Intergovernmental
Oceanographic
Commission

Manuals and Guides

35

IUGG/IOC TIME PROJECT

NUMERICAL METHOD OF TSUNAMI SIMULATION WITH THE LEAP-FROG SCHEME

PART 1 SHALLOW WATER THEORY AND ITS DIFFERENCE SCHEME

PART 2 PROPAGATION IN THE OCEAN IN THE SPHERICAL CO-ORDINATES TSUNAMI-F1 AND ITS PROGRAMME LIST

by C. Goto and Y. Ogawa
Department of Civil Engineering
Faculty of Engineering
Tohoku University
Sendai 980-77 - Japan

Translated and prepared for the TIME PROJECT
by N. Shuto
Disaster Control Research Center
Faculty of Engineering
Tohoku University
Sendai 980-77 - Japan

PART 3 PROGRAMME LISTS FOR NEAR-FIELD TSUNAMI

PART 4 PROGRAMME LIST FOR FAR-FIELD TSUNAMI

by F. Imamura
Tohoku University
Sendai 980-77 - Japan

1997 UNESCO

SC-97/WS-37

For bibliographic purposes
this document should be cited as follows:

IUGG/IOC Time Project
IOC Manuals and Guides No. 35. UNESCO 1997
(English)

Published in 1997
by the United Nations Educational,
Scientific and Cultural Organization
7, place de Fontenoy, 75352 Paris 07 SP, France

Printed in UNESCO's Workshops

Printed in France

IUGG/IOC TIME PROJECT

PART 1: SHALLOW WATER THEORY AND ITS DIFFERENCE SCHEME

CHAPTER 1: METHOD OF NUMERICAL SIMULATION

CHAPTER 2: TYPICAL PROGRAMMES

CONTENTS

CHAPTER 1: METHOD OF NUMERICAL SIMULATION

CHAPTER 2: TYPICAL PROGRAMMES

CHAPTER 1. METHOD OF NUMERICAL SIMULATION

1.1 EQUATIONS AND DIFFERENCE SCHEMES

(1) Shallow Water Theory

Water waves are classified into many types from a hydraulic point of view. This paper considers tsunamis, which are generated by the movement of sea bottom due to earthquakes. Tsunamis belong to long waves.

The theory of long waves is an approximate theory applicable to waves of small relative depth (the ratio of water depth to wave length), for which the vertical acceleration of water particles is negligible compared to the gravitational acceleration and the curvature of trajectories of water particles is sufficiently small. Consequently, the vertical motion of water particles has no effect on the pressure distribution. It is a good approximation that the pressure is hydrostatic. In addition, the horizontal velocity of water particles are vertically uniform.

Based upon these approximations, the motion of long waves is well expressed by the following shallow water theory.

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+g\frac{\partial \eta}{\partial x}+\frac{\tau_x}{\rho}=0$$

$$\frac{\partial \eta}{\partial t}+\frac{\partial[u(h+\eta)]}{\partial x}+\frac{\partial[v(h+\eta)]}{\partial y}=0$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+g\frac{\partial \eta}{\partial y}+\frac{\tau_y}{\rho}=0$$

where x and y are horizontal coordinates, t time, h the still water depth, η the vertical displacement of water surface above still water level, u and v water particle velocities in the x- and y-directions, g the gravitational acceleration, and τ_x/ρ and τ_y/ρ bottom frictions in the x- and y-directions.

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

$$\frac{\tau_x}{\rho}=\frac{1}{2g}\frac{f}{D}u\sqrt{u^2+v^2} \quad , \quad \frac{\tau_y}{\rho}=\frac{1}{2g}\frac{f}{D}v\sqrt{u^2+v^2}$$

where D is the total water depth given by h + η and f is the friction coefficient. Without any detailed discussion of the value of the authors prefer to use Manning's roughness n which are familiar among civil engineers.

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

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

and, the bottom frictions are finally expressed by

$$\frac{\tau_x}{\rho}=\frac{gn^2}{D^{4/3}}u\sqrt{u^2+v^2} \quad , \quad \frac{\tau_y}{\rho}=\frac{gn^2}{D^{4/3}}v\sqrt{u^2+v^2}$$

throughout the present paper.

The next step is the introduction of discharge fluxes (M, N) in the x- and y-directions. M and N are related to u and v by the following expressions.

$$M=u(h+\eta)=uD \quad , \quad N=v(h+\eta)=vD$$

Integrating Eqs.(1) through (3) from sea bottom to water surface, the following shallow water theory is obtained for discharge fluxes M and N.

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

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

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

These equations are the fundamental equations used in the present paper.

The authors would not like to recommend the use of Eqs.(1) through (3) but the use of Eqs.(6) through (8). When discretized, the former set of equations sometimes does not satisfy the conservation of mass. This fact which is often disregarded becomes the source of large computation errors. On the other hand, the latter set of equations, Eqs.(6) through (8), has no defect in the conservation of mass, and it also satisfies the conservation of momentum fairly well.

(2) Difference Scheme and Stability

The leap-frog scheme used in the present paper is a central difference scheme with the truncation error of the second order.

Values of F(x) at discrete points with an spatial interval Δx are expressed as follows, on referring Figure 1.

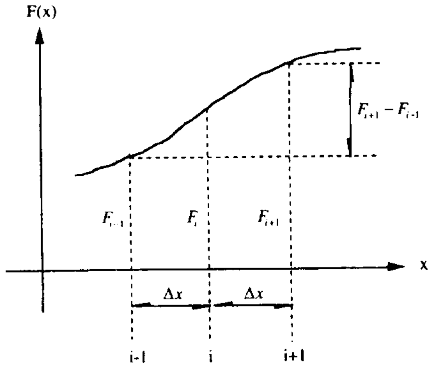


Figure 1-1. Central difference

$$F\{(i-1)\Delta x\}=F_{i-1}\ ,\ F(i\Delta x)=F_i\ ,\ F\{(i+1)\Delta x\}=F_{i+1}$$

The Taylor series expressions for $_{\text{If-1}}$ and $_{\text{If+1}}$ are

$$F_{i-1}=F_i-\Delta x\left.\frac{\partial F}{\partial x}\right|_i+\frac{(\Delta x)^2}{2}\left.\frac{\partial^2 F}{\partial x^2}\right|_i-\frac{(\Delta x)^3}{6}\left.\frac{\partial^3 F}{\partial x^3}\right|_i+\frac{(\Delta x)^4}{24}\left.\frac{\partial^4 F}{\partial x^4}\right|_i+O(\Delta x^5)$$

$$F_{i+1}=F_i+\Delta x\left.\frac{\partial F}{\partial x}\right|_i+\frac{(\Delta x)^2}{2}\left.\frac{\partial^2 F}{\partial x^2}\right|_i+\frac{(\Delta x)^3}{6}\left.\frac{\partial^3 F}{\partial x^3}\right|_i+\frac{(\Delta x)^4}{24}\left.\frac{\partial^4 F}{\partial x^4}\right|_i+O(\Delta x^5)$$

and the difference between the two Taylor series expressions yield a central difference of the first-order derivative as Eq.(9).

$$\left.\frac{\partial F}{\partial x}\right|_i = \frac{1}{2\Delta x}\left[F_{i+1} - F_{i-1}\right] + O(\Delta x^2)$$

Apply the above difference scheme to Eqs.(6) through (8). In order to make it easy to set the boundary conditions, the leap-frog scheme assumes that the computation point for η does not coincide with the computation point for M and N, as shown in Figure-2. In Figure1-2, suffixes (I, j, k) are used to express the spatial position (x, y) and the time t.

First, the equation of the continuity is approximated by a difference equation. With the central difference scheme, three terms in Eq.(6) are given by

$$\frac{\partial \eta}{\partial t} = \frac{1}{\Delta t}\left[\eta_{i,j}^{k+1} - \eta_{i,j}^{k}\right]$$

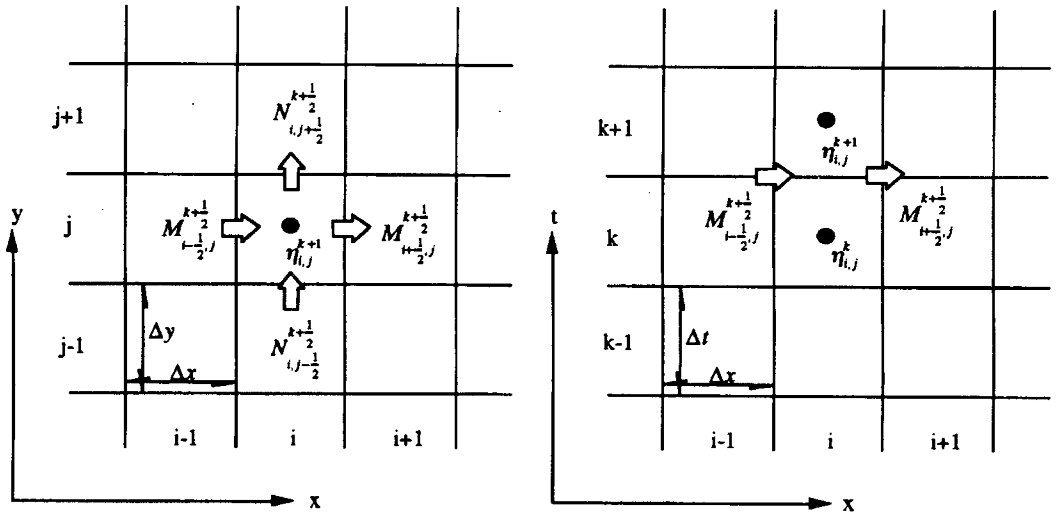


Figure 1-2. Arrangement of points for computation in the leap-frog method.

$$\frac{\partial M}{\partial x} = \frac{1}{\Delta x}\left| M_{i+\frac{1}{2},j}^{k+\frac{1}{2}} - M_{i-\frac{1}{2},j}^{k+\frac{1}{2}} \right|$$

$$\frac{\partial N}{\partial y} = \frac{1}{\Delta y}\left[N_{i,j+\frac{1}{2}}^{k+\frac{1}{2}} - N_{i,j-\frac{1}{2}}^{k+\frac{1}{2}} \right]$$

On assuming that values at k and k+1/2 time steps are known, the only unknown η(I, j, k+1) is solved by

$$\eta_{i,j}^{k+1} = \eta_{i,j}^{k} - \frac{\Delta t}{\Delta x}\left| M_{i+\frac{1}{2},j}^{k+\frac{1}{2}} - M_{i-\frac{1}{2},j}^{k+\frac{1}{2}} \right| - \frac{\Delta t}{\Delta y}\left| N_{i,j+\frac{1}{2}}^{k+\frac{1}{2}} - N_{i,j-\frac{1}{2}}^{k+\frac{1}{2}} \right|$$

Second, the equation of motion is approximated. Explanation consists of three parts; approximation of linear terms, approximation and stability of convection terms, and approximation of friction terms.

The linear equation of motion of in the x-direction is written as

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

A central difference at the point (I+1/2, j, k) yields the following equation for an unknown M(I+1/2, j, k-1/2)

$$M^{k+\frac{1}{2}}_{i+\frac{1}{2}j}=M^{k-\frac{1}{2}}_{i+\frac{1}{2}j}-gD^{k}_{i+\frac{1}{2}j}\frac{\Delta t}{\Delta x}\left[\eta^{k}_{i+1,j}-\eta^{k}_{i,j}\right]$$

where the total water depth D(I+1/2, j, k) is expressed by

$$D^{k}_{i+\frac{1}{2}j}=h_{i+\frac{1}{2}j}-\eta^{k}_{i+\frac{1}{2}j}=h_{i+\frac{1}{2}j}+\frac{1}{2}\left[\eta^{k}_{i+1,j}+\eta^{k}_{i,j}\right]$$

The similar manipulation yields the following difference equation for the linear equation of motion in the y-direction.

$$N^{k+\frac{1}{2}}_{i,j+\frac{1}{2}}=N^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}-gD^{k}_{i,j-\frac{1}{2}}\frac{\Delta t}{\Delta y}\left[\eta^{k}_{i,j+1}+\eta^{k}_{i,j}\right]$$

$$D^{k}_{i,j+\frac{1}{2}}=h_{i,j+\frac{1}{2}}+\eta^{k}_{i,j+\frac{1}{2}}=h_{i,j+\frac{1}{2}}\frac{1}{2}\left[\eta^{k}_{i,j+1}+\eta^{k}_{i,j}\right]$$

It is now possible to solve Eqs.(10), (12) and (14) simultaneously and obtain the solution of linear long waves. A comment is necessary to explain a difference between the original equations and Eqs.(12) and (14). The original equations of linear long waves uses h (still water depth), but Eqs.(12) and (14) use D (total water depth). If h is sufficiently larger then η, a linear computation with Eqs.(12) and (14) can yield reliable results, It should be kept in mind that this linear computation may become unstable if h is smaller than η.

In the leap-frog scheme, an upwind 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 briefly explained by taking a simple convection equation in the following.

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

Here, the coefficient C is the propagation velocity and is assumed constant. The arrangement of computation points in the leap-frog scheme requires the forward difference scheme for the first-order time derivative (see Figure-3). This yields

$$\frac{\partial F}{\partial t}=\frac{1}{\Delta t}\left[F^{k+\frac{1}{2}}_{i+\frac{1}{2}}-F^{k+\frac{1}{2}}_{i+\frac{1}{2}}\right]-\frac{\Delta t}{2}\frac{\partial^{2}F}{\partial t^{2}}+O(\Delta t^{2})$$

The central difference is applied to the space derivative.

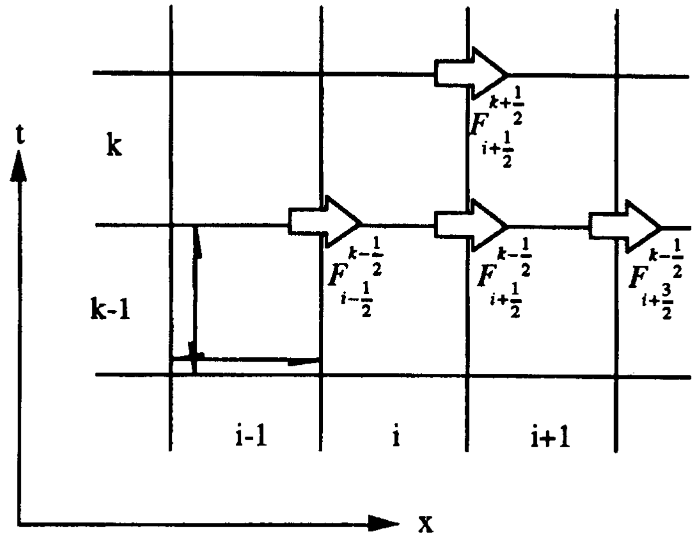


Figure 1-3. Arrangement of points for convection terms.

$$C\frac{\partial F}{\partial x}=\frac{C}{2\Delta x}\left[F_{i+\frac{3}{2}}^{k-\frac{1}{2}}-F_{i-\frac{1}{2}}^{k-\frac{1}{2}}\right]+O(\Delta x^{2})$$

As a result, the unknown F(I+1/2, k+1/2) is given by Eq.(17)

$$F_{i+\frac{1}{2}}^{k+\frac{1}{2}}=F_{i+\frac{1}{2}}^{k-\frac{1}{2}}-C\frac{\Delta t}{2\Delta x}\left[F_{i+\frac{3}{2}}^{k-\frac{1}{2}}-F_{i-\frac{1}{2}}^{k-\frac{1}{2}}\right]$$

Or, inversely speaking, the solution of Eq.(17) is implicitly equivalent the solution of Eq.(18) with an truncation error of ($\Delta x^2+\Delta y^2$)

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

If the second-order derivative with respect to time is rewritten by using the following relationship,

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

the solution of Eq.(18) is the same as the solution of the following diffusion equation in which the diffusion coefficient is negative.

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

A negative diffusion works to collect round-off errors to lead to an instability. Therefore, Eq.(17) is an unstable difference scheme.

In order to obtain a stable scheme, the space derivative term is approximated either by a forward or by a backward difference. With the forward difference, we have

$$C\frac{\partial F}{\partial x}=\frac{C}{\Delta x}\left[F_{i+\frac{3}{2}}^{k-\frac{1}{2}}-F_{i+\frac{1}{2}}^{k-\frac{1}{2}}\right]-\frac{\Delta x}{2}C\frac{\partial^2 F}{\partial x^2}+O(\Delta x^2)$$

and with the backward difference

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

The corresponding differential equations we are going to solve are, within the truncation error of $(\Delta x^2+\Delta y^2)$, for the forward difference

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

and for the backward difference

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

Therefore, in order to keep the virtual diffusion coefficient positive (or say, in order to ensure the stability of the computation), we have use the backward difference in case of positive C, and the forward difference in case of negative C, in addition to set $\Delta x/\Delta t \geq C$.

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. It is usually said that the leap-frog scheme has the truncation error of the order of Δx^2. However, as long as the convection term concerns, the truncation error is of the order of Δx.

The convection terms in Eqs.(7) and (8) are expressed as follows with the upwind scheme mentioned above.

$$\frac{\partial}{\partial x}\left(\frac{M^2}{D}\right)=\frac{1}{\Delta x}\left[\lambda_{11}\frac{\left(M_{i+\frac{3}{2}j}^{k-\frac{1}{2}}\right)^2}{D_{i+\frac{3}{2}j}^{k-\frac{1}{2}}}+\lambda_{21}\frac{\left(M_{i+\frac{1}{2}j}^{k-\frac{1}{2}}\right)^2}{D_{i+\frac{1}{2}j}^{k-\frac{1}{2}}}+\lambda_{31}\frac{\left(M_{i-\frac{1}{2}j}^{k-\frac{1}{2}}\right)^2}{D_{i-\frac{1}{2}j}^{k-\frac{1}{2}}}\right]$$

$$\frac{\partial}{\partial y}\left(\frac{MN}{D}\right)=\frac{1}{\Delta y}\left[\nu_{11}\frac{M^{k-\frac{1}{2}}_{i+\frac{1}{2},j+1}N^{k-\frac{1}{2}}_{i+\frac{1}{2},j+1}}{D^{k-\frac{1}{2}}_{i+\frac{1}{2},j+1}}+\nu_{21}\frac{M^{k-\frac{1}{2}}_{i+\frac{1}{2},j}N^{k-\frac{1}{2}}_{i+\frac{1}{2},j}}{D^{k-\frac{1}{2}}_{i+\frac{1}{2},j}}+\nu_{31}\frac{M^{k-\frac{1}{2}}_{i+\frac{1}{2},j-1}N^{k-\frac{1}{2}}_{i+\frac{1}{2},j-1}}{D^{k-\frac{1}{2}}_{i+\frac{1}{2},j-1}}\right]$$

$$\frac{\partial}{\partial x}\left(\frac{MN}{D}\right)=\frac{1}{\Delta x}\left[\lambda_{12}\frac{M^{k-\frac{1}{2}}_{i+1,j+\frac{1}{2}}N^{k-\frac{1}{2}}_{i+1,j+\frac{1}{2}}}{D^{k-\frac{1}{2}}_{i+1,j+\frac{1}{2}}}+\lambda_{22}\frac{M^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}N^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}}{D^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}}+\lambda_{32}\frac{MN}{D}\frac{M^{k-\frac{1}{2}}_{i-1,j+\frac{1}{2}}N^{k-\frac{1}{2}}_{i-1,j+\frac{1}{2}}}{D^{k-\frac{1}{2}}_{i-1,j+\frac{1}{2}}}\right]$$

$$\frac{\partial}{\partial y}\left(\frac{N^2}{D}\right)=\frac{1}{\Delta y}\left[\nu_{12}\frac{\left(N^{k-\frac{1}{2}}_{i,j+\frac{3}{2}}\right)^2}{D^{k-\frac{1}{2}}_{i,j+\frac{3}{2}}}+\nu_{22}\frac{\left(N^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}\right)^2}{D^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}}+\nu_{32}\frac{\left(N^{k-\frac{1}{2}}_{i,j-\frac{1}{2}}\right)^2}{D^{k-\frac{1}{2}}_{i,j-\frac{1}{2}}}\right]$$

where

$$M^{k-\frac{1}{2}}_{i+\frac{1}{2},j} \geq 0,\quad \lambda_{11}=0,\quad \lambda_{21}=1,\quad \lambda_{31}=-1$$
$$< 0,\quad \lambda_{11}=1,\quad \lambda_{21}=-1,\quad \lambda_{31}=0$$

$$N^{k-\frac{1}{2}}_{i+\frac{1}{2},j} \geq 0,\quad \nu_{11}=0,\quad \nu_{21}=1,\quad \nu_{31}=-1$$
$$< 0,\quad \nu_{11}=1,\quad \nu_{21}=-1,\quad \nu_{31}=0$$

$$M^{k-\frac{1}{2}}_{i,j+\frac{1}{2}} \geq 0,\quad \lambda_{12}=0,\quad \lambda_{22}=1,\quad \lambda_{32}=-1$$
$$< 0,\quad \lambda_{12}=1,\quad \lambda_{22}=-1,\quad \lambda_{32}=0$$

$$N^{k-\frac{1}{2}}_{i,j+\frac{1}{2}} \geq 0,\quad \nu_{12}=0,\quad \nu_{22}=1,\quad \nu_{32}=-1$$
$$< 0,\quad \nu_{12}=1,\quad \nu_{22}=-1,\quad \nu_{32}=0$$

Next is the friction term. The friction term becomes a source of instability if it is decretized with an explicit scheme. The present authors use an implicit scheme; i.e.,

$$\frac{gn^2}{D^{7/3}}M\sqrt{M^2+N^2}=\frac{gn^2}{\left(D^{k-\frac{1}{2}}_{i+\frac{1}{2}j}\right)^{7/3}}\frac{1}{2}\left(M^{k+\frac{1}{2}}_{i+\frac{1}{2}j}+M^{k-\frac{1}{2}}_{i+\frac{1}{2}j}\right)\sqrt{\left(M^{k-\frac{1}{2}}_{i+\frac{1}{2}j}\right)^2+\left(N^{k-\frac{1}{2}}_{i+\frac{1}{2}j}\right)^2}$$

$$\frac{gn^2}{D^{7/3}}N\sqrt{M^2+N^2}=\frac{gn^2}{\left(D^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}\right)^{7/3}}\frac{1}{2}\left(N^{k+\frac{1}{2}}_{i,j+\frac{1}{2}}+N^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}\right)\sqrt{\left(M^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}\right)^2+\left(N^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}\right)^2}$$

As a summary of the present section, the set of differential equations is given by

$$\eta^{k+1}_{i,j}=\eta^{k}_{i,j}-\frac{\Delta t}{\Delta x}\left[M^{k+\frac{1}{2}}_{i+\frac{1}{2}j}-M^{k+\frac{1}{2}}_{i-\frac{1}{2}j}\right]-\frac{\Delta t}{\Delta y}\left[N^{k+\frac{1}{2}}_{i,j+\frac{1}{2}}-N^{k+\frac{1}{2}}_{i,j-\frac{1}{2}}\right]$$

$$\frac{1}{1+{}^{x}\mu^{k-\frac{1}{2}}_{i+\frac{1}{2}j}}\left[\left(1-{}^{x}\mu^{k-\frac{1}{2}}_{i+\frac{1}{2}j}\right)M^{k-\frac{1}{2}}_{i+\frac{1}{2}j}-\frac{\Delta t}{\Delta x}\left\{\lambda_{11}\frac{\left(M^{k-\frac{1}{2}}_{i+\frac{3}{2}j}\right)^2}{D^{k-\frac{1}{2}}_{i+\frac{3}{2}j}}+\lambda_{21}\frac{\left(M^{k-\frac{1}{2}}_{i+\frac{1}{2}j}\right)^2}{D^{k-\frac{1}{2}}_{i+\frac{1}{2}j}}+\lambda_{31}\frac{\left(M\right.}{D}\right.\right.$$

$$-\left\{\nu_{11}\frac{M^{k-\frac{1}{2}}_{i+\frac{1}{2}j+1}N^{k-\frac{1}{2}}_{i+\frac{1}{2}j+1}}{D^{k-\frac{1}{2}}_{i+\frac{1}{2}j+1}}+\nu_{21}\frac{M^{k-\frac{1}{2}}_{i+\frac{1}{2}j}N^{k-\frac{1}{2}}_{i+\frac{1}{2}j}}{D^{k-\frac{1}{2}}_{i+\frac{1}{2}j}}+\nu_{31}\frac{M^{k-\frac{1}{2}}_{i+\frac{1}{2}j-1}N^{k-\frac{1}{2}}_{i+\frac{1}{2}j-1}}{D^{k-\frac{1}{2}}_{i+\frac{1}{2}j-1}}\right\}-gD^{k}_{i+\frac{1}{2}j}\frac{\Delta t}{\Delta x}\left\{\eta^{k}_{i+1}\right.$$

$$\frac{1}{{}^{y}\mu^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}}\left[\left(1-{}^{y}\mu^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}\right)N^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}-\frac{\Delta t}{\Delta x}\left\{\lambda_{12}\frac{M^{k-\frac{1}{2}}_{i+1,j+\frac{1}{2}}N^{k-\frac{1}{2}}_{i+1,j+\frac{1}{2}}}{D^{k-\frac{1}{2}}_{i+1,j+\frac{1}{2}}}+\lambda_{22}\frac{M^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}N^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}}{D^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}}+\lambda_{32}\frac{M^{k}_{i}}{}\right.\right.$$

$$-\frac{\Delta t}{\Delta y}\left\{\nu_{12}\frac{\left(N^{k-\frac{1}{2}}_{i,j+\frac{3}{2}}\right)^2}{D^{k-\frac{1}{2}}_{i,j+\frac{3}{2}}}+\nu_{22}\frac{\left(N^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}\right)^2}{D^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}}+\nu_{32}\frac{\left(N^{k-\frac{1}{2}}_{i,j-\frac{1}{2}}\right)^2}{D^{k-\frac{1}{2}}_{i,j-\frac{1}{2}}}\right\}-gD^{k}_{i,j+\frac{1}{2}}\frac{\Delta t}{\Delta y}\left\{\eta^{k}_{i,j+1}-\eta^{k}_{i,j}\right\}\Bigg]$$

where

$$\mu\, x_1{}^{k-\frac{1}{2}}_{i+\frac{1}{2}j}=\frac{1}{2}\frac{gn^2}{\left(D^{k-\frac{1}{2}}_{i+\frac{1}{2}j}\right)^2}\sqrt{\left(M^{k-\frac{1}{2}}_{i+\frac{1}{2}j}\right)^2+\left(N^{k-\frac{1}{2}}_{i+\frac{1}{2}j}\right)^2}$$

$$\mu y^{k-\frac{1}{2}}_{i,j+\frac{1}{2}} = \frac{1}{2}\frac{gn^2}{\left(D^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}\right)^2}\sqrt{\left(M^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}\right)^2+\left(N^{k-\frac{1}{2}}_{i,j+\frac{1}{2}}\right)^2}$$

$$D^{k}_{i+\frac{1}{2},j} = \frac{1}{2}\left(D^{k}_{i+1,j}+D^{k}_{i,j}\right)=\frac{1}{2}\left(\eta^{k}_{i+1,j}+\eta^{k}_{i,j}\right)+h_{i+\frac{1}{2},j}$$

$$\begin{aligned}D^{k-\frac{1}{2}}_{i+\frac{1}{2},j} &= \frac{1}{4}\left(D^{k}_{i+1,j}+D^{k-1}_{i+1,j}+D^{k}_{i,j}+D^{k-1}_{i,j}\right)\\ &= \frac{1}{4}\left(\eta^{k}_{i+1,j}+\eta^{k-1}_{i+1,j}+\eta^{k}_{i,j}+\eta^{k-1}_{i,j}\right)+h_{i+\frac{1}{2},j}\end{aligned}$$

$$D^{k}_{i,j+\frac{1}{2}} = \frac{1}{2}\left(D^{k}_{i,j+1}+D^{k}_{i,j}\right)=\frac{1}{2}\left(\eta^{k}_{i,j+1}+\eta^{k}_{i,j}\right)+h_{i,j+\frac{1}{2}}$$

$$\begin{aligned}D^{k-\frac{1}{2}}_{i,j+\frac{1}{2}} &= \frac{1}{4}\left(D^{k}_{i,j+1}+D^{k-1}_{i,j+1}+D^{k}_{i,j}+D^{k-1}_{i,j}\right)\\ &= \frac{1}{4}\left(\eta^{k}_{i,j+1}+\eta^{k-1}_{i,j+1}+\eta^{k}_{i,j}+\eta^{k-1}_{i,j}\right)+h_{i,j+\frac{1}{2}}\end{aligned}$$

with λ and ν given by Eq.(23).

1.2 INITIAL CONDITIONS AND BOUNDARY CONDITIONS

(1) Initial Conditions

The present programme is only for tsunamis. No wind waves and tides are included. The still water level is given by tides and is assumed constant during the computation of tsunamis. This means that no motion is assumed up to the time k-1/2. Therefore, we set the initial conditions in sea as,

$$\eta^{k-1}_{i,j}\ ,\quad M^{k-\frac{1}{2}}_{i+\frac{1}{2},j}\ ,\quad N^{k-\frac{1}{2}}_{i,j+\frac{1}{2}} = 0$$

For Run up computation on land, the initial water level η(I, j, k-1) is equal to ground height $h_{i,j}$.

$$\eta^{k-1}_{i,j} = -h_{i,j}$$

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

(2) Conditions for a Simple Harmonic Waves Train at an Offshore Open Boundary

In the following, a method of input at an offshore boundary is given for a progressive sine wave train.

Actual motion of the water at the offshore boundary is not given by the sine wave but by a resultant of the advancing and receding sine wave trains. If the sine motion is given at the boundary, no reflected wave can pass through the boundary, and a forced oscillation is inevitably introduced. At the open boundary, it is necessary to allow the reflected wave freely pass the boundary and go out of the region of computation. This is easily solved if the method of characteristics is used at the boundary.

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

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

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

Equation (36) is reduced to

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

and Eq.(37) is reduced to

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

Addition and subtraction of the two equations yield

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

This leads to

$$u\pm\sqrt{\frac{g}{h}}\eta=Const, \qquad on \qquad \frac{dx}{dt}=\pm\sqrt{gh}$$

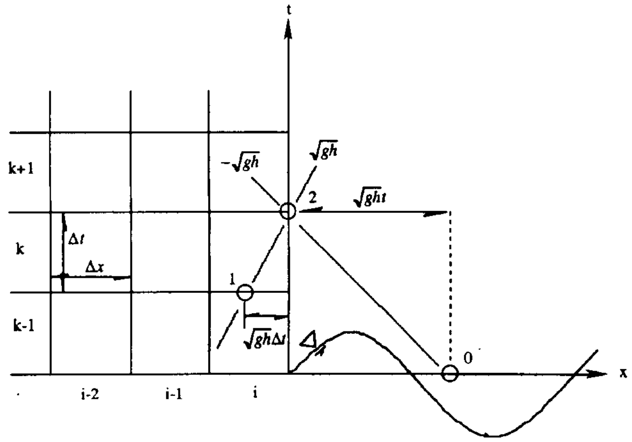


Figure 4. An offshore open boundary and characteristics when a simple harmonic wave train is propagating in the negative x direction.

Assume a simple harmonic wave train advancing in the negative x-direction having its front at x=x$_0$ at t=0. On referring Figure-4, equation (39) gives

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

along a positive characteristics, and

$$u_{2_2}-\sqrt{\frac{g}{h}}\eta_2=u_0-\sqrt{\frac{g}{h}}\eta_0$$

along a negative characteristics. The given wave train is propagating in the negative x-direction. It satisfies the following relationship between the horizontal water particle velocity u_0 and the water surface elevation η_0

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

With this relationship substituted, Eq.(41) is reduced to

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

and from Eqs.(40) and (41'), we have

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

If we assume the incident wave train η_0 is given by

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

where k_0 is the wave number, then the corresponding water particle velocity is expressed by

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

Consequently, the value of u_2 at the boundary is composed of two parts as follows,

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

where the first term on the right-hand side corresponds to the given incident wave train and the second term corresponds to the receding wave trains.

If Eq.(45) is rewritten in terms of water discharge, we have

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

where M_1 and η_1 are calculated by

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

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

Second, we consider a two-dimensional case in which the relationship of characteristics is, in principle, given on characteristic surface. In the present text, however, an extension of the one-dimensional case mentioned above is used. 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 he 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.

(3) Open Boundary Conditions for Forced Inputs

When the boundary condition itself is already composed of progressive and reflected waves, it is given with no modification at 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 a linear problem, either the displacement of water surface or discharge flux is used as the input at the boundary.

(4) Open Boundary Conditions for Free Transmission

A method is given to make waves in the computation region go outward freely passing through an open boundary. The characteristics relationship in (2) above is used. On referring Figure 3-4, at the boundary $x=x_0$, the relationships are

$$u_0=\eta_0=0 \ \ and \ \ u_1=\sqrt{g/h}\,\eta_1$$

This yields

$$u_2=u_1 \qquad or \qquad M^{k+\frac{1}{2}}_{i+\frac{1}{2},j}=M_1$$

In case of two-dimensional propagation, the boundary condition should be set in the same way as discussed above in (2).

(5) Boundary Conditions at Run up Fronts

Run up is taken into consideration only in nonlinear computations but not in linear computations.

Whether a computation cell is dry or submerged is judged in terms of the total water depth, as follows.

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

A wave front is located between the dry and submerged cells. The discharge across the boundary between the two cells is computed 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) 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 a structures is given by

$$Q=\mu h_1\sqrt{2gh_1} \ , \qquad if \ \ h_2\leq\frac{2}{3}h_1$$

$$=\mu' h_1\sqrt{2g(h_1-h_2)} \ , \ if \ \ h_2>\frac{2}{3}h_1$$

where h_1 and h_2 are the water depths in front of and behind structure measured above the top of structure, and coefficients $\mu=0.35$ and $\mu'=2.6\mu$

1.3 CONTINUATION OF REGIONS

(1) Necessity of Continuation of Regions in Numerical Computation

In the design of numerical computations for long waves such as storm surges and tsunamis, it is recommended to set an open sea boundary in the deep ocean where the boundary conditions can be accurately and easily given. In addition, in order to save the CPU time, also recommended are the proper use of linear and nonlinear theories according to the degree of non-linearity of the phenomena, and (2) coarse grids in the deep sea and fine grids in the near shore zone. These selection and use require the continuation of computation at the boundary of regions of different grid length.

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

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

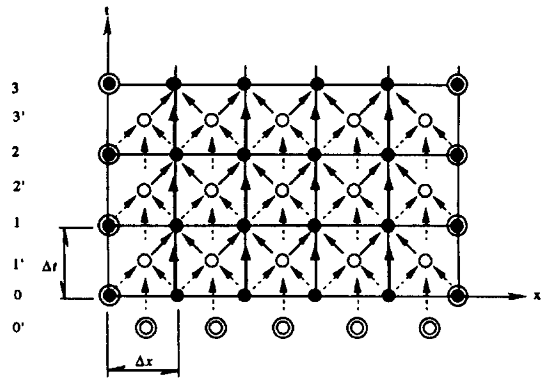
where Δt and Δx are the temporal and spatial grid lengths, and h_{max} is the maximum still water depth in a computation region. Approaching the shore, h_{max} becomes smaller, then a smaller Δx is selected to satisfy the CFL condition on keeping Δt constant. This is an ordinary way of how to select 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 different computation regions.

Methods are given, in the following, to connect water level and discharge flux between regions of different temporal and spatial grid lengths, in order to carry out a computation continuously.

(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 Δt is constant in any regions of different spatial grid length.

In section (a), a method of continuation and its importance is explained by taking a one-dimensional cases as an example. In section (b), a method of continuation is given for the spatially two-dimensional problem.



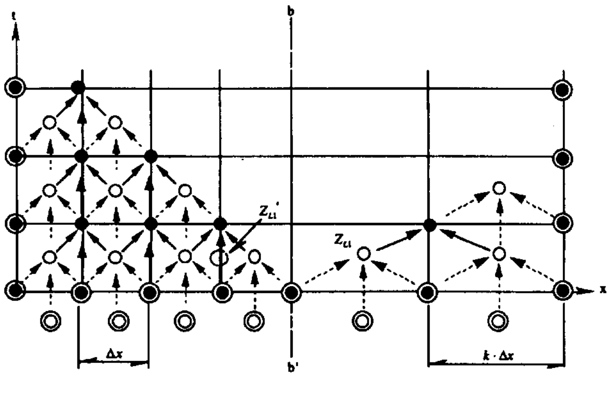


Figure 1-5. Computation procedure in the x-t plane, (A) if the spatial grid length is constant in the whole region, and (B) if two regions of different grid length are connected at the boundary b-b'.

(a) Continuation of Regions in the x-t plane

Figure 1-5 shows the process of computation in the x-t plane in a one-dimensional case. Arrows given by solid lines are for the computation of the discharge, and those by broken lines are for the computation of the water level. Numerals without prime on the t-axis correspond to the time step when the discharge is computed. Numerals with prime is for the computation of the water level. The discharge M^k is defined as the discharge at the positive x side of the computation cell(i,k).

