is friction velocity.

2.3 Finite volume method (FVM)

The finite volume method (FVM) is a numerical method for approximate solution of partial differential equations (PDEs) in integral form (Toro, 1999; LeVeque, 2002). Take the cell-centered one-dimensional finite volume method (FVM) as an example, just as shown in Fig2.2, the numerical solution $\mathbf{q}_i^n$ is an approximation to the average value of the solution in the i-th mesh $[x_{i-1/2}, x_{i+1/2}]$ at time t_n.

$$\mathbf{q}_i^n \approx \frac{1}{\Delta x_i} \int_{x-1/2}^{x+1/2} q\left(x, t_n\right) dx \tag{2.6}$$

In which, Δx_i is the volume of the mesh. In one-dimensional case, the $\Delta x_i = x_{i-1/2} - x_{i+1/2}$ is the length of the mesh. The wave propagation algorithm updates the numerical solution

from t_n to t_{n+1} by solving Riemann problems at mesh boundaries $x_{i-1/2}$ and $x_{i+1/2}$ (LeVeque et al., 2011). The method is applied to conservation laws in this manner.

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^n - \frac{\Delta t}{\Delta x_i}\left(\mathbf{F}_{i+1/2}^n - \mathbf{F}_{i-1/2}^n\right) \tag{2.7}$$

Here, $\mathbf{F}^n_{i-1/2}$ and $\mathbf{F}^n_{i+1/2}$ are the numerical flux approximating the time average of the true flux across the left boundary and right boundary.

The finite volume method (FVM) can be applied using either structured mesh or unstructured mesh in spatial discretization, which provide flexibility in reproducing arbitrary geometries. In this research, a first order Roe scheme and a second order SRNH scheme are used in solving two-dimensional shallow water equations using unstructured triangular mesh.

2.3.1 First order scheme (Roe)

Roe scheme is an approximate Riemann solver based on Godunov's method (Roe, 1981). Fig2.3 shows the numerical flux located in a single triangular mesh and the related parameter arrangement. Time evolution is given by

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^n - \frac{\Delta t}{\Delta s_i}\left(\sum_{j=1}^{3}\left(\mathbf{E}_j \cdot \mathbf{n}_j dl_j\right) - \Delta s_i \mathbf{S}_{b_j}\right)_i^n + \Delta t \mathbf{S}_{f_i}^{\mathbf{n}} \tag{2.8}$$

Where Δs is the area of mesh, $\mathbf{E}_j=(\mathbf{F}, \mathbf{G})_j$ is the numerical flux tensor, $\mathbf{n}_j=(n_x, n_y)_j$ is the outward unit normal vector in each boundary (surface), and $\mathbf{S}_b=\Sigma^3_{j=1}\mathbf{S}_b$ and $\mathbf{S}^n_f$ are integral source term in every cell. The numerical flux is calculated using

$$\mathbf{E}_j \cdot \mathbf{n}_j = \frac{1}{2}\left[\mathbf{E}_R \cdot \mathbf{n} + \mathbf{E}_L \cdot \mathbf{n} - \mathbf{R}|\Lambda|\mathbf{R}^{-1}\left(\mathbf{q}_R - \mathbf{q}_L\right)\right] \tag{2.9.1}$$

$$\mathbf{S}^*_{b_j} = \mathbf{R}\left(\mathbf{I} - |\Lambda|\Lambda^{-1}\right)\mathbf{R}^{-1}\mathbf{S}_{b_j} \tag{2.9.2}$$

with $\mathbf{I}$ suggests unit matrix, $\mathbf{R}$ refer to eigenvector of Jacobian matrix, and Λ suggests diagonal matrix constituted by eigenvalues.

$$\mathbf{R}=\begin{bmatrix} 1 & 0 & 1 \\ \bar{u}+\bar{c}\cdot n_{xj} & -\bar{c}\cdot n_{yj} & \bar{u}-\bar{c}\cdot n_{xj} \\ \bar{v}+\bar{c}\cdot n_{yj} & \bar{c}\cdot n_{xj} & \bar{v}-\bar{c}\cdot n_{yj} \end{bmatrix} \tag{2.10.1}$$

$$\mathbf{R}^{-1}=\frac{1}{2\bar{c}}\begin{bmatrix} \frac{1}{2}\left(-\bar{u}\cdot n_{xj}-\bar{v}\cdot n_{yj}+\bar{c}\right) & \frac{1}{2}n_{xj} & \frac{1}{2}n_{yj} \\ \bar{u}\cdot n_{yj}-\bar{v}\cdot n_{xj} & -n_{yj} & n_{xj} \\ \frac{1}{2}\left(\bar{u}\cdot n_{xj}+\bar{v}\cdot n_{yj}+\bar{c}\right) & -\frac{1}{2}n_{xj} & -\frac{1}{2}n_{yj} \end{bmatrix} \tag{2.10.2}$$

$$\left|\Lambda\right|=\begin{bmatrix} \left|\bar{u}+\bar{c}\right| & 0 & 0 \\ 0 & \left|\bar{u}\right| & 0 \\ 0 & 0 & \left|\bar{u}-\bar{c}\right| \end{bmatrix} \tag{2.10.3}$$

$$\mathbf{S}_{b_j}=\left[0,-g\frac{h_L+h_R}{2}\Delta z\cdot n_x,-g\frac{h_L+h_R}{2}\Delta z\cdot n_y\right] \tag{2.10.4}$$

Where, $\bar{u}$ and $\bar{v}$ are Roe's average velocity, and $\bar{c}=\sqrt{g\bar{h}}$ is average wave velocity.

$$\bar{u}=\frac{\sqrt{h_R}u_R+\sqrt{h_L}u_L}{\sqrt{h_R}+\sqrt{h_L}},\quad \bar{v}=\frac{\sqrt{h_R}v_R+\sqrt{h_L}v_L}{\sqrt{h_R}+\sqrt{h_L}} \tag{2.10.5}$$

2.3.2 Second order scheme (SRNH)

The SRNH scheme is a recently developed finite volume non-homogeneous Riemann solver for solving shallow water equations on non-flat bottom using unstructured mesh (Benkhalldoum et al., 2010). Here, the acronym is originated from French words: "Solveur de Riemann Non Homogène ".

The SRNH scheme consists of two stages: Predictor stage (2.11.1) and Corrector (2.11.2). It can be expressed by

$$\mathbf{q}_{ij}^{n}=\frac{1}{2}\left(\mathbf{q}_i^n+\mathbf{q}_j^n\right)-\frac{1}{2}\mathrm{sgn}\left[\nabla\xi\left(\bar{\mathbf{q}}_{ij}^n;\mathbf{n}_{ij}\right)\right]\left(\mathbf{q}_j^n-\mathbf{q}_i^n\right)+\frac{1}{2}\left|\nabla\xi\left(\bar{\mathbf{q}}_{ij}^n;\mathbf{n}_{ij}\right)^{-1}\right|\mathbf{S}_{ij} \tag{2.11.1}$$

$$\mathbf{q}_i^{n+1}=\mathbf{q}_i^n-\frac{\Delta t}{\Delta s_i}\sum_{j=1}^{3}\xi\left(\mathbf{q}_{ij}^n;\mathbf{n}_j dl_j\right)+\frac{\Delta t}{\Delta s_i}\mathbf{S}_i^n \tag{2.11.2}$$

In which, sgn[**A**] denotes the sign matrix of **A**.

A MUSCL method incorporating Minmod slope limiter is used to realize second order accuracy in spatial.

$$\mathbf{q}_{ij} = \mathbf{q}_i + \frac{1}{2}\lim\left(\frac{2}{3}\nabla\mathbf{q}_i \cdot \vec{\mathbf{X}}_i\vec{\mathbf{X}}_j + \frac{1}{3}(\mathbf{q}_j - \mathbf{q}_i), \mathbf{q}_j - \mathbf{q}_i\right) \times (\mathbf{q}_j - \mathbf{q}_i) \tag{2.12.1}$$

$$\mathbf{q}_{ji} = \mathbf{q}_j - \frac{1}{2}\lim\left(\frac{2}{3}\nabla\mathbf{q}_j \cdot \vec{\mathbf{X}}_i\vec{\mathbf{X}}_j + \frac{1}{3}(\mathbf{q}_j - \mathbf{q}_i), \mathbf{q}_j - \mathbf{q}_i\right) \times (\mathbf{q}_j - \mathbf{q}_i) \tag{2.12.2}$$

Here, $\mathbf{X}_i$ and $\mathbf{X}_j$ are the barycenter coordinates of mesh i and j, the Minmod limiter given by

$$\lim(a,b) = \max\left(0, \min\left(1, \frac{a}{b}\right)\right) \tag{2.12.3}$$

And the mesh gradients are defined as

$$\nabla\mathbf{q}_i = (\partial_x\mathbf{q}_i, \partial_y\mathbf{q}_i)^T \tag{2.12.4}$$

Project the shallow water equations on the local mesh outward normal η and tangential normal τ, we can get the average state as

$$\overline{\mathbf{q}} = \begin{pmatrix} \overline{h} \\ \overline{h}\overline{u}_\eta \\ \overline{h}\overline{u}_\tau \end{pmatrix} = \frac{1}{2}(h_i + h_j)\begin{pmatrix} 1 \\ \dfrac{\sqrt{h_i}u_i + \sqrt{h_j}u_j}{\sqrt{h_i} + \sqrt{h_j}}\eta_x + \dfrac{\sqrt{h_i}v_i + \sqrt{h_j}v_j}{\sqrt{h_i} + \sqrt{h_j}}\eta_y \\ -\dfrac{\sqrt{h_i}u_i + \sqrt{h_j}u_j}{\sqrt{h_i} + \sqrt{h_j}}\eta_y + \dfrac{\sqrt{h_i}v_i + \sqrt{h_j}v_j}{\sqrt{h_i} + \sqrt{h_j}}\eta_x \end{pmatrix} \tag{2.13}$$

The sign and inverse matrices of the Jacobian matrix are defined as

$$\mathrm{sgn}\left[\nabla\xi_\eta(\overline{\mathbf{q}})\right] = \mathbf{R}(\overline{\mathbf{q}})\mathrm{sgn}\left[\Lambda(\overline{\mathbf{q}})\right]\mathbf{R}^{-1}(\overline{\mathbf{q}}) \tag{2.14.1}$$

$$\left|\nabla\xi_\eta(\overline{\mathbf{q}})^{-1}\right| = \mathbf{R}(\overline{\mathbf{q}})\mathrm{sgn}\left|(\Lambda(\overline{\mathbf{q}}))^{-1}\right|\mathbf{R}^{-1}(\overline{\mathbf{q}}) \tag{2.14.2}$$

Where, the right eigenvector and the diagonal matrices are reconstructed as

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 1 \\ \overline{u}_\eta - \overline{c} & 0 & \overline{u}_\eta + \overline{c} \\ \overline{u}_\tau & 1 & \overline{u}_\tau \end{bmatrix} \tag{2.14.3}$$

$$\mathbf{R}^{-1}=\begin{bmatrix} \frac{1}{2\bar{c}}\left(\bar{u}_\eta+\bar{c}\right) & -\frac{1}{2\bar{c}} & 0 \\ -\bar{u}_\tau & 0 & 1 \\ -\frac{1}{2\bar{c}}\left(\bar{u}_\eta-\bar{c}\right) & \frac{1}{2\bar{c}} & 0 \end{bmatrix} \tag{2.14.4}$$

$$\Lambda=\begin{bmatrix} \bar{u}_\eta-\bar{c} & 0 & 0 \\ 0 & \bar{u}_\eta & 0 \\ 0 & 0 & \bar{u}_\eta+\bar{c} \end{bmatrix} \tag{2.14.5}$$

2.4 Wet/Dry front

In this research, the wet/dry front is handled using the method proposed by Castro (2005), which are defined as:

a. Set the criterion of water depth ε, the mesh that satisfy $h \le \varepsilon$ is tagged as dry mesh, otherwise, is defined as wet mesh.

b. Set the variables **q** of dry mesh equal to 0, and set the wet/dry front as wet boundary.

c. Redefine the bottom level as follow

$$z_R'=\begin{cases} h_L+z_L & \textbf{if} h_L+z_L<z_R \\ z_R & \textbf{otherwise} \end{cases} \tag{2.15}$$

The concept of this procedure is also shown in Fig2.4.

2.5 Numerical experiment

In order to verify the abilities of developed tsunami simulator, as well as the necessary conditions for high-resolution tsunami simulation, three kinds of idealized numerical experiments are conducted with several meshes of different size.

2.5.1 2D tsunami propagate on flat-bed ocean

This numerical experiment is designed to investigate the accuracy of Roe scheme and SRNH scheme in simulating two-dimensional tsunami propagate by means of

unstructured mesh. Initial conditions are shown in Fig2.5, the computational domain is 5000 km × 5000 km square and the bottom is flat, the water depth is set to be 4000 meters which is close to average depth of ocean. Furthermore, the wave length is set to be 800 km and the maximum initial wave height is set to be 20 meters in consideration of the wave height of 2011 Tōhoku earthquake tsunami. The initial wave height is given by a Guass distribution.

$$h=\begin{cases}20\times\exp\left(-\left(\left(x-2.5\times10^{6}\right)^{2}+\left(y-2.5\times10^{6}\right)^{2}\right)/10^{11}\right) & \textbf{if}\ r\le 8\times10^{5}\\ 0 & \textbf{otherwise}\end{cases} \tag{2.16}$$

In this numerical experiment, four kinds of meshes are used just as shown in Table2.1 and Fig2.6 (just shown F1 and F2) that the mesh size are various from 6 km to 50 km with mesh number up to 8.7×10^{5}. Time step is set to 0.5s. The Reynolds stress and bottom friction are not considered. In addition, for the case of SRNH application, the Minmod slope limiter is not used. All the four boundaries are set as outflow boundary.

Time series results of mesh F4 are shown in Fig2.7 by contour figure, in which the color represents wave height. Fig2.8 shows the results in t=30 min along middle-line (refer to the red line A-B in third figure of Fig2.7) using Roe scheme (solid line) and the SRNH scheme (dash line) with different meshes. The green, blue, red and brown refer to the results using mesh F1, F2, F3 and F4 respectively. The upper two figures are the enlarged view of right wave crest that shows the difference of Roe scheme's results using different mesh are much bigger than SRNH scheme, which indicate that the accuracy of tsunami simulation is depend on numerical scheme. In addition, for the case of SRNH scheme, even the coarsest mesh F1 is used, we can also get almost the same result as Roe's second finest mesh's result. In other words, the accuracy of Roe scheme is much more depend on mesh size compared with SRNH scheme.

Considering the accuracy of SRNH scheme is higher than Roe scheme, the result of SRNH using finest mesh F4 is used as the "theoretical wave height: H_{theo}" to calculate the relative errors (Err=(H_{theo}- H_{cal})/ H_{theo}). Just as shown in Fig2.9, the Roe scheme is verified to be first-order accuracy and the SRNH scheme is second-order accuracy.

In conclusion, the developed tsunami simulator with unstructured mesh based on finite volume method is effective in reproduce two-dimensional tsunami propagation. Besides, the accuracy of Roe scheme and SRNH scheme is confirmed.

2.5.2 Tsunami propagate on deep-ocean

The above numerical experiment (2.5.1) is the most idealized case in which the wave is propagating symmetrically from the epicenter. In order to find out the necessary conditions in achieving high-resolution tsunami simulation, the simulation of tsunami propagate on deep-ocean is also conducted in this research.

Fig2.10 shows the initial conditions of this numerical experiment. The computational domain is set to be 8000 km long and 800 km wide. The water depth is also 4000 meters and the bottom is flat-bed. The wave length is set to be 400 km in view of a possible macro-quake happening in deep-ocean. The maximum height of initial wave is 2 meters (relative wave height: 1/2000) and the wave height is given by

$$h = \begin{cases} \cos\left(\dfrac{2\pi}{4\times10^5}\left(x-\dfrac{8\times10^6}{2}\right)\right)+1 & \textbf{if} \; 3.8\times10^6 \le x \le 4.2\times10^6 \\ 0 & \textbf{otherwise} \end{cases} \tag{2.17}$$

Bottom friction and Reynolds stress are considered, and the Manning's roughness coefficient is set as 0.02. The time step is set to ensure the CFL<0.1. For the case of SRNH application, the Minmode slope limiter is enforced. The two boundaries perpendicular to the direction of tsunami movement is set as outflow boundary, and the rest two boundaries are set as periodic boundary. Five kinds of meshes are used which are various from 5 km to 80 km just as shown in Table2.2 and Fig2.11.

Because the relative wave height is 1/2000 << 1, this wave can be approximately treated as small amplitude wave that will split into two sub-waves and both of them with half of initial wave height (1 meter), theoretically. Fig2.12 shows the time series results along transverse middle-line using mesh A1-A5. Although the results of SRNH scheme are better than Roe scheme in all five cases, but sharply wave decay is observed

both in Roe scheme and SRNH scheme's result when using coarse mesh such as A1, A2 and A3. For the case of coarsest mesh A1, the attenuation increases quickly as the propagate distance increase.

Here, the attenuation is defined as H_{cal}/ H_{theo} and its relationships with mesh size and propagate distance are shown in Fig2.13, which indicate that as the propagate distance increases, the difference of attenuation among different mesh will increase. While, the difference of attenuation between Roe and SRNH will also increases according distance.

Furthermore, if "N" is defined as the mesh number in one wave length, we can get its relationship with relative error according to propagation distance (350km, 1000km and 2100km) just as shown in Fig2.14. These figures show clearly that the relative error decreases accompany with mesh number "N" increases, which illustrate that the mesh size will affect the resolution of tsunami simulation seriously. In detail, Hasegawa et al. (1987) proposed a criterion that if the relative error is smaller than 0.1, the result is seemed to be exact enough in tsunami simulation, which is also accepted by the Ministry of Land, infrastructure, Transport and Tourism (MLIT) and have already been applied to guide tsunami prevention over Japan. According to this criterion, the required mesh number "N_{req}" for tsunami propagate on deep-ocean is summarized in Table2.3, which indicates that the required mesh number "N_{req}" also increases along with the propagating distance increase. Note that the distance 350km is chose in view of the farthest distance between main disaster area and epicenter of tsunami in 2011 Tōhoku Earthquake. Besides, as the propagating distances increase, the increment of Roe scheme for the "N_{req}" is much bigger than SRNH scheme. For example, the "N_{req}" of Roe scheme grow tenfold (from 30 to 400) accompany with the propagating distance change from 350 km to 2100 km. By contrast, the "N_{req}" of SRNH just grow two times (from 15 to 35) for the same case.

In conclusion, the numerical method and mesh size as well as propagate distance will affect the tsunami simulation. The required mesh number "N_{req}" for deep-ocean tsunami propagation is summarized.

2.5.3 Run-up Tsunami on sloping beach

This numerical experiment is designed to testify the ability in treating sharply increased wave height in coastline. In this simulation, five kinds of meshes are used just as shown in Table2.4 and Fig2.16. The computational domain is set to be 200 km long and 20 km wide. The detail of the initial conditions are shown in Fig2.15, which are set to meet the Carrier & Greenspan's (1957) theory to realize the comparison between computational results and theoretical solutions. Besides, considering the bottom slope around coastline is very gentle (e.g., the steepest bottom is about 1/100 in Ryouri Bay in Ōfunato, Northeast Japan), the bottom slope is set to be 1/500 in this experiment. The initial wave height is given by

$$h=\begin{cases}1.005917\times\exp\left(-4.56976\times10^{-9}\times\left(x-10^{5}\right)^{2}\right) & \textbf{if}\ 5.0\times10^{4}\le x\le1.5\times10^{5}\\ 0 & \textbf{otherwise}\end{cases} \tag{2.18}$$

Bottom friction and Reynolds stress are considered in this calculation, and the Manning's roughness coefficient is set as 0.02. The time step is set to ensure the CFL<0.1. For the case of SRNH application, the Minmode slope limiter is enforced. The boundary of land side is set as reflection boundary, and the boundary perpendicular to the direction of tsunami movement is set as outflow boundary, the rest two boundaries are set as periodic boundary.

The time series results along the middle-line that parallel to the direction of tsunami movement using different mesh are shown and compared with Carrier & Greenspan's theoretical solution in Fig2.17 (i.e., 5 figures correspond to 5 meshes from coarse to fine). The dependence of simulation results on mesh size can also be verified again from Fig2.17.1 to Fig2.17.5. The results of first 40 minutes' tsunami movement fit the Carrier & Greenspan's solution much better than the results of 50-80 minutes in all cases. However, numerical oscillations arise in the results of 50-80 minutes. The possible reason is that the wave arrived at coast and crashed with the sloping beach around 50 minutes, while the wave broken and behaved as discontinuous wave.

Fig2.18 shown us the quantitative comparison between Roe scheme and SRNH scheme, in which, Fig2.18.1 shows the relative error according to mesh size from 10-50 minutes, and Fig2.18.2 shows the relative error according to mesh number "N" per wave length. As seen from these figures, the results fit Carrier & Greenspan's solution very well except the result of 50 minutes. In addition, the difference between Roe scheme and SRNH scheme is not so big which is considered as a result that there is no transformation of the topography along longitudinal direction.

The estimation of maximum wave height in coastline with tiny water depth is still a challenge for tsunami simulation. Usually, the maximum wave height is estimated by using Green's law (Synolakis, 1991) instead of direct computation. Note that the Green's law is also applied by Japanese government on forecasting tsunami (MLIT, FDMA, and JMA, 1998). Specifically, the maximum wave height is estimated using $H_{max}=k\times H/h^{1/4}$, where k is statistical ratio which is equal to 1 usually, h is the undisturbed local water depth and H is the wave height. In this research, Green's law is used to estimate the maximum wave height in coastline at t=50 minutes, and the results is shown in Fig2.19. As seen from the figure, about 200 meshes per one wave length is required to get accurate results for the run-up tsunami on sloping beach, not only for the Roe scheme, but also for the SRNH scheme. The details of required mesh number "N_{req}" are summarized in Table2.5.

In conclusion, the ability of the developed simulator in treating sharply increased wave height in coastline is verified. The required mesh number "N_{req}" for run-up tsunami propagation is summarized.

2.6 Application to Tsunami in 2011 Tōhoku Earthquake

2.6.1 Initial conditions

For the sake of validating the developed simulator in solving real-tsunami movement, it is also applied to simulate the Tsunami in 2011 Tōhoku Earthquake. Fig2.20.1 shows the computational domain and the initial water wave distribution, which is calculated by

Tohoku University model (Imamura et al., 2011). The topography is shown in Fig2.20.2. The starting time is set as 14:52 11th March, and the time step is adjusted according to CFL condition (<0.3) at each step. The Reynolds stress and bottom friction are considered in this application, which Manning's roughness coefficient is set as 0.02 in ocean and 0.04 in land respectively. A one hour's tsunami propagation is simulated using three kinds of meshes, which are shown in Table2.6 and Fig2.21.

The GPS observational data of Nationwide Ocean Wave information network for Ports and HArbourS (NOWPHAS) are used for the verification simulation. The GPS data is record every minutes, and six observation station (Fig2.21) are choose for the following discussion: a, north part of Iwate prefecture (NI, 17km away from the coastline); b, center part of Iwate prefecture (CI, 14km away from the coastline); c, south part of Iwate prefecture (SI, 14km away from the coastline); d, north part of Miyagi prefecture (NM, 18km away from the coastline); e, center part of Miyagi prefecture (CM, 13km away from the coastline); f, Fukushima prefecture (FK, 20km away from the coastline). In addition, the observational data of water gauge located 1km away from the Tone River mouth is also used in comparison (TR).

2.6.2 Results and discussion

Fig2.22 shows the time series contour figures of tsunami propagating in ocean till 60 minutes after the earthquake (meshT1, Roe scheme). Fig2.23.1 shows the computational results using Roe scheme with coarse mesh T1 and finer mesh T3 comparing with GPS observational data in each observation location. It can be seen from the figures that although the arriving time of first wave fits observational data well, the wave height of computational results are much smaller than the GPS observational data. Furthermore, the improvement of using finer mesh T3 is not so big. The possible reason is that in mesh T3, just the meshes around epicenter and coastline are refined, which cause the near shore (the region between epicenter and coastline) size of mesh T3 is almost the same as mesh T1. On this occasion, considering the conclusion of Section2.5.2 that the required mesh number for tsunami propagating less than 350km is 30 (see Table2.3, Roe

scheme), mesh T3R is made that the meshes in near-shore region are refined to almost the same size as coastline (Fig2.21.2, Table2.6). In consequence, the mesh number is upgraded from 7 to 35 according to the mesh size in near-shore area. The comparison between results of mesh T3 and mesh T3R is shown in Fig2.23.2, in which the wave heights of mesh T3R are lean towards the observational data much more than result of mesh T3. Moreover, as shown in Fig2.23.3, it is easy to observe the improvement of using finer mesh in tsunami simulation. And this improvement has demonstrated that the conclusions of the former numerical experiments (Section2.5.2, Table2.3) have guiding significance for achieving high-resolution tsunami simulation indeed.

What calls for special attention is that, the improvement of mesh T3R over T3 are concentrated in NM and CM (see Fig2.23.2e). Here, let's take the observation station CM as example to investigate the possible reason. First, draw a cross line A-A' just as shown in Fig2.24. The time variation of tsunami propagation along A-A' is shown in Fig2.25, which indicate that the usage of finer mesh have made the wave more compact thus the wave height is improved. This tendency could also be observed in the enlarged view of tsunami propagate near CM (Fig2.26) using mesh T3 and T3R respectively. In addition, consider the CM is very close to the coastline, the wave height is also very easy to be affected by the reflection wave.

On the other hand, although the finer mesh is used, the computational results are still much lower than observational data especially in NI, CI and SI. A possible reason is that the resolution of topography data in these areas is coarse.

Furthermore, let's have a look at the comparison between Roe scheme and SRNH scheme (Fig2.27). There is no big difference between Roe scheme and SRNH scheme not only for the results of mesh T1, but also for the results of mesh T3R. The possible reason is shown in Fig2.28, which is the mesh in epicenter area is not fine enough even for SRNH scheme (N_{req}>15, see Table2.3) to achieve high-resolution tsunami simulation.

At last, the run-time of each cases are also recorded, and for the case using mesh T3R, the run-time for one hour's tsunami propagation will cost more than 5 hours, which is unacceptable for tsunami prevention. The urgent need for accelerating the computational speed leads to the topics of AMR (Chapter3) and GPGPU (Chapter4).

2.7 Conclusion

In this chapter, three kinds of idealized numerical experiment are conducted with several meshes of different spatial size to verify the effectiveness of the developed tsunami simulator as well as the required conditions for high-resolution tsunami simulation. The quantitative error analysis is conducted and indicates that not only the numerical scheme, but also the mesh size as well as propagate distance will affect the accuracy of tsunami simulation. The required mesh number "N_{req}" per wavelength are summarized for deep-ocean tsunami movement and run-up tsunami respectively. In addition, the simulation of the Tsunami in 2011 Tōhoku Earthquake is also conducted, and the summarized required mesh number "N_{req}" is applied in this simulation and is proved to be effective in achieving high-resolution tsunami simulation.

REFERENCE

Toro, E. F.: Riemann Solvers and Numerical Methods for Fluid Dynamics – A Practical Introduction (3rd Edition), *Springer-Verlag Berlin Heidelberg*, 1999.

LeVeque, R. J.: Finite Volume Methods for Hyperbolic Problems, *Cambridge University Press*, 2002.

LeVeque, R. J., George, D. L., and Berger, M. J.: Tsunami modeling with adaptively refined finite volume methods, *Acta Numerica*, Vol. 20, pp. 211-289, 2011.

Roe, P. L.: Approximate Riemann Solvers, Parameter Vectors and Difference Scheme, *Journal of Computational Physics*, Vol. 43, pp. 357-372, 1981.

Benkhaldoun, F., Elmahi, I., Seaïd, M.: A new finite volume method for flux-gradient and source-term balancing in shallow water equations, *Comput. Methods Appl. Mech. Engrg.*, Vol. 199, pp. 3324-3335, 2010.

Castro, M. J., Ferreiro Ferreiro, A. M., García-Rodríguez, J. A., González-Vida, J. M., Macías, J., Parés, C., Elena Vázquez-Cendón, M.: The Numerical Treatment of Wet/Dry Fronts in Shallow Flows: Application to One-Layer and Two-Layer Systems, *Mathematical and Computer Modelling*, Vol. 42(3-4), pp. 419-439, 2005.

長谷川賢一，鈴木孝夫，稲垣和男，首藤伸夫：津波の数値実験における格子間隔と時間積分間隔に関する研究，土木学会論文集，No.381/II-7，pp.111-120，1987.

Carrier, G. F., and Greenspan, H. P.: Water waves of finite amplitude on a sloping beach, *Journal of Fluid Mechanics*, vol. 4, pp.97-109, 1957.

Synolakis, C. E.: Green's law and the evolution of solitary waves, *Physics of Fluids A: Fluid Dynamics*, Vol. 3, pp. 490-491, 1991.

国土庁，気象庁，消防庁: 津波災害予測マニュアル一地域防災計学における津波対策強化の手引き, 1998.

今村文彦，越村俊一，大家隆行，馬渕幸雄，村嶋陽一(2011): 東北地方太平洋沖地震を対象とした津波シミュレーションの実施 東北大学モデル(Version1.0)，東北大学大学院工学研究科 付属災害制御研究センター.

Ministry of Land, Infrastructure, Transport and tourism, NOWPHAS, URL: http://www.mlit.go.jp/kowan/nowphas/index_eng.html

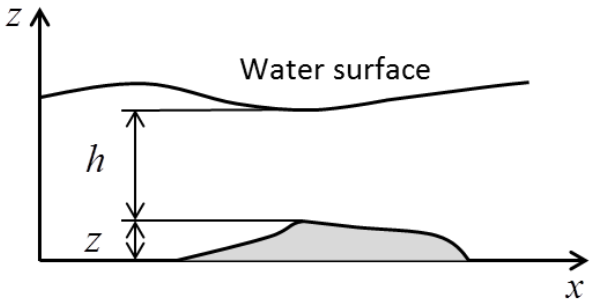


Fig2.1 Schematic diagram of shallow water equations

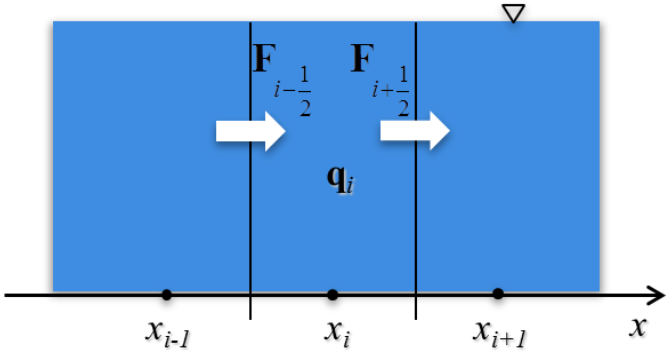


Fig2.2 Concept map of cell-centered Finite Volume Method (FVM) in one-dimensional

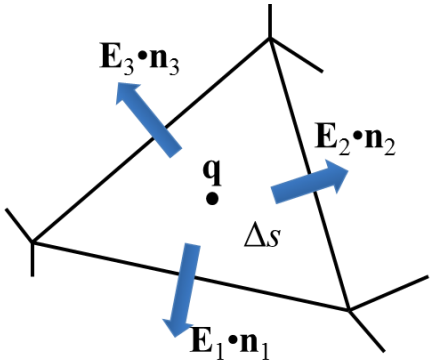
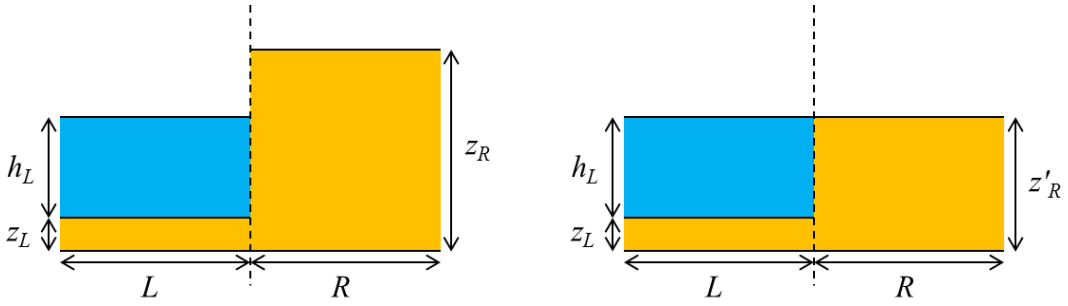


Fig2.3 Numerical flux located in a single triangle and parameter arrangements



(a) Wet/Dry front (b) Redefinition of the bottom function

Fig2.4 The wet/dry front treatment

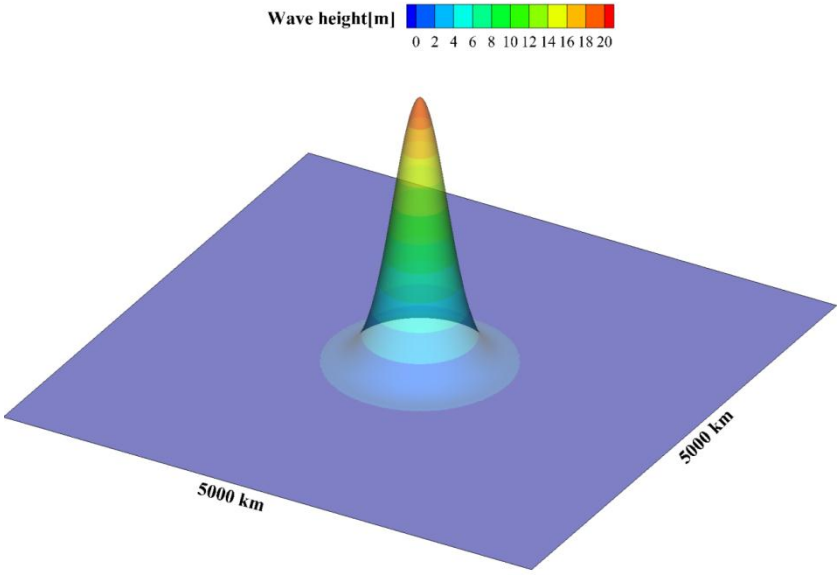


Fig2.5.1 Initial conditions of 2D tsunami propagate on flat-bed ocean

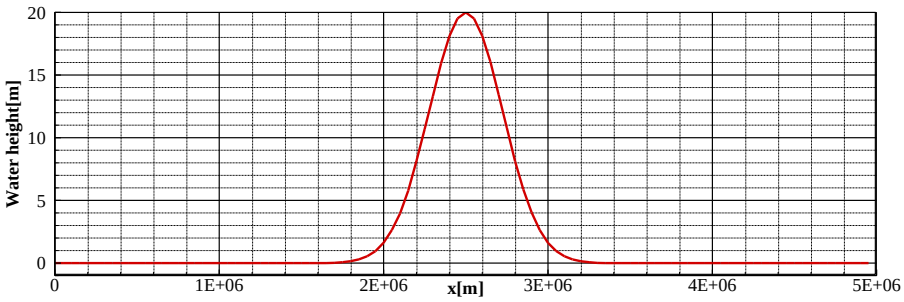


Fig2.5.2 Cross-section view of initial wave for 2D tsunami propagate

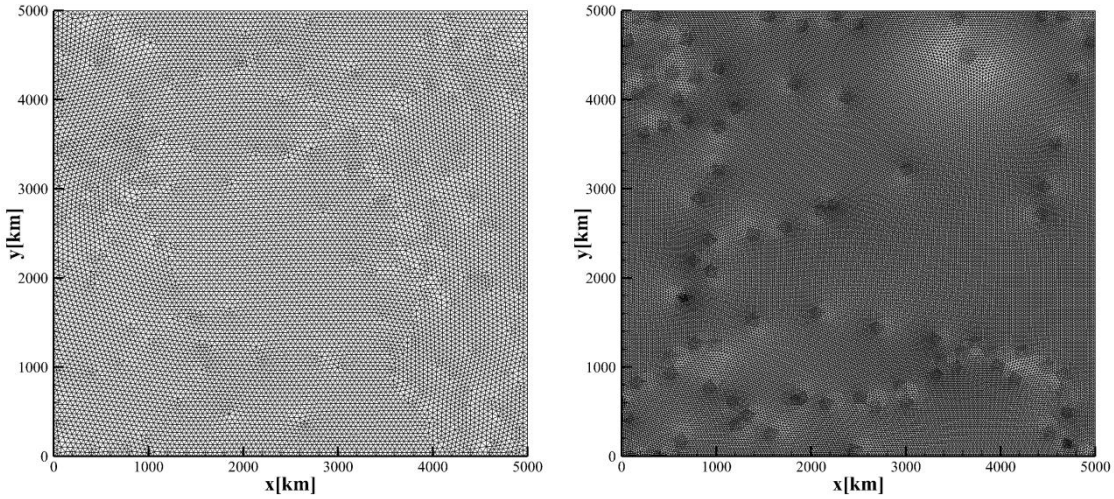


Fig2.6 Mesh used for 2D tsunami propagate (Left: mesh F1; Rigt: mesh F2)

