TSUNAMI MODELLING MANUAL

(TUNAMI model)

by

Dr.Fumihiko Imamura
Prof. of Tsunami Engineering
School of Civil Engineering, Asian Inst. Tech .(1993-1995) and
Disaster Control Research Center, Tohoku University., Sendai, Japan

Dr. Ahmet Cevdet Yalciner
Assoc. Prof. in Middle East Technical University, Civil Enigneering Department, Ocean Engineering Research Center, Ankara Turkey

Res. Assist. Gulizar Ozyurt
Research Assistant in Middle East Technical University, Civil Enigneering Department, Ocean Engineering Research Center, Ankara Turkey

Prepared in JUNE 1995 by Imamura
Revised in JUNE 2005 by Imamura
Revised in AUGUST 2005 by Yalciner and Ozyurt
Revised in APRIL 2006 by Imamura, Yalciner and Ozyurt

Revision due on APRIL 2006

INTRODUCTION

This manual is based on “TSUNAMI NUMERICAL SIMULATION with the staggered leap-frog scheme (Numerical code of TUNAMI-N1)” of Dr. Fumihiko Imamura, Prof. of Tsunami Engineering School of Civil Engineering, Asian Inst. Tech. and Disaster Control Research Center, Tohoku University prepared in June, 1995 for TIME project. The TIME (Tsunami Inundation Modeling Exchange) started in 1991 as a joint effort of IUGG and IOC/UNESCO during IDNDR. The Disaster Control Research Center (DCRC), Tohoku University, Japan has been acting as the center of TIME, to transfer numerical technique of tsunami simulation to the countries which suffered or will suffer tsunami hazards. Fifteen institutions of twelve countries obtained the computer programs and manuals developed and prepared by DCRC through mails or by training directly from DCRC. Four institutions of four countries obtained the technique through Mr. Ortiz, one of the trainees of the TIME project. As of 2003, the TUNAMI code was transferred to nineteen institutions of fifteen countries.

In 1997, the manual was published by UNESCO as IOC Manuals and Guides No.30 “IUGG/IOC TIME PROJECT: NUMERICAL METHOD OF TSUNAMI SIMULATION WITH THE LEAP-FROG SCHEME”.

The TUNAMI code consists of;

TUNAMI-N1 (Tohoku University’s Numerical Analysis Model for Investigation of Near-field tsunamis, No.1) (linear theory with constant grids),

TUNAMI-N2 (linear theory in deep sea, shallow-water theory in shallow sea and runup on land with constant grids),

TUNAMI-N3 (linear theory with varying grids),

TUNAMI-F1 (linear theory for propagation in the ocean in the spherical co-ordinates) and

TUNAMI-F2 (linear theory for propagation in the ocean and coastal waters).

The mentioned manual is revised and extended by Dr. Fumihiko Imamuıra tohoku Univesity, Japan, Dr. Ahmet Cevdet Yalciner and Research Assistant Civil Eng. Gulizar Ozyurt, Middle East Technical University, Turkey by including the manual for TUNAMI-N2, processing bathymetry file, input procedures and a glossary.

Chapters 1 to 7 assess the governing equations, numerical scheme, errors, initial and boundary conditions. Chapters 8 to 10 assess the input files and data necessary for the software. Chapter 11 assesses the TUNAMI-N2 program in detail. Chapter 12 assesses the interpretation of the output files. Chapter 13 includes some sample and real-time examples. The glossary is an important source for the whole document including earthquake, tsunami and wave terminology.

CONTENTS

1. GOVERNING EQUATIONS

1.1 Shallow Water Theory

Tsunamis which are mainly generated by the movement of sea bottom due to earthquakes belong to long waves. In the theory of such waves, the vertical acceleration of water particles are negligible compared to the gravitational acceleration except for an oceanic propagation of tsunami [Kajiura, 1963]. Consequently, the vertical motion of water particles has no effect on the pressure distribution. It is a good approximation that the pressure is hydrostatic.

Based upon these approximations and neglecting the vertical acceleration, the equations of mass conservation and momentum in the three dimensional problem (see Fig 1.1) are expressed by the following theory:

$$
\begin{aligned}
&\frac{\partial \eta}{\partial t}+\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \\
&\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}+\frac{1}{\rho}\frac{\partial p}{\partial x}+\frac{1}{\rho}\left(\frac{\partial \tau_{xx}}{\partial x}+\frac{\partial \tau_{xy}}{\partial y}+\frac{\partial \tau_{xz}}{\partial z}\right)=0 \\
&\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}+\frac{1}{\rho}\frac{\partial p}{\partial y}+\frac{1}{\rho}\left(\frac{\partial \tau_{xy}}{\partial x}+\frac{\partial \tau_{yy}}{\partial y}+\frac{\partial \tau_{yz}}{\partial z}\right)=0 \\
&\qquad g+\frac{1}{\rho}\frac{\partial \rho}{\partial z}=0
\end{aligned}
\tag{1.1}
$$

where *x* and *y* are horizontal axes, *z* the vertical axis, *t* time, *h* the still water depth, *η* the vertical displacement of water surface above the still water surface, *u*, *v* and *w* are water particle velocities in the *x*, *y* and *z* directions, *g* the gravitational acceleration, and τ_{ij} the normal or tangential shear stress in the *i* direction on the *j* normal plane.

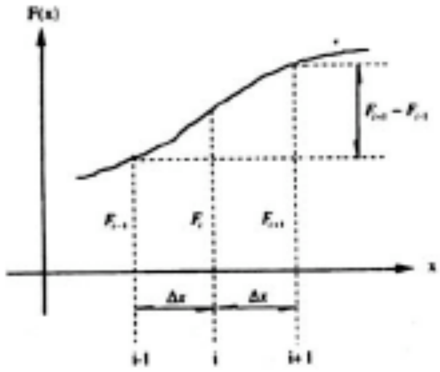


Figure 1-1. Central difference

The equation of momentum in the *z*-direction with the dynamic condition at a surface that *p* = 0 yields the hydrostatic pressure $p = \rho g(\eta - z)$.

We can solve any wave propagation problems by using the governing equations - Eq.(1.1) with boundary conditions. The dynamic and kinetic conditions at surface and bottom are given as follows:

$$p = 0 \qquad \text{at } z = \eta \qquad (1.2)$$

$$w = \frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y} \qquad \text{at } z = \eta \qquad (1.3)$$

$$w = -u\frac{\partial h}{\partial x} - v\frac{\partial h}{\partial y} \qquad \text{at } z = -h \qquad (1.4)$$

Now, let us integrate Eq. (1.1) from the bottom to the surface using by Liebnitz rule. For example, the first term of the momentum equation in the *x*-direction is rewritten as follows:

$$\int_{-h}^{\eta} \frac{\partial u}{\partial t} dz = \frac{\partial}{\partial t}\int_{-h}^{\eta} u dz - u\left.\frac{\partial \eta}{\partial t}\right|_{z=\eta} + u\left.\frac{\partial(-h)}{\partial t}\right|_{z=-h}$$

With dynamic and kinetic condition - Eqs. (1.2)-(1.4), we finally obtain the following two dimensional equations (this is called the shallow water theory) :

$$\frac{\partial \eta}{\partial t} + \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} = 0$$

$$\frac{\partial M}{\partial t}+\frac{\partial}{\partial x}\left(\frac{M^2}{D}\right)+\frac{\partial}{\partial y}\left(\frac{MN}{D}\right)+gD\frac{\partial \eta}{\partial x}+\frac{\tau_x}{\rho}=A\left(\frac{\partial^2 M}{\partial x^2}+\frac{\partial^2 M}{\partial y^2}\right)$$

$$\frac{\partial N}{\partial t}+\frac{\partial}{\partial x}\left(\frac{MN}{D}\right)+\frac{\partial}{\partial y}\left(\frac{N^2}{D}\right)+gD\frac{\partial \eta}{\partial y}+\frac{\tau_y}{\rho}=A\left(\frac{\partial^2 N}{\partial x^2}+\frac{\partial^2 N}{\partial y^2}\right) \tag{1.5}$$

where *D* is the total water depth given by *h*+*η*, *τ_x* and *τ_y* the bottom frictions in the *x*- and *y*-directions, *A* the horizontal eddy viscosity which is assumed to be constant in space, the shear stress on a surface wave is neglected. *M* and *N* are the discharge fluxes in the *x*- and *y*- directions which are given by,

$$M=\int_{-h}^{\eta} udz = u(h+\eta)=uD$$

$$N=\int_{-h}^{\eta} vdz = v(h+\eta)=vD \tag{1.6}$$

1.2 Bottom friction

The bottom friction is generally expressed as follows, in an analogy to the uniform flow,

$$\frac{\tau_x}{\rho}=\frac{1}{2g}\frac{f}{D^2}M\sqrt{M^2+N^2}$$

$$\frac{\tau_y}{\rho}=\frac{1}{2g}\frac{f}{D^2}N\sqrt{M^2+N^2} \tag{1.7}$$

where *f* is the friction coefficient. Without any detailed discussion of the value of *f* it is preferred to use Manning's roughness *n* which is familiar among civil engineers. Values of *n* are given in Table 1.1 [Chow, 1960].

The friction coefficient *f* and Manning's roughness *n* are related by

$$n=\sqrt{\frac{fD^{\frac{1}{3}}}{2g}} \tag{1.8}$$

This implies that *f* becomes rather large when the total depth *D* is small as *n* remains almost a constant. Thus, the bottom friction terms are expressed by

$$\frac{\tau_x}{\rho} = \frac{fn^2}{D^{7/3}} M\sqrt{M^2 + N^2}$$

$$\frac{\tau_y}{\rho} = \frac{fn^2}{D^{7/3}} N\sqrt{M^2 + N^2} \tag{1.9}$$

Throughout the present model, the expression of bottom friction in Eq.(1.9) is being used. *n* should be selected depending on the condition of the bottom surface according to the Table 1.1.

Table 1.1 Values of Coefficient of Bottom Friction *n* (after Linsley and Franzini, 1979)

Channel Material	**n**	**Channel Material**	**n**
Neat cement, smooth metal	0.010	Natural channels in good condition	0.025
Rubble masonry	0.017	Natural channels with stones and weeds	0.035
Smooth earth	0.018	Very poor natural channels	0.060

1.3 Governing equation

For the propagation of tsunami in the shallow water, the horizontal eddy turbulence could be negligible compared to the bottom friction except for run-up on the land. The following equations are therefore given as the fundamental equations in the present model.

$$\frac{\partial \eta}{\partial t} + \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} = 0$$

$$\frac{\partial M}{\partial t} + \frac{\partial}{\partial x}\left(\frac{M^2}{D}\right) + \frac{\partial}{\partial y}\left(\frac{MN}{D}\right) + gD\frac{\partial \eta}{\partial x} + \frac{gn^2}{D^{7/3}} M\sqrt{M^2 + N^2} = 0$$

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial x}\left(\frac{MN}{D}\right) + \frac{\partial}{\partial y}\left(\frac{N^2}{D}\right) + gD\frac{\partial \eta}{\partial y} + \frac{gn^2}{D^{7/3}} N\sqrt{M^2 + N^2} = 0 \tag{1.10}$$

1.4 Note on convection terms

The other expression of the shallow water equation using the averaged velocities in *x*- and *y*-directions (*u* and *v*) are often introduced by

$$\frac{\partial \eta}{\partial t}+\frac{\partial \overline{u}D}{\partial x}+\frac{\partial \overline{v}D}{\partial y}+\frac{\partial \overline{w}D}{\partial z}=0$$

$$\frac{\partial \overline{u}}{\partial t}+\overline{u}\frac{\partial \overline{u}}{\partial x}+\overline{v}\frac{\partial \overline{u}}{\partial y}+\overline{w}\frac{\partial \overline{u}}{\partial z}+g\frac{\partial \eta}{\partial x}+\frac{gn^2}{D^{1/3}}\overline{u}\sqrt{\overline{u}^2+\overline{v}^2}=0$$

$$\frac{\partial \overline{v}}{\partial t}+\overline{u}\frac{\partial \overline{v}}{\partial x}+\overline{v}\frac{\partial \overline{v}}{\partial y}+\overline{w}\frac{\partial \overline{v}}{\partial z}+g\frac{\partial \eta}{\partial y}+\frac{gn^2}{D^{1/3}}\overline{v}\sqrt{\overline{u}^2+\overline{v}^2}=0 \qquad (1.11)$$

We should remind that the above equation can not be applied to a numerical model on run-up because it does not satisfy the conservation of momentum. For example, the convection terms in the momentum equation in the *x*-direction in Eq. (1.10) divided by *D* can be modified as

$$\frac{1}{D}\left\{\frac{\partial}{\partial x}\left(\frac{M^2}{D}\right)+\frac{\partial}{\partial y}\left(\frac{MN}{D}\right)\right\}=\overline{u}\frac{\partial \overline{u}}{\partial x}+\overline{v}\frac{\partial \overline{u}}{\partial x}+\frac{\overline{u}}{D}\left\{\frac{\partial(\overline{u}D)}{\partial x}+\frac{\partial(\overline{v}D)}{\partial y}\right\}=\overline{u}\frac{\partial \overline{u}}{\partial x}+\overline{v}\frac{\partial \overline{u}}{\partial x}+\frac{\overline{u}}{D}\frac{\partial \eta}{\partial t}$$

(1.12)

The convection terms in Eq.(1.10) are not the same as those in Eq.(1.11) and even if the mass conservation equation in Eq.(1.10), is applied, the third term in left side of Eq.(1.12) can not be eliminated. Note that in the case of tidal current with longer wave period than tsunami in which acceleration term, the third term in left side of Eq.(1.12) is neglected.

2. NUMERICAL SCHEME

2.1 Numerical scheme for linearized equation

For the first step to describe the numerical scheme for the tsunami model, the linearized long wave equation without bottom frictions in one dimensional propagation, Eq.(2.1), is introduced.

$$\frac{\partial \eta}{\partial t}+\frac{\partial M}{\partial x}=0 \qquad (2.1)$$

$$\frac{\partial M}{\partial t}+gD\frac{\partial \eta}{\partial x}=0$$

Let us introduce the finite difference method to solve the above equation numerically. The finite difference method based upon the Taylor expansion series is shown as follows.

$$\eta(x,t+\Delta t)=\eta(x,t)+\Delta t\frac{\partial\eta(x,t)}{\partial t}+\frac{\Delta t^2}{2}\frac{\partial^2\eta(x,t)}{\partial t^2}+\frac{\Delta t^3}{3!}\frac{\partial^3\eta(x,t)}{\partial t^3}+\ldots.. \tag{2.2}$$

where *Δt* is the grid interval. We can form the "forward " difference by rearranging Eq.(2.2)

$$\frac{\partial\eta(x,t)}{\partial t}=\frac{\eta(x,t+\Delta t)-\eta(x,t)}{\Delta t}+O(\Delta t) \tag{2.3}$$

where the first term in the right side of Eq.(2.3) is obviously the finite difference representation for the first order of time derivative at t=*t* (see Fig.2.1).