Figure 1-5(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, and either the discharge M at t=0 as the initial condition, and either the discharge or water level along $x = 0$ and $x = n\Delta x$ 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.

Figure 1-5(B) is a case of continuation of regions of different grid length. In the region S (of small grid length) to the left of the line b-b', the spatial grid length is Δx, and in the right region L (of larger grid length) the spatial grid length is larger and is equal to $k\Delta x$ ($k>1$). The computation procedure in each region is the same as in the former case. However, if values are not calculated and not connected along the line, the region where the solution is given becomes narrower with the lapse of time as shown in Figure-5(B).

In order to obtain the solution in the whole x-t plane, the discharge should be known on the boundary b-b' 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} in the region L with respect to the line b-b'. An interpolation may be used to determine $Z_{1'}$ from values obtained in the region S. However, in place of interpolation, the present authors recommend to set k be an odd number then the necessary value is already calculated in the computation in the region S.

In the authors' programme, they take the following assumptions are made.

(i) The ratio k is 3;
(ii) For connection of water level, the region L needs an extra cell in the region S beyond the line b-b';
(iii) In more general case than Figure-5(b), the region S needs an extra cell in the region L beyond the line b-b'.

Otherwise, when the direction of the x-axis is taken inversely to the case in Figure-5(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 cell.

In conclusion, the computation procedure for the continuation of region is summarized as shown in Figure-6 for a one-dimensional case.

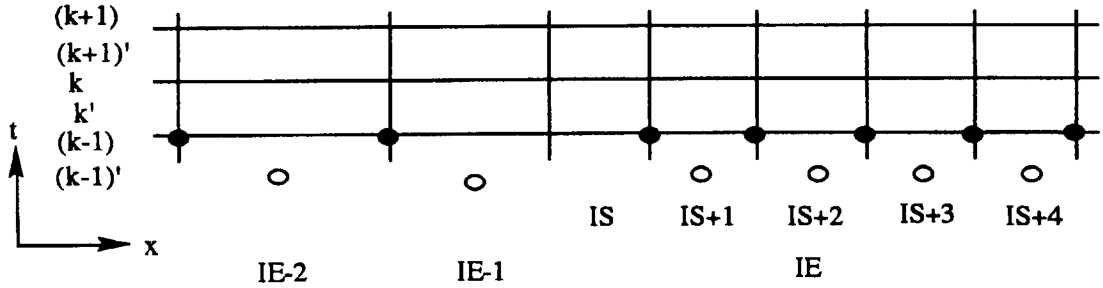


Figure 1-6(a). Details of computation continuation in the x-t plane in case of k = 3 and Δt =const. Open (filled) circles are points where the water level (discharge) is computed.

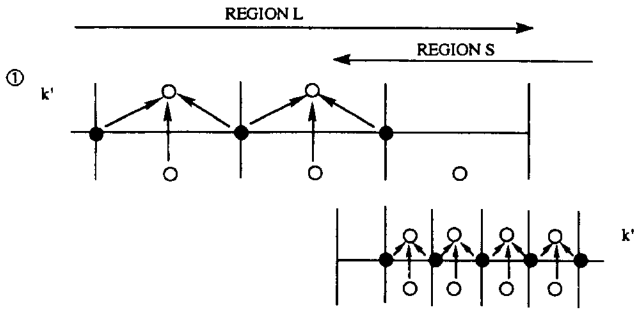


Figure 1-6(b). With the equation of continuity, $Z_S(I,K')$ and $Z_L(I',K')$ are computed. Suffixes $_S$ and $_L$ denote values in the region S of Δx and in the region L of $3\Delta x$

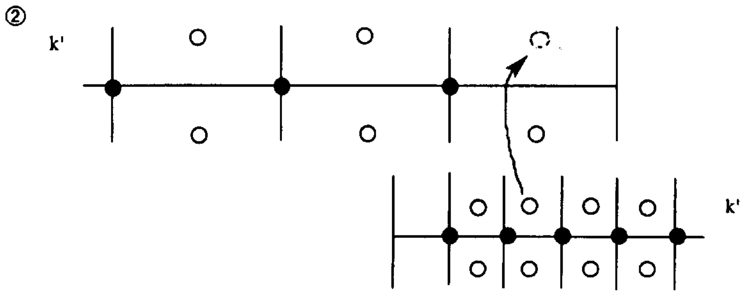


Figure 1-6(c). The water level $Z_L(IE,K')$ is set equal to $Z_S(IS+'',K')$

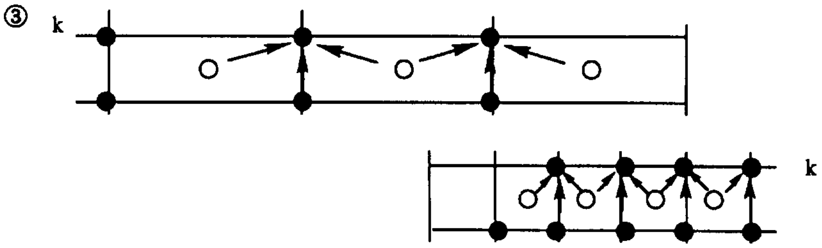


Figure 1-6(d). With the equation of motion, $M_S(I,K)$ and $M_L(I',K)$ are computed.

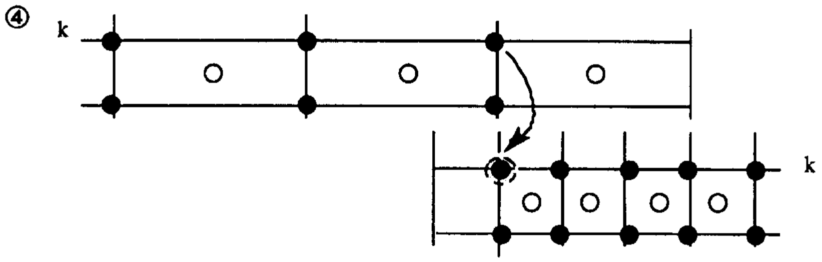


Figure 1-6(e). The water discharge $M_S(IS,K)$ is set to $M_L(IE-1,K)$

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

Discussion in the preceding section give the way of connection 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 water 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 Figure-7 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 cell is composed of 9 small cells of the region S. The central point (marked by double circles in Figure-7) of the 9 cells is located at the central point of the extra cell.

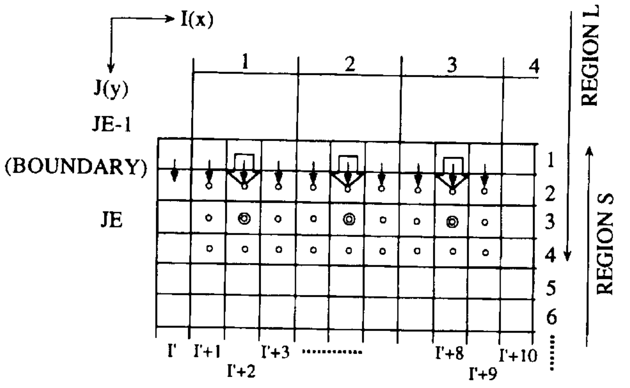


Figure 1-7. Continuation between two regions S and L in the (x,y-t) space, where k = 3 and Δt = const.

Therefore, we set

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

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

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

Different from the water level, the connection of the water discharge requires either interpolation or extrapolation. Along the boundary between the regions L and S, discharges area connected as follows, on referring Figure-7.

(i) At points from (I'+2) to (I'+8) along the boundary, discharges are calculated by an interpolation. For example,

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

(ii) At other points, an interpolation is used if the region L still exists. For example, the discharge at the point (I'+10) on the boundary is given by,

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

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

$$M_S(I',\ 1) = (5.0*M_L(1,JE-1) - 2.0*M_L(2,JE-1))/3.0$$

(3) Continuation of Regions of Different Δt

Assume that the time grid changes from 3Δt to Δt. Figure 1-8 shows the computation procedure in the x-t plane.

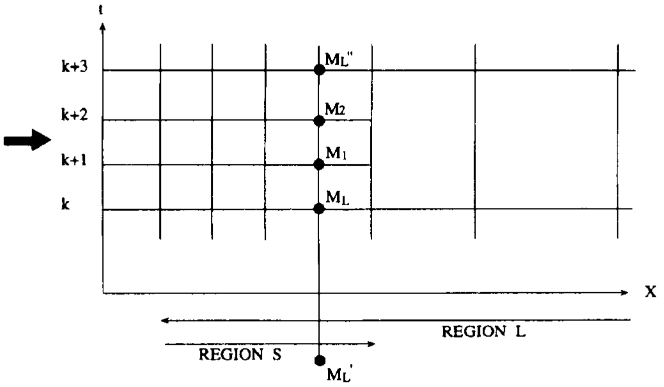


Figure 1-8. Computation continuation between two regions S and L in the x-t plane when Δt varies at the ration of 1:3.

As for the water level, values in the region S is used in the computation in the region L. This means that values are connected at the time step shown by an arrow in Figure 1-8; i.e., at the middle of 3 cells in the region S.

As for the discharge, values in the region L is 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.

The same extrapolation is applicable to M_2. However, an interpolation is preferable for M_2 in order to reduce numerical errors. For this interpolation, value of M_L'' should be known beforehand at time (K+2).

Figure 1-9 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.

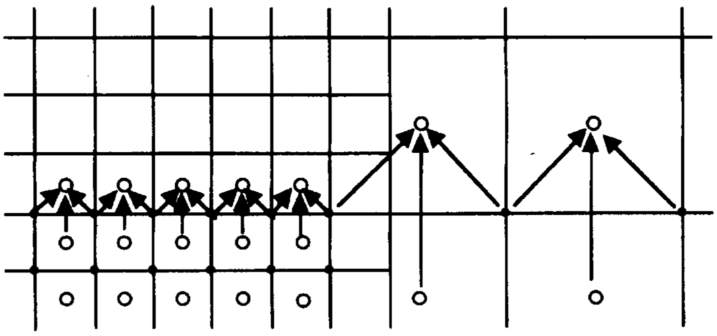


Figure 1-9(a). Details of computation continuation in the x-t plane, when Δt varies at the ratio of 1:3. Computation of the water level Z in the regions S and L, with the equation of continuity.

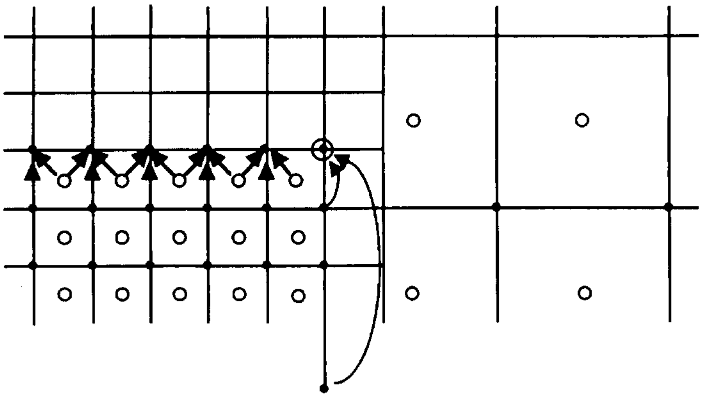


Figure 1-9(b). Computation of the water discharge M in the region S, with the equation of motion. An extrapolation in the region L to determine value of M at the boundary. Connection of the water discharge from the region L to the region S.

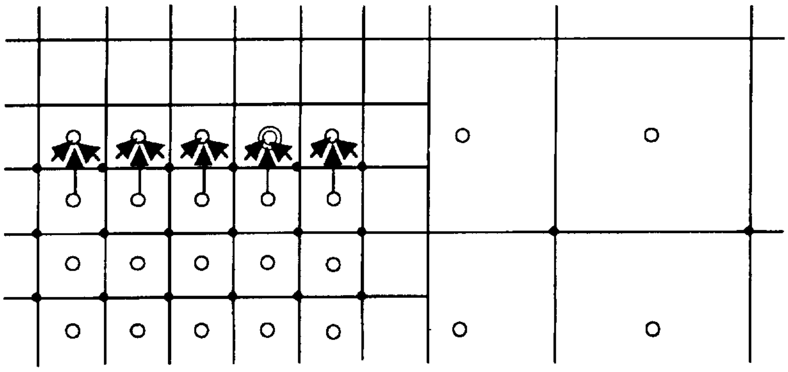


Figure 1-9(c). Computation of the water level Z in the region S, with the equation of continuity. Connection of the water level from the region S to the region L.

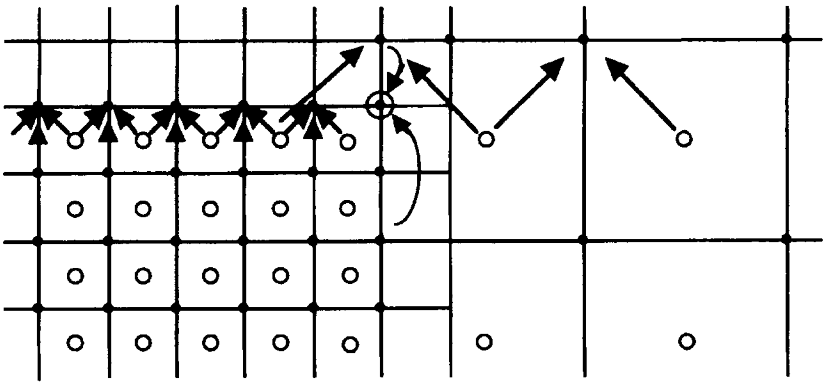


Figure 1-9(d). Computation of the water discharge M in the region S and L, with the equation of motion. An interpolation in the region L to determine values of M at the boundary. Connection of the water discharge from the region L to the region S.

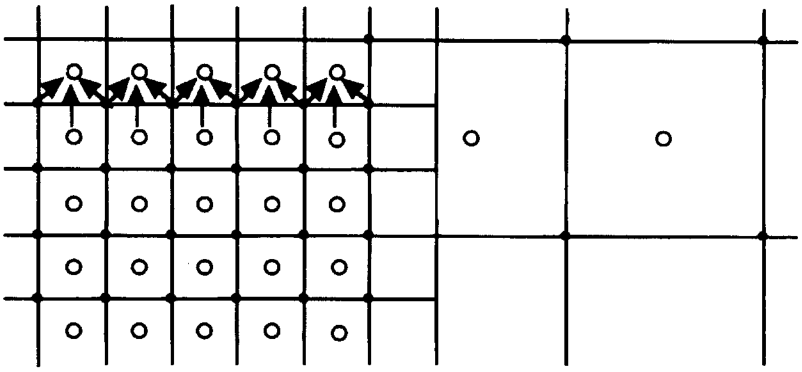


Figure 1-9(e). Computation of the water level Z in the region S, with the equation of continuity

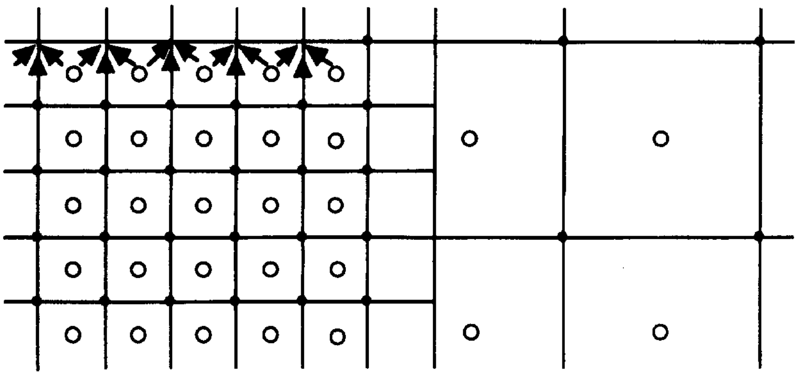


Figure 1-9(f). Computation of the water discharge M in the region S, with the equation of motion.

CHAPTER 2. TYPICAL PROGRAMMES

2.1 ASSUMPTIONS

The following three assumptions are made.

(1) The astronomical tide does not vary with respect to time throughout the tsunami computation. The still water level in the computation is set equal to the water level at the beginning of computation.

(2) Both the spatial and temporal grid lengths vary only at the ratio of 1:3:9: and so on, if the change of them is necessary.

(3) In the linear computation, no run up can be included, and therefore the computation is not carried out for the water depth shallower than 0.1cm, and vertical walls are set in place of the actual slope.

2.2 CONSTRUCTION OF MAIN PROGRAMMES

In order to make it easy to understand the flow of computation, the main programme is composed of calls of sub-programmes. For simplicity of explanation, we take a region of computation shown in Figure 2-1 as an example. The region is of a constant water depth, surrounded by vertical walls (shown by hatches) and has a small bay also bounded by vertical walls. The region is divided into three subregions, A, B and C, the spatial and temporal grid lengths of which vary as 9:3:1. In Table 2-1, letters used in the main and subroutines are summarized and explained. The main programme is given after Table 2-1 from page 30 to page 34. In the following, explanations and hints in programming are given, in the order of the flow in the main programme.

(1) Specification Statement

- REAL M and N should be stated.
- Declare, in each domains, three-dimensional arrays for Z, M, N, DZ, DM, and DN, and two-dimensional arrays for HZ, HM, HN, IR and IB. Declaration of BT(10) should not be missed (This will be explained later). Dimensions for space should be increased by one in order to include an extra row or column outside the domain under consideration, as shown in Figure 2-1. Otherwise, discharge on the boundary, or IB and IR maps (both of them will be explained later) are sometimes not definable, according to the way of selection of the I and J axes. Dimensions for time are always taken to be 2, because values are changed by a subroutine CHANGE.

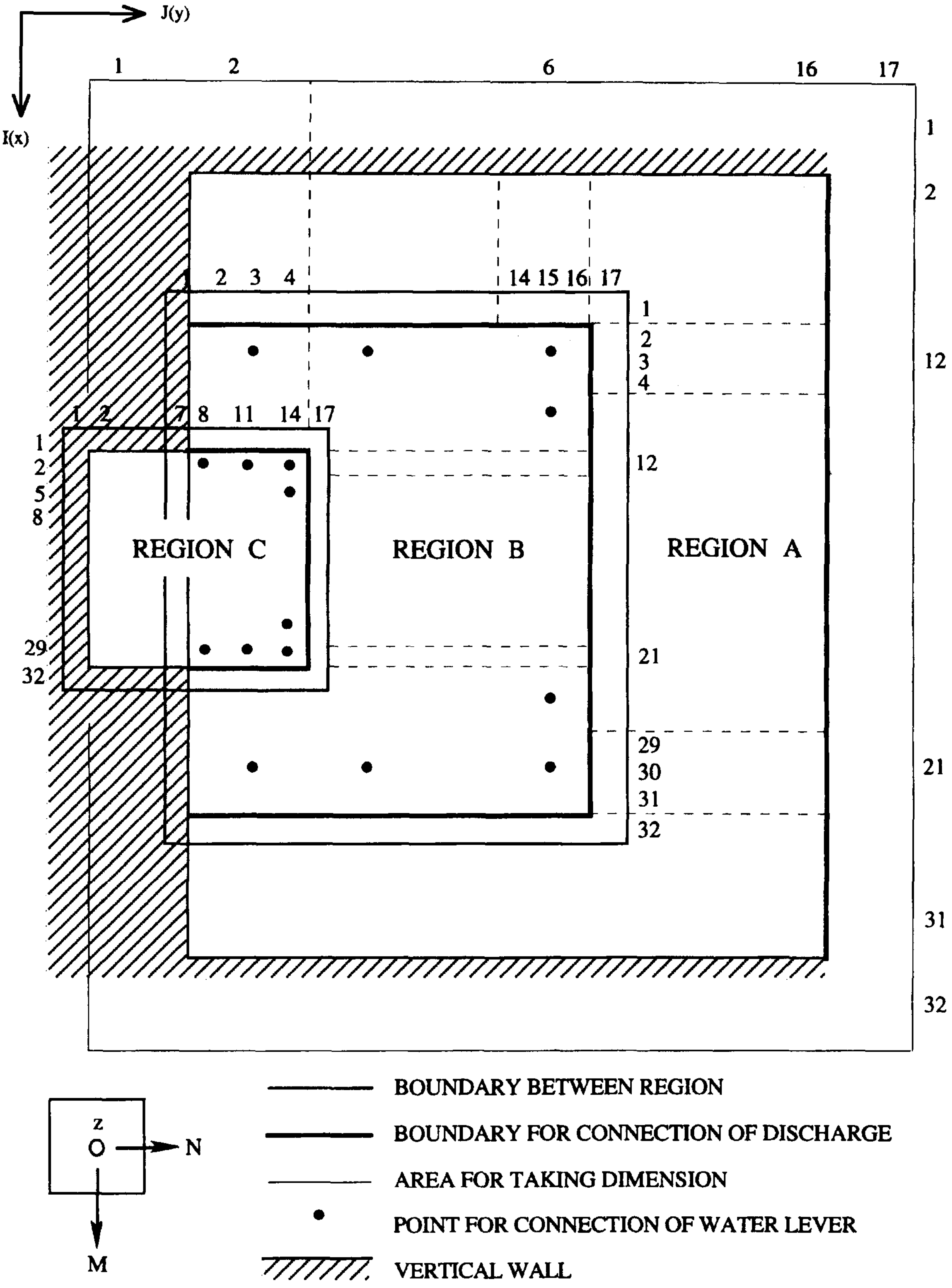


Figure 2-1: A model basin for explanation of the programme

(2) Input of Setting Values

- Values of DX, DT AND R(=DT/DX) should be input in every domains.

(3) Setting of Initial Value

(i) CALL DEPTH0

With this command, data of water depth HZ, IB and IR map are input in every domain. To build this subroutine for data input, the following points should be taken into consideration.

- Water depth

Read water depths HZ on hydrographic charts with the z-axis positive downwards.

- IB map (a two-dimensional array)

An IB map gives the method of computation and the existence of vertical walls. According to the following rule, positive integers of one or two figures are allotted to and input into every grid point in every domain.

The unit digit = 0, the computation is with the linear theory without the convection term.

The unit digit = 1, the computation is with the nonlinear theory with the convection term included.

The tenth digit = 1, the discharge flux M in the I-direction is zero, owing to the existence of a vertical wall.

The tenth digit = 2, the discharge flux N in the J-direction is zero, owing to the existence of a vertical wall.

The tenth digit = 3, the discharge fluxes M and N in the I- and J-directions are zero, owing to the existence of a vertical wall.

The tenth digit = 4, no computation.

For example (See Figure 2-1), if the computation in the domain A is carried out with the linear theory, the IB(I,J) is set as follows.

IB(I,J) = 10, for $1 \leq I \leq 31$ and $1 \leq J \leq 16$, because of the vertical wall.

IB(I,J) = 20, for $2 \leq I \leq 31$ and J = 1, because of the vertical wall.

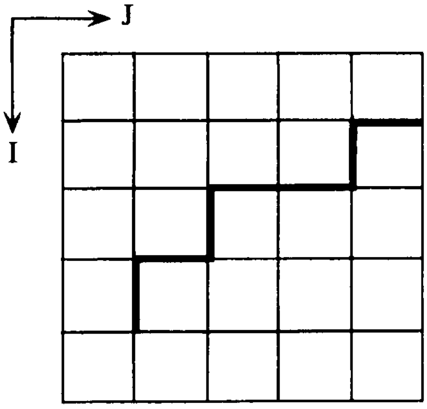
IB(I,J) = 40, for $12 \leq I \leq 20$ and $1 \leq J \leq 5$, the computation in the domain A is not necessary because it is carried out in the domain B.

IB(I,J) = 0, for other combination of I and J.

- IR map (a two-dimensional array)

This map shows the existence of such structures as sea walls of finite crown height. According to the following rule, positive integers of one or two figures are allotted to and input into every grid point in every domain.

The unit digit -------- assign the address I (=1~9) of BT(I), data of the crown height of sea walls.
No tenth digit -------- no sea wall.
The tenth digit = 1, there is a sea wall on the computation line of discharge M in the I-direction.
The tenth digit = 2, there is a sea wall on the computation line of discharge N in the J-direction.
The tenth digit = 3, there is a sea wall both on the computation lines of discharge M and N in the I- and J-directions.



0	0	0	0	11
0	0	11	31	0
0	31	0	0	0
21	0	0	0	0
0	0	0	0	0

Figure 2-2: An example of the IR map.

An example of IR map is shown in Figure 2.2 The thick line in Figure 2.2(a) is the axis of the sea wall. Figure.2.2(b) is the corresponding IR map. In BT(1), stored is the crown height of sea wall measured positive upward above still water level. Attention should be paid on the fact that the positive direction for the crown height is different from that for HZ, HM and HN.

(ii) CALL DEPTH

Water depths HM and HN at the point where the discharge is computed are calculated. A call statement is necessary for a computation domain.

(iii) CALL SETZRO

Initial values are set for Z, M, N and DZ, all of which are set equal to zero. A call statement is necessary for a computation domain.

(4) Computation Repeated with Respect to Time

(i) CALL CONTIN

The water depth Z at the next time step is computed with the equation of continuity. A call statement is necessary for a computation domain.

(ii) CALL JOINTZ

The water depth is connected between domains of different time and space grid lengths. A call statement is necessary for a line of connection. In Figure 2-1, six call statements are required in total :three call statements between the domain A and the domain B, and other three call statements between the domain B and the domain C.

(iii) CALL MOTION

The discharges M and N at the next time step is computed with the equation of motion. Discharges over sea walls is evaluated with the Hom-ma formula. A call statement is necessary for a commutation domain.

(iv) CALL BOUND

The conditions are input at the seaside boundary. Input data of a tsunami should be prepared, on referring Chapter 1, Section 1-1.

(v) CALL JOINTQ

The discharge is connected between domains of different time and space grid lengths. A call statement is necessary for a line of connection.

(vi) CALL OUTPUT

A subroutine is added at need, to output the computed results.

(vii) CALL CHANGE

Old data one time step before are changed with new data. For example, newly obtained Z(I, J, 2) replaces old data and become Z(I, J, 1) which is used in the next computation.

- Time Step Index

If the time step Δt varies from a domain to another, the computation procedures from (i) to (vii) mentioned above is not always carried out at every time step except in the domain of the smallest Δt. The time step at which the computation is carried out in the other domains should be controlled by introducing the "time step index". On referring Figure 1-9 and the subroutine given later in 2.3, the time step index is as follows in case of Figure 2-1.

KT in CONTIN, MOTION, CHANGE	1, 3, 9 in domains C, B and A, respectively.
KT in JOINTZ	3 in connection of domains C and B. 9 in connection of domains B and A.
KT1 and KT2 in JOINTQ	1 and 3 in connection of domains C and B. 3 and 9 in connection of domains B and A.

Table 2-1: Notations in the Main Programme

Notation	Type	Definition
Z	R (3-D array)	water level
M	R (3-D array)	discharge in the I-direction
N	R (3-D array)	discharge in the J-direction
DZ	R (3-D array)	total water depth at point for Z
DM	R (3-D array)	total water depth at point for M
DN	R (3-D array)	total water depth at point for N
HZ	R (2-D array)	still water depth at point for Z
HM	R (2-D array)	still water depth at point for M
HN	R (2-D array)	still water depth at point for N
IB	I (2-D array)	map of the selection of theory (linear of nonlinear, and the existence of vertical walls.
IR	I (2-D array)	map of the existence of structures
BT	R (1-D array)	crown height of structures
DX	R	space grid length
DT	R	time step length
R	R	ration DT / DX
GG	R	gravitational acceleration
PP	R	2 Pi
K	I	time step of the computation
K1	I	argument to call subroutines, taken as the same value as K
KE	I	the last time step
(KOUT)	I	time step for output procedure
(WP)	R	wave period
(WD)	R	water depth
(LL)	R	index for output
(LX)	R	index for output

Note: Suffixes A, B and C denotes domains. Notations in parentheses are used only in this programme

```
C
C        MAIN PROGRAMME
C
         REAL MA,NA,MB,NB,MC,NC
         DIMENSION ZA(32,17,2),MA(32,17,2),NA(32,17,2),DZA(32,17,2), &
         DMA(32,17,2),DNA(32,17,2),HZA(32,17),HMA(32,17),HNA(32,17) DIMENSION
         ZB(32,17,2),MB(32,17,2),NB(32,17,2),DZB(32,17,2), &
         DMB(32,17,2),DNB(32,17,2),HZB(32,17),HMB(32,17),HNB(32,17) DIMENSION
         ZC(32,17,2),MC(32,17,2),NC(32,17,2),DZC(32,17,2), &
         DMC(32,17,2),DNC(32,17,2),HZC(32,17),HMC(32,17),HNC(32,17) DIMENSION
         IRA(32,17),IRB(32,17),IRC(32,17) DIMENSION IBA(32,17),IBB(32,17),IBC(32,17) DIMENSION
         BT(10)
C
C        READ(5,*), DXC,DTC,WP,WD,KOUT
         DXC = 50.0
         DTC = 1.0
         WP = 600.0
         WD = 50.0
         KOUT = 5
         DXB=3.0*DXC
         DXA=3.0*DXB
         DTB=3.0*DTC
         OTA=3.0*DTB
         RA=DTA/DXA
         RB=DTB/DXB
         RC=DTC/DXC
         GG=9.8
         PP=6.283185
         KE=IFIX(2.0*WP/DTC)
         DO 20 I=1,10
         BT(I)=0.0
    20   CONTINUE
C
         CALL DEPTHO(32,17,HZA,IRA,IBA,WD,1)
         CALL DEPTHO(32,17,HZB,IRB,IBB,WD,2)
         CALL DEPTHO(32,17,HZC,IRC,IBC,WD,3)
C
         CALL DEPTH(32,17,HZA,HMA,HNA,BT,IRA)
         CALL DEPTH(32,17,HZB,HMB,HNB,BT,IRB)
         CALL DEPTH(32,17,HZC,HMC,HNC,BT,IRC)

         CALL SETZRO(32,17,ZA,MA,NA,DZA,HZA)
         CALL SETZRO(32,17,ZB,MB,NB,DZB,HZB)
         CALL SETZRO(32,17,ZC,MC,NC,DZC,HZC)
         LL=1
         LX=0
C
         DO 10 K=1,KE
         K1=K
         CALL CONTIN(32,17,2,31,2,16,ZA,MA,NA,DZA,HZA,RA,IBA,K1,9)
         CALL CONTIN(32,17,2,31,2,16,ZB,MB,NB,DZB,HZB,RB,IBB,K1,3)
         CALL CONTIN(32,17,2,31,2,16,ZC,MC,NC,DZC,HZC,RC,IBC,K1,1)
C
         CALL JOINTZ(32,17,32,17,ZB,ZA,DZB,DZA,12,12,2,6,3,3,K1,9,DXA,DXB)
         CALL JOINTZ(32,17,32,17,ZB,ZA,DZB,DZA,12,21,6,6,3,15,K1,9,DXA,DXB)
         CALL JOINTZ(32,17,32,17,ZB,ZA,DZB,DZA,21,21,2,6,30,3,K1,9,DXA,DXB)
C
         CALL JOINTZ(32,17,32,17,ZC,ZB,DZC,DZB,12,12,2,4,3,9,K1,3,DXB,DXC)
         CALL JOINTZ(32,17,32,17,ZC,ZB,DZC,DZB,12,21,4,4,3,15,K1,3,DXB,DXC)
         CALL JOINTZ(32,17,32,17,ZC,ZB,DZC,DZB,21,21,2,4,30,9,K1,3,DXB,DXC)
```

```
C
           CALL    MOTION(32,17,2,31,2,16,ZA,MA,NA,DZA,DMA,DNA,HZA,HMA,HNA,IRA,     &
           IBA,RA,0.0,DTA,K1,9)
           CALL    MOTION(32,17,2,31,2,16,ZB,MB,NB,DZB,DMB,DNB,HZB,HMB,HNB,IRB,     &
           IBB,RB,0.0,DTB,K1,3)
           CALL    MOTION(32,17,2,31,2,16,ZC,MC,NC,DZC,DMC,DNC,HZC,HMC,HNC,IRC,     &
           IBC,RC,0.0,DTC,K1,1)

C
           CALL BOUND(32,17,2,31,NA,ZA,WD,WP,DTA,DXA,K1,9)
C
           CALL JOINTQ(32,17,32,17,MA,MB,NA,NB,1,1,2,16,11,2,0,1,0,2, & K1,3,9,DXA,DXB)
           CALL JOINTQ(32,17,32,17,MA,MB,NA,NB,2,31,16,16,12,6,0,2,0,2, & K1,3,9,DXA,DXB)
           CALL JOINTQ(32,17,32,17,MA,MB,NA,NB,31,31,2,16,21,2,0,1,0,2, & K1,3,9,DXA,DXB)
           CALL JOINTQ(32,17,32,17,MB,MC,NB,NC,1,1,8,16,11,2,0,1,0,2, & K1,1,3,DXB,DXC)
           CALL JOINTQ(32,17,32,17,MB,MC,NB,NC,2,31,16,16,12,4,0,2,0,2, & K1,1,3,DXB,DXC)
           CALL JOINTQ(32,17,32,17,MB,MC,NB,NC,31,31,8,16,21,2,0,1,0,2, & K1,1,3,DXB,DXC)
C
           IF(MOD(K1,9).EQ.5) LL=LL+1
           IF(MOD(LL,KOUT).NE.0) GO TO 30
           IF(LL.EQ.LX) GO TO 30
C
           CALL OUTPUT(32,17,ZA,DZA,K1)
           CALL OUTPUT(32,17,ZB,DZB,K1)
           CALL OUTPUT(32,17,ZC,DZC,K1)
           WRITE(60,1000) K
      1000 FORMAT(10X,'K=',I5)
      30   CONTINUE
           LX=LL
C
           CALL CHANGE(32,17,ZA,MA,NA,DZA,K1,9)
           CALL CHANGE(32,17,ZB,MB,NB,DZB,K1,3)
           CALL CHANGE(32,17,ZC,MC,NC,DZC,K1,1)
C
      10   CONTINUE
C
           STOP
           END

           SUBROUTINE BOUND(IG,JG,IS,IE,N,Z,WD,WP,DT,DX,KK,KT)
           REAL N
           DIMENSION N(IG,JG,2),Z(IG,JG,2)
           GG=9.8
           PP=6.283185
           IF(KT.EQ.1) GO TO 200
           IF(MOD(KK,KT).NE.KT/2+1) RETURN
      200  CONTINUE
           DO 10 I=IS,IE
           CC=SQRT(GG*WD)
           XMM=(CC*DT*N(I,15,1)+(DX-CC*DT)*N(1,16,1))/DX
           ZZ2=(Z(I,15,2)+Z(I,15,1))*0.5
           ZZ1=(Z(I,16,2)+Z(I,16,1))*0.5
           ZZ=((CC*DT-0.5*DX)*ZZ2+(1.5*DX-CC*DT)*ZZ1)/DX
           EZ=-2.0*SIN(PP*FLOAT(KK)*DT/(WP*FLOAT(KT)))
           N(I,16,2)=CC*EZ+0.5*(XMM+CC*ZZ)
      10   CONTINUE
           RETURN
           END
```

```
      SUBROUTINE DEPTH (IG,JG,HZ,HM,HN,BT,IR)
      DIMENSION HZ(IG,JG),HM(IG,JG),HN(IG,JG)
      DIMENSION IR(IG,JG),BT(10)
      DO 10 I=1,IG
      DO 10 J=1,JG
      IF(I.EQ.IG) GO TO 11
      HM(I,J)=0.5*(HZ(I,J)+HZ(I+1,J))
      GO TO 12
11    HM(I,J)=HZ(I,J)
12    IF(J.EQ.JG) GO TO 13
      HN(I,J)=0.5*(HZ(I,J)+HZ(I,J+l))
      GO TO 10
13    HN(I,J)=HZ(I,J)
10    CONTINUE
      DO 14 I=1,IG
      DO 14 J=1,JG
      IRR=MOD(IR(I,J),10)
      IRM=IR(I,J)/10
      IF(IRM.EQ.0) GO TO 14
      IF(IRM.EQ.2) GO TO 15
      HM(I,J)=-BT(IRR)
      IF(IRM.EQ.1) GO TO 14
15    HN(I,J)=-BT(IRR)
14    CONTINUE
      RETURN
      END

      SUBROUTINE OUTPUT(IG,JG,Z,DZ,KK)
C
      DIMENSION LW(20),Z(IG,JG,2),DZ(lG,JG,2)
      WRITE(6,100) KK,(J,J=2,l6)
100   FORMAT(1H ,2X,'K=',I4,/,6X,15I4)
      IG1=IG-1
      JGl=JG-1
      DO 10 I=2,IG1
      DO 11 J=2,JG1
      LW(J)=0
      IF(DZ(I,J,2).GT.1.0E-4) LW(J)=IFIX(100.0*Z(I,J,2)+0.5)
11    CONTINUE
      WRITE(6,101) I,(LW(J).J=2,16)
101   FORMAT(1H ,I5,15I4)
10    CONTINUE
      RETURN
      END
```

2.3 Explanation of Subroutines

(1) DEPTH (Computation of the water depth at points for discharge)

Objective

The still water depth at points for discharge is calculated, based upon the still water depth at point for water depth. Information of the existence of structures from the map of breakwaters is also input.