Table2.1 Mesh for 2D tsunami propagate on flat-bed ocean

Mesh	F1	F2	F3	F4
Total number	22,804	90,590	362,360	870,110
Mesh size [km]	50	23	13	6

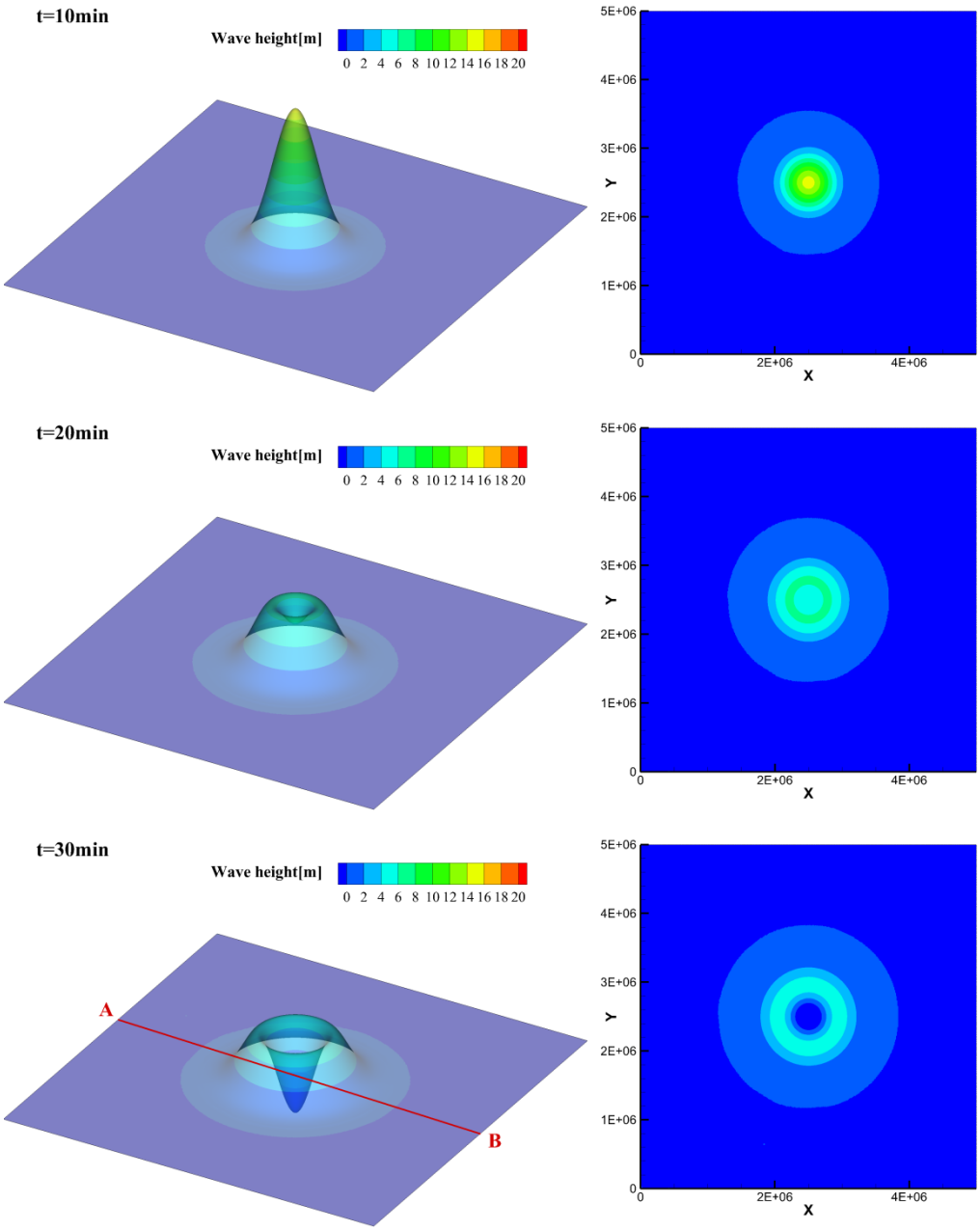


Fig2.7 Time series results of 2D wave propagate on flat-bed ocean (Mesh F4)

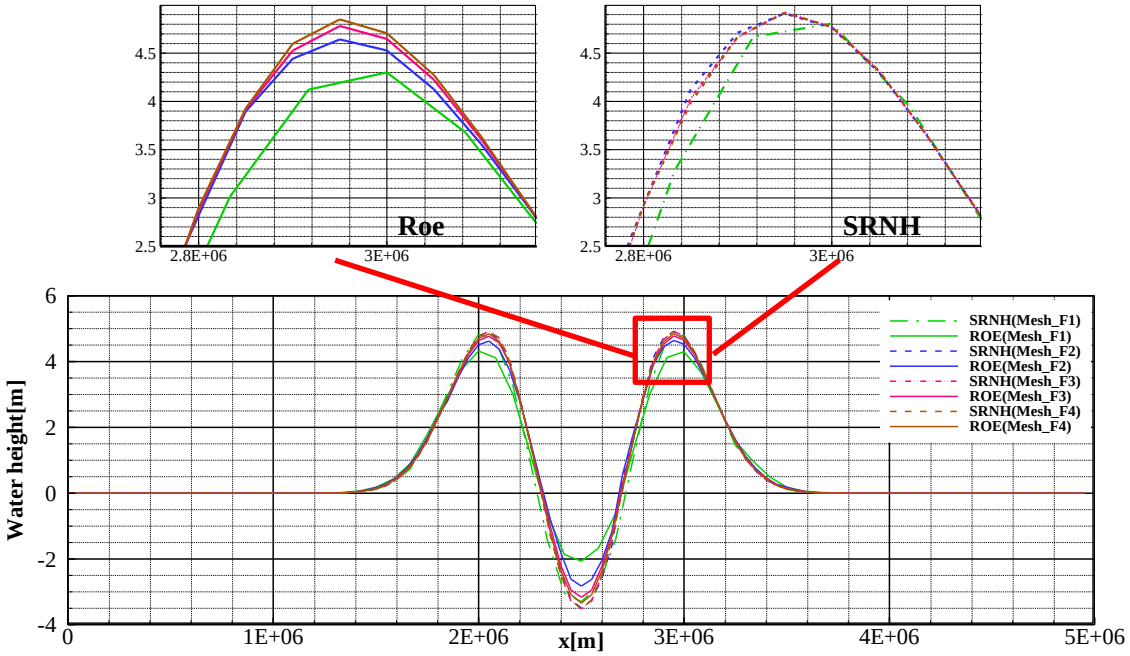


Fig2.8 Result of Roe and SRNH in *t*=30min along transverse middle-line (A-B)

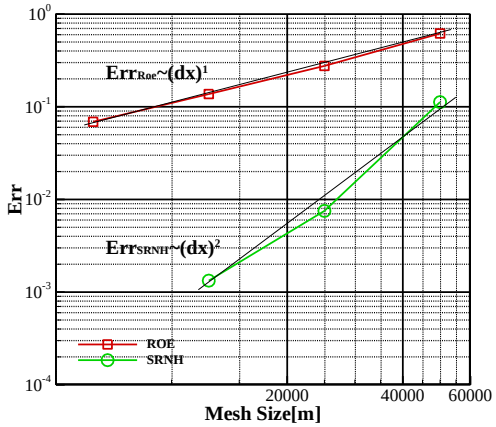


Fig2.9 Relative Error of the maximum wave height

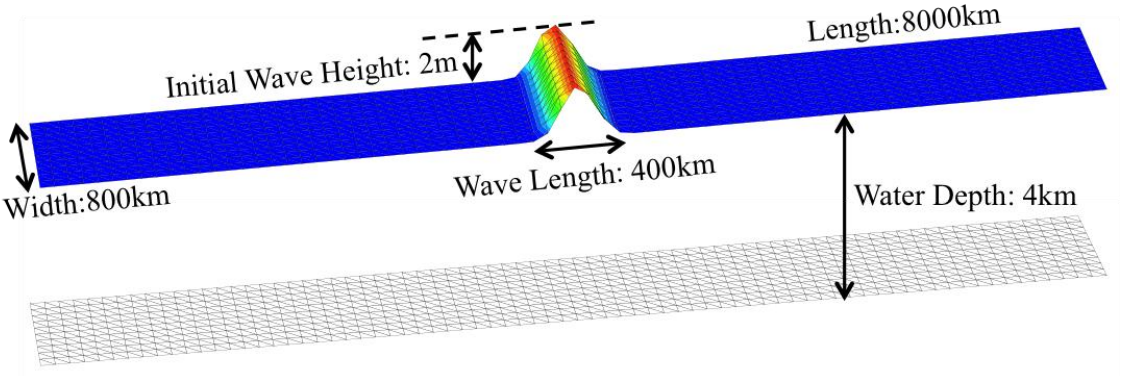


Fig2.10 Initial conditions for tsunami propagate in long narrow flat-bed channel

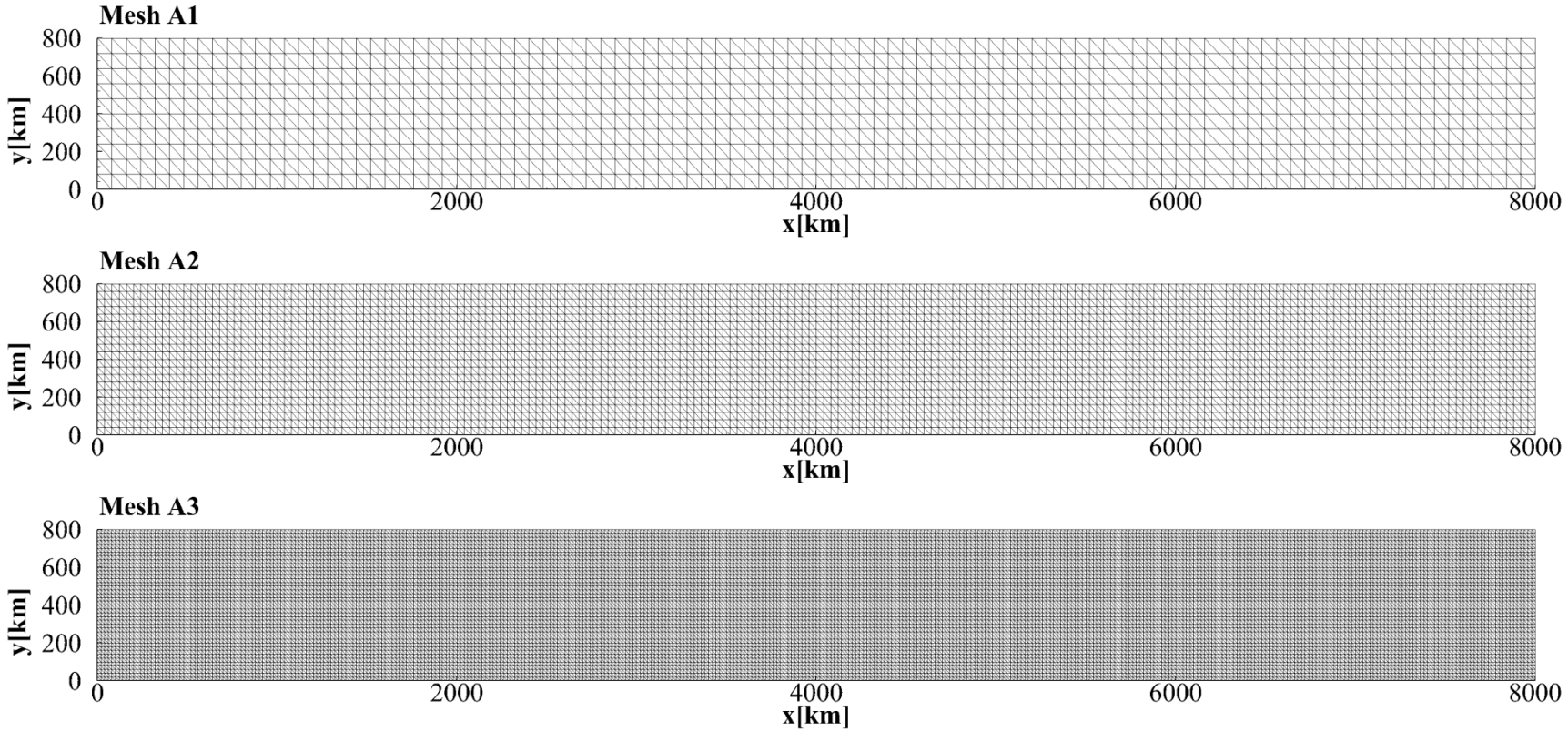


Fig2.11 Mesh used for tsunami propagate in long narrow flat-bed channel

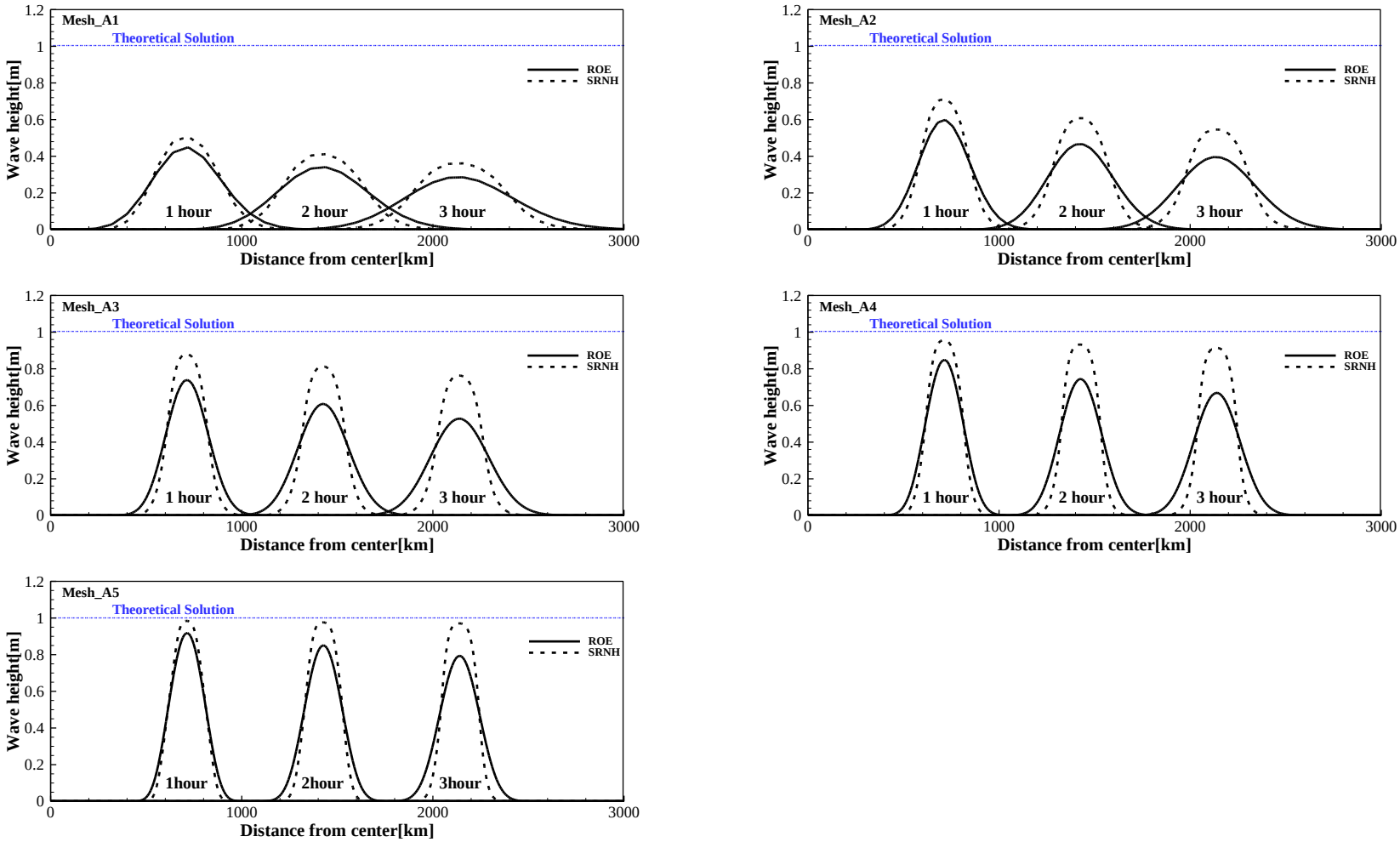


Fig2.12 Time series results of tsunami propagate (along transverse middle-line) on deep-ocean using different mesh

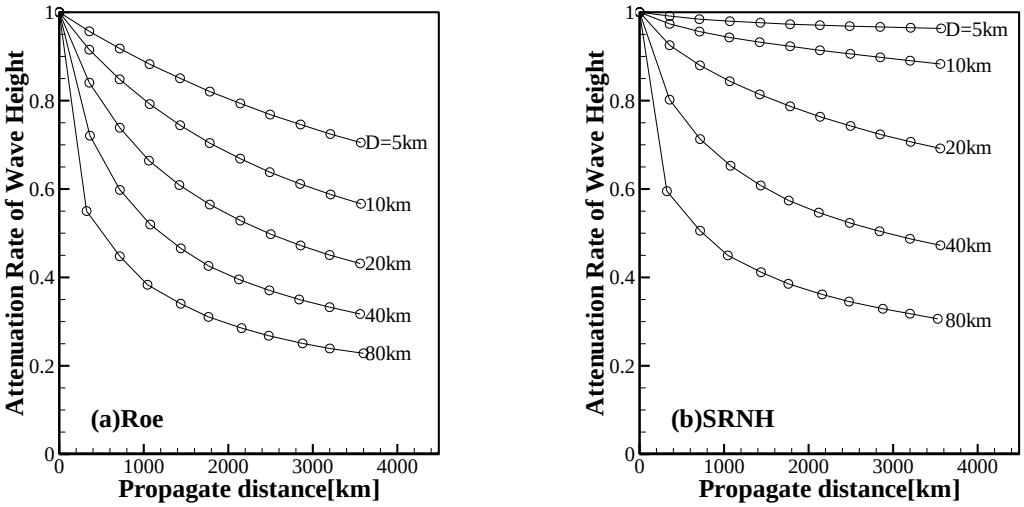


Fig2.13 Attenuation rate of computational wave using different meshes

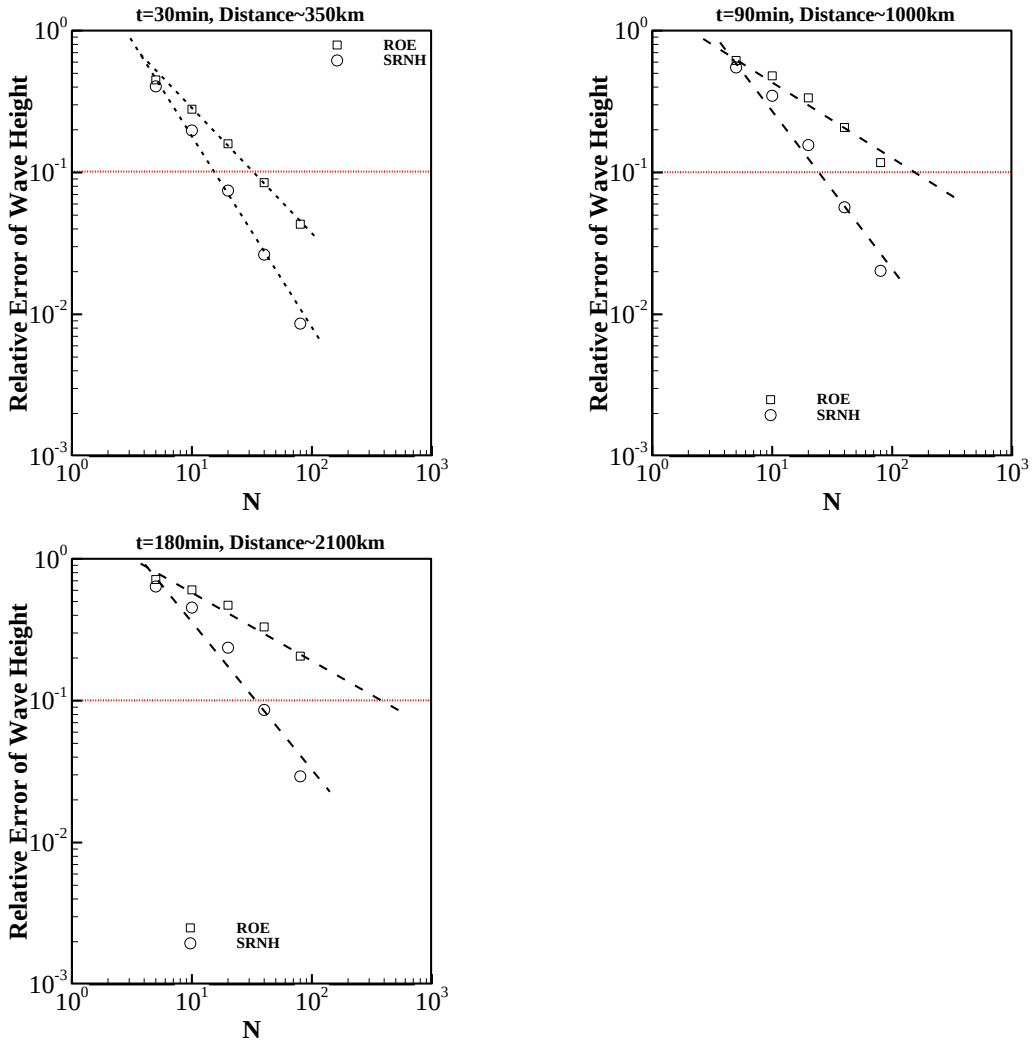


Fig2.14 Relationship of relative error and mesh number per wave length

Table2.2 Mesh used for tsunami propagate in long narrow flat-bed channel

Mesh	A1	A2	A3	A4	A5
Mesh number	2000	8000	32000	128000	512000
Mesh size [km]	80	40	20	10	5

Table2.3 Required mesh number "N_{req}" for tsunami propagate on deep-ocean (Err < 0.1)

Numerical scheme / Propagate distance	Roe	SRNH
350km	30	15
1000km	150	25
2100km	400	35

Table2.4 Mesh for run-up tsunami on a sloping beach

Mesh	R1	R2	R3	R4	R5
Total number	2000	8000	32000	128000	512000
Mesh size[km]	2	1	0.5	0.25	0.125