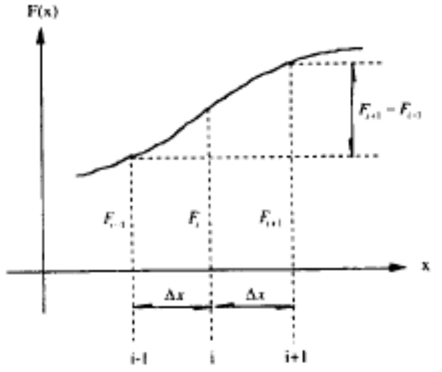


Figure 2.1 Central finite difference representations

The truncation error which have the order of *Δt*, (O(*Δt*)) is the difference between the partial derivative and its finite difference representation. Moreover we can rearrange the Taylor expansion series in Eq.(2.2) by replacing *Δt* by +*Δt* /2 and -*Δt*/2 and then we obtain "central" difference with the second order of truncation error.

$$\frac{\partial\eta(x,t)}{\partial t}=\frac{\eta\left(x,t+\frac{1}{2}\Delta t\right)-\eta\left(x,t-\frac{1}{2}\Delta t\right)}{\Delta t}+O(\Delta t^2) \tag{2.4}$$

It is interesting that although the expression of the finite difference representations in Eqs. (2.3) and (2.4) are similar, the order of truncation errors are different. By using the above "central" difference method with the staggered numerical points for water level and discharges, which is called the staggered leap-frog scheme, we can descretize Eq.(2.1) as follows.

$$\frac{1}{\Delta t}\left(\eta_i^{n+1}-\eta_i^n\right)+\frac{1}{\Delta x}\left(M_{i+\frac{1}{2}}^{n+\frac{1}{2}}-M_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right)+O\left(\Delta x^2\right)=0$$

$$\frac{1}{\Delta t}\left(M_{i+\frac{1}{2}}^{n+\frac{1}{2}}-M_{i+\frac{1}{2}}^{n-\frac{1}{2}}\right)+g\frac{\left(D_{i+1}^n+D_i^n\right)}{2}\frac{1}{\Delta x}\left(\eta_{i+1}^n-\eta_i^n\right)+O\left(\Delta x^2\right)=0 \qquad (2.5)$$

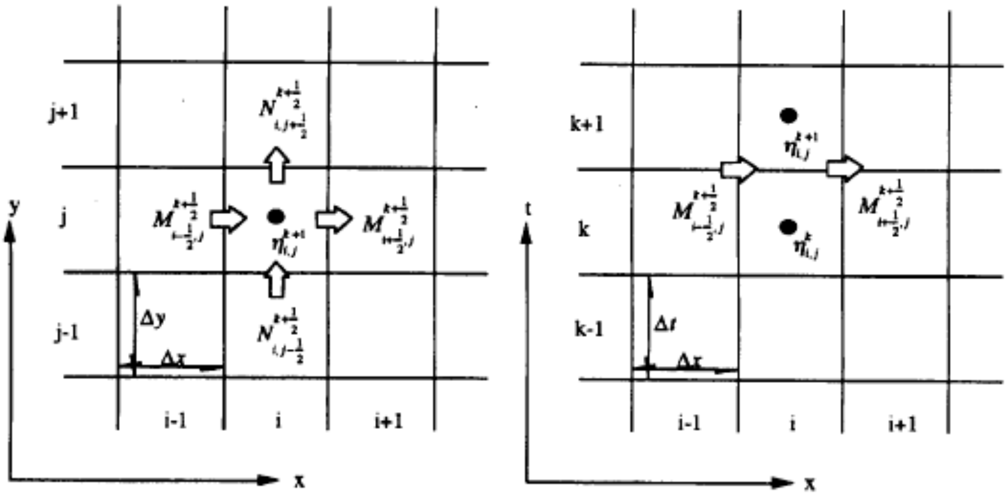
For dealing with discrete values in numerical computations, *η*(*x,t*) and *M*(*x,t*) are expressed for the case of the staggered leap-frog scheme as

$$\eta(x,t)=\eta(i\Delta x,n\Delta t)=\eta_i^n$$

$$M(x,t)=M\left\{\left(i+\tfrac{1}{2}\right)\Delta x,\left(n+\tfrac{1}{2}\right)\Delta t\right\}=M_{i+\frac{1}{2}}^{n+\frac{1}{2}} \qquad (2.6)$$

where *Δx* and *Δt* are the grid sizes in *x* direction and in time *t*. The point schematics for the numerical scheme are illustrated in Fig.2.2. The points for water depth *h* is the same as those for water elevation, *η*.

$$D_i^n=\eta_i^n+h_i$$



j+1
j
j-1
y
x
$N_{i,j+\frac{1}{2}}^{k+\frac{1}{2}}$
$M_{i-\frac{1}{2},j}^{k+\frac{1}{2}}$
$\eta_{i,j}^{k+1}$
$M_{i+\frac{1}{2},j}^{k+\frac{1}{2}}$
$N_{i,j-\frac{1}{2}}^{k+\frac{1}{2}}$
Δy
Δx
i-1
i
i+1
k+1
k
k-1
t
x
$\eta_{i,j}^{k+1}$
$M_{i-\frac{1}{2},j}^{k+\frac{1}{2}}$
$M_{i+\frac{1}{2},j}^{k+\frac{1}{2}}$
$\eta_{i,j}^{k}$
Δt
Δx
i-1
i
i+1

Figure 2.2 The point schematics for the numerical scheme

The above finite method provide stable result as long as the C.F.L condition is satisfied:

C (celerity) < *Δx /Δt*

Details of the stable condition will be discussed in the Chapter 3.1. Imamura & Goto (1988) investigated the truncation errors in three kinds of typical scheme for long waves simulations and showed that in term of numerical accuracy the staggered leap-frog scheme is the best among them.

2.2 Numerical scheme for convection terms

In the present numerical scheme, an "upwind" difference scheme is applied to the convection terms in order to make the computation stable. The reason why this scheme ensures the stability of computation is explained by taking a simple convection equation in the following:

$$\frac{\partial F}{\partial t} + C\frac{\partial F}{\partial x} = 0 \tag{2.7}$$

Here the coefficient *C* is the propagation velocity and is assumed constant. The arrangement of computation points in the present scheme requires the forward difference scheme for the first order time derivations. This yields

$$\frac{\partial F}{\partial t} = \frac{1}{\Delta t}\left(F_i^{n+1} - F_i^n\right) - \frac{\Delta t}{2}\frac{\partial^2 F}{\partial t^2} + O\left(\Delta t^2\right) \tag{2.8}$$

In addition, the central difference is applied to the space derivative.

$$C\frac{\partial F}{\partial x} = \frac{C}{2\Delta x}\left(F_{i+1}^n - F_{i-1}^n\right) + O\left(\Delta x^2\right) \tag{2.9}$$

As a result, F_i^{n+1} is given by

$$F_i^{n+1} = F_i^k - C\frac{\Delta t}{2\Delta x}\left(F_{i+1}^n - F_{i-1}^n\right) \tag{2.10}$$

The solution of Eq.(2.10) is implicitly equivalent the solution of Eq.(2.11) with an truncation error of ($\Delta t^2+\Delta x^2$). Substituting Eqs.(2.8) and (2.9) into Eq.(2.7) yields

$$\frac{\partial F}{\partial t}+\frac{\Delta t}{2}\frac{\partial^2 F}{\partial t^2}+C\frac{\partial F}{\partial x}=0 \tag{2.11}$$

If the second-order derivative with respect to time is rewritten by using the following relationship (this assumption is valid for the progressive waves),

$$\frac{\partial^2 F}{\partial t^2}=\frac{\partial}{\partial t}\left(-C\frac{\partial F}{\partial x}\right)=C^2\frac{\partial^2 F}{\partial x^2}$$

The solution of Eq.(2.11) is the same as the solution of the following diffusion equation in which the diffusion coefficient is negative.

$$\frac{\partial F}{\partial t}+C\frac{\partial F}{\partial x}=-\frac{\Delta t}{2}C^2\frac{\partial^2 F}{\partial t^2} \tag{2.12}$$

A negative diffusion works to amplify round-off errors with time leading to an instability. Therefore, Eq.(2.10) is an unstable difference scheme. The more detail about stable and unstable scheme will be discussed in chapter 3.1.

In order to obtain a stable scheme, the space derivative term is approximated by either forward or backward difference depending on the sign of coefficient *C*. With the forward difference, we have

$$C\frac{\partial F}{\partial x}=\frac{C}{\Delta x}\left(F_{i+1}^{n}-F_{i}^{n}\right)-\frac{\Delta x}{2}C\frac{\partial^2 F}{\partial x^2}+O\left(\Delta x^2\right)$$

and with the backward difference

$$C\frac{\partial F}{\partial x}=\frac{C}{\Delta x}\left(F_{i}^{n}-F_{i-1}^{n}\right)+\frac{\Delta x}{2}C\frac{\partial^2 F}{\partial x^2}+O\left(\Delta x^2\right)$$

The corresponding differential equations we are going to solve are within the truncation error of

O(Δt^2+ Δx^2), for the forward difference

$$\frac{\partial F}{\partial t} + C\frac{\partial F}{\partial x} = -\frac{C}{2}\left(C\Delta t + \Delta x\right)\frac{\partial^2 F}{\partial x^2} \tag{2.13}$$

and for the backward difference

$$\frac{\partial F}{\partial t} + C\frac{\partial F}{\partial x} = -\frac{C}{2}\left(- C\Delta t + \Delta x\right)\frac{\partial^2 F}{\partial x^2} \tag{2.14}$$

Therefore, to keep the virtual diffusion coefficient positive (or say to ensure the stability of the computation), we have used the backward difference in case of positive *C*, and the forward difference in case of negative *C*, in addition to setting $\frac{\Delta x}{\Delta t} > |C|$. In other words, the difference should be taken in the direction of the flow. This is the reason why this scheme is called the "upwind" difference. Although the leap-frog scheme has the truncation error of the order of Δx^2, as long as the convection term concerns, its order become large as Δx.

2.3 Numerical scheme for bottom friction term

The friction term becomes a source of instability if it is discretized with an explicit scheme. To make the discussion of instability simple, let us consider the following momentum equation without convection terms:

$$\frac{\partial M}{\partial t} + gD\frac{\partial \eta}{\partial x} + \frac{gn^2}{D^{7/3}}M|M| = 0 \tag{2.15}$$

The explicit form of Eq.(2.15) is :

$$M^{n+1} = \left(1 - \frac{gn^2}{D^{7/3}}|M|\right)M^n - gD\frac{\partial \eta}{\partial x} \tag{2.16}$$

When a velocity become large or a total depth is small in a very shallow water, the absolute of coefficient (amplification factor) of the first term on the right hand side of Eq.(2.16) become more than unity, which leads to numerical instability. In order to overcome this problem, an implicit scheme to set a friction term can be basically introduced. For example, a simple implicit form,

$$M^{n+1} = \frac{M^n}{\left(1 + \frac{gn^2}{D^{7/3}}|M|\right)} - \frac{gD\frac{\partial \eta}{\partial x}}{\left(1 + \frac{gn^2}{D^{7/3}}|M|\right)} \tag{2.17}$$

ensures numerical stability, because the amplification factor in Eq.(2.17) is always less than unity. However the effect of friction in shallow water becomes so large that numerical results are dumped. Another implicit form, a combined implicit one to the friction term is given by,

$$M^{n+1} = \frac{M^n\left(1 - \frac{gn^2}{2D^{7/3}}|M|\right)}{\left(1 + \frac{gn^2}{2D^{7/3}}|M|\right)} - \frac{gD\frac{\partial \eta}{\partial x}}{\left(1 + \frac{gn^2}{2D^{7/3}}|M|\right)} \tag{2.18}$$

This scheme also gives a stable result. It is, however, noted that the above scheme causes a numerical oscillation at the wave front because the amplification factor could be negative.

We should select the best scheme among some implicit ones to apply the bottom friction term with Manning's roughness. Considering the fact that the numerical scheme of convection terms also involve artificial or numerical dissipation, selection of Eq.(2.17) causes much damping in the result. Therefore the present model uses the combined implicit scheme, Eq.(2.18).

3. STABILITY AND CONSISTENCY

3.1 Stability

Tsunami propagation is one of marching problems that water level and discharges are varied with time. In our experiences in numerical simulations, a numerical result is un-expectedly diverged depending on grid size and time step, which is caused by instability in numerical simulations. In order to avoid such instability, a concept of stability and its condition is discussed in this chapter. A stable numerical scheme is one for which errors from any sources (round-off, truncation and so on) are not permitted to grow in the sequence of numerical procedures as the calculation proceeds from one marching step to the next.

Fourier or von Nuemann analysis is applied to obtain the stability condition for the given numerical scheme. Let *ε* represent the error in the numerical solution due to round-off errors. The numerical solution actually calculated may be written as,

$$F = D + \varepsilon \tag{3.1}$$

where *D* is the exact solution of finite difference equation. The computed numerical solution *F* must satisfy the differential equation. For example, substituting Eq.(3.1) into Eq.(2.10) yields :

$$D_j^{n+1} + \varepsilon_j^{n+1} = D_j^n + \varepsilon_j^n - C\frac{\Delta t}{2\Delta x}\left(D_{j+1}^n + \varepsilon_{j+1}^n - D_{j-1}^n - \varepsilon_{j-1}^n\right) \tag{3.2}$$

Since the exact solution *D* must satisfy the difference equation, the same is true for,

$$\varepsilon_j^{n+1} = \varepsilon_j^n - C\frac{\Delta t}{2\Delta x}\left(\varepsilon_{j+1}^n - \varepsilon_{j-1}^n\right) \tag{3.3}$$

In this case, the exact solution *D* and the error must both satisfy the same differential equation. This means that the numerical error and the exact numerical solution both possess the same growth property in time.

The error can be expressed in a Fourier series as follows:

$$\varepsilon(x,t) = \sum_{m}^{\infty} e^{at} e^{ik_m x} \tag{3.4}$$

where *k* is real but *a* may be complex. If Eq.(3.4) is substituted into Eq.(3.3), we get,

$$e^{a(t+\Delta t)} e^{ik_m x} = e^{at} e^{ik_m x} - \frac{r}{2}\left(e^{at} e^{ik_m (x+\Delta x)} - e^{at} e^{ik_m (x-\Delta x)}\right) \tag{3.5}$$

where *r* = *CΔt*/*Δx*. Dividing by $e^{at}e^{ik_m x}$, above expression becomes

$$e^{a\Delta t} = 1 - \frac{r}{2}\left(e^{ik_m \Delta x} - e^{-ik_m \Delta x}\right) \tag{3.6}$$

where β = *k*$k_m\Delta x$. The coefficient of $e^{a\Delta t}$ represents the amplification factor. It is clear that if $\left|e^{a\Delta t}\right|$ is less than or equal to one, the error will not grow from one time step to the next, which is the stability condition. This method is called von Nuemann analysis. However, this method is not always satisfied with the stability condition, as a result this numerical scheme lead to instable results. In the same way using the Fourier series for the error, the stability condition can be discussed as long as numerical scheme is linear. It is noted that von Nuemann analysis can not be directly applied to nonlinear equations. For instance, Eq.(2.10) is replaced by the following equation :

$$D_j^{n+1} = D_j^n - \frac{\Delta t}{2\Delta x}\left({D_{j+1}^n}^2 - {D_{j-1}^n}^2\right) \tag{3.7}$$

Substituting Eq.(31) yields :

$$\varepsilon_j^{n+1} = \varepsilon_j^n - \frac{\Delta t}{2\Delta x}\left({\varepsilon_{j+1}^n}^2 - {\varepsilon_{j-1}^n}^2\right) - \frac{\Delta t}{\Delta x}\left(D_{j+1}^n \varepsilon_{j+1}^n - D_{j-1}^n \varepsilon_{j-1}^n\right) \tag{3.8}$$

The above equation indicates that the error no longer satisfy the original differential equation, Eq.(3.7), and the error depends on the value D^n in the previous time step. An initial value for a numerical simulation affects the stability condition.