Method of Connection

CALL DEPTH (IG, JG, HZ, HM, HN, BT, IR)

Parameter	Type	Content Before Calling the Subroutine	Content When Returned from the Subroutine
IG, JG	I	Indices of HZ,HM,HN and IR in the main programme	No change
HZ	R (2-D array) (JG, IG)	Still water depth	No change
HM	R (2-D array) (JG, IG)	No input	Still water depth at points for discharge in the I-direction
HN	R (2-D array) (JG, IG)	No input	Still water depth at points for discharge in the J-direction
BT	R (1-D array) (10)	Crown height of breakwaters	No change
IR	I (2-D array) (JG, IG)	Map of existence of breakwaters (integer of two figures)	No change

```
      SUBROUTINE DEPTH (IG,JG,HZ,HM,HN,BT,IR)
      DIMENSION HZ(IG,JG),HM(IG,JG),HN(IG,JG)
      DIMENSION IR(IG,JG),BT(10)
      DO 10 I=1,IG
      DO 10 J=1,JG
      IF(I.EQ.IG) GO TO 11
      HM(I,J)=0.5*(HZ(I,J)+HZ(I+1,J))
      GO TO 12
11    HM(I,J)=HZ(I,J)
12    IF(J.EQ.JG) GO TO 13
      HN(I,J)=0.5*(HZ(I,J)+HZ(I,J+l))
      GO TO 10
13    HN(I,J)=HZ(I,J)
10    CONTINUE
      DO 14 I=1,IG
      DO 14 J=1,JG
      IRR=MOD(IR(I,J),10)
      IRM=IR(I,J)/10
      IF(IRM.EQ.0) GO TO 14
      IF(IRM.EQ.2) GO TO 15
      HM(I,J)=-BT(IRR)
      IF(IRM.EQ.1) GO TO 14
15    HN(I,J)=-BT(IRR)
14    CONTINUE
      RETURN
      END
```

(2) SETZRO (Setting of initial condition)

Objective

Input of initial values of water level, water discharge, and total water depth at points for water level.

Method of Connection

CALL SETZRO (IG, JG, Z, M, N, DZ, HZ)

Parameter	Type	Content Before Calling the Subroutine	Content When Returned from the Subroutine
IG, JG	I	Indices of Z, M, N, DZ and HZ in the main programme	No change
Z	R (3-D array) (JG, IG, 2)	No input	Initial water level
M	R (3-D array) (JG ,IG, 2)	No input	Initial discharge in the I-direction
N	R (3-D array) (JG ,IG, 2)	No input	Initial discharge in the J-direction
DZ	R (3-D array)	No input	Initial total water depth at point for
HZ	R (2-D array) (JG, IG)	Still water depth at point for water level	No change

```
          SUBROUTINE SETZRO (IG,JG,Z,M,N,DZ,HZ)
          REAL M,N
          DIMENSION Z(IG,JG,2),M(IG,JG,2),N(IG,JG,2)
          DIMENSION DZ(IG,JG,2),HZ(IG,JG)
          DO 10 K=1,2
          DO 10 I=1,IG
          DO 10 J=1,JG
          M(I,J,K)=0.0
          N(I,J,K)=0.0
          DZ(I,J,K)=HZ(I,J)
          IF(HZ(I,J).LT.1.0E-5) DZ(I,J,K)=0.0
          Z(I,J,K)=DZ(I,J,K)-HZ(I,J)
    10    CONTINUE

          RETURN
          END
```

(3) CONTIN (Computation of the equation of continuity)

Objective

Cotation of the water level and total water depth at the next time step with the equation of continuity.

Method of Connection

CALL CONTIN (IG, JG, IS, IE, JS, JE, Z, M, N, DZ, HZ, R, IB, KK, KT)

Parameter	**Type**	**Content Before Calling the Subroutine**	**Content When Returned from the Subroutine**
IG, JG	I	Indices of Z, M, N, DZ and HZ in the main programme	No change
IS, IE	I	Co-ordinates of the start of imputation (IS, JS) and of end (IE, JE)	No change
JS, JE	I		No change
Z	R (3-D array) (JG, IG, 2)	Water level given as Z(I, J, 2)=Z(I, J, 1)	Z(I, J, 2) newly computed
M	R (3-D array) (JG, IG, 2)	Discharge in the I-direction	No change
N	R (3-D array) (JG, IG, 2)	Discharge in the J-direction	No change
DZ	R (3-D array) (JG, IG, 2)	Total water depth at points for water level	DZ(I, J, 2) newly computed
HZ	R (2-D array) (JG, IG)	Still water depth at points for water level	No change
R	R	DT/DX; ratio of time-to-space grid length	No change
IB	I (2-D array) (JG, IG)	Map of the selection of theory (linear or nonlinear) and of the existence of vertical walls	No change
KK	I	Time step	No change
KT	I	Time step index	No change

```
          SUBROUTINE CONTIN (IG,JG,IS,IE,JS,JE,Z,M,N,DZ,HZ,R,IB, & KK,KT)
C
          REAL M,N
          DIMENSION Z(IG,JG,2),M(IG,JG,2),N(IG,JG,2),HZ(IG,JG)
          DIMENSION DZ(IG,JG,2),IB(IG,JG)
          DATA GX/1.0E-5/
          IF(KT.EQ.1)GO TO 200
          IF(MOD(KK,KT).NE.1)RETURN
      200 CONTINUE
          DO 10 I=IS,IE
          DO 10 J=JS,JE
          IF(IB(I,J).EQ.40) GO TO 11
          XM=0.0
          XN=0.0
          IF(I.NE.1)XM=M(I-1,J,2)
          IF(J.NE.1)XN=N(I,J-1,2)
          ZZ=Z(I,J,1)-R*(M(I,J,2)-XM+N(I,J,2)-XN)
          IF(ABS(ZZ).LT.1.0E-10)ZZ=0.0
          DD=ZZ+HZ(I,J)
          IF(DD.LT.GX) GO TO 11
          DZ(I,J,2)=DD
          Z(I,J,2)=ZZ
          GO TO 10
      11  DD = 0.0
          DZ(I,J,2)=DD
          Z(1,J,2)=DD-HZ(I,J)
      10  CONTINUE
          RETURN
          END
```

(4) JOINTZ (Connection of the water level in space and time)

Objective

Connection of the water level between computation domains of different Δx and Δt.

Method of Connection

CALL JOINTZ (IG1, JG1, IG2, JG2, Z1, Z2, DZ1, DZ2, IS, IE, JS, JE, ISS, JSS, KK, KT, DX1, DX2)

Parameter	**Type**	**Content Before Calling the Subroutine**	**Content When Returned from the Subroutine**
IG1, JG1	I	Indices of Z1 and DZ1 in the main programme	No change
IG2, JG2	I	Indices of Z2 and DZ2 in the main programme	No change
Z1	R (3-D array) (JG1, IG1, 2)	Water level in the sender (domain of fine grids)	No change
Z2	R (3-D array) (JG2,IG2,2)	Water level in the receiver (domain of coarse grids)	Values after connection
DZ!	R (2-D array) (JG1, IG1)	Water depth at points for water level in the sender	No change
DZ2	R (2-D array) (JG2, IG2)	Water depth at points for water level in the receiver	No change
IS, IE	I	Co-ordinates of the start of connection	No change
JS, JE	I	(IS, JS) and the end (IE, JE) in the receiver	No change
ISS, JSS	I	Co-ordinates of the start of connection in the sender	No change
KK	I	Time step	No change
KT	I	Time step index of the receiver	No change
DX1, DX2	R	Space grid length in the sender (DX1) and in the receiver (DX2)	No change

```
      SUBROUTINE      JOINTZ      (IG1,JG1,IG2,JG2,Z1,Z2,DZ1,DZ2,IS,IE,JS,JE,      &
      ISS,JSS,KK,KT,DX1,DX2)
      DIMENSION Z1(IG1,JG1,2),Z2(IG2,JG2,2)
      DIMENSION DZI(IG1,JG1,2),DZ2(IG2,JG2,2)
      IF(KT.EQ.1) GO TO 200
      IF(MOD(KK,KT).NE.KT/2+1)RETURN
200   CONTINUE
      DO 10 I=IS,IE
      DO 10 J=JS,JE
      IF(DX2-DX1) 20,21,20
20    II=ISS+3*(I-IS)
      JJ=JSS+3*(J-JS)
      GO TO 22
21    II=ISS+(I-IS)
      JJ=JSS+(J-JS)
22    Z2(I,J,2)=Z1(II,JJ,2)
      DZ2(I,J,2)=DZ1(II,JJ,2)
10    CONTINUE
      RETURN
      END
```

(5) MOTION (Computation of the equation of motion)

Objective

Computation of the water discharge at the next time step with the equation of motion.

Method of Connection

CALL MOTION (IG, JG, Z, M, N, DZ, DM, DN, HZ, HM, HN, IR, IB, R, FM, DT, KK, KT)

Parameter	Type	Content before calling the subroutine	Content when returned from the subroutine
JG, IG	I	Indices of Z, M, N, DZ, DM, DN, HZ, HM, HN, IR, IB, in the main programme	No change
Z	R (3-D array) (JG, IG, 2)	Water level	No change
M	R (3-D array) (JG, IG, 2)	Discharge in the I-direction given as M(I, J, 1)=M(I, J, 2)	M(I, J, 2) newly computed
N	R (3-D array) (JG, IG, 2)	Discharge in the j-direction given as N(I, J, 1)=N(I, J, 2)	N(I, J, 2) newly computed
DZ	R (3-D array) (JG, IG, 2)	Total water depth at points for water level	No change
DM	R (3-D array) (JG, IG, 2)	No need of input	Total water depth at points for M
DN	R (3-D array) (JG, IG, 2)	No need of input	No change
HZ	R (2-D array) (JG, IG)	Still water depth at points for water level	No change
HM	R (2-D array) (JG, IG)	Still water depth at points for M	No change
HN	R (2-D array) (JG, IG)	Still water depth at points for N	No change
IR	R (2-D array) (JG, IG)	Map of the existence of breakwaters (positive integer of two figures)	No change
IB	R (2-D array) (JG, IG)	Map of the selection of theory (linear or nonlinear) and of the existence of vertical walls (positive integer of two figures)	No change
R	R	DT/DX; RATIO of time-to-space grid length	No change
FM	R	Manning`s roughness in s/m$^{1/3}$	No change
DT	R	Time step length	
KK	I	Time step	
KT	I	Time step index	

```
         ROUTINEMOTION(IG,JG,IS,IE,JS,JE,Z,M,N,DZ,DM,DN,HZ,HM,HN,& IR,IB,R,FM,DT,KK,KT)
         REAL M,N
         DIMENSION Z(IG,JG,2),M(IG,JG,2),N(IG,JG,2)
         DIMENSION DZ(IG,JG,2),DM(IG,JG,2),DN(IG,JG,2)
         DIMENSION HZ(IG,JG),HM(IG,JG),HN(IG,JG)
         DIMENSION IR(IG,JC),IB(IG,JG)
         DATA GG,GX/9.8,1.0E-5/
         IF(KT.EQ.1)GO TO 200
         IF(MOD(KK,KT).NE.KT/2+1)RETURN
   200   CONTINUE
         DO 10 I=IS,IE
         DO 10 J=JS,JE
         DMl=0.25*(Z(I,J,1)+Z(I,J,2)+Z(I+1,J,1)+Z(I+1,J,2))+HM(I,J)
         DM2=0.5*(Z(I,J,2)+Z(I+1,J,2))+HM(I,J)
         DNl=0.25*(Z(I,J,1)+Z(I,J,2)+Z(I,J+1,1)+Z(I,J+1,2))+HN(I,J)
         DN2=0.5*(Z(I,J,2)+Z(I,J+1,2))+HN(I,J)
         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
         FN=0.5*DT*GG*FM**2
         DO 20 I=IS,IE
         DO 20 J=JS,JE
         IBB=IB(I,J)/10
         IBR=MOD(IB(I,J),10)
         IF(IBB.EQ.4)GO TO 20
         IF(IBB.EQ.1.0R.IBB.EQ.3)GO TO 30
         IF(I.EQ.IG)GO TO 30
         IRR=IR(I,J)/10
         IF(IRR.EQ.1.OR.IRR.EQ.3)GO TO 60
         IF(IBR.EQ.0)GO TO 33
         IF(DZ(I.J.2))31,31,32
   31    IF(DZ(I+1,J,2))30,30,34
   32    IF(DZ(I+1,J,2))35,35,36
   34    IF(Z(I+1,J,2)+HZ(I,J))30,30,37
   35    IF(Z(I,J,2)+HZ(I+1,J))30,30,38
   36    DD=DM(I,J,2)
         GO TO 39
   37    DD=0.5*(Z(I,J,2)+Z(I+1,J,2))+HZ(I,J)
         GO TO 39
   38    DD=0.5*(Z(I,J,2)+Z(I+1,J,2))+HZ(I+1,J)
         GO TO 39
   33    DD=HM(I,J)
          IF(DD.LT.0.1)GO TO 30
   39    XNN=0.25*(N(I,J,1)+N(I+1,J,1)+N(I,J-1,1)+N(I+1,J-1,1))
         DF=DD
         IF(DF.LT.1.0E-2)DF=1.0E-2
         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))
         IF(IBR.EQ.0)GO TO 40
         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 40
         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
42    IF(DM(I-1,J,1).LT.GX)CO TO 40
      XM=XM-R*(M(I,J,1)**2/DM(I,J,1)-M(I-1,J,1)**2/DM(I-1,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 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
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).GT.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))
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
      GO TO 100
60    IRR=MOD(IR(I,J),10)
      Z1=Z(I,J,2)+HM(I,J)
      Z2=Z(I+1,J,2)+HM(I,J)
      ZZ=Z1
      ZX=Z2
      IF(Z1.GT.Z2)GO TO 61
      ZZ=Z2
      ZX=Z1
61    IF(ZZ.LT.GX)GO TO 30
      IF(ZZ*0.66667.LT.ZX)GO TO 62
      XM=1.55*ZZ**1.5
      GO TO 63
62    XM=4.029*ZX*SQRT(ZZ-ZX)
63    IF(Z2.GT.Z1)XM=-XM
      M(I,J,2)=XM
100   CONTINUE
      IF(IBB.GE.2)GO TO 130
      IF(J.EQ.JG)GO TO 130
      IRR=IR(I,J)/10
      IF(IRR.GE.2)GO TO 160
      IF(IBR.EQ.0)GO TO 133
      IF(DZ(I,J,2))131,131,132
131   IF(DZ(I,J+1,2))130,130,134
132   IF(DZ(I,J+1,2))135,135,136
134   IF(Z(I,J+1,2)+HZ(I,J))130,130,137
135   IF(Z(I,J,2)+HZ(I,J+1))130,130,138
136   DD=DN(I,J,2)
      GO TO 139
137   DD=0.5*(Z(I,J,2)+Z(I,J+1,2))+HZ(I,J)
      GO TO 139
138   DD=0.5*(Z(I,J,2)+Z(I,J+1,2))+HZ(I,J+1)
      GO TO 139
133   DD=HN(I,J)
      IF(DD.LT.0.1)GO TO 130
139   XMM=0.25*(M(I,J,1)+M(I,J+1,1)+M(I-1,J,1)+M(I-1,J+1,1))
      DF=DD
      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))
      IF(IBR.EQ.0)GO TO 140
      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 140
      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
142   IF(DN(I,J-1,1).LT.GX)GO TO 140
      XN=XN-R*(N(I,J,1)**2/DN(I,J,1)-N(I,J-1,1)**2/DN(I,J-1,1))
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 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
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 140
      XN=XN-R*(N(I,J,1)*XMM/DN(I,J,1) & -N(I-1,J,1)*XME/DN(I-1,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 20
130   N(I,J,2)=0.0
      GO TO 20
160   IRR=MOD(IR(I,J),10)
      Z1=Z(I,J,2)+HN(I,J)
      Z2=Z(I,J+1,2)+HN(I,J)
      ZZ=Z1
      ZX=Z2
      IF(Z1.GT.Z2)GO TO 161
      ZZ=Z2
      ZX=Z1
161   IF(ZZ.LT.GX)GO TO 130
      IF(ZZ*0.66667.LT.ZX)GO TO 162
      XN=1.55*ZZ**1.5
      GO TO 163
162   XN=4.029*ZX*SQRT(ZZ-ZX)
163   IF(Z2.GT.Z1)XN=-XN
      N(I,J,2)=XN
20    CONTINUE
      RETURN
      END
```

(6) JOINTQ (Connection of the water discharge in space and time)

Objective

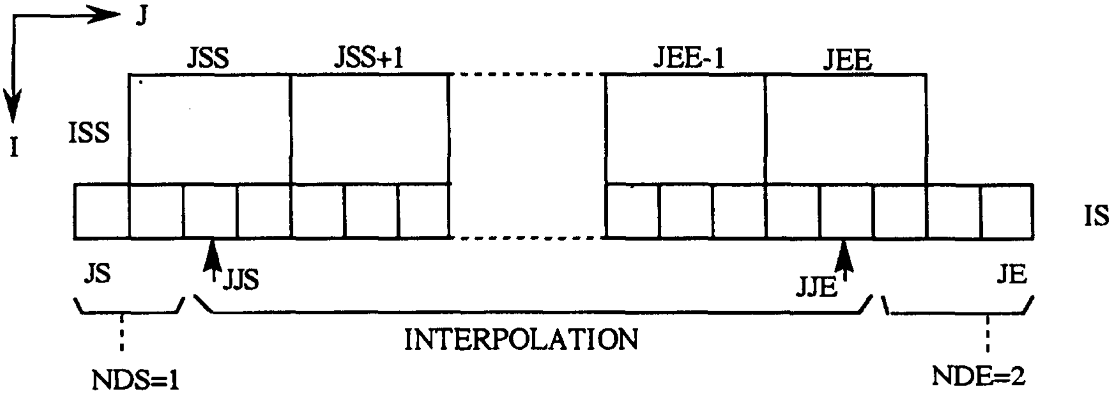
Connection of the water level between computation domains of different Δx and Δt.

Method of Connection

CALL JOINTQ (IG1, JG1, IG2, JG2, M1, M2, N1, N2, IS, IE, JS, JE, ISS, JSS, NDS, INS, NDE, INE, KK, KT1, KT2, DX1, DX2)

Parameter	Type	Content Before Calling the Subroutine	Content When Returned from the Subroutine
IG1, IG2	I	Indice of M1 and N1 in the main programme	No change
JG1, JG2	I	Indice of M2 and N2 in the main programme	No change
M1, N1	R (3-D array) (JG1, IG1, 2)	Discharge in the I-and J-directions in the sender (domain of coarse grids)	No change
M2, N2	R (3-D array) (JG2, IG2, 2)	Discharge in the I-and J-directions in the receiver (domain of fine grids)	Value after connection
IS, IE JS, JE	I	Co-ordinates of the start of connection (IS, JS) and the end (IE, JE) in the receiver	No change
ISS, JSS	I	Co-ordinates of the start of connection in the sender	No change
NDS, NDE	I	Number of extra grids at the start and end in the receiver	No change
INS, INE	I	Computation at the connection boundary; 1 for extrapolation and 2 for interpolation	No change
KK	I	Time step	No change
KT1, KT2	I	Time step index (1,3,9) (KT1<KT2)	No change
DX1, DX2	R	Space grid length in the sender (DX1) and in the receiver (DX2)	No change

Note: For the meanings of NDS and NDE, refer the following figure.



```
      SUBROUTINE JOINTQ(IG1,JG1,IG2,JG2,M1,M2,N1,N2,IS,IE, & JS, JE, ISS, JSS, NDS, INS,
      NDE, INE, KK, KT1, KT2, DX1, DX2)
      REAL M1,N1,M2,N2
      DIMENSION
      M1(IG1,JG1,2),M2(IG2,JG2,2),N1(IG1,JG1,2),N2(IG2,JG2,2)
      KY=KT1/2+1
      IF(KT1.EQ.1) GO TO 100
      IF(MOD(KK,KT1).NE.KY) RETURN
 100  IF(IS.NE.IE) GO TO 200
      J3=NDS+1
      J4=NDE+1
      JJS=JS+J3
      JJE=JE-J4
      JEE=JSS+(JJE-JJS)/3
      DO 10 J=JS,JE
      IF(DX1-DX2) 51,50,51
 51   IF(J.GE.JJS.AND.J.LE.JJE) GO TO 20
      IF(J.LT.JJS) GO TO 30
      IF(J.GT.JJE) GO TO 40
 20   JF=JSS+(J-JJS)/3
      F=FLOAT(3-MOD(J-JJS,3))
      GO TO 60
 30   IF(INS.EQ.2) GO TO 35
      JF=JSS
      F=3.0+FLOAT(JJS-J)
      GO TO 60
 35   JF=JSS-1
      F=FLOAT(JJS-J)
      GO TO 60
 40   IF(INE.EQ.2) GO TO 45
      JF=JEE-1
      F=FLOAT(JJE-J)
      GO TO 60
 45   JF=JEE
      F=FLOAT(3-(J-JJE))
      GO TO 60
 50   JF=J-(JS-JSS+NDS)
      F=3.0
 60   XM2=(F*M1(ISS,JF,2)+(3.0-F)*M1(ISS,JF+1,2))/3.0
      XM1=(F*M1(ISS,JF,1)+(3.0-F)*M1(ISS,JF+1,1))/3.0
      IF(KT1.EQ.KT2) GO TO 101
      KX=MOD(KK,KT2)
      IF(KT1.EQ.1) GO TO 103
      IF(KX.EQ.KY) GO TO 102
      IF(KX.EQ.KY+2*KT1) GO TO 101
      GO TO 104
 103  IF(KX.EQ.0) GO TO 101
      IF(KX.EQ.1) GO TO 102
 104  M2(IS,J,2)=(2.0*XM2+XM1)/3.0
      GO TO 10
 101  M2(IS,J,2)=XM2
      GO TO 10
 102  M2(IS,J,2)=(4.0*XM2-XM1)/3.0
 10   CONTINUE
      RETURN
 200  I3=NDS+1
      I4=NDE+1
      IIS=IS+I3
      IIE=IE-I4
      IEE=ISS+(IIE-IIS)/3
```

```
      DO 70 I=IS,IE
      IF(DX1-DX2) 151,150,151
151   IF(I.GE.IIS.AND.I.LE.IIE) GO TO 120
      IF(I.LT.IIS) GO TO 130
      IF(I.GT.IIE) CO TO 140
120   IF=ISS+(I-IIS)/3
      F=FLOAT(3-MOD(I-IIS,3))
      GO TO 160
130   IF(INS.EQ.2) GO TO 135
      IF=ISS
      F=3.0+FLOAT(IIS-I)
      GO TO 160
135   IF=ISS-I
      F=FLOAT(IIS-I)
      GO TO 160
140   IF(INE.EQ.2) GO TO 145
      IF=IEE-I
      F=FLOAT(IIE-I)
      GO TO 160
145   IF=IEE
      F=FLOAT(3-(I-IIE))
      GO TO 160
150   IF=I-(IS-ISS+NDS)
      F=3.0
160   XN2=(F*N1(IF,JSS,2)+(3.0-F)*N1(IF+1,JSS,2))/3.0
      XN1=(F*N1(IF,JSS,1)+(3.0-F)*N1(IF+1,JSS,1))/3.0
      IF(KT1.EQ.KT2) GO TO 201
      KX=MOD(KK,KT2)
      IF(KT1.EQ.1) GO TO 203
      IF(KX.EQ.KY) GO TO 202
      IF(KX.EQ.KY+2*KT1) GO TO 201
      GO TO 204
203   IF(KX.EQ.0) GO TO 201
      IF(KX.EQ.1) GO TO 202
204   N2(I,JS,2)=(2.0*XN2+XN1)/3.0
      GO TO 70
201   N2(I,JS,2)=XN2
      GO TO 70
202   N2(I,JS,2)=(4.0*XN2-XN1)/3.0
70    CONTINUE
      RETURN
      END
```

(7) CHANGE (Change of dat)

Objective

Change the index of time index from 2 to 1, at every time step of computation.

Method of Connection

CALL CHANGE (IG, JG, Z, M, N, DZ, KK, KT).

Parameter	Type	Content Before Calling the Subroutine	Content When Returned from the Subroutine
IG, JG	I	Indices of Z,M,N and DZ and in the main programme	No change
Z	R (3-D array) (JG, IG, 2)	Water level	Z(I, J, 2) is equal to Z(I, J, 1)
M	R (3-D array) (JG, IG, 2)	Discharge in the I-direction	M(I, J, 2) is equal to M(I, J, 1)
N	R (3-D array) (JG, IG, 2)	Discharge in the I-direction	N(I, J, 2) is equal to N(I, J, 1)
DZ	R (3-D array) (JG, IG, 2)	Total water depth at points for water level	DZ(I, J, 2) is equal to DZ(I, J, 1)
KK	I	Time step	No change
KT	I	Time step index	No change

```
      SUBROUTINE  CHANGE (IG,JG,Z,M,N,DZ,KK,KT)
      REAL M,N
      DIMENSION Z(IG,JG,2),M(IG,JG,2),N(IG,JG,2),DZ(IG,JG,2)
      LL=MOD(KK,KT)
      LX=KT/2
      IF(LL.NE.0.AND.LL-LX.NE.0)RETURN
      DO 10 I=1,IG
      DO 10 J=1,JG
      IF(LL.NE.0) GO TO 11
      Z(I,J,1)=Z(I,J,2)
      DZ(I,J,1)=DZ(I,J,2)
11    IF(LL-LX.NE.0)GO TO 10
      M(I,J,1)=M(I,J,2)
      N(I,J,1)=N(I,J,2)
10    CONTINUE
      RETURN
      END
```

IUGG/IOC TIME PROJECT

PART 2. PROPAGATION IN THE OCEAN IN THE SPHERICAL CO-ORDINATES TSUNAMI-F1 AND ITS PROGRAMME LIST

CONTENTS

1. INTRODUCTION

Numerical simulations of far-field tsunamis require longer CPU time and larger memory capacity than those of near-field tsunamis, because the computation covers wide areas. The long travel distance may yield dispersion of wave components, and therefore the physical dispersion term should be included; i.e. the equations of higher order approximation should be used. The long travel time yields an inevitable accumulation of numerical error. These factors suggest that the computation programme should be carefully designed.

In the present text, a method of simulation is explained. The linear long wave theory is expressed in the latitude-longitude co-ordinates. The equations are different. A method of setting initial and boundary conditions is stated. A process of computation is explained by an example. A method of output is given. In addition, variables, constants and subroutines are explained. This model is TUNAMI-F1 (Tohoku University's Numerical Analysis Model for Investigation of Far-field Tsunamis, 1).

When the linear theory is used, it is also easy to attain a high rate of vectorization even with the ordinary programming method. The present programme for transoceanic propagation is composed to fully utilize the vectorization ability of super computer. The rate of vectorization of higher than 99% is a result of elimination of both the IF-sentences in DO-groups and the division operation.

2. METHOD OF COMPUTATION

2.1 CO-ORDINATES AND AREA FOR COMPUTATION

For near field tsunamis, the Cartesian co-ordinate system is used, because areas included in simulations are not wide. On the other hand, in numerical simulations of far-field tsunamis which travel more than 1,000 km over the ocean, the polar co-ordinates should be used. As shown in Figure 1, the earth is considered as a sphere of radius R, covered by the latitude and longitude (θ,λ).

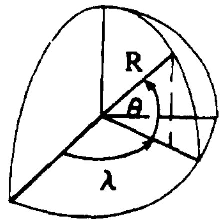


Figure 1. Spherical Co-ordinates

The area in computation is taken wide enough to simulate the whole process of the transoceanic propagation of such tsunamis as the 1960 Chilean tsunami and the 1964 Alaska tsunami. Figure 2 shows the area used in the present example. Since the figure is drawn in Mercator's projection, the length at high latitude in this map is different from the actual length. The map covers the area from

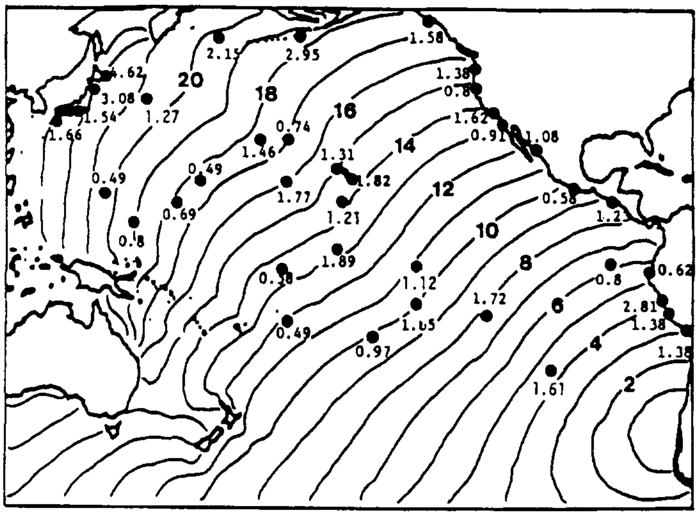


Figure 2. Area for Computation, and Propagation of the Chilean tsunami

S60° to N63° and from E120° to W70°. The water depth is given at every 5' in longitude and latitude. The space grid length of 10' is used in the computation. The total number of grids amounts to 1,020 x 738 = 752,760.

2.2 GOVERNING EQUATIONS AND GRID LENGTH

In computation of a far-field tsunami, the dispersion term becomes important because the long travel distance acts to disperse wave components. The linear Boussinesq equation which includes the physical dispersion term is considered appropriate to express this effect. In the following, another simple method is explained to replace the physical dispersion term by the numerical dispersion term which is inevitably resulted as the truncation error of a numerical scheme. This replacement is possible, if the grid length is appropriately selected. Then, the linear long wave theory of lower order of approximation becomes almost equivalent to the linear Boussinesq equation of higher order of approximation. In addition, due to this replacement, the CPU time and the computer memory are much saved.

The linear long wave theory is given by the following expression in the latitude-longitude co-ordinates.

$$\frac{\partial\eta}{\partial t}+\frac{1}{R\cos\theta}\left[\frac{\partial M}{\partial\lambda}+\frac{\partial}{\partial\theta}(N\cos\theta)\right]=0 \qquad (1)$$

$$\frac{\partial M}{\partial t}+\frac{gh}{R\cos\theta}\frac{\partial\eta}{\partial\lambda}=fN \qquad (2)$$

$$\frac{\partial N}{\partial t}+\frac{gh}{R}\frac{\partial\eta}{\partial\theta}=-fM \qquad (3)$$

where η is the water level, M and N are discharge fluxes in the λ (along a parallel of latitude) and θ (along a circle of longitude) directions, g is the gravitational acceleration, and f ($2\omega\sin\theta$) is the Coriolis coefficient.

The leap-frog scheme is applied to obtain difference expressions of Eqs.(1) through (3). Then, we have

$$\frac{\eta^{n+\frac{1}{2}}_{j,m}-\eta^{n-\frac{1}{2}}_{j,m}}{\Delta t}+\frac{1}{R\cos\theta_m}\left[\frac{M^n_{j+\frac{1}{2},m}-M^n_{j-\frac{1}{2},m}}{\Delta\lambda}+\frac{N^n_{j,m+\frac{1}{2}}\cos\theta_{m+\frac{1}{2}}-N^n_{j,m-\frac{1}{2}}\cos\theta_{m-\frac{1}{2}}}{\Delta\theta}\right]=0 \qquad (4)$$

$$\frac{M^{n+1}_{j+\frac{1}{2},m}-M^n_{j+\frac{1}{2},m}}{\Delta t}+\frac{gh_{j+\frac{1}{2},m}}{R\cos\theta_m}\frac{\eta^{n+\frac{1}{2}}_{j+1,m}-\eta^{n+\frac{1}{2}}_{j,m}}{\Delta\lambda}=fN' \qquad (5)$$

$$\frac{N^{n+1}_{j+\frac{1}{2},m} - N^{n}_{j+\frac{1}{2},m}}{\Delta t} + \frac{gh_{j+\frac{1}{2},m}}{R\sin\theta_m}\frac{\eta^{n+\frac{1}{2}}_{j,m+1} - \eta^{n+\frac{1}{2}}_{j,m}}{\Delta\theta} = -fM' \qquad (6)$$

where

$$N' = \frac{1}{4}\left[N^{n}_{j+1,m+\frac{1}{2}} + N^{n}_{j+1,m-\frac{1}{2}} + N^{n}_{j,m+\frac{1}{2}} + N^{n}_{j,m-\frac{1}{2}}\right]$$

$$M' = \frac{1}{4}\left[M^{n}_{j+\frac{1}{2},m+\frac{1}{2}} + M^{n}_{j+\frac{1}{2},m} + M^{n}_{j-\frac{1}{2},m+1} + M^{n}_{j-\frac{1}{2},m}\right] \qquad (7)$$

The unkowns, η, M and N, are given by the following explicit expressions.

$$\eta^{n+\frac{1}{2}}_{j,m} = \eta^{n-\frac{1}{2}}_{j,m} - R_1\left[M^{n}_{j+\frac{1}{2},m} - M^{n}_{j-\frac{1}{2},m} + N^{n}_{j,m+\frac{1}{2}}\cos\theta_{m+\frac{1}{2}} - N^{n}_{j,m-\frac{1}{2}}\cos\theta_{m-\frac{1}{2}}\right] \qquad (8)$$

$$M^{n+1}_{j+\frac{1}{2},m} = M^{n}_{j+\frac{1}{2},m} - R_2 \cdot h_{j+\frac{1}{2},m}\left[\eta^{n+\frac{1}{2}}_{j+1,m} - \eta^{n+\frac{1}{2}}_{j,m}\right] + R_3 N' \qquad (9)$$

$$N^{n+1}_{j,m+\frac{1}{2}} = M^{n}_{j,m+\frac{1}{2}} - R_4 \cdot h_{j,m+\frac{1}{2}}\left[\eta^{n+\frac{1}{2}}_{j+1,m} - \eta^{n+\frac{1}{2}}_{j,m}\right] + R_5 M' \qquad (10)$$

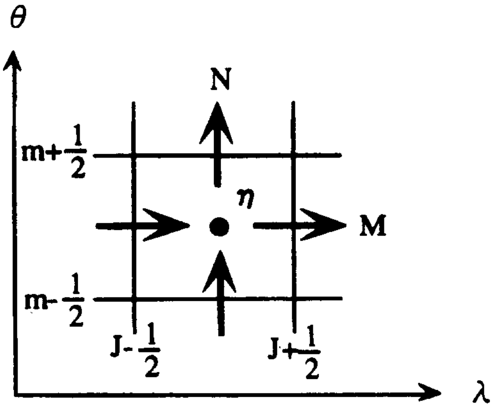


Figure 3. Computation Points for Water Level and Discharge

where R'_ns a re given as follows.

$$R_1 = \Delta t/\left(R\cos\theta_m \Delta s\right)$$

$$R_2 = g\Delta t/\left(R\cos\theta_m \Delta s\right)$$

$$R_3 = 2\Delta t\omega\sin\theta_m$$

$$R_4 = g\Delta t/\left(R\Delta s\right)$$

$$R_5 = 2\Delta t\omega\sin\theta_{m+\frac{1}{2}} \qquad (11)$$

A point of computation is numbered as (j, m, n) in the (θ, λ, t) directions. The grid lengths are (Δθ, Δλ, Δt) In the present computation grid lengths in the latitude and longitude directions are taken equal: i.e. Δs = Δθ = Δλ. The angular velocity of earth rotation is given by ω.

The spatial grid length is taken to be 10', and the time step interval is taken to be 20 seconds. This spatial grid length is selected to satisfy the condition that the Imamura number Im given below is nearly equal to unity, or in other words, the numerical and physical dispersion effects are nearly the same.

$$Im = \Delta x / 2h\sqrt{1 - K^2} \qquad \textbf{(12)}$$

where K is the Courant number (=(gh)$^{1/2}$Δt/Δx.) and h the mean water depth in the ocean under consideration. The time step interval is determined to satisfy the C.F.L. condition for the spatial grid length thus determined.

2.3 INITIAL AND BOUNDARY CONDITIONS

The initial condition, the vertical displacement of water surface is usually considered the same as the vertical displacement of sea bottom which is calculated with the Mansinha and Smylie method (1972), if fault parameters are given. When the actual vertical displacement of ground is available even though it is very rare, as in case of the 1964 Great Alaska earthquake, the better is to use this actual displacement. In case of a huge earthquake, other effects of dynamic fault motion such as the rise time and the rupture velocity may not be negligible. However, when the effect of a tsunami at a very remote place is studied, these dynamic factors are not influential.

The boundary condition on land is the perfect reflection. The velocity component (or, discharge flux component) normal to the land boundary is made equal to zero.