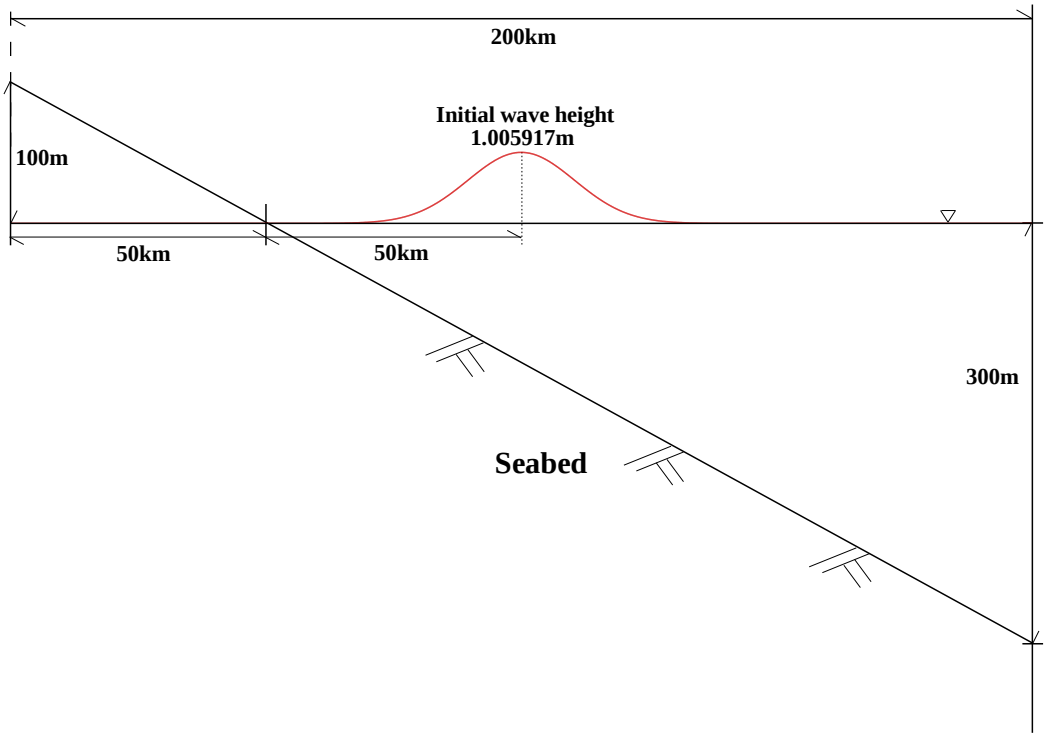


Fig2.15 Initial computational conditions of Run-up tsunami on slopping beach

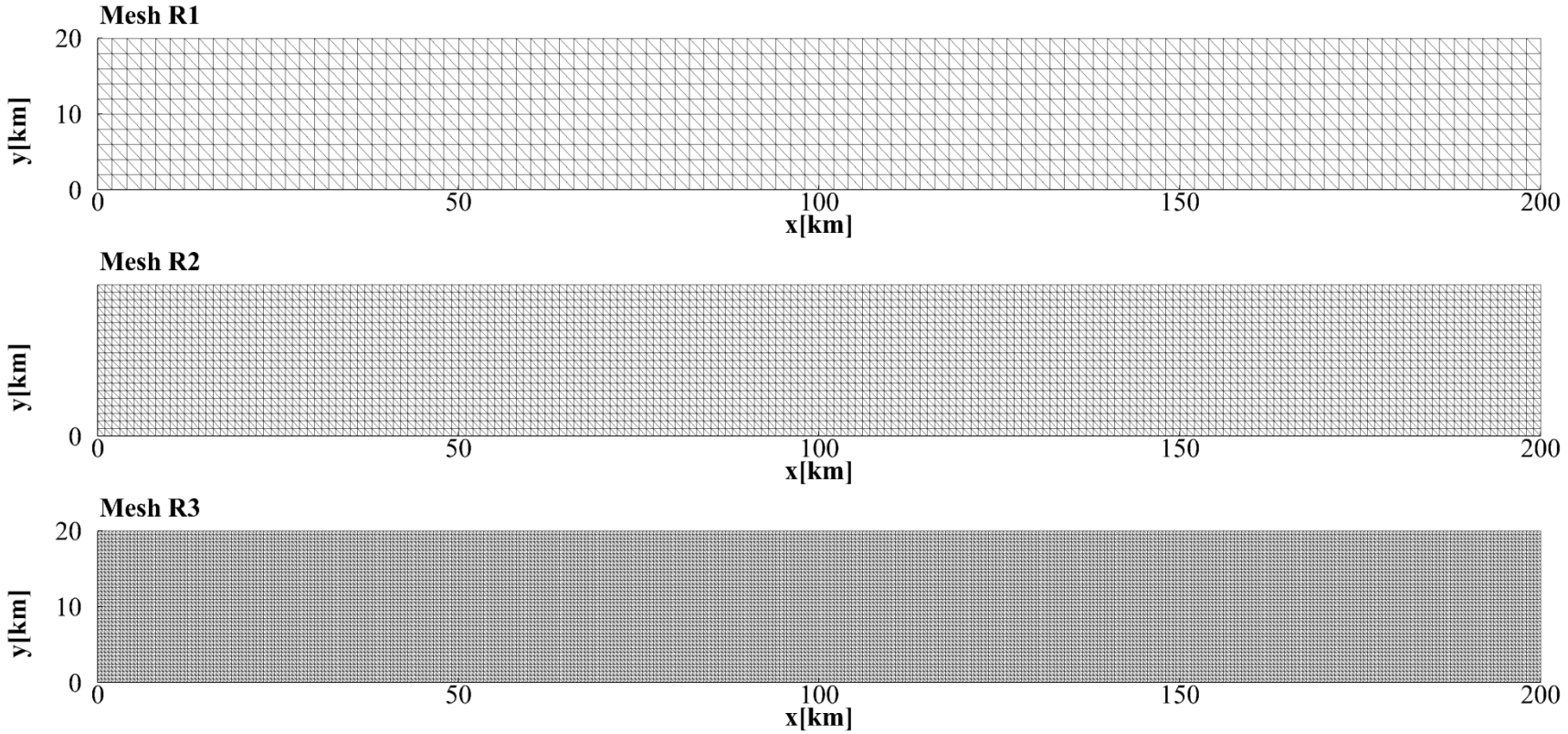


Fig2.16 Mesh used for Run-up tsunami on slopping beach

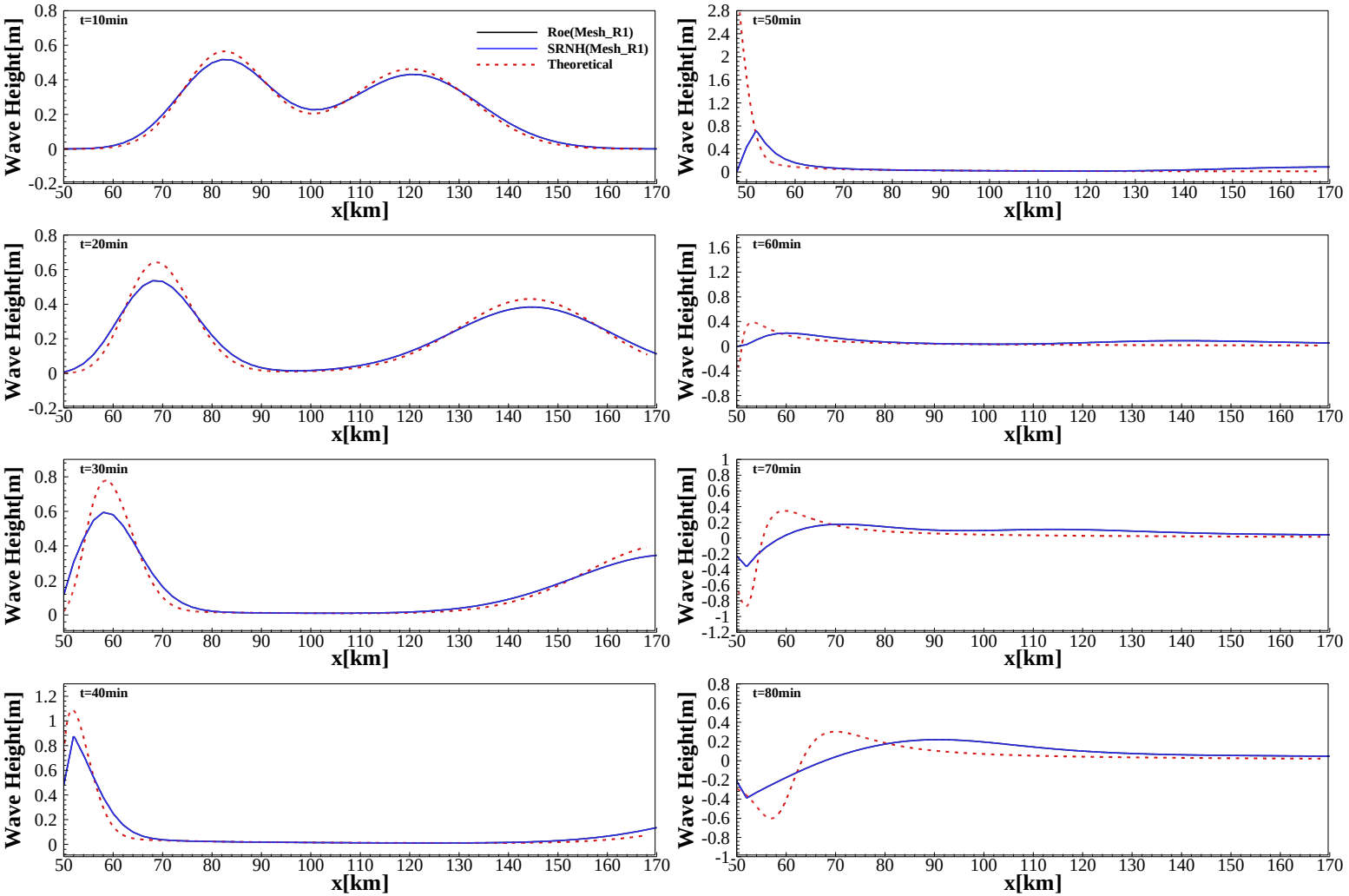


Fig2.17.1 Results of run-up tsunami on slopping beach using Mesh R1

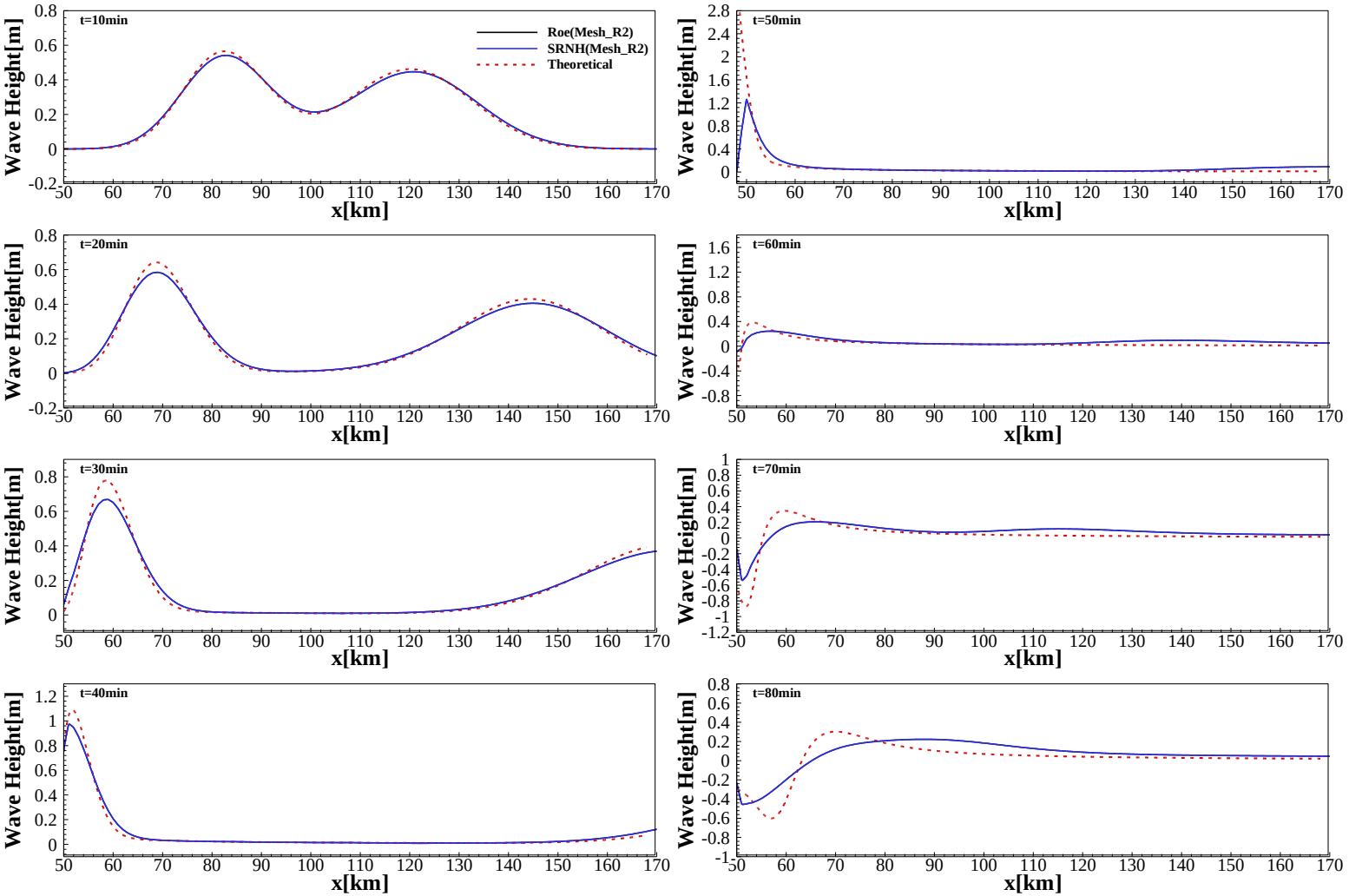


Fig2.17.2 Results of run-up tsunami on slopping beach using Mesh R2

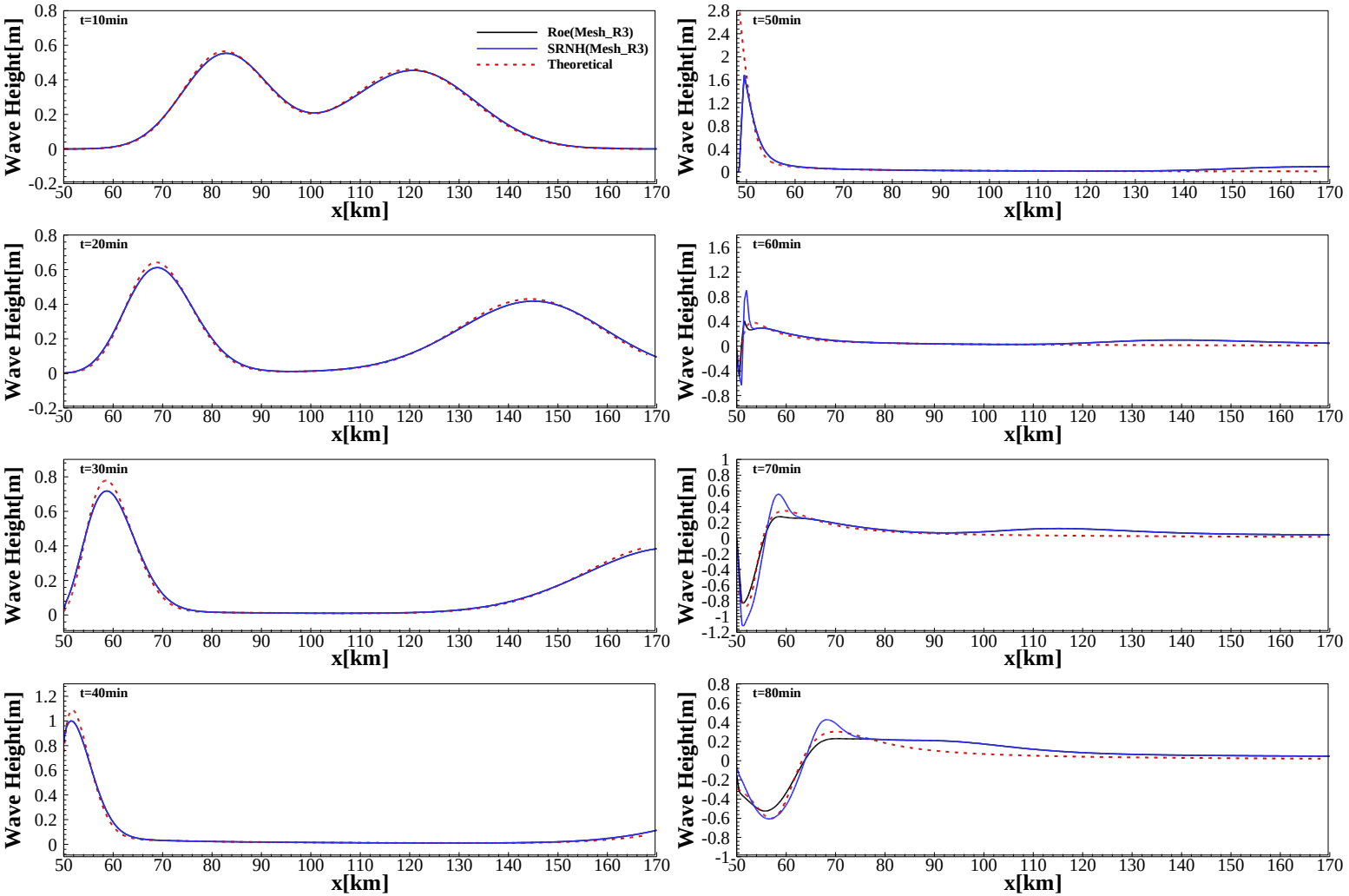


Fig2.17.3 Results of run-up tsunami on slopping beach using Mesh R3

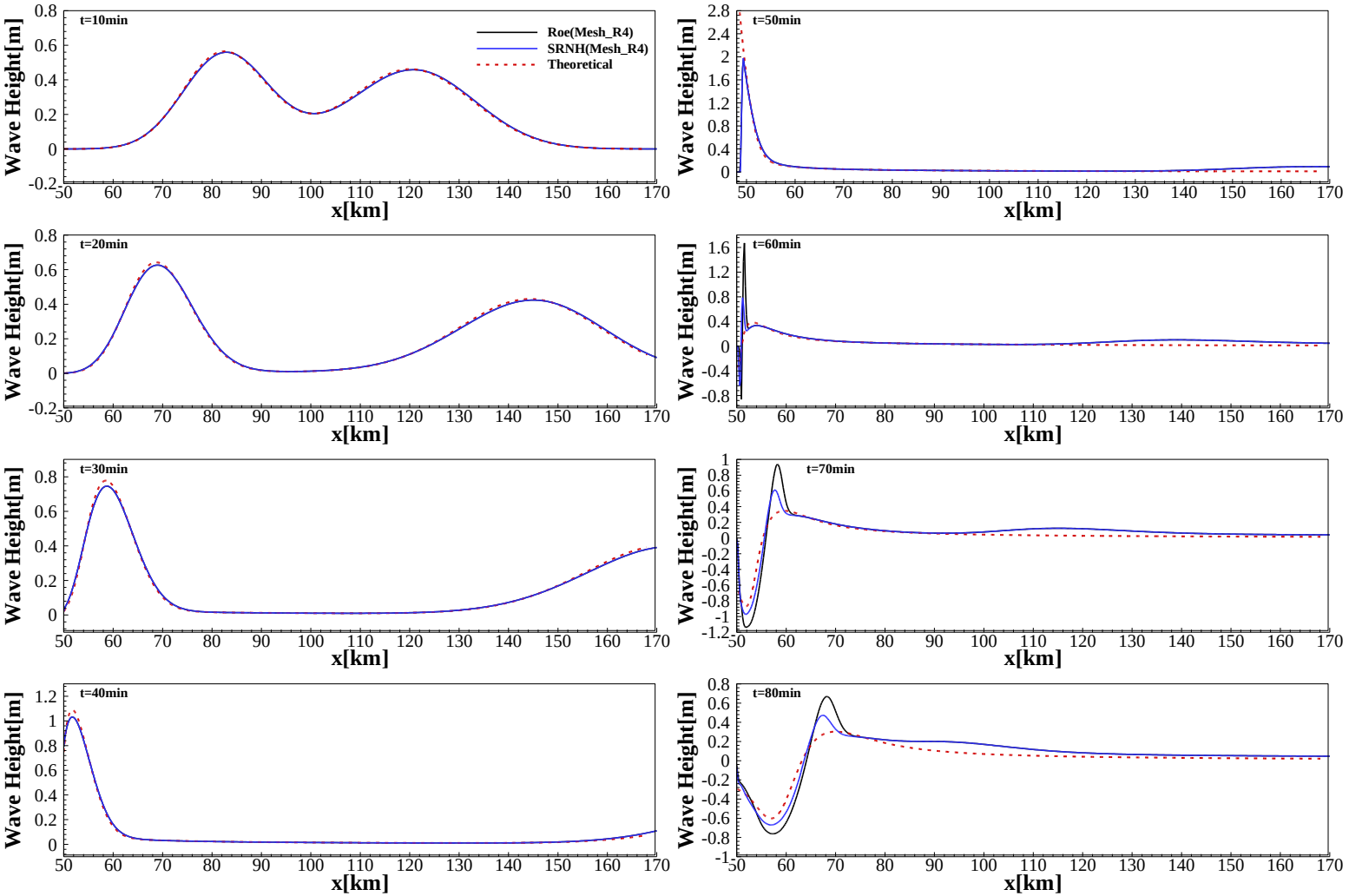


Fig2.17.4 Results of run-up tsunami on slopping beach using Mesh R4

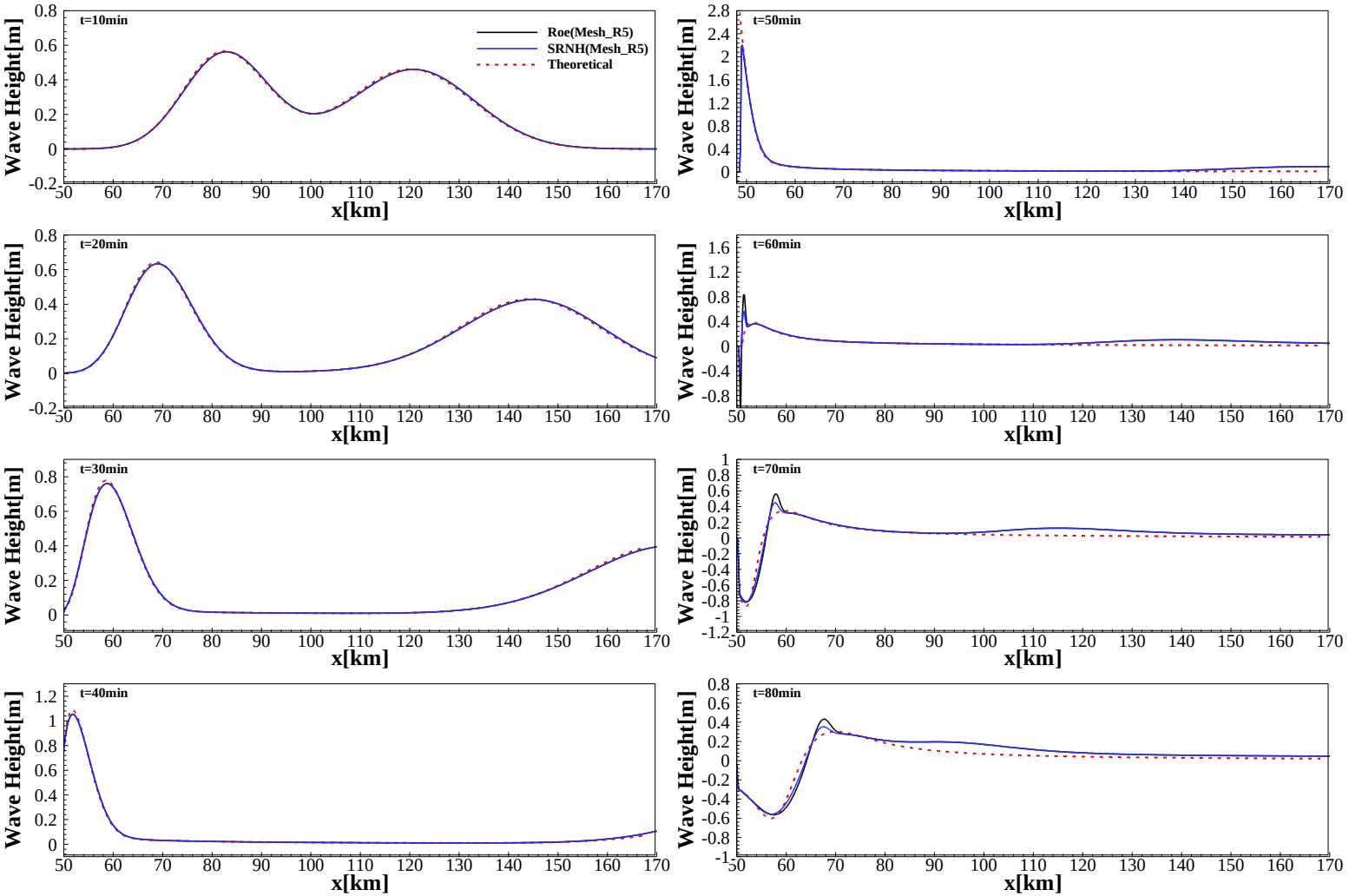


Fig2.17.5 Results of run-up tsunami on slopping beach using Mesh R5

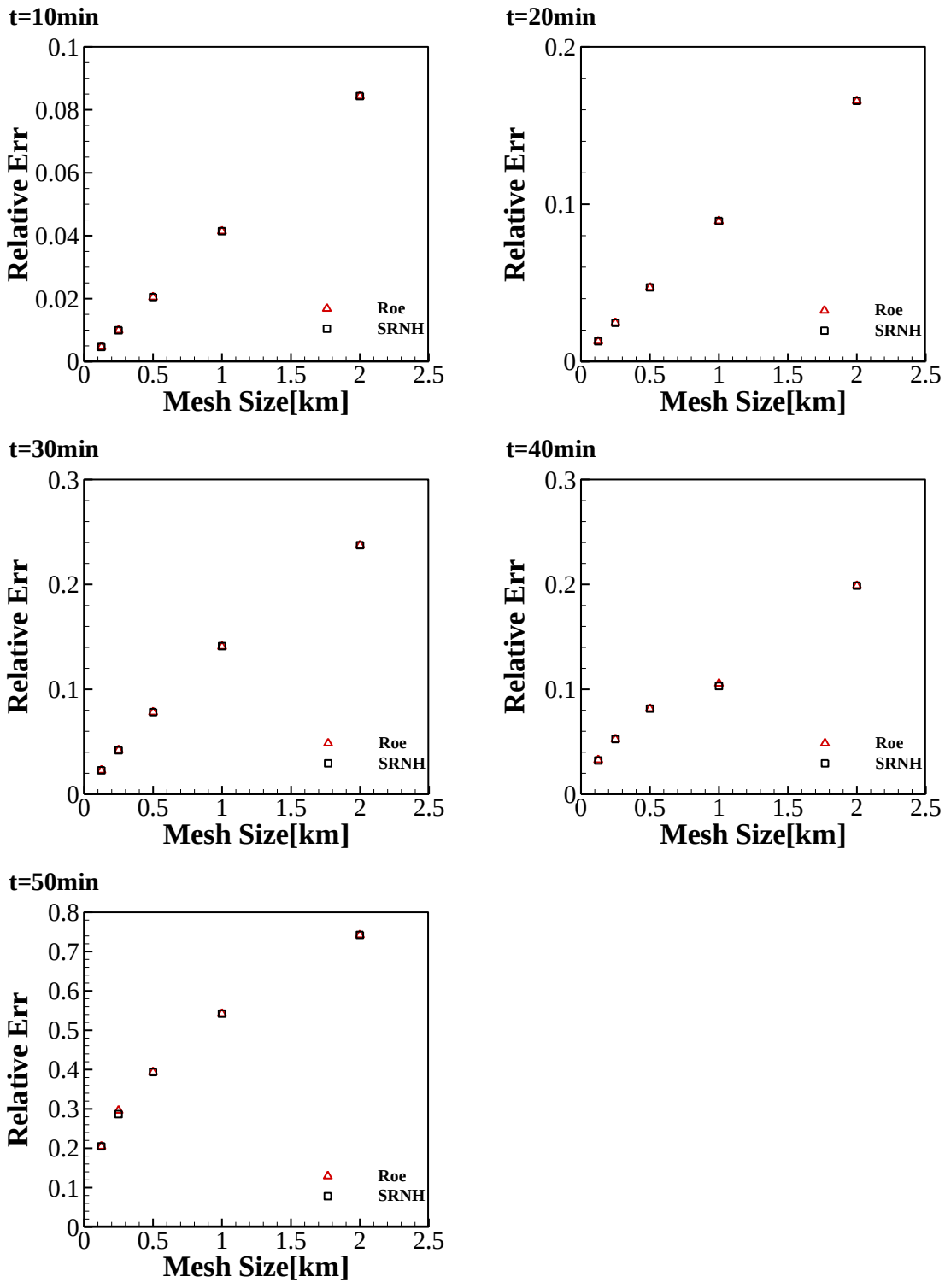


Fig2.18.1 Relative error according to mesh size (Run-up tsunami)

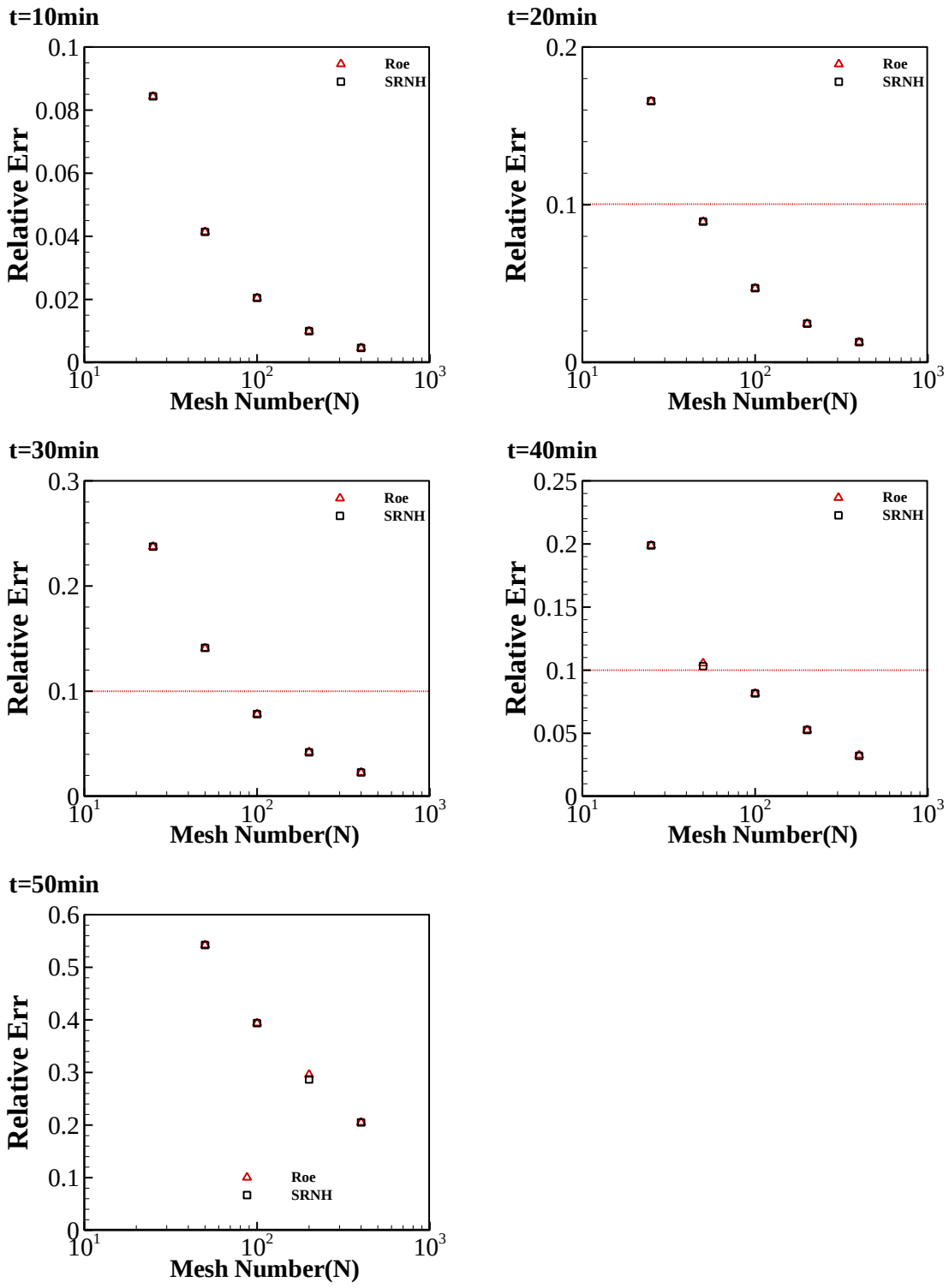


Fig2.18.2 Relative errors according to mesh number (Run-up tsunami)

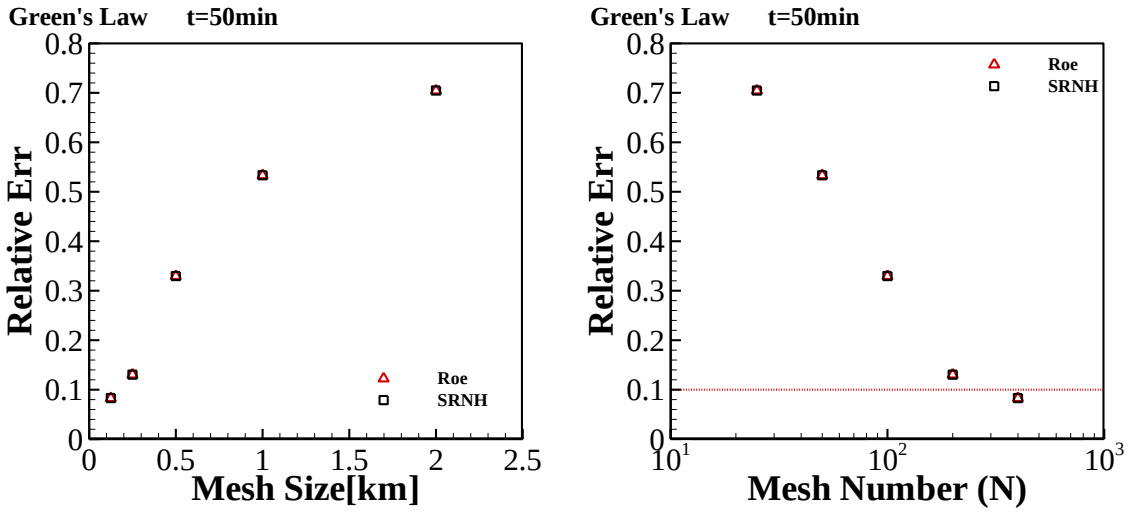


Fig2.19 Relative error of wave height estimated with Green's Law

Table2.5 Required mesh number "N_{req}" for run-up tsunami on sloping beach

Scheme / Time (minutes)	Roe	SRNH
10	20	20
20	40	40
30	70	70
40	50	50
50(Green's Law)	Non(200)	Non(200)

Table2.6 Mesh used for tsunami in 2011 Tōhoku earthquake

Mesh		T1	T3	T3R
Total number of mesh		34,060	266,194	296,552
Size of mesh [km]	Ocean	35~75	8~75	4~75
	Epicenter	14	6	6
	Near shore	5~18	2~12	2~3

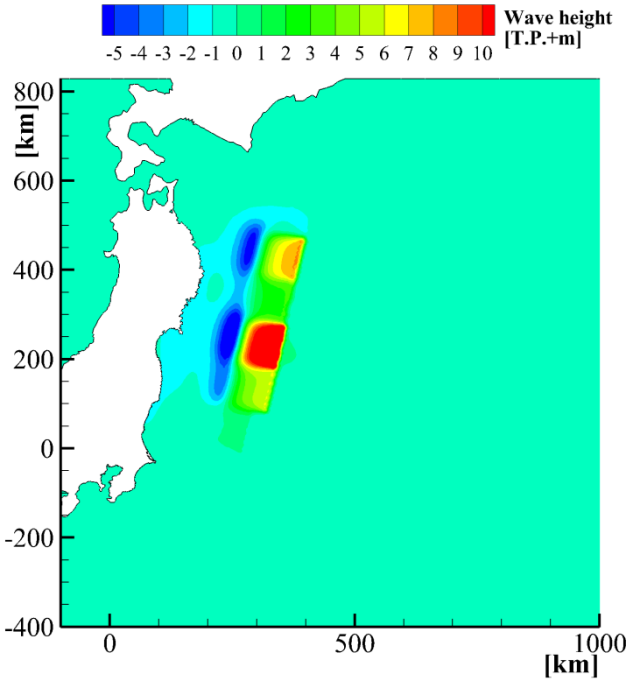


Fig2.20.1 Computational domain and initial wave height for tsunami in 2011 Tōhoku earthquake

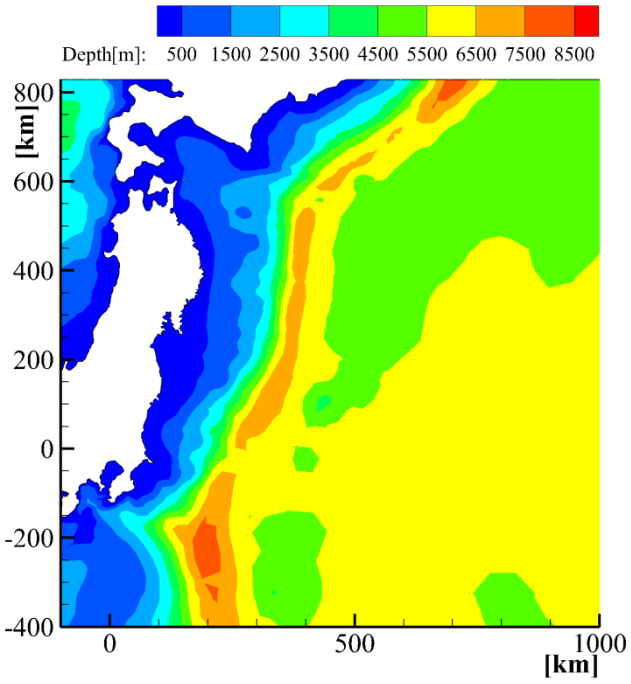
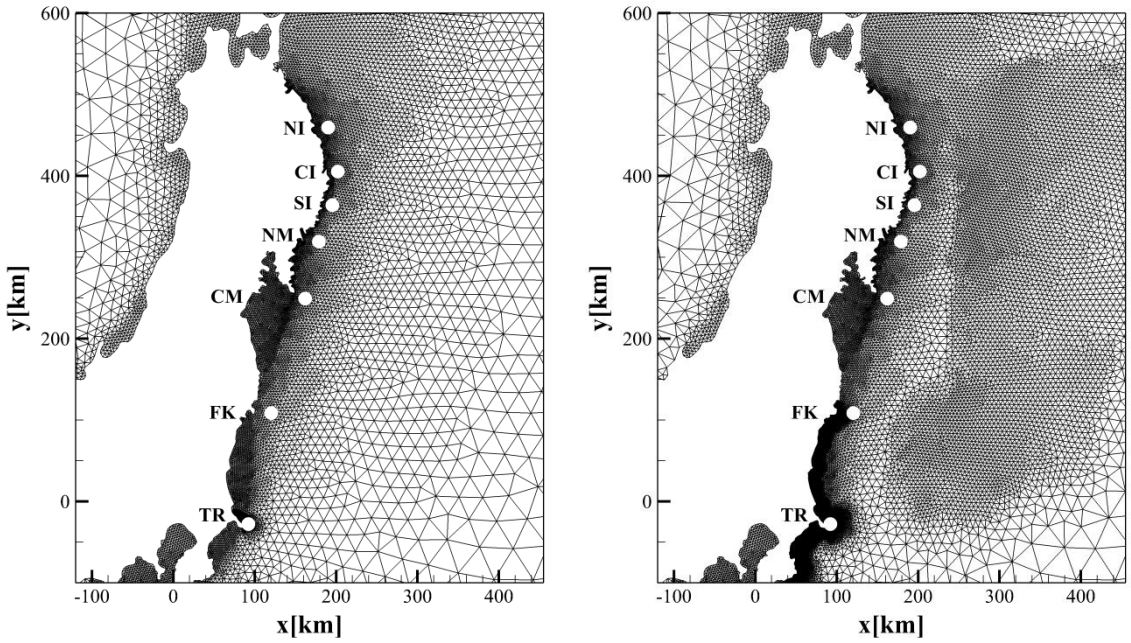


Fig2.20.2 Topography for simulation of tsunami in 2011 Tōhoku earthquake



a) Mesh T1 b) Mesh T3

Fig2.21.1 Computational mesh (T1 and T3) and location of GPS observational station

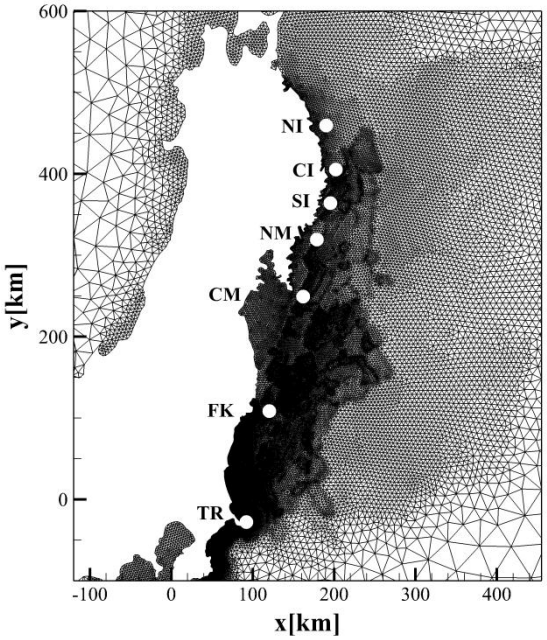


Fig2.21.2 Computational mesh T3R

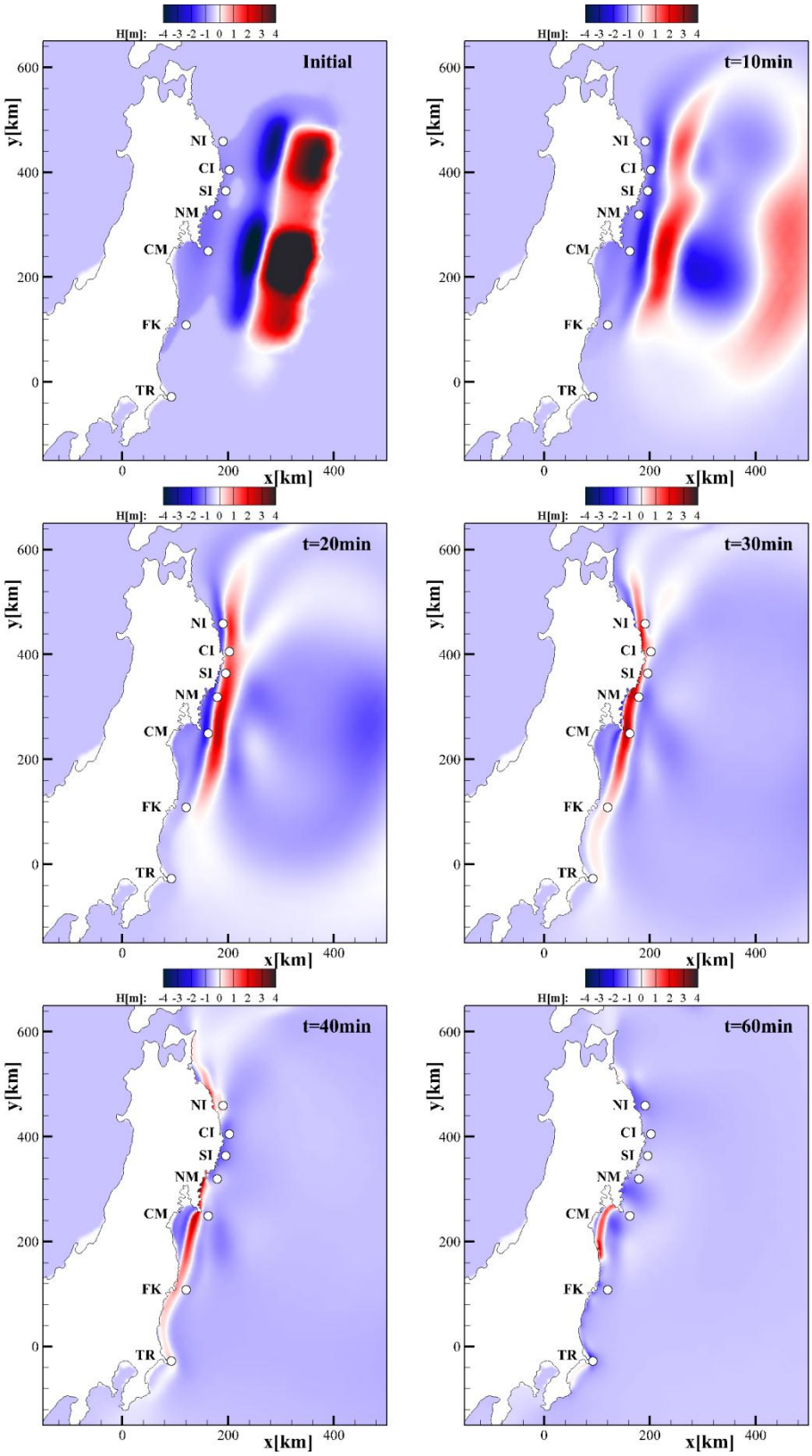


Fig2.22 Time series figure using Roe scheme with mesh T1

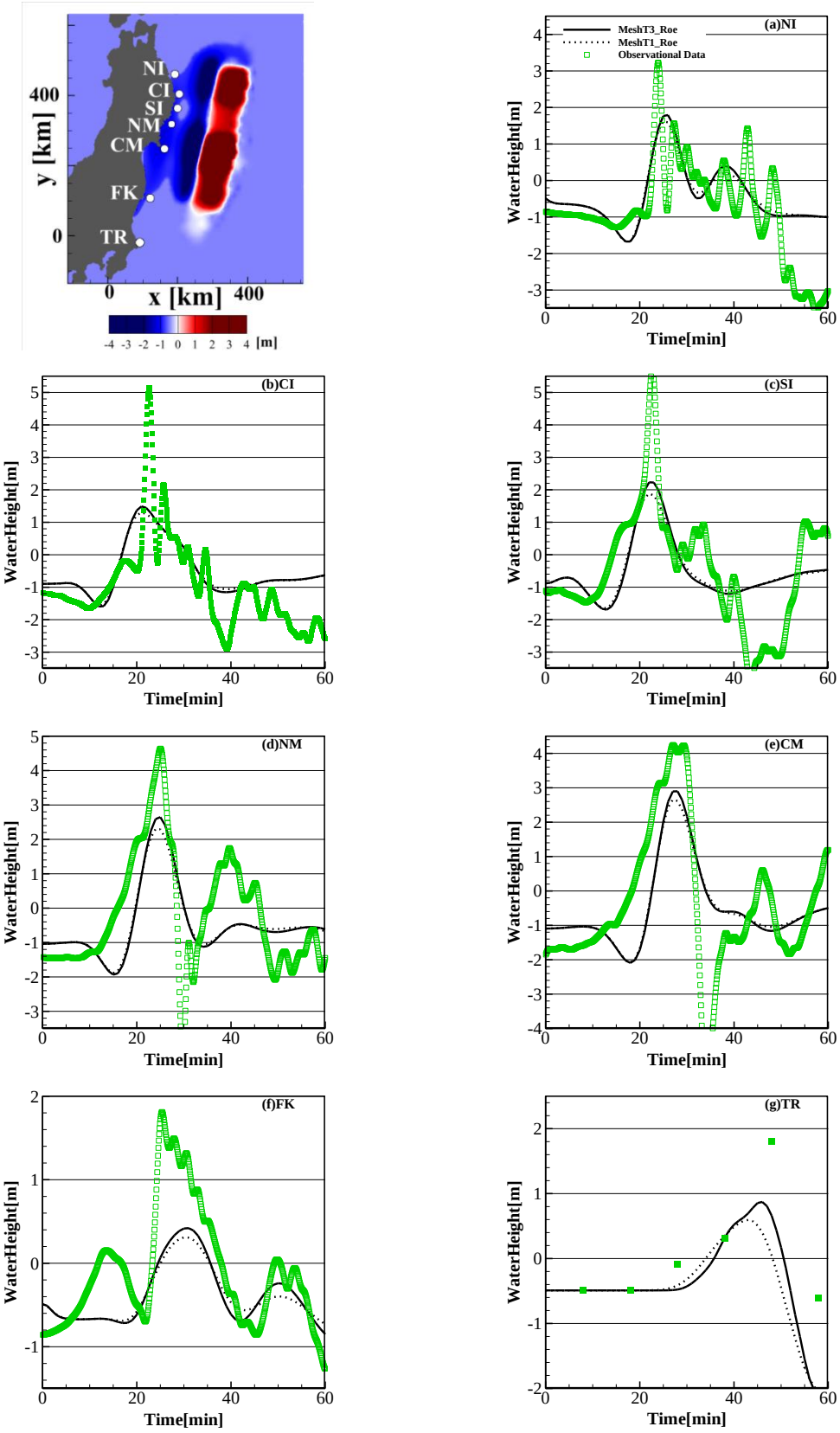


Fig2.23.1 Computational results of tsunami in 2011 Tōhoku earthquake using mesh T1 and T3 (Roe scheme)