3.2 Consistency

Consistency deals with the extent to which the finite difference equations approximate the partial differential equations (PDE). The difference between the PDE and the finite difference approximation has already been defined as the truncation error of the difference representation as shown in Eq.(2.3). A finite-difference representation of a PDE is said to be consistent if we can show that the truncation error vanishes as the mesh is refined. This should always be the case if the order of the truncation error vanishes under grid refinement [ie., O(*Δt*), O(*Δx*)]. An example of a questionable scheme would be one for which the truncation error was O(*Δt/Δx*) where the scheme would not formally be consistent unless the mesh were refined in a manner such that *Δt/Δx* approach to 0.

On the other hand, it is our experience that, even when the computations are stably carried out, the computed results often behave against what is expected from the PDE used. These errors are introduced by the numerical scheme and the result in the damping of wave height or numerical oscillations behind the main wave. For the analysis of truncation error to know the accuracy obtained by a numerical scheme, the von Nuemann analysis already introduced in the stability condition can be applied. Because the finite difference equations to obtain the exact solution is the same as those for numerical errors, which has been shown in Eq.(3.3). In the case of stability problem, only absolute value of amplification factor of the error expressed by Fourier series is concerned. However, for the analysis of the truncation error, a phase angle Φ, as well as amplitude, which are defined by

$$e^{a\Delta t} = |G| e^{i\Phi} \tag{3.8}$$

should be discussed. For example, the staggered leap-frog scheme has

$$e^{a\Delta t} = e^{ik_m C_1 \Delta t}$$

where, $C_1 = C_0 \dfrac{2}{K k_m \Delta x} \sin^{-1}\left\{ K \sin\left(\dfrac{k_m \Delta x}{2} \right) \right\}$ should be discussed.

4. INITIAL CONDITIONS

4.1 Initial Conditions

The present program is only for tsunamis. No wind waves and tides are included. The still water level is given by tides and is assumed constant during tsunamis are computed. Accordingly, no motion is assumed up to the time *n-1*. It means, therefore, in sea,

$$\eta_{i,j}^{n-1}, \qquad M_{i+\frac{1}{2},j}^{n-\frac{1}{2}}, \qquad N_{i,j+\frac{1}{2}}^{n-\frac{1}{2}} = 0 \tag{4.1}$$

For run-up computation on land, the initial water level *η* is equal to the ground height *h*.

$$\eta_{i,j}^{n-1} = -h_{i,j} \tag{4.2}$$

It should be kept in mind that values of *h* take negative sign on land.

4.2 Static Source

4.2.1. Bottom Deformation due to the fault motion

The fault break parameters are

i) The location of the fault coordinates as starting and ending points consequently (length of the fault)

ii) The width of the fault

iii) The direction of the fault axis from North (Clockwise)

iv) The dip angle

v) The slip angle

vi) Vertical displacement of the fault

vii) Focal depth

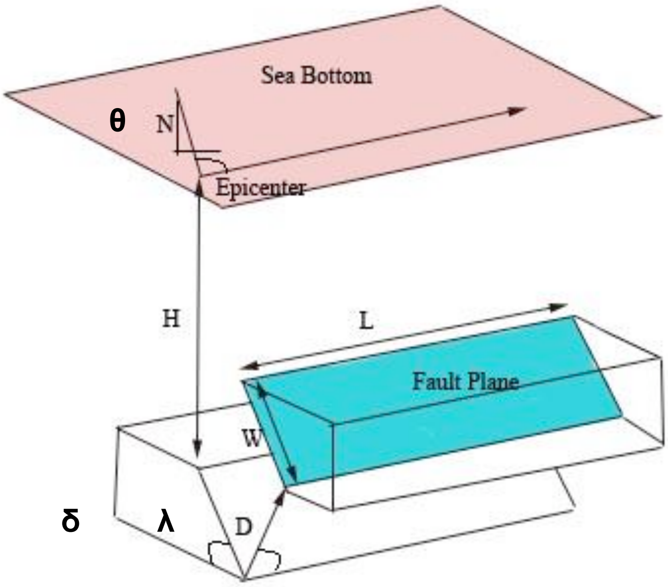


Figure 4.1 Fault break parameters

These parameters are inputted to the software FAULTWAVE.F for determination of the static tsunami source (initial wave) in the domain. This procedure is described separately under the subject of FAULTWAVE.F

4.2.2. Sea Bottom Subsidence

Sea bottom subsidence may be observed along the strike slip faults at some locations where the step over occurs. This is called pull apart mechanism.

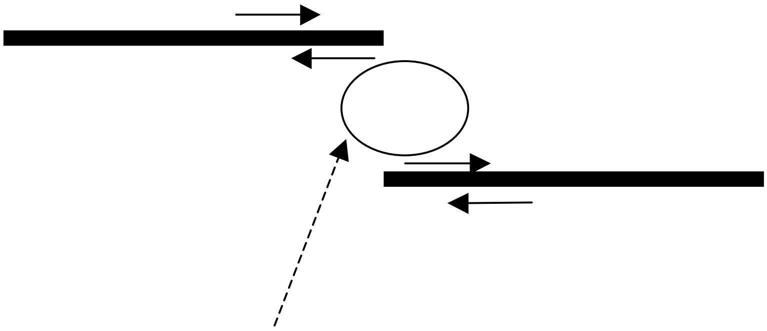


Figure 4.2 Subsidence at the tensile stress area

4.2.3. Dynamic Input of Wave Source

As for research purposes or understanding the basin response such as in special wave conditions or determination of resonance characteristics, the dynamic input can also be inputted in the basin. It is specifically controlled by Subroutine BNC1. The subroutine BNC1 must be modified according to the specified wave conditions. Help from Dr. Yalciner may be necessary.

5. OPEN BOUNDARY CONDITIONS

5.1 Open Boundary Conditions for Regular Waves and Forced Input

A method of input at an offshore boundary is given in the following when a pure sinusoidal wave train is propagating. Actual motion of the water on the offshore boundary is different from a pure sinusoidal one but is given as a resultant of the motion caused by the advancing purely sinusoidal wave train and receding waves. If it is assumed that the motion is sinusoidal at the boundary, this inevitably introduces a forced oscillation and no reflected wave can pass the boundary. It is, however, necessary to allow the reflected wave freely pass the boundary. This is easily solved if the method of characteristics is used at the boundary.

First, consider a one-dimensional case. The equations for linear long waves in a channel of constant depth are,

$$\frac{\partial u}{\partial t} + g\frac{\partial \eta}{\partial x} = 0 \tag{5.1}$$

$$\frac{\partial \eta}{\partial t} + h\frac{\partial u}{\partial x} = 0 \tag{5.2}$$

Eq. (5.1) is reduced to

$$\frac{\partial u}{\partial t} + \sqrt{gh}\frac{\partial}{\partial x}\left(\sqrt{\frac{g}{h}}\eta\right) = 0 \tag{5.3}$$

and Eq. (5.2) is reduced to

$$\frac{\partial}{\partial t}\left(\sqrt{\frac{g}{h}}\eta\right) + \sqrt{gh}\frac{\partial u}{\partial x} = 0 \tag{5.4}$$

Addition and subtraction of the two equations yield

$$\left\{\frac{\partial}{\partial t} \pm \sqrt{gh}\frac{\partial}{\partial x}\right\}\left(u \pm \sqrt{\frac{g}{h}}\eta\right) = 0 \tag{5.5}$$

The mathematical meaning of Eq. (5.4) is stated as that on characteristics certain values are kept constant. That is,

$$u \pm \sqrt{\frac{g}{h}}\eta \text{ is a constant on } \quad \frac{dx}{dt} = \pm\sqrt{gh} \tag{5.6}$$

Assume a sinusoidal wave train advancing in the negative *x*-direction having its front at $x=x_0$ at t=0, as shown in Fig.5.1. Equation (5.6) gives,

$$u_2 + \sqrt{\frac{g}{h}}\eta_2 = u_1 + \sqrt{\frac{g}{h}}\eta_1 \tag{5.7}$$

along a positive characteristics and

$$u_2 - \sqrt{\frac{g}{h}}\eta_2 = u_0 - \sqrt{\frac{g}{h}}\eta_0 \tag{5.8}$$

along a negative characteristics.

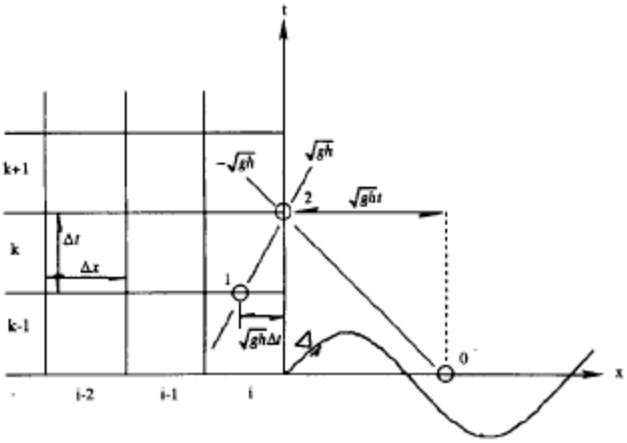


Figure 5.1 An offshore open boundary and characteristics when a simple harmonic wave train is propagating in the negative x direction.

The wave train is propagating in the negative *x*-direction, and satisfies the following relationship between the horizontal water particle velocity u_0 and the elevation of water surface η_0.

$$u_0 = -\sqrt{\frac{g}{h}}\eta_0 \tag{5.9}$$

With this relationship substituted, Eq.(5.9) is reduced to

$$u_2 = -\sqrt{\frac{g}{h}}\eta_2 = 2u_0 \tag{5.10}$$

and from Eqs.(5.9) and (5.10), we have

$$u_2 = u_0 + \frac{1}{2}\left(u_1 + \sqrt{\frac{g}{h}}\eta_1 \right) \tag{5.11}$$

If we assume the wave form of the incident wave train *η₀* is given by

$$\eta_0 = a\sin\left(k_0\left(x - x_0\right)\right)$$

where *k₀* is the wave number, then the corresponding water particle velocity is expressed by

$$u_0 = a\sqrt{\frac{g}{h}}\sin\left(k_0\left(x - x_0\right)\right) = a\sqrt{\frac{g}{h}}\sin\left(k_0\sqrt{gh}t\right) \qquad (5.12).$$

Consequently, the value of *u* at the boundary is composed of two parts as follows,

$$u_2 = a\sqrt{\frac{g}{h}}\sin\left(k_0\sqrt{gh}t\right) + \frac{1}{2}\left(u_1 + \sqrt{\frac{g}{h}}\eta_1 \right) \tag{5.13}$$

where the first term on the right-hand side is the given incident sinusoidal wave train and the second term is the receding wave.

If Eq.(5.13) is expressed in the present difference scheme, we have

$$M^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = a\sqrt{gh}\sin\left(k_0\sqrt{gh}\left(n + \frac{1}{2}\right)\Delta t \right) + \frac{1}{2}\left(M_1 + \sqrt{gh}\eta_1\right) \tag{5.14}$$

where M_1 and η_1 are calculated by

$$M_1 = \frac{1}{\Delta x}\left[\sqrt{gh}\Delta t M^{n-\frac{1}{2}}_{i-\frac{1}{2},j} + \left(\Delta x - \sqrt{gh}\Delta t\right)\right] M^{n-\frac{1}{2}}_{i+\frac{1}{2},j} \tag{5.15}$$

$$\eta_1 = \frac{1}{\Delta x}\left[\frac{1}{2}\left(\sqrt{gh}\Delta t - \frac{\Delta x}{2}\right)\left(\eta^{n}_{i-1,j} - \eta^{n-1}_{i-1,j}\right) + \frac{1}{2}\left(\frac{3}{2}\Delta x - \sqrt{gh}\Delta t\right)\left(\eta^{n}_{i,j} - \eta^{n-1}_{i,j}\right)\right] \tag{5.16}$$

Second, we consider a two-dimensional case in which the relationship of characteristics is given on characteristic surface. Extension of the one-dimensional case mentioned above is used here. Different from a one dimensional problem, the direction of wave propagation should be determined carefully. In general, the propagation direction of the incident waves is given and constant and therefore negative characteristics have a constant direction. On the other hand the direction of positive characteristics corresponding to the reflected waves may be different from that of the incident waves. The direction of positive characteristics is determined as the direction of the resultant of M(I, j, k-1/2) and N(I, j, k-1/2). In this way, negative and positive characteristics should be computed with the same method as in the one-dimensional problem on taking their propagation direction into consideration.

5.2 Open Boundary Conditions for Forced Input

When the boundary condition itself is already composed of progressive and reflected waves, it is given with no modification on the boundary. There is no need to follow the method developed in the precedent subsection. No consideration is required to make the reflected wave freely pass the boundary. In case of linear problem, either displacement of water surface or discharge flux is used as the input at the boundary.

5.3 Open Boundary Conditions for Free Transmission

A method is given to make waves in the computation region go outward by freely passing the open boundary. The characteristics relationship in (2) above is used. At the boundary $x = x_0$, the relationships are

$$u_0 = h_0 = 0 \quad \text{and} \quad u_1 = \sqrt{\frac{g}{h}}\eta_1$$

This yields

$$u_2 = u_1 \quad \text{or} \quad M^{n+1/2}_{i+1/2,j} = M_1$$

In case of two-dimensional case, the direction of discharge should be taken into consideration. Assume that an incident wave propagates toward the negative *x*-direction. The direction of characteristics corresponding to the wave is given and is constant with respect to both time and space. On the other hand, characteristics corresponding to waves reflected from land boundary may change their direction with respect to time and place. The direction of characteristics of reflected waves is determined as the direction of the resultant of *M* and *N*. With the directions of incident and reflected waves taken into consideration, the same method as in case of one-dimensional cases is applied.

6. BOUNDARY CONDITIONS AT RUN-UP FRONT

6.1 Wave front condition

Run-up is taken into consideration only in nonlinear computations but not in linear computations. Whether a cell is dry or submerged is judged as follows.

$D = h + \eta > 0$, then the cell is submerged and

≤ 0, then the cell is dry.

The wave front is located between the dry and submerged cells. Discharge across the boundary between the two cells is carried out if the ground height in the dry cell is lower than the water level in the submerged cell. In other cases, discharge is considered zero.

6.2 Boundary Conditions When Water Overflows Structures

The Hom-ma formula is used when water overflows breakwaters and sea walls in the computation region. Discharge overflowing the structures is given by

$$
\begin{aligned}
Q &= \mu \ h_1\sqrt{2gh_1}\ , && \text{if } h_2 \leq \frac{2}{3}h_1 \\
Q &= \mu' h_1\sqrt{2g(h_1 - h_2)}, && \text{if } h_2 > \frac{2}{3}h_1 \qquad (6.1)
\end{aligned}
$$

where h_1 and h_2 are the water depths in front of and behind structures measured above the top of structures, μ=0.35 and μ'=2.6μ.

A tsunami front runs up and down the land. It is not easy to express this moving boundary with the coordinate system in the Eulerian description. If equations in the Lagrangian description are used, the moving condition can be expressed with no approximation (Shuto & Goto, 1978). Even with equation in the Eulerian description, a variable transformation in which the origin of the new coordinates is located at the front can easily express the moving boundary (Takeda, 1984). It is unfortunate, however, that these two methods are well applicable only to one dimensional problem, but poorly to any two-dimensional practical problem with complicated topography.