At the open sea boundary, the boundary condition is the free transmission. In terms of u the linear free transmission condition is,

$$\begin{aligned} \eta &= +\sqrt{h/g}\cdot && \textit{for advancing waves} \\ &= -\sqrt{h/g}\cdot u && \textit{for receding waves} \qquad \textbf{(13)} \end{aligned}$$

where η is the water level and u the velocity. In terms of water discharge Q, the condition is given as follows, on assuming that η << h.

$$\begin{aligned} \eta &= +Q/\sqrt{gh} && \textit{for advancing waves} \\ &= -Q/\sqrt{gh} && \textit{for receding waves} \qquad \textbf{(14)} \end{aligned}$$

Here, two judgements are necessary One is the evaluation of the discharge flux Q. Another is the judgement of the direction of wave propagation. On referring Figure 4, the resultant discharge flux is simply determined by

$$Q = \sqrt{\frac{(M_1 + M_2)^2}{4} + N_2^2} \qquad \textbf{(15)}$$

T

wave propagation direction can be judged by the sign of N_2 in relation to the alignment of the open sea boundary.

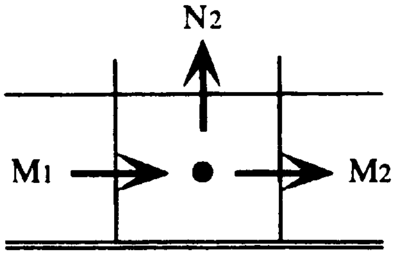


Figure 4. Distinction of Advancing and Receding Waves by the sign of N_2

2.4 FLOW CHART

In Figure 5, the flow of computation is shown with the names of subroutines, the details of which are described in Section 3.

3. TSUNAMI SIMULATION PROGRAMMES

3.1 AN EXAMPLE OF PROGRAMME

Attached are the programme sheets of TUNAMI-F1 applied to the 1990 Mariana tsunami, with brief explanations and comments. Pages 1 to 10 of the programme sheets are the tsunami simulation with the method describe above. Pages 11 to 13 are for the computation to determine the initial tsunami profile with the method given in the following subsection.

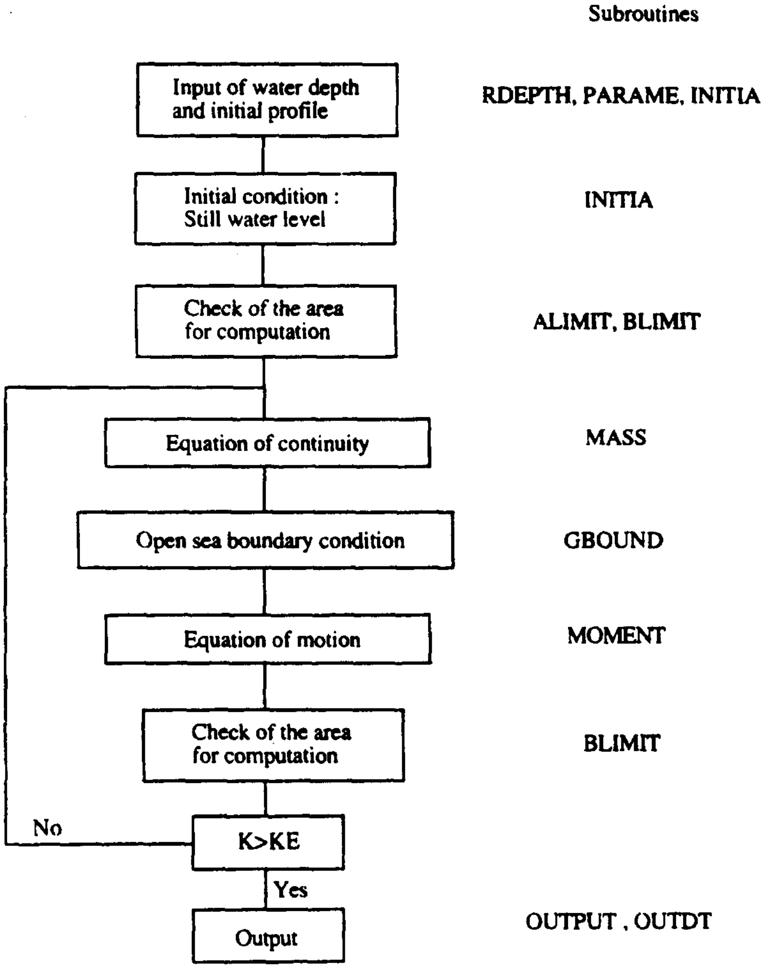


Figure 5. Flow Chart of the Computation

3.2 VARIABLES AND CONSTANTS IN TSUNAMI SIMULATION

Variables in the programme are;

Z: Water level.
M,N: Discharge flux.
H: Still water depth.
PZ: Time history of water level.
IP,JP: Co-ordinates of points for output of the history of water level.
V1,V2,V3,V4,V5,V6,V7: Working arrays for vector operation.

Coefficients are;

ZM: Highest water level.
ZN: Lowest water level.
LF: Arrival time in hour.
R1,R2,R3,R4,R5,R6: Coefficients given by Eq.(11), and R6 = $\cos\theta_{m+1/2}$
C1= $\theta_{m+1/2}$ in radian.
C2= θ_m in radian.
C3= $\theta_{m-1/2}$ in radian.
C4= h : Water depth.

Constants are;

GG: Gravitational acceleration.
PP: Circular constant π (= 3.1415926).
R: Radius of the Earth.

The computation is controlled by the following conditions.

IG,JG: Size of the area for computation in longitude and latitude.
FL: Latitude of the southernmost end of the area for computation.
IS,JS,IE,JE: Area where the tsunami exists and the computation is carried out.
DS,DT: Grid length in minute, and time step length in second.
KS,KE: Time steps of beginning and end of the computation.
NG: Number of spatial points where the time history of water level is output.
KC: Time step length in outputting the time history of water level.
KD: Time step length to output spatial wave profiles.

3.3 SUBROUTINES IN TSUNAMI SIMULATION

Subroutines indicated in Figure 5 are as follows.

RDEPTH: Data input of water depth.
PARAME: Setting of parameters required in vectorized computation.
INITIA: Input of the initial condition and the initial profile.
ALIMIT: Making the area of computation be within the area under consideration.
BLIMINT: Enlargement of the area of computation as the tsunami propagates.
OUTPUT: Output and display of the spatial distribution of water level at an instant.
CLOCK: Time from the beginning of the computation, special for a NEC SX-1.
MASS: Computation of the equation of continuity.
GBOUND: Open sea, boundary condition.
MOMENT: Computation of the equation of motion.
MAX: Check of the highest and lowest water level.
POINT: Output of the time history of water level at the point (IP, JP).
PROPA: Output of the tsunami arrival time in hour.
OUTDT: Output of the highest and lowest water levels, and the arrival time.
FILEOT: Output of the water level and the discharge flux.

3.4 VARIABLES AND CONSTANTS IN INITIAL PROFILE COMPUTATION

The vertical displacement of sea bottom is calculated with the Mansinha and Smylie method (1972), and is assumed equal to the tsunami initial profile with no modification of hydraulic effect because the horizontal size of the initial profile is sufficiently large compared to the water depth at the tsunami source.

In the attached programme sheets, the subroutine DEFORM is used to calculate the initial tsunami profile from the given fault parameters.

Variables in the programme are, on referring Figs. 7 through 9;

DX: Grid length in meter for the Cartesian co-ordinates.
DR: Grid length in degree for the spherical co-ordinates.
H: Depth of a corner of the fault plane in meter.
D: Dislocation of the upper plane (u) in meter.
DL: Dip angle (8) in degree
TH: Strike angle (O) in degree measured clockwise from north.
RD: Direction of dislocation (A) in degree.
L: Length of fault plane in meter.
W: Width of fault plane in meter.
Y0, X0: Co-ordinate of the origin in the area for tsunami computation.

$\bar{Y}0$, $\bar{X}0$: Co-ordinate of the origin of the fault plane.

Constants are;

A: Circular constant π (= 3.1415926).
RR: Radius of the earth (= 6.37E + 6 (m)).
E: = 1.7453E - 3(m).

3.5 SUBROUTINES IN INITIAL PROFILE COMPUTATION

Subroutines are;

DEFORM: Computation of the initial profile.
USCAL: Computation of the vertical displacement due to the strike slip component.
UDCAL: Computation of the vertical displacement due to the dip slip component.

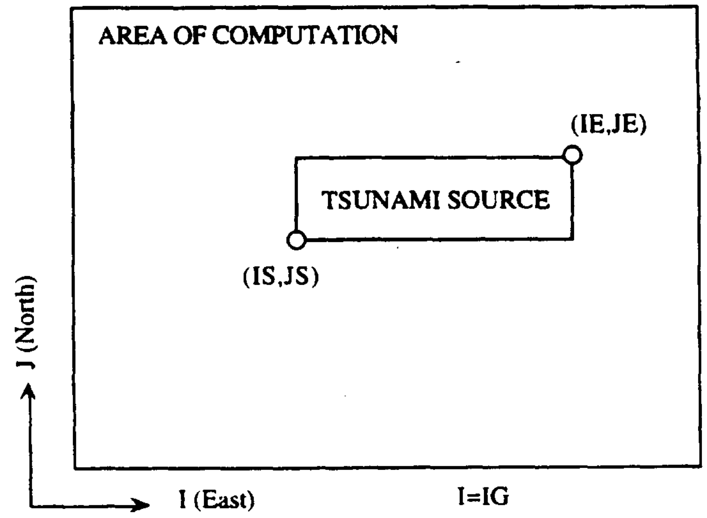


Figure 6. Co-ordinates in the Areas for Computation of Tsunami and Initial Profile

4. CONCLUDING REMARKS

In addition to TUNAMI-F1, the Disaster Control Research Center, Tohoku University provides four programs.

TUNAMI-F2 is for the simulation of a far-field tsunami, combining the transoceanic propagation and the detailed simulation in coastal water with the linear long wave equations.

For a near-field tsunami, there are three programs.

The first is TUNAMI-N1, with the linear theory with constant spatial grid length. The authors recommend to begin with this programme, because it is easy to understand programs. Reader may refer "NUMERICAL METHOD OF TSUNAMI SIMULATION WITH THE LEAP-FROG SCHEME" written by Goto and Ogawa, and translated by Shuto.

The second is TUNAMI-N2, with the linear theory in deep sea 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 programme.

The third is TUNAMI-N3, in which the spatial grid length varies from coarse grids in deep sea to fine grids in shallow sea. The linear long wave theory is used.

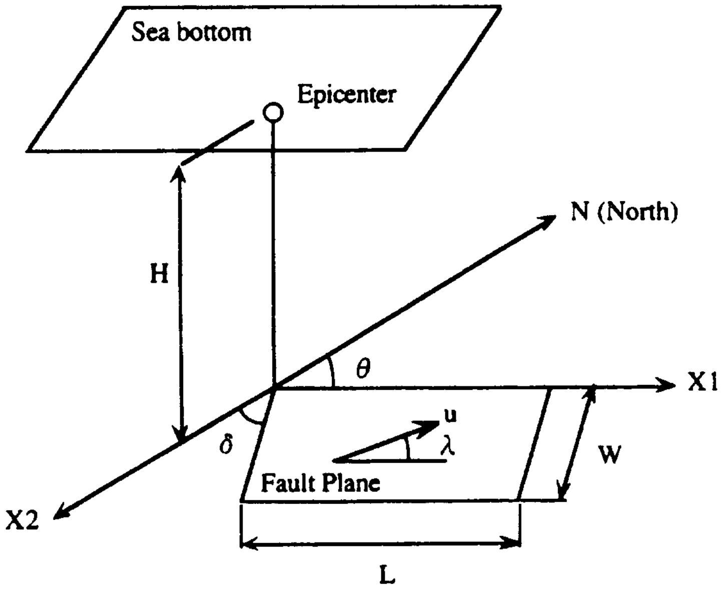


Figure 7. Fault Parameter

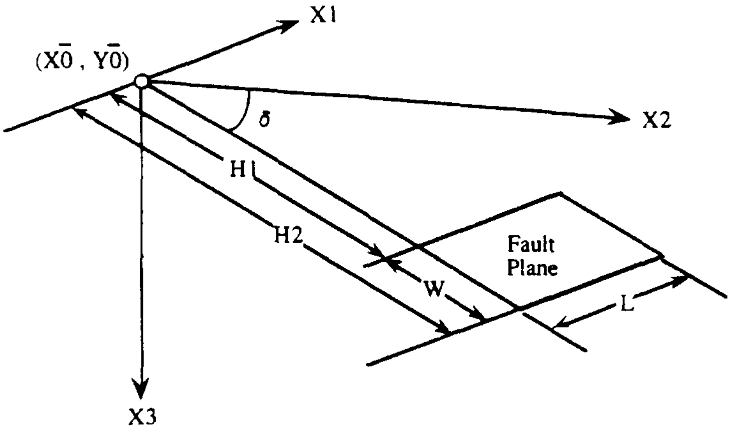


Figure 8 Plan of Fault Plane

Figure 9. Fault Plane in Three Dimensions

```
C
C    *************** TUNAMI-F1 ****************
C      Tohoku University's Numerical-Analysis Model
C         for Investigation of tsunami
C          Far-field Tsunami version
C
C          1990 MARIANA  TSUNAMI (DX=5')
C         --- LINEAR LONG WAVE THEORY ----
C              NUMERICAL EXPERIMENT
C                                  91.4.11
C
      PARAMETER (IG=480,JG=600)
      PARAMETER (DS=5.,DT=10.)
      PARAMETER (KS=1,KE=360*10,KD=360*2,KC=1)
C
      REAL M,N
      DOUBLE PRECISION TS,TE,TD
      DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG),H(IG,JG)
      DIMENSION R1(IG,JG),R2(IG,JG),R3(IG,JG)
      DIMENSION R4(IG,JG),R5(IG,JG),R6(JG)
      DIMENSION C1(IG),C2(JG),C3(IG),C4(JG)
      DIMENSION V1(JG),V2(JG),V3(JG),V4(JG),V5(JG)
      DIMENSION ZM(IG,JG),ZN(IG,JG),LF(IG,JG)
C
      CALL RDEPTH (IG,JG,H)
      WRITE(6,255)
  255 FORMAT(1H1,5X,"READ DEPTH OK!")
      CALL PARAME (IG,JG,IR,JR,ID,JD,DS,DT,
     &          R1,R2,R3,R4,R5,R6,C1,C2,C3,C4,
     &          V1,V2,V3,V4,V5,H)
      CALL INITIA (IG,JG,IR,JR,ID,JD,
     &          Z,M,N,H)
      CALL ALIMIT (IG,JG,IS,JS,IE,JE,IR,JR,ID,JD)
      CALL BLIMIT (IG,JG,IS,JS,IE,JE,Z)
C     CALL OUTPUT (IG,JG,IS,JS,IE,JE,0,KD,Z,H)
      CALL CLOCK (TS)
      DO 20 K=KS,KE
      KK=K  ; IF(MOD(KK,50).EQ.0)WRITE(6,*)KK
      CALL MASS (IG,JG,IS,JS,IE,JE,
     &          Z,M,N,R1,R6)
      CALL GBOUND (IG,JG,IS,JS,IE,JE,
     &          Z,M,N,C1,C2,C3,C4)
      CALL MOMENT (IG,JG,IS,JS,IE,JE,
     &          Z,M,N,R2,R3,R4,R5,
     &          V1,V2)
      CALL OUTPUT (IG,JG,IS,JS,IE,JE,KK,KD,Z,H)
      CALL BLIMIT (IG,JG,IS,JS,IE,JE,Z)
C
      CALL MAX (IG,JG,IS,JS,IE,JE,Z,H,ZM,ZN)
      CALL POINT (IG,JG,Z,KK,KC)
      CALL PROPA (IG,JG,IS,JS,IE,JE,Z,H,LF,KK)
C
   20 CONTINUE
      CALL OUTDT (IG,JG,ZM,ZN,LF)
      CALL CLOCK (TE) ; TD=TE-TS
      WRITE(6,*)KK,TD
      STOP
      END
C
```

Data and initial condition input

Main program

Main computation

Output of computed results

```
      SUBROUTINE RDEPTH (IG,JG,H) ------ Input of water depth
C
      DIMENSION H(IG,JG)
C
      DO 10 J=1,JG
      READ(20,100)(H(I,J),I=1,IG)
  100 FORMAT(20F6.1)
   10 CONTINUE
      RETURN
      END
C
      SUBROUTINE PARAME (IG,JG,IR,JR,ID,JD,DS,DT, ----- Setting of parameters
     &                  R1,R2,R3,R4,R5,R6,C1,C2,C3,C4,
     &                  V1,V2,V3,V4,V5,H)
C
      PARAMETER (GG=9.8)
      PARAMETER (FL=0.0,PP=3.1415926)
      PARAMETER (R=6.37E+6,WW=7.2722E-5)
      PARAMETER (IS=240,JS=120,IE=360,JE=240)
C
      DIMENSION R1(IG,JG),R2(IG,JG),R3(IG,JG),R4(IG,JG),R5(IG,JG)
      DIMENSION R6(JG),C1(IG),C2(JG),C3(IG),C4(JG)
      DIMENSION V1(JG),V2(JG),V3(JG),V4(JG),V5(JG)
      DIMENSION H(IG,JG)
C
      IR=IS
      JR=JS
      ID=IE
      JD=JE
      PX=PP/180.
      DX=R*(DS/60.)*PX
      RR=DT/DX
      RS=0.5*GG*RR
      RT=0.5*DT*WW
      DO 10 J=1,JG
      C2(J)=(FL+(DS*J-0.5*DS)/60.)*PX
      C4(J)=(FL+DS*J/60.)*PX                  -- Setting of C1 and C2
   10 CONTINUE
      DO 11 J=1,JG
      R6(J)=COS(C2(J))
      V1(J)=RR/R6(J)
      V2(J)=RS/R6(J)
      V3(J)=RT*SIN(C2(J))                     -- Setting of C3 and R6
      V4(J)=RS
      V5(J)=RT*SIN(C4(J))
   11 CONTINUE
      DO 12 J=1,JG
      C4(J)=0.
      DO 12 I=1,IG                            -- Setting of C4
      IF(H(I,J).GT.C4(J))C4(J)=H(I,J)
   12 CONTINUE
```

```
1110       DTMIN=1.E+20
1120       DXMAX=0.
1130       DXMIN=1.E+20
1140       DO 13 J=1,JG
1150       DXM=DX*R6(J)
1160       DTM=SQRT((1./(GG*C4(J)))/(1./DX**2+1./DXM**2))
1170       IF(DXM.GT.DXMAX)DXMAX=DXM
1180       IF(DXM.LT.DXMIN)DXMIN=DXM
1190       IF(DTM.LT.DTMIN)DTMIN=DTM
1200    13 CONTINUE
1210       WRITE(6,110)DX,DXMAX,DXMIN,DTMIN
1220       DO 20 J=1,JG
1230       DO 20 I=1,IG
1240       R1(I,J)=0.
1250       R2(I,J)=0.
1260       R3(I,J)=0.
1270       R4(I,J)=0.
1280       R5(I,J)=0.
1290    20 CONTINUE
1300       DO 30 J=1,JG
1310       DO 30 I=1,IG
1320       IF(H(I,J).LT.0.)GOTO 30
1330       R1(I,J)=V1(J)
1340       IF(I.NE.IG)THEN
1350       IF(H(I+1,J).LT.0.)GOTO 31
1360       R2(I,J)=V2(J)*(H(I,J)+H(I+1,J))
1370       R3(I,J)=V3(J)
1380       ELSE
1390       R2(I,J)=V2(J)*H(I,J)*2
1400       R3(I,J)=V3(J)
1410       ENDIF
1420    31 IF(J.NE.JG)THEN
1430       IF(H(I,J+1).LT.0.)GOTO 30
1440       R4(I,J)=V4(J)*(H(I,J)+H(I,J+1))
1450       R5(I,J)=V5(J)
1460       ELSE
1470       R4(I,J)=V4(J)*H(I,J)*2
1480       R5(I,J)=V5(J)
1490       ENDIF
1500    30 CONTINUE
1510       DO 41 I=1,IG
1520    41 C1(I)=0.
1530       DO 42 I=1,IG
1540    42 IF(H(I,1).GT.0.)C1(I)=1./SQRT(GG*H(I,1))
1550       DO 43 J=1,JG
1560    43 C2(J)=0.
1570       DO 44 J=1,JG
1580    44 IF(H(1,J).GT.0.)C2(J)=1./SQRT(GG*H(1,J))
1590       DO 45 I=1,IG
1600    45 C3(I)=0.
1610       DO 46 I=1,IG
1620    46 IF(H(I,JG).GT.0.)C3(I)=1./SQRT(GG*H(I,JG))
1630       DO 47 J=1,JG
1640    47 C4(J)=0.
1650       DO 48 J=1,JG
1660    48 IF(H(IG,J).GT.0.)C4(J)=1./SQRT(GG*H(IG,J))
1670       RETURN
1680   110 FORMAT(1H ,"DX,DXMAX,DXMIN,DTMIN=",2X,4E12.4)
1690       END
1700C
```

Max. and min. of DX
Min. of $DT=DX'/(gh)^{1/2}$

Setting of R1 through R5

Setting of coefficients used at the open sea boundary

```
      SUBROUTINE BLIMIT (IG,JG,IS,JS,IE,JE,Z)
C
      PARAMETER (GX=1.E-4)
      DIMENSION Z(IG,JG)
C
      IF(IS.EQ.2)GOTO 61
      L=0
      DO 10 J=JS,JE
      IF(ABS(Z(IS+2,J)).GT.GX)L=1
   10 CONTINUE
      IF(L.EQ.1)THEN
      IS=IS-1
      IF(IS.LE.2)IS=2
      ENDIF
   61 IF(IE.EQ.IG-1)GOTO 62
      L=0
      DO 20 J=JS,JE
      IF(ABS(Z(IE-2,J)).GT.GX)L=1
   20 CONTINUE
      IF(L.EQ.1)THEN
      IE=IE+1
      IF(IE.GE.IG-1)IE=IG-1
      ENDIF
   62 IF(JS.EQ.2)GOTO 63
      L=0
      DO 30 I=IS,IE
      IF(ABS(Z(I,JS+2)).GT.GX)L=1
   30 CONTINUE
      IF(L.EQ.1)THEN
      JS=JS-1
      IF(JS.LE.2)JS=2
      ENDIF
   63 IF(JE.EQ.JG-1)GOTO 64
      L=0
      DO 40 I=IS,IE
      IF(ABS(Z(I,JE-2)).GT.GX)L=1
   40 CONTINUE
      IF(L.EQ.1)THEN
      JE=JE+1
      IF(JE.GE.JG-1)JE=JG-1
      ENDIF
   64 RETURN
      END
C
```

Setting of the area for computation.

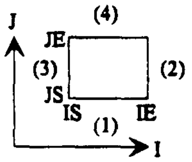
(Enlarge the area if water level is higher than GX(1.0E-4).)

Enlargement in (1) direction.

Enlargement in (2)irection.

Enlargement in (3) direction.

Enlargement in (4) direction.



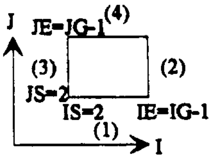
```
      SUBROUTINE INITIA (IG,JG,IS,JS,IE,JE, -----  Setting of the initial
     &                  Z,M,N,H)                    condition
C
      REAL M,N
      DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG),H(IG,JG)
C
      DO 10 J=1,JG ----------+
      DO 10 I=1,IG           |
      Z(I,J)=0.              |--- Still water level is assumed.
      M(I,J)=0.              |
      N(I,J)=0.              |
   10 CONTINUE --------------+
C
      CALL DEFORM (IG,JG,IS,JS,IE,JE,Z) --------- Input of the initial profile.
C
      DO 30 J=JS,JE -----------------+
      DO 30 I=IS,IE                  |
      IF(H(I,J).LT.0.)Z(I,J)=0.      |--- Water level is set zero on land.
   30 CONTINUE ----------------------+
      RETURN
      END
C
      SUBROUTINE ALIMIT (IG,JG,IS,JS,IE,JE,IR,JR,ID,JD)--+
C                                                        |
      IS=IR-1                                            |
      JS=JR-1                                            |
      IE=ID+1                          The area for computation is limited
      JE=JD+1                          within the area of concern.
      IF(IS.LE.2)IS=2
      IF(JS.LE.2)JS=2                  (ALIMIT is not necessary.
      IF(IE.GE.IG-1)IE=IG-1             BLIMIT given below is sufficient)
      IF(JE.GE.JG-1)JE=JG-1
      WRITE(6,100)IS,JS,IE,JE
      RETURN
  100 FORMAT(1H ,"IS=",I5,2X,"JS=",I5,2X,"IE=",I5,2X,"JE=",I5)
      END
C
```

```
      SUBROUTINE MASS (IG,JG,IS,JS,IE,JE,Z,M,N,R1,R6)
C
      REAL M,N
      DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG)
      DIMENSION R1(IG,JG),R6(JG)
C
      DO 10 J=JS,JE
      DO 10 I=IS,IE
      Z(I,J)=Z(I,J)-R1(I,J)*(M(I,J)-M(I-1,J))
     &            -R1(I,J)*(N(I,J)*R6(J)-N(I,J-1)*R6(J-1))
   10 CONTINUE
      DO 20 J=JS,JE
      DO 20 I=IS,IE
      IF(ABS(Z(I,J)).LT.1.0E-5)Z(I,J)=0.0
   20 CONTINUE
      RETURN
      END
C
```

Computation of the equation of continuity.

```
      SUBROUTINE GBOUND (IG,JG,IS,JS,IE,JE,
     &                   Z,M,N,C1,C2,C3,C4)
C
      REAL M,N
      DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG)
      DIMENSION C1(IG),C2(JG),C3(IG),C4(JG)
C
```

Setting of the open sea boundary condition.



```
      IF(JS.LE.2)THEN
      DO 10 I=2,IG-1
      Z(I,1)=SQRT(N(I,1)**2
          &+0.25*(M(I,1)+M(I-1,1))**2)*C1(I)
   10 IF(N(I,1).GT.0.)Z(I,1)=-Z(I,1)
      ENDIF
```

Setting on the line (1).

```
      IF(IS.LE.2)THEN
      DO 20 J=2,JG-1
      Z(1,J)=SQRT(M(1,J)**2
          &+0.25*(N(1,J)+N(1,J-1))**2)*C2(J)
C     IF(Z(1,J).GT.1.0)Z(1,J)=1.0
C     IF(Z(1,J).LT.-1.0)Z(1,J)=-1.0
   20 IF(M(1,J).GT.0.)Z(1,J)=-Z(1,J)
      ENDIF
```

Setting on the line (2).

```
      IF(JE.GE.JG-1)THEN
      DO 30 I=2,IG-1
      Z(I,JG)=SQRT(N(I,JG-1)**2
          &+0.25*(M(I,JG)+M(I,JG-1))**2)*C3(I)
      IF(N(I,JG-1).LT.0.)Z(I,JG)=-Z(I,JG)
   30 CONTINUE
C
      ENDIF
```

Setting on the line (3).

```
      IF(IE.GE.IG-1)THEN
      DO 40 J=2,JG-1
      Z(IG,J)=SQRT(M(IG-1,J)**2
          &+0.25*(N(IG,J)+N(IG,J-1))**2)*C4(J)
   40 IF(M(IG-1,J).LT.0.)Z(IG,J)=-Z(IG,J)
      ENDIF
```

Setting on the line (4).

```
      Z(1,1)=SQRT(M(1,1)**2+N(1,1)**2)*C1(1)
      IF(N(1,1).GT.0.0)Z(1,1)=-Z(1,1)
      Z(IG,1)=SQRT(M(IG-1,1)**2+N(IG,1)**2)*C1(IG)
      IF(N(IG,1).GT.0.0)Z(IG,1)=-Z(IG,1)
      Z(1,JG)=SQRT(M(1,JG)**2+N(1,JG-1)**2)*C3(1)
      IF(N(1,JG-1).LT.0.0)Z(1,JG)=-Z(1,JG)
      Z(IG,JG)=SQRT(M(IG-1,JG)**2+N(IG,JG-1)**2)*C3(IG)
      IF(N(IG,JG-1).LT.0.0)Z(IG,JG)=-Z(IG,JG)
      RETURN
      END
```

```
C
      SUBROUTINE MOMENT (IG,JG,IS,JS,IE,JE,
     &                   Z,M,N,R2,R3,R4,R5,
     &                   V1,V2)
```

Computation of the equation of motion.

```
C
      REAL M,N
      DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG)
      DIMENSION R2(IG,JG),R3(IG,JG),R4(IG,JG),R5(IG,JG)
      DIMENSION V1(JG),V2(JG)
C
      DO 10 J=JS,JE
      DO 10 I=IS,IE
      V1(I)=Z(I+1,J)-Z(I,J)
      V2(I)=N(I,J-1)+N(I,J)+N(I+1,J-1)+N(I+1,J)
      M(I,J)=M(I,J)-R2(I,J)*V1(I)+R3(I,J)*V2(I)
   10 CONTINUE
```

Computation of M.

```
      IF(JS.LE.2)THEN
      DO 15 I=1,IG-1
   15 M(I,1)=M(I,1)-R2(I,1)*(Z(I+1,1)-Z(I,1))
      ENDIF
      IF(JE.GE.JG-1)THEN
      DO 16 I=1,IG-1
   16 M(I,JG)=M(I,JG)-R2(I,JG)*(Z(I+1,JG)-Z(I,JG))
      ENDIF
      IF(IS.LE.2)THEN
      DO 17 J=1,JG
   17 M(1,J)=M(1,J)-R2(1,J)*(Z(2,J)-Z(1,J))
      ENDIF
```

Computation of M along the boundary line.

```
C
      DO 20 J=JS,JE
      DO 20 I=IS,IE
      V1(I)=Z(I,J+1)-Z(I,J)
      V2(I)=M(I-1,J)+M(I,J)+M(I-1,J+1)+M(I,J+1)
      N(I,J)=N(I,J)-R4(I,J)*V1(I)-R5(I,J)*V2(I)
   20 CONTINUE
```

Computation of N.

```
      IF(IS.LE.2)THEN
      DO 25 J=1,JG-1
   25 N(1,J)=N(1,J)-R4(1,J)*(Z(1,J+1)-Z(1,J))
      ENDIF
      IF(IE.GE.IG-1)THEN
      DO 26 J=1,JG-1
   26 N(IG,J)=N(IG,J)-R4(IG,J)*(Z(IG,J+1)-Z(IG,J))
      ENDIF
      IF(JS.LE.2)THEN
      DO 27 I=1,IG
   27 N(I,1)=N(I,1)-R4(I,1)*(Z(I,2)-Z(I,1))
      ENDIF
```

Computation of N along the boundary line.

```
C
      RETURN
      END
C
```

```
      SUBROUTINE OUTPUT (IG,JG,IS,JS,IE,JE,KK,KD,Z,H)
C
      PARAMETER (GX=1.E-4)
      DIMENSION Z(IG,JG),H(IG,JG)
      DIMENSION KW(100),KM(23)
C
      DATA KM/1HJ,1HI,1HH,1HG,1HF,1HE,1HD,1HC,
     &       1HB,1HA,1H.,1H1,1H2,1H3,1H4,1H5,
     &       1H6,1H7,1H8,1H9,1H0,1H*,1H /
      IF(KK.EQ.0)GOTO 120
C     IF(MOD(KK,180).EQ.0.AND.KK.LE.360*4)THEN
C     WRITE(26,102)IS,JS,IE,JE
C 102 FORMAT(4I10)
C     DO 15 J=JS,JE
C     WRITE(26,101)(Z(I,J),I=IS,IE)
C 101 FORMAT(20F6.3)
C  15 CONTINUE
C     ENDIF
 120  IF(MOD(KK,KD).NE.0)RETURN
      WRITE(6,410)KK,(KI,KI=120,160,10)
      DO 30 KY=1,100
      KJ=100-KY+1
      J1=6*(KJ-1)+1
      J2=J1+5
      DO 40 KI=1,80
      I1=6*(KI-1)+1
      I2=I1+5
      L=0
      S=0.
      DO 41 J=J1,J2
      DO 41 I=I1,I2
      IF(H(I,J).GT.0.)THEN
      L=L+1
      S=S+Z(I,J)
      ENDIF
   41 CONTINUE
      IF(L.GE.20)THEN
      S=S/L
      KMM=11+INT(S/0.02)
      IF(KMM.GT.21)KMM=21
      IF(KMM.LT.1)KMM=1
      IF(ABS(S).LT.GX)KMM=23
      ELSE
      KMM=22
      ENDIF
      KW(KI)=KM(KMM)
   40 CONTINUE
      JK=(KY+1)/2
      WRITE(6,400)JK,(KW(I),I=1,80)
   30 CONTINUE
      WRITE(6,100)KK,IS,JS,IE,JE
      RETURN
  100 FORMAT(1H ,"K=",I5,2X,"IS,JS,IE,JE=",4I5)
  400 FORMAT(1H ,I3,2X,80A1)
  410 FORMAT(1H1,/,2X,"MAP AT K=",I5,//,3X,4I20)
      END
C
```

Output of the spatial distribution of water level.

Output of the spatial distribution of water level in the area for computation.

Computation of the mean water level in each area of 2' x 2'.

Output of the mean water level on the display.

```
      SUBROUTINE FILEOT (IG,JG,Z,M,N)---
```

Output of the water level and discharge.

```
C
      REAL M,N
      DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG)
C
      DO 10 J=1,JG
      WRITE(25,100)(Z(I,J),I=1,IG)
      WRITE(25,100)(M(I,J),I=1,IG)
      WRITE(25,100)(N(I,J),I=1,IG)
   10 CONTINUE
  100 FORMAT(10E12.5)
      RETURN
      END
C
      SUBROUTINE MAX (IG,JG,IS,JS,IE,JE,Z,H,ZM,ZN)---
```

Computation of the highest and lowest water levels.

```
C
      DIMENSION Z(IG,JG),H(IG,JG),ZM(IG,JG),ZN(IG,JG)
      DO 10 I=IS,IE
      DO 10 J=JS,JE
      IF(H(I,J).LT.0.0)GOTO 10
      IF(ZM(I,J).LT.Z(I,J))ZM(I,J)=Z(I,J)
      IF(ZN(I,J).GT.Z(I,J))ZN(I,J)=Z(I,J)
   10 CONTINUE
      RETURN
      END
C
      SUBROUTINE POINT (IG,JG,Z,KK,KC)---
```

Output of the time histroy of water level.

```
C
      DIMENSION Z(IG,JG)
      IF(MOD(KK,KC).NE.0)RETURN
      WRITE(24,100)Z(211,405),Z(252,417),
                 &Z(249,417),Z(246,418),
                 &Z(219,415),Z(238,419),
                 &Z(237,397),Z(266,325),
                 &Z(133,377),Z(92,315),
                 &Z(50,292),
                 &Z(480,339),Z(382,89),
                 &Z(480,231),Z(269,161)
  100 FORMAT(15F9.5)
      RETURN
      END
C
```

```
4650      SUBROUTINE PROPA (IG,JG,IS,JS,IE,JE,Z,H,LF,KK)
4660      DIMENSION Z(IG,JG),H(IG,JG),LF(IG,JG)
4670      LK=KK/60+1
4680      GX=0.002
4690      IF(LK.LE.3)GX=0.02
4700      IF(LK.GT.3.AND.LK.LE.6)GX=0.005
4710      DO 10 I=IS,IE
4720      DO 10 J=JS,JE
4730      IF(H(I,J).GT.0.0)THEN
4740      IF(LF(I,J).EQ.0)THEN
4750      IF(ABS(Z(I,J)).GT.GX)LF(I,J)=LK
4760      ENDIF
4770      ELSE
4780      LF(I,J)=-9
4790      ENDIF
4800   10 CONTINUE
4810      RETURN
4820      END
4830C
4840      SUBROUTINE OUTDT (IG,JG,ZM,ZN,LK)
4850      DIMENSION ZM(IG,JG),ZN(IG,JG),LK(IG,JG)
4860      DO 20 I=1,IG
4870      DO 20 J=1,JG
4880      ZN(I,J)=ZM(I,J)-ZN(I,J)
4890      IF(ZN(I,J).GT.100.0)ZN(I,J)=100.0
4900      IF(ZM(I,J).GT.100.0)ZM(I,J)=100.0
4910   20 CONTINUE
4920      DO 10 J=1,JG
4930C     WRITE(31,100)(ZM(I,J),I=1,IG)
4940C     WRITE(32,100)(ZN(I,J),I=1,IG)
4950      WRITE(33,200)(LK(I,J),I=1,IG)
4960   10 CONTINUE
4970C 100 FORMAT(20F6.3)
4980  200 FORMAT(40I3)
4990      RETURN
5000      END
5010C
```

Check of the arrival time.

Travel time at present.

Give the travel time to a point, if the water level at the point exceeds 1cm.