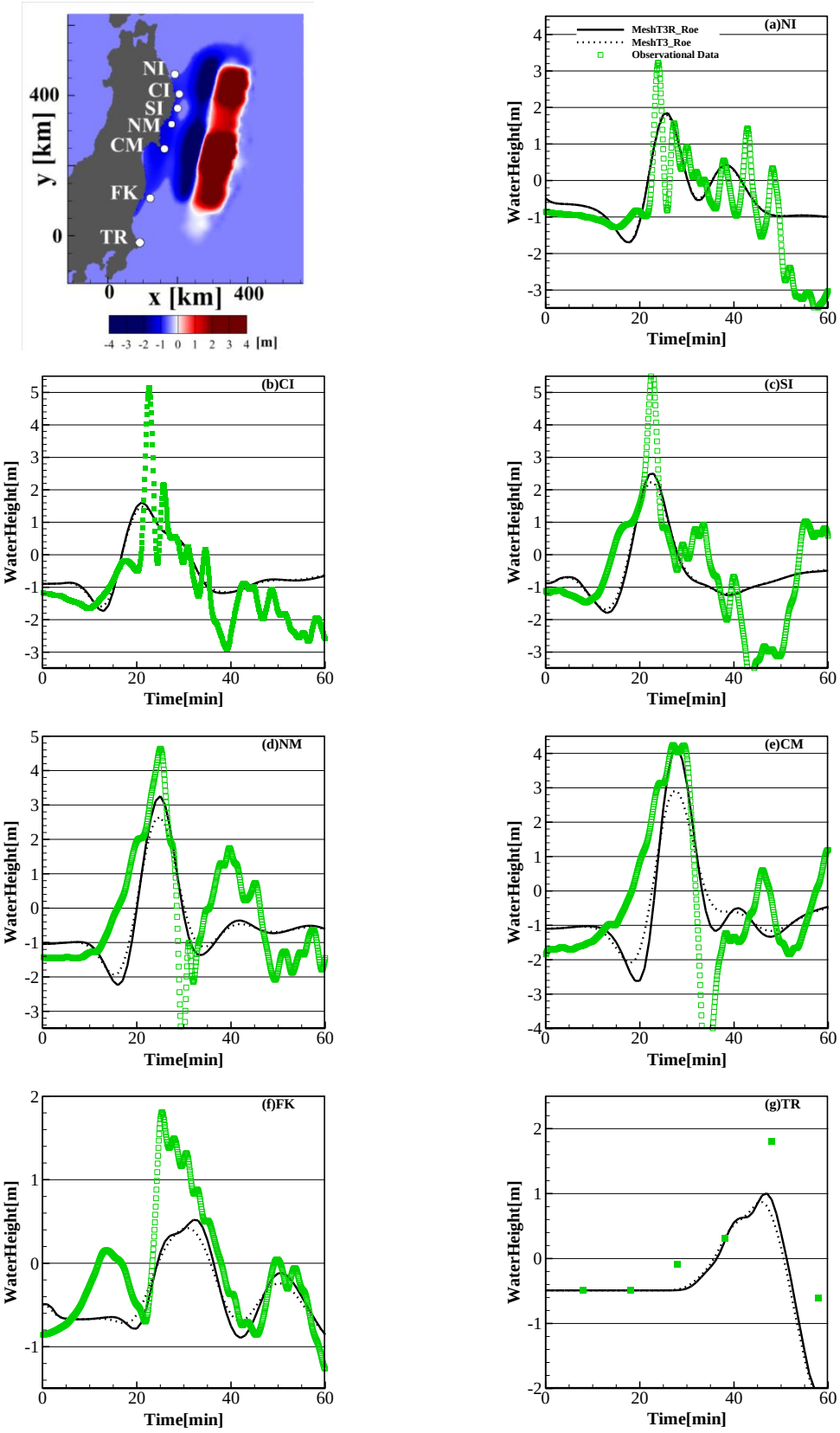


Fig2.23.2 Computational results of tsunami in 2011 Tōhoku earthquake using mesh T3 and T3R (Roe scheme)

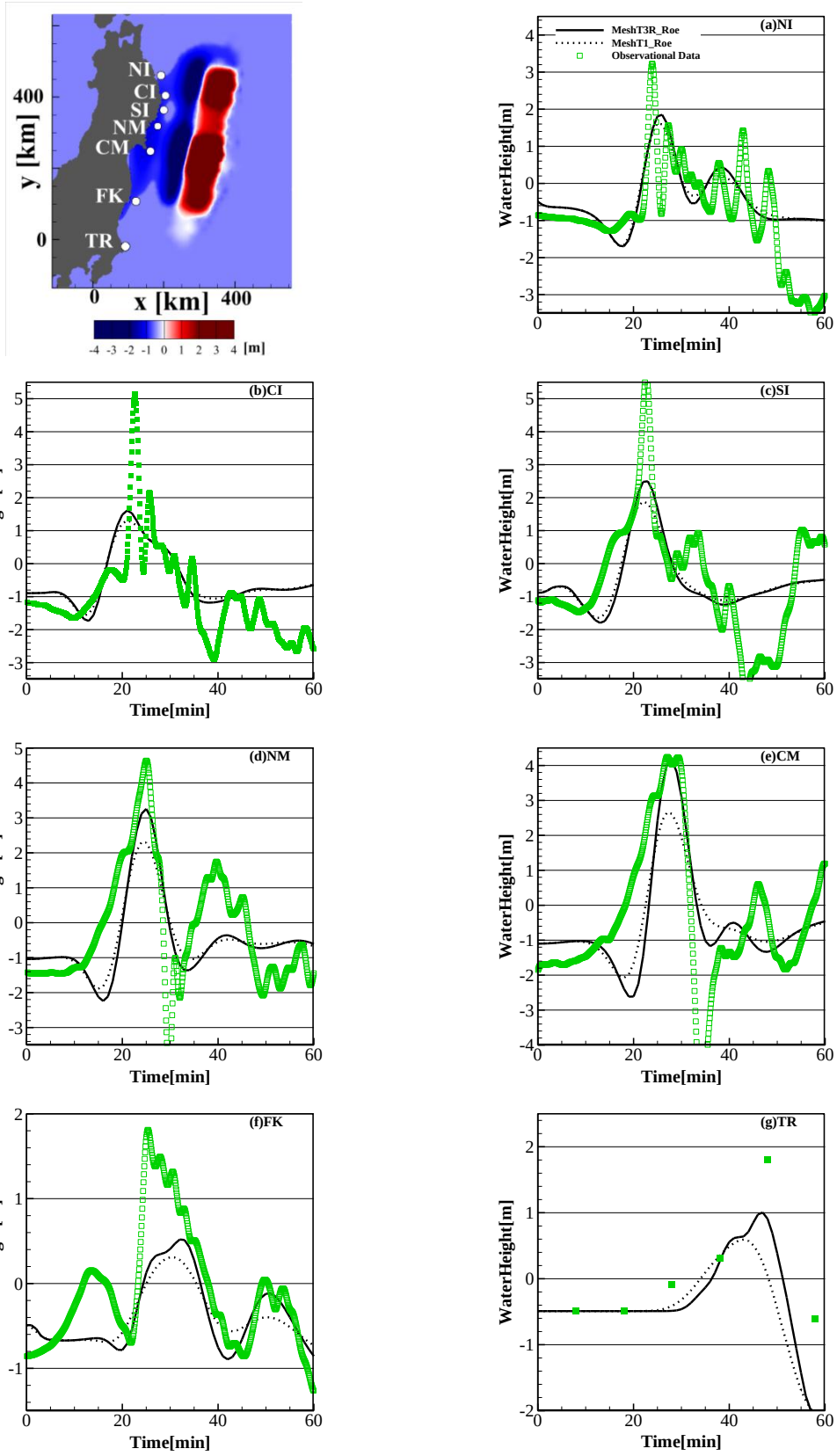


Fig2.23.3 Computational results of tsunami in 2011 Tōhoku earthquake using mesh T1 and T3R (Roe scheme)

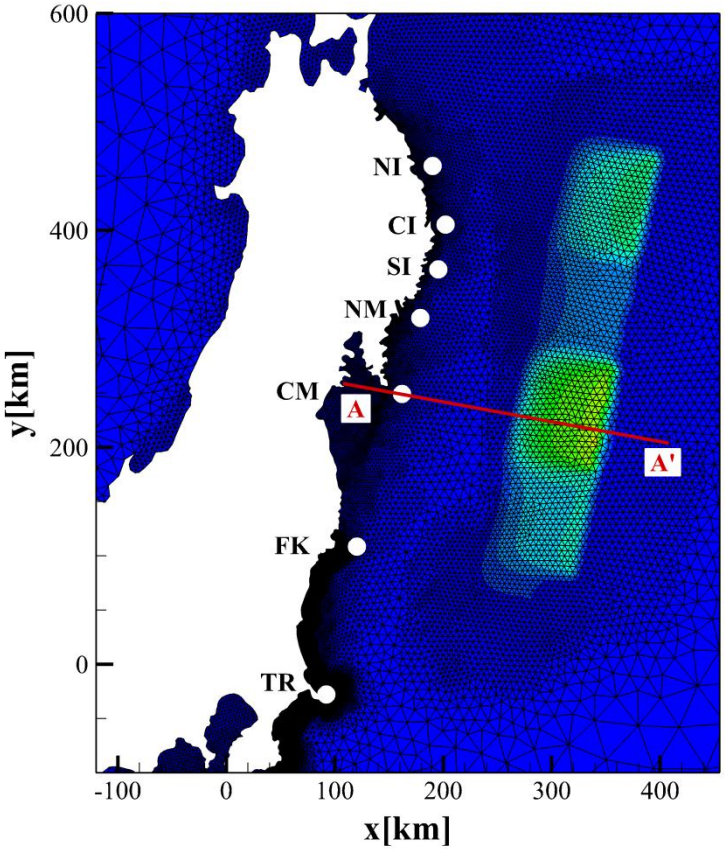


Fig2.24 Cross line (A-A') near CM observation station

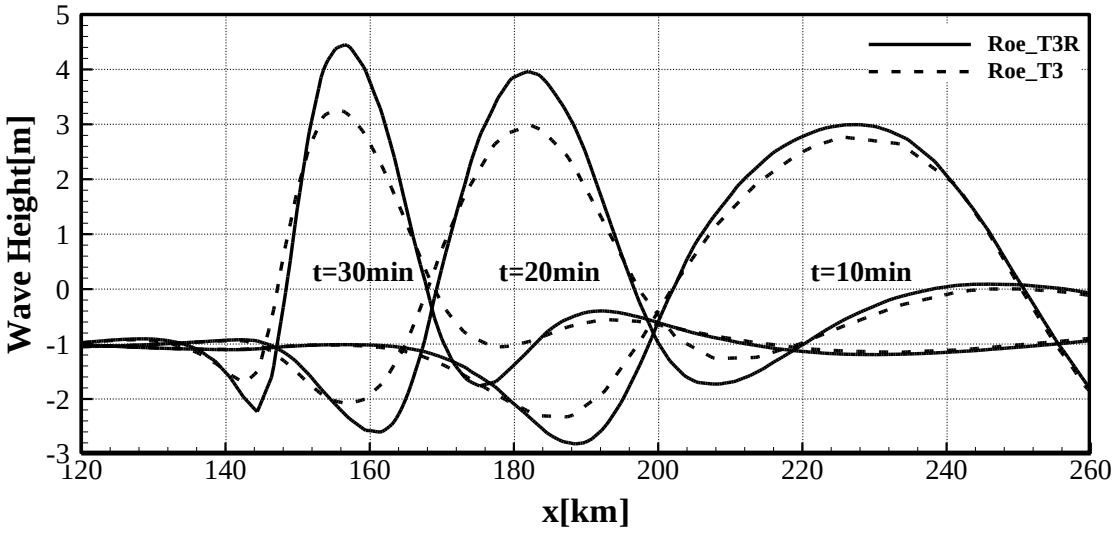


Fig2.25 Time variation of tsunami propagation along cross line A-A'

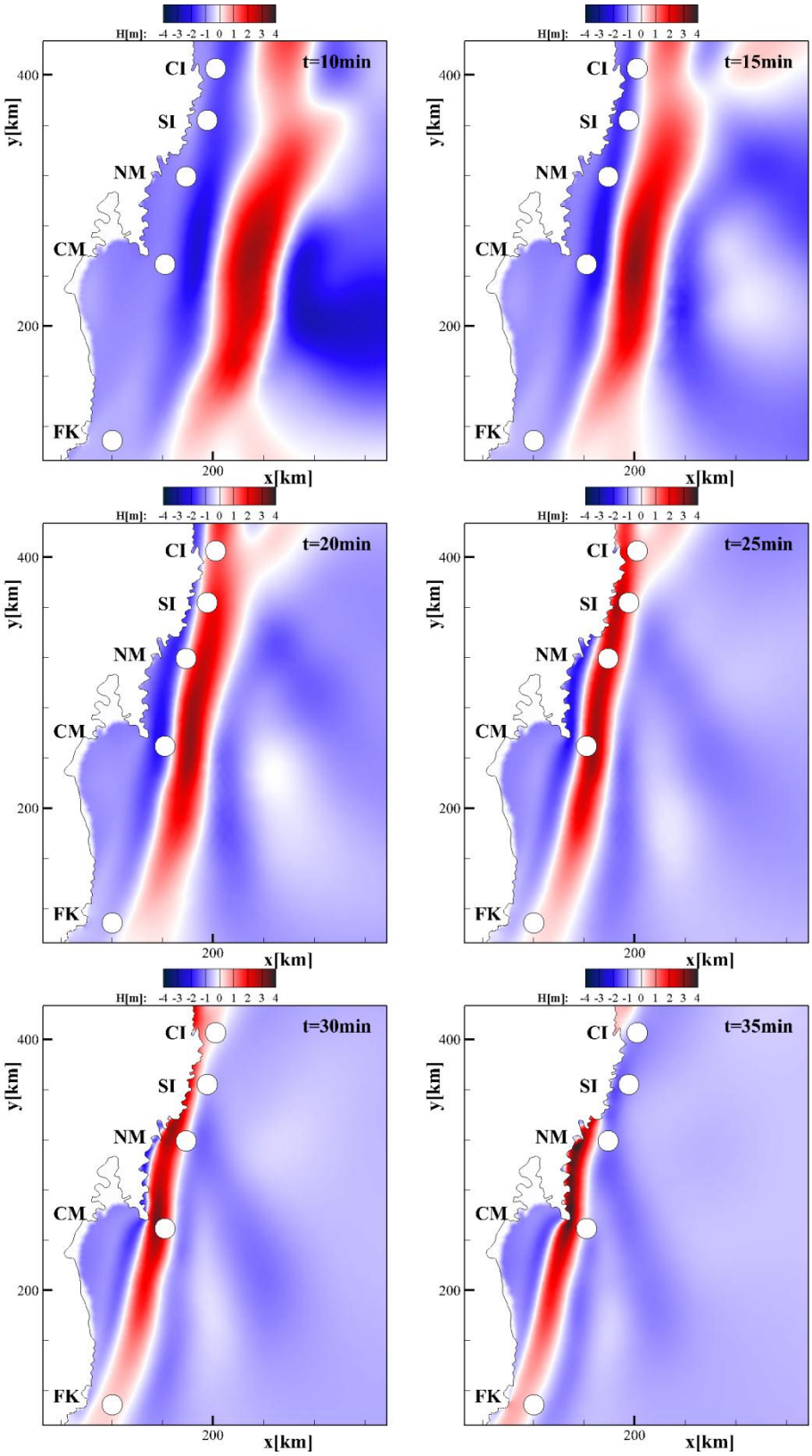


Fig2.26.1 Tsunami propagation near CM station (Roe scheme, mesh T3)

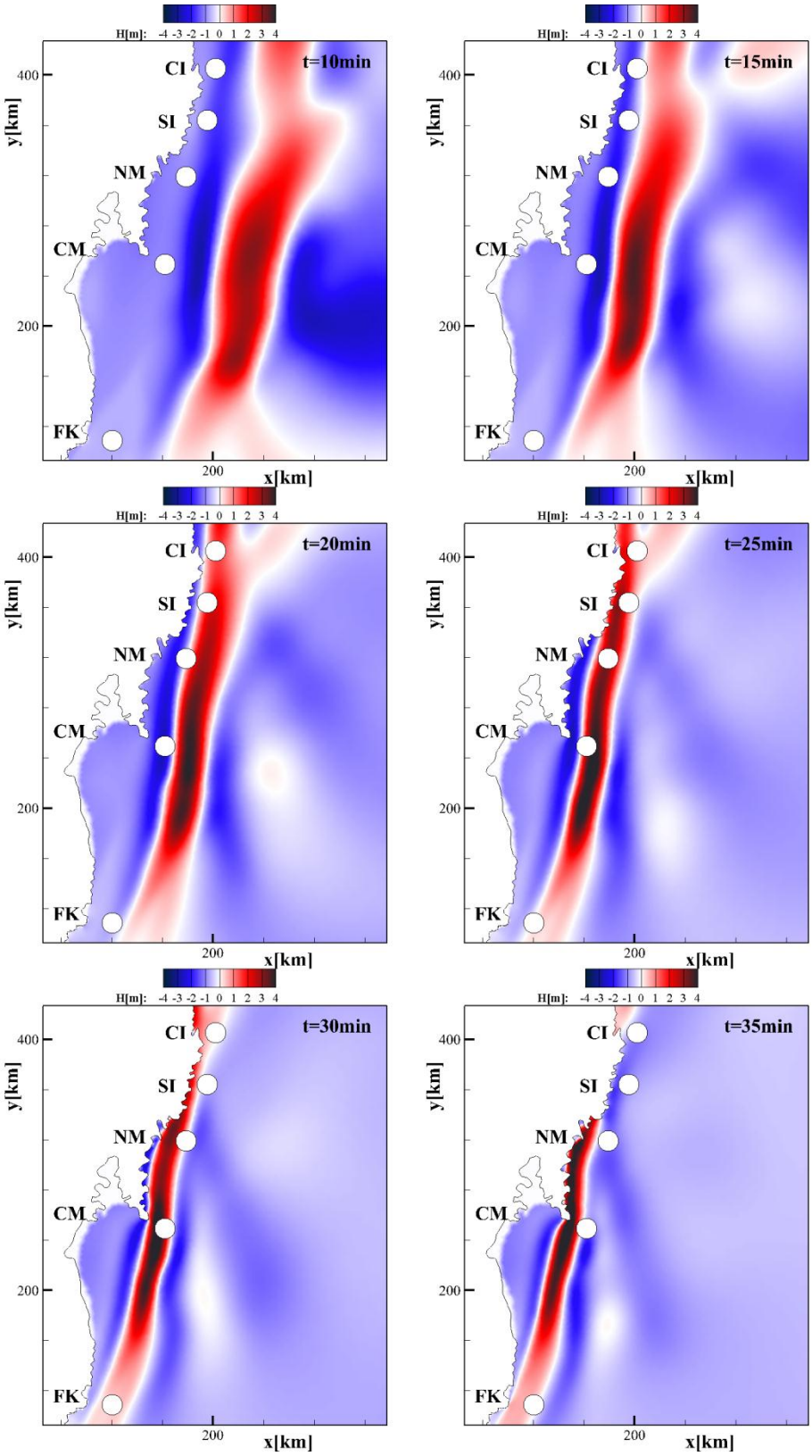


Fig2.26.2 Tsunami propagation near CM station (Roe scheme, mesh T3R)

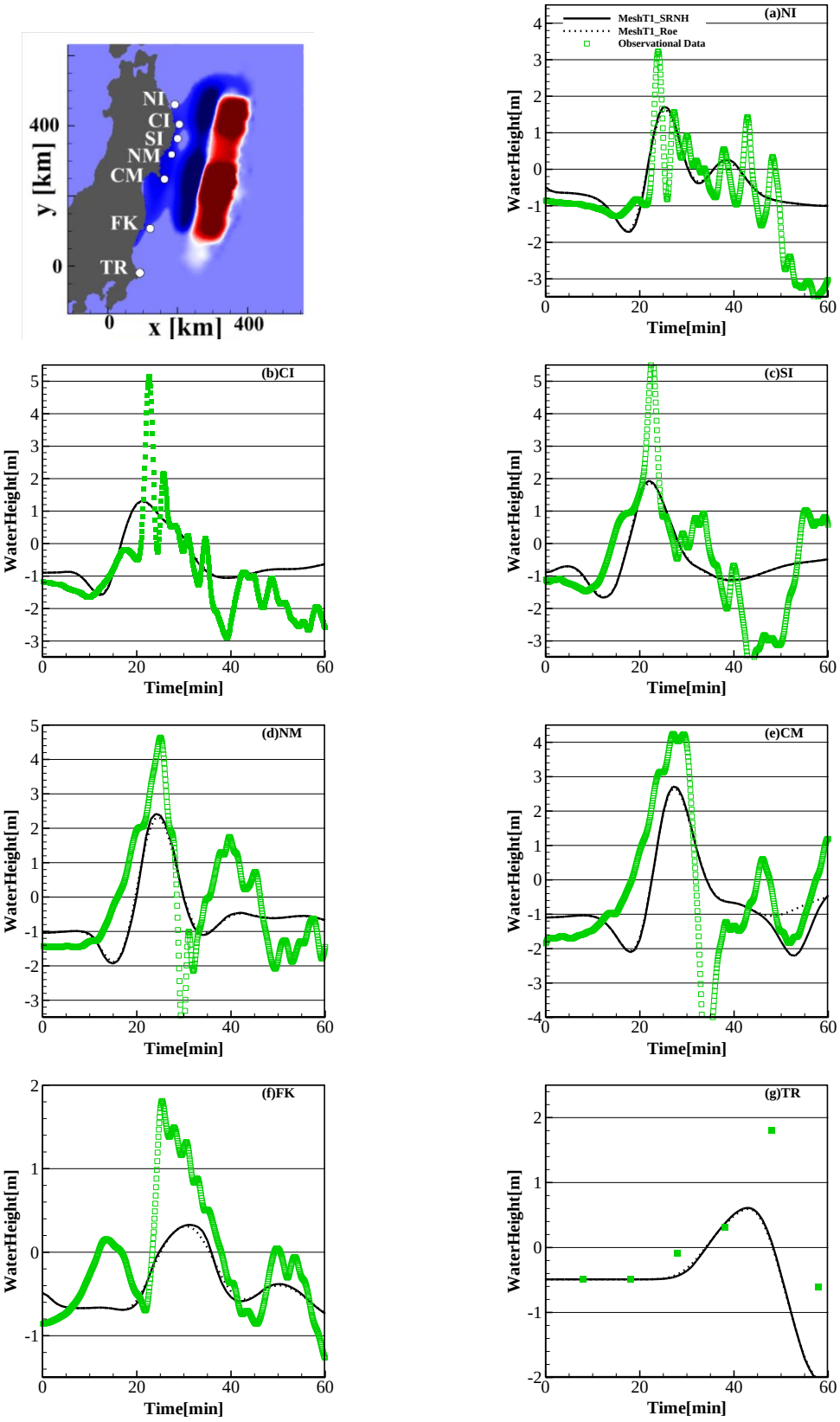


Fig2.27.1 Results of tsunami in 2011 Tōhoku earthquake using different schemes (mesh T1)

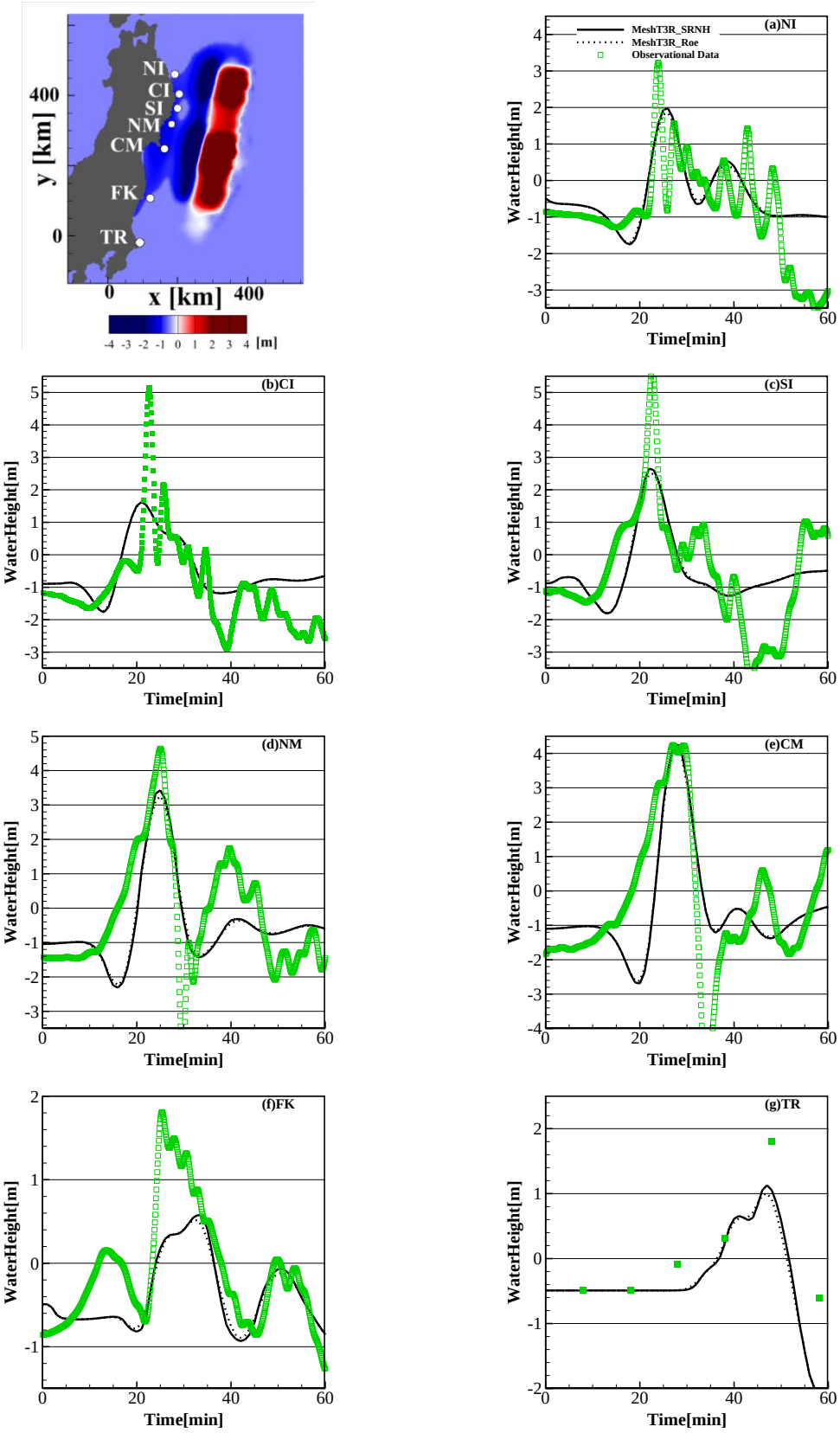
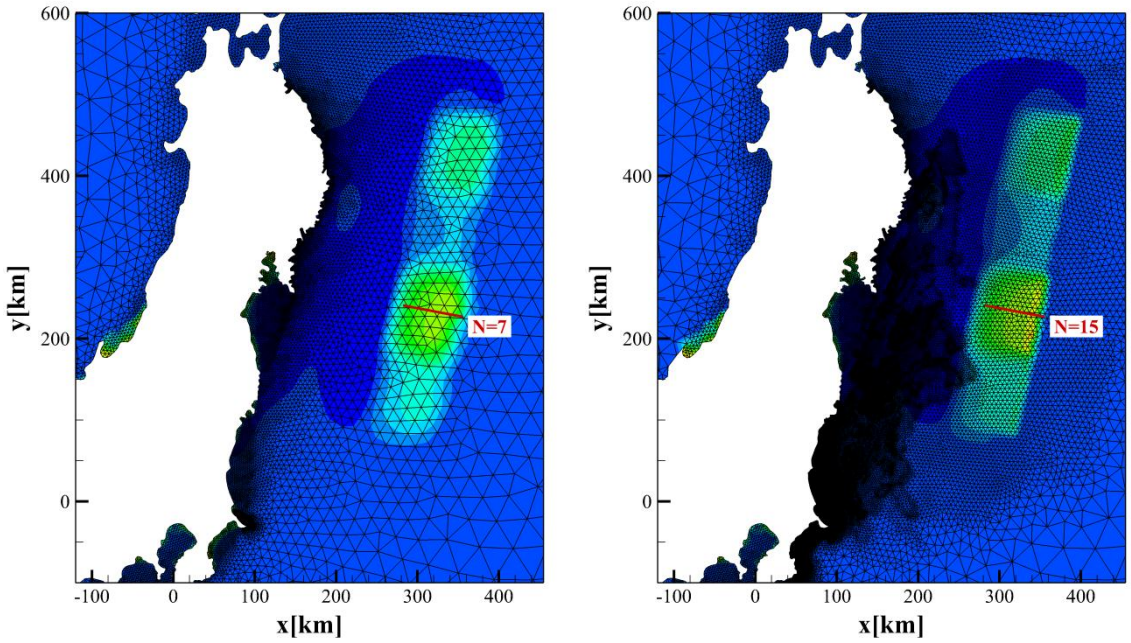


Fig2.27.2 Results of tsunami in 2011 Tōhoku earthquake using different schemes (mesh T3R)



a) Mesh T1 b) Mesh T3R

Fig2.28 "*N*" value for tsunami in 2011 Tōhoku earthquake

Chapter3 Rapid and high-resolution tsunami simulator by Adaptive Mesh Refinement (AMR)

3.1 Introduction

On account of the diverse spatial scales of tsunami movement, an additional difficulty arises for high-resolution tsunami simulation. More specifically, on the deep ocean, the wavelength of tsunami is about hundreds kilometers which is up to hundred times larger than that on near-shore region. But as discussed in Chapter2, finer mesh is required to get accurate tsunami simulation results. Here comes the dilemma, how can we get high-resolution results but with low computational cost?

The traditional way for solving the above problem is using fixed telescoping meshes at the near-shore area (George et al., 2006). For example, the famous tsunami simulator MOST model (Titov & Gonzalez, 1997; Titov & Synolakis, 1998; Titov et al., 2005) is using fixed telescoping meshes to reduce computational load. However, this approach has some drawbacks:

a. Some priori knowledge is required to determine the local mesh should be refined to what extent.
b. The potential of mesh refinement in reducing computational load is not fully exploited. In other words, the refined mesh is locked at some places through whole computation no matter whether there are waves.
c. The fixed mesh refinement just for coastline will also affect the tsunami simulation results. The results comparison between mesh T3 and T3R (Fig2.23.2) in Chapter2 is a strong evidence that the mesh on the tsunami propagate route will also affect the simulation results.

Adaptive mesh refinement (AMR) is a method that can be used to refine the mesh locally in regions of greater interest during every time step or selected time steps of

computation. Simultaneously, the rest meshes will retain the same size or be coarsened back. The AMR is an adaptive method which is effective in capturing wave movement to surpass numerical diffusion error, as well as reducing computational load. In consideration of simpleness, most of the AMR application on tsunami simulation is based on structured mesh system. Thereupon, in this research, a new strategy (include the strategy and criterion of mesh refinement and coarsening) of applying AMR on unstructured mesh system based on finite volume method (FVM) is proposed to achieve rapid and high-resolution tsunami simulation. The effectiveness of the developed tsunami simulator is also investigated.

3.2 Methodology

3.2.1 Strategies of refining and coarsening

As mentioned in Chapter2, the basic idea of finite volume method (FVM) is that the numerical flux is calculating through surfaces. Whereas, one surface should just only connecting 2 meshes is required by the FVM method. In consideration of this restriction, the application of adaptive mesh refinement (AMR) on FVM is much more complicate compared with the application on finite difference method (FDM).

Bisection-type mesh refinement is widely used for triangular mesh refinement (Bank et al., 1983). The basic concept of this refinement method is shown in Fig3.1.1. The middle points (red circle in Fig3.1.1.a) of the three surfaces are used as the vertex of new refined mesh. By connecting these three middle points, the original mesh is divided into four sub-meshes with half spatial size. However, there is a problem that the highlight red surface (Fig3.1.1.b) is shared by three meshes which is not allowed by the finite volume method (FVM). Therefore, some special treatment for the neighbor mesh is needed to guarantee that one surface is just shared by two meshes rigorously, which will cause the refinement strategy become complicate to apply.

In view of above problem, a new strategy of local mesh refining and coarsening is proposed to combine the adaptive mesh refinement (AMR) with unstructured mesh

system. The proposed approach is not only simple but also efficient that there is no need to treat neighbor mesh additionally.

Fig3.1.2 shows the concept of the proposed new approach. The refining degree of mesh is indexed by “Level”, in which “Level 1” refers to the original state of mesh. The refinement route consists of three steps:

a. Use the refining criterion (see next section for detail) to search and label the “Level 1” meshes that are needed for refining. For the convenience of instruction, the selected meshes are painted to blue just as shown in Fig3.1.2.a.
b. Take the barycenter of the selected meshes as the apex of the new triangles, and divide one “Level 1” mesh into three “Level 2” meshes by connecting the barycenter to the three apex of “Level 1” mesh. However, as seen in Fig3.1.2.b, the longest surface of “Level 2” is the same as “Level 1” mesh. Therefore, the directly usage of “Level 2” meshes can’t be expected to improve the resolution of tsunami simulation.
c. The couple of “Level 2” mesh (see highlighted two red meshes in Fig3.1.2.b) which is connected by long surface is partitioned again to get the “Level 3” meshes (see highlighted red meshes in Fig3.1.2.c).

By using the above approach, the local mesh refinement can be achieved effectively as well as automatically guarantee that one surface just connecting two meshes. If we using $D=\sqrt{2\times\Delta s}$ in representing the mesh size (Δs is the area of the triangular mesh), it will be minimized to $1/\sqrt{3}$ times of the “Level 1” mesh size by using above refinement (from “Level 1” to “Level 3”) only once. In addition, if you apply the above refinement to “Level 3” mesh again, the mesh will be refined to 1/3 times (“Level 5”) of the original mesh size. By applying the above approach recursively, the mesh can be refined to any size you want. Furthermore, in order to avoid unnecessary waste of computing resources, the refined mesh could also be coarse back to bigger size or even original size, in case of the mesh is too small for current computation. It should be noted that, considering the sharp size change will cause additional numerical error, the

partition and combining process are limited to odd “Level”. For example, as shown in Fig3.1.2.c, the original “Level 1” meshes are adjacent to “Level 2” meshes, and “Level 2” meshes are adjacent to “Level 3” meshes, which insured that every mesh is adjacent to mesh that is just one “Level” bigger or smaller than itself.

3.2.2 Criterions of refining and coarsening

There are varieties of criterions can be used for labeling the meshes that are necessary to be refined, in which the error estimation procedure is widely used for tsunami simulation (Behrens & Bader, 2009, Popinet, 2012). The basic idea of this method is to achieve a uniform distribution of local error. However, this method is weak in treat complex topography.

In this research, a new criterion is proposed to determine which mesh is needed to be refined. The concept is shown in Fig3.2. At a time step during the computation, the first-order differential of water level $\partial H/\partial x$ and $\partial H/\partial y$ as well as second-order differential $\partial^2 H/\partial x^2$, $\partial^2 H/\partial xy$ and $\partial^2 H/\partial y^2$ are used to calculate the maximum change of water level ΔH_E in local mesh by the following equation.

$$H_E(\xi)=\left(\left|\frac{\partial H}{\partial x}\right|+\left|\frac{\partial H}{\partial y}\right|\right)\xi+\frac{1}{2}\left(\left|\frac{\partial^2 H}{\partial x^2}\right|+\left|\frac{\partial^2 H}{\partial xy}\right|+\left|\frac{\partial^2 H}{\partial y^2}\right|\right)\xi^2 \tag{3.1.1}$$

$$\Delta H_E = H_E(D) \tag{3.1.2}$$

If the ΔH_E is bigger than the preset water level reference ΔH_C, the mesh size D is considered to be too big for the sharply changing water level. Then, the mesh is labeled to be refined at next step. On the other hand, if the change of water level meet condition

$$H_E\left(\sqrt{3}D\right) < \Delta H_C \tag{3.1.3}$$

The mesh is considered to be big enough to represent the water level change, and is labeled to be coarse back to the upper grade mesh. In case of the mesh is refined or coarsened, the solution variables **q** of new meshes will be rearranged from the **q** of former mesh, which will maintain conservation of mass locally.

3.3 Numerical experiment

In order to investigate the effectiveness of the proposed adaptive mesh refinement (AMR) approach on unstructured triangular mesh system in solving tsunami movement, two kinds of idealized numerical experiments are conducted. Note that the adaptive mesh refinement (AMR) method is just applied to SRNH scheme in all computing case, and unless explicitly stated, it is called "AMR" for short in following discussion.