There are several approximate moving boundary conditions. In the leap-frog scheme, grid points are alternatively located for velocity and water level assuming that the water level is already computed as a computational cell. If the water level is higher than the latter, the water may flow into the landward cell. Figure 6.1 explains the way to estimate the inflow velocity or discharge.

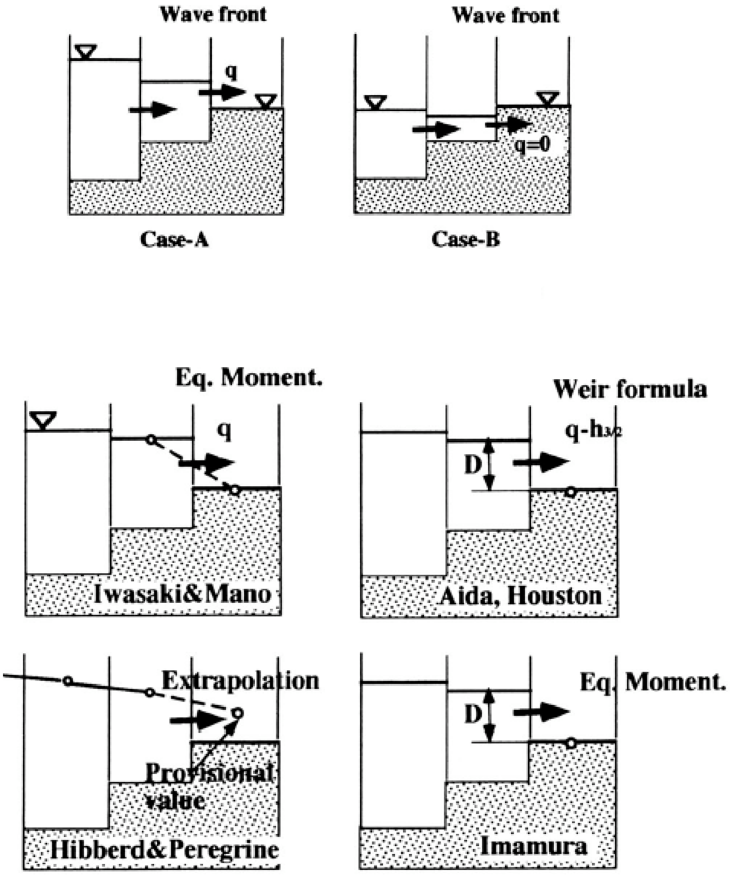


Figure 6.1 Various moving boundary conditions at a wave front

Iwasaki and Mano (1979) assume that the line connecting the water level and the bottom height gives the surface slope to the first-order approximation. Hibberd and Peregrine (1979) give a provisional water level in the dry cell on a linearly extrapolated water surface. Then, the discharge calculated with this provisional water level gives the total amount of water into the dry cell and the water depth in the cell. If necessary, the computation will be repeated with the water level thus modified. Aida (1977) and Houston and Butler (1979) evaluate the discharge into the dry cell with broad-crested weir formulas in which the water depth above the bottom of the dry cell is substituted. The coefficient of the discharge to $\sqrt{gh}$ should be determined by a flow condition such as the Froude number $\left(\frac{u}{\sqrt{gh}}\right)$.

These approximations are convenient to handle but introduce numerical errors (Goto and Shuto, 1983a). The run-up height computed with the Iwasaki-Mano method agrees with the theoretical solution with a 5% range of error if the following condition is satisfied

$$\Delta x / \alpha g T^2 < 4 \times 10^{-4}$$

With the Aida method, the condition is given by

$$\Delta x / \alpha g T^2 < 10^{-3}$$

in which is the *α* angle of slope, *g* is the gravitational acceleration, *Δx* is the spatial grid length, and *T* is the wave period.

There are three methods to include the effects of buildings into tsunami simulation. The simple method in a hind casting is to allot a large friction coefficient *f*, which ranges from 0.2 to 1, to the residential areas. This method is however, not applicable to a forecasting, because an appropriate value of the friction coefficient is determined only after comparisons of the computed inundated area with the recorded.

The second method which can be used in a forecasting is to determine an equivalent friction coefficient by summing up the drag of individual building. This is allowed to use the different values of the friction coefficient depending on a surface condition, for example, C_f=5x10^{-3} for coastal water and 1x10^{-2} on land without obstacles.

The third and the best one is to use very fine grids in the city area. If grids are less than 5 m wide, most of the large building can be expressed as impermeable boundaries. This inevitably increases the number of grid points. Fine grids also introduce the question of whether or not a map used in discretization is accurate enough for this detailed computation.

In the neighborhood of a large obstacle covered by fine grids, the water flow can be numerically simulated. However, this leaves the question of whether the computed result is reliable. Uda et al.(1988a) computed a tsunami that overflowed a model sand dune. They compared the results with measured data in hydraulic experiment. Agreement was poor just behind the dune where the water flow varied from supercritical to sub-critical flows through a jump. The, agreement recovered further behind the dune.

7. CONTINUATION OF REGIONS

- Continuation of Spatial Regions
- Continuation of Time Regions

7.1 Necessity of Continuation of Regions in Numerical Computation

In the design of numerical computations for long waves such as storm surges and tsunamis, the open sea boundary had better be set in the deep ocean where the boundary conditions can be accurately and easily given. On the other hand, for economy of CPU time, (1) linear or nonlinear theories are used according to the degree of nonlinearity of the phenomena, and (2) coarse grids are used in the deep sea and fine grids are in the nearshore zone. This requires the continuation of computation at the boundary of regions of different grid length.

The equations belong to the wave equation, for which the CFL condition should be satisfied for stability of numerical computation.

$$\frac{\Delta x}{\Delta t} = \sqrt{2gh_{\max}} \tag{7.1}$$

where Δt and Δx are the temporal and spatial grid lengths, and h_{max} is the maximum still water depth in a computation region. Approaching the shore, h_{max} becomes smaller, then a smaller x is selected to satisfy the CFL condition on keeping Δt constant. This is an ordinary way of how to select the temporal and spatial grids, if run-up is not included in simulations. If run-up is included, it is sometimes very hard to satisfy the CFL condition by changing only the spatial grid length. In this case, not only *Δx* but also *Δt* are changed in the different computation regions.

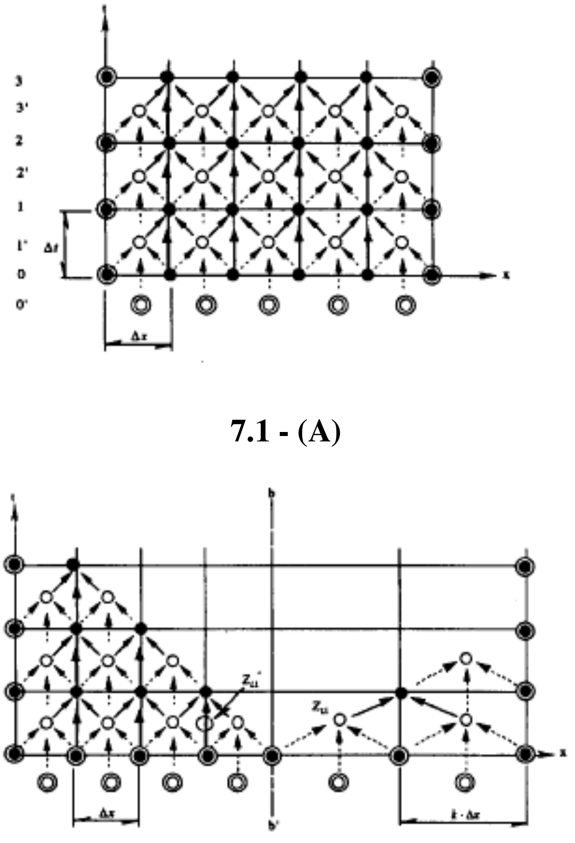
Continuation methods of water level and discharge between regions of different temporal and spatial grid lengths are given in the following, in order to carry out the computation continuously and smoothly. Eq. (7.1) must certainly be satisfied when selection of *Δx* and *Δt* for stability.

7.2 Continuation of Regions of Different *Δx*

In a problem which is two-dimensional in space, three independent variables, *x*, *y* and *t*, should be taken into consideration. In the following, it is assumed that *t* is constant in any region of different spatial grid length. In section (a), a method of continuation is explained by using one-dimensional propagation. In section (b), a method of continuation is given for the spatially two-dimensional problem.

(a) Continuation of Region in the *x* - *t* plane

Figure 7.1 shows the process of computation in the *x* - *t* plane in case of a one-dimensional propagation. Arrows given by solid lines are for the computation of the discharge, and that by broken lines is for the computation of the water levels. Numerals without and with prime ' on the *t*-axis corresponds to the time step when the discharge or the water level is computed. The discharge corresponding to a computation cell denotes the discharge through the positive *x* - side of the cell.



7.1 - (A)

7.1 - (B)

Figure 7.1 Computation procedure in the x-t plane, (A) if the spatial grid length is constant in the whole region, (B) if two regions of different grid length are connected at the boundary b-b’.

Figure 7.1(A) is the mesh when *Δx* is constant in the whole region of computation. In order to begin the computation, values at points of double circles on boundaries should be known; they are, the water level *Z* at *t* = 0' and the discharge *M* at *t* = 0 as the initial condition, the discharge or water level along *x* = 0 and *x* = *nΔx* as the boundary condition. If they are given, the water level Z_1 at *t* = 1' is calculated with the equation of continuity, then the discharge M_1 at *t* = 1 is obtained with the equation of motion. The same procedure is repeated to determine *Z* and *M* in the direction of time.

Fig.7.1 (B) is a case of continuation of regions of different grid length. In the region S to the left of the line b-b', the spatial grid length is *Δx,* and in the right region L the spatial grid length is larger and is *kΔx* (*k* > 1). The computation procedure in each region is the same as in the former case. However, if values on the line b-b' are not calculated, the region where the solution is given becomes narrower with the lapse of time as shown in Fig.7.1 (B).

In order to obtain the solution in the whole *x* - *t* plane, discharge should be known on the boundary b-b' at first. This value of discharge can be calculated either in the region S or in the region L. Assume now that the discharge is calculated in the computation for the region L. In order to calculate the discharge at *t* = 1 on the line b-b', we need the value of the water level $Z_{L1'}$ at a point in the region S, the position of which is symmetric to the point for Z_{L1} with respect to the line b-b'. An interpolation may be used to determine ZL1' from values obtained in the region S. However, in place of interpolation, the present authors recommend to set *k* be an odd number.

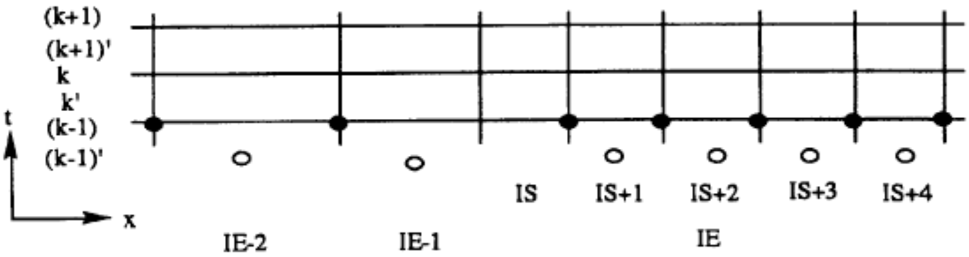
In the authors' program, the following assumptions are being taken.

(i) The ratio *k* is 3.

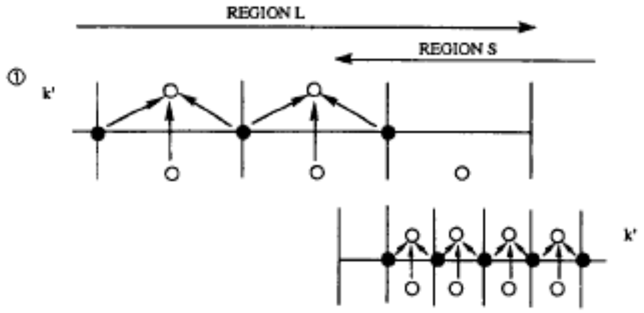
(ii) For continuation, the region L needs an extra mesh in the region S beyond the line b-b'.

(iii) In more general case than Fig. 7.1 (B) the region S needs an extra mesh in the region L beyond the line b-b'. Otherwise, when the direction of the x-axis is taken inversely to the case in Fig. 7.1(B) values of discharge on the right boundary in the region S can not be calculated, because the point of computation for discharge is located on the left side of the computation mesh.

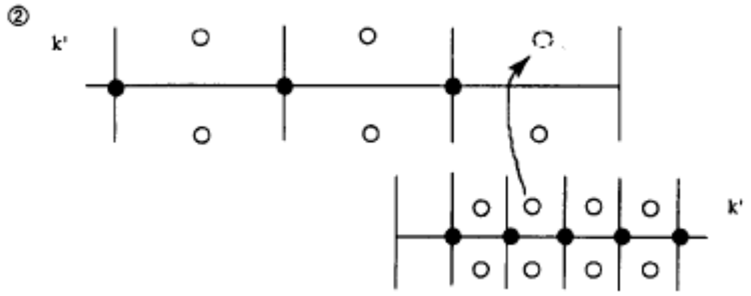
In conclusion, the computation procedure for the continuation of region is summarized and shown as follows for a one-dimensional case.



7.2 - (A)



7.2 - (B)



7.2 - (C)

Figures. 7.2 (A), (B) and (C) : the computation procedure for the continuation of region is summarized as shown for a one-dimensional case.

(b) Continuation of Region in the *x*,*y*-*t* Space

Discussion in the preceding section gives the way of selection of values when a difference equation is solved across the boundary between two regions of different grid length.

(i) The water level in the region S in the neighborhood of the line b-b' is used in the computation in the region L.

(ii) The discharge in the region L in the neighborhood of the line b-b' is used in the computation in the region S.

The method is explained for a two-dimensional case, on referring Fig. 7.3 where circles are computation points for water level and arrows are those for discharge.

As for the water level, two methods are possible. Since *k* is taken equal to 3, when the region L is extended by an extra cell into the region S, this extra mesh is composed of 9 small cells of the region S. The central point (marked by double circles in Fig.7.3) of the 9 cells is located at the central point of the extra cell.

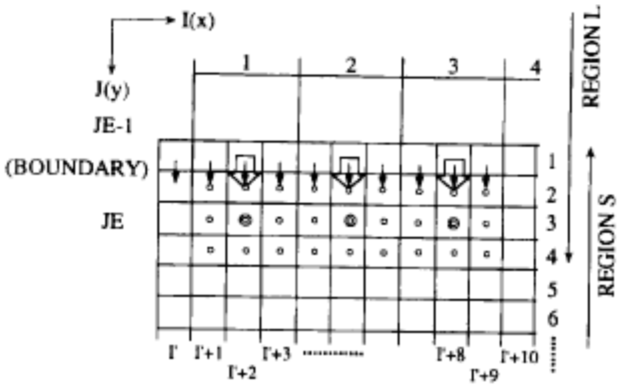


Figure 7.3 Continuation between two regions S and L in the (x, y-t) space, where k = 3 and Δt = const.