Output of the highest and lowest water level, and the arrival time.

```
5020       SUBROUTINE DEFORM(IG,JG,IS,JS,IE,JE,Z)
5030       REAL L
5040       PARAMETER (DX=15.0E+3,DR=5.0)
5050       PARAMETER (A=3.141592,B=4.848E-06,RR=6.37E+6,E=1.7453E-2)
5060       PARAMETER (HH=1.0E+3,D=1.2,DL=45.0,TH=10.0,RD=-90.0)
5070       PARAMETER (L=100.0E+3,W=50.0E+3)
5080       PARAMETER (YO=10.0,XO=140.0)
5090       PARAMETER (Y0=15.4,X0=147.3)
5100       DIMENSION Z(IG,JG)
5110       XL=A*RR*(X0-XO)*COS(E*YO)/180.0 ; YL=A*RR*(Y0-YO)/180.0
5120       H1=HH/SIN(E*DL)           ; H2=HH/SIN(E*DL)+W
5130       DS=D*COS(E*RD)            ; DD=D*SIN(E*RD)
5140       WRITE(6,*)XL,YL,H1,H2,DS,DD
5150       DO 10 I=1,IE-IS+1
5160       DO 10 J=1,JE-JS+1
5170C      XX=DX*(I-1) ; YY=DX*(J-1)
5180       YY=A*RR*DR*(J-1)/(60.0*180)
5190       XX=A*RR*DR*(I-1)*COS(E*(YO+DR*(J-1)/60.0))/(60.0*180)
5200       X1=(XX-XL)*SIN(E*TH)+(YY-YL)*COS(E*TH)-L/2.0
5210       X2=(XX-XL)*COS(E*TH)-(YY-YL)*SIN(E*TH)+HH/TAN(E*DL)
5220       X3=0.0
5230       CALL USCAL (X1,X2,X3, L/2.,H2,E*DL,F1)
5240       CALL USCAL (X1,X2,X3, L/2.,H1,E*DL,F2)
5250       CALL USCAL (X1,X2,X3,-L/2.,H2,E*DL,F3)
5260       CALL USCAL (X1,X2,X3,-L/2.,H1,E*DL,F4)
5270       CALL UDCAL (X1,X2,X3, L/2.,H2,E*DL,G1)
5280       CALL UDCAL (X1,X2,X3, L/2.,H1,E*DL,G2)
5290       CALL UDCAL (X1,X2,X3,-L/2.,H2,E*DL,G3)
5300       CALL UDCAL (X1,X2,X3,-L/2.,H1,E*DL,G4)
5310       US=(F1-F2-F3+F4)*DS/(12.0*A)
5320       UD=(G1-G2-G3+G4)*DD/(12.0*A)
5330       IF=I+IS-1  ; JF=J+JS-1
5340       Z(IF,JF)=US+UD
5350    10 CONTINUE
5360       RETURN
5370       END

5380C
```

Computation of the initial profile

Computations of parameters. Refer Figs.8 and 9.

Co-ordinates transform between (X,Y) and (X1,Y1).

Vertical displacement due to strike slip

Vertical displacement due to dip slip

```
5390      SUBROUTINE USCAL (X1,X2,X3,C,CC,DP,F)
5400      REAL K
5410C
5420      SN=SIN(DP)   ; CS=COS(DP)
5430      C1=C ; C2=CC*CS ; C3=CC*SN
5440      R=SQRT((X1-C1)**2+(X2-C2)**2+(X3-C3)**2)
5450      Q=SQRT((X1-C1)**2+(X2-C2)**2+(X3+C3)**2)
5460      R2=X2*SN-X3*CS
5470      R3=X2*CS+X3*SN
5480      Q2=X2*SN+X3*CS
5490      Q3=-X2*CS+X3*SN
5500      H=SQRT(Q2**2+(Q3+CC)**2)
5510      K=SQRT((X1-C1)**2+Q2**2)
5520      A1=LOG(R+R3-CC)
5530      A2=LOG(Q+Q3+CC)
5540      A3=LOG(Q+X3+C3)
5550      B1=1+3.0*(TAN(DP))**2
5560      B2=3.0*TAN(DP)/CS
5570      B3=2.0*R2*SN
5580      B4=Q2+X2*SN
5590      B5=2.0*R2**2*CS
5600      B6=R*(R+R3-CC)
5610      B7=4.0*Q2*X3*SN**2
5620      B8=2.0*(Q2+X2*SN)*(X3+Q3*SN)
5630      B9=Q*(Q+Q3+CC)
5640      B10=4.0*Q2*X3*SN
5650      B11=(X3+C3)-Q3*SN
5660      B12=4.0*Q2**2*Q3*X3*CS*SN
5670      B13=2.0*Q+Q3+CC
5680      B14=Q**3*(Q+Q3+CC)**2
5690      F=CS*(A1+B1*A2-B2*A3)
5700     & +B3/R+2*SN*B4/Q-B5/B6+(B7-B8)/B9+B10*B11/Q**3-B12*B13/B14
5710      RETURN
5720      END
5730C
```

Computation of the vertical displacement due to strike slip.

```
      SUBROUTINE UDCAL (X1,X2,X3,C,CC,DP,F)
      REAL K
C
      SN=SIN(DP)   ; CS=COS(DP)
      C1=C ; C2=CC*CS ; C3=CC*SN
      R=SQRT((X1-C1)**2+(X2-C2)**2+(X3-C3)**2)
      Q=SQRT((X1-C1)**2+(X2-C2)**2+(X3+C3)**2)
      R2=X2*SN-X3*CS
      R3=X2*CS+X3*SN
      Q2=X2*SN+X3*CS
      Q3=-X2*CS+X3*SN
      H=SQRT(Q2**2+(Q3+CC)**2)
      K=SQRT((X1-C1)**2+Q2**2)
      A1=LOG(R+X1-C1)
      A2=LOG(Q+X1-C1)
      B1=Q*(Q+X1-C1)
      B2=R*(R+X1-C1)
      B3=Q*(Q+Q3+CC)
      D1=X1-C1   ; D2=X2-C2   ; D3=X3-C3
      D4=X3+C3   ; D5=R3-CC   ; D6=Q3+CC
      T1=ATN(D1*D2,(H+D4)*(Q+H))
      T2=ATN(D1*D5,R2*R)
      T3=ATN(D1*D6,Q2*Q)
      F=SN*(D2*(2*D3/B2+4*D3/B1-4*C3*X3*D4*(2*Q+D1)/(B1**2*Q))
         &  -6*T1+3*T2-6*T3)
      &+CS*(A1-A2-2*(D3**2)/B2
         &  -4*(D4**2-C3*X3)/B1-4*C3*X3*D4**2*(2*Q+X1-C1)/(B1**2*Q))
       &+6*X3*(CS*SN*(2*D6/B1+D1/B3)-Q2*(SN**2-CS**2)/B1)
      RETURN
      END
C
      REAL FUNCTION ATN (AX,AY)
      DATA GX/1.0E-6/
      AAX=ABS(AX) ; AAY=ABS(AY)
      P=AX*AY
      IF(AAX.LE.GX.AND.AAY.LE.GX)GOTO 10
      SR=ATAN2(AAX,AAY)
      ATN=SIGN(SR,P)
      RETURN
   10 WRITE(6,100)AX,AY
  100 FORMAT(1H ,"ATAN --    AX=",E15.7,2X,"AY=",E15.7)
      ATN=0.2
      RETURN
      END
```

Computation of the vertical displacement due to dip slip.