3.3.1 Tsunami propagate on deep-ocean

3.3.1.1 Computational conditions

This experiment is conducted to check the effectiveness of the proposed adaptive mesh refinement (AMR) in reproducing tsunami propagating on deep-ocean. The computational domain and the initial conditions please refer to Section2.5.2. The meshes are listed in Table3.1, in which the coarsest mesh A1 is used for the implementation of AMR and acting as the "Level 1" mesh. For the convenience of confirm the accuracy of AMR method, the maximum refine level "ML" are changed to 3, 5, 7 or 11 to guarantee the minimum mesh size are close to the size of mesh A2, A3, A4 and A5 respectively. The ΔH_C is set to be 0.01 meter. The time steps are changed adaptively to make sure the CFL is equal to 0.1. Six hours' tsunami propagation is simulated using Roe scheme, SRNH scheme as well as the AMR implementation respectively.

3.3.1.2 Results and discussion

Fig3.3 shows the time series results of AMR implementation with ML=11. As seen from the figures, the mesh is refined adaptively tracing the wave propagation, which indicates that the proposed AMR method could capture the wave movement accurately. And we can also confirm that every mesh is adjacent to mesh that is just one "Level" bigger or smaller than itself. In other words, there is no sharply size change all over the computational domain. Fig3.4 shows time series results of tsunami propagate along transverse middle line, which are sorted by mesh size. As seen from case (a), (b) and (c),

the wave height calculated using AMR method is a little lower than SRNH scheme's, but higher than Roe scheme's result. A possible reason is that the finest mesh size of AMR implementation is still a little bigger than the mesh used by Roe and SRNH scheme (Table3.1), the result of AMR can't achieve the same resolution as SRNH scheme. However, the proposed AMR strategy is effective in capturing wave movement, which lead to the result of AMR is better than first-order Roe scheme. As you see in case (d) of Fig3.4, if the finest mesh size is equal to that used by SRNH scheme, the application of AMR could achieve almost the same resolution as SRNH scheme do. In addition, no matter what mesh or numerical scheme is used, the relative error of computational wave velocity is smaller than 1%.

The relationship between the attenuation (H_{cal}/H_{theo}), mesh size and propagate distance are shown in Fig3.5. The result of AMR implementation shows the same trend as SRNH scheme that attenuation increases with increasing propagate distance. Meanwhile the difference of attenuation among meshes with different spatial size will also increase.

The relationship between relative error and mesh number "N" at different propagating distances are shown in Fig3.6. There figures also reveals a fact that the proposed AMR strategy could get almost the same result with second-order SRNH scheme if the mesh resolution is the same.

Fig3.7 shows the computing load of each case, in which the horizontal axis represents finest mesh size. As seen in Fig3.7(a), considering the AMR method just refine the mesh in interesting region, the computing load of AMR application (red circle) is much lower than that is discretized by uniform mesh (white circle). The CPU time summarized in Fig3.7(b) confirmed that the computing load is reduced by AMR method indeed. For example, in case of ML=11 in AMR application, both of the computing load and CPU time are reduced to one tenth.

3.3.2 Run-up tsunami on sloping beach

3.3.2.1 Computational conditions

This experiment is designed to testify the ability of proposed AMR strategy in treating sharply increased wave height in near-shore region. Please refer to Section2.5.3 for the initial conditions and other settings. As shown in Table3.2, five kinds of uniform meshes (R1-R5) are used in this experiment as well as the four kinds of refined meshes (ML=3, 5, 7, 11) which is refined based on mesh R1.

3.3.2.2 Results and discussion

The time series results of wave height using AMR method with different refine levels are compared with Carrier & Greenspan's (1957) theoretical solution in Fig3.8. According to the finest mesh size (from big to small), the results are sorted to four groups (uniform mesh R2-R5 corresponding to AMR ML=3, 5, 7, 11). For the three cases that mesh are coarse (Fig3.8.1, 3.8.2, 3.8.3), the result of AMR application trend to be a little lower than the SRNH scheme because the finest mesh size of AMR application is still bigger than the corresponding uniform mesh (see Table3.2), but the advantages of adaptive local mesh refinement are represented specially after the wave crashed (wave crest at t=60min, 70min, 80min) with beach that the numerical oscillation is reduced in AMR application compared with SRNH scheme. Furthermore, if refine the uniform mesh R1 to ML=11, the result of AMR application after wave crash (t=60min, 70min, 80min) fit theoretical solution much better than SRNH scheme. Fig3.9 summarized the relative error of wave crests (t=10min-50min) calculated using Roe scheme, SRNH scheme and AMR method. The results of first 40 minutes show that the AMR application can get almost the same results with Roe and SRNH scheme. In addition, the AMR method could achieve much higher resolution than Roe and SRNH scheme even the Green's law is not applied to estimate wave height at 50 minutes. And the relative errors of all the four cases of AMR implementation are below 0.1.

3.4Application to tsunami in 2011 Tōhoku Earthquake

The proposed AMR method is also applied to simulate the tsunami in 2011 Tōhoku Earthquake to verify its ability in solving real tsunami movement.

3.4.1 Computational conditions

The computational domain and related conditions please refer to Section2.6.1. In this simulation, two kinds of meshes (mesh T1 and T3) are used for SRNH applications as well as mesh T1 is refined to ML=5 for AMR application. The detail of mesh is listed in Table3.3, in which the finest mesh size of AMR in epicenter is almost the same with mesh T3, and the finest mesh size in coast for AMR is half of the mesh T3. Considering that the maximum observational wave height in observation stations various from 2 to 6 meters (refer to Fig2.23), the ΔH_C is decided as follows to guarantee the 2 meters wave can be represented exactly in this simulation. Assume the water depth near-shore is 10 meters order and 1000 meters order offshore which lead to a ratio about 1/100. And in view of the shallow water deformation, the 2 meters wave height near-shore is correspond to $2\times(1/100)^{1/2}$=0.2 meter offshore. As discussed in Section2.5.2, N=15 meshes per wave (8 for half wave length) length is required for SRNH scheme to ensure the relative error smaller than 0.1 for offshore wave propagation. Therefore, if we apply this condition to reproduce 0.2 meter wave's propagation, an averaged 0.2/8=0.025 meter wave height change per mesh is required. Based on above-mentioned reasons, the water level reference ΔH_C is set to be 0.025 meter for the simulation of tsunami in 2011 Tōhoku Earthquake.

3.4.2 Results and discussion

3.4.2.1 Results of wave height

Fig3.10 shows time series results of wave propagation as well as the refined meshes, which indicate that the meshes are refined exactly according to the wave propagation meanwhile the meshes will also be coarsened back after the wave passed through (t=30, 40, 60 minutes). This verified the proposed AMR method could also capture the wave

propagation in real tsunami simulation. Fig3.11 shows the comparison between computing results and observational data. As seen in these figures, the results of AMR implementation are much closer to observational data compared with that not using AMR method. Specially, for the result in NM and CM, the AMR implementation has improved the wave height more than 2 meters higher than that without AMR. Fig3.12 shows the enlarged view around NM and CM, which is seemed to reveal the reason for this big difference on wave height result. As you see in the t=10 minutes' result, in response to the wave front, the mesh of AMR implementation is refined locally to about 1/3 of the size of mesh T3. This adaptively mesh refinement is considered to have reduced the numerical diffusion error and improved the resolution of real tsunami simulation. Based on the difference of numerical diffusion error and the results of t=20min shown in Fig3.12, we can predict that the first wave length computed using mesh T3 without AMR is much bigger than AMR implementation. In addition, as seen from (b) and (c) of Fig3.11, no matter what method is used, the computing wave height in CI and SI are much lower than the observational data. The initial wave distribution and topography data should be checked and verified to find the reason, which lead to a special topic for the future study.

3.4.2.2 Comparison of CPU time

The total mesh number and run-time (CPU time) for one hour's tsunami propagation are listed in Table3.3. By using AMR method, the computational load (total mesh number) is reduced a lot comparing with mesh T3. In detail, the CPU time is reduced to about 1/3 of the mesh T3 without AMR.

3.4.2.3 Effect of ΔH_C

In order to investigate the effect of ΔH_C on real tsunami simulation, the same calculation is also conducted using 10 time's bigger $\Delta H_C = 0.25m$ and 10 time's smaller $\Delta H_C = 0.0025m$. The mesh states and computing results are shown in Fig3.13 and Fig3.14 respectively. As seen from Fig3.13 (a), although the refine area is enlarged a lot, there is no obvious improvement of the computing results (see Fig3.14 (a)).

However, the CPU time have increased about 9% comparing with case (b) $\Delta H_C = 0.025$, which indicate that the usage of too small ΔH_C can't be expected to greatly improve the resolution of tsunami simulation but just increase the computational load. On the other hand, if enlarge the ΔH_C 10 time's bigger to 0.25 meter, the refine area is minimized (see Fig3.13 (c)) and the CPU time is reduced to 30% of the case (b) $\Delta H_C = 0.025$. Unfortunately, just as shown in Fig3.14 (c), the results deteriorate sharply. In conclusion, the appropriate value of ΔH_C plays an important role in achieving rapid and high-resolution tsunami simulation using AMR method.

3.5 Conclusions

In this chapter, a new approach of adaptive mesh refinement (AMR) is proposed. Two kinds of idealized numerical experiment are conducted with several meshes of different spatial size to verify the abilities of the proposed new AMR method in solving wave propagate on deep-ocean and slopping beach respectively. The results indicate that the proposed AMR method works well in capturing wave propagation and could reduce the computational load as well as reduce the CPU time down to 1/10 of the finest mesh without AMR. Furthermore, the proposed AMR method is also applied to the simulation of tsunami in 2011 Tōhoku Earthquake. The computational wave height got improved by using AMR method indeed, but the CPU time reduced to about 1/3 of the mesh T3 without AMR. In addition, the effect of ΔH_C is also investigated which indicate that the appropriate value of ΔH_C plays an important role in achieving rapid and high-resolution tsunami simulation using AMR method.

REFERENCE

George, D. L., and LeVeque R. J.: Finite volume methods and adaptive refinement for global tsunami propagation and local inundation, *Science of Tsunami Hazards*, Vol. 24, pp. 319-328, 2006.

Titov, V. V., and Gonzalez F. I.: Implementation and testing of theMethod of Splitting Tsunami (MOST) model, *NOAA Tech. Memo.*, ERLPMEL-112, 11 pp., Pac. Mar. Environ. Lab., NOAA, Seattle, Wash., 1997.

Titov, V. V., and Synolakis, C. E.: Numerical modeling of tidal wave runup, *J. Waterway. Port Coastal Ocean Eng.*, Vol. 124(4), pp. 157–171, 1998.

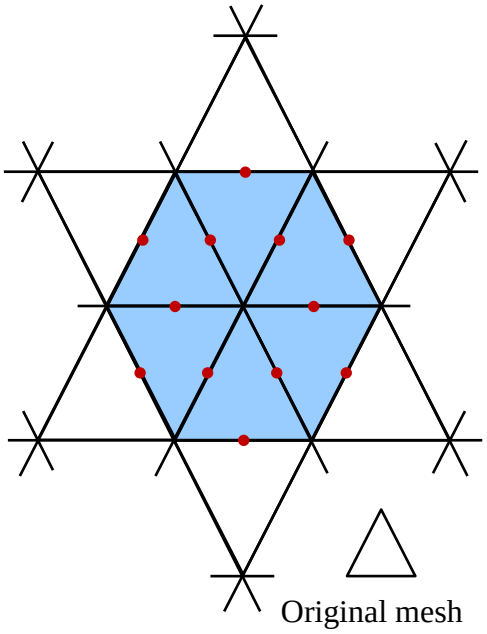
Titov, V. V., Rabinovich, A. B., Mofjeld, H. O., Thomson, R. E., and Gonzalez, F. I.: The Global Reach of the 26 December 2004 Sumatra Tsunami, *Science*, Vol. 309(5743), pp. 2045-2048, 2005.

Bank, R. E., Sherman, A. H., and Weiser, A.: Some refinement algorithms and data structures for regular local mesh refinement, *Scientific Computing, Applications of Mathematics and Computing to the Physical Sciences*, Vol. 1, pp. 3-17, 1983.

Behrens, J., and Bader, M.: Efficiency considerations in triangular adaptive mesh refinement, *Phil. Trans. R. Soc. A*, Vol. 367, pp. 4577-4589, 2009.

Popinet, S.: Adaptive modeling of long-distance wave propagation and fine-scale flooding during the Tōhoku tsunami, *Natural Hazards And Earth System Science*, Vol.12, pp. 1213-1227, 2012.

(a)



(b)

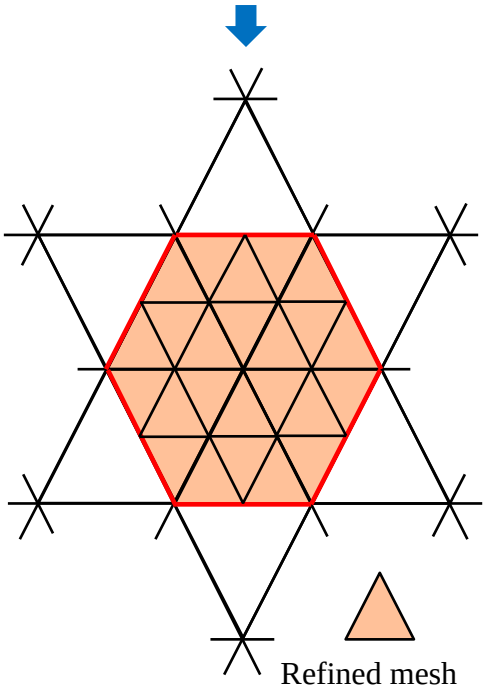


Fig3.1.1 Concept of bisection-type mesh refinement applied on triangular mesh (Bank et al., 1983)

(a)

(b)

(c)

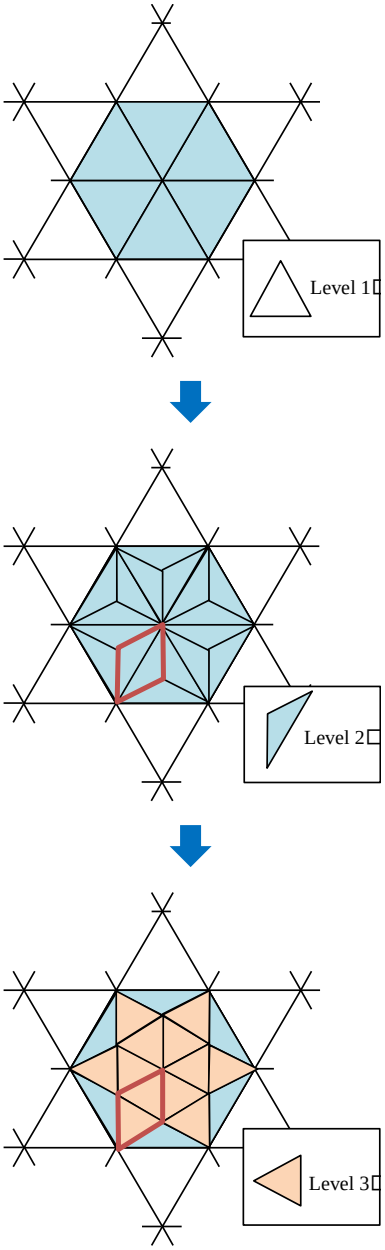


Fig3.1.2 Concept of the proposed new AMR strategy

(a)

(b)

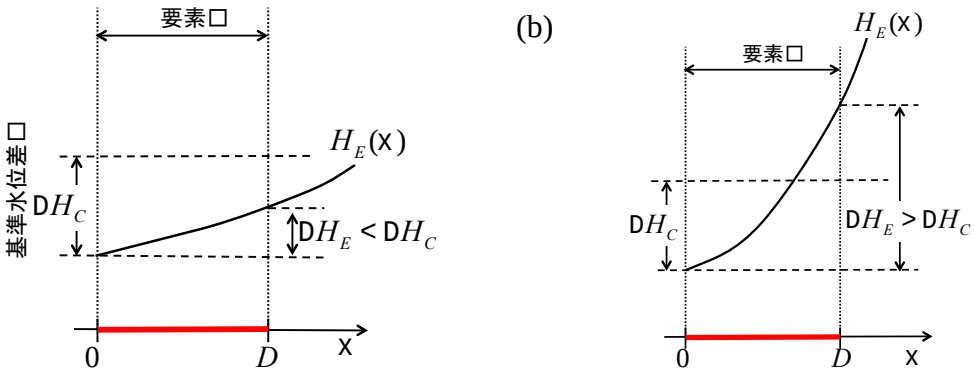


Fig3.2 Criterions of mesh refining

Table3.1 Mesh used for tsunami propagating on deep-ocean

Mesh	A1	A2	A3	A4	A5	AMR ML=3	AMR ML=5	AMR ML=7	AMR ML=11
Max size [km]	80	40	20	10	5	80			
Min size [km]						46	27	15	5

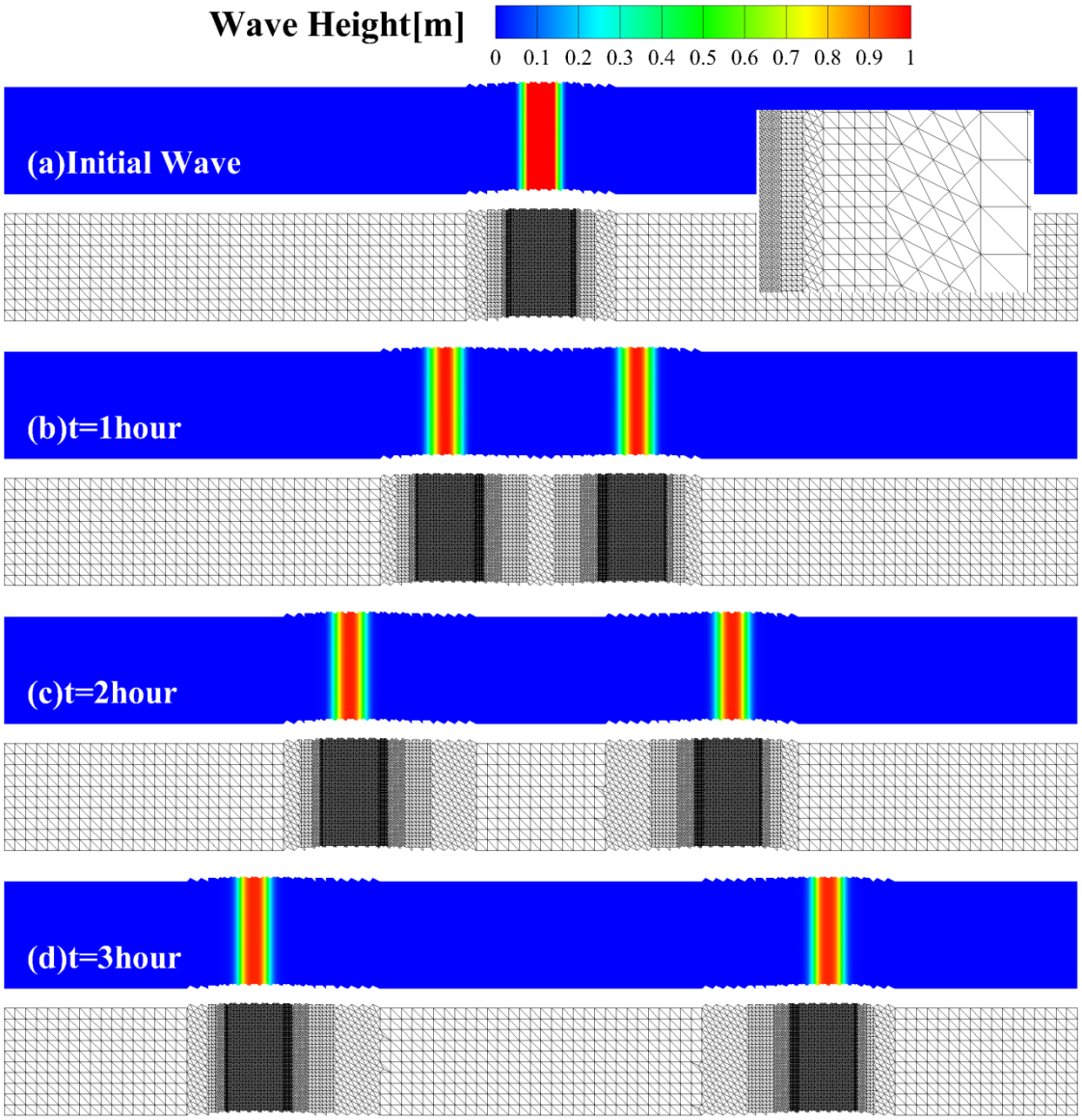
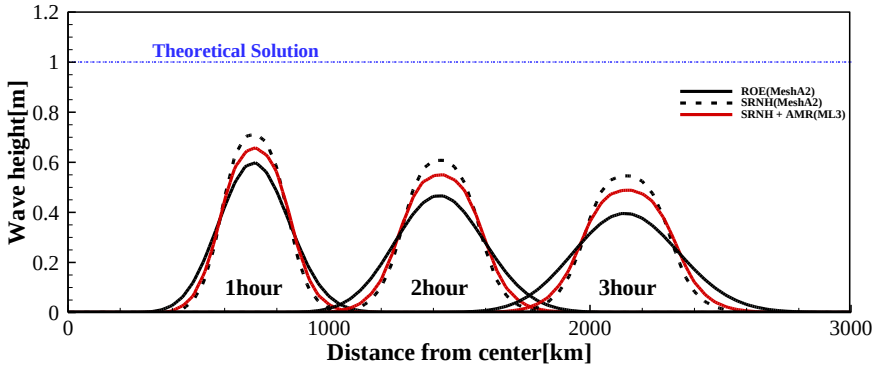
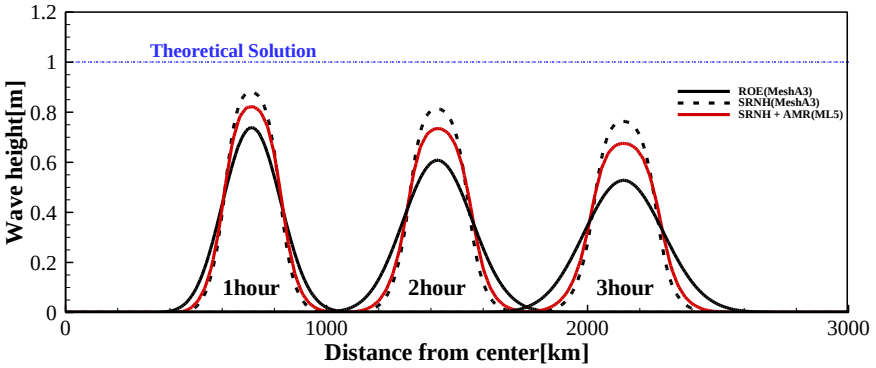


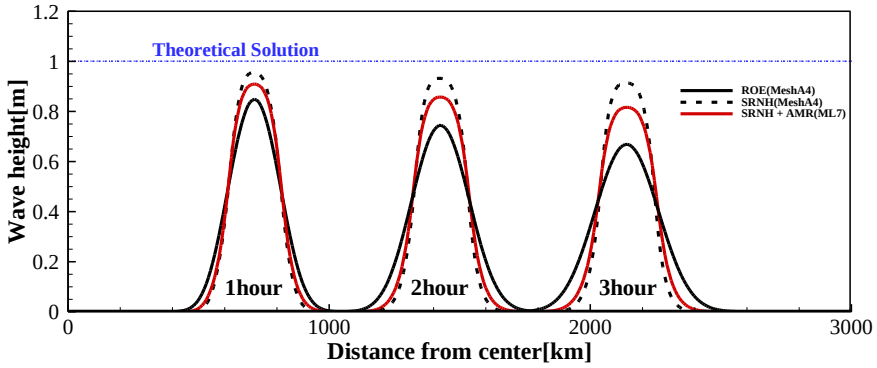
Fig3.3 Actual meshes according to AMR (ML11)



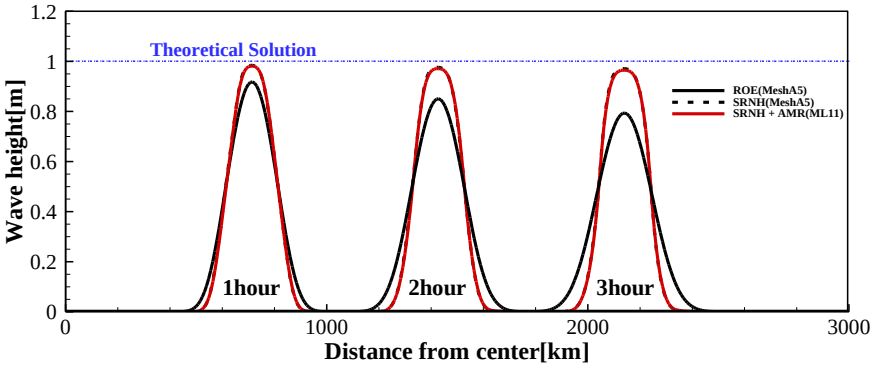
(a) AMR (ML=3)



(b) AMR (ML=5)



(c) AMR (ML=7)



(d) AMR (ML=11)

Fig3.4 Time series results of tsunami propagate (along transverse middle-line) on deep-ocean using AMR

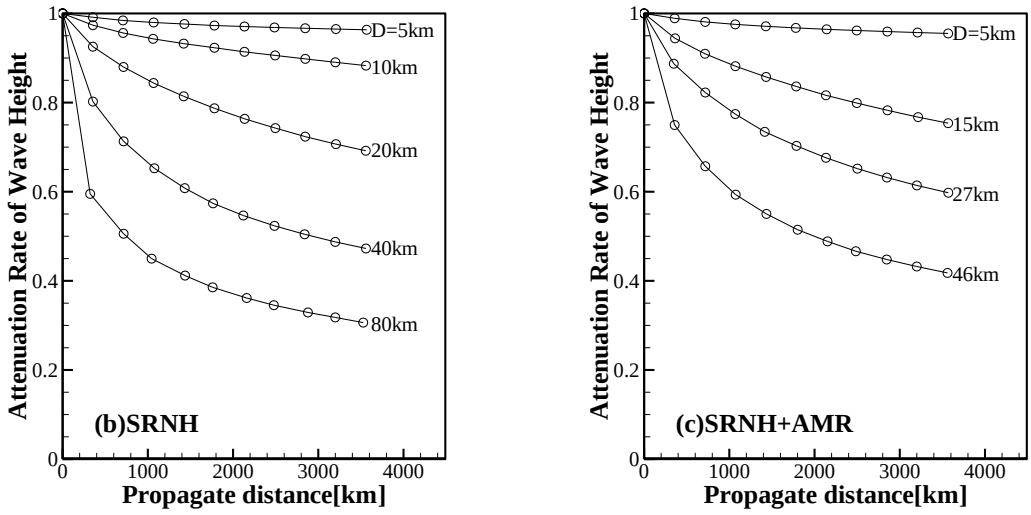


Fig3.5 Attenuation rate of computational wave

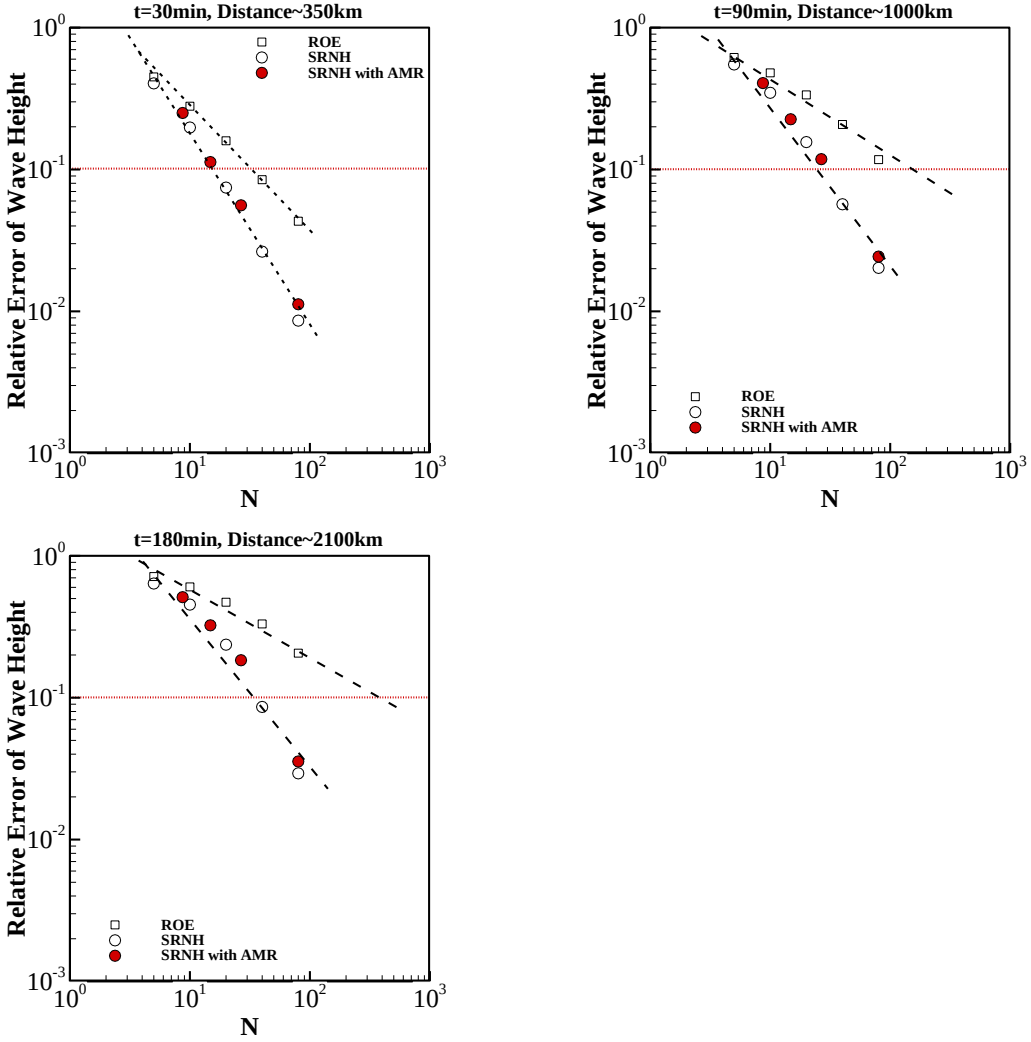
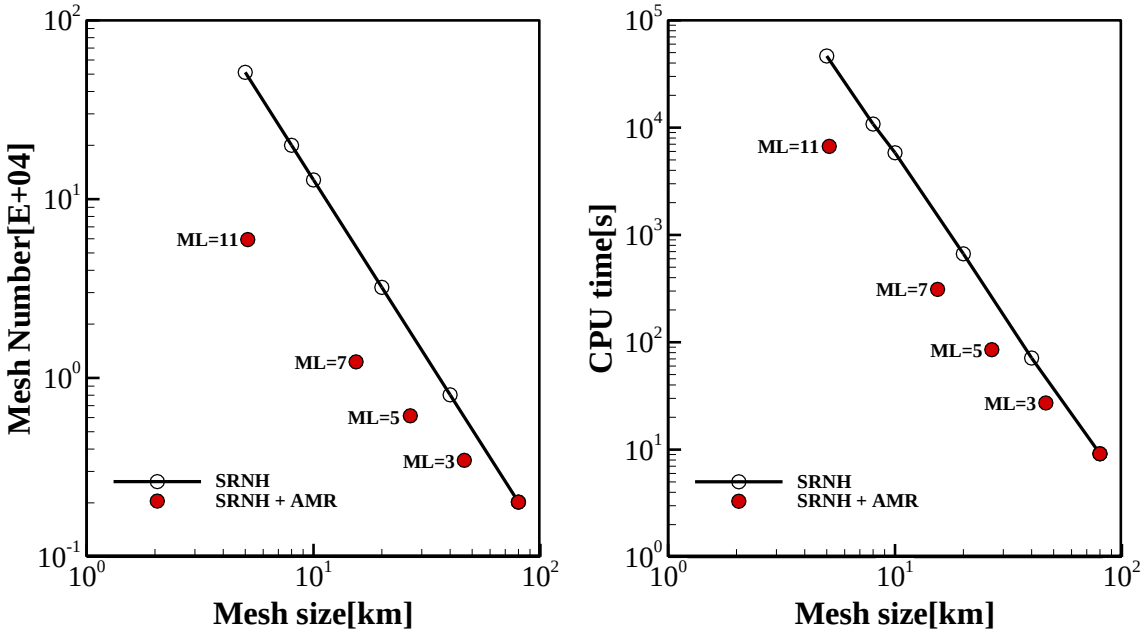


Fig3.6 Relative error of wave height using AMR



(a) Relationship of mesh size and mesh number (b) Relationship of mesh size and CPU time

Fig3.7 Computational load and CPU time of AMR

Table3.2 Mesh for run-up tsunami on slopping beach

Mesh	A1	A2	A3	A4	A5	AMR ML=3	AMR ML=5	AMR ML=7	AMR ML=11
Max size [km]	2	1	0.5	0.25	0.125	2			
Min size [km]						1.15	0.67	0.38	0.22

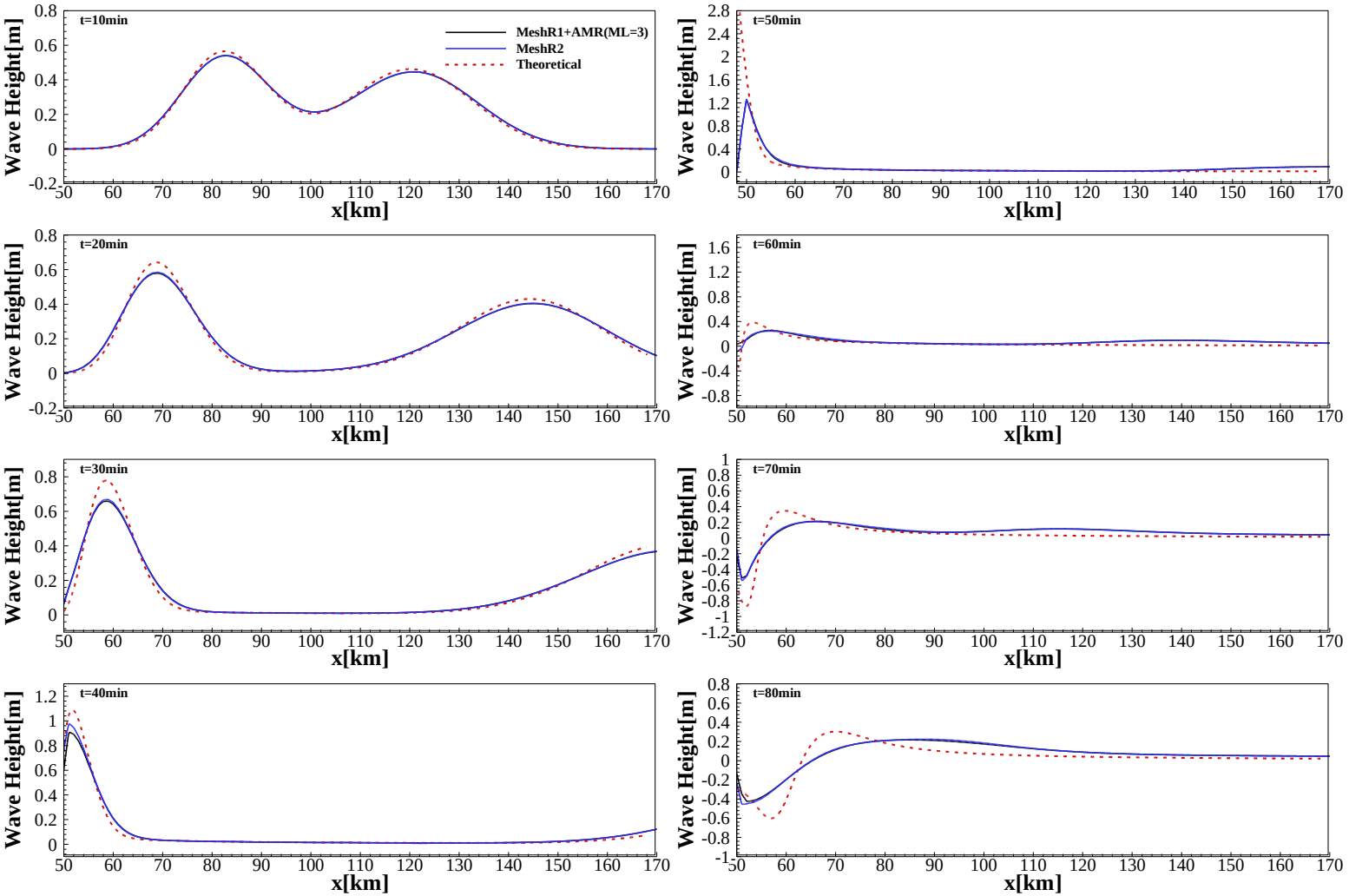


Fig3.8.1 Comparison between MeshR2 and MeshR1+AMR (ML=3)