Therefore, we set

$$Z_L(1, JE) \quad = \quad Z_S(I'+2, 3)$$

or, taking the mean value averaged over 9 cells, we have

$$Z_L(1, JE) = \frac{1}{9}\sum_{I'=I'+1}^{I'+3}\sum_{J'=2}^{4} Z_S(I', J')$$

As for the discharge, different from the water level, either interpolation or extrapolation is inevitable. Along the boundary between the regions L and S, discharges are connected as follows.

i) At points from *I'*+2 to *I'*+8 along the boundary in Fig.7.3, discharges are calculated by an interpolation as Eq. for *MS*

$$M_s(I'+3,1) = (2.0M_l(1, JE-1) + 1.0M_L(2, JE-1))/3.0$$

ii) At other points than stated above, an interpolation is used if the region L still exists. For example, the discharge at the point *I'*+10 on the boundary in Fig.7.3 is given by, Eq. for *M*(*I'*+10, 1)

$$M_s(I'+10,1) = (1.0M_L(3, JE-1) + 2.0M_L(4, JE-1))/3.0$$

If the region L does not exist as the point *I'* in Fig.7.3, an extrapolation is used. Then we have,

Eq. for *M*(*I'*, 1)

$$M_s(I',1) = \left(5.0M_l(1,JE-1) - 2.0M_L(2,JE-1)\right)/3.0$$

(2) Continuation of Regions of Different *Δt*

Assume that the time grid changes from 3*Δt* to *Δt*. Figure 7-4 shows the computation procedure in the *x* - *t* plane.

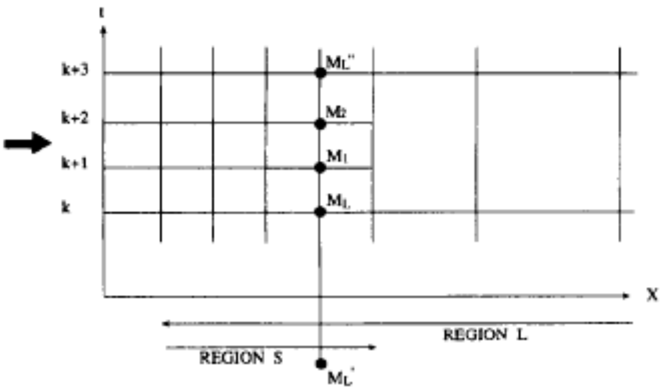


Figure 7.4 Computation continuation between two regions S and L in the x-t plane when *Δt* varies at the ration of 1:3.

As for the water level, values in the region S are used in the computation in the region L. This means that values are connected at the time shown by an arrow in Fig.7.4; i.e., at the middle of 3 cells in the region S.

As for the discharge, values in the region L are used in the computation in the region S. Therefore, at each time step in the region S (at *K*, *K*+1, K+2, and so on), values of discharge such as M_1 and M_2 should be given. Let M_L at time *K* and M_L' at time K-3 be known. Then, M1 is calculated by an extrapolation. Even though the same extrapolation is applied to M2, an interpolation is preferable for *M*2 in order to reduce numerical error. For this interpolation, value of M_L'' should be known beforehand at time K+2. Fig. 7.5 shows the way of computation in detail.

A comment is added here that the position of time-region continuation is not necessarily the same as the position of space-region continuation.

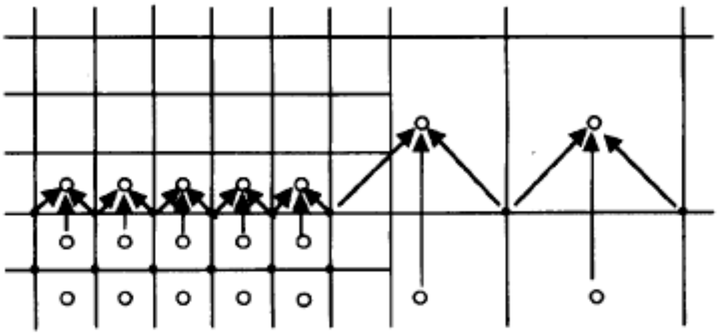


Figure 7.5 (a) Details of computation continuation in the x-t plane, when Δt varies at the ratio of 1:3. Computation of the water level Z in the regions S and L, with the equation of continuity.

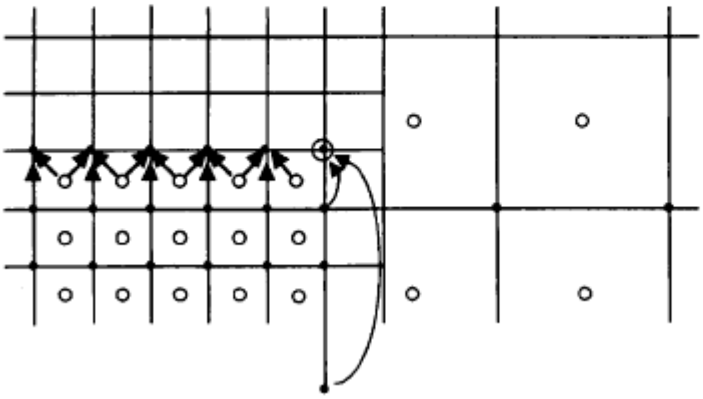


Figure 7.5(b) Computation of the water discharge M in the region S, with the equation of motion. An extrapolation in the region L to determine value of M at the boundary. Connection of the water discharge from the region L to the region S.

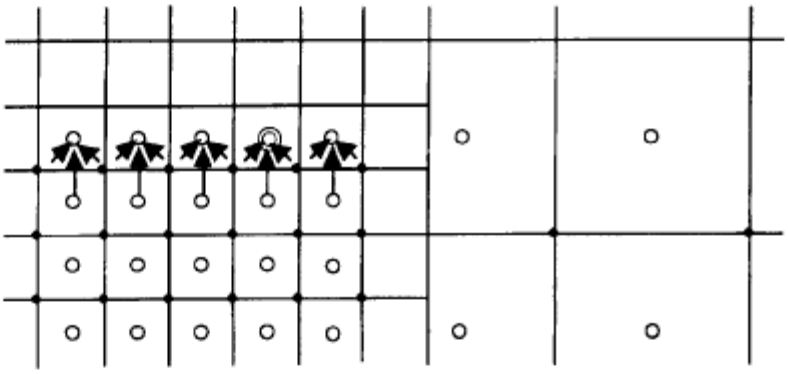


Figure 7.5(c) Computation of the water level Z in the region S, with the equation of continuity. Connection of the water level from the region S to the region L.

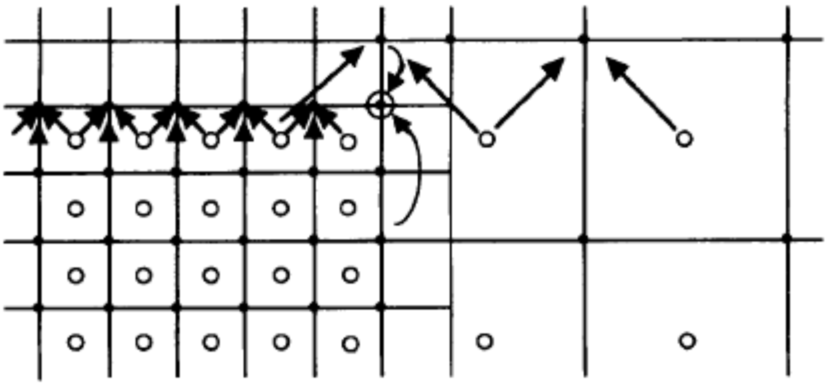


Figure 7.5 (d) Computation of the water discharge M in the region S and L, with the equation of motion. An interpolation in the region L to determine values of M at the boundary. Connection of the water discharge from the region L to the region S.

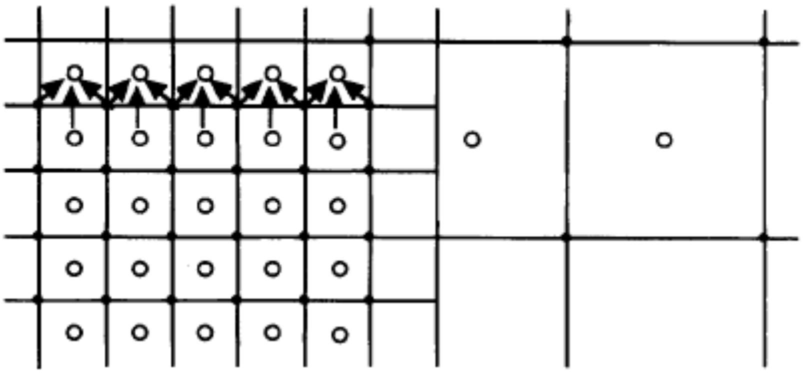


Figure 7.5 (e) Computation of the water level Z in the region S, with the equation of continuity.

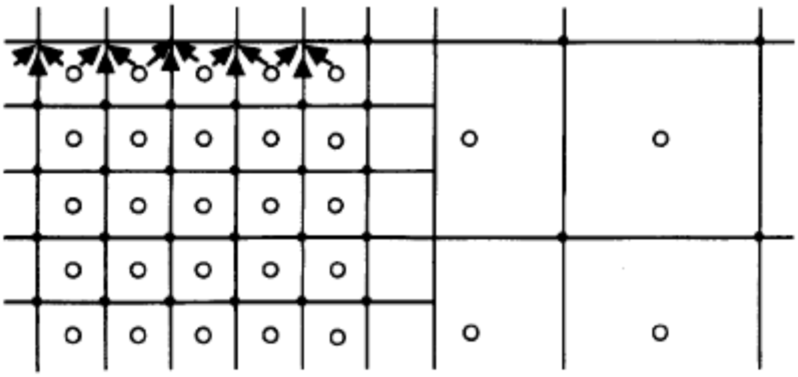


Figure 7.6 (f) Computation of the water discharge M in the region S, with the equation of motion

8. PROCESSING BATHYMETRIC DATA

TUNAMI N2 program uses the bathymetry of the area as input data. The bathymetry of the area is usually stored as data files. This file consists of three values; x coordinate, y coordinate and the depth values. However data files are typically randomly spaced files, and this data must be converted into an evenly spaced grid before using as input file of the program. To convert into a grid file, a program called Surfer is used.

Surfer is a contouring and 3D surface mapping program that runs under Microsoft Windows. It quickly and easily converts your data into outstanding contour, surface, wireframe, vector, image, shaded relief, and post maps. Further information can be found at the website of Golden Software; www.goldensoftware.com .

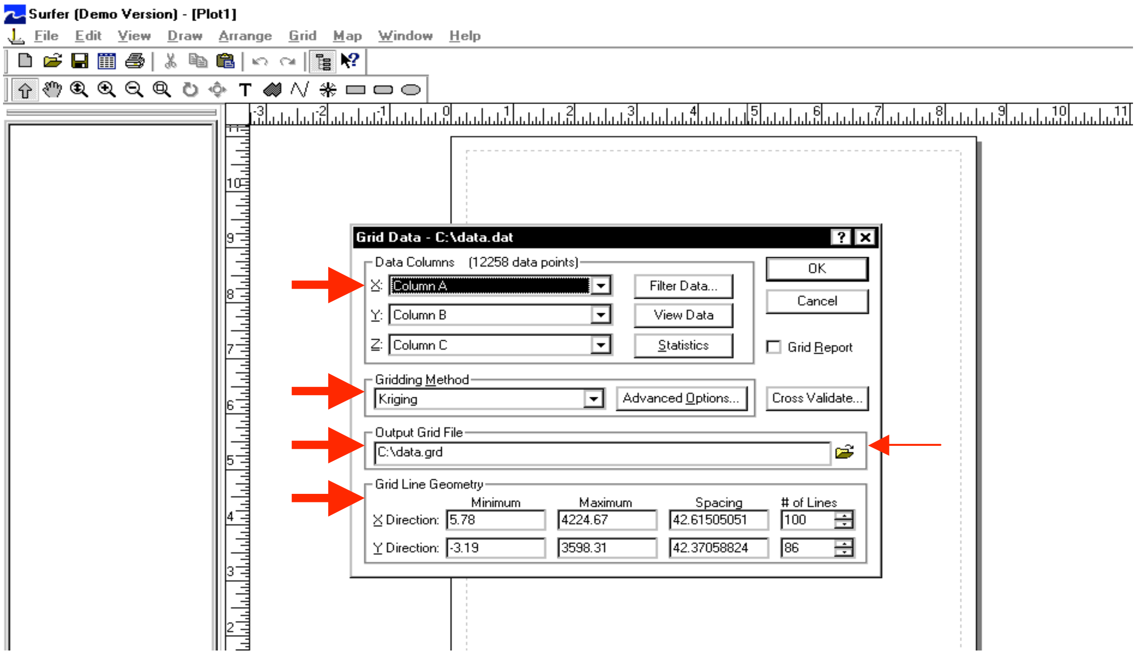
Below is the procedure for converting the bathymetry data file to grid file by SURFER.

1. Start Surfer.

2. Click on the **Grid | Data** command to display the **Open** dialog.

3. Specify the name of the XYZ data file which is the bathymetry data of the area, and then click OK.

4. In the **Grid Data** dialog, specify the parameters for the type of grid file you want to produce.



Data Columns (12258 data points)

X: Column A

Y: Column B

Z: Column C

□ Grid Report

Gridding Method: Kriging

Output Grid File: C:\data.grd

Grid Line Geometry

	Minimum	Maximum	Spacing	# of Lines
X Direction:	5.78	4224.67	42.61505051	100
Y Direction:	-3.19	3598.31	42.37058824	86

The Grid Data Dialog

When creating a grid file you can usually accept all of the default gridding parameters

Data Columns

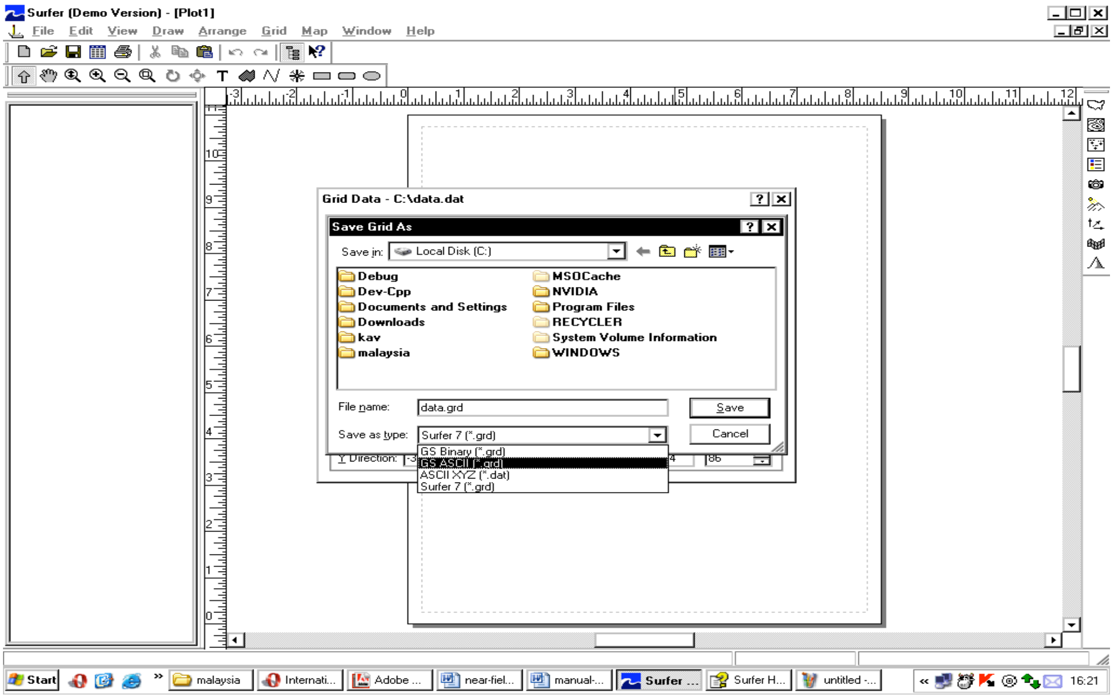
Individually specify the columns for the X data, the Y data, and the Z data. **Surfer** defaults to *X:* Column A, *Y:* Column B, and *Z:* Column C, which represents the x coordinate, y coordinate and the depth respectively.