□

IUGG/IOC TIME PROJECT

PART 3. PROGRAMME LISTS FOR NEAR-FIELD TSUNAMI

TUNAMI-N1: 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

CONTENTS

```
*#RUN : IAP OPT=2 F=/SNRK/MM2.7(02)
C
C   *************** TUNAMI-N1 ****************
C     Tohoku University's Numerical-Analysis Model
C        for Investigation of tsunami
C         Near-field Tsunami version
C
C       1968 TOKACHI-OKI EARTHQUAKE TSUNAMI
C         USING BY LINEAR LONG WAVE THEORY
C
C               CODED  BY
C
C         F.IMAMURA, TOHOKU UNIV., JAPAN
C
C                         1991.6.25
C
C     Z; WATER SURFACE LEVEL   M,N; WATER DISCHARGE
C     H; STILL WATER DEPTH      ZD; TOTAL WATER DEPTH
C   IF,JF; DIMENSION          DX,DT; SPATIAL GRID & TIME STEP SIZE
C     G; GRAVITATIONAL ACCES.  KL; TOTAL TIME STEP
C   IP,JP; POSITION OF OUTPUT POINT
C   I0,J0; ORIGIN OF EPICENTER
C
      REAL M,N
      PARAMETER(IF=260,JF=180,I0=111,J0=188)
      PARAMETER(DX=2700.0,DT=5.0,G=9.8,KD=1,KL=360)
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
      DIMENSION ZM(IF,JF),ZD(IF,JF),IH(IF,JF)
C      DIMENSION IP(16),JP(16),PZ(16)
C      DATA IP/62,76,86,96,104,114,124,132,142,152,162,172,182
C     &     ,192,202,212/
C      DATA JP/60,60,62,66,72,76,80,84,88,86,80,78,78,78,80,78/
C
      R=DT/DX ; KK=0
C
C  ********* INITIAL CONDITION *************
C
      CALL DEPTH(IF,JF,H,IH)
      CALL INTL(IF,JF,Z,M,N)
      CALL DEFORM(IF,JF,Z,I0,J0,DX)
C      CALL OUT6 (IF,JF,Z,H,KK)
      CALL PTIME(CT1)
C
C  ********* MAIN CALCULATION *************
C
      DO 10 K=1,KL
       KK=K
       IF(MOD(KK,10).EQ.0)WRITE(6,*)KK
       CALL MASS(IF,JF,Z,M,N,H,R,ZD)
       CALL OPEN(IF,JF,Z,M,N,H)
       CALL MOMENT(IF,JF,Z,M,N,H,R,G)
       CALL ZMAX(IF,JF,Z,ZM)
C       CALL OUT3 (IF,JF,16,Z,IP,JP,PZ)
       IF(MOD(K,50).NE.0)GOTO 20
       CALL OUT6(IF,JF,Z,H,KK)
C       CALL OUT1 (IF,JF,Z)
  20   CALL CHANGE(IF,JF,Z,M,N)
  10 CONTINUE
C
C      CALL OUT66 (IF,JF,ZM,H)
C      CALL OUT11 (IF,JF,ZM)
      CALL PTIME(CT2)
      CT=CT2-CT1
      WRITE(6,100)CT
 100 FORMAT(1H ,3X,"CPU TIME =",E12.4)
      STOP
      END
C
C ******** READ DEPTH DATA *********************
C
      SUBROUTINE DEPTH(IF,JF,H,IH)
C
```

```
      DIMENSION H(IF,JF),IH(IF,JF)
      L=JF/10+1
      IF(MOD(JF,10).EQ.0)L=L-1
      DO 10 LL=1,L
       J1=10*(LL-1)+1
       J2=J1+9
       IF(J2.GT.JF)J2=JF
       DO 10 I=1,IF
        READ(2,100)(IH(I,J),J=J1,J2)
 100     FORMAT(5X,10I5)
        DO 10 J=J1,J2
         H(I,J)=FLOAT(IH(I,J))
         IF(IH(I,J).GT.0.0.AND.IH(I,J).LT.500)H(I,J)=500.0
  10 CONTINUE
      RETURN
      END
C
C ********* INITIAL CONDITION **************
C
      SUBROUTINE INTL(IF,JF,Z,M,N)
C
      REAL M,N
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
      DO 10 K=1,2
       DO 10 J=1,JF
        DO 10 I=1,IF
         Z(I,J,K)=0.0
         M(I,J,K)=0.0
         N(I,J,K)=0.0
  10 CONTINUE
      RETURN
      END
C
C ********** MASS CONSERVATION **************
C
      SUBROUTINE MASS(IF,JF,Z,M,N,H,R,ZD)
C
      REAL M,N
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
      DIMENSION ZD(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))
        ZZ=ZZ+ZD(I,J)
        IF(ABS(ZZ).LT.GX)ZZ=0.0
        Z(I,J,2)=ZZ
  10 CONTINUE
      RETURN
      END
C
C *********** MOMENTUM CONSERVATION *****************
C
      SUBROUTINE MOMENT(IF,JF,Z,M,N,H,R,G)
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/
      DO 10 J=1,JF
       DO 10 I=1,IF
        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
        HM=0.5*(H(I,J)+H(I+1,J))
        XM=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
        IF(H(I,J+1).LT.0.0)GO TO 10
        HN=0.5*(H(I,J)+H(I,J+1))
        XN=N(I,J,1)-G*R*HN*(Z(I,J+1,2)-Z(I,J,2))
```

```
1450       IF(ABS(XN).LT.GX)XN=0.0
1460       N(I,J,2)=XN
1470  10 CONTINUE
1480     RETURN
1490     END
1500C
1510C ****** OUTPUT OF COMPUTED RESULTS ON THE DISPLAY *******
1520C
1530     SUBROUTINE OUT6(IF,JF,Z,H,K)
1540C
1550     DIMENSION KM(46),LW(140),Z(IF,JF,2)
1560     DIMENSION H(IF,JF)
1570     DATA KM/1H*,1HW,1HV,1HU,1HT,1HS,1HR,1HP,1HO,1HN,1HM,1HL,1HK,1HJ,
1580    &  1HI,1HH,1HG,1HF,1HE,1HD,1HC,1HB,1HA,1H.,1H1,1H2,1H3,1H4,1H5,1H6,
1590    &  1H7,1H8,1H9,1H0,1H.,1H-,1H+,1H/,1H#,1H$,1H%,1H&,1H=,1H?,1H(,1H)/
1600     WRITE(6,100)K
1610 100 FORMAT(1H ,2X,"K=",I4,/)
1620     WRITE(6,200)(J,J=10,70,10)
1630 200 FORMAT(1H ,4X,7I10)
1640     DO 10 I=1,130
1650       DO 30 J=1,90
1660         II=2*(I-1)+1
1670         JJ=2*(J-1)+1
1680         ZZZ=0.0
1690         DO 999 IX=II,II+1
1700           DO 999 JX=JJ,JJ+1
1710             ZZZ=ZZZ+0.25*Z(IX,JX,2)
1720             IF(H(IX,JX).LT.0.0)GOTO 998
1730 999       CONTINUE
1740         KKM=24
1750         IF(ZZZ.GT.0.0)KKM=IFIX((ZZZ+0.05)/0.1)+24
1760         IF(ZZZ.LT.0.0)KKM=IFIX((ZZZ-0.05)/0.1)+24
1770         IF(KKM.LT.2)KKM=2
1780         IF(KKM.GT.46)KKM=46
1790         LW(J)=KM(KKM)
1800         GOTO 30
1810 998     LW(J)=KM(1)
1820  30   CONTINUE
1830       WRITE(6,110)I,(LW(J),J=1,70)
1840  110  FORMAT(1H ,I3,1X,90A1)
1850  10 CONTINUE
1860     RETURN
1870     END
1880C
1890C  *********** OUTPUT OF DATA *************
1900C
1910     SUBROUTINE OUT1(IF,JF,Z)
1920     DIMENSION Z(IF,JF,2)
1930     DO 10 I=1,IF
1940       WRITE(4)(Z(I,J,2),J=1,JF)
1950  10 CONTINUE
1960     RETURN
1970     END
1980C
1990     SUBROUTINE OUT3(IF,JF,IM,Z,IP,JP,PZ)
2000     DIMENSION Z(IF,JF,2),IP(IM),JP(IM),PZ(IM)
2010     DO 10 I=1,IM
2020       II=IP(I)
2030       JJ=JP(I)
2040       PZ(I)=Z(II,JJ,2)
2050  10 CONTINUE
2060     WRITE(3)(PZ(I),I=1,IM)
2070     RETURN
2080     END
2090C
2100C *********** OPEN BOUNDARY CONDITION *************
2110C
2120     SUBROUTINE OPEN(IF,JF,Z,M,N,H)
2130C
2140     REAL M,N
2150     DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
2160     DO 10 KK=1,2
```

```
        J=1
        IF(KK.EQ.2)J=JF
        DO 10 I=2,IF
         IF(H(I,J).LT.0.0)GO TO 10
         CC=SQRT(9.8*H(I,J))
         UU=0.5*ABS(M(I,J,2)+M(I-1,J,2))
         IF(J.EQ.1)UU=SQRT(UU**2+N(I,J,2)**2)
         IF(J.EQ.JF)UU=SQRT(UU**2+N(I,J-1,2)**2)
         ZZ=UU/CC
         IF(J.EQ.1.AND.N(I,J,2).GT.0.0)ZZ=-ZZ
         IF(J.EQ.JF.AND.N(I,J-1,2).LT.0.0)ZZ=-ZZ
         Z(I,J,2)=ZZ
   10 CONTINUE
       DO 20 KK=1,2
        I=1
        IF(KK.EQ.2)I=IF
        DO 20 J=2,JF
         IF(H(I,J).LT.0.0)GO TO 20
         CC=SQRT(9.8*H(I,J))
         UU=0.5*ABS(N(I,J,2)+N(I,J-1,2))
         IF(I.EQ.1)UU=SQRT(UU**2+M(I,J,2)**2)
         IF(I.EQ.IF)UU=SQRT(UU**2+M(I-1,J,2)**2)
         ZZ=UU/CC
         IF(I.EQ.1.AND.M(I,J,2).GT.0.0)ZZ=-ZZ
         IF(I.EQ.IF.AND.M(I-1,J,2).LT.0.0)ZZ=-ZZ
         Z(I,J,2)=ZZ
   20 CONTINUE
      RETURN
      END
C
C *** EXCHANGE FOR LAST STEP DATA TO NEXT STEP DATA *****
C
      SUBROUTINE CHANGE(IF,JF,Z,M,N)
C
      REAL M,N
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
      DO 10 J=1,JF
       DO 10 I=1,IF
        Z(I,J,1)=Z(I,J,2)
        M(I,J,1)=M(I,J,2)
        N(I,J,1)=N(I,J,2)
   10 CONTINUE
      RETURN
      END
C
C
C *** CALCULATOIN FOR INITIAL TSUNAMI SOURCE ******
C       (= SEA BOTTOM DEFORMATION)
C
C     RR; RADIUS OF EARTH
C     L ; FAULT LENGTH,    W ; FAULT WIDTH
C     TH; DIP DIRECTION,   DL; DIP ANGLE
C     RD; SLIP ANGLE       D ; DISLOCATION
C     HH; DEPTH
C
      SUBROUTINE DEFORM(IF,JF,Z,I0,J0,DX)
      REAL L
      PARAMETER(A=3.141592,B=4.848E-06,RR=6.37E+6,E=1.7453E-2)
      PARAMETER(L=188.2E+3,W=92.0E+3,TH=145.0,DL=20.0)
      PARAMETER(RD=148.0,D=3.2,HH=1.0E+3)
      DIMENSION Z(IF,JF,2)
      XL=DX*(I0-1) ; YL=DX*(J0-1) ; H1=HH/SIN(E*DL)
      H2=HH/SIN(E*DL)+W ; DS=D*COS(E*RD) ; DD=D*SIN(E*RD)
      DO 10 I=1,IF
       DO 10 J=1,JF
        XX=DX*(I-1) ; YY=DX*(J-1)
        X1=(XX-XL)*SIN(E*TH)+(YY-YL)*COS(E*TH)-L/2.0
        X2=(XX-XL)*COS(E*TH)-(YY-YL)*SIN(E*TH)+HH/TAN(E*DL)
        X3=0.0
        CALL USCAL(X1,X2,X3,L/2.,H2,E*DL,F1)
        CALL USCAL(X1,X2,X3,L/2.,H1,E*DL,F2)
        CALL USCAL(X1,X2,X3,-L/2.,H2,E*DL,F3)
```

```
2890       CALL USCAL(X1,X2,X3,-L/2.,H1,E*DL,F4)
2900       CALL UDCAL(X1,X2,X3,L/2.,H2,E*DL,G1)
2910       CALL UDCAL(X1,X2,X3,L/2.,H1,E*DL,G2)
2920       CALL UDCAL(X1,X2,X3,-L/2.,H2,E*DL,G3)
2930       CALL UDCAL(X1,X2,X3,-L/2.,H1,E*DL,G4)
2940       US=(F1-F2-F3+F4)*DS/(12.0*A)
2950       UD=(G1-G2-G3+G4)*DD/(12.0*A)
2960       Z(I,J,1)=US+UD
2970       Z(I,J,2)=US+UD
2980  10 CONTINUE
2990     RETURN
3000     END
3010C
3020     SUBROUTINE USCAL(X1,X2,X3,C,CC,DP,F)
3030     REAL K
3040C
3050     SN=SIN(DP) ; CS=COS(DP) ; C1=C ; C2=CC*CS ; C3=CC*SN
3060     R=SQRT((X1-C1)**2+(X2-C2)**2+(X3-C3)**2)
3070     Q=SQRT((X1-C1)**2+(X2-C2)**2+(X3+C3)**2)
3080     R2=X2*SN-X3*CS
3090     R3=X2*CS+X3*SN
3100     Q2=X2*SN+X3*CS
3110     Q3=-X2*CS+X3*SN
3120     H=SQRT(Q2**2+(Q3+CC)**2)
3130     K=SQRT((X1-C1)**2+Q2**2)
3140     A1=LOG(R+R3-CC)
3150     A2=LOG(Q+Q3+CC)
3160     A3=LOG(Q+X3+C3)
3170     B1=1+3.0*(TAN(DP))**2
3180     B2=3.0*TAN(DP)/CS
3190     B3=2.0*R2*SN
3200     B4=Q2+X2*SN
3210     B5=2.0*R2**2*CS
3220     B6=R*(R+R3-CC)
3230     B7=4.0*Q2*X3*SN**2
3240     B8=2.0*(Q2+X2*SN)*(X3+Q3*SN)
3250     B9=Q*(Q+Q3+CC)
3260     B10=4.0*Q2*X3*SN
3270     B11=(X3+C3)-Q3*SN
3280     B12=4.0*Q2**2*Q3*X3*CS*SN
3290     B13=2.0*Q+Q3+CC
3300     B14=Q**3*(Q+Q3+CC)**2
3310     F=CS*(A1+B1*A2-B2*A3)+B3/R+2*SN*B4/Q-B5/B6+(B7-B8)/B9+B10*B11/
3320    & Q**3-B12*B13/B14
3330     RETURN
3340     END
3350C
3360     SUBROUTINE UDCAL(X1,X2,X3,C,CC,DP,F)
3370     REAL K
3380C
3390     SN=SIN(DP) ; CS=COS(DP) ; C1=C ; C2=CC*CS ; C3=CC*SN
3400     R=SQRT((X1-C1)**2+(X2-C2)**2+(X3-C3)**2)
3410     Q=SQRT((X1-C1)**2+(X2-C2)**2+(X3+C3)**2)
3420     R2=X2*SN-X3*CS
3430     R3=X2*CS+X3*SN
3440     Q2=X2*SN+X3*CS
3450     Q3=-X2*CS+X3*SN
3460     H=SQRT(Q2**2+(Q3+CC)**2)
3470     K=SQRT((X1-C1)**2+Q2**2)
3480     A1=LOG(R+X1-C1)
3490     A2=LOG(Q+X1-C1)
3500     B1=Q*(Q+X1-C1)
3510     B2=R*(R+X1-C1)
3520     B3=Q*(Q+Q3+CC)
3530     D1=X1-C1 ; D2=X2-C2 ; D3=X3-C3 ; D4=X3+C3 ; D5=R3-CC ; D6=Q3+CC
3540     T1=ATN(D1*D2,(H+D4)*(Q+H))
3550     T2=ATN(D1*D5,R2*R)
3560     T3=ATN(D1*D6,Q2*Q)
3570     F=SN*(D2*(2*D3/B2+4*D3/B1-4*C3*X3*D4*(2*Q+D1)/(B1**2*Q))
3580    & -6*T1+3*T2-6*T3)+CS*(A1-A2-2*(D3**2)/B2-4*(D4**2-C3*X3)/
3590    & B1-4*C3*X3*D4**2*(2*Q+X1-C1)/(B1**2*Q))+6*X3*(CS*SN*(2*D6/B1+D1/
3600    & B3)-Q2*(SN**2-CS**2)/B1)
```

```
      RETURN
      END
C
      REAL FUNCTION ATN(AX,AY)
      DATA GX/1.0E-6/
      AAX=ABS(AX) ; AAY=ABS(AY)
      P=AX*AY
      IF(AAX.LE.GX.AND.AAY.LE.GX)GOTO 10
      SR=ATAN2(AAX,AAY)
      ATN=SIGN(SR,P)
      RETURN
   10 WRITE(6,100)AX,AY
  100 FORMAT(1H ,"ATAN --   AX=",E15.7,2X,"AY=",E15.7)
      ATN=0.2
      END
C
      SUBROUTINE ZMAX(IF,JF,Z,ZM)
      DIMENSION Z(IF,JF),ZM(IF,JF)
C
      DO 10 I=1,IF
        DO 10 J=1,JF
          IF(ZM(I,J).LT.Z(I,J))ZM(I,J)=Z(I,J)
   10 CONTINUE
      RETURN
      END
```

```
*#RUN : IAP OPT=2 F=/SNRK/GD-MK(10) F=/SNRK/GD-BT1(11);
*#             F=/SNRK/TBN-MYK(12)
C
C  ************** TUNAMI-N2 ****************
C     Tohoku University's Numerical-Analysis Model
C        for Investigation of tsunami
C         Near-field Tsunami version
C
C        FOR 1896 MEIJI SANTRIKU TSUNAMI
C          AT MIYAKO BAY (DX=50M)
C          WITH SHALLOW WATER THEORY
C        INCLUDING EFFECTS OF RUNUP AND STRUCTURES
C
C                        1991.1.25
C               BY
C
C        F.IMAMURA, TOHOKU UNIV., JAPAN
C
C     Z; WATER SURFACE LEVEL   M,N; WATER DISCHARGE
C   HM,HN; STILL WATER DEPTH AT POINT OF WATER DISCHARGE
C   DM,DN; TOTAL WATER DEPTH AT POINT OF WATER DISCHARGE
C     HZ; STILL WATER DEPTH     ZD; TOTAL WATER DEPTH
C   IF,JF; DIMENSION PARAME   DX,DT; SPATIAL GRID & TIME STEP SIZE
C     G; GRAVITATIONAL ACCES.  KL; TOTAL TIME STEP
C     IR; MAP OF WAVE-BRAKER    BT; HEIGHT OF WAVE-BRAKER
C   IP,JP; POSITION OF OUTPUT POINT
C     ZM; MAXIMUM WATER LEVEL
C
      REAL M,N
      PARAMETER(IF=170,JF=150)
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),HZ(IF,JF)
      DIMENSION DZ(IF,JF,2),DM(IF,JF,2),DN(IF,JF,2)
      DIMENSION HM(IF,JF),HN(IF,JF),B(1800,14),ZM(IF,JF)
      DIMENSION IP(13),JP(13),PZ(13)
      DIMENSION IR(IF,JF),BT(10)
      DATA IP/14,27,50,62,78,114,123,144,144,132,116,102,85/
      DATA JP/73,80,79,78,79,52,40,48,72,71,78,85,103/
      DATA BT/5.26,6.70,8.0,5.5,6.0,10.0,10.0,10.0,10.0,10.0/
      G=9.8 ; DX=50.0 ; DT=1.0 ; R=DT/DX
      KL=5400
C
C  ********* INITIAL CONDITION *************
C
      CALL READ(10,11,IF,JF,HZ,IR)
      CALL DATA(12,B)
      CALL HMN(IF,JF,HZ,HM,HN,IR,BT)
      CALL INTL(IF,JF,Z,M,N,DZ,HZ)
      CALL PTIME(CT1)
C
C  ********* MAIN CALCULATION *************
C
      DO 10 K=1,KL
        KK=K
        CALL NLMASS(IF,JF,Z,M,N,DZ,HZ,R)
        CALL BNC(IF,JF,Z,B,2,101,150,KK,DT)
        CALL NLMMT(IF,JF,Z,M,N,DZ,DM,DN,HZ,HM,HN,R,DT,0.025,IR)
        CALL MAX(IF,JF,Z,ZM)
        IF(MOD(K,30).NE.0)GOTO 20
        CALL POINT(17,IF,JF,13,Z,IP,JP,PZ)
        CALL OUT1(15,IF,JF,Z,M,N,DM,DN,HZ)
        CALL OUT6(IF,JF,Z,HZ,KK)
   20   CALL CHANGE(IF,JF,Z,M,N,DZ)
   10 CONTINUE
      CALL OUT3(16,IF,JF,ZM)
C     CALL OUT4 (18,IF,JF,Z,M,N,HZ,ZM)
      CALL PTIME(CT2)
      CT=CT2-CT1
      WRITE(6,180)CT
  180 FORMAT(1H ,3X,"CPU TIME=",E12.4)
      STOP
      END
C
```

```
C ******** READ DEPTH DATA *********************
C
      SUBROUTINE READ(N1,N2,IF,JF,HZ,IR)
      DIMENSION HZ(IF,JF),IR(IF,JF)
      L=IF/10
      IF(MOD(IF,10).NE.0)L=L+1
      DO 10 LL=1,L
       I1=10*(LL-1)+1
       I2=I1+9
       IF(I2.GE.IF)I2=IF
       DO 10 J=1,JF
         READ(N1,100)(HZ(I,JF-J+1),I=I1,I2)
         READ(N2,200)(IR(I,JF-J+1),I=I1,I2)
 100       FORMAT(5X,10F5.1)
 200       FORMAT(4X,10I5)
  10 CONTINUE
      RETURN
      END
C
C ********** INITIAL CONDITION **************
C
      SUBROUTINE INTL(IF,JF,Z,M,N,D,H)
      REAL M,N
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
      DIMENSION H(IF,JF),D(IF,JF,2)
      DO 10 K=1,2
       DO 10 J=1,JF
        DO 10 I=1,IF
         Z(I,J,K)=0.0
         M(I,J,K)=0.0
         N(I,J,K)=0.0
         D(I,J,K)=H(I,J)
         IF(H(I,J).GT.0.0) GOTO 10
         D(I,J,K)=0.0
         Z(I,J,K)=-H(I,J)
  10 CONTINUE
      RETURN
      END
C
C ****** CAL. OF WATER DEPTH AT POINT OF DISCHARGE ********
C
      SUBROUTINE HMN(IF,JF,HZ,HM,HN,IR,BT)
      DIMENSION HZ(IF,JF),HM(IF,JF),HN(IF,JF),IR(IF,JF),BT(10)
      DO 10 J=1,JF
       DO 10 I=1,IF
        IF(I.EQ.IF)GOTO 11
        HM(I,J)=0.5*(HZ(I,J)+HZ(I+1,J))
        GOTO 12
 11      HM(I,J)=HZ(I,J)
 12      IF(J.EQ.JF)GOTO 13
        HN(I,J)=0.5*(HZ(I,J)+HZ(I,J+1))
        GOTO 10
 13      HN(I,J)=HZ(I,J)
  10 CONTINUE
      DO 20 I=1,IF
       DO 20 J=1,JF
        IRR=MOD(IR(I,J),10)
        IRM=IR(I,J)/10
        IF(IRM.EQ.0) GOTO 20
        IF(IRM.EQ.2) GOTO 22
        HM(I,J)=-BT(IRR)
        IF(IRM.EQ.1) GOTO 23
 22      HN(I,J)=-BT(IRR)
 23      IF(IRM.EQ.2) GOTO 24
        IF(HM(I,J).GT.HZ(I,J))HM(I,J)=HZ(I,J)
        IF(HM(I,J).GT.HZ(I+1,J))HM(I,J)=HZ(I+1,J)
        IF(IRM.EQ.1) GOTO 20
 24      IF(HN(I,J).GT.HZ(I,J))HN(I,J)=HZ(I,J)
        IF(HN(I,J).GT.HZ(I,J+1))HN(I,J)=HZ(I,J+1)
  20 CONTINUE
      RETURN
      END
```

```
C
C ******** MASS CONSERVATION ***********
C
      SUBROUTINE NLMASS(IF,JF,Z,M,N,DZ,HZ,R)
      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=3,IF
        IF(HZ(I,J).LT.-9.9)GOTO 11
        XM=0.0
        XN=0.0
        IF(I.NE.1)XM=M(I-1,J,1)
        IF(J.NE.1)XN=N(I,J-1,1)
        ZZZ=Z(I,J,1)-R*(M(I,J,1)-XM+N(I,J,1)-XN)
        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
C
C *********** MOMENTUM CONSERVATION *****************
C
      SUBROUTINE NLMMT(IF,JF,Z,M,N,DZ,DM,DN,HZ,HM,HN,R,DT,FM,IR)
      REAL M,N
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
      DIMENSION DZ(IF,JF,2),DM(IF,JF,2),DN(IF,JF,2)
      DIMENSION HZ(IF,JF),HM(IF,JF),HN(IF,JF),IR(IF,JF)
      DATA GG,GX/9.8,1.0E-5/
C
C  ------ CAL. OF TOTAL DEPTH AT POINT OF DISCHARGE -------
C
      DO 10 I=2,IF
       DO 10 J=2,JF
        IRR=IR(I,J)/10
        IF(IRR.EQ.1.OR.IRR.EQ.3)GO TO 11
        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))
        GO TO 12
   11    IF(DZ(I,J,1))13,13,14
   13    IF(DZ(I+1,J,1))15,15,16
   14    IF(DZ(I+1,J,1))17,17,18
   16    IF(Z(I+1,J,1)+HM(I,J))15,15,19
   17    IF(Z(I,J,1)+HM(I,J))15,15,20
   18    IF(Z(I,J,1)+HM(I,J))21,21,22
   21    IF(Z(I+1,J,1)+HM(I,J))15,15,19
   22    IF(Z(I+1,J,1)+HM(I,J))20,20,23
   15    DM2=0.0
        DM1=0.0
        GO TO 12
   19    DM2=Z(I+1,J,1)+HM(I,J)
        DM1=DM2
        GO TO 12
   20    DM2=Z(I,J,1)+HM(I,J)
        DM1=DM2
        GO TO 12
   23    DM2=0.5*(Z(I,J,1)+Z(I+1,J,1))+HM(I,J)
        DM1=DM2
   12    IF(IRR.GE.2)GO TO 112
        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))
        GO TO 24
  112     IF(DZ(I,J,1))113,113,114
  113     IF(DZ(I,J+1,1))115,115,116
```

```
 114  IF(DZ(I,J+1,1))117,117,118
 116  IF(Z(I,J+1,1)+HN(I,J))115,115,119
 117  IF(Z(I,J,1)+HN(I,J))115,115,120
 118  IF(Z(I,J,1)+HN(I,J))121,121,122
 121  IF(Z(I,J+1,1)+HN(I,J))115,115,119
 122  IF(Z(I,J+1,1)+HN(I,J))120,120,123
 115  DN2=0.0
      DN1=0.0
      GO TO 24
 119  DN2=Z(I,J+1,1)+HN(I,J)
      DN1=DN2
      GO TO 24
 120  DN2=Z(I,J,1)+HN(I,J)
      DN1=DN2
      GO TO 24
 123  DN2=0.5*(Z(I,J,1)+DZ(I,J+1,1))+HN(I,J)
      DN1=DN2
  24  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
C  ------- CAL. OF LINEAR TERMS (X-DIRECTION) -------
C
      FN=0.5*DT*GG*FM**2
      DO 200 I=2,IF
       DO 200 J=2,JF
        IF(HZ(I,J).LT.-8.1)GO TO 200
        IF(I.EQ.IF)GO TO 30
        IRR=IR(I,J)/10
        IF(IRR.EQ.1.OR.IRR.EQ.3)GO TO 60
        IF(HM(I,J).LT.-8.1)GO TO 30
        IF(DZ(I,J,2))31,31,32
  31    IF(DZ(I+1,J,2))30,30,34
  32    IF(DZ(I+1,J,2))35,35,36
  34    IF(Z(I+1,J,2)+HZ(I,J))30,30,37
  35    IF(Z(I,J,2)+HZ(I+1,J))30,30,38
  36    DD=DM(I,J,2)
        GO TO 39
  37    DD=Z(I+1,J,2)+HZ(I,J)
        GO TO 39
  38    DD=Z(I,J,2)+HZ(I+1,J)
  39    XNN=0.25*(N(I,J,1)+N(I+1,J,1)+N(I,J-1,1)+N(I+1,J-1,1))
        DF=DD
        IF(DF.LT.1.0E-2)DF=1.0E-2
        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
C ----- CAL. OF NON-LINEAR TERMS (CONVECTION TERMS) ------
C
        IF(I.LE.6.OR.J.LE.6)GO TO 40
        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 40
        IF(DZ(I+2,J,2).LT.GX)GO TO 40
        IF(DZ(I+1,J,2).LT.GX)GO TO 40
        IF(IR(I+1,J)/10.EQ.1.OR.IR(I+1,J)/10.EQ.3)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 40
        IF(DZ(I-1,J,2).LT.GX)GO TO 40
        IF(DZ(I,J,2).LT.GX)GO TO 40
        IF(IR(I-1,J)/10.EQ.1.OR.IR(I-1,J)/10.EQ.3)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))
```

```
2890       GO TO 43
2900 1002    XM=XM-R*(M(I,J,1)**2/DM(I,J,1))
2910  43    IF(XNN)44,44,45
2920  44    XNE=0.25*(N(I,J+1,1)+N(I+1,J+1,1)+N(I,J,1)+N(I+1,J,1))
2930       IF(DM(I,J+1,1).LT.GX)GO TO 40
2940       IF(DZ(I,J+1,2).LT.GX)GO TO 40
2950       IF(DZ(I,J+2,2).LT.GX)GO TO 40
2960       IF(DZ(I+1,J+1,2).LT.GX)GO TO 40
2970       IF(DZ(I+1,J+2,2).LT.GX)GO TO 40
2980       IF(IR(I,J)/10.GE.2.OR.IR(I+1,J)/10.GE.2)GO TO 40
2990       XM=XM-R*(M(I,J+1,1)*XNE/DM(I,J+1,1)-M(I,J,1)*XNN/DM(I,J,1))
3000       GO TO 40
3010  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))
3020       IF(DM(I,J-1,1).LT.GX)GO TO 40
3030       IF(DZ(I,J-2,2).LT.GX)GO TO 40
3040       IF(DZ(I,J-1,2).LT.GX)GO TO 40
3050       IF(DZ(I+1,J-1,2).LT.GX)GO TO 40
3060       IF(DZ(I+1,J-2,2).LT.GX)GO TO 40
3070       IF(IR(I,J-1)/10.GE.2.OR.IR(I+1,J-1)/10.GE.2)GO TO 40
3080       XM=XM-R*(M(I,J,1)*XNN/DM(I,J,1)-M(I,J-1,1)*XNE/DM(I,J-1,1))
3090  40    XM=XM/(1.0+FF)
3100       IF(ABS(XM).LT.1.0E-10)XM=0.0
3110C
3120C  ------ LIMITING OF DISCHARGE -------
3130C
3140       IF(XM.GT.7.0*DD)XM=7.0*DD
3150       IF(XM.LT.-7.0*DD)XM=-7.0*DD
3160       M(I,J,2)=XM
3170       GO TO 100
3180  30    M(I,J,2)=0.0
3190       GO TO 100
3200C
3210C  ------ CAL. OF DISCHARGE OF OVERFLOW ------
3220C
3230  60    Z1=Z(I,J,2)+HM(I,J)
3240       Z2=Z(I+1,J,2)+HM(I,J)
3250       ZZZ=Z1
3260       ZX=Z2
3270       IF(Z1.GT.Z2)GO TO 61
3280       ZZZ=Z2
3290       ZX=Z1
3300  61    IF(ZZZ.LT.GX)GO TO 30
3310       IF(ZZZ*0.66667.LT.ZX)GO TO 62
3320       XM=1.55*ZZZ**1.5
3330       GO TO 63
3340  62    XM=4.029*ZX*SQRT(ZZZ-ZX)
3350  63    IF(Z2.GT.Z1)XM=-XM
3360       M(I,J,2)=XM
3370 100    CONTINUE
3380C
3390C  ------- CAL. OF LINEAR TERMS (Y-DIRECTION) -------
3400C
3410       IF(J.EQ.JF)GO TO 130
3420       IRR=IR(I,J)/10
3430       IF(IRR.GE.2)GO TO 160
3440       IF(HN(I,J).LT.-8.1)GO TO 130
3450       IF(DZ(I,J,2))131,131,132
3460 131    IF(DZ(I,J+1,2))130,130,134
3470 132    IF(DZ(I,J+1,2))135,135,136
3480 134    IF(Z(I,J+1,2)+HZ(I,J))130,130,137
3490 135    IF(Z(I,J,2)+HZ(I,J+1))130,130,138
3500 136    DD=DN(I,J,2)
3510       GO TO 139
3520 137    DD=Z(I,J+1,2)+HZ(I,J)
3530       GO TO 139
3540 138    DD=Z(I,J,2)+HZ(I,J+1)
3550 139    XMM=0.25*(M(I,J,1)+M(I,J+1,1)+M(I-1,J,1)+M(I-1,J+1,1))
3560       DF=DD
3570       IF(DF.LT.1.0E-2)DF=1.0E-2
3580       FF=FN*SQRT(N(I,J,1)**2+XMM**2)/DF**(7.0/3.0)
3590       IF(DD.LT.GX)GO TO 130
3600       XN=(1.0-FF)*N(I,J,1)-GG*R*DD*(Z(I,J+1,2)-Z(I,J,2))
```

```
3610C
3620C ----- CAL. OF NON-LINEAR TERMS (CONVECTION TERMS) ------
3630C
3640       IF(I.LE.6.OR.J.LE.6)GO TO 140
3650       IF(DN(I,J,1).LT.GX)GO TO 140
3660       IF(N(I,J,1))141,141,142
3670 141    IF(DN(I,J+1,1).LT.GX)GO TO 140
3680       IF(DZ(I,J+2,2).LT.GX)GO TO 140
3690       IF(DZ(I,J+1,2).LT.GX)GO TO 140
3700       IF(IR(I,J+1)/10.GE.2)GO TO 1003
3710       XN=XN-R*(N(I,J+1,1)**2/DN(I,J+1,1)-N(I,J,1)**2/DN(I,J,1))
3720       GO TO 143
3730 1003    XN=XN-R*(-N(I,J,1)**2/DN(I,J,1))
3740       GO TO 143
3750 142    IF(DN(I,J-1,1).LT.GX)GO TO 140
3760       IF(DZ(I,J-1,2).LT.GX)GO TO 140
3770       IF(DZ(I,J,2).LT.GX)GO TO 140
3780       IF(IR(I,J-1)/10.GE.2)GO TO 1004
3790       XN=XN-R*(N(I,J,1)**2/DN(I,J,1)-N(I,J-1,1)**2/DN(I,J-1,1))
3800       GO TO 143
3810 1004    XN=XN-R*(N(I,J,1)**2/DN(I,J,1))
3820 143    IF(XMM)144,144,145
3830 144    XME=0.25*(M(I+1,J,1)+M(I+1,J+1,1)+M(I,J,1)+M(I,J+1,1))
3840       IF(DN(I+1,J,1).LT.GX)GO TO 140
3850       IF(DZ(I+1,J,2).LT.GX)GO TO 140
3860       IF(DZ(I+2,J,2).LT.GX)GO TO 140
3870       IF(DZ(I+1,J+1,2).LT.GX)GO TO 140
3880       IF(DZ(I+2,J+1,2).LT.GX)GO TO 140
3890       IF(IR(I,J)/10.EQ.1)GO TO 140
3900       IF(IR(I,J+1)/10.EQ.1.OR.IR(I,J+1)/10.EQ.3)GO TO 140
3910       XN=XN-R*(N(I+1,J,1)*XME/DN(I+1,J,1)-N(I,J,1)*XMM/DN(I,J,1))
3920       GO TO 140
3930 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))
3940       IF(DN(I-1,J,1).LT.GX)GO TO 140
3950       IF(DZ(I-2,J,2).LT.GX)GO TO 140
3960       IF(DZ(I-2,J+1,2).LT.GX)GO TO 140
3970       IF(DZ(I-1,J,2).LT.GX)GO TO 140
3980       IF(DZ(I-1,J+1,2).LT.GX)GO TO 140
3990       IF(IR(I-1,J)/10.EQ.1.OR.IR(I-1,J)/10.EQ.3)GO TO 140
4000       IF(IR(I-1,J+1)/10.EQ.1.OR.IR(I-1,J+1)/10.EQ.3)GO TO 140
4010       XN=XN-R*(N(I,J,1)*XMM/DN(I,J,1)-N(I-1,J,1)*XME/DN(I-1,J,1))
4020 140    XN=XN/(1.0+FF)
4030       IF(ABS(XN).LT.1.0E-10)XN=0.0
4040C
4050C ------ LIMITING OF DISCHARGE -------
4060C
4070       IF(XN.GT.7.0*DD)XN=7.0*DD
4080       IF(XN.LT.-7.0*DD)XN=-7.0*DD
4090       N(I,J,2)=XN
4100       GO TO 200
4110 130    N(I,J,2)=0.0
4120       GO TO 200
4130C
4140C ------ CAL. OF DISCHARGE OF OVERFLOW ------
4150C
4160 160    Z1=Z(I,J,2)+HN(I,J)
4170       Z2=Z(I,J+1,2)+HN(I,J)
4180       ZZZ=Z1
4190       ZX=Z2
4200       IF(Z1.GT.Z2)GO TO 161
4210       ZZZ=Z2
4220       ZX=Z1
4230 161    IF(ZZZ.LT.GX)GO TO 130
4240       IF(ZZZ*0.66667.LT.ZX)GO TO 162
4250       XN=1.55*ZZZ**1.5
4260       GO TO 163
4270 162    XN=4.029*ZX*SQRT(ZZZ-ZX)
4280 163    IF(Z2.GT.Z1)XN=-XN
4290       N(I,J,2)=XN
4300 200 CONTINUE
4310    RETURN
4320    END
```

```
4330C
4340C ***** EXCHANGE FOR LAST STEP DATA TO NEXT STEP DATA *****
4350C
4360    SUBROUTINE CHANGE(IF,JF,Z,M,N,D)
4370    REAL M,N
4380    DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
4390    DIMENSION D(IF,JF,2)
4400    DO 10 J=1,JF
4410     DO 10 I=1,IF
4420      Z(I,J,1)=Z(I,J,2)
4430      M(I,J,1)=M(I,J,2)
4440      N(I,J,1)=N(I,J,2)
4450      D(I,J,1)=D(I,J,2)
4460  10 CONTINUE
4470    RETURN
4480    END
4490C
4500C ******* INPUT OF BOUNDARY DATA (WATER LEVEL) *******
4510C
4520    SUBROUTINE DATA(NN,B)
4530    DIMENSION B(1800,14)
4540    DO 10 K=1,1800
4550     READ(NN)(B(K,J),J=1,14)
4560  10 CONTINUE
4570    RETURN
4580    END
4590C
4600C  ************** OUTPUT OF DATA ******************
4610C
4620    SUBROUTINE OUT3(NN,IF,JF,ZM)
4630    DIMENSION ZM(IF,JF)
4640    DO 10 I=1,IF
4650     WRITE(NN)(ZM(I,J),J=1,JF)
4660  10 CONTINUE
4670    RETURN
4680    END
4690C
4700C ********* OUTPUT OF WATER LEVE;L & DISCHARGE **********
4710C
4720    SUBROUTINE OUT1(NN,IF,JF,Z,M,N,DM,DN,H)
4730    REAL M,N
4740    DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
4750    DIMENSION DM(IF,JF,2),DN(IF,JF,2)
4760    DIMENSION VEL(200,200,2)
4770    DATA GX/1.0E-3/
4780    DO 20 J=2,JF
4790     DO 20 I=2,IF
4800      IF(H(I,J).LT.-9.9)GOTO 21
4810      UU=0.0
4820      VV=0.0
4830      IF(DM(I,J,1).GT.GX)UU=M(I,J,1)/DM(I,J,1)
4840      IF(DN(I,J,1).GT.GX)VV=N(I,J,1)/DN(I,J,1)
4850      UUU=0.0
4860      VVV=0.0
4870      IF(DM(I-1,J,1).GT.GX)UUU=M(I-1,J,1)/DM(I-1,J,1)
4880      IF(DN(I,J-1,1).GT.GX)VVV=N(I,J-1,1)/DN(I,J-1,1)
4890      VEL(I,J,1)=0.5*(UU+UUU)
4900      VEL(I,J,2)=0.5*(VV+VVV)
4910      GOTO 20
4920  21   VEL(I,J,1)=0.0
4930       VEL(I,J,2)=0.0
4940  20 CONTINUE
4950     DO 30 I=1,IF
4960      WRITE(NN)(Z(I,J,1),J=1,JF)
4970      WRITE(NN)(VEL(I,J,1),J=1,JF)
4980      WRITE(NN)(VEL(I,J,2),J=1,JF)
4990  30 CONTINUE
5000    RETURN
5010    END
5020C
5030    SUBROUTINE OUT2(NN,IF,JF,Z,M,N,ZZ)
5040    REAL M,N
```

```
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),ZZ(IF,JF,3)
      DO 10 I=1,IF
       DO 10 J=1,JF
        ZZ(I,J,1)=Z(I,J,2)
        ZZ(I,J,2)=M(I,J,2)
        ZZ(I,J,3)=N(I,J,2)
   10 CONTINUE
      DO 20 L=1,3
       DO 20 I=1,IF
        WRITE(NN)(ZZ(I,J,L),J=1,JF)
   20 CONTINUE
      RETURN
      END
C
C ****** OUTPUT OF COMPUTED RESULTS ON THE DISPLAY *******
C
      SUBROUTINE OUT6(IF,JF,Z,H,K)
      DIMENSION KM(47),LW(120),Z(IF,JF,2)
      DIMENSION H(IF,JF)
      DATA KM/1H ,1HW,1HV,1HU,1HT,1HS,1HR,1HP,1HO,1HN,1HM,1HL,1HK,1HJ,
     & 1HI,1HH,1HG,1HF,1HE,1HD,1HC,1HB,1HA,1H*,1H1,1H2,1H3,1H4,1H5,1H6,
     & 1H7,1H8,1H9,1H0,1H,,1H-,1H+,1H/,1H#,1H$,1H%,1H&,1H=,1H?,1H(,1H),
     & 1H./
      WRITE(6,110)K
  110 FORMAT(//1H ,2X,"K=",I5)
      LLL=JF/120+1
      IF(MOD(JF,120).EQ.0)LLL=LLL-1
      DO 100 L=1,LLL
       J1=120*(L-1)+1
       J2=J1+119
       IF(J2.GT.JF)J2=JF
       JX=J1+9
       WRITE(6,200)(J,J=JX,J2,10)
  200  FORMAT(1H ,4X,12I10)
       DO 90 I=1,IF
        DO 80 J=J1,J2
         KKM=24
         IF(Z(I,J,2).GT.0.0)KKM=IFIX((Z(I,J,2)+0.50)/1.0)+24
         IF(Z(I,J,2).LT.0.0)KKM=IFIX((Z(I,J,2)-0.50)/1.0)+24
         IF(KKM.LT.2)KKM=2
         IF(KKM.GT.46)KKM=46
         IF(H(I,J).LT.0.0)KKM=1
         LW(J)=KM(KKM)
   80    CONTINUE
        WRITE(6,101)I,(LW(LL),LL=J1,J2)
  101    FORMAT(1H ,I3,1X,120A1)
   90  CONTINUE
  100 CONTINUE
      RETURN
      END
C
C     ********* CHECK OF MAXIMUM VALUE ***********
C