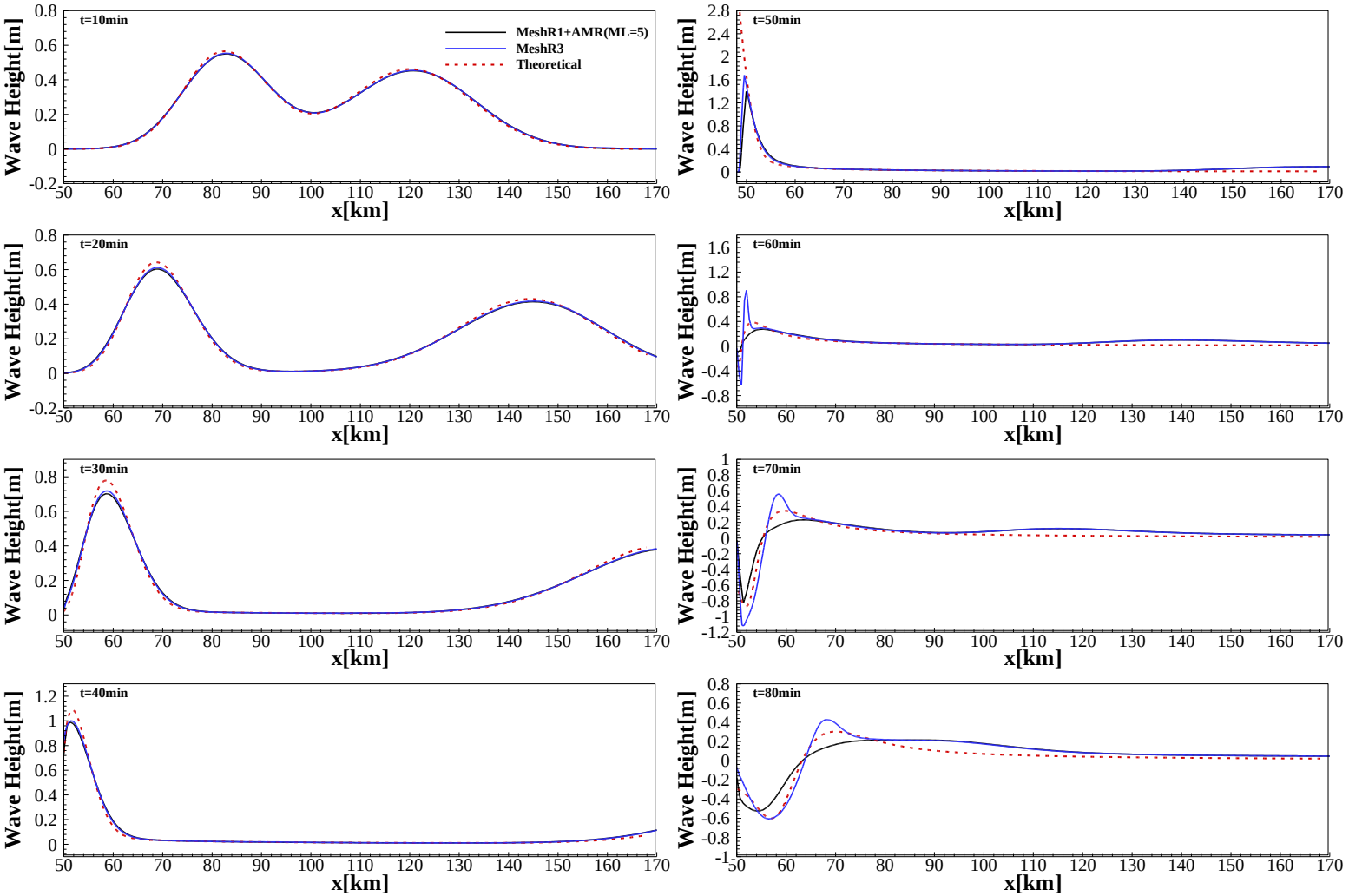


Fig3.8.2 Comparison between MeshR3 and MeshR1+AMR (ML=5)

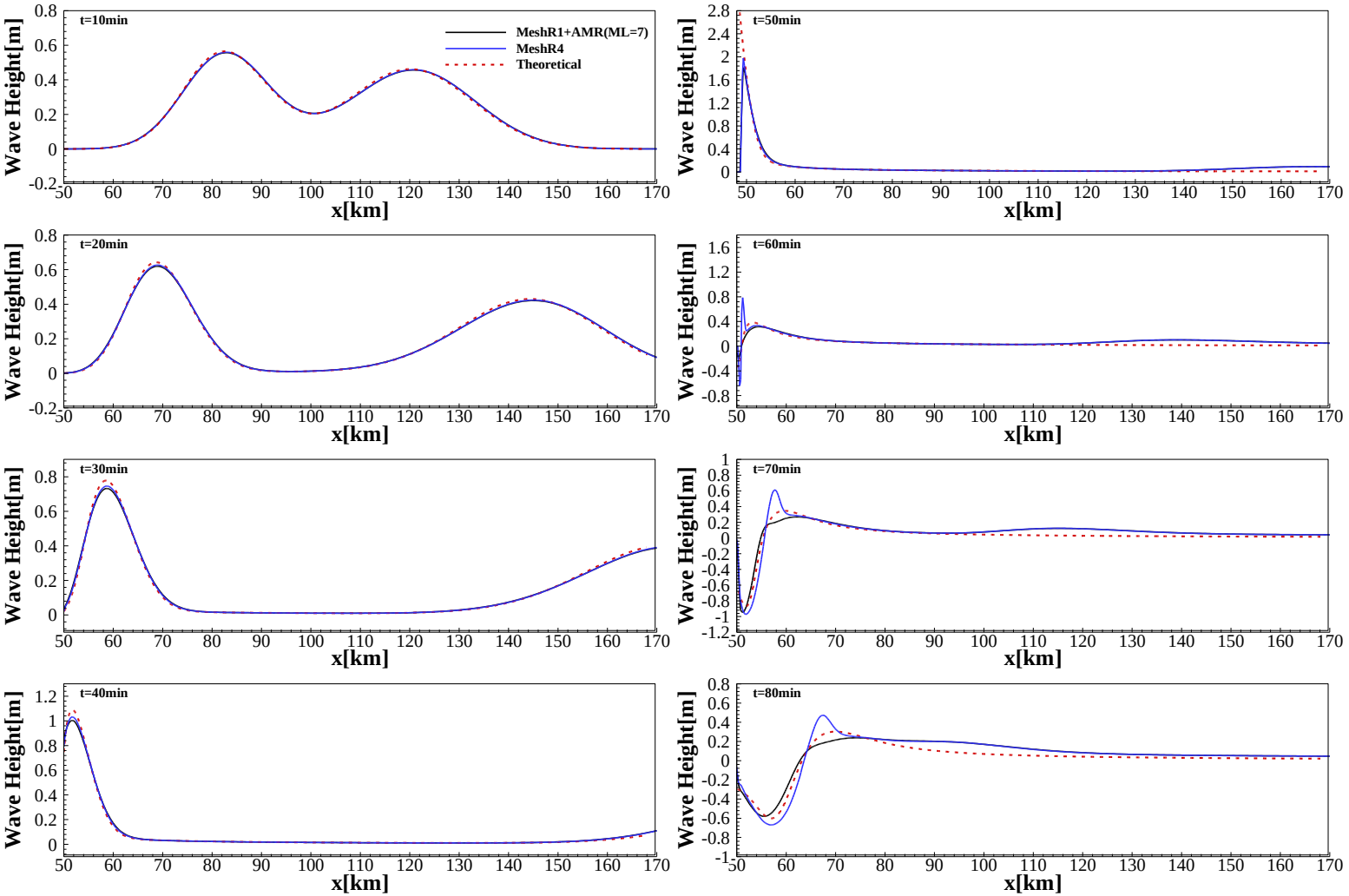


Fig3.8.3 Comparison between MeshR4 and MeshR1+AMR (ML=7)

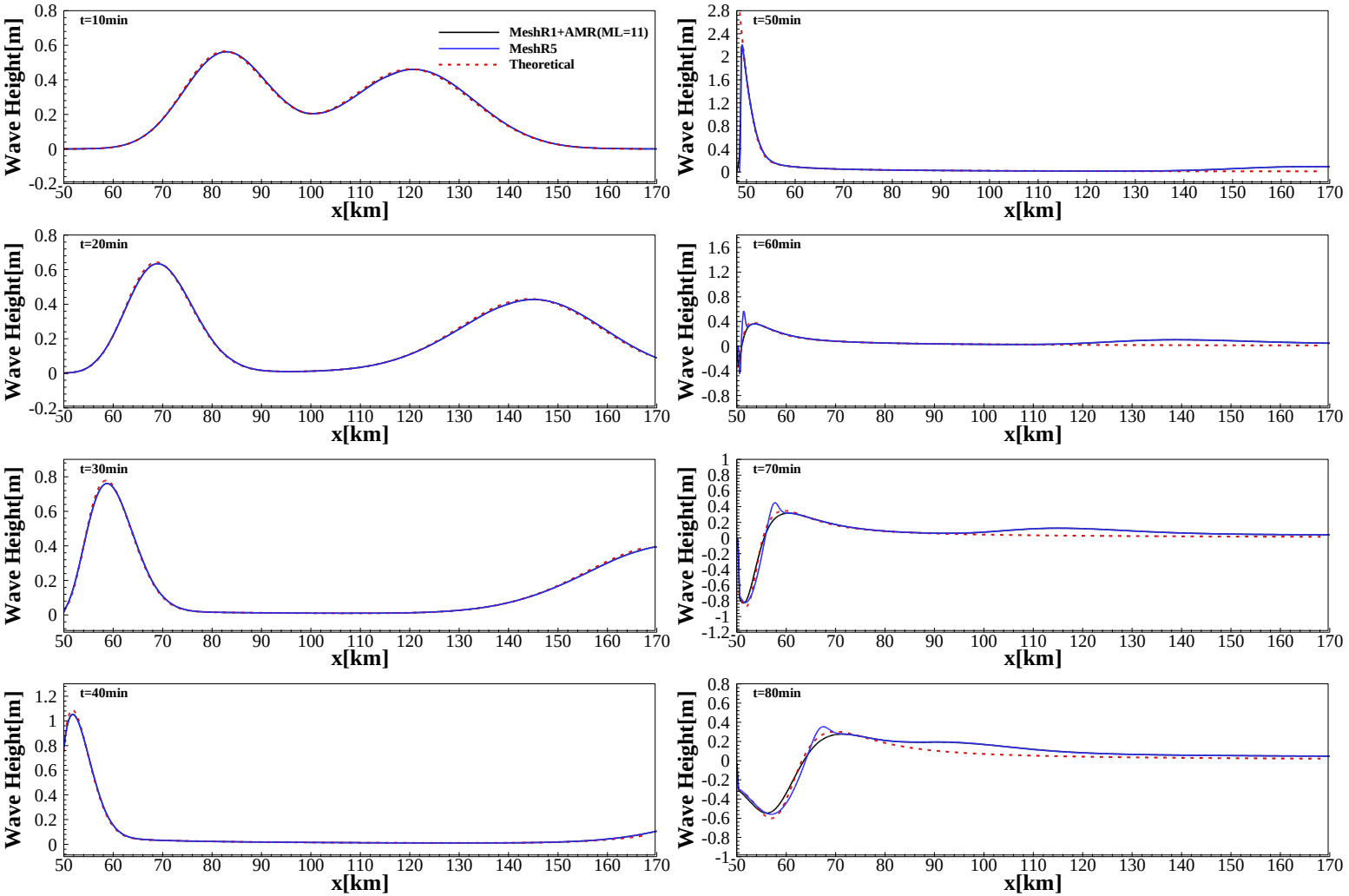


Fig3.8.4 Comparison between MeshR5 and MeshR1+AMR (ML=11)

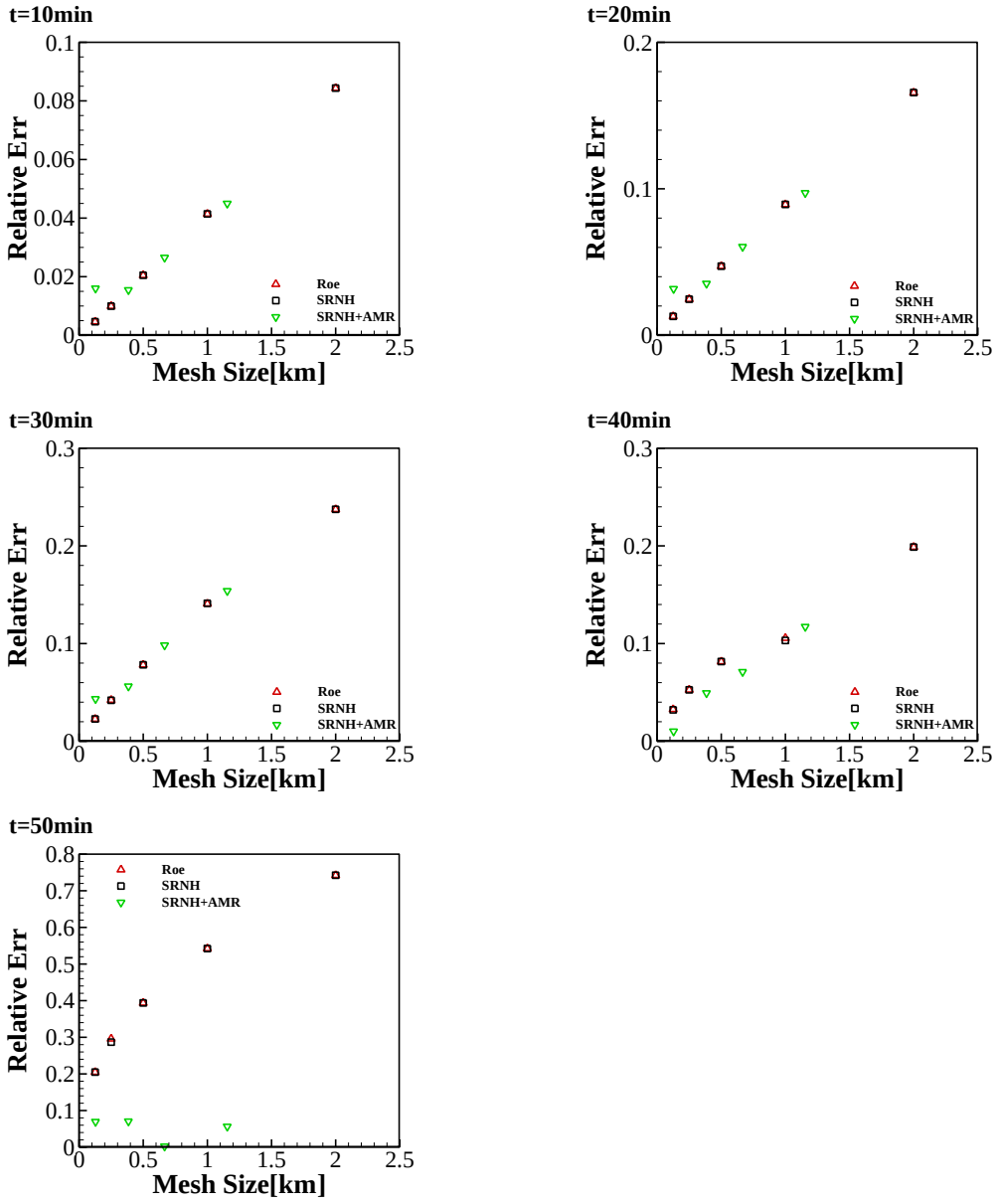
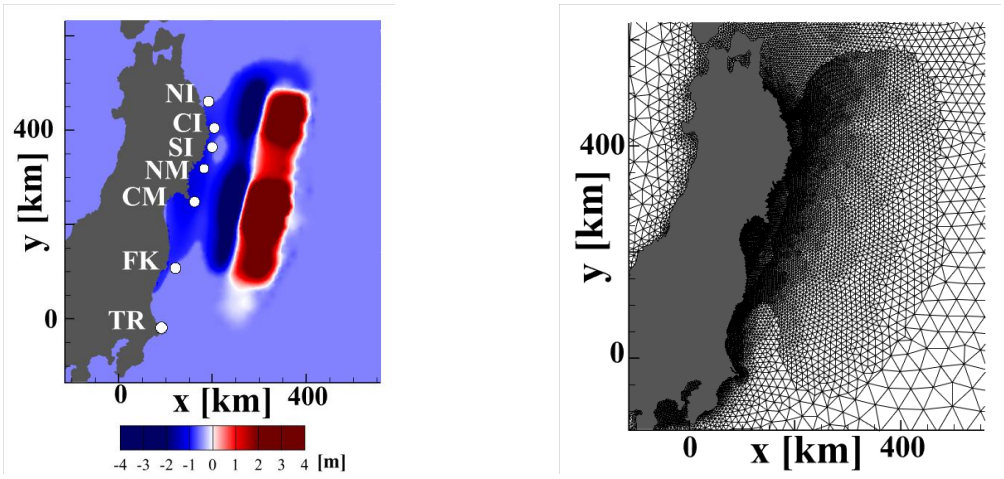


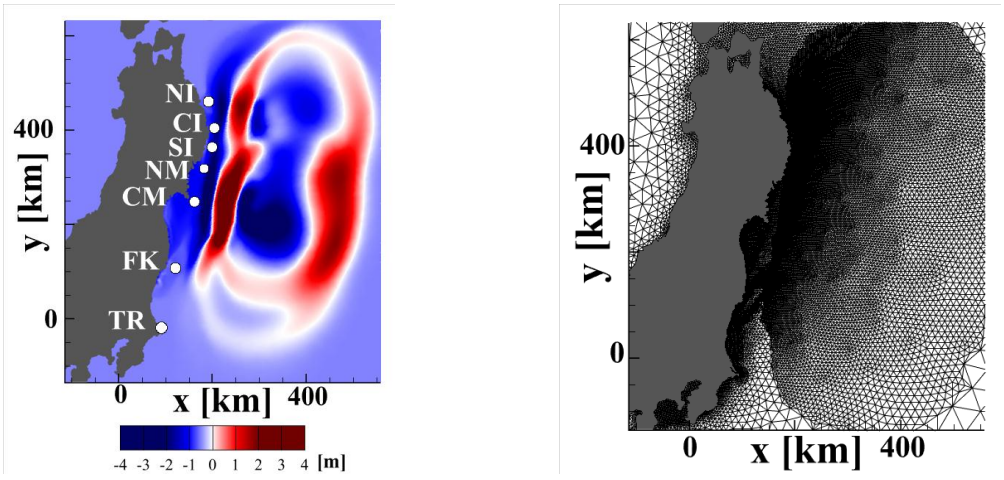
Fig3.9 Time series results of relative error

Table3.3 Mesh used for tsunami in 2011 Tōhoku earthquake

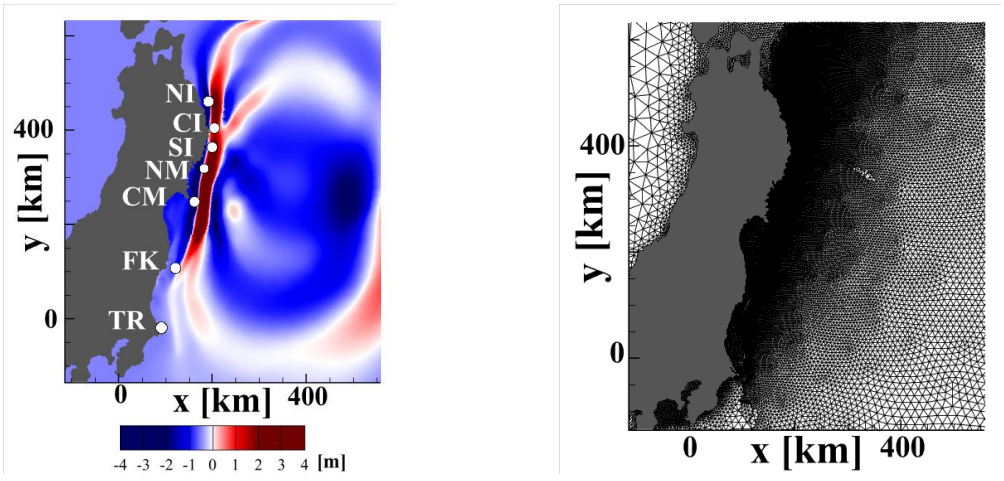
Mesh		Mesh T1	Mesh T3	Mesh T1+AMR(ML=5)
Mesh size D (km)	Epicenter	14.0	6.0	14.0(Max)~6.0(Min)
	Near shore	2.5	1.5	2.5(Max)~0.8(Min)
Mesh number		34,060	266,194	34,060(Min)~188,078(Max)
N (mesh number/wave length)		7	15	7~15
CPU time (min)		12	480	162



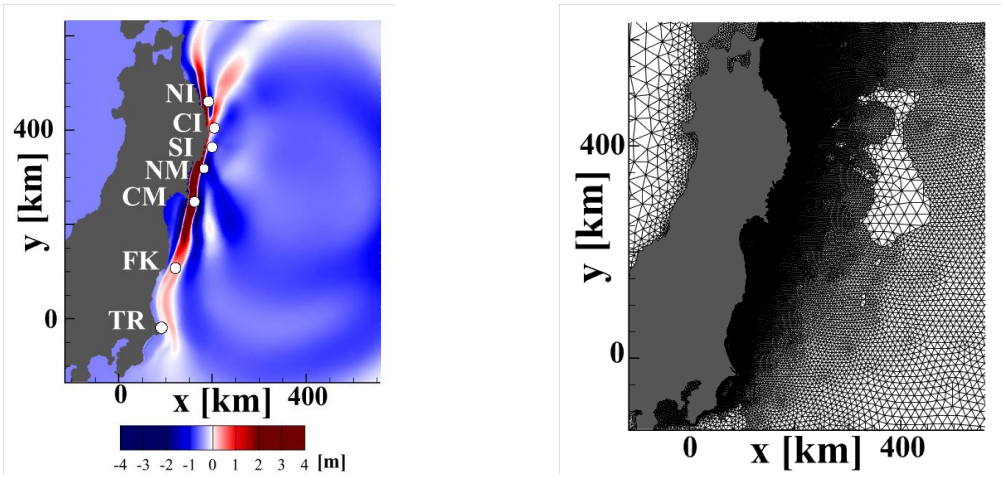
(a) Initial



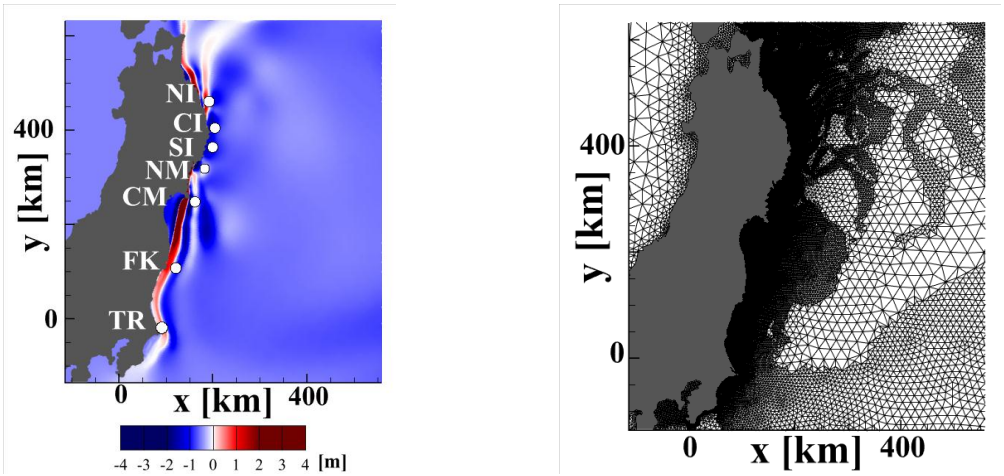
(b) t=10min



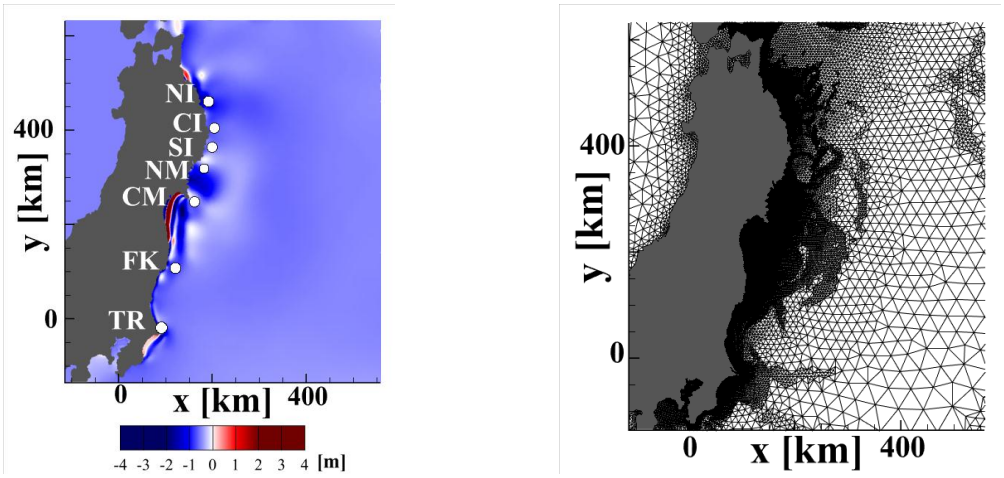
(c) t=20min



(d) t=30min



(e) t=40min



(f) t=60min

Fig3.10 Computational results of wave propagation and mesh status using AMR

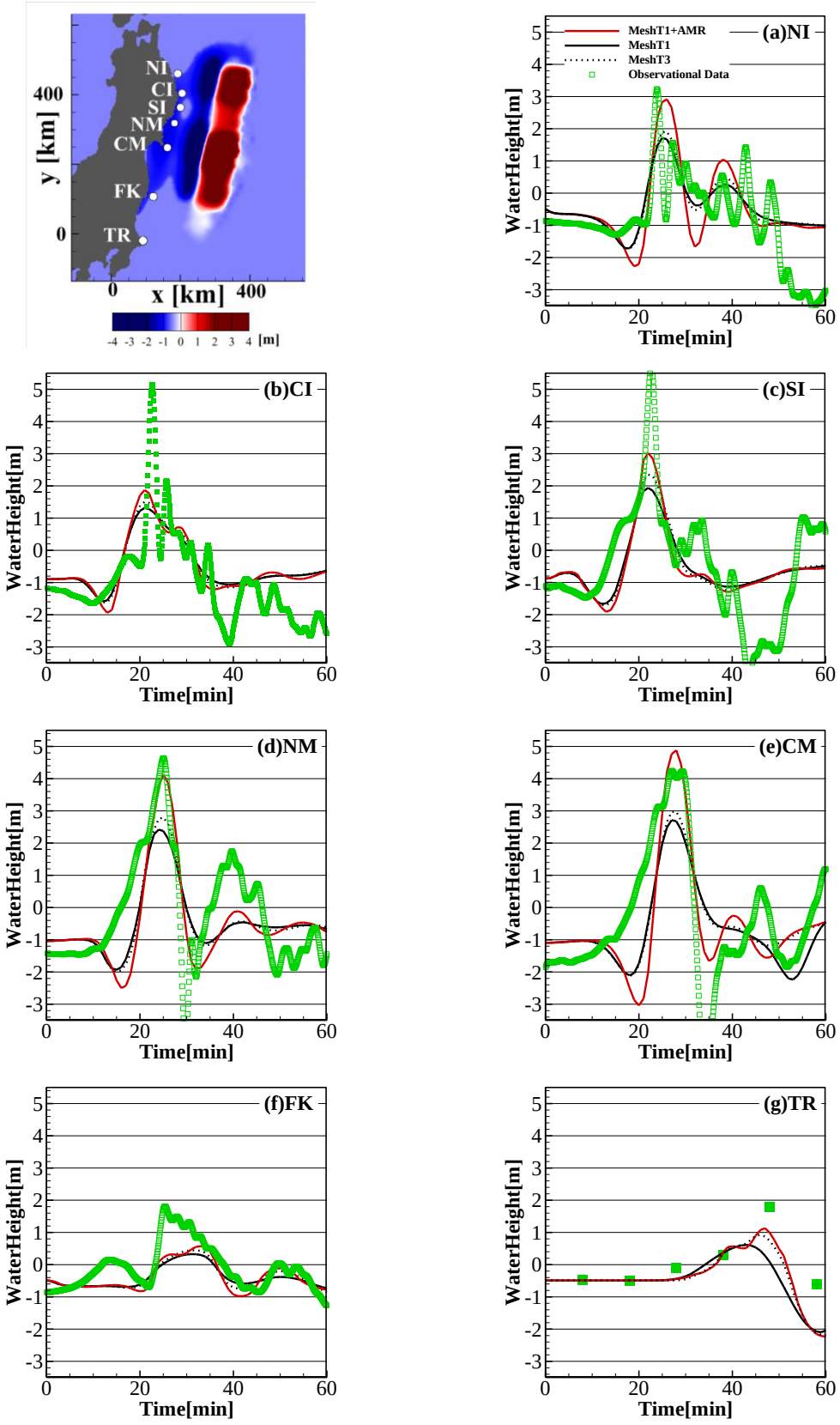
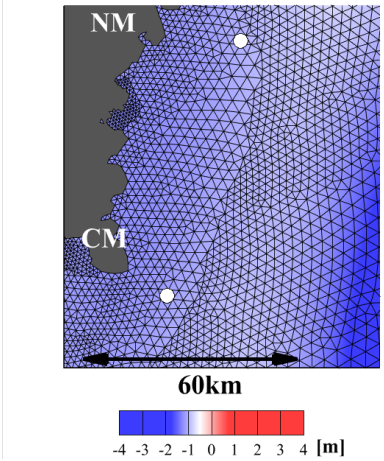
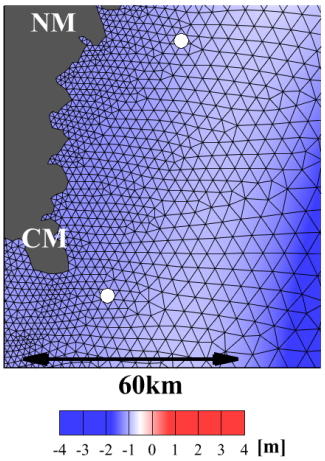


Fig3.11 Results comparison between SRNH (MeshT1, T3) and AMR (ML=5)

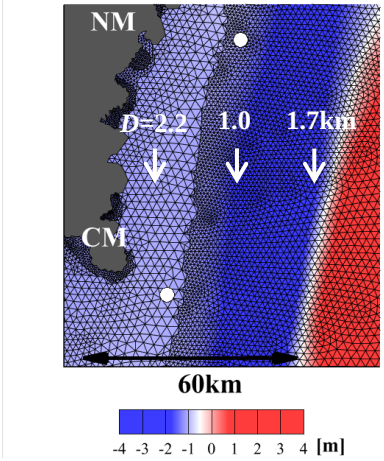
MeshT1+AMR (ML=5)

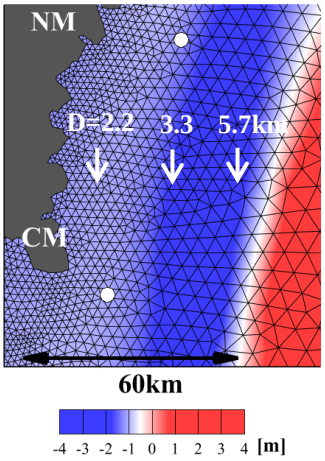
MeshT3



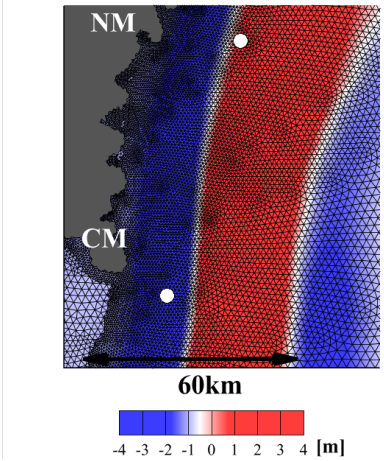


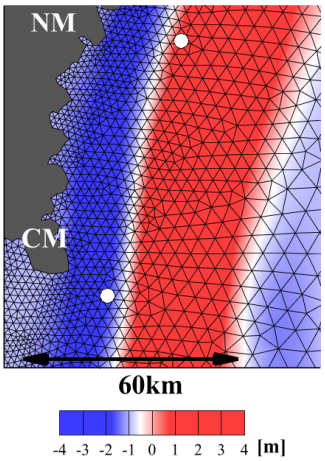
Initial





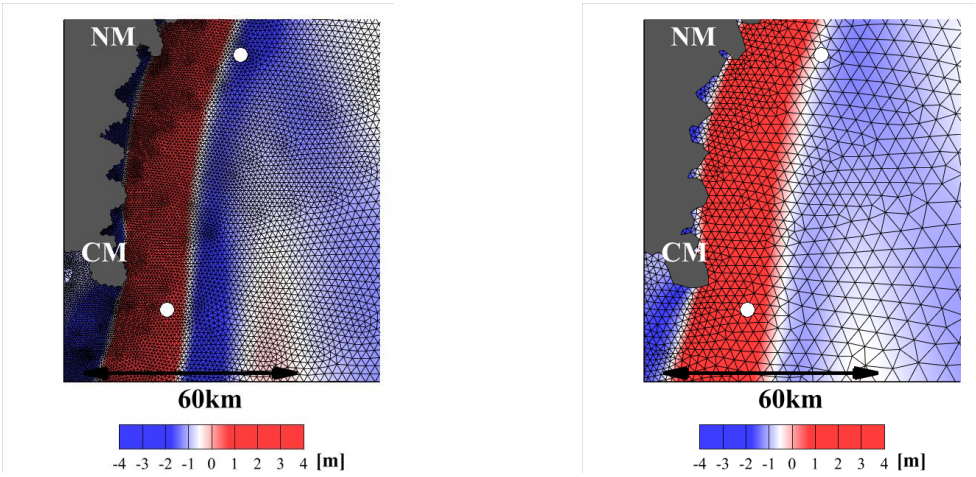
t=10min



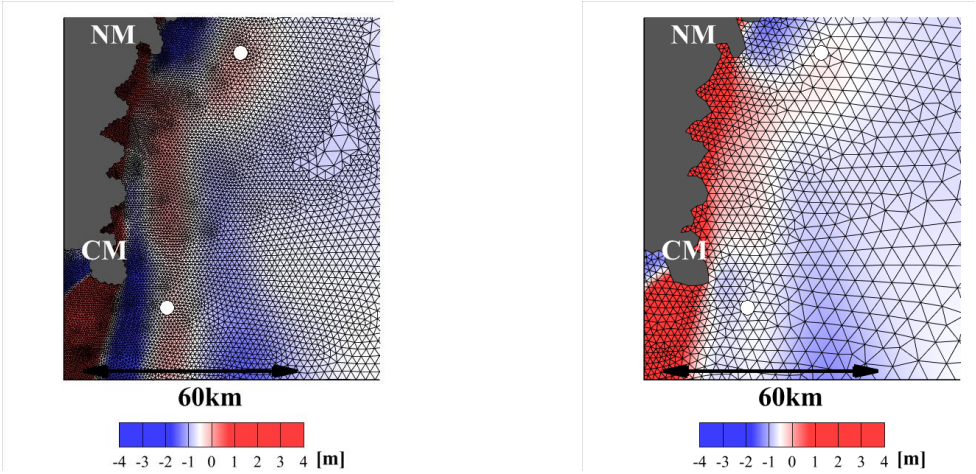


t=20min

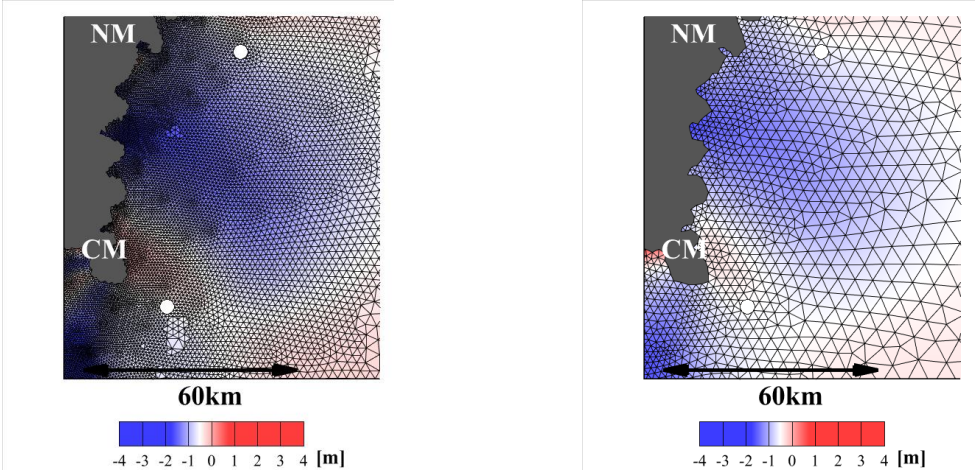
MeshT1+AMR (ML=5)　　MeshT3



t=30min



t=40min



t=60min

Fig3.12 Comparison of mesh status between MeshT1+AMR (ML=5) and MeshT3

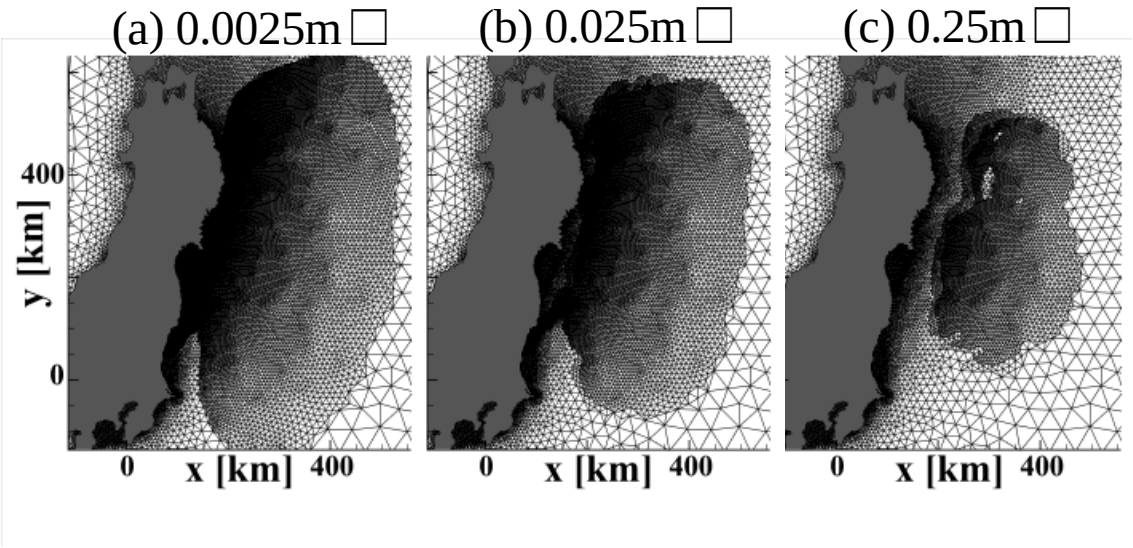


Fig3.13 Comparison of refined mesh area using different ΔH_C (a. ΔH_C=0.0025m; b. ΔH_C=0.025m; c. ΔH_C=0.25m)