Gridding Method

The gridding method should be set to Kriging which is the recommended gridding method with the default linear variogram. This is actually the selected default gridding method because it gives good results for most XYZ data sets.

Output Grid File

Choose a path and file name for the grid in the *Output Grid File* group by clicking the button 📂 (shown by an arrow). **Save Grid As** dialog will be opened. You can type a path and file name or browse to a new path and enter a file name in the box. TUNAMI N2 requires a specific file format for the output grid file which is ASCII grid file format. To change the file format, use the drop down Save as type menu and choose GS ASCII (*.grd) format as shown below. Then click Save and return to Grid Data menu.



Surfer (Demo Version) - [Plot1]

Grid Data - C:\data.dat

Save Grid As

Save in: Local Disk (C:)

File name: data.grd

Save as type: Surfer 7 (*.grd)

ASCII Grid File Format

ASCII grid files [.GRD] contain five header lines that provide information about the size and limits of the grid, followed by a list of Z values. The fields within ASCII grid files must be space delimited.

The listing of Z values follows the header information in the file. The Z values are stored in row-major order starting with the minimum Y coordinate. The first Z value in the grid file corresponds to the lower left corner of the map. This can also be thought of as the southwest corner of the map, or, more specifically, the grid node of minimum X and minimum Y. The second Z value is the next adjacent grid node in the same row (the same Y coordinate but the next higher X coordinate). When the maximum X value is reached in the row, the list of Z values continues with the next higher row, until all the rows of Z values have been included.

The general format of an ASCII grid file is:

id	The identification string DSAA that identifies the file as an ASCII grid file.
nx ny	nx is the integer number of grid lines along the X axis (columns)
	ny is the integer number of grid lines along the Y axis (rows)
xlo xhi	xlo is the minimum X value of the grid
	xhi is the maximum X value of the grid
ylo yhi	ylo is the minimum Y value of the grid
	yhi is the maximum Y value of the grid
zlo zhi	zlo is the minimum Z value of the grid
	zhi is the maximum Z value of the grid
grid row 1 grid row 2 grid row 3 ………..	These are the rows of Z values of the grid, organized in row order. Each row has a constant Y coordinate. Grid row 1 corresponds to ylo and the last grid row corresponds to yhi(i.e. the first row in the grid file is the last row on the map). Within each row, the Z values are arranged from xlo to xhi.

The following example grid file is ten rows high by ten columns wide. The first five lines of the file contain header information. X ranges from 0 to 9, Y ranges from 0 to 7, and Z ranges from 25 to 97.19. The first Z value shown corresponds to the lower left corner of the map and the following values correspond to the increasing X positions along the bottom row of the grid file. This file has a total of 100 Z values.

```
DSAA
10 10
0.0 9.0
0.0 7.0
25.00 97.19
91.03 77.21 60.55 46.67 52.73 64.05 41.19 54.99 44.30 25.00
96.04 81.10 62.38 48.74 57.50 63.27 48.67 60.81 51.78 33.63
92.10 85.05 65.09 53.01 64.44 65.64 52.53 66.54 59.29 41.33
94.04 85.63 65.56 55.32 73.18 70.88 55.35 76.27 67.20 45.78
97.19 82.00 64.21 61.97 82.99 80.34 58.55 86.28 75.02 48.75
91.36 78.73 64.05 65.60 82.58 81.37 61.16 89.09 81.36 54.87
86.31 77.58 67.71 68.50 73.37 74.84 65.35 95.55 85.92 55.76
80.88 75.56 74.35 72.47 66.93 75.49 86.39 92.10 84.41 55.00
74.77 66.02 70.29 75.16 60.56 65.56 85.07 89.81 74.53 51.69
70.00 54.19 62.27 74.51 55.95 55.42 71.21 74.63 63.14 44.99
```

Grid Line Geometry

Grid line geometry defines the grid limits and grid density. Grid limits are the minimum and maximum X and Y coordinates for the grid. Grid density is usually defined by the number of columns and rows in the grid. The *# of Lines* in the *X Direction* is the number of grid columns, and the *# of Lines* in the *Y Direction* is the number of grid rows. By defining the grid limits and the number of rows and columns, the *Spacing* values are automatically determined as the distance in data units between adjacent rows and adjacent columns.

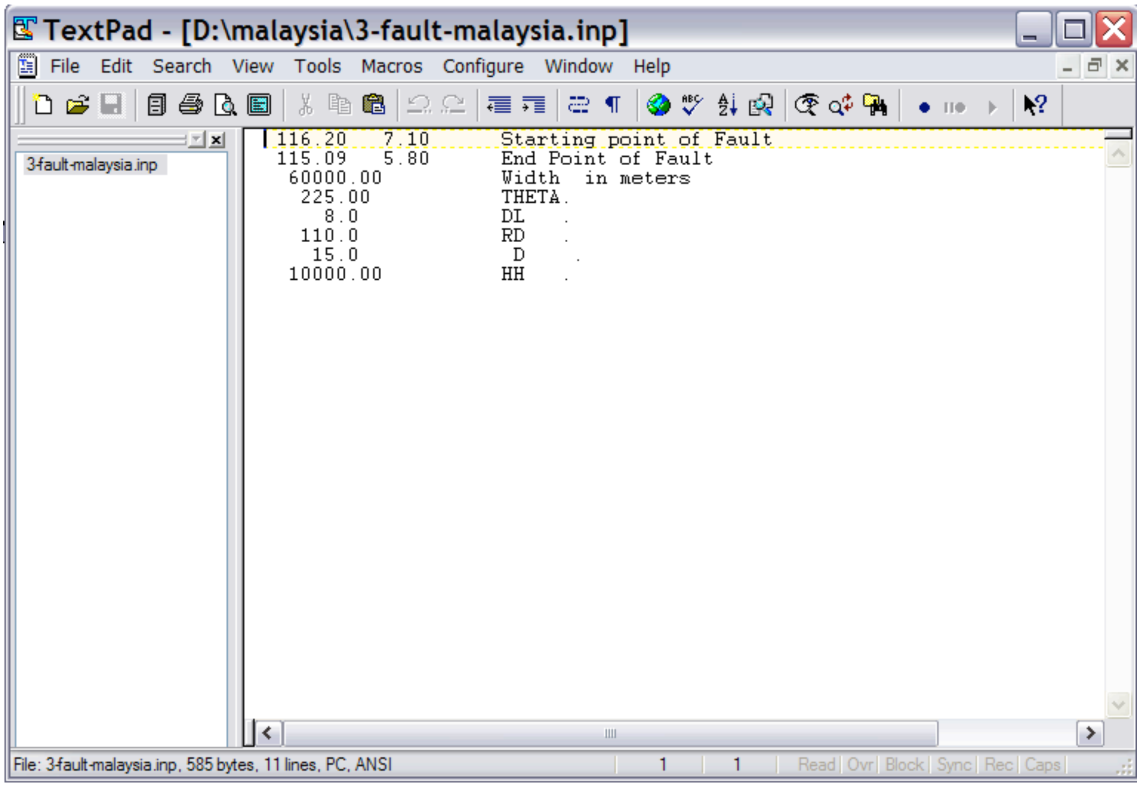
5. Click OK and the grid file is created. During gridding, the status bar at the bottom of the **Surfer** window provides you with information about the progress of the gridding process.

9. SOFTWARE FAULTWAVE.F

TUNAMI does not use the fault data as it is. There is another software called FAULTWAVE.f which runs the fault data and produces the initial wave by the fault. TUNAMI uses this initial wave data to start with the modeling.

9.1. Input Files and Parameters

The software requires the bathymetry of the area as well as the fault data as the input. The data concerning the fault consists of the following parameters. The coordinates of the starting point and the end point of the fault, the width of the fault in meters, the dip direction, dip angle, slip angle in degrees, the dislocation and the depth in meters. This information will be stored in a txt file as shown below. Then the file will be saved as *.inp.



TextPad - [D:\malaysia\3-fault-malaysia.inp]
File Edit Search View Tools Macros Configure Window Help
3-fault-malaysia.inp
File: 3-fault-malaysia.inp, 585 bytes, 11 lines, PC, ANSI
1
1
Read Ovr Block Sync Rec Caps

```
116.20    7.10          Starting point of Fault
115.09    5.80          End Point of Fault
 60000.00               Width  in meters
  225.00                THETA.
    8.0                 DL    .
  110.0                 RD    .
   15.0                  D     .
 10000.00               HH    .
```

Parameters of the program is listed below (which are also the input values in the *.inp file)

RR : radius of earth

L : fault length (m)

W : fault width (m)

TH : dip direction (degrees)

DL : dip angle (degrees)

RD : slip angle (degrees)

D : dislocation

HH : depth (m)

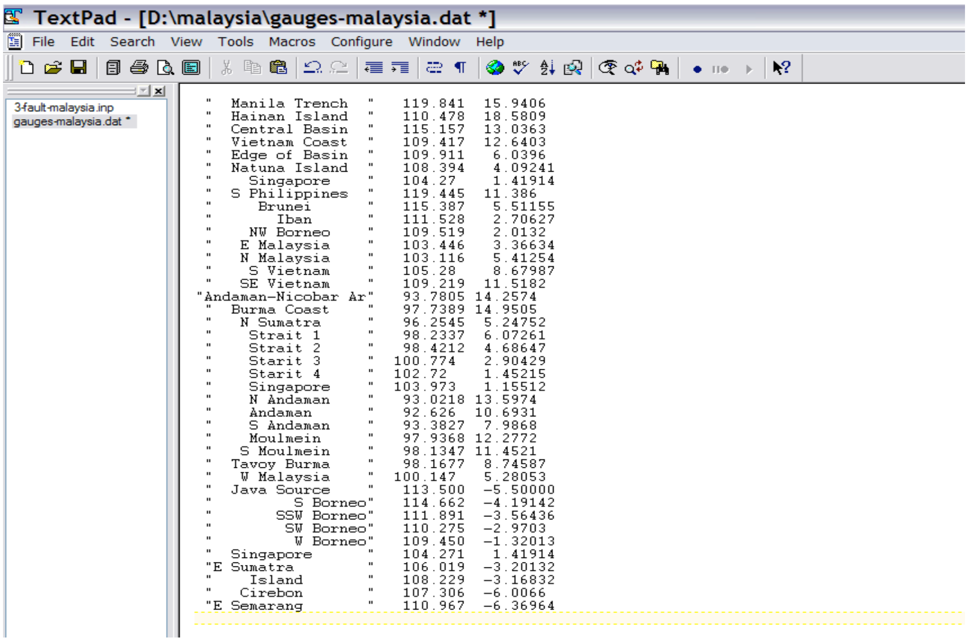
Other than the input file, the dimensional parameters in the FORTRAN code are needed to be modified according to the number of grids of the bathymetry file.

9.2. Output files

The output file of the program is the initial wave in the sea which contains the water surface elevations at each grid point of the bathymetry data. This output file which is usually named as fault***.grd is also one of the input files of the TUNAMI program.

10. GAUGES FILE CONCEPT

The gauge location data is saved as *.dat file and contains the names, x and y coordinates of the gauge locations within the area. Names of the locations will be written within quotation marks. Next, the x coordinate and the y coordinate of the location will be written in decimals. South and west coordinates will be shown with a minus sign.