      SUBROUTINE MAX(IF,JF,Z,ZM)
      DIMENSION Z(IF,JF,2),ZM(IF,JF)
      DO 10 I=2,IF
       DO 10 J=2,JF
        IF(ZM(I,J).LT.Z(I,J,2))ZM(I,J)=Z(I,J,2)
   10 CONTINUE
      RETURN
      END
C
      SUBROUTINE BNC(IF,JF,Z,B,ISS,JSS,JES,KK,DT)
      DIMENSION Z(IF,JF,2),B(1800,14),BC(14)
      KT=(KK-1)*DT
      K1=KT/3+1
      IF(K1+1.GE.1800)K1=1799
      K2=MOD(KT,3)
      DO 10 J=1,14
   10 BC(J)=(K2*B(K1+1,J)+(3-K2)*B(K1,J))/3.0
      DO 20 J=JSS+2,JES-2
       JX=J-(JSS+1)
```

```
        JJ=JX/4+1
        JY=2*MOD(JX,4)+1
        Z(ISS,J,2)=(JY*BC(JJ+1)+(8-JY)*BC(JJ))/8.0
   20 CONTINUE
      Z(ISS,JSS,2)=BC(1)
      Z(ISS,JSS+1,2)=BC(1)
      Z(ISS,JES-1,2)=BC(14)
      Z(ISS,JES,2)=BC(14)
      RETURN
      END
C
C ******** OUTPUT OF WATER LEVEL TIMEHISTORIES *********
C
      SUBROUTINE POINT(NN,IF,JF,IM,Z,IP,JP,PZ)
      DIMENSION Z(IF,JF,2),IP(IM),JP(IM),PZ(IM)
      DO 10 I=1,IM
        II=IP(I)
        JJ=JP(I)
        PZ(I)=Z(II,JJ,2)
   10 CONTINUE
      WRITE(NN)(PZ(I),I=1,IM)
      RETURN
      END
C
C *********** OUTPUT OF LAST STEP DATA **************
C
      SUBROUTINE OUT4(NN,IF,JF,Z,M,N,H,ZM)
      REAL M,N
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
      DIMENSION ZM(IF,JF,3)
      DO 10 I=1,IF
        WRITE(NN)(Z(I,J,2),J=1,JF)
        WRITE(NN)(M(I,J,2),J=1,JF)
        WRITE(NN)(N(I,J,2),J=1,JF)
        WRITE(NN)(H(I,J),J=1,JF)
        DO 20 K=1,3
   20   WRITE(NN)(ZM(I,J,K),J=1,JF)
   10 CONTINUE
      RETURN
      END
C
C *********** INPUT OF LAST STEP DATA **************
C
      SUBROUTINE INDATA(NN,IF,JF,Z,M,N,H,ZM)
      REAL M,N
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
      DIMENSION ZM(IF,JF,3)
      DO 10 I=1,IF
        READ(NN)(Z(I,J,1),J=1,JF)
        READ(NN)(M(I,J,1),J=1,JF)
        READ(NN)(N(I,J,1),J=1,JF)
        READ(NN)(H(I,J),J=1,JF)
        DO 20 K=1,3
   20   READ(NN)(ZM(I,J,K),J=1,JF)
   10 CONTINUE
      RETURN
      END
```

```
*#RUN : IAP OPT=2 F=/SNRK/JP-MYK(10)
C
C    *************** TUNAMI-N3 ****************
C      Tohoku University's Numerical-Analysis Model
C         for Investigation of tsunami
C          Near-field Tsunami version
C
C       1896 SANRIKU TSUNAMIS NEAR MIYAKO BAY
C         --- LINEAR LONG WAVE THEORY ----
C             NUMERICAL EXPERIMENT
C                            91.7.25
C                  BY
C
C           F.IMAMURA, TOHOKU UNIV.
C
C
      REAL MA,NA,MB,NB,MC,NC,MD,ND
      DOUBLE PRECISION T1,T2,T3
      PARAMETER(IA=90,JA=130,IB=37,JB=85)
      PARAMETER(IC=43,JC=163,ID=61,JD=73)
      PARAMETER(DA=5400,DB=1800,DC=600,DD=200)
      PARAMETER(TA=3.0,TB=3.0,TC=3.0,TD=3.0)
      PARAMETER(KS=1,KE=1000,KD=200,KC=3)
      COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
      COMMON /HA/ HA(IA,JA)
      COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
      COMMON /HB/ HB(IB,JB)
      COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
      COMMON /HC/ HC(IC,JC)
      COMMON /XD/ ZD(ID,JD,2),MD(ID,JD,2),ND(ID,JD,2)
      COMMON /HD/ HD(ID,JD)
C
C ******** INITIAL CONDITION *********
C
      CALL CDEPTH
      WRITE(6,*)"DATA OK"
      CALL CINTL
      WRITE(6,*)"INITIAL OK"
      CALL OUTPUT(IA,JA,ZA,HA,1,1,1)
C       CALL COUT  (0,KD)
      CALL CLOCK(T1)
C
C ******** MAIN CALCULATION ********
C
      DO 10 K=KS,KE
        KK=K
        CALL CMASS
        CALL CJNZ
        CALL CMMNT
        CALL CJNQ
        CALL COUT(KK,KD)
        CALL CCHNG
   10 CONTINUE
      CALL CLOCK(T2)
      T3=T2-T1
      WRITE(6,*)KK,T3
      STOP
      END
C
      SUBROUTINE CDEPTH
C
      PARAMETER(IA=90,JA=130,IB=37,JB=85)
      PARAMETER(IC=43,JC=163,ID=61,JD=73)
      COMMON /HA/ HA(IA,JA)
      COMMON /HB/ HB(IB,JB)
      COMMON /HC/ HC(IC,JC)
      COMMON /HD/ HD(ID,JD)
      DIMENSION LAB(4),LBC(4),LCD(4)
      DATA LAB/27,44,38,71/
      DATA LBC/12,11,25,64/
      DATA LCD/16,79,35,102/
C
```

```
      NN=10
      DO 10 I=1,IA
   10 READ(NN,100)(HA(I,J),J=1,JA)
      DO 20 I=1,IB
   20 READ(NN,100)(HB(I,J),J=1,JB)
      DO 30 I=1,IC
   30 READ(NN,100)(HC(I,J),J=1,JC)
      DO 40 I=1,ID
   40 READ(NN,100)(HD(I,J),J=1,JD)
  100 FORMAT(10F8.1)
C
C ****** CAHNGE OF DEPTH DATA ********
C
      CALL CHH(IA,JA,IB,JB,HA,HB,LAB,1110)
      CALL CHH(IB,JB,IC,JC,HB,HC,LBC,1110)
      CALL CHH(IC,JC,ID,JD,HC,HD,LCD,1110)
      RETURN
      END
C
      SUBROUTINE CINTL
C
      REAL MA,NA,MB,NB,MC,NC,MD,ND
      PARAMETER(IA=90,JA=130,IB=37,JB=85)
      PARAMETER(IC=43,JC=163,ID=61,JD=73)
      PARAMETER(DA=5400,DB=1800,DC=600,DD=200)
      PARAMETER(TA=3.0,TB=3.0,TC=3.0,TD=3.0)
      COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
      COMMON /HA/ HA(IA,JA)
      COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
      COMMON /HB/ HB(IB,JB)
      COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
      COMMON /HC/ HC(IC,JC)
      COMMON /XD/ ZD(ID,JD,2),MD(ID,JD,2),ND(ID,JD,2)
      COMMON /HD/ HD(ID,JD)
C
      DO 100 L=1,2
        DO 10 J=1,JA
          DO 10 I=1,IA
            ZA(I,J,L)=0.0 ; MA(I,J,L)=0.0 ; NA(I,J,L)=0.0
   10   CONTINUE
        DO 20 J=1,JB
          DO 20 I=1,IB
            ZB(I,J,L)=0.0 ; MB(I,J,L)=0.0 ; NB(I,J,L)=0.0
   20   CONTINUE
        DO 30 J=1,JC
          DO 30 I=1,IC
            ZC(I,J,L)=0.0 ; MC(I,J,L)=0.0 ; NC(I,J,L)=0.0
   30   CONTINUE
        DO 40 J=1,JD
          DO 40 I=1,ID
            ZD(I,J,L)=0.0 ; MD(I,J,L)=0.0 ; ND(I,J,L)=0.0
   40   CONTINUE
  100 CONTINUE
C
      IS=20 ; JS=20 ; IE=90 ; JE=120
      CALL DEFORM(IA,JA,IS,JS,IE,JE,ZA,DA)
C
      RETURN
      END
C
      SUBROUTINE CMASS
C
      REAL MA,NA,MB,NB,MC,NC,MD,ND
      PARAMETER(IA=90,JA=130,IB=37,JB=85)
      PARAMETER(IC=43,JC=163,ID=61,JD=73)
      PARAMETER(DA=5400,DB=1800,DC=600,DD=200)
      PARAMETER(TA=3.0,TB=3.0,TC=3.0,TD=3.0)
      COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
      COMMON /HA/ HA(IA,JA)
      COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
      COMMON /HB/ HB(IB,JB)
      COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
```

```
1450    COMMON /HC/ HC(IC,JC)
1460    COMMON /XD/ ZD(ID,JD,2),MD(ID,JD,2),ND(ID,JD,2)
1470    COMMON /HD/ HD(ID,JD)
1480    LP=1
1490    CALL MASS(IA,JA,ZA,MA,NA,HA,DA,TA,LP,1)
1500    CALL MASS(IB,JB,ZB,MB,NB,HB,DB,TB,LP,1)
1510    CALL MASS(IC,JC,ZC,MC,NC,HC,DC,TC,LP,1)
1520    CALL MASS(ID,JD,ZD,MD,ND,HD,DD,TD,LP,1)
1530    RETURN
1540    END
1550C
1560    SUBROUTINE CJNZ
1570C
1580    REAL MA,NA,MB,NB,MC,NC,MD,ND
1590    PARAMETER(IA=90,JA=130,IB=37,JB=85)
1600    PARAMETER(IC=43,JC=163,ID=61,JD=73)
1610    PARAMETER(DA=5400,DB=1800,DC=600,DD=200)
1620    PARAMETER(TA=3.0,TB=3.0,TC=3.0,TD=3.0)
1630    COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
1640    COMMON /HA/ HA(IA,JA)
1650    COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
1660    COMMON /HB/ HB(IB,JB)
1670    COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
1680    COMMON /HC/ HC(IC,JC)
1690    COMMON /XD/ ZD(ID,JD,2),MD(ID,JD,2),ND(ID,JD,2)
1700    COMMON /HD/ HD(ID,JD)
1710    DIMENSION LAB(4),LBC(4),LCD(4)
1720    DATA LAB/27,44,38,71/
1730    DATA LBC/12,11,25,64/
1740    DATA LCD/16,79,35,102/
1750    LP=1
1760    CALL BOUT(IA,JA,ZA,MA,LP,1)
1770    CALL JNZ(IA,JA,IB,JB,ZA,ZB,HB,LAB,1110,LP,1)
1780    CALL JNZ(IB,JB,IC,JC,ZB,ZC,HC,LBC,1110,LP,1)
1790    CALL JNZ(IC,JC,ID,JD,ZC,ZD,HD,LCD,1110,LP,1)
1800    RETURN
1810    END
1820C
1830    SUBROUTINE CMMNT
1840C
1850    REAL MA,NA,MB,NB,MC,NC,MD,ND
1860    PARAMETER(IA=90,JA=130,IB=37,JB=85)
1870    PARAMETER(IC=43,JC=163,ID=61,JD=73)
1880    PARAMETER(DA=5400,DB=1800,DC=600,DD=200)
1890    PARAMETER(TA=3.0,TB=3.0,TC=3.0,TD=3.0)
1900    COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
1910    COMMON /HA/ HA(IA,JA)
1920    COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
1930    COMMON /HB/ HB(IB,JB)
1940    COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
1950    COMMON /HC/ HC(IC,JC)
1960    COMMON /XD/ ZD(ID,JD,2),MD(ID,JD,2),ND(ID,JD,2)
1970    COMMON /HD/ HD(ID,JD)
1980    LP=1
1990    CALL MMNT(IA,JA,ZA,MA,NA,HA,DA,TA,LP,1)
2000    CALL MMNT(IB,JB,ZB,MB,NB,HB,DB,TB,LP,1)
2010    CALL MMNT(IC,JC,ZC,MC,NC,HC,DC,TC,LP,1)
2020    CALL MMNT(ID,JD,ZD,MD,ND,HD,DD,TD,LP,1)
2030    RETURN
2040    END
2050C
2060    SUBROUTINE CJNQ
2070C
2080    REAL MA,NA,MB,NB,MC,NC,MD,ND
2090    PARAMETER(IA=90,JA=130,IB=37,JB=85)
2100    PARAMETER(IC=43,JC=163,ID=61,JD=73)
2110    COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
2120    COMMON /HA/ HA(IA,JA)
2130    COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
2140    COMMON /HB/ HB(IB,JB)
2150    COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
2160    COMMON /HC/ HC(IC,JC)
```

```
2170     COMMON /XD/ ZD(ID,JD,2),MD(ID,JD,2),ND(ID,JD,2)
2180     COMMON /HD/ HD(ID,JD)
2190     DIMENSION LAB(4),LBC(4),LCD(4)
2200     DATA LAB/27,44,38,71/
2210     DATA LBC/12,11,25,64/
2220     DATA LCD/16,79,35,102/
2230     LP=1
2240     CALL JNQ(IA,JA,IB,JB,MA,NA,MB,NB,HB,LAB,1110,LP,1)
2250     CALL JNQ(IB,JB,IC,JC,MB,NB,MC,NC,HC,LBC,1110,LP,1)
2260     CALL JNQ(IC,JC,ID,JD,MC,NC,MD,ND,HD,LCD,1110,LP,1)
2270     RETURN
2280     END
2290C
2300     SUBROUTINE CCHNG
2310C
2320     REAL MA,NA,MB,NB,MC,NC,MD,ND
2330     PARAMETER(IA=90,JA=130,IB=37,JB=85)
2340     PARAMETER(IC=43,JC=163,ID=61,JD=73)
2350     PARAMETER(TA=3.0,TB=3.0,TC=3.0,TD=3.0)
2360     COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
2370     COMMON /HA/ HA(IA,JA)
2380     COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
2390     COMMON /HB/ HB(IB,JB)
2400     COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
2410     COMMON /HC/ HC(IC,JC)
2420     COMMON /XD/ ZD(ID,JD,2),MD(ID,JD,2),ND(ID,JD,2)
2430     COMMON /HD/ HD(ID,JD)
2440     LP=1
2450     CALL CHANG(IA,JA,ZA,MA,NA,LP,1)
2460     CALL CHANG(IB,JB,ZB,MB,NB,LP,1)
2470     CALL CHANG(IC,JC,ZC,MC,NC,LP,1)
2480     CALL CHANG(ID,JD,ZD,MD,ND,LP,1)
2490     RETURN
2500     END
2510C
2520C
2530     SUBROUTINE MASS(IF,JF,Z,M,N,H,DX,DT,LP,LT)
2540C
2550     REAL M,N
2560     DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
2570     DIMENSION H(IF,JF)
2580     IF(MOD(LP,LT).NE.0)RETURN
2590     DO 10 J=2,JF
2600      DO 10 I=2,IF
2610       IF(H(I,J).GT.0.0)THEN
2620        Z(I,J,2)=Z(I,J,1)-DT/DX*(M(I,J,1)-M(I-1,J,1)+N(I,J,1)-N(I,
2630    &       J-1,1))
2640       ELSE
2650        Z(I,J,2)=0.0
2660       ENDIF
2670   10 CONTINUE
2680     DO 20 J=2,JF
2690      DO 20 I=2,IF
2700       IF(ABS(Z(I,J,2)).LT.1.0E-5)Z(I,J,2)=0.0
2710   20 CONTINUE
2720     RETURN
2730     END
2740C
2750     SUBROUTINE MMNT(IF,JF,Z,M,N,H,DX,DT,LP,LT)
2760C
2770     REAL M,N
2780     PARAMETER(GG=9.8)
2790     DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
2800     DIMENSION H(IF,JF)
2810     IF(MOD(LP,LT).NE.0)RETURN
2820     DO 10 J=2,JF-1
2830      DO 10 I=2,IF-1
2840       HH=0.5*(H(I,J)+H(I+1,J))*GG*DT/DX
2850       IF(HH.GT.0.0)THEN
2860        M(I,J,2)=M(I,J,1)-HH*(Z(I+1,J,2)-Z(I,J,2))
2870       ELSE
2880        M(I,J,2)=0.0
```

```
            ENDIF
   10 CONTINUE
        DO 20 J=2,JF-1
          DO 20 I=2,IF-1
            HH=0.5*(H(I,J)+H(I,J+1))*GG*DT/DX
            IF(HH.GT.0.0)THEN
              N(I,J,2)=N(I,J,1)-HH*(Z(I,J+1,2)-Z(I,J,2))
            ELSE
              N(I,J,2)=0.0
            ENDIF
   20 CONTINUE
C
        DO 16 I=1,IF-1
          HH=0.5*(H(I,JF)+H(I+1,JF))*GG*DT/DX
          IF(HH.GT.0.0)THEN
            M(I,JF,2)=M(I,JF,1)-HH*(Z(I+1,JF,2)-Z(I,JF,2))
          ELSE
            M(I,JF,2)=0.0
          ENDIF
   16 CONTINUE
        DO 26 J=1,JF-1
          HH=0.5*(H(IF,J)+H(IF,J+1))*GG*DT/DX
          IF(HH.GT.0.0)THEN
            N(IF,J,2)=N(IF,J,1)-HH*(Z(IF,J+1,2)-Z(IF,J,2))
          ELSE
            N(IF,J,2)=0.0
          ENDIF
   26 CONTINUE
        RETURN
        END
C
        SUBROUTINE BOUT
C
        REAL MA,NA
        PARAMETER(IA=90,JA=130)
        COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
        COMMON /HA/ HA(IA,JA)
        DO 10 KK=1,2
          J=2
          IF(KK.EQ.2)J=JA
          DO 10 I=3,IA
            IF(HA(I,J).LT.0.0)GOTO 10
            CC=SQRT(9.8*HA(I,J))
            UU=0.5*ABS(MA(I,J,2)+MA(I-1,J,2))
            IF(J.EQ.2)UU=SQRT(UU**2+NA(I,J,2)**2)
            IF(J.EQ.JA)UU=SQRT(UU**2+NA(I,J-1,2)**2)
            ZZ=UU/CC
            IF(J.EQ.2.AND.NA(I,J,2).GT.0.0)ZZ=-ZZ
            IF(J.EQ.JA.AND.NA(I,J-1,2).LT.0.0)ZZ=-ZZ
            ZA(I,J,2)=ZZ
   10 CONTINUE
        DO 20 KK=1,2
          I=2
          IF(KK.EQ.2)I=IA
          DO 20 J=3,JA
            IF(HA(I,J).LT.0.0)GOTO 20
            CC=SQRT(9.8*HA(I,J))
            UU=0.5*ABS(NA(I,J,2)+NA(I,J-1,2))
            IF(I.EQ.2)UU=SQRT(UU**2+MA(I,J,2)**2)
            IF(I.EQ.IA)UU=SQRT(UU**2+MA(I-1,J,2)**2)
            ZZ=UU/CC
            IF(I.EQ.2.AND.MA(I,J,2).GT.0.0)ZZ=-ZZ
            IF(I.EQ.IA.AND.MA(I-1,J,2).LT.0.0)ZZ=-ZZ
            ZA(I,J,2)=ZZ
   20 CONTINUE
        RETURN
        END
C
        SUBROUTINE CHH(IX,JX,IY,JY,HX,HY,L0,BCHK)
C
        INTEGER BCHK,CHK
        DIMENSION HX(IX,JX),HY(IY,JY),L0(4)
```

```
C
      ISS=2 ; JSS=2 ; IES=IY ; JES=JY ; ISL=L0(1) ; JSL=L0(2)
      IEL=L0(3) ; JEL=L0(4)
      CHK=BCHK
      KB=CHK/1000
      IF(KB.EQ.1)THEN
        CHK=CHK-1000
        II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
        DO WHILE(I1+2.LE.IES)
          S=0.0 ; L=0
          DO 10 I=I1,I1+2
            DO 10 J=J1,J1+2
              IF(HY(I,J).GT.0.0)THEN
                S=S+HY(I,J)
                L=L+1
              ENDIF
   10       CONTINUE
          IF(L.GE.5)THEN
            HX(II,JJ)=S/L
          ENDIF
          II=II+1 ; I1=I1+3
        ENDDO
      ENDIF
C
      KB=CHK/100
      IF(KB.EQ.1)THEN
        CHK=CHK-100
        II=IEL ; JJ=JSL ; I1=IES-2 ; J1=JSS
        DO WHILE(J1+1.LE.JES)
          S=0.0 ; L=0
          DO 20 I=I1,I1+2
            DO 20 J=J1,J1+2
              IF(HY(I,J).GT.0.0)THEN
                S=S+HY(I,J)
                L=L+1
              ENDIF
   20       CONTINUE
          IF(L.GE.5)THEN
            HX(II,JJ)=S/L
          ENDIF
          JJ=JJ+1 ; J1=J1+3
        ENDDO
      ENDIF
C
      KB=CHK/10
      IF(KB.EQ.1)THEN
        CHK=CHK-10
        II=ISL ; JJ=JEL ; I1=ISS ; J1=JES-2
        DO WHILE(I1+2.LE.IES)
          S=0.0 ; L=0
          DO 30 I=I1,I1+2
            DO 30 J=J1,J1+2
              IF(HY(I,J).GT.0.0)THEN
                S=S+HY(I,J)
                L=L+1
              ENDIF
   30       CONTINUE
          IF(L.GE.5)THEN
            HX(II,JJ)=S/L
          ENDIF
          II=II+1 ; I1=I1+3
        ENDDO
      ENDIF
C
      IF(CHK.EQ.1)THEN
        II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
        DO WHILE(J1+2.LE.JES)
          S=0.0 ; L=0
          DO 40 I=I1,I1+2
            DO 40 J=J1,J1+2
              IF(HY(I,J).GT.0.0)THEN
                S=S+HY(I,J)
```

```
           L=L+1
          ENDIF
   40    CONTINUE
        IF(L.GE.5)THEN
         HX(II,JJ)=S/L
        ENDIF
        JJ=JJ+1 ; J1=J1+3
       ENDDO
      ENDIF
      RETURN
      END
C
      SUBROUTINE JNZ(IX,JX,IY,JY,ZX,ZY,HY,L0,BCHK,LP,LT)
C
      INTEGER BCHK,CHK
      DIMENSION ZX(IX,JX,2),ZY(IY,JY,2)
      DIMENSION HY(IY,JY),L0(4)
      IF(MOD(LP,LT).NE.0)RETURN
      ISS=2 ; JSS=2 ; IES=IY ; JES=JY ; ISL=L0(1) ; JSL=L0(2)
      IEL=L0(3) ; JEL=L0(4)
      CHK=BCHK
      KB=CHK/1000
      IF(KB.EQ.1)THEN
       CHK=CHK-1000
       II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
       DO WHILE(I1+2.LE.IES)
        S=0.0 ; L=0
        DO 10 I=I1,I1+2
         DO 10 J=J1,J1+2
          IF(HY(I,J).GT.0.0)THEN
           S=S+ZY(I,J,2)
           L=L+1
          ENDIF
   10    CONTINUE
        IF(L.GE.5)THEN
         ZX(II,JJ,2)=S/L
        ELSE
         ZX(II,JJ,2)=0.0
        ENDIF
        II=II+1 ; I1=I1+3
       ENDDO
      ENDIF
C
      KB=CHK/100
      IF(KB.EQ.1)THEN
       CHK=CHK-100
       II=IEL ; JJ=JSL ; I1=IES-2 ; J1=JSS
       DO WHILE(J1+1.LE.JES)
        S=0.0 ; L=0
        DO 20 I=I1,I1+2
         DO 20 J=J1,J1+2
          IF(HY(I,J).GT.0.0)THEN
           S=S+ZY(I,J,2)
           L=L+1
          ENDIF
   20    CONTINUE
        IF(L.GE.5)THEN
         ZX(II,JJ,2)=S/L
        ELSE
         ZX(II,JJ,2)=0.0
        ENDIF
        JJ=JJ+1 ; J1=J1+3
       ENDDO
      ENDIF
C
      KB=CHK/10
      IF(KB.EQ.1)THEN
       CHK=CHK-10
       II=ISL ; JJ=JEL ; I1=ISS ; J1=JES-2
       DO WHILE(I1+2.LE.IES)
        S=0.0 ; L=0
        DO 30 I=I1,I1+2
```

```
         DO 30 J=J1,J1+2
          IF(HY(I,J).GT.0.0)THEN
           S=S+ZY(I,J,2)
           L=L+1
          ENDIF
   30    CONTINUE
        IF(L.GE.5)THEN
         ZX(II,JJ,2)=S/L
        ELSE
         ZX(II,JJ,2)=0.0
        ENDIF
        II=II+1 ; I1=I1+3
       ENDDO
      ENDIF
C
      IF(CHK.EQ.1)THEN
       II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
       DO WHILE(J1+2.LE.JES)
        S=0.0 ; L=0
        DO 40 I=I1,I1+2
         DO 40 J=J1,J1+2
          IF(HY(I,J).GT.0.0)THEN
           S=S+ZY(I,J,2)
           L=L+1
          ENDIF
   40    CONTINUE
        IF(L.GE.5)THEN
         ZX(II,JJ,2)=S/L
        ELSE
         ZX(II,JJ,2)=0.0
        ENDIF
        JJ=JJ+1 ; J1=J1+3
       ENDDO
      ENDIF
      RETURN
      END
C
      SUBROUTINE JNQ(IX,JX,IY,JY,MX,NX,MY,NY,HY,L0,BCHK,LP,LT)
C
      INTEGER BCHK,CHK
      REAL MX,NX,MY,NY
      DIMENSION MX(IX,JX,2),NX(IX,JX,2),HY(IY,JY)
      DIMENSION MY(IY,JY,2),NY(IY,JY,2),L0(4)
      IF(MOD(LP,LT).NE.0)RETURN
      ISS=2 ; JSS=2 ; IES=IY ; JES=JY ; ISL=L0(1) ; JSL=L0(2)
      IEL=L0(3) ; JEL=L0(4)
      CHK=BCHK
      KB=CHK/1000
      IF(KB.EQ.1)THEN
       CHK=CHK-1000
       I=ISS ; J=JSS-1 ; II=ISL-1 ; JJ=JSL-1
       DO WHILE(I.LE.IES)
        SI=(I-ISS+2)/3.0
        IS=IFIX(SI) ; DI=SI-IS ; II=IS+ISL-1
        NY(I,J,2)=(1-DI)*NX(II,JJ,2)+DI*NX(II+1,JJ,2)
        IF(HY(I,J+1).LT.0.0)NY(I,J,2)=0.0
        I=I+1
       ENDDO
      ENDIF
C
      KB=CHK/100
      IF(KB.EQ.1)THEN
       CHK=CHK-100
       I=IES ; J=JSS ; II=IEL ; JJ=JSL-1
       DO WHILE(J.LE.JES)
        SJ=(J-JSS+2)/3.0
        JS=IFIX(SJ) ; DJ=SJ-JS ; JJ=JS+JSL-1
        MY(I,J,2)=(1-DJ)*MX(II,JJ,2)+DJ*MX(II,JJ+1,2)
        IF(HY(I,J).LT.0.0)MY(I,J,2)=0.0
        J=J+1
       ENDDO
      ENDIF
```

```
5770C
5780      KB=CHK/10
5790      IF(KB.EQ.1)THEN
5800        CHK=CHK-10
5810        I=ISS ; J=JES ; II=ISL-1 ; JJ=JEL
5820        DO WHILE(I.LE.IES)
5830          SI=(I-ISS+2)/3.0
5840          IS=IFIX(SI) ; DI=SI-IS ; II=IS+ISL-1
5850          NY(I,J,2)=(1-DI)*NX(II,JJ,2)+DI*NX(II+1,JJ,2)
5860          IF(HY(I,J).LT.0.0)NY(I,J,2)=0.0
5870          I=I+1
5880        ENDDO
5890      ENDIF
5900C
5910      IF(CHK.EQ.1)THEN
5920        I=ISS-1 ; J=JSS ; II=ISL-1 ; JJ=JSL-1
5930        DO WHILE(J.LE.JES)
5940          SJ=(J-JSS+2)/3.0
5950          JS=IFIX(SJ) ; DJ=SJ-JS ; JJ=JS+JSL-1
5960          MY(I,J,2)=(1-DJ)*MX(II,JJ,2)+DJ*MX(II,JJ+1,2)
5970          IF(HY(I+1,J).LT.0.0)MY(I,J,2)=0.0
5980          J=J+1
5990        ENDDO
6000      ENDIF
6010      RETURN
6020      END
6030C
6040      SUBROUTINE COUT(KK,KD)
6050C
6060      REAL MA,NA,MB,NB,MC,NC,MD,ND
6070      PARAMETER(IA=90,JA=130,IB=37,JB=85)
6080      PARAMETER(IC=43,JC=163,ID=61,JD=73)
6090      COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
6100      COMMON /HA/ HA(IA,JA)
6110      COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
6120      COMMON /HB/ HB(IB,JB)
6130      COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
6140      COMMON /HC/ HC(IC,JC)
6150      COMMON /XD/ ZD(ID,JD,2),MD(ID,JD,2),ND(ID,JD,2)
6160      COMMON /HD/ HD(ID,JD)
6170      CALL OUTPUT(IA,JA,ZA,HA,KK,KD,1)
6180      CALL OUTPUT(IB,JB,ZB,HB,KK,KD,2)
6190      CALL OUTPUT(IC,JC,ZC,HC,KK,KD,3)
6200      CALL OUTPUT(ID,JD,ZD,HD,KK,KD,4)
6210      RETURN
6220      END
6230C
6240      SUBROUTINE OUTPUT(IF,JF,Z,H,KK,KD,CHR)
6250C
6260      INTEGER CHR
6270      PARAMETER(GX=1.E-2)
6280      DIMENSION Z(IF,JF,2),H(IF,JF)
6290      DIMENSION KW(100),KM(23)
6300      CHARACTER NAME(4)*10
6310      DATA KM/1HJ,1HI,1HH,1HG,1HF,1HE,1HD,1HC,1HB,1HA,1H.,1H1,1H2,1H3,
6320     & 1H4,1H5,1H6,1H7,1H8,1H9,1H0,1H*,1H /
6330      NAME(1)="A REGION"
6340      NAME(2)="B REGION"
6350      NAME(3)="C REGION"
6360      NAME(4)="D REGION"
6370      IF(MOD(KK,KD).NE.0)RETURN
6380      WRITE(6,450)NAME(CHR)
6390  450 FORMAT(1H ,/,10X,A10)
6400      IN=IF/20
6410      WRITE(6,410)KK,(KI,KI=20,IN*20,20)
6420      DO 10 J=JF,1,-1
6430        JS=J-1 ; JE=J
6440        IF(JS.LE.0)JS=1
6450        DO 20 I=1,IF
6460C         IS=2*I-1 ; IE=IS+1 ; IF(IE.GE.IF)IE=IF
6470C         S=0.0 ; L=0
6480C           DO 30 II=IS,IE
```

```
6490C        DO 30 JJ=JS,JE
6500C          IF(H(II,JJ).GT.0.)THEN
6510       IF(H(I,J).GT.0.)THEN
6520C          L=L+1
6530C          S=S+Z(II,JJ,2)
6540C          ENDIF
6550C      30 CONTINUE
6560C       IF(L.GE.2)THEN
6570C        S=S/L
6580         S=Z(I,J,2)
6590         KMM=11+INT(S/0.2)
6600         IF(KMM.GT.21)KMM=21
6610         IF(KMM.LT.1)KMM=1
6620         IF(ABS(S).LT.GX)KMM=23
6630       ELSE
6640         KMM=22
6650       ENDIF
6660       KW(I)=KM(KMM)
6670  20   CONTINUE
6680      II=IF
6690      WRITE(6,400)J,(KW(I),I=1,II)
6700  10 CONTINUE
6710    RETURN
6720  400 FORMAT(1H ,I3,2X,100A1)
6730  410 FORMAT(1H ,/,2X,"MAP AT K=",I5,//,6X,11I10)
6740    END
6750C
6760C
6770    SUBROUTINE CHANG(IF,JF,Z,M,N,LP,LT)
6780C
6790    REAL M,N
6800    DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
6810    IF(MOD(LP,LT).NE.0)RETURN
6820    DO 10 J=1,JF
6830      DO 10 I=1,IF
6840        Z(I,J,1)=Z(I,J,2)
6850        M(I,J,1)=M(I,J,2)
6860        N(I,J,1)=N(I,J,2)
6870  10 CONTINUE
6880    RETURN
6890    END
6900C
6910    SUBROUTINE FILEOT(IF,JF,Z,M,N)
6920C
6930    REAL M,N
6940    DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
6950C
6960    DO 10 J=1,JF
6970      WRITE(25,100)(Z(I,J,2),I=1,IF)
6980      WRITE(25,100)(M(I,J,2),I=1,IF)
6990      WRITE(25,100)(N(I,J,2),I=1,IF)
7000  10 CONTINUE
7010  100 FORMAT(10E12.5)
7020    RETURN
7030    END
7040C
7050    SUBROUTINE MAX(IF,JF,IS,JS,IE,JE,Z,H,ZM,ZN)
7060C
7070    DIMENSION Z(IF,JF,2),H(IF,JF),ZM(IF,JF),ZN(IF,JF)
7080    DO 10 I=IS,IE
7090      DO 10 J=JS,JE
7100        IF(H(I,J).LT.0.0)GOTO 10
7110        IF(ZM(I,J).LT.Z(I,J,2))ZM(I,J)=Z(I,J,2)
7120        IF(ZN(I,J).GT.Z(I,J,2))ZN(I,J)=Z(I,J,2)
7130  10 CONTINUE
7140    RETURN
7150    END
7160C
7170    SUBROUTINE POINT(IF,JF,Z,NG,PZ,IP,JP,KK,KC)
7180C
7190    DIMENSION Z(IF,JF,2),PZ(NG),IP(NG),JP(NG)
7200C
```

```
      IF(MOD(KK,KC).NE.0)RETURN
      DO 10 K=1,NG
        PZ(K)=Z(IP(K),JP(K),2)
   10 CONTINUE
      WRITE(24,100)(PZ(K),K=1,NG)
  100 FORMAT(10F7.3)
      RETURN
      END
C
      SUBROUTINE PROPA(IF,JF,IS,JS,IE,JE,Z,H,LF,KK)
C
      DIMENSION Z(IF,JF,2),H(IF,JF),LF(IF,JF)
C
      LK=KK/180+1
      DO 10 I=IS,IE
        DO 10 J=JS,JE
          IF(H(I,J).GT.0.0)THEN
            IF(LF(I,J).EQ.0)THEN
              IF(ABS(Z(I,J,2)).GT.0.01)LF(I,J)=LK
            ENDIF
          ELSE
            LF(I,J)=-9
          ENDIF
   10 CONTINUE
      RETURN
      END
C
      SUBROUTINE OUTDT(IF,JF,ZM,ZN,LK)
      DIMENSION ZM(IF,JF),ZN(IF,JF),LK(IF,JF)
      DO 20 I=1,IF
        DO 20 J=1,JF
          ZN(I,J)=ZM(I,J)-ZN(I,J)
          IF(ZN(I,J).GT.100.0)ZN(I,J)=100.0
          IF(ZM(I,J).GT.100.0)ZM(I,J)=100.0
   20 CONTINUE
      DO 10 J=1,JF
        WRITE(31,100)(ZM(I,J),I=1,IF)
        WRITE(32,100)(ZN(I,J),I=1,IF)
        WRITE(33,200)(LK(I,J),I=1,IF)
   10 CONTINUE
  100 FORMAT(20F6.3)
  200 FORMAT(40I3)
      RETURN
      END
C
      SUBROUTINE DEFORM(IO,JO,IS,JS,IE,JE,Z,DX)
      REAL L
      PARAMETER(A=3.141592,B=4.848E-06)
      PARAMETER(RR=6.37E+6,E=1.7453E-2)
      PARAMETER(HH=1.0E+3,D=12.5,DL=20.0,TH=156.0,RD=58.0)
      PARAMETER(L=210.0E+3,W=50.0E+3)
      PARAMETER(YO=37.0,XO=139.67)
      PARAMETER(Y0=40.9,X0=143.3)
      DIMENSION Z(IO,JO,2)
      XL=A*RR*(X0-XO)*COS(E*YO)/180.0 ; YL=A*RR*(Y0-YO)/180.0
      H1=HH/SIN(E*DL) ; H2=HH/SIN(E*DL)+W ; DS=D*COS(E*RD)
      DD=D*SIN(E*RD)
      WRITE(6,*)XL,YL,H1,H2,DS,DD
      DO 10 I=IS,IE
        DO 10 J=JS,JE
          XX=DX*(I-1) ; YY=DX*(J-1)
C           YY=A*RR*DR*(J-1)/(60.0*180)
C           XX=A*RR*DR*(I-1)*COS(E*(YO+DR*(J-1)/60.0))/(60.0*180)
          X1=(XX-XL)*SIN(E*TH)+(YY-YL)*COS(E*TH)
          X2=(XX-XL)*COS(E*TH)-(YY-YL)*SIN(E*TH)+HH/TAN(E*DL)
          X3=0.0
          CALL USCAL(X1,X2,X3,L,H2,E*DL,F1)
          CALL USCAL(X1,X2,X3,L,H1,E*DL,F2)
          CALL USCAL(X1,X2,X3,0.,H2,E*DL,F3)
          CALL USCAL(X1,X2,X3,0.,H1,E*DL,F4)
          CALL UDCAL(X1,X2,X3,L,H2,E*DL,G1)
          CALL UDCAL(X1,X2,X3,L,H1,E*DL,G2)
```

```
7930       CALL UDCAL(X1,X2,X3,0.,H2,E*DL,G3)
7940       CALL UDCAL(X1,X2,X3,0.,H1,E*DL,G4)
7950       US=(F1-F2-F3+F4)*DS/(12.0*A)
7960       UD=(G1-G2-G3+G4)*DD/(12.0*A)
7970       Z(I,J,1)=US+UD
7980       Z(I,J,2)=US+UD
7990  10 CONTINUE
8000     RETURN
8010     END
8020C
8030     SUBROUTINE USCAL(X1,X2,X3,C,CC,DP,F)
8040     REAL K
8050C
8060     SN=SIN(DP) ; CS=COS(DP) ; C1=C ; C2=CC*CS ; C3=CC*SN
8070     R=SQRT((X1-C1)**2+(X2-C2)**2+(X3-C3)**2)
8080     Q=SQRT((X1-C1)**2+(X2-C2)**2+(X3+C3)**2)
8090     R2=X2*SN-X3*CS
8100     R3=X2*CS+X3*SN
8110     Q2=X2*SN+X3*CS
8120     Q3=-X2*CS+X3*SN
8130     H=SQRT(Q2**2+(Q3+CC)**2)
8140     K=SQRT((X1-C1)**2+Q2**2)
8150     A1=LOG(R+R3-CC)
8160     A2=LOG(Q+Q3+CC)
8170     A3=LOG(Q+X3+C3)
8180     B1=1+3.0*(TAN(DP))**2
8190     B2=3.0*TAN(DP)/CS
8200     B3=2.0*R2*SN
8210     B4=Q2+X2*SN
8220     B5=2.0*R2**2*CS
8230     B6=R*(R+R3-CC)
8240     B7=4.0*Q2*X3*SN**2
8250     B8=2.0*(Q2+X2*SN)*(X3+Q3*SN)
8260     B9=Q*(Q+Q3+CC)
8270     B10=4.0*Q2*X3*SN
8280     B11=(X3+C3)-Q3*SN
8290     B12=4.0*Q2**2*Q3*X3*CS*SN
8300     B13=2.0*Q+Q3+CC
8310     B14=Q**3*(Q+Q3+CC)**2
8320     F=CS*(A1+B1*A2-B2*A3)+B3/R+2*SN*B4/Q-B5/B6+(B7-B8)/B9+B10*B11/
8330    & Q**3-B12*B13/B14
8340     RETURN
8350     END
8360C
8370     SUBROUTINE UDCAL(X1,X2,X3,C,CC,DP,F)
8380     REAL K
8390C
8400     SN=SIN(DP) ; CS=COS(DP) ; C1=C ; C2=CC*CS ; C3=CC*SN
8410     R=SQRT((X1-C1)**2+(X2-C2)**2+(X3-C3)**2)
8420     Q=SQRT((X1-C1)**2+(X2-C2)**2+(X3+C3)**2)
8430     R2=X2*SN-X3*CS
8440     R3=X2*CS+X3*SN
8450     Q2=X2*SN+X3*CS
8460     Q3=-X2*CS+X3*SN
8470     H=SQRT(Q2**2+(Q3+CC)**2)
8480     K=SQRT((X1-C1)**2+Q2**2)
8490     A1=LOG(R+X1-C1)
8500     A2=LOG(Q+X1-C1)
8510     B1=Q*(Q+X1-C1)
8520     B2=R*(R+X1-C1)
8530     B3=Q*(Q+Q3+CC)
8540     D1=X1-C1 ; D2=X2-C2 ; D3=X3-C3 ; D4=X3+C3 ; D5=R3-CC ; D6=Q3+CC
8550     T1=ATN(D1*D2,(H+D4)*(Q+H))
8560     T2=ATN(D1*D5,R2*R)
8570     T3=ATN(D1*D6,Q2*Q)
8580     F=SN*(D2*(2*D3/B2+4*D3/B1-4*C3*X3*D4*(2*Q+D1)/(B1**2*Q))
8590    & -6*T1+3*T2-6*T3)+CS*(A1-A2-2*(D3**2)/B2-4*(D4**2-C3*X3)/
8600    & B1-4*C3*X3*D4**2*(2*Q+X1-C1)/(B1**2*Q))+6*X3*(CS*SN*(2*D6/B1+D1/
8610    & B3)-Q2*(SN**2-CS**2)/B1)
8620     RETURN
8630     END
8640C
```

```
      REAL FUNCTION ATN(AX,AY)
      DATA GX/1.0E-6/
      AAX=ABS(AX) ; AAY=ABS(AY)
      P=AX*AY
      IF(AAX.LE.GX.AND.AAY.LE.GX)GOTO 10
      SR=ATAN2(AAX,AAY)
      ATN=SIGN(SR,P)
      RETURN
   10 WRITE(6,100)AX,AY
  100 FORMAT(1H ,"ATAN --    AX=",E15.7,2X,"AY=",E15.7)
      ATN=0.2
      RETURN
      END
```