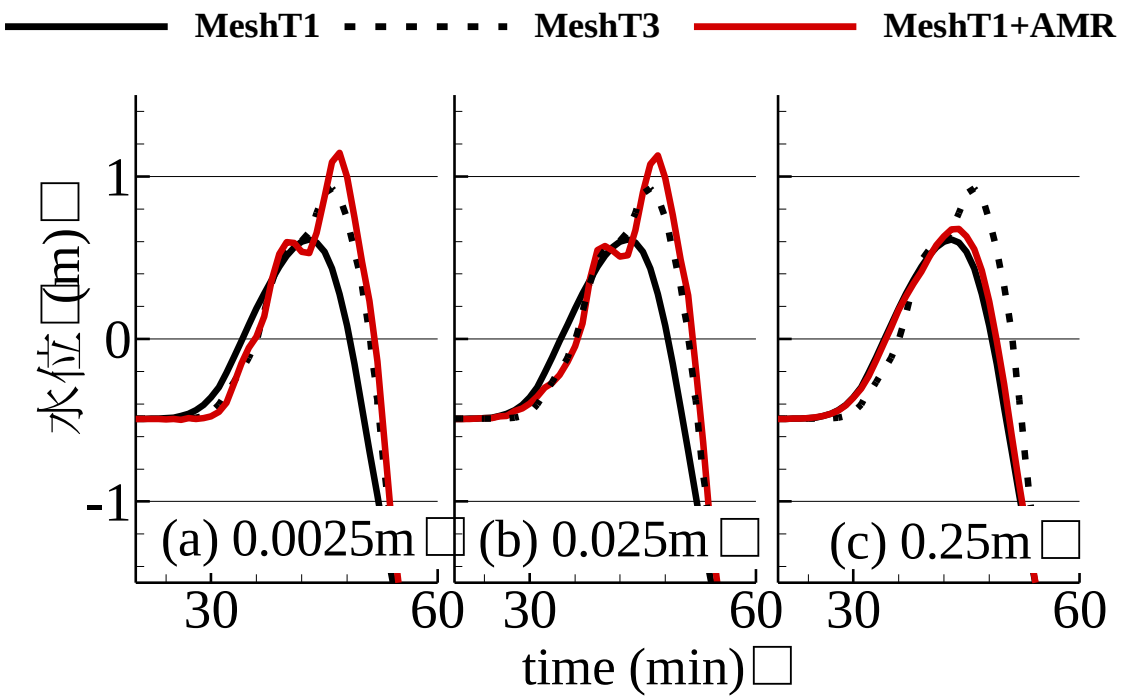


Fig3.14 Comparison of results in TR using different ΔH_C (a. ΔH_C=0.0025m; b. ΔH_C=0.025m; c. ΔH_C=0.25m)

Chapter4 Rapid and high-resolution tsunami simulator based on General-Purpose computing on GPU (GPGPU)

4.1 Motivation

As mentioned in Chapter1, run-up tsunami in rivers is observed after offshore earthquake especially in 2011 Tōhoku Earthquake, which is seemed to have extended the inland flood area. Therefore, the numerical simulation of run-up tsunami in rivers is becoming a hot topic recently. However, as summarized in Section2.5.3, more than 200 meshes per one wavelength are required for high-resolution run-up tsunami simulation, which is inefficient for AMR implementation. Considering that AMR method has limitation for the case that the spatial scale changing dramatically, parallel computation using GPGPU is selected to achieve rapid and high-resolution tsunami simulation.

4.2 Introduction of parallel computing using GPU

General-Purpose computing on GPU (GPGPU) is a parallel computing technique using GPU which have gained widespread attention in recent years. The Graphics Processing Unit (GPU, nVIDIA TESLA C1060, Fig4.1a) is originally designed to accelerate the operation of digital images for output to a display rather than large scale numerical computations. However, GPU are reasonably suitable for large scale numerical computations especially the computational fluid dynamics (CFD).

a. GPU has hundreds of cores to process parallel computations simultaneously while a CPU just consists of a few cores (see Table4.1). In addition, the CPU is optimized for sequential serial processing such as word processing, web seafaring and so on. By contrast, GPU is designed to stream a mass of data in, do a few computations on the data and then output for displaying rapidly, all of which are conducted in parallel.

b. The memory bandwidth inner GPU is much faster than CPU (see Table4.1), which means GPU could transfer data much quickly than CPU. This advantage is also crucial for large scale parallel computation because the data transfer may create a bottleneck in some application.

c. GPU is powerful in extensibility comparing with CPU. As shown in Fig4.1b, multiple GPUs (three GPUs) could be installed in mainboard easily.

d. GPU have relative low price, but high performance. For example, although a nVIDIA TESLA C1060 will cost about 5 times price over an intel Core i7-920 CPU, the performance of TESLA C1060 is more than 18 times over intel CPU (see Table4.1, flops column). Note that the nVIDIA TESLA C1060 is the first generation that specialized for GPGPU.

GPGPU has already been widely used in scientific simulations especially for the computational fluid dynamics (Khajeh-Saeed & Perot, 2013; Li et al., 2013; Brodtkorb et al., 2012; Lobeiras et, al., 2013). However, to the best knowledge of author, there still merely is study about the GPU-accelerated tsunami simulator using unstructured mesh based on finite volume method. It is urgent to investigate the necessary conditions or treatment (such as the margin mesh and the data transmission) for GPU implementation on high-resolution tsunami simulation using unstructured mesh.

4.3 Basic concept of parallel computing using GPU

GPU is traditionally installed in mainboard (Fig4.1b) as an expansion card to assemble a Personal Computer (PC). The GPU is connected via the inner bus (PCI Express) to the CPU. Fig4.2 shows the internal structure diagram of nVIDIA GPU. As shown in Fig4.2a, there is several GB's (e.g. 4GB for a TESLA C1060) global memory on GPU, which can be used to store the variables **q** as well as other intermediate variables for computation according time evolution. But, the global memory is specified to the usage within GPU, it is not accessible to the rest part of PC. A GPU is composed of several Multi-Processors (MP). The MPs are arranged on GPU in pairs. And each

MP is consists of eight Streaming Processors (SP). Each SP can access the shared memory located in the MP. For example, the TESLA C1060 features 30 MP and totally 240 SP (Table4.1, Fig4.2b). Where, SP is the process unit in charge of computation that corresponding to a process core in ordinary CPU. That means, there are 240 process cores can be used parallel to achieve high-speed computation even by single C1060. Actually, with the GPU developing rapidly, the SPs are increased to more than 400 in single GPU, which have made GPU a compelling platform for parallel computation.

4.4 Program implementation

As shown in Fig4.2b, the shared memory in each MP has low storage capacity but high transfer speed, which can be accessed for the SPs in the same MP. On the other hand, the shared memory in different MP can't be accessed by each other. This suggests that, the "MP" is corresponds to a PC in the parallel computation using CPU cluster, and the shared memory in MP is correspond to the memory in PC.

For the actual program implementation of parallel computation using GPU, the computational domain is firstly partitioned into several small subdomains by some domain decomposition method. Fig4.3 has shown the concept graph of parallel computing using GPU. Although the actual tsunami simulation is conducted in two-dimensional space, for the sake of clarity, here just giving an one-dimensional example. As seen from Fig4.3.a, at first, the partitioned subdomains are allocated to MPs, and the necessary variables $\mathbf{q}^n$ are read from global memory to the shared memory in MPs. Then, the SPs within the same MP will calculate the variables in next time step $\mathbf{q}^{n+1}$ cross-referencing the variables stored in shared memory. Finally, the results $\mathbf{q}^{n+1}$ will be written back to the global memory.

As mentioned in Chapter2, the numerical flux $\mathbf{E}_j$ is calculated explicitly using former step's value $\mathbf{q}^n$, there is no need to reference the results $\mathbf{q}^{n+1}$ stored in different MP. Consequently, the computations conducted by MPs are entirely independent with each other, and fully parallelized computation without overhead of data transfer is realized.

Meanwhile, in consideration of that the numerical flux $\mathbf{E}_j$ is calculated through mesh surface, it is necessary to read the $\mathbf{q}^n$ of margin mesh (blue circle, Fig4.3b) additionally from global memory to current shared memory to calculate the numerical flux $\mathbf{E}_j$ (blue arrow, Fig4.3b) through the boundary surface between subdomains. This additional data transfer for the "margin mesh" can result in degradation of the computational speed. The actual margin mesh number in two-dimensional space is decided by the domain decomposition method that will be discussed in following section.

In conclusion, the parallel computation by GPU is achieved using MP as the processor unit. Therefore, in order to achieve efficient high-speed parallel computation, computational load should be distributed to each MP equally and the communication between MPs should be minimized. In other words, the mesh number should as evenly as possible among these subdomains as well as minimize the number of margin mesh. Besides, for the actual program implementation of GPU based parallel computation, special consideration such as data transfer and synchronous process and so on are required in writing programs. In this research, the CUDA™ library (nVIDIA CUDA home page) is used for the GPGPU code.

4.5 Domain decomposition

Considering the irregular connections of unstructured mesh, the effective domain decomposition method for unstructured mesh system is still in its exploration stage, and it is almost impossible to totally satisfy the above conditions so far. Therefore, in this research, two kinds of domain decomposition methods (recursive spectral bisection & multi-level spectral bisection) are used to investigate their effect on tsunami simulation.

4.5.1 Recursive spectral bisection (RSB)

Recursive spectral bisection (RSB) method (Pothen & Simon, 1990; Simon, 1991) is widely used to partition subdomains to achieve efficient large scale parallel computations which are based on unstructured mesh system.

The RSB method is considering the unstructured meshes are represented as undirected graphs using sparse-matrix data structures (Barnard & Simon, 1994). It is based on computing the eigenvalue of the Laplacian matrix $L(G)=(l_{ij})$ of the graph $G=(V,E)$. The Laplacian matrix is defined as

$$l_{ij}=\begin{cases}-1 & \textbf{if } (v_i,v_j)\in E\\ \deg(v_i) & \textbf{if } i=j\\ 0 & \textbf{otherwise}\end{cases} \tag{4.1}$$

Here $i,j=1,2,3,\ldots n$, $\deg(v_i)$ is the degree of the vertex v_i, defined as the number of edges connected with vertex v_i. A simple example for one-dimension is defined as

Graph G

Lapalacian Matrix $L(G)$

$$\begin{bmatrix}1 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1\\ & & -1 & 2\end{bmatrix}$$

The Laplacian matrix can be written as

$$x^tLx=-\sum_{(v_i,v_j)\in E}\left(x_{v_i}-x_{v_j}\right)^2 \tag{4.2}$$

From the definition of L, the eigenvalue is nonnegative

$$\lambda_1=0\le\lambda_2\le\lambda_3\ldots\le\lambda_n \tag{4.3}$$

The number of connected components in G is equal to the number of $\lambda_i=0$. Particularly, the second eigenvalue $\lambda_2\neq 0$ if and only if the graph G is connected. Therefore, the eigenvector $\vec{x}_2$ (also known as Fiedler vector) associated with the second eigenvalue λ_2 yield a weighting for the vertices. Differences in this weight give the distance information about the vertices of the graph G. The domain decomposition can be achieved by sorting the vertices according to this weight. The procedure of RSB method can be summarized as follow recursive steps:

a. Compute the Fiedler vector of the graph G;

b. Sort vertices according to Fiedler vector;

c. Assign half of the vertices to each subdomain.

4.5.2 Multi-level spectral bisection (MSB)

In order to give better partition results at reasonable computational cost, the multi-level *k-way* spectral bisection (MSB) method is proposed (Karypis & Kumar, 1998, 1999), which is simple but powerful for refining a *k-way* partitioning in the multilevel context.

The basic concept of multilevel partition method is not so complicate. The original graph $G=(V,E)$ is firstly coarsened down to a series of successively graphs G_1, G_2, G_3,…G_k. And then, the coarsened small graphs are bisected using spectral method. At last, the partitioned graphs are projected back to the original graph as well as been successively refined at each intermediate level. This refinement is seemed effective in decreasing the edge-cut (defined as Fig4.4, i.e. if the two connected vertices are located in different subdomain, there is one edge-cut).

The algorithm of multilevel graph bisection normally consists of three phases:

a. Coarsening phase. The weighted graph $G_0=(V_0,E_0)$ is transformed into a series of coarse graphs G_1, G_2, G_3,…G_k with with $|V_0|>|V_1|>|V_2|>\ldots>|V_k|$.

b. The V_k is partitioned into two parts by the bisection P_k of the graph $G_k=(V_k,E_k)$, and each of the partitioned part containing half of the vertices of G_0.

c. The partition P_k of the graph G_k is projected back to the G_{k-1} and refined at the same time, this process will repeat till and terminate at G_0.

4.6 Application to run-up tsunami in 2011 Tōhoku Earthquake

Although there is urgent need for run-up tsunami simulation in guarding human's life and property, it is hard to evaluate the simulation results because there is merely observation data was recorded. Fortunately, the run-up tsunami observed in Tone River right after the 2011 Tōhoku Earthquake was recorded. Based on this situation, a simulation of run-up tsunami in downstream Tone River is conducted using GPGPU, and the effectiveness of the GPGPU implementation is verified comparing with the observational data. In addition, the necessary conditions of GPGPU implementation on tsunami simulation with unstructured mesh are also investigated.

4.6.1 Computational conditions

The starting time is set as 14:52 11th March, a three hours tsunami propagation (till 17:52) is conducted using single GPU (TESLA C1060) and CPU (Core i7 920, 2.67GHz) respectively. In order to investigate the dependency of efficiency of GPGPU on mesh size, the simulations are carried out with the four kinds of meshes with different spatial size (mesh T1-T4 listed in Table4.2, also refer to the comparison between T1 and T4 in Fig4.5). The time increment is set to be 0.4s. The initial water wave distribution is calculated by Tohoku University model (Version1.0, Imamura et al., 2011). The water depth of Tone river estuary T.P.-0.49m is set as the base level, which is interpolated from the water level between 14:00 and 15:00, 11th March. The computational domain and topography please refer to Fig2.20.1 and Fig2.20.2. The boundary in deep-ocean is set as outflow boundary, and the rest is set to be reflective boundary. In order to trace the water front movement, emergence and submergence are determined on each mesh based on the water depth at each time step. Actually, if the water depth is lesser then 0.01 meter, the mesh is regarded as a land mesh, otherwise, the mesh is seemed to be submerged. Furthermore, on the basis of the land use type, the Manning's roughness coefficient is set to be 0.02 in ocean/river channel, and 0.04 in land and flood plain.

It should be noted that, although the SRNH has higher accuracy, it is more complicate than Roe scheme, which means doubled variables for computation and much more judgment functions are needed in Program implementation. On the other hand, the shared memory in MP is very small (16KB) that is considered not enough to support realizing efficient GPU implementation. That is the reason why just the Roe scheme is applied in GPU-based tsunami simulator in this research.

4.6.2 Comparing with observational data

In following discussion, if there is no special instruction, the result is calculated using mesh T4. Fig4.6 shows the tsunami propagation (0-100 minutes) in Tōhoku coastline. The results indicate that the first wave arrived in Tōhoku coastline is caused by the

initial wave. And, the second waves as well as subsequent waves are resulted from the reflecting wave around the coastline. Fig4.7 shows the highest water level along Ibaraki coastline. The red circles represent the calculation results of the developed model, and the blue rectangles represent observed results which are estimated from a trace survey. As shown in Fig4.7b, the calculation results consistent with the observational results.

Fig4.8 indicates the observation stations in Tone River (black circle). "KP" means the distance from river mouth. The tidal weir is located in 18.5KP, and the gates were closed right after the earthquake. Fig4.9 shows the contour figure of wave height in Tone River. As seen from these figures, the first wave arrived at Tone River mouth just 40 minutes after the earthquake. The run-up tsunami is reflected by the tidal weir, and the going-back first wave bumped up with second wave at 14KP could also be identified clearly. Fig4.10 shows the time series results of wave height in observation points which are located in Tone River. In which, the green rectangles represent observation data, the red line is the calculation results using mesh T4, and the blue line represent the calculation result using mesh T1. As seen from these figures, in the three observational points (-1.0KP, 0.76KP and 1.5KP) near the river mouth, the results of mesh T4 fit the observational data well, not only the wave height but also the arriving time. However, along with the run-up tsunami move toward the upper stream, the wave height decreased sharply. A possible reason is that, the Manning's roughness coefficient is set to be 0.02 no matter it is in ocean or in river channel. Besides, the Manning's roughness coefficient is likely different in land and flood plain either. This effect of Manning's roughness coefficient should be studied much more in detail in future. On the other hand, comparison between the results of mesh T1 and mesh T4 shows clearly that the wave height decayed distinctly on account of the numerical diffusion error brought on by the difference of mesh size. As seen in Taable4.2 (N column) that jus the mesh T4 satisfied the criterion in run-up tsunami simulation (N_{req}>200, refer to Table2.5) This reminds us again that, it is necessary to choose appropriate mesh size to get better evaluation of run-up Tsunami in river.

Generally speaking, the simulation results of GPGPU implementation is worth to be used in tsunami prevention if appropriate mesh is used.

4.6.3 Validity of GPGPU acceleration

4.6.3.1 Run-time comparison-without output

In this research, the same run-up tsunami simulations are conducted using CPU and GPU respectively using four kinds of meshes with different spatial size (mesh T1-T4). In view of the data output will have influence on the speed-up of parallel computing, the same simulations are also conducted without or with output (1times per 1minutes). Here, the speed-up of GPGPU implementation without output is discussed firstly. Table4.3 shows the elapsed time for three hour's tsunami propagation. In the cases of mesh T2, T3 and T4, the elapsed time using CPU is even much more than the actual tsunami propagation time (e.g. more than 16 hours are needed for CPU computation). Furthermore, although the CPU computation with mesh T1 can finish within three hours, the wave decayed sharply that the results cannot be used to estimate the wave height. By contrast, even though the mesh T4 is used, the elapsed time of GPU computation will just cost about 20 minutes. As shown in Table4.3, it is very clear that about 50 times speed-up is achieved for all mesh cases even using single TESLA C1060 GPU. This fact suggests that the GPGPU implementation is effective in accelerating the tsunami simulation.

4.6.3.2 Run-time comparison-with output (1time/1min)

Table4.3 also shown us the elapse time with data (wave height, velocity and so on) output 1 time per 1 minutes' tsunami propagation, in which the speed-up decreased to about 25 times in all of the four cases. Table4.4 have listed the computing time (A) and output time (B) as well as the ratio of output time on total run-time (=B/(A+B)). The data output cannot be conducted in parallel, which cause the ratio of output time increased dramatically (i.e. almost the same as the computing time) for the GPGPU implementation. In addition, for the data output in GPGPU implementation, the data

must be transferred trough the low speed inner bus that will also cost additional run-time. Therefore, the output time of GPGPU is considered to be increased much bigger than CPU. However, there is no obvious difference between CPU and GPGPU implementation on the output time. A possible reason is that, just as shown in Table4.2, the memory volume is so small that the data transfer time from GPU to CPU is negligible compared with output time. In conclusion, the data output time and data transfer time will in charge of the speed of whole GPGPU implementation, which indicate that the appropriate output time interval is crucial for the efficient GPGPU based parallel computation.

4.6.3.3 Influence of subdomain size

Here, we will discuss the influence of subdomain size on GPGPU based parallel computation. If there is no special declaration, the following discussion is based on the computation without data output. Considering that the number of mesh contained in subdomain (sub-mesh number) indicates the size of the subdomain. Therefore, the sub-mesh number is changed based on the RSB method to investigate its influences on tsunami simulation. The results are shown in Fig4.11, in which the horizontal axis shows average sub-mesh number as well as the vertical axis represent the run-time spent for 100 steps tsunami simulation. This figure clarified that along with average sub-mesh number increasing (subdomain become bigger), run- time will reduce and efficient high-speed tsunami simulation will be achieved. If compare the calculation times using 20 sub-mesh and that using 200 sub-mesh (pair endpoints of each curves), it is quite clear that the run-time difference are about 4 times in every cases. There are several possible reasons are considered to be adequate to explain this difference (see Section 4.6.3.4 & 4.6.3.5).

4.6.3.4 Influence of load balance

Unevenness of sub-mesh number is always significant accompany with the domain decomposition on unstructured mesh system, which means it is very hard to achieve load balance among subdomains. Therefore, the subdomain with lager sub-mesh

number will cost much more computational time compared with other subdomains with smaller sub-mesh number. As a result, the non-uniformity of sub-mesh number will deteriorate the global computational efficiency. If define the influence of unevenness on sub-mesh number as

$$\frac{\sigma}{N_{Avg}} = \frac{1}{N_{sub}} \sqrt{\sum \left(N_i - N_{Avg}\right)^2} \Big/ \frac{\sum N_i}{N_{sub}} \tag{4.4}$$

Where N_{sub} is the number of subdomain, N_{Avg} is the averaged sub-mesh number in one subdomain, N_i is the number of mesh in subdomain i, and σ is the standard deviation. The relationship between influence of unevenness and averaged number in subdomain is shown in Fig4.12. It is considered that the calculation load on single MP is in proportion to the average sub-mesh number, and the difference of calculation load among MPs is in proportion to the unevenness of sub-mesh number. Accordingly, seen from the influence of unevenness, the bigger the subdomain (larger sub-mesh number) is, the smaller the influence of unevenness will be. That is, following the usage of bigger subdomain, the inequality calculation load caused by unevenness of sub-mesh number becomes relatively smaller compared with the total calculation load. This is considered to be one possible reason to explain the results shown in Fig4.11 (also refer to Section 4.6.3.3) that bigger subdomain will accelerate the calculation.

4.6.3.5 Influence of Edge-cut (margin cell)

As mentioned above, the numerical flux is calculated through the boundary surface so that it is necessary to read the variables of additional one margin from the Global memory to the Shared memory of MP. Therefore, corresponding with the decrease of average sub-mesh number, the ratio of this additional “margin mesh” amount to sub-mesh becomes much bigger (see Fig4.13). It is considered to be another reason of speed drop discussed in Section4.6.3.3.

In summary, using bigger subdomain to a certain degree is the key-point of GPGPU application for high-speed calculation.

4.6.3.6 Influence of partition method

Fig4.14 shows the actual domain decomposition result by RSB and MRB method respectively, in which the color represents the serial number of subdomains. From the intuitive appreciation of these two figures, the MRB method is giving much better partition result than RSB method because the boundary is much smooth that the edge-cut is looks like restrained successfully. The quantitative evaluations are given in Fig4.15 and Table4.5. As seen from Table4.5, not only the influence of uneven, but also the edge-cut of MRB partition result is smaller than that of RSB method, which lead to an additional 20% speed-up by the application of MRB method.

4.7 Extension to Multi-GPU

4.7.1 Multi-GPU implementation

Considering the computational domain of actual tsunami simulation is so huge (e.g. sometimes the global tsunami simulation is also required) and the much more detailed topography data will be available in near future, which are likely to surpass the ability of single GPU in parallel computing. Thereupon, the developed tsunami simulator is also extended to multi-GPU.

A THIN node of Tsubame 2.0 is used in this study (see Table4.6), which is consist of three M2050 GPU and one CPU. As shown in Fig4.16, the computational domain is partitioned into several subdomains according to the number of GPU at first. And then, the partitioned subdomains will be partitioned into much smaller subdomains and distribute to each GPU. The following procedure is like the same as single GPU implementation. Here comes the problem that data transfer between GPUs (Fig4.16) is inevitable. However, as discussed above (refer to Section4.4), data transfer between GPUs will decrease the efficiency of GPGPU implementation.

Overwrap technique is effective in solving the above problem. The strategy of over wrap is shown in Fig4.17. In which, the meshes are labeled as two kinds: inner mesh and boundary mesh (red meshes), and the computation of inner mesh is separated from

the boundary mesh. Under the circumstances, from t_n-t_{n+1}, if the computational load is big enough (inner mesh number is considerably larger than boundary mesh number), the data transfer time of boundary meshes will be overwrapped.

4.7.2 Run-time comparison

Table4.7 has summarized the run-time of multi-GPU implementation. For the case without overwrap, although three GPUs are used, the speed-up are still liked the same as single GPU. On the other hand, with the power of overwrap technique, the computing speed is improved a lot that 120 times speed-up is achieved using three GPUs. Table4.8 has listed the computing time and data transfer time as well as the ratio of data transfer time on total run-time. It is very clear that the data transfer time increased dramatically from single GPU to multi-GPU implementation. For example, the ratio of transfer time on total run-time increased from 0% of single GPU to 54% of two GPUs. It is considered to be the reason of the problem occurred in multi-GPU implementation. And, we can get a conclusion that the overwrap is effective and essential in multi-GPU implementation.

4.8 Conclusions

In this chapter, a GPGPU based tsunami simulator is developed and is used to conduct the simulation of run-up tsunami (2011) in Tone River with several meshes of different spatial size. As a result of comparison with field observation data, it is found that the solver can restore run-up tsunami well. Furthermore, the acceleration of GPGPU is verified that 50 times speed-up is achieved by using single GPU (TESLA C1060). Meanwhile, the influence of output interval, subdomain size, load balance, edge-cut as well as the decomposition method are evaluated quantitatively. In detail, the bigger the subdomain is, the higher speed-up it will be. And the usage of MRB method will bring 20% additional speed-up comparing with RSB method. At last, the solver is extended to multi-GPUs, and 120 times speed-up is achieved by using three GPUs (Tsubame 2.0, TESLA C2050).

REFERENCE

Khajeh-Saeed, A., Perot, J. B.: Direct numerical simulation of turbulence using GPU accelerated supercomputers, *J. Computational Physicis*, Vol. 235, pp. 241-257, 2013.

Li, X., Zhang, Y., Wang, X. W., Ge, W.: GPU-based numerical simulation of multi-phase flow in porous media using multiple-relaxation-time lattice Boltzmann method, *Chemical Engineering Science*, Vol. 102, pp. 209-219, 2013.

Brodtkorb, A., Sætra, M., Altinakar, M.: Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation, *Computers & Fluids*, Vol.55, pp. 1-12, 2012.

Lobeiras, J., Viñas, M., Amor, M., Fraguela, B. B., García, A. J., Castro, M. J.: Parallelization of shallow water simulations on current multi-threaded systems, *International Journal of High Performance Computing Applications*, Vol. 27(4), pp. 493-512, 2013.

Pothen, A., Simon, H. D., and Liou, K. P.: Partitioning sparse matrices with eigenvectors of graphs, *SIAM Journal of Matrix Analysis and Applications*, Vol. 11, pp. 430-452, 1990.

Simon, H. D.: Partitioning of unstructured problems for parallel processing, *Computer Systems in Engineering*, Vol. 2, pp. 135-148, 1991.

Barnard, S. T., and Simon, H. D.: A Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems, *Concurrency: Practice and Experience*, Vol. 6, pp. 101-117, 1994.

Karypis, G., and Kumar V.: Multilevel k-way partitioning scheme for irregular graphs, *Journal of Parallel and Distributed Computing*, Vol. 48(1), pp. 96–129, 1998.

Karypis, G., and Kumar V.: A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM Journal on Scientific Computing*, Vol. 20(1), pp. 359–392, 1999.

今村文彦，越村俊一，大家隆行，馬渕幸雄，村嶋陽一(2011): 東北地方太平洋沖地震を対象とした津波シミュレーションの実施 東北大学モデル(Version1.0), 東北大学大学院工学研究科 付属災害制御研究センター.

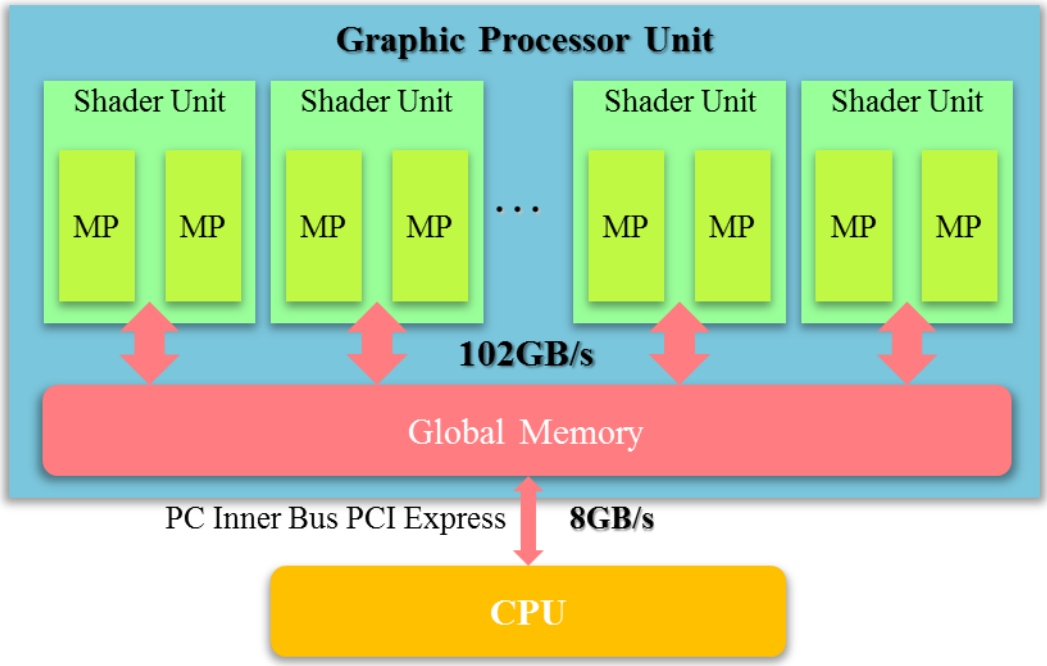
(a)

(b)

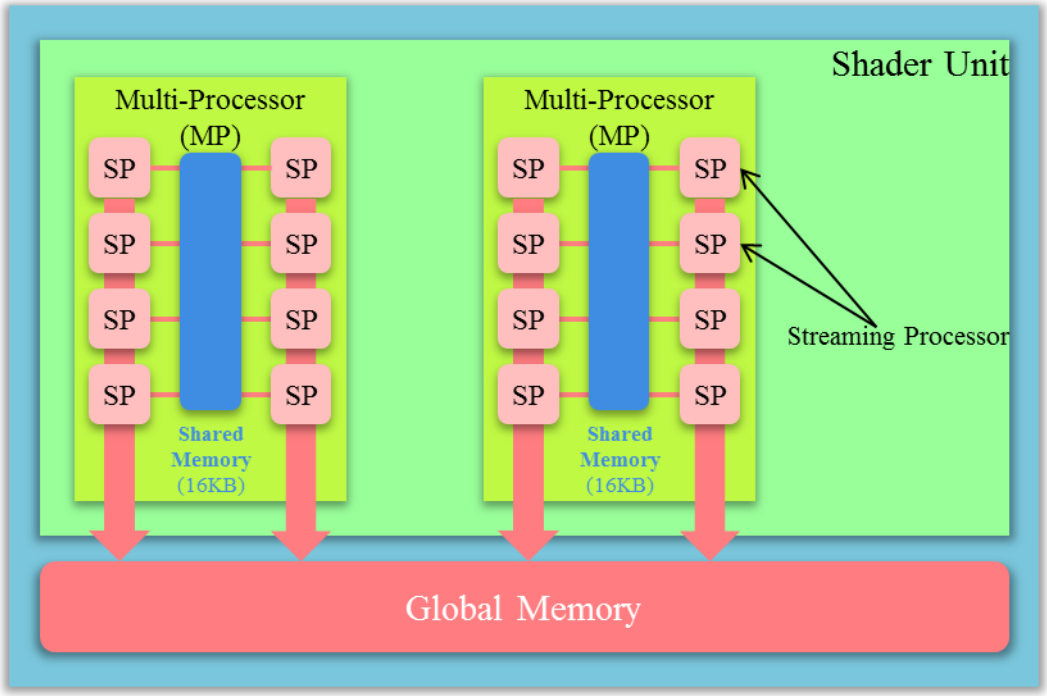
Fig4.1 GPU (a. Nvidia TESLA C1060; b. Three GPUs installed in mainboard)

Table4.1 Comparison between CPU and GPU

Specification	CPU (Core2 i7 920)	GPU (TESLA C1060)
Processor clock	2670MHz	1296MHz
Number of processor cores	4	240 (SP)
Memory size	Up to 32 GB	4 GB
Memory bandwidth	25.6 GB/s	102.4 GB/s
flops	51.2 Gflops	933 Gflops
Price(JPY)	40000	200000

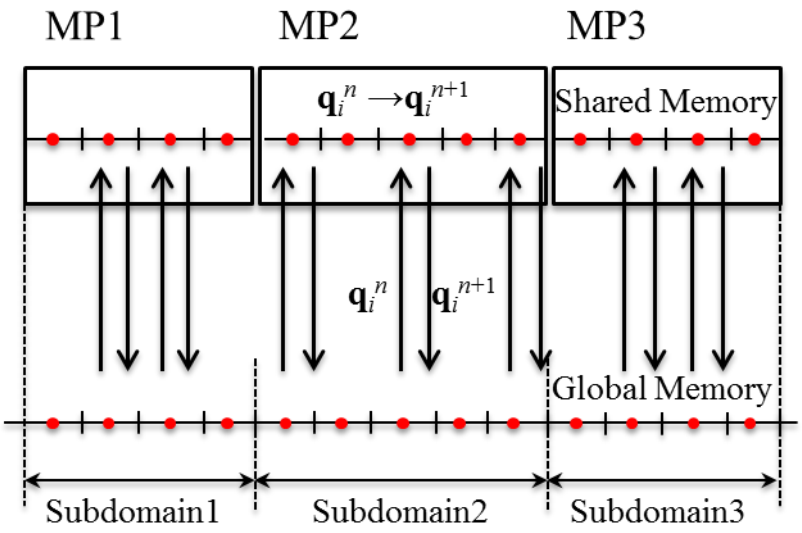


(a) Internal structure of GPU

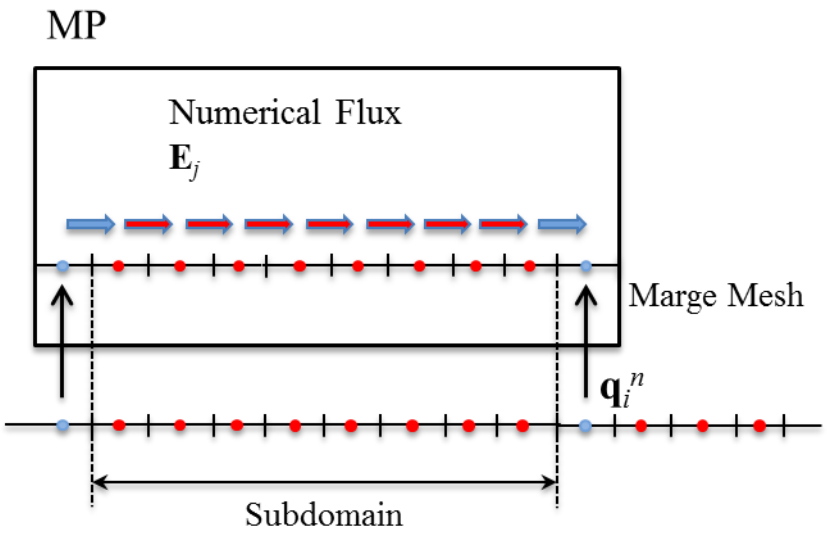


(b) Structure of shader unit

Fig4.2 GPU internal structure diagram



(a)