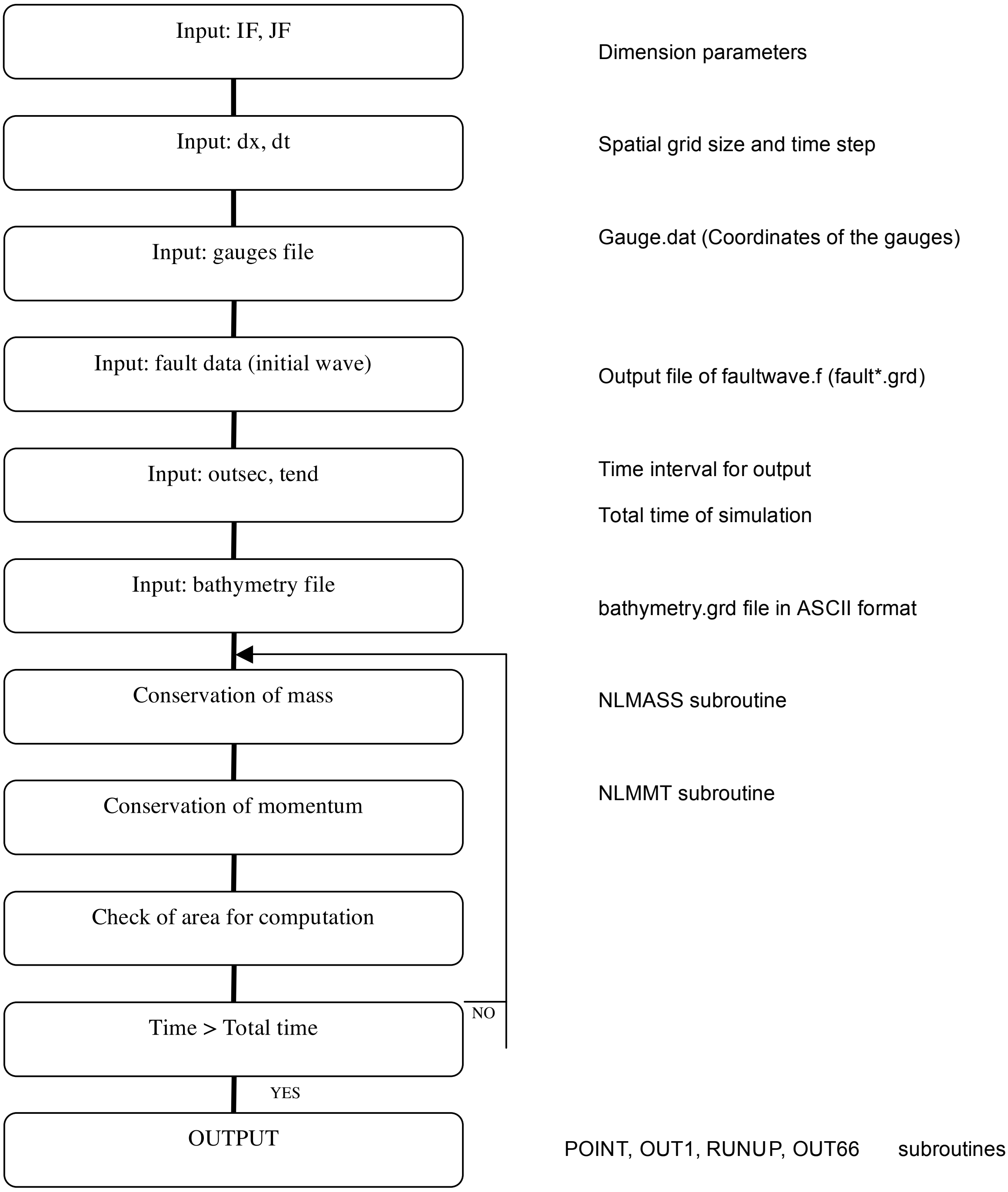

TextPad - [D:\malaysia\gauges-malaysia.dat *]
File Edit Search View Tools Macros Configure Window Help
3-fault-malaysia.inp
gauges-malaysia.dat *

```
 "  Manila Trench  "   119.841  15.9406
 "  Hainan Island  "   110.478  18.5809
 "  Central Basin  "   115.157  13.0363
 "  Vietnam Coast  "   109.417  12.6403
 "  Edge of Basin  "   109.911   6.0396
 "  Natuna Island  "   108.394   4.09241
 "    Singapore    "   104.27    1.41914
 "  S Philippines  "   119.445  11.386
 "     Brunei      "   115.387   5.51155
 "       Iban      "   111.528   2.70627
 "    NW Borneo    "   109.519   2.0132
 "   E Malaysia    "   103.446   3.36634
 "   N Malaysia    "   103.116   5.41254
 "    S Vietnam    "   105.28    8.67987
 "   SE Vietnam    "   109.219  11.5182
"Andaman-Nicobar Ar"   93.7805 14.2574
 "  Burma Coast    "   97.7389 14.9505
 "   N Sumatra     "   96.2545  5.24752
 "    Strait 1     "   98.2337  6.07261
 "    Strait 2     "   98.4212  4.68647
 "    Starit 3     "  100.774   2.90429
 "    Starit 4     "  102.72    1.45215
 "    Singapore    "  103.973   1.15512
 "    N Andaman    "   93.0218 13.5974
 "    Andaman      "   92.626  10.6931
 "    S Andaman    "   93.3827  7.9868
 "    Moulmein     "   97.9368 12.2772
 "   S Moulmein    "   98.1347 11.4521
 "  Tavoy Burma    "   98.1677  8.74587
 "   W Malaysia    "  100.147   5.28053
 "  Java Source    "   113.500  -5.50000
 "         S Borneo"   114.662  -4.19142
 "       SSW Borneo"   111.891  -3.56436
 "        SW Borneo"   110.275  -2.9703
 "         W Borneo"   109.450  -1.32013
 "  Singapore      "   104.271   1.41914
 "E Sumatra        "   106.019  -3.20132
 "    Island       "   108.229  -3.16832
 "   Cirebon       "   107.306  -6.0066
 "E Semarang       "   110.967  -6.36964
```

11. SOFTWARE TUNAMI N2

TUNAMI-N2 is a tsunami numerical simulation program with the linear theory in deep sez and with the shallow water theory in shallow sea and on land with constant grid length in the whole region. The run up can be computed with this program.

The basic procedure of the program can be seen from the flowchart given below. The detailed information of some parts is either given in previous chapters or will be given in following chapters.



11.1 Input files and variables

Dimension parameters are denoted as IF and JF. The values of the dimension parameters depend on the bathymetry grid file. IF is the number of rows and JF is the number of columns of the grid file.

Spatial grid size which is denoted as dx is the distance between two grids of the bathymetry file. During processing of the bathymetric data, the grid interval size is determined by the user.

Outsec is the time interval to output the sea state (the wave profile). This sea state is stored to t****.grd file tend is the total simulation time of the numerical analysis in minutes.

Gauges file, initial wave parameters and the bathymetry file are explained in detail in previous chapters.

11.2 Parameters

Z : wave surface level

M, N : water discharge

HM, HN : still water depth at point of water discharge

DM, DN : total water depth at point of water discharge

HZ : still water depth

ZD : total water depth

IF, JF : dimensional parameters

DX, DT : spatial grid and time step size

G : gravitational acceleration

KL : total time step

IR : map of wave-braker

BT :height of wave braker

IP, JP : position of output point

rip, rjp : coordinates of the output point in the map in cm.

ZM : maximum water level

xleft, xright, yleft, yright : limits of input file in degrees

tplate* : time where the fault* is broken in seconds. There is only one fault broken at initial time t = 0 sec. tplate2, tplate3 and tplate4 are given so that they do not break during simulation.

zmxsec : time when maximum elevations in the sea occurred

rupmin : time in minutes. At every rupmin, program calculates maximum water surface elevations near shoreline. It is written to files *.dat.

11.3 Subroutines

There are several subroutines of TUNAMI-N2 having different purposes. The list of the subroutines as well as the codes of the main ones is presented below.

READ – reads the depth data
INTL – the initial condition as input
HMN – calculates the water depth at point of discharge
NLMASS – conservation of mass
NLMMT – conservation of momentum
CHANGE – exchanges the last step data to next step data
OUT3 – output of data in a given format
OUT1 – output of water level and discharge
OUT6 – output of computed results on the display
mxMAX – checks the maximum value
POINT and POINTc – output of water level time histories
RUNUP – output of run up values

<u>List of program Codes (Linearized model)</u>

(a) Mass conservation equation (for linear)

```
      SUBROUTINE   MASS (IF,JF,Z,M,N,H,R)
C
C   Z; water surface level,        M,N; water discharge
C   H; still water depth                 R; ration of Dt to Dx
C   IF,JF: dimension
      REAL M,N
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
      DATA GX/1.0E-10/
```

```
          DO 10 J=2,JF
                    DO 10 I=2,IF
                    IF(H(I,J).LT.0.0)GO TO 10
                    ZZ=Z(I,J,1)-R*(M(I,J,1)-M(I-1,J,1)+N(I,J,1)-N(I,J-1,1))
                    IF(ABS(ZZ).LT.GX)ZZ=0.0
                    Z(I,J,2)=ZZ
   10   CONTINUE
          RETURN
           END
```

(b) Linearized momentum equation with bottom friction

```
      SUBROUTINE MOMENT(IF,JF,Z,M,N,H,R,G)
C      G; gravitational acces.   FM; Manning's roughness
C
          REAL M,N
          DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
          DATA GX/1.0E-10/
          FN=0.5*DT*G*FM**2
          DO 10 J=1,JF
          DO 10 I=1,IF
C        cal. of   Linear term (x-direction)
           IF(I.EQ.IF)GOTO 20
           IF(H(I,J).LT.0.0)GO TO 10
           IF(H(I+1,J).LT.0.0)GO TO 20
          IF(J.EQ.1)THEN
          XNN=0.5*(N(I,J,1)+N(I+1,J,1))
          ELSE
          XNN=0.25*(N(I,J,1)+N(I+1,J,1)+N(I,J-1,1)+N(I+1,J-1,1))
          ENDIF
          HM=0.5*(H(I,J)+H(I+1,J))
          FF=FN*SQRT(M(I,J,1)**2+XNN**2)/HM**(7/3.0)
           XM=(1-FF)*M(I,J,1)-G*R*HM*(Z(I+1,J,2)-Z(I,J,2))
           IF(ABS(XM).LT.GX)XM=0.0
           M(I,J,2)=XM
    20    IF(J.EQ.JF)GOTO 10
C        cal. of  Linear term (y-direction)
           IF(H(I,J+1).LT.0.0)GO TO 10
          IF(I.EQ.1)THEN
          XMM=0.5*(M(I,J,1)+M(I,J+1,1))
          ELSE
          XMM=0.25*(M(I,J,1)+M(I,J+1,1)+N(I-1,J,1)+N(I-1,J+1,1))
          ENDIF
          HN=0.5*(H(I,J)+H(I,J+1))
          FF=FN*SQRT(N(I,J,1)**2+XMM**2)/HN**(7/3.0)
          XN=(1-FF)*N(I,J,1)-G*R*HN*(Z(I,J+1,2)-Z(I,J,2))
          IF(ABS(XN).LT.GX)XN=0.0
           N(I,J,2)=XN
    10   CONTINUE
          RETURN
          END
```

(c) Mass conservation equation (for nonlinear)

```
        SUBROUTINE NLMASS(IF,JF,Z,M,N,DZ,HZ,R)
C          DZ; total depth
         REAL M,N
        DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
        DIMENSION DZ(IF,JF,2),HZ(IF,JF)
        DATA GX,GY/1.0E-5,1.0E-10/
        DO 10 J=2,JF
        DO 10 I=2,IF
        IF(HZ(I,J).LT.-9.9)GOTO 11
        ZZZ=Z(I,J,1)-R*(M(I,J,1)-M(I-1,J,1)+N(I,J,1)-N(I,J-1,1))
        IF(ABS(ZZZ).LT.GY)ZZZ=0.0
        DD=ZZZ+HZ(I,J)
```

```
            IF(DD.LT.GX)GOTO 11
            DZ(I,J,2)=DD
            Z(I,J,2)=ZZZ
            GOTO 10
   11     DD=0.0
            DZ(I,J,2)=DD
            Z(I,J,2)=DD-HZ(I,J)
   10 CONTINUE
        RETURN
        END
```

(d) Nonlinear momentum equation with bottom friction

```
      SUBROUTINE NLMMT(IF,JF,Z,M,N,DZ,HZ,HM,HN,R,DT,FM)
C           G; gravitational acces.     FM; Manning's roughness
        REAL M,N
        DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
        DIMENSION HZ(IF,JF),HM(IF,JF),HN(IF,JF)
        PARAMETER (NET=3)
        DATA GG,GX/9.8,1.0E-5/
C        ------ CAL. OF TOTAL DEPTH AT POINT OF DISCHARGE -------
        DO 10 I=1,IF
          DO 10 J=1,JF
            IF(I.LT.IF)THEN
            DM2=0.5*(DZ(I,J,2)+DZ(I+1,J,2))
            DM1=0.25*(DZ(I,J,2)+DZ(I,J,1)+DZ(I+1,J,2)+DZ(I+1,J,1))
            ENDIF
            IF(J.LT.JF)THEN
            DN2=0.5*(DZ(I,J,2)+DZ(I,J+1,2))
            DN1=0.25*(DZ(I,J,2)+DZ(I,J,1)+DZ(I,J+1,2)+DZ(I,J+1,1))
            ENDIF
            IF(DM1.LT.GX)DM1=0.0
            IF(DM2.LT.GX)DM2=0.0
            IF(DN1.LT.GX)DN1=0.0
            IF(DN2.LT.GX)DN2=0.0
            DM(I,J,1)=DM1
            DM(I,J,2)=DM2
            DN(I,J,1)=DN1
            DN(I,J,2)=DN2
     10 CONTINUE
C    ------- CAL. OF LINEAR TERMS (X-DIRECTION) -------
        FN=0.5*DT*GG*FM**2
        DO 200 I=1,IF
          DO 200 J=1,JF
            DF=DM(I,J,2)
            IF(J.EQ.1)THEN
            XNN=0.5*(N(I,J,1)+N(I+1,J,1))
            ELSE
            XNN=0.25*(N(I,J,1)+N(I+1,J,1)+N(I,J-1,1)+N(I+1,J-1,1))
            ENDIF
            IF(DF.LT.1.0E-2)DF=1.0E-2          ----- minimum depth
            FF=FN*SQRT(M(I,J,1)**2+XNN**2)/DF**(7.0/3.0)
            IF(DD.LT.GX)GO TO 30
            XM=(1.0-FF)*M(I,J,1)-GG*R*DD*(Z(I+1,J,2)-Z(I,J,2))
C   ----- CAL. OF NON-LINEAR TERMS (CONVECTION TERMS) ------
         IF(DM(I,J,1).LT.GX)GO TO 40
            IF(M(I,J,1))41,41,42
    41     IF(DM(I+1,J,1).LT.GX)GO TO 1001
            IF(DZ(I+2,J,2).LT.GX)GO TO 1001
            IF(DZ(I+1,J,2).LT.GX)GO TO 1001
            XM=XM-R*(M(I+1,J,1)**2/DM(I+1,J,1)-M(I,J,1)**2/DM(I,J,1))
            GO TO 43
   1001     XM=XM-R*(-M(I,J,1)**2/DM(I,J,1))
            GO TO 43
     42     IF(DM(I-1,J,1).LT.GX)GO TO 1002
            IF(DZ(I-1,J,2).LT.GX)GO TO 1002
            IF(DZ(I,J,2).LT.GX)GO TO 1002
            XM=XM-R*(M(I,J,1)**2/DM(I,J,1)-M(I-1,J,1)**2/DM(I-1,J,1))
            GO TO 43
   1002     XM=XM-R*(M(I,J,1)**2/DM(I,J,1))
     43     IF(XNN)44,44,45
     44     XNE=0.25*(N(I,J+1,1)+N(I+1,J+1,1)+N(I,J,1)+N(I+1,J,1))
            IF(DM(I,J+1,1).LT.GX)GO TO 1003
            IF(DZ(I,J+1,2).LT.GX)GO TO 1003
            IF(DZ(I,J+2,2).LT.GX)GO TO 40
            IF(DZ(I+1,J+1,2).LT.GX)GO TO 40
            IF(DZ(I+1,J+2,2).LT.GX)GO TO 40
            XM=XM-R*(M(I,J+1,1)*XNE/DM(I,J+1,1)-M(I,J,1)*XNN/DM(I,J,1))
            GO TO 40
   1003     XM=XM-R*(-M(I,J,1)*XNN/DM(I,J,1))
            GO TO 40
     45     XNE=0.25*(N(I,J-1,1)+N(I+1,J-1,1)+N(I,J-2,1)+N(I+1,J-2,1))
            IF(DM(I,J-1,1).LT.GX)GO TO 1004
            IF(DZ(I,J-2,2).LT.GX)GO TO 1004
            IF(DZ(I,J-1,2).LT.GX)GO TO 40
            IF(DZ(I+1,J-1,2).LT.GX)GO TO 40
            IF(DZ(I+1,J-2,2).LT.GX)GO TO 40
            XM=XM-R*(M(I,J,1)*XNN/DM(I,J,1)-M(I,J-1,1)*XNE/DM(I,J-1,1))
            GO TO 40
   1004     XM=XM-R*(M(I,J,1)*XNN/DM(I,J,1))
     40     XM=XM/(1.0+FF)
            IF(ABS(XM).LT.1.0E-10)XM=0.0
            M(I,J,2)=XM
            GO TO 100
     30     M(I,J,2)=0.0
    100     CONTINUE
C    ------- CAL. OF LINEAR TERMS (Y-DIRECTION) -------
            IF(J.EQ.JF)GO TO 130
         DF=DN(I,J,2)
```

```
              IF(I.EQ.1)THEN
             XMM=0.5*(M(I,J,1)+M(I,J+1,1))
             ELSE
             XMM=0.25*(M(I,J,1)+M(I,J+1,1)+M(I-1,J,1)+M(I-1,J+1,1))
             ENDIF
             IF(DF.LT.1.0E-2)DF=1.0E-2
             FF=FN*SQRT(N(I,J,1)**2+XMM**2)/DF**(7.0/3.0)
             IF(DD.LT.GX)GO TO 130
             XN=(1.0-FF)*N(I,J,1)-GG*R*DD*(Z(I,J+1,2)-Z(I,J,2))
C   ----- CAL. OF NON-LINEAR TERMS (CONVECTION TERMS) ------
             IF(DN(I,J,1).LT.GX)GO TO 140
             IF(N(I,J,1))141,141,142
   141       IF(DN(I,J+1,1).LT.GX)GO TO 1011
             IF(DZ(I,J+2,2).LT.GX)GO TO 1011
             IF(DZ(I,J+1,2).LT.GX)GO TO 1011
             XN=XN-R*(N(I,J+1,1)**2/DN(I,J+1,1)-N(I,J,1)**2/DN(I,J,1))
             GO TO 143
  1011       XN=XN-R*(-N(I,J,1)**2/DN(I,J,1))
             GO TO 143
   142       IF(DN(I,J-1,1).LT.GX)GO TO 1012
             IF(DZ(I,J-1,2).LT.GX)GO TO 1012
             IF(DZ(I,J,2).LT.GX)GO TO 1012
             XN=XN-R*(N(I,J,1)**2/DN(I,J,1)-N(I,J-1,1)**2/DN(I,J-1,1))
             GO TO 143
  1012       XN=XN-R*(N(I,J,1)**2/DN(I,J,1))
   143       IF(XMM)144,144,145
   144       XME=0.25*(M(I+1,J,1)+M(I+1,J+1,1)+M(I,J,1)+M(I,J+1,1))
             IF(DN(I+1,J,1).LT.GX)GO TO 1013
             IF(DZ(I+1,J,2).LT.GX)GO TO 1013
             IF(DZ(I+2,J,2).LT.GX)GO TO 140
             IF(DZ(I+1,J+1,2).LT.GX)GO TO 140
             IF(DZ(I+2,J+1,2).LT.GX)GO TO 140
             XN=XN-R*(N(I+1,J,1)*XME/DN(I+1,J,1)-N(I,J,1)*XMM/DN(I,J,1))
```

```
          GO TO 140
 1013      XN=XN-R*(-N(I,J,1)*XMM/DN(I,J,1))
          GO TO 140
  145      XME=0.25*(M(I-1,J,1)+M(I-1,J+1,1)+M(I-2,J,1)+M(I-2,J+1,1))
          IF(DN(I-1,J,1).LT.GX)GO TO 1014
          IF(DZ(I-2,J,2).LT.GX)GO TO 1014
          IF(DZ(I-2,J+1,2).LT.GX)GO TO 140
          IF(DZ(I-1,J,2).LT.GX)GO TO 140
          IF(DZ(I-1,J+1,2).LT.GX)GO TO 140
          XN=XN-R*(N(I,J,1)*XMM/DN(I,J,1)-N(I-1,J,1)*XME/DN(I-1,J,1))
          GO TO 140
 1014      XN=XN-R*(N(I,J,1)*XMM/DN(I,J,1))
  140      XN=XN/(1.0+FF)
          IF(ABS(XN).LT.1.0E-10)XN=0.0
          N(I,J,2)=XN
          GO TO 200
  130      N(I,J,2)=0.0
  200 CONTINUE
      RETURN
      END
```