(b)

Fig4.3 Concept graph of GPU program implementation

Fig4.4 Definition of Edge-cut

Table4.2 Meshes used for run-up tsunami simulation

Mesh		T1	T2	T3	T4
Mesh number		34,060	139,119	266,194	452,901
Mesh size	Ocean	75km	40km	75km	40km
	Tone river	150m	30m	25m	15m
N (mesh number per wave length in Tone river)		31	153	184	307
Memory volume [MB]		5	24	46	79

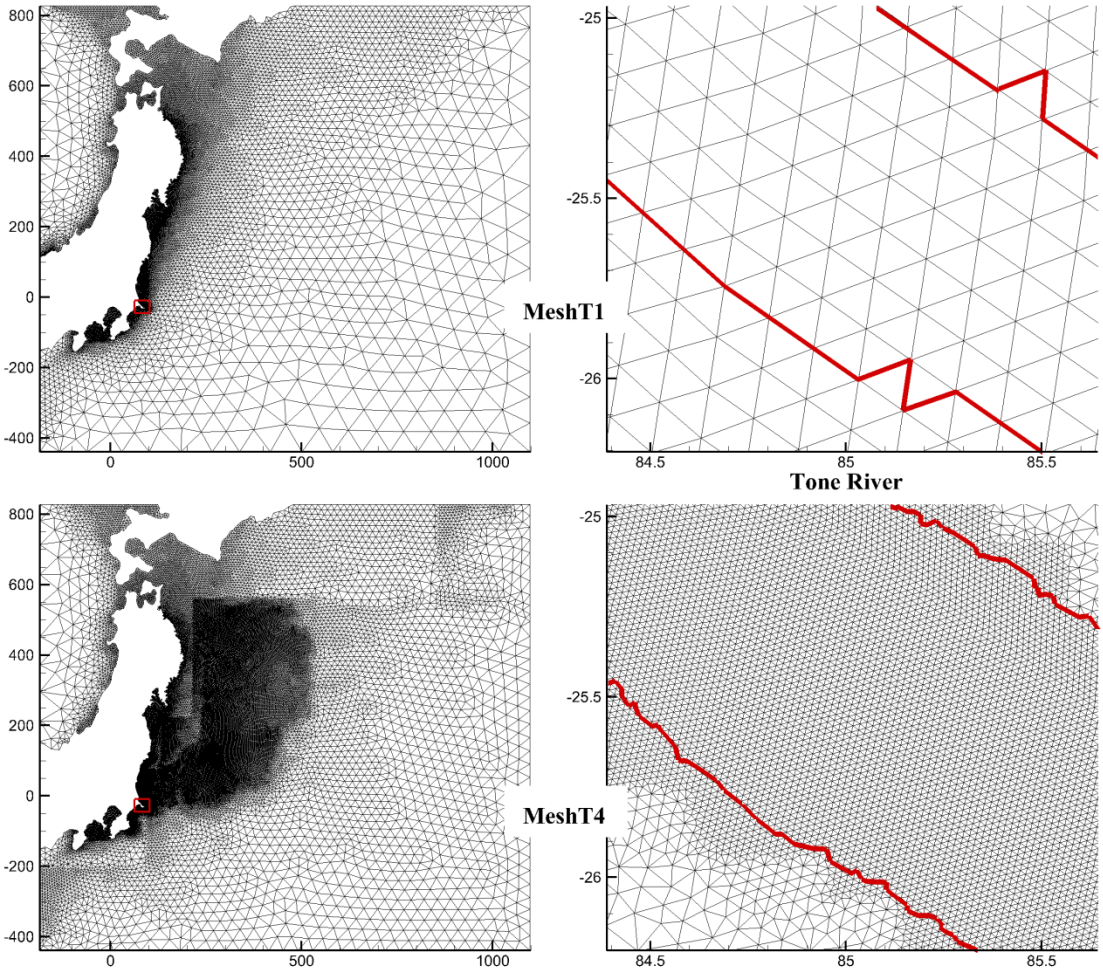


Fig4.5 Comparison between mesh T1 and mesh T4

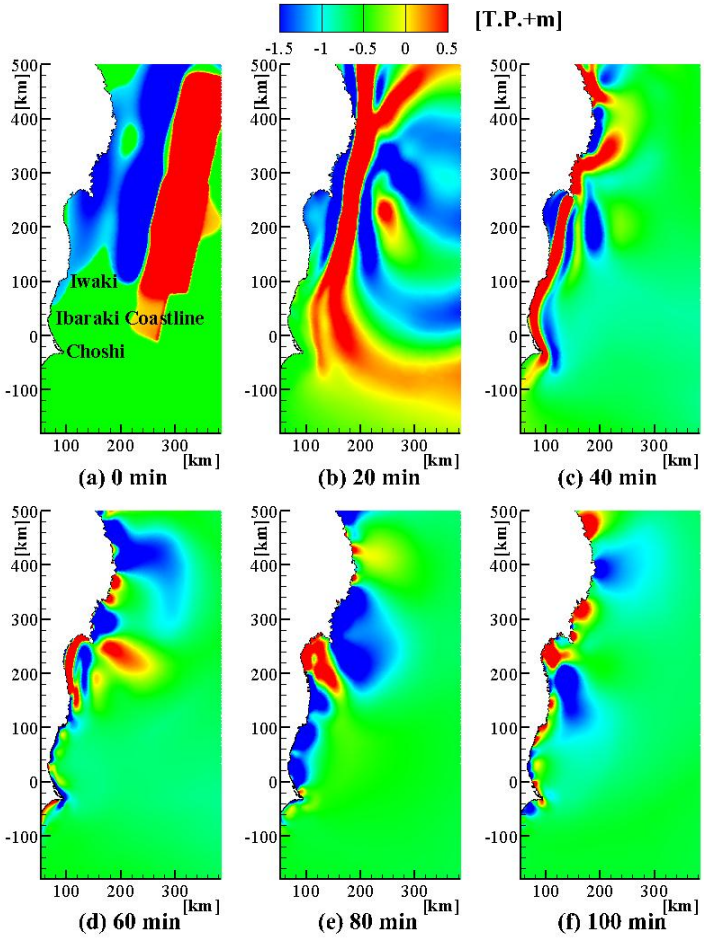


Fig4.6 Tsunami propagation

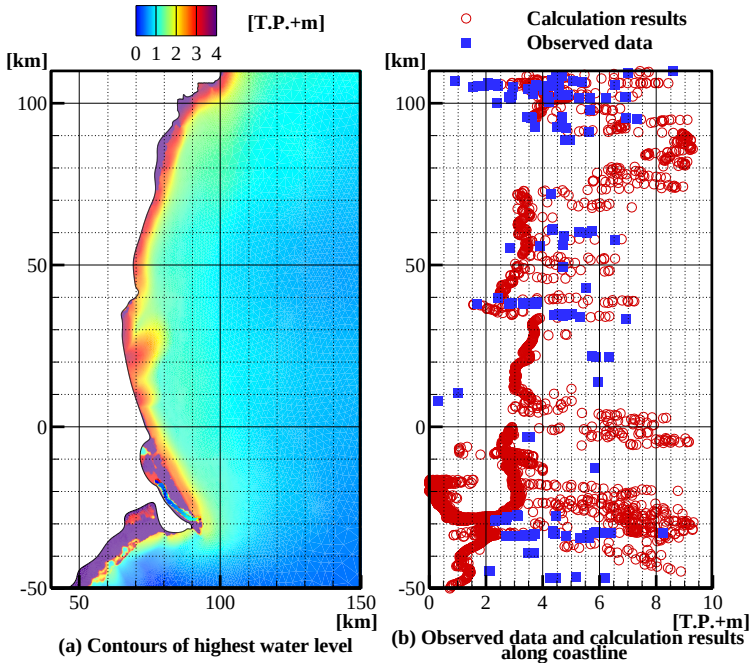


Fig4.7 Highest water level in Ibaraki coast

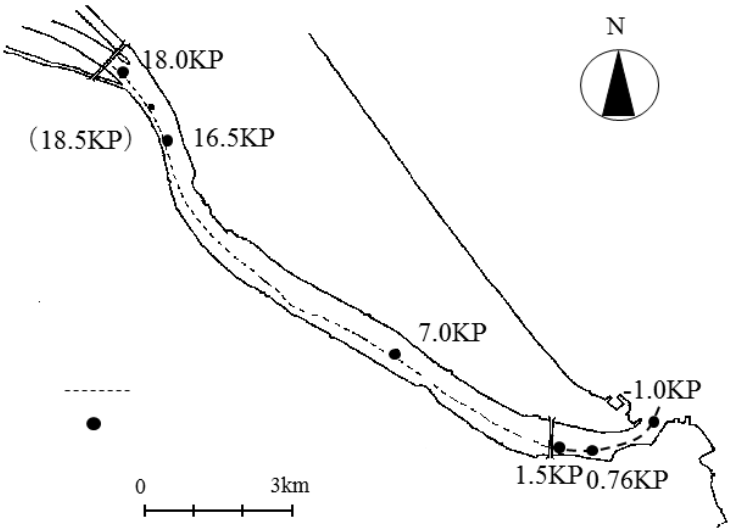


Fig4.8 Location of observational point in Tone river

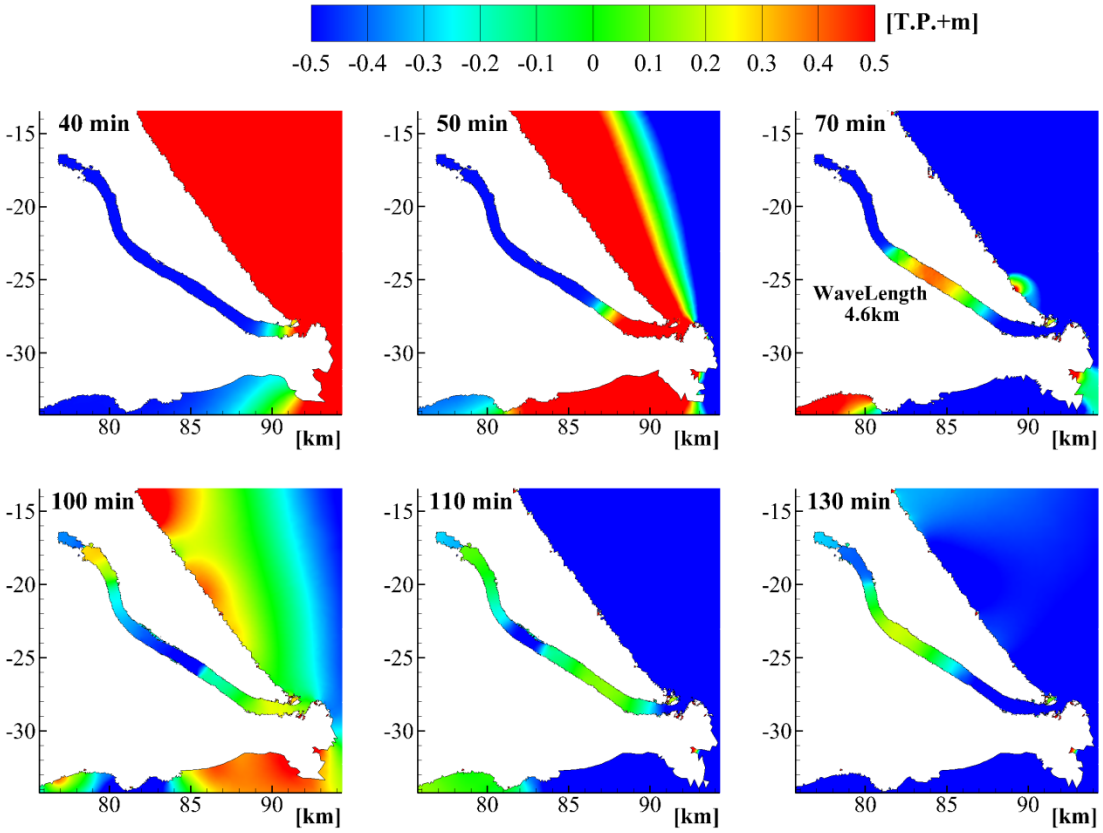


Fig4.9 Run-up Tsunami in Tone river

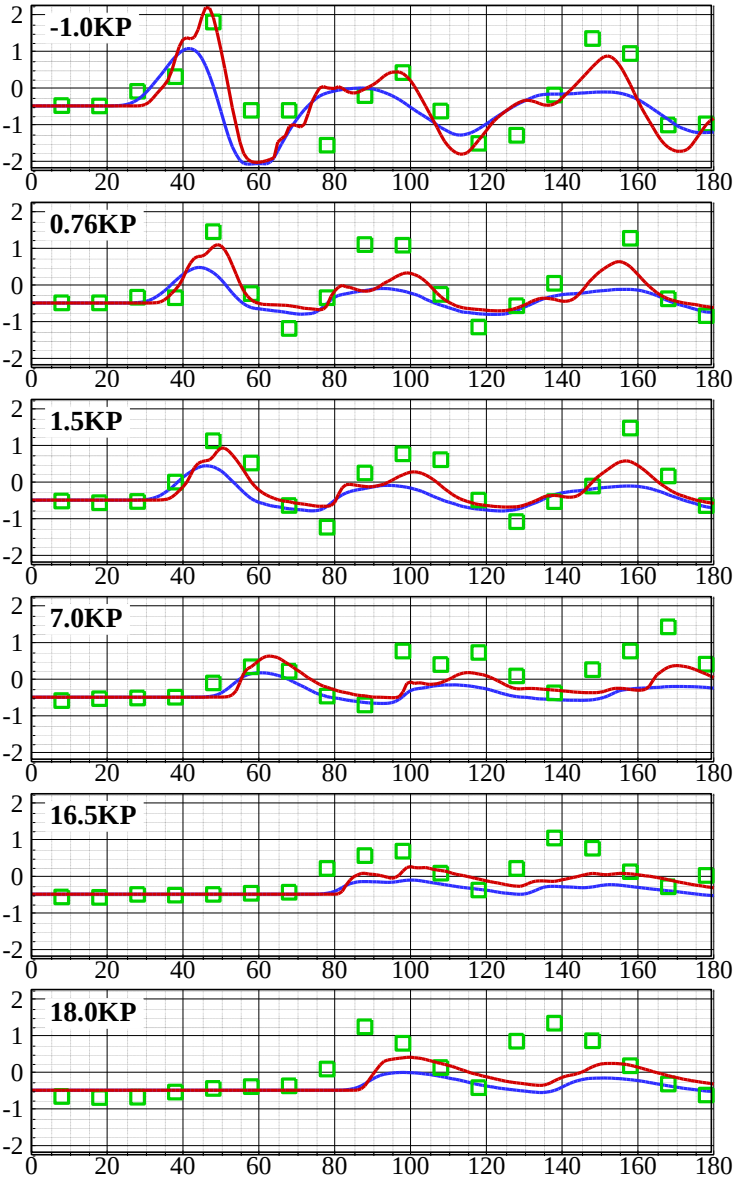


Fig4.10 Comparison between observational data and computational results

(Green square: Observational data; Blue line: Results of Mesh1; Red line: Results of mesh4)

Table4.3 Run-time comparison

Mesh		T1		T2		T3		T4	
Mesh number		34,060		139,119		266,194		452,901	
Subdomain number		256		1,024		2,048		4,096	
Output [1time/1min]		No	Output	No	Output	No	Output	No	Output
Run time [min]	GPU[A]	1.3	(2.5)	5.2	(9.7)	10.9	(20)	21.4	(37.1)
	CPU[B]	71.2	(72.3)	246.7	(251.6)	535.4	(544.6)	964.2	(979.8)
Speed-up[B/A]		53	(28)	47	(25)	49	(27)	45	(26)

Table4.4 Run-time for computation and data output

Mesh		T1		T2		T3		T4	
Computing system		GPGPU	CPU	GPGPU	CPU	GPGPU	CPU	GPGPU	CPU
Run-time [min]	Computation(A)	1.3	71.2	5.2	246.7	10.9	535.4	21.4	964.2
	Outpu(B)	1.2	1.1	4.5	4.9	9.1	9.2	15.7	15.6
Ratio of output (B/(A+B))		48%	1.5%	46%	1.9%	46%	1.7%	42%	1.6%

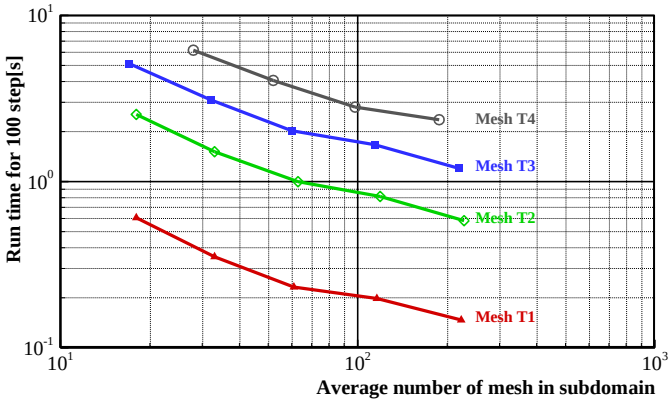


Fig4.11 Influence of subdomain size (RSB)

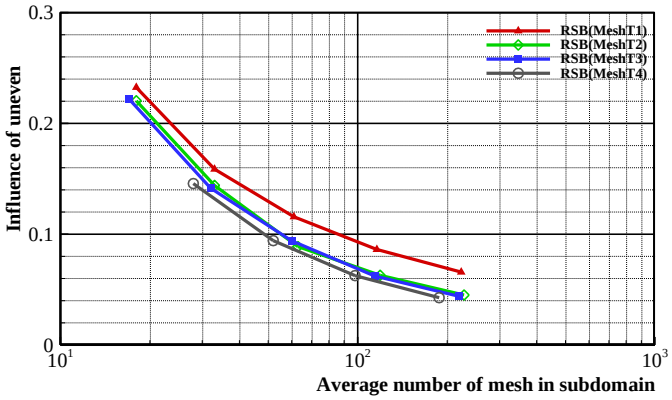


Fig4.12 Influence of unevenness of mesh number (load balance)

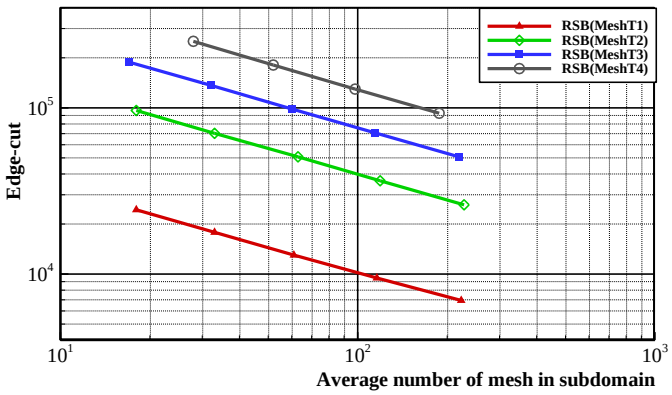
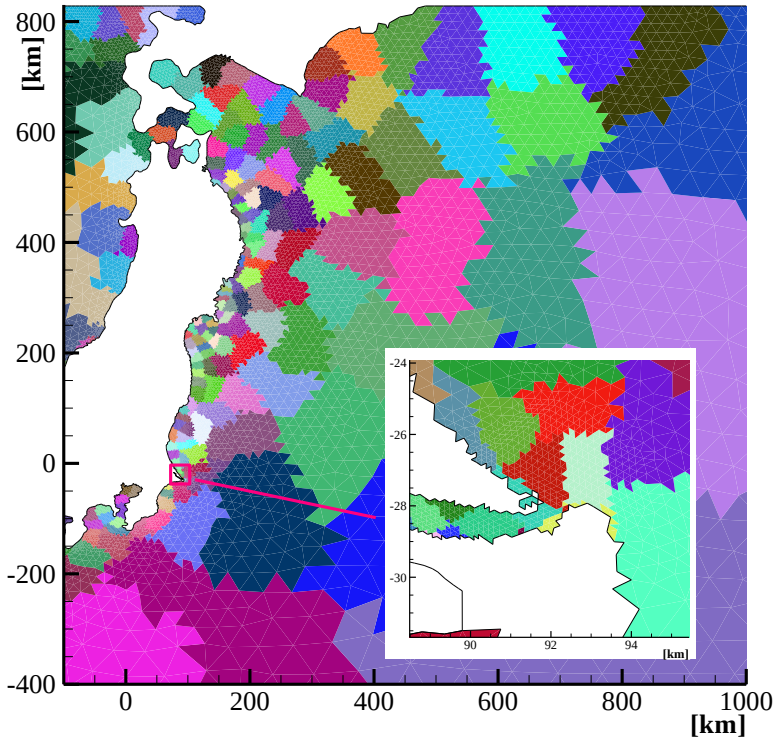
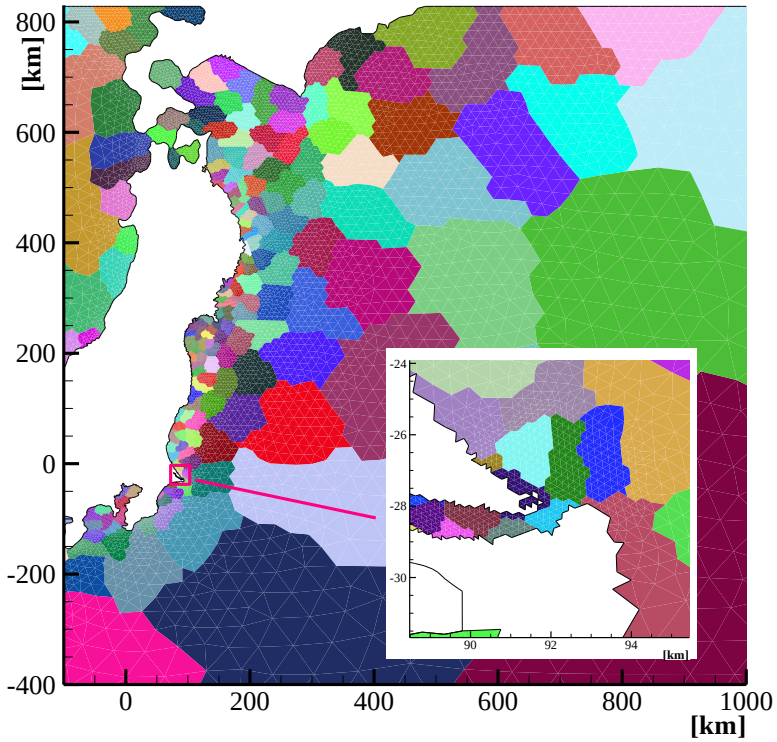


Fig4.13 Influence of edge-cut



(a) Domain decomposition result of recursive spectral bisection method (RSB)



(b) Domain decomposition result of multilevel recursive bisection method (MRB)

Fig 4.14 Domain decomposition results

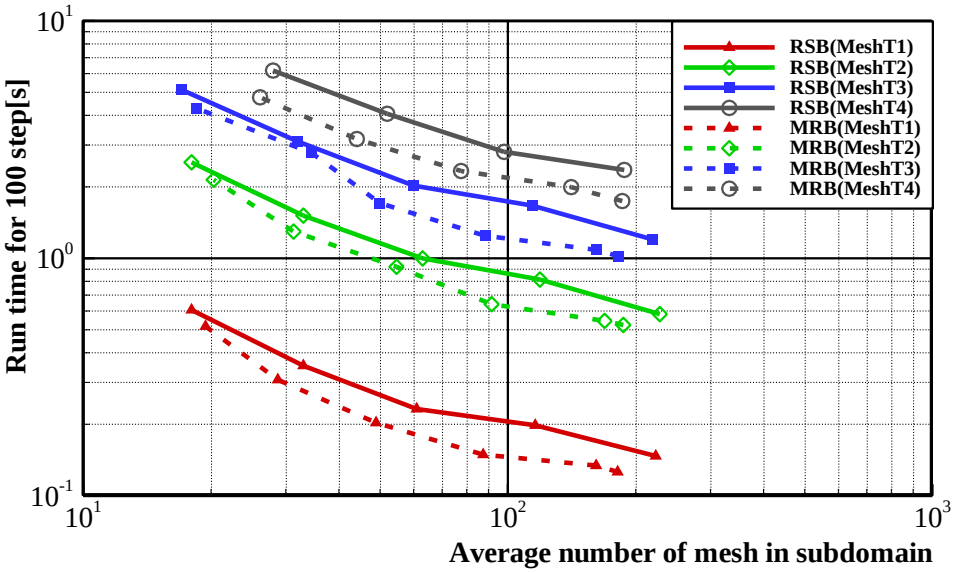


Fig4.15 Influence of subdomain size

Table4.5 Comparison of RSB and MRB

Mesh T4 Number of subdomain: 4096

Decomposition method	Uneven factor	Edge-cut	Run time for 100 steps [s]
RSB	6.0×10^{-1}	92829	2.357
MRB	1.5×10^{-1}	64404	1.995

Table4.6 Specification of Tsubame 2.0 (THIN node)

GPU	MP	SP	Memory	Memory bandwidth	flops (single)
M2050 (3GPU/node)	14	448	3GB	148GB/s	1.03 Tflops

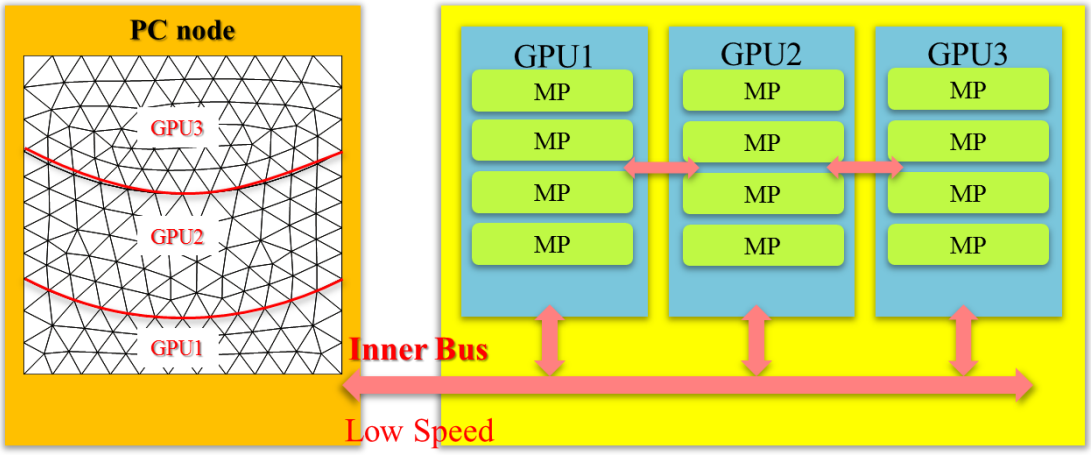


Fig4.16 Internal structure diagram of Tsubame2.0 (THIN node)

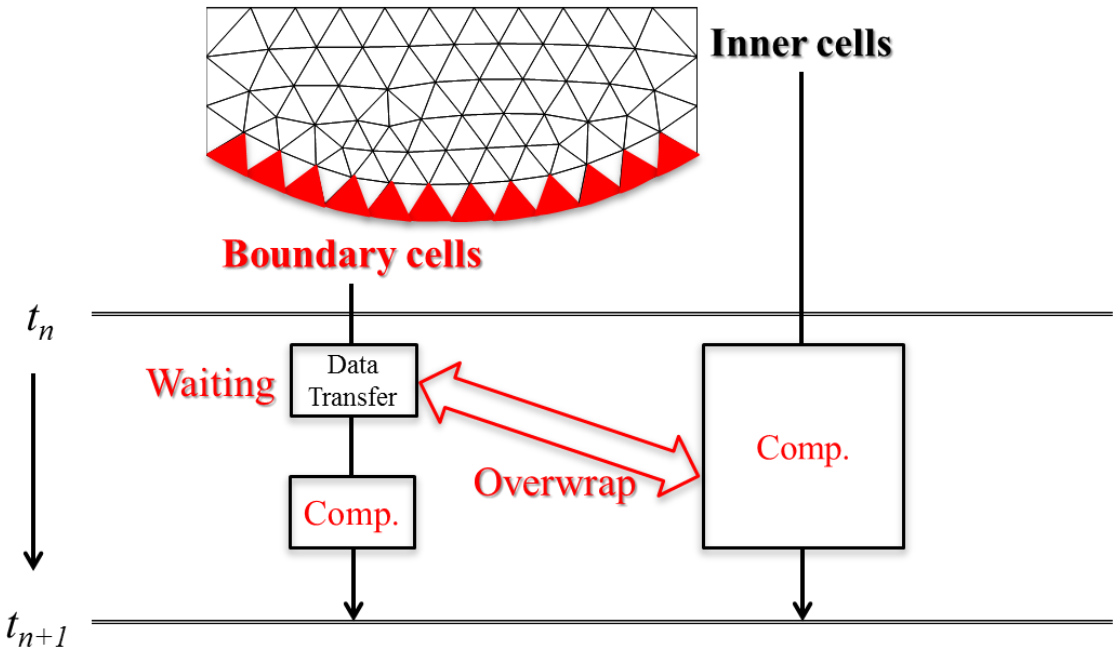


Fig4.17 Multi-GPU implementation with overwrap

Table4.7 Run-time comparison of Multi-GPU

Computational system		No overwrap		With Overwrap	
		Time	Speed-up	Time	Speed-up
Run time for 30min's propagation [min]	1CPU	89.23	/	89.23	/
	1GPU	1.17	76	1.19	75
	2GPUs	1.24	72	0.69	129
	3GPUs	1.07	83	0.72	123

Table4.8 Data transfer time of Multi-GPU

Computing system	1CPU	1GPU	2 GPUs	3 GPUs
Computation time[min]	89.23	$\approx 1.17\times10^{0}$	5.71×10^{-1}	3.64×10^{-1}
Transfer time(B) [min]	0	2.73×10^{-6}	6.69×10^{-1}	7.10×10^{-1}
Total time (C) [min]	89.23	1.17×10^{0}	1.24×10^{0}	1.07×10^{0}
Ratio of transfer time(B/C)	/	≈0%	54%	66%
Speed-up	/	76	72	83

Chapter5 Summary and future work

5.1 Summary

A rapid and high-resolution tsunami simulator based on finite volume method with unstructured mesh system is developed in this research. Three kinds of idealized numerical experiment are conducted with several meshes of different spatial size to verify the effectiveness of the developed tsunami simulator, as well as to investigate necessary conditions for high-resolution tsunami simulation. Quantitative error analysis is also conducted and the required mesh number "*N*" per wavelength is investigated and summarized for deep-ocean tsunami movement and run-up tsunami respectively. These criterions are applied to guide the simulations of tsunami in Tōhoku earthquake (2011).

A new approach of adaptive mesh refinement (AMR) is proposed based on the unstructured triangular mesh system. In order to verify the abilities of proposed new AMR strategy in capturing tsunami propagation either on deep-ocean or slopping beach, two kinds of numerical experiment is conducted respectively. In addition, the proposed AMR strategy is also applied to the simulation of tsunami in Tōhoku earthquake (2011).

A GPGPU based tsunami simulator is developed and used to conduct the simulation of run-up tsunami in Tone River with several meshes of different size. Meanwhile, the necessary conditions of GPGPU implementation for rapid and high-resolution tsunami simulation are investigated. At last, the solver is extended to multi-GPUs.

5.2 Conclusion

The developed tsunami simulator is effective in get high-resolution result. Not only the numerical scheme, but also the mesh size as well as propagate distance will affect the resolution of tsunami simulation.

1-1 In order to get high-resolution result on deep-ocean tsunami simulation, at least 15 meshes is required for SRNH scheme, and 30 meshes for Roe scheme; For the run-up tsunami simulation, at least 200 meshes is required both for SRNH and Roe scheme.

1-2 As the propagate distance increasing, the numerical error will also increase.

The proposed AMR strategy is effective in capturing tsunami movement as well as reducing computational load.

2.1 For idealized numerical experiments, if the minimum mesh size of AMR implementation is equal to the mesh used for SRNH scheme, AMR could get almost the same result as SRNH scheme.

2.2 For the simulation of tsunami in 2011 Tōhoku Earthquake, the computational results are improved by using the AMR method especially in the NM and CM observational station.

2.3 For idealize numerical experiment, the proposed AMR method could reduce the computational load to 1/10 at best; For real-tsunami simulation, the computational load just could be reduced to 1/3 at best.

The developed GPGPU based tsunami simulator is effective in achieving rapid and high-resolution tsunami simulation, especially for the case that the spatial scale changing dramatically. Output interval, subdomain size, load balance, edge-cut as well as the decomposition method will affect the acceleration of GPGPU implementation.

3.1 The bigger the subdomain is, the higher efficient will be achieved.

3.2 The usage of MRB method will bring 20% additional speed-up comparing with RSB method.

3.3 A 50 times speed-up is achieved using single GPU (TESLA C1060).

3.4 A 120 times speed-up is achieved using three GPUs (Tsubame 2.0, TESLA C2050).

5.3 Future work

In the present study, the initial wave distribution is calculated by Tohoku University model (Imamura et al., 2011), which is distributed right after the Tōhoku Earthquake. Therefore, it should be checked and verified that if there are any changes.

As discussed in Chapter1, the tsunami movement is sensitive to the changed of sea bed especially near shore. However, the resolution of the topography data used in this research is not so high. For example, the original topography mesh size in north part of the computational domain is about 2000 meters big, which is seemed to be a possible reason that the computational wave height decreased sharply in CI and SI.

Although the effect of ΔH_C is evaluated in present study, it is based on experience or trial computation. A more simpler and clear criterion should be investigated to achieve efficient AMR implementation in tsunami simulation.

For the multi-GPU implementation, considering that the 75 times speed-up can be achieved even use single TESLA C2050, there still have potential to get much higher acceleration using three GPUs.

Acknowledgements

First and foremost, I would like to express my deepest thanks to my supervisor, Professor Nakamura. You guided me on my academic and daily life in Japan. Plenty of discussions give me the original ideas of this research. Your precious advice and encouragement let this research progress successfully. Your precise attitude for the scientific research is worth learning in all my life.

My vice supervisor, Professor Ishikawa, also gave his kind support and advice on this study. Your broad expert knowledge of science helps me establish my knowledge framework. Your strong thirsts for knowledge of language as well as international communication are worthy to be revered.

I would like to thank Dr. Akoh for his helpful guidance on the model study and data processing. Thanks for your experiential knowledge on model establishment, I am able to avoid detours on my study.

I also would like to thank Professor Kinouchi on his sincerely care on my study life as well as the valuable comments and advices on my research.

Moreover, the lab members and also my good friends gave their strong support on my scientific life and brought me uncountable happiness. Dr. Xu and Mr. He as well as Mr. Ishii gave their valuable advices on my study. Dr. Minoura, Dr. Wang Z. J., Dr. Liu C., Dr. Liu T., Ms. Li, Ms. Lin, Mr. Gao, Mr. Yamazaki, Mr. Yamaguchi, Mr. Fukuda, Mrs. Xin, Ms. Zhang and Ms. Wang M. Y., thank you for your perpetual support and friendship during my Tokyo life. I will never forget everybody's happy face and the exciting moment you gave me.

Last but not least, many thanks to my lovely parents and my grandparents. Your endless loves give me inspiration and motivation to start and to continue this journal. I am deeply grateful for my fiancée, Ms. Zou's support. The happiness life together with you is the best memory of my life.