(e) Run up along a coastline

```
     SUBROUTINE RUNUP(IF,JF,ZM,RUI,RUJ,DX,HZ,TM,III,JJJ,
     *XL,XR,YB,YT,XINC,YINC,NR1,NR2,NR3,NR4,NR5,NR6,NR7,NR8,NR9,
     *NR10,NR11,NR12)

           DIMENSION ZM(IF,JF),RUI(IF,5),RUJ(JF,5),HZ(IF,JF),TM(IF,JF)
      DIMENSION III(IF,2),JJJ(JF,2)
      DY=DX
           WRITE(*,2205)
           WRITE(NR1,2205)
```

```
              NNN=-1
C        X=98.962     X=102.882
          JSTART=(9.099-YB)/YINC
              JEND=(1.612-YB)/YINC
          ISTART=(98.962-XL)/XINC+1
              DO 399 J=JSTART,JEND,-1
C             WRITE(*,*)J
              NNN=NNN+1
              RUJ(J,1)=YB+(J-1)*YINC
          RMAX=-10000
          YJ=(J-1)*YINC+YB

             ISTART=ISTART+1
C             WRITE(*,*) ISTART
C           IF(YJ.LE.3)ISTART=+(96.7-XL)/XINC+1
C             SAYI=(J-1)*YINC+YB
C             IF(SAYI.LT.40.6)ISTART=(29.15-XL)/XINC+1
C           WRITE(*,*)J
          DO 3991 I=ISTART,ISTART-100,-1
C           WRITE(*,*)'ISTART',ISTART,I, HZ(I,J),ZM(I,J)
C           WRITE(*,*)I,J,HZ(I,J)

C           IF(J.LE.60.AND.I.GT.(IF-10))GO TO 4399
C           IF(J.LE.60.AND.I.LT.(IF-10))GO TO 1399
C           WRITE(*,*)I,J
          IF (HZ(I,J).GT.300.)GO TO 1399
C           WRITE(*,*)I,J
          IF(ZM(I,J).LT.RMAX) GO TO 3991
C IF(SAYI.LE.40.7838.AND.SAYI.GE.40.7056.AND.HZ(I,J).GT.10.)
C        *GO TO 1399

          RUJ(J,2)=ZM(I,J)
          RUJ(J,3)=TM(I,J)
```

```
      RUJ(J,5)=YB+(J-1)*YINC
      RUJ(J,4)=XL+(I-1)*XINC
      JJJ(J,1)=I
      JJJ(J,2)=J
      RMAX=ZM(I,J)
C     WRITE(*,*)I,J,ZM(I,J),JJJ(I,1),JJJ(I,2)
C     WRITE(*,*)RUJ(I,1),RUJ(I,2),RUJ(I,3),RUJ(I,4),RUJ(I,5),
C     *HZ(JJJ(I,1),JJJ(I,2))
3991  CONTINUE
1399  CONTINUE
C     WRITE(*,*)I,J
C     WRITE(*,*)I,J,JJJ(J,1),JJJ(J,2),HZ(I,J),ZM(I,J),RMAX
C     WRITE(*,202)RUJ(J,1),RUJ(J,2),RUJ(J,3),RUJ(J,4),RUJ(J,5)
      WRITE(NR1,202)RUJ(J,1),RUJ(J,2),RUJ(J,3),RUJ(J,4),RUJ(J,5),
     *HZ(JJJ(J,1),JJJ(J,2))
 399 CONTINUE
     WRITE(*,*)'     RUNUP ALONG WEST COAST OF THAILAND IS OK'
```

11.4 Output Files of TUNAMI-N2

There are several output files which are mostly grid files showing sea state at different times and locations. These files are:

t*****.grd files: these files show the sea state at a specified instant. Stars in the file name represents the time at which the sea state is stored. These files are generated according to the time interval given as outsec from the beginning of the simulation until the total simulation time, tend.

elevsurface.dat: shows the time histories of water surface elevations of the locations that are given in the gauges file.

elevcurrent.dat: shows the time histories of current velocities at the locations that are given in the gauges file.

azmax.grd: shows the maximum water level at each grid during simulation at time = tend. Using this file and coordinates of the start and end points of a specified coastline, the runup subroutine as a separate program, calculates the runup levels and the results are stored in r-u-name.dat file.

azmin.grd: shows the minimum water level at each grid during simulation at time = tend.

Curmax.grd: shows the maximum current velocity at each grid during simulation at time = tend.

azWH.grd: shows the wave height at each grid during simulation till time = tend.

12. INTERPRETATION OF TUNAMI-N2 OUTPUT FILES

The output of TUNAMI-N2 program is not easily interpreted as given in the output files. In order to get the best results, the output files are needed to be converted in to diagrams or graphs so that the interpretation and the comparison of different data can be achieved easily.

Using t***.grd files and the bathymetry file of the area, the simulation of the tsunami is accomplished. Plotting the elevsurface.dat for specific coordinates where the tidal gauges are located, comparison of the real time data and the numerical analysis of the tsunami can be done which is a way of checking the accuracy of the simulation.

Plotting the runup file enables another check of accuracy for the simulation and the input parameters by comparing the post tsunami survey measurements and the output of the runup routine.

The diagrams or the graphs of the output files are mostly prepared by using the programs Surfer, Grapher and/or Microsoft Office Excel. To learn about these programs, please refer to the help menus of each software.

13. SAMPLE AND REAL-TIME APPLICATIONS

13.1 Pool Application

A grid data is created as below in order to present some of the output files in simple form.

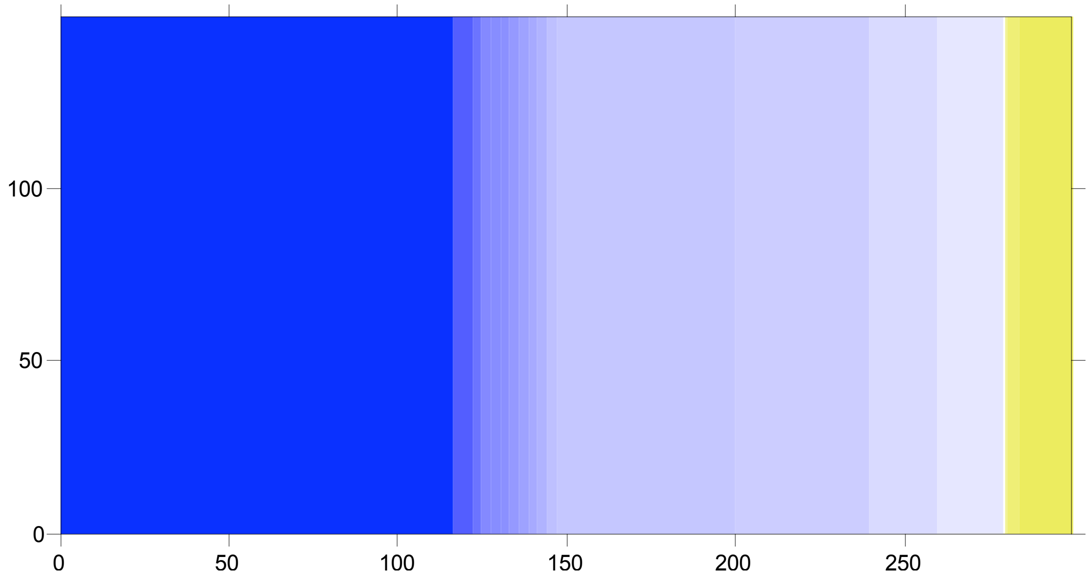


Figure 13.1 The bathymetry of the sample application by Surfer

Using the below parameters an initial wave is created for this sample.

128.8	36.2358	Start point of the fault
73.73	97.4	End point of the fault
5000.00	Width.	
300.00	THETA.	
20.00	DL .	
45.00	RD .	
20.00	D .	
10000.00	HH .	

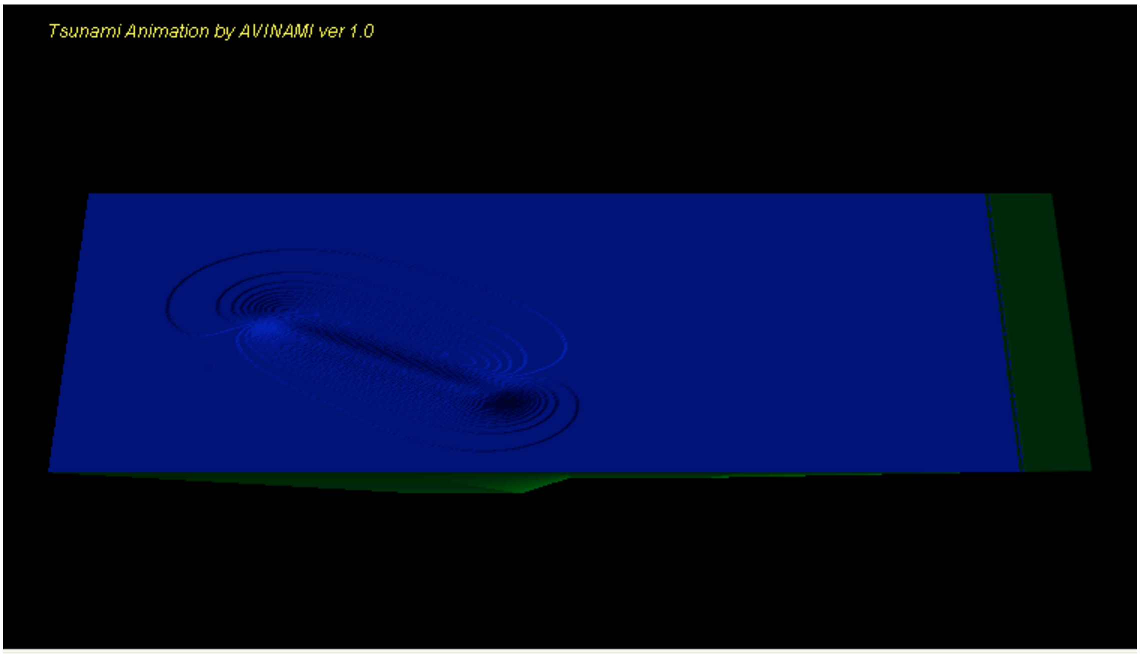
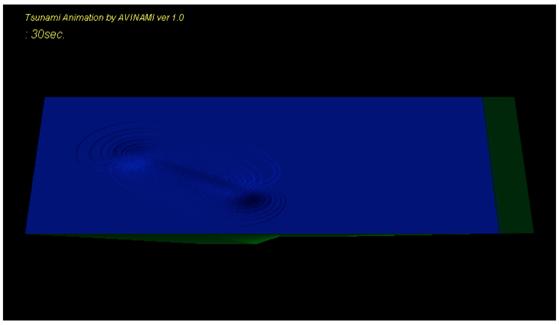
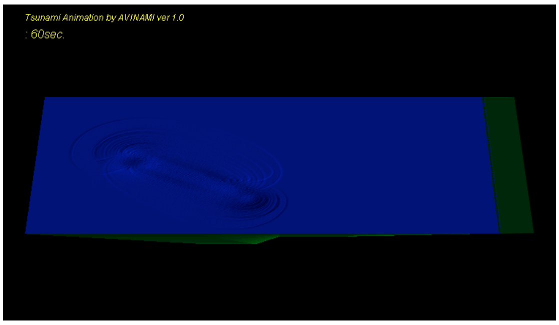
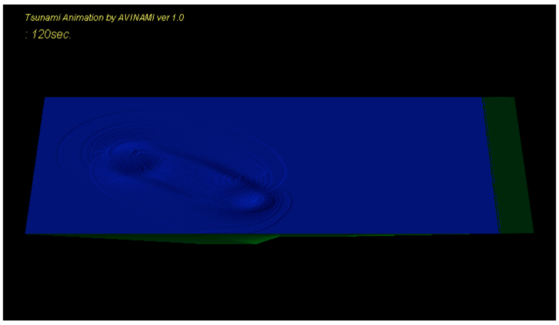


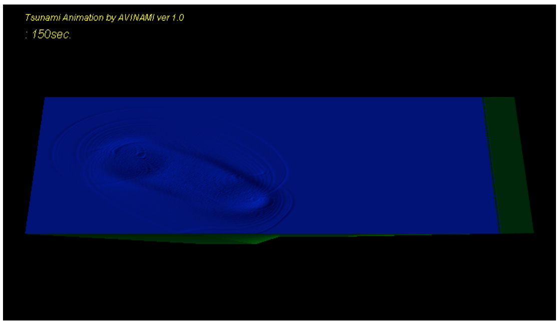
Figure 13.2 The initial wave created by using the above parameters using AVINAMI

The propagation of the initial wave can be plotted by using the output files at any time interval. Some examples for this sample application are given below.


Tsunami Animation by AVINAMI ver 1.0
: 30sec.


Tsunami Animation by AVINAMI ver 1.0
: 60sec.


Tsunami Animation by AVINAMI ver 1.0
: 120sec.


Tsunami Animation by AVINAMI ver 1.0
: 150sec.