IUGG/IOC TIME PROJECT

PART 4. PROGRAMME LIST FOR FAR-FIELD TSUNAMI

TUNAMI-F2: LINEAR THEORY FOR PROPAGATION IN THE OCEAN AND COASTAL WATERS

CONTENTS

```
C
C  *************** TUNAMI-F2 ****************
C    Tohoku University's Numerical-Analysis Model
C       for Investigation of tsunami
C        Far-field Tsunami version
C
C        1960 CHILEAN  TSUNAMI (DX=10')
C           OHMA VERSION 1.2
C                         91.7.25
C
C                BY
C
C           F.IMAMURA, TOHOKU UNIV.
C
C    Z; WATER SURFACE LEVEL   M,N;WATER DISCHARGE
C    DX; GRID LENGTH
C
      REAL M,N,MA,NA,MB,NB,MC,NC,ME,NE,MF,NF
      REAL MD1,ND1,MD2,ND2,MD3,ND3,MD4,ND4,MD5,ND5
      DOUBLE PRECISION T1,T2,T3
      PARAMETER (IO=1020,JO=738,IA=76,JA=76)
      PARAMETER (IB=136,JB=136,IC=154,JC=307)
      PARAMETER (ID1=28,JD1=55,ID2=55,JD2=40)
      PARAMETER (ID3=37,JD3=31,ID4=13,JD4=40)
      PARAMETER (ID5=46,JD5=40)
      PARAMETER (IE1=34,JE1=67,IF1=46,JF1=58)
      PARAMETER (DO=10.0,DA=4.0,DB=1.33,DC=833)
      PARAMETER (DD=278.0,DE=93.0,DF=31.0)
      PARAMETER (TO=20.0,TA=10.0,TB=5.0,TC=5.0)
      PARAMETER (TD=5.0,TE=2.5,TF=1.25)
      PARAMETER (KS=1,KE=180*32,KD=180*2,KC=3)
      DATA IS,JS,IE,JE/961,61,1020,180/
      COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
      COMMON /HH/ H(IO,JO)
      COMMON /RR/ R1(IO,JO),R2(IO,JO),R3(IO,JO),
     &R4(IO,JO),R5(IO,JO),R6(IO)
      COMMON /CC/ C1(IO),C2(IO),C3(IO),C4(IO)
      COMMON /VV/ V1(IO),V2(IO),V3(IO),V4(IO),V5(IO)
      COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
      COMMON /HA/ HA(IA,JA)
      COMMON /RA/ RA1(IA,JA),RA2(IA,JA),RA3(IA,JA),
     &RA4(IA,JA),RA5(IA,JA),RA6(IA)
      COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
      COMMON /HB/ HB(IB,JB)
      COMMON /RB/ RB1(IB,JB),RB2(IB,JB),RB3(IB,JB),
     &RB4(IB,JB),RB5(IB,JB),RB6(IB)
      COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
      COMMON /HC/ HC(IC,JC)
      COMMON /XD1/ ZD1(ID1,JD1,2),MD1(ID1,JD1,2),
     &ND1(ID1,JD1,2),HD1(ID1,JD1)
      COMMON /XD2/ ZD2(ID2,JD2,2),MD2(ID2,JD2,2),
     &ND2(ID2,JD2,2),HD2(ID2,JD2)
      COMMON /XD3/ ZD3(ID3,JD3,2),MD3(ID3,JD3,2),
     &ND3(ID3,JD3,2),HD3(ID3,JD3)
      COMMON /XD4/ ZD4(ID4,JD4,2),MD4(ID4,JD4,2),
     &ND4(ID4,JD4,2),HD4(ID4,JD4)
      COMMON /XD5/ ZD5(ID5,JD5,2),MD5(ID5,JD5,2),
     &ND5(ID5,JD5,2),HD5(ID5,JD5)
      COMMON /XE/  ZE(IE1,JE1,2),ME(IE1,JE1,2),
     &NE(IE1,JE1,2),HE(IE1,JE1)
      COMMON /XF/  ZF(IF1,JF1,2),MF(IF1,JF1,2),
     &NF(IF1,JF1,2),HF(IF1,JF1)
```

```
C
C ******** INITIAL CONDITION *********
C
      CALL RDEPTH
      CALL PARAME
      CALL INITIA (IS,JS,IE,JE)
C      CALL COUT   (0,KD)
      CALL CLOCK  (T1)
C
C  ******** MAIN CALCULATION ********
C
      DO 10 K=KS,KE
      KK=K
C
C ******** TRANSOCEANIC PROPAGATION *******
C
      CALL MASS1 (IS,JS,IE,JE)
      CALL BOUND (IS,JS,IE,JE)
      CALL LIMIT (IS,JS,IE,JE)
C
C ******** CAL. NEAR NORTH JAPAN ************
C
      IF(IS.LE.121.AND.JE.GE.592)THEN
      CALL CAL   (IS,JS,IE,JE,KK) ; ELSE
      CALL MMNT1 (IS,JS,IE,JE)    ; ENDIF
      CALL COUT  (KK,KD)
      CALL CHANGE
   10 CONTINUE
      CALL CLOCK (T2) ; T3=T2-T1
      WRITE(6,*)KK,T3
      STOP  ; END
C
C ********* CAL. NEAR NORTH JAPAN ********
C
C
      SUBROUTINE CAL (IS,JS,IE,JE,KK)
C
      REAL M,N,MA,NA,MB,NB,MC,NC,ME,NE,MF,NF
      REAL MD1,ND1,MD2,ND2,MD3,ND3,MD4,ND4,MD5,ND5
      PARAMETER (IO=1020,JO=738,IA=76,JA=76)
      PARAMETER (IB=136,JB=136,IC=154,JC=307)
      PARAMETER (ID1=28,JD1=55,ID2=55,JD2=40)
      PARAMETER (ID3=37,JD3=31,ID4=13,JD4=40)
      PARAMETER (ID5=46,JD5=40)
      PARAMETER (IE1=34,JE1=67,IF1=46,JF1=58)
      PARAMETER (DC=833.0,DD=278.0,DE=93.0,DF=31.0)
      PARAMETER (TC=5.0,TD=5.0,TE=2.5,TF=1.25)
      COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
      COMMON /HH/ H(IO,JO)
      COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
      COMMON /HA/ HA(IA,JA)
      COMMON /RA/ RA1(IA,JA),RA2(IA,JA),RA3(IA,JA),
     &RA4(IA,JA),RA5(IA,JA),RA6(IA)
      COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
      COMMON /HB/ HB(IB,JB)
      COMMON /RB/ RB1(IB,JB),RB2(IB,JB),RB3(IB,JB),
     &RB4(IB,JB),RB5(IB,JB),RB6(IB)
      COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
      COMMON /HC/ HC(IC,JC)
      COMMON /XD1/ ZD1(ID1,JD1,2),MD1(ID1,JD1,2),
     &ND1(ID1,JD1,2),HD1(ID1,JD1)
      COMMON /XD2/ ZD2(ID2,JD2,2),MD2(ID2,JD2,2),
     &ND2(ID2,JD2,2),HD2(ID2,JD2)
      COMMON /XD3/ ZD3(ID3,JD3,2),MD3(ID3,JD3,2),
     &ND3(ID3,JD3,2),HD3(ID3,JD3)
      COMMON /XD4/ ZD4(ID4,JD4,2),MD4(ID4,JD4,2),
     &ND4(ID4,JD4,2),HD4(ID4,JD4)
      COMMON /XD5/ ZD5(ID5,JD5,2),MD5(ID5,JD5,2),
     &ND5(ID5,JD5,2),HD5(ID5,JD5)
      COMMON /XE/  ZE(IE1,JE1,2),ME(IE1,JE1,2),
     &NE(IE1,JE1,2),HE(IE1,JE1)
      COMMON /XF/  ZF(IF1,JF1,2),MF(IF1,JF1,2),
     &NF(IF1,JF1,2),HF(IF1,JF1)
      DIMENSION PD(14)
```

```
1370C
1380C ----- PARAMETERS FOR CONNECTION ------
1390C
1400    DIMENSION LAB(4),LBC(4),LDE(4),LEF(4)
1410    DIMENSION LD1(4),LD2(4),LD3(4),LD4(4),LD5(4)
1420    DATA LAB/5,20,49,64/
1430    DATA LBC/6,33,74,134/
1440    DATA LD1/60,144,68,161/
1450    DATA LD2/35,182,52,194/
1460    DATA LD3/90,133,101,142/
1470    DATA LD4/114,103,117,115/
1480    DATA LD5/122,23,136,35/
1490    DATA LDE/11,19,21,40/
1500    DATA LEF/14,26,28,44/
1510C
1520    LP=1 ; DO WHILE (LP.LE.16)
1530    CALL MASS2 (IA,JA,ZA,MA,NA,RA1,RA6,LP,8)
1540    CALL MASS2 (IB,JB,ZB,MB,NB,RB1,RB6,LP,4)
1550    CALL MASS3 (IC,JC,ZC,MC,NC,HC,DC,TC,LP,4)
1560    CALL MASS3 (ID1,JD1,ZD1,MD1,ND1,HD1,DD,TD,LP,4)
1570    CALL MASS3 (ID2,JD2,ZD2,MD2,ND2,HD2,DD,TD,LP,4)
1580    CALL MASS3 (ID3,JD3,ZD3,MD3,ND3,HD3,DD,TD,LP,4)
1590    CALL MASS3 (ID4,JD4,ZD4,MD4,ND4,HD4,DD,TD,LP,4)
1600    CALL MASS3 (ID5,JD5,ZD5,MD5,ND5,HD5,DD,TD,LP,4)
1610    CALL MASS3 (IE1,JE1,ZE,ME,NE,HE,DE,TE,LP,2)
1620C    CALL MASS3 (IF1,JF1,ZF,MF,NF,HF,DF,TF,LP,1)
1630C
1640    CALL JNZOA (LP,16)
1650    CALL JNZ (IA,JA,IB,JB,ZA,ZB,HB,LAB,1101,LP,8)
1660    CALL J2Z (IB,JB,IC,JC,ZB,ZC,HC,LBC,1111,LP,4)
1670    CALL J2Z (IC,JC,ID1,JD1,ZC,ZD1,HD1,LD1,1111,LP,4)
1680    CALL J2Z (IC,JC,ID2,JD2,ZC,ZD2,HD2,LD2,1101,LP,4)
1690    CALL J2Z (IC,JC,ID3,JD3,ZC,ZD3,HD3,LD3,0110,LP,4)
1700    CALL J2Z (IC,JC,ID4,JD4,ZC,ZD4,HD4,LD4,1110,LP,4)
1710    CALL J2Z (IC,JC,ID5,JD5,ZC,ZD5,HD5,LD5,0110,LP,4)
1720    CALL JNZ (ID1,JD1,IE1,JE1,ZD1,ZE,HE,LDE,1011,LP,4)
1730C    CALL JNZ (IE1,JE1,IF1,JF1,ZE,ZF,HF,LEF,1011,LP,2)
1740C
1750    IF(LP.EQ.16) CALL MMNT1 (IS,JS,IE,JE)
1760    CALL MMNT2 (IA,JA,ZA,MA,NA,RA2,RA3,RA4,RA5,LP,8)
1770    CALL MMNT2 (IB,JB,ZB,MB,NB,RB2,RB3,RB4,RB5,LP,4)
1780    CALL MMNT3 (IC,JC,ZC,MC,NC,HC,DC,TC,LP,4)
1790    CALL MMNT3 (ID1,JD1,ZD1,MD1,ND1,HD1,DD,TD,LP,4)
1800    CALL MMNT3 (ID2,JD2,ZD2,MD2,ND2,HD2,DD,TD,LP,4)
1810    CALL MMNT3 (ID3,JD3,ZD3,MD3,ND3,HD3,DD,TD,LP,4)
1820    CALL MMNT3 (ID4,JD4,ZD4,MD4,ND4,HD4,DD,TD,LP,4)
1830    CALL MMNT3 (ID5,JD5,ZD5,MD5,ND5,HD5,DD,TD,LP,4)
1840    CALL MMNT3 (IE1,JE1,ZE,ME,NE,HE,DE,TE,LP,2)
1850C    CALL MMNT3 (IF1,JF1,ZF,MF,NF,HF,DF,TF,LP,1)
1860C
1870    CALL JNQOA (LP,8)
1880    CALL JNQ (IA,JA,IB,JB,MA,NA,MB,NB,HB,LAB,1101,LP,4)
1890    CALL JNQ (IB,JB,IC,JC,MB,NB,MC,NC,HC,LBC,1111,LP,4)
1900    CALL JNQ (IC,JC,ID1,JD1,MC,NC,MD1,ND1,HD1,LD1,1111,LP,4)
1910    CALL JNQ (IC,JC,ID2,JD2,MC,NC,MD2,ND2,HD2,LD2,1101,LP,4)
1920    CALL JNQ (IC,JC,ID3,JD3,MC,NC,MD3,ND3,HD3,LD3,0110,LP,4)
1930    CALL JNQ (IC,JC,ID4,JD4,MC,NC,MD4,ND4,HD4,LD4,1110,LP,4)
1940    CALL JNQ (IC,JC,ID5,JD5,MC,NC,MD5,ND5,HD5,LD5,0110,LP,4)
1950    CALL JNQ (ID1,JD1,IE1,JE1,MD1,ND1,ME,NE,HE,LDE,1011,LP,2)
1960C    CALL JNQ (IE1,JE1,IF1,JF1,ME,NE,MF,NF,HF,LEF,1011,LP,1)
1970C
1980    CALL CHANG (IA,JA,ZA,MA,NA,LP,8)
1990    CALL CHANG (IB,JB,ZB,MB,NB,LP,4)
2000    CALL CHANG (IC,JC,ZC,MC,NC,LP,4)
2010    CALL CHANG (ID1,JD1,ZD1,MD1,ND1,LP,4)
2020    CALL CHANG (ID2,JD2,ZD2,MD2,ND2,LP,4)
2030    CALL CHANG (ID3,JD3,ZD3,MD3,ND3,LP,4)
2040    CALL CHANG (ID4,JD4,ZD4,MD4,ND4,LP,4)
2050    CALL CHANG (ID5,JD5,ZD5,MD5,ND5,LP,4)
2060    CALL CHANG (IE1,JE1,ZE,ME,NE,LP,2)
2070C    CALL CHANG (IF1,JF1,ZF,MF,NF,LP,1)
2080    LP=LP+1 ; ENDDO
2090C
2100    IF(KK.GE.180*22.AND.MOD(KK,3).EQ.0)THEN
```

```
      PD(1)=ZD1(24,39,2)  ; PD(2)=ZD1(22,43,2)
      PD(3)=ZD1(19,42,2)  ; PD(4)=ZD1(13,14,2)
      PD(5)=ZD1(10,7,2)   ; PD(6)=ZD2(41,16,2)
      PD(7)=ZD2(35,19,2)  ; PD(8)=ZD2(39,23,2)
      PD(9)=ZD2(18,39,2)  ; PD(10)=ZD3(6,25,2)
      PD(11)=ZD3(26,7,2)  ; PD(12)=ZD4(4,27,2)
      PD(13)=ZD5(22,4,2)  ; PD(14)=ZD5(32,6,2)
      WRITE(30,100)(PD(LL),LL=1,14)
  100 FORMAT(7F8.3)
      ENDIF
C
      RETURN ; END
C
C ******* READ DEPTH DATA *******
C
      SUBROUTINE RDEPTH
C
      REAL MD1,ND1,MD2,ND2,MD3,ND3,MD4,ND4,MD5,ND5
      REAL ME,NE,MF,NF
      PARAMETER (IO=1020,JO=738,IA=76,JA=76)
      PARAMETER (IB=136,JB=136,IC=154,JC=307)
      PARAMETER (ID1=28,JD1=55,ID2=55,JD2=40)
      PARAMETER (ID3=37,JD3=31,ID4=13,JD4=40)
      PARAMETER (ID5=46,JD5=40)
      PARAMETER (IE1=34,JE1=67,IF1=46,JF1=58)
      COMMON /HH/ H(IO,JO)
      COMMON /HA/ HA(IA,JA)
      COMMON /HB/ HB(IB,JB)
      COMMON /HC/ HC(IC,JC)
      COMMON /XD1/ ZD1(ID1,JD1,2),MD1(ID1,JD1,2),
     &ND1(ID1,JD1,2),HD1(ID1,JD1)
      COMMON /XD2/ ZD2(ID2,JD2,2),MD2(ID2,JD2,2),
     &ND2(ID2,JD2,2),HD2(ID2,JD2)
      COMMON /XD3/ ZD3(ID3,JD3,2),MD3(ID3,JD3,2),
     &ND3(ID3,JD3,2),HD3(ID3,JD3)
      COMMON /XD4/ ZD4(ID4,JD4,2),MD4(ID4,JD4,2),
     &ND4(ID4,JD4,2),HD4(ID4,JD4)
      COMMON /XD5/ ZD5(ID5,JD5,2),MD5(ID5,JD5,2),
     &ND5(ID5,JD5,2),HD5(ID5,JD5)
      COMMON /XE/  ZE(IE1,JE1,2),ME(IE1,JE1,2),
     &NE(IE1,JE1,2),HE(IE1,JE1)
      COMMON /XF/  ZF(IF1,JF1,2),MF(IF1,JF1,2),
     &NF(IF1,JF1,2),HF(IF1,JF1)
      DIMENSION LAB(4),LBC(4),LDE(4),LEF(4)
      DIMENSION LD1(4),LD2(4),LD3(4),LD4(4),LD5(4)
      DATA LAB/5,20,49,64/
      DATA LBC/6,33,74,134/
      DATA LD1/60,144,68,161/
      DATA LD2/35,182,52,194/
      DATA LD3/90,133,101,142/
      DATA LD4/114,103,117,115/
      DATA LD5/122,23,136,35/
      DATA LDE/11,19,21,40/
      DATA LEF/14,26,28,44/
      DO 10 J=1,JO
   10 READ(20,900)(H(I,J),I=1,IO)
      DO 20 J=1,JA
   20 READ(21,900)(HA(I,J),I=1,IA)
      DO 30 J=1,JB
   30 READ(22,900)(HB(I,J),I=1,IB)
      DO 40 J=1,JC
   40 READ(23,900)(HC(I,J),I=1,IC)
      DO 51 J=1,JD1
   51 READ(24,900)(HD1(I,J),I=1,ID1)
      DO 52 J=1,JD2
   52 READ(24,900)(HD2(I,J),I=1,ID2)
      DO 53 J=1,JD3
   53 READ(24,900)(HD3(I,J),I=1,ID3)
      DO 54 J=1,JD4
   54 READ(24,900)(HD4(I,J),I=1,ID4)
      DO 55 J=1,JD5
   55 READ(24,900)(HD5(I,J),I=1,ID5)
      DO 60 J=1,JE1
   60 READ(25,900)(HE(I,J),I=1,IE1)
```

```
      DO 70 J=1,JF1
   70 READ(26,900)(HF(I,J),I=1,IF1)
C
  900 FORMAT(20F6.1)
      DO 101 J=1,JD1
      DO 101 I=1,ID1
      IF(HD1(I,J).LT.0.0)THEN ; HD1(I,J)=-90.0
      ELSE ; IF(HD1(I,J).LT.10.0)HD1(I,J)=10.0
      ENDIF
  101 CONTINUE
      DO 102 J=1,JD2
      DO 102 I=1,ID2
      IF(HD2(I,J).LT.0.0)THEN ; HD2(I,J)=-90.0
      ELSE ; IF(HD2(I,J).LT.10.0)HD2(I,J)=10.0
      ENDIF
  102 CONTINUE
      DO 103 J=1,JD3
      DO 103 I=1,ID3
      IF(HD3(I,J).LT.0.0)THEN ; HD3(I,J)=-90.0
      ELSE ; IF(HD3(I,J).LT.10.0)HD3(I,J)=10.0
      ENDIF
  103 CONTINUE
      DO 104 J=1,JD4
      DO 104 I=1,ID4
      IF(HD4(I,J).LT.0.0)THEN ; HD4(I,J)=-90.0
      ELSE ; IF(HD4(I,J).LT.10.0)HD4(I,J)=10.0
      ENDIF
  104 CONTINUE
      DO 105 J=1,JD5
      DO 105 I=1,ID5
      IF(HD5(I,J).LT.0.0)THEN ; HD5(I,J)=-90.0
      ELSE ; IF(HD5(I,J).LT.10.0)HD5(I,J)=10.0
      ENDIF
  105 CONTINUE
C
      DO 110 J=1,JE1
      DO 110 I=1,IE1
      IF(HE(I,J).LT.0.0)THEN ; HE(I,J)=-90.0
      ELSE ; IF(HE(I,J).LT.10.0)HE(I,J)=10.0
      ENDIF
  110 CONTINUE
      DO 120 J=1,JF1
      DO 120 I=1,IF1
      IF(HF(I,J).LT.0.0)THEN ; HF(I,J)=-90.0
      ELSE ; IF(HF(I,J).LT.10.0)HF(I,J)=10.0
      ENDIF
  120 CONTINUE
C
C ****** CAHNGE OF DEPTH DATA ********
C
      CALL CHH (IA,JA,IB,JB,HA,HB,LAB,1101)
      CALL CHH (IB,JB,IC,JC,HB,HC,LBC,0101)
      CALL CHH (IC,JC,ID1,JD1,HC,HD1,LD1,1111)
      CALL CHH (IC,JC,ID2,JD2,HC,HD2,LD2,1101)
      CALL CHH (IC,JC,ID3,JD3,HC,HD3,LD3,0110)
      CALL CHH (IC,JC,ID4,JD4,HC,HD4,LD4,1110)
      CALL CHH (IC,JC,ID5,JD5,HC,HD5,LD5,0110)
      CALL CHH (ID1,JD1,IE1,JE1,HD1,HE,LDE,1011)
      CALL CHH (IE1,JE1,IF1,JF1,HE,HF,LEF,1011)
      RETURN
      END
C
C ******** PARAMETERS FOR VECTOR OPERATION *******
C
      SUBROUTINE PARAME
      PARAMETER (GG=9.8)
      PARAMETER (FO=-60.,FA=38.5,FB=39.7,PP=3.1415926)
      PARAMETER (R=6.37E+6,WW=7.2722E-5)
      PARAMETER (IO=1020,JO=738,IA=76,JA=76)
      PARAMETER (IB=136,JB=136,IC=154,JC=307)
      PARAMETER (DO=10.0,DA=4.0,DB=1.33)
      PARAMETER (TO=20.0,TA=10.0,TB=5.0)
      COMMON /HH/ H(IO,JO)
      COMMON /RR/ R1(IO,JO),R2(IO,JO),R3(IO,JO),
```

```
             &R4(IO,JO),R5(IO,JO),R6(IO)
      COMMON /CC/ C1(IO),C2(IO),C3(IO),C4(IO)
      COMMON /VV/ V1(IO),V2(IO),V3(IO),V4(IO),V5(IO)
      COMMON /HA/ HA(IA,JA)
      COMMON /RA/ RA1(IA,JA),RA2(IA,JA),RA3(IA,JA),
             &RA4(IA,JA),RA5(IA,JA),RA6(IA)
      COMMON /HB/ HB(IB,JB)
      COMMON /RB/ RB1(IB,JB),RB2(IB,JB),RB3(IB,JB),
             &RB4(IB,JB),RB5(IB,JB),RB6(IB)
C
      PX=PP/180.
      DX=R*(DO/60.)*PX
      RR=TO/DX
      RS=0.5*GG*RR
      RT=0.5*TO*WW
      DO 10 J=1,JO
      C1(J)=(FO+(DO*J-0.5*DO)/60.)*PX
      C2(J)=(FO+DO*J/60.)*PX
   10 CONTINUE
      DO 11 J=1,JO
      C3(J)=COS(C1(J))
      V1(J)=RR/C3(J)
      V2(J)=RS/C3(J)
      V3(J)=RT*SIN(C1(J))
      V4(J)=RS
      V5(J)=RT*SIN(C2(J))
      R6(J)=C3(J)
   11 CONTINUE
      DO 12 J=1,JO
      C4(J)=0.
      DO 12 I=1,IO
      IF(H(I,J).GT.C4(J))C4(J)=H(I,J)
   12 CONTINUE
      DO 15 J=1,JO
      DO 15 I=1,IO
      IF(H(I,J).LT.0.)GOTO 15
      R1(I,J)=V1(J)
      IF(I.NE.IO)THEN
      IF(H(I+1,J).LT.0.)GOTO 16
      R2(I,J)=V2(J)*(H(I,J)+H(I+1,J))
      R3(I,J)=V3(J)
      ELSE
      R2(I,J)=V2(J)*H(I,J)*2
      R3(I,J)=V3(J)
      ENDIF
   16 IF(J.NE.JO)THEN
      IF(H(I,J+1).LT.0.)GOTO 15
      R4(I,J)=V4(J)*(H(I,J)+H(I,J+1))
      R5(I,J)=V5(J)
      ELSE
      R4(I,J)=V4(J)*H(I,J)*2
      R5(I,J)=V5(J)
      ENDIF
   15 CONTINUE
      DO 17 I=1,IO
   17 C1(I)=0.
      DO 18 I=1,IO
   18 IF(H(I,1).GT.0.)C1(I)=1./SQRT(GG*H(I,1))
      DO 19 J=1,JO
   19 C2(J)=0.
      DO 20 J=1,JO
   20 IF(H(1,J).GT.0.)C2(J)=1./SQRT(GG*H(1,J))
      DO 21 I=1,IO
   21 C3(I)=0.
      DO 22 I=1,IO
   22 IF(H(I,JO).GT.0.)C3(I)=1./SQRT(GG*H(I,JO))
      DO 23 J=1,JO
   23 C4(J)=0.
      DO 24 J=1,JO
   24 IF(H(IO,J).GT.0.)C4(J)=1./SQRT(GG*H(IO,J))
C
      DX=R*(DA/60.)*PX
      RR=TA/DX
      RS=0.5*GG*RR
```

```
4330    RT=0.5*TA*WW
4340    DO 30 J=1,JA
4350    CC1=(FA+(DA*J-0.5*DA)/60.)*PX
4360    CC2=(FA+DA*J/60.)*PX
4370    CC3=COS(CC1)
4380    V1(J)=RR/CC3
4390    V2(J)=RS/CC3
4400    V3(J)=RT*SIN(CC1)
4410    V4(J)=RS
4420    V5(J)=RT*SIN(CC2)
4430    RA6(J)=CC3
4440  30 CONTINUE
4450    DO 32 J=1,JA
4460    DO 32 I=1,IA
4470    IF(HA(I,J).LT.0.)GOTO 32
4480    RA1(I,J)=V1(J)
4490    IF(I.NE.IA)THEN
4500    IF(HA(I+1,J).LT.0.)GOTO 33
4510    RA2(I,J)=V2(J)*(HA(I,J)+HA(I+1,J))
4520    RA3(I,J)=V3(J)
4530    ELSE
4540    RA2(I,J)=V2(J)*HA(I,J)*2
4550    RA3(I,J)=V3(J)
4560    ENDIF
4570  33 IF(J.NE.JA)THEN
4580    IF(HA(I,J+1).LT.0.)GOTO 32
4590    RA4(I,J)=V4(J)*(HA(I,J)+HA(I,J+1))
4600    RA5(I,J)=V5(J)
4610    ELSE
4620    RA4(I,J)=V4(J)*HA(I,J)*2
4630    RA5(I,J)=V5(J)
4640    ENDIF
4650  32 CONTINUE
4660C
4670    DX=R*(DB/60.)*PX
4680    RR=TB/DX
4690    RS=0.5*GG*RR
4700    RT=0.5*TB*WW
4710    DO 40 J=1,JB
4720    CC1=(FB+(DB*J-0.5*DB)/60.)*PX
4730    CC2=(FB+DB*J/60.)*PX
4740    CC3=COS(CC1)
4750    V1(J)=RR/CC3
4760    V2(J)=RS/CC3
4770    V3(J)=RT*SIN(CC1)
4780    V4(J)=RS
4790    V5(J)=RT*SIN(CC2)
4800    RB6(J)=CC3
4810  40 CONTINUE
4820    DO 42 J=1,JB
4830    DO 42 I=1,IB
4840    IF(HB(I,J).LT.0.)GOTO 42
4850    RB1(I,J)=V1(J)
4860    IF(I.NE.IB)THEN
4870    IF(HB(I+1,J).LT.0.)GOTO 43
4880    RB2(I,J)=V2(J)*(HB(I,J)+HB(I+1,J))
4890    RB3(I,J)=V3(J)
4900    ELSE
4910    RB2(I,J)=V2(J)*HB(I,J)*2
4920    RB3(I,J)=V3(J)
4930    ENDIF
4940  43 IF(J.NE.JB)THEN
4950    IF(HB(I,J+1).LT.0.)GOTO 42
4960    RB4(I,J)=V4(J)*(HB(I,J)+HB(I,J+1))
4970    RB5(I,J)=V5(J)
4980    ELSE
4990    RB4(I,J)=V4(J)*HB(I,J)*2
5000    RB5(I,J)=V5(J)
5010    ENDIF
5020  42 CONTINUE
5030    RETURN ; END
5040C
5050    SUBROUTINE INITIA  (IS,JS,IE,JE)
5060C
```

```
5070     REAL M,N,MA,NA,MB,NB,MC,NC,ME,NE,MF,NF
5080     REAL MD1,ND1,MD2,ND2,MD3,ND3,MD4,ND4,MD5,ND5
5090     PARAMETER (IO=1020,JO=738,IA=76,JA=76)
5100     PARAMETER (IB=136,JB=136,IC=154,JC=307)
5110     PARAMETER (ID1=28,JD1=55,ID2=55,JD2=40)
5120     PARAMETER (ID3=37,JD3=31,ID4=13,JD4=40)
5130     PARAMETER (ID5=46,JD5=40)
5140     PARAMETER (IE1=34,JE1=67,IF1=46,JF1=58)
5150     COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
5160     COMMON /HH/ H(IO,JO)
5170     COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
5180     COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
5190     COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
5200     COMMON /XD1/ ZD1(ID1,JD1,2),MD1(ID1,JD1,2),
5210              &ND1(ID1,JD1,2),HD1(ID1,JD1)
5220     COMMON /XD2/ ZD2(ID2,JD2,2),MD2(ID2,JD2,2),
5230              &ND2(ID2,JD2,2),HD2(ID2,JD2)
5240     COMMON /XD3/ ZD3(ID3,JD3,2),MD3(ID3,JD3,2),
5250              &ND3(ID3,JD3,2),HD3(ID3,JD3)
5260     COMMON /XD4/ ZD4(ID4,JD4,2),MD4(ID4,JD4,2),
5270              &ND4(ID4,JD4,2),HD4(ID4,JD4)
5280     COMMON /XD5/ ZD5(ID5,JD5,2),MD5(ID5,JD5,2),
5290              &ND5(ID5,JD5,2),HD5(ID5,JD5)
5300     COMMON /XE/  ZE(IE1,JE1,2),ME(IE1,JE1,2),
5310              &NE(IE1,JE1,2),HE(IE1,JE1)
5320     COMMON /XF/  ZF(IF1,JF1,2),MF(IF1,JF1,2),
5330              &NF(IF1,JF1,2),HF(IF1,JF1)
5340C
5350     DO 200 L=1,2
5360     DO 10 J=1,JO ; DO 10 I=1,IO
5370     Z(I,J,L)=0.0 ; M(I,J,L)=0.0 ; N(I,J,L)=0.0
5380   10 CONTINUE
5390     DO 20 J=1,JA ; DO 20 I=1,IA
5400     ZA(I,J,2)=0.0 ; MA(I,J,L)=0.0 ; NA(I,J,L)=0.0
5410   20 CONTINUE
5420     DO 30 J=1,JB ; DO 30 I=1,IB
5430     ZB(I,J,L)=0.0 ; MB(I,J,L)=0.0 ; NB(I,J,L)=0.0
5440   30 CONTINUE
5450     DO 40 J=1,JC ; DO 40 I=1,IC
5460     ZC(I,J,L)=0.0 ; MC(I,J,L)=0.0 ; NC(I,J,L)=0.0
5470   40 CONTINUE
5480     DO 51 J=1,JD1 ; DO 51 I=1,ID1
5490     ZD1(I,J,L)=0. ; MD1(I,J,L)=0. ; ND1(I,J,L)=0.
5500   51 CONTINUE
5510     DO 52 J=1,JD2 ; DO 52 I=1,ID2
5520     ZD2(I,J,L)=0. ; MD2(I,J,L)=0. ; ND2(I,J,L)=0.
5530   52 CONTINUE
5540     DO 53 J=1,JD3 ; DO 53 I=1,ID3
5550     ZD3(I,J,L)=0. ; MD3(I,J,L)=0. ; ND3(I,J,L)=0.
5560   53 CONTINUE
5570     DO 54 J=1,JD4 ; DO 54 I=1,ID4
5580     ZD4(I,J,L)=0. ; MD4(I,J,L)=0. ; ND4(I,J,L)=0.
5590   54 CONTINUE
5600     DO 55 J=1,JD5 ; DO 55 I=1,ID5
5610     ZD5(I,J,L)=0. ; MD5(I,J,L)=0. ; ND5(I,J,L)=0.
5620   55 CONTINUE
5630     DO 60 J=1,JE1 ; DO 60 I=1,IE1
5640     ZE(I,J,L)=0. ; ME(I,J,L)=0. ; NE(I,J,L)=0.
5650   60 CONTINUE
5660     DO 70 J=1,JF1 ; DO 70 I=1,IF1
5670     ZF(I,J,L)=0. ; MF(I,J,L)=0. ; NF(I,J,L)=0.
5680   70 CONTINUE
5690  200 CONTINUE
5700C
5710     CALL DEFORM (IO,JO,IS,JS,IE,JE,Z)
5720     DO 110 J=JS,JE ; DO 110 I=IS,IE
5730     IF(H(I,J).LT.0.)Z(I,J,1)=0.
5740  110 CONTINUE
5750     RETURN
5760     END
5770C
5780     SUBROUTINE CHH (IX,JX,IY,JY,HX,HY,L0,BCHK)
5790C
5800     INTEGER BCHK,CHK
```

```
      DIMENSION HX(IX,JX),HY(IY,JY),L0(4)
C
      ISS=2   ; JSS=2   ; IES=IY   ; JES=JY
      ISL=L0(1) ; JSL=L0(2) ; IEL=L0(3) ; JEL=L0(4)
      CHK=BCHK
      KB=CHK/1000
      IF(KB.EQ.1)THEN
      CHK=CHK-1000
      II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
      DO WHILE (I1+2.LE.IES)
      S=0.0 ; L=0
      DO 10 I=I1,I1+2
      DO 10 J=J1,J1+2
      IF(HY(I,J).GT.0.0)THEN
      S=S+HY(I,J)
      L=L+1 ; ENDIF
   10 CONTINUE
      IF(L.GE.5)THEN
      HX(II,JJ)=S/L  ; ENDIF
      II=II+1 ; I1=I1+3
      ENDDO   ; ENDIF
C
      KB=CHK/100
      IF(KB.EQ.1)THEN
      CHK=CHK-100
      II=IEL ; JJ=JSL ; I1=IES-2 ; J1=JSS
      DO WHILE (J1+1.LE.JES)
      S=0.0 ; L=0
      DO 20 I=I1,I1+2
      DO 20 J=J1,J1+2
      IF(HY(I,J).GT.0.0)THEN
      S=S+HY(I,J)
      L=L+1 ; ENDIF
   20 CONTINUE
      IF(L.GE.5)THEN
      HX(II,JJ)=S/L  ; ENDIF
      JJ=JJ+1 ; J1=J1+3
      ENDDO   ; ENDIF
C
      KB=CHK/10
      IF(KB.EQ.1)THEN
      CHK=CHK-10
      II=ISL ; JJ=JEL ; I1=ISS ; J1=JES-2
      DO WHILE (I1+2.LE.IES)
      S=0.0 ; L=0
      DO 30 I=I1,I1+2
      DO 30 J=J1,J1+2
      IF(HY(I,J).GT.0.0)THEN
      S=S+HY(I,J)
      L=L+1 ; ENDIF
   30 CONTINUE
      IF(L.GE.5)THEN
      HX(II,JJ)=S/L  ; ENDIF
      II=II+1 ; I1=I1+3
      ENDDO   ; ENDIF
C
      IF(CHK.EQ.1)THEN
      II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
      DO WHILE (J1+2.LE.JES)
      S=0.0 ; L=0
      DO 40 I=I1,I1+2
      DO 40 J=J1,J1+2
      IF(HY(I,J).GT.0.0)THEN
      S=S+HY(I,J)
      L=L+1 ; ENDIF
   40 CONTINUE
      IF(L.GE.5)THEN
      HX(II,JJ)=S/L  ; ENDIF
      JJ=JJ+1 ; J1=J1+3
      ENDDO  ; ENDIF
      RETURN ; END
C
      SUBROUTINE LIMIT (IS,JS,IE,JE)
C
```

```
6550     REAL M,N
6560     PARAMETER (GX=1.E-4)
6570     PARAMETER (IO=1020,JO=738)
6580     COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
6590     IF(IS.EQ.2)GOTO 61
6600     L=0
6610     DO 10 J=JS,JE
6620     IF(ABS(Z(IS+2,J,2)).GT.GX)L=1
6630  10 CONTINUE
6640     IF(L.EQ.1)THEN
6650     IS=IS-1
6660     IF(IS.LE.2)IS=2
6670     ENDIF
6680  61 IF(IE.EQ.IO-1)GOTO 62
6690     L=0
6700     DO 20 J=JS,JE
6710     IF(ABS(Z(IE-2,J,2)).GT.GX)L=1
6720  20 CONTINUE
6730     IF(L.EQ.1)THEN
6740     IE=IE+1
6750     IF(IE.GE.IO-1)IE=IO-1
6760     ENDIF
6770  62 IF(JS.EQ.2)GOTO 63
6780     L=0
6790     DO 30 I=IS,IE
6800     IF(ABS(Z(I,JS+2,2)).GT.GX)L=1
6810  30 CONTINUE
6820     IF(L.EQ.1)THEN
6830     JS=JS-1
6840     IF(JS.LE.2)JS=2
6850     ENDIF
6860  63 IF(JE.EQ.JO-1)GOTO 64
6870     L=0
6880     DO 40 I=IS,IE
6890     IF(ABS(Z(I,JE-2,2)).GT.GX)L=1
6900  40 CONTINUE
6910     IF(L.EQ.1)THEN
6920     JE=JE+1
6930     IF(JE.GE.JO-1)JE=JO-1
6940     ENDIF
6950  64 RETURN
6960     END
6970C
6980     SUBROUTINE MASS1 (IS,JS,IE,JE)
6990C
7000     REAL M,N
7010     PARAMETER (IO=1020,JO=738)
7020     COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
7030     COMMON /HH/ H(IO,JO)
7040     COMMON /RR/ R1(IO,JO),R2(IO,JO),R3(IO,JO),
7050              &R4(IO,JO),R5(IO,JO),R6(IO)
7060     DO 10 J=JS,JE
7070     DO 10 I=IS,IE
7080     Z(I,J,2)=Z(I,J,1)-R1(I,J)*(M(I,J,1)-M(I-1,J,1))
7090     &  -R1(I,J)*(N(I,J,1)*R6(J)-N(I,J-1,1)*R6(J-1))
7100  10 CONTINUE
7110     DO 20 J=JS,JE
7120     DO 20 I=IS,IE
7130     IF(ABS(Z(I,J,2)).LT.1.0E-5)Z(I,J,2)=0.0
7140  20 CONTINUE
7150     RETURN
7160     END
7170C
7180     SUBROUTINE MASS2 (IF,JF,Z,M,N,R1,R6,LP,LT)
7190C
7200     REAL M,N
7210     DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
7220     DIMENSION R1(IF,JF),R6(IF)
7230     IF(MOD(LP,LT).NE.0)RETURN
7240     DO 10 J=2,JF
7250     DO 10 I=2,IF
7260     Z(I,J,2)=Z(I,J,1)-R1(I,J)*(M(I,J,1)-M(I-1,J,1))
7270     &  -R1(I,J)*(N(I,J,1)*R6(J)-N(I,J-1,1)*R6(J-1))
7280  10 CONTINUE
```

```
      DO 20 J=2,JF
      DO 20 I=2,IF
      IF(ABS(Z(I,J,2)).LT.1.0E-5)Z(I,J,2)=0.0
   20 CONTINUE
      RETURN
      END
C
      SUBROUTINE MASS3 (IF,JF,Z,M,N,H,DX,DT,LP,LT)
C
      REAL M,N
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
      DIMENSION H(IF,JF)
      IF(MOD(LP,LT).NE.0)RETURN
      DO 10 J=2,JF
      DO 10 I=2,IF
      IF(H(I,J).GT.0.0)THEN
      Z(I,J,2)=Z(I,J,1)-DT/DX*(M(I,J,1)-M(I-1,J,1)
      &                 +N(I,J,1)-N(I,J-1,1))
      ELSE
      Z(I,J,2)=0.0
      ENDIF
   10 CONTINUE
      DO 20 J=2,JF
      DO 20 I=2,IF
      IF(ABS(Z(I,J,2)).LT.1.0E-5)Z(I,J,2)=0.0
   20 CONTINUE
      RETURN ; END
C
      SUBROUTINE MMNT1 (IS,JS,IE,JE)
C
      REAL M,N
      PARAMETER (IO=1020,JO=738)
      COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
      COMMON /HH/ H(IO,JO)
      COMMON /RR/ R1(IO,JO),R2(IO,JO),R3(IO,JO),
             &R4(IO,JO),R5(IO,JO),R6(IO)
      COMMON /VV/ V1(IO),V2(IO),V3(IO),V4(IO),V5(IO)
      DO 10 J=JS,JE
      DO 10 I=IS,IE
      V1(I)=Z(I+1,J,2)-Z(I,J,2)
      V2(I)=N(I,J-1,1)+N(I,J,1)+N(I+1,J-1,1)+N(I+1,J,1)
      M(I,J,2)=M(I,J,1)-R2(I,J)*V1(I)+R3(I,J)*V2(I)
   10 CONTINUE
      IF(JS.LE.2)THEN
      DO 15 I=1,IO-1
   15 M(I,1,2)=M(I,1,1)-R2(I,1)*(Z(I+1,1,2)-Z(I,1,2))
      ENDIF
      IF(JE.GE.JO-1)THEN
      DO 16 I=1,IO-1
   16 M(I,JO,2)=M(I,JO,1)-R2(I,JO)*(Z(I+1,JO,2)-Z(I,JO,2))
      ENDIF
      IF(IS.LE.2)THEN
      DO 17 J=1,JO
   17 M(1,J,2)=M(1,J,1)-R2(1,J)*(Z(2,J,2)-Z(1,J,2))
      ENDIF
C
      DO 20 J=JS,JE
      DO 20 I=IS,IE
      V1(I)=Z(I,J+1,2)-Z(I,J,2)
      V2(I)=M(I-1,J,1)+M(I,J,1)+M(I-1,J+1,1)+M(I,J+1,1)
      N(I,J,2)=N(I,J,1)-R4(I,J)*V1(I)-R5(I,J)*V2(I)
   20 CONTINUE
      IF(IS.LE.2)THEN
      DO 25 J=1,JO-1
   25 N(1,J,2)=N(1,J,1)-R4(1,J)*(Z(1,J+1,2)-Z(1,J,2))
      ENDIF
      IF(IE.GE.IO-1)THEN
      DO 26 J=1,JO-1
   26 N(IO,J,2)=N(IO,J,1)-R4(IO,J)*(Z(IO,J+1,2)-Z(IO,J,2))
      ENDIF
      IF(JS.LE.2)THEN
      DO 27 I=1,IO
   27 N(I,1,2)=N(I,1,1)-R4(I,1)*(Z(I,2,2)-Z(I,1,2))
      ENDIF
```

```
8030      RETURN ; END
8040C
8050      SUBROUTINE MMNT2 (IF,JF,Z,M,N,R2,R3,R4,R5,LP,LT)
8060C
8070      REAL M,N
8080      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
8090      DIMENSION R2(IF,JF),R3(IF,JF),R4(IF,JF),
8100          &R5(IF,JF)
8110      IF(MOD(LP,LT).NE.0)RETURN
8120      DO 10 J=2,JF-1
8130      DO 10 I=2,IF-1
8140      M(I,J,2)=M(I,J,1)-R2(I,J)*(Z(I+1,J,2)-Z(I,J,2))
8150     &+R3(I,J)*(N(I,J-1,1)+N(I,J,1)+N(I+1,J-1,1)+N(I+1,J,1))
8160      N(I,J,2)=N(I,J,1)-R4(I,J)*(Z(I,J+1,2)-Z(I,J,2))
8170     &+R5(I,J)*(M(I-1,J,1)+M(I,J,1)+M(I-1,J+1,1)+M(I,J+1,1))
8180   10 CONTINUE
8190C
8200      DO 16 I=1,IF-1
8210   16 M(I,JF,2)=M(I,JF,1)-R2(I,JF)*(Z(I+1,JF,2)-Z(I,JF,2))
8220      DO 26 J=1,JF-1
8230   26 N(IF,J,2)=N(IF,J,1)-R4(IF,J)*(Z(IF,J+1,2)-Z(IF,J,2))
8240      RETURN  ; END
8250C
8260      SUBROUTINE MMNT3 (IF,JF,Z,M,N,H,DX,DT,LP,LT)
8270C
8280      REAL M,N
8290      PARAMETER (GG=9.8)
8300      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
8310      DIMENSION H(IF,JF)
8320      IF(MOD(LP,LT).NE.0)RETURN
8330      DO 10 J=2,JF-1
8340      DO 10 I=2,IF-1
8350      HH=0.5*(H(I,J)+H(I+1,J))*GG*DT/DX
8360      IF(HH.GT.0.0)THEN
8370      M(I,J,2)=M(I,J,1)-HH*(Z(I+1,J,2)-Z(I,J,2))
8380      ELSE ; M(I,J,2)=0.0 ; ENDIF
8390   10 CONTINUE
8400      DO 20 J=2,JF-1
8410      DO 20 I=2,IF-1
8420      HH=0.5*(H(I,J)+H(I,J+1))*GG*DT/DX
8430      IF(HH.GT.0.0)THEN
8440      N(I,J,2)=N(I,J,1)-HH*(Z(I,J+1,2)-Z(I,J,2))
8450      ELSE ; N(I,J,2)=0.0 ; ENDIF
8460   20 CONTINUE
8470C
8480      DO 16 I=1,IF-1
8490      HH=0.5*(H(I,JF)+H(I+1,JF))*GG*DT/DX
8500      IF(HH.GT.0.0)THEN
8510      M(I,JF,2)=M(I,JF,1)-HH*(Z(I+1,JF,2)-Z(I,JF,2))
8520      ELSE ; M(I,JF,2)=0.0 ; ENDIF
8530   16 CONTINUE
8540      DO 26 J=1,JF-1
8550      HH=0.5*(H(IF,J)+H(IF,J+1))*GG*DT/DX
8560      IF(HH.GT.0.0)THEN
8570      N(IF,J,2)=N(IF,J,1)-HH*(Z(IF,J+1,2)-Z(IF,J,2))
8580      ELSE ; N(IF,J,2)=0.0 ; ENDIF
8590   26 CONTINUE
8600      RETURN  ; END
8610C
8620      SUBROUTINE BOUND (IS,JS,IE,JE)
8630C
8640      REAL M,N
8650      PARAMETER (IO=1020,JO=738)
8660      COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
8670      COMMON /HH/ H(IO,JO)
8680      COMMON /CC/ C1(IO),C2(IO),C3(IO),C4(IO)
8690      IF(JS.LE.2)THEN
8700      DO 10 I=2,IO-1
8710      Z(I,1,2)=SQRT(N(I,1,1)**2
8720         &+0.25*(M(I,1,1)+M(I-1,1,1))**2)*C1(I)
8730   10 IF(N(I,1,1).GT.0.)Z(I,1,2)=-Z(I,1,2)
8740      ENDIF
8750      IF(IS.LE.2)THEN
8760      DO 20 J=2,JO-1
```

```
8770     Z(1,J,2)=SQRT(M(1,J,1)**2
8780        &+0.25*(N(1,J,1)+N(1,J-1,1))**2)*C2(J)
8790     IF(Z(1,J,2).GT.1.0)Z(1,J,2)=1.0
8800     IF(Z(1,J,2).LT.-1.0)Z(1,J,2)=-1.0
8810   20 IF(M(1,J,1).GT.0.)Z(1,J,2)=-Z(1,J,2)
8820     ENDIF
8830     IF(JE.GE.JO-1)THEN
8840     DO 30 I=2,IO-1
8850     Z(I,JO,2)=SQRT(N(I,JO-1,1)**2
8860        &+0.25*(M(I,JO,1)+M(I,JO-1,1))**2)*C3(I)
8870   30 IF(N(I,JO-1,1).LT.0.)Z(I,JO,2)=-Z(I,JO,2)
8880     ENDIF
8890     IF(IE.GE.IO-1)THEN
8900     DO 40 J=2,JO-1
8910     Z(IO,J,2)=SQRT(M(IO-1,J,1)**2
8920        &+0.25*(N(I,JO,1)+N(I,JO-1,1))**2)*C4(J)
8930   40 IF(M(IO-1,J,1).LT.0.)Z(IO,J,2)=-Z(IO,J,2)
8940     ENDIF
8950     Z(1,1,2)=SQRT(M(1,1,1)**2+N(1,1,1)**2)*C1(1)
8960     IF(N(1,1,1).GT.0.0)Z(1,1,2)=-Z(1,1,2)
8970     Z(IO,1,2)=SQRT(M(IO-1,1,1)**2+N(IO,1,1)**2)*C1(IO)
8980     IF(N(IO,1,1).GT.0.0)Z(IO,1,2)=-Z(IO,1,2)
8990     Z(1,JO,2)=SQRT(M(1,JO,1)**2+N(1,JO-1,1)**2)*C3(1)
9000     IF(N(1,JO-1,1).LT.0.0)Z(1,JO,2)=-Z(1,JO,2)
9010     Z(IO,JO,2)=SQRT(M(IO-1,JO,1)**2+N(IO,JO-1,1)**2)*C3(JO)
9020     IF(N(IO,JO-1,1).LT.0.0)Z(IO,JO,2)=-Z(IO,JO,2)
9030     RETURN
9040     END
9050C
9060     SUBROUTINE JNZOA (LP,LT)
9070C
9080     REAL M,MA,N,NA
9090     PARAMETER (IO=1020,JO=738,IA=76,JA=76)
9100     PARAMETER (ISS=2,JSS=2,IES=76,JES=76)
9110     PARAMETER (ISL=121,JSL=592,IEL=150,JEL=621)
9120     COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
9130     COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
9140     COMMON /HA/ HA(IA,JA)
9150     IF(MOD(LP,LT).NE.0)RETURN
9160C
9170     II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
9180     DO WHILE (II.LE.IEL)
9190     S=0.0 ; L=0
9200     IE=I1+2 ; IF(IE.GE.IES)IE=IES
9210     DO 10 I=I1,IE
9220     DO 10 J=J1,J1+2
9230     IF(HA(I,J).GT.0.0)THEN
9240     S=S+0.5*(ZA(I,J,1)+ZA(I,J,2))
9250     L=L+1 ; ENDIF
9260    10 CONTINUE
9270     IF(L.GE.1)THEN
9280     Z(II,JJ,2)=S/L  ; ELSE
9290     Z(II,JJ,2)=0.0  ; ENDIF
9300     II=II+1 ; I1=I1+2
9310     IF(MOD(II,2).EQ.1)I1=I1+1
9320     ENDDO
9330C
9340     II=IEL ; JJ=JSL ; I1=IES-2 ; J1=JSS
9350     DO WHILE (JJ.LE.JEL)
9360     S=0.0 ; L=0
9370     JE=J1+2 ; IF(JE.GE.JES)JE=JES
9380     DO 20 I=I1,I1+2
9390     DO 20 J=J1,JE
9400     IF(HA(I,J).GT.0.0)THEN
9410     S=S+0.5*(ZA(I,J,1)+ZA(I,J,2))
9420     L=L+1 ; ENDIF
9430    20 CONTINUE
9440     IF(L.GE.1)THEN
9450     Z(II,JJ,2)=S/L  ; ELSE
9460     Z(II,JJ,2)=0.0  ; ENDIF
9470     JJ=JJ+1 ; J1=J1+2
9480     IF(MOD(JJ,2).EQ.0)J1=J1+1
9490     ENDDO
9500C
```

```
9510    II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
9520    DO WHILE (JJ.LE.JEL)
9530    S=0.0 ; L=0
9540    JE=J1+2 ; IF(JE.GE.JES)JE=JES
9550    DO 30 I=I1,I1+2
9560    DO 30 J=J1,JE
9570    IF(HA(I,J).GT.0.0)THEN
9580    S=S+0.5*(ZA(I,J,1)+ZA(I,J,2))
9590    L=L+1 ; ENDIF
9600  30 CONTINUE
9610    IF(L.GE.1)THEN
9620    Z(II,JJ,2)=S/L  ; ELSE
9630    Z(II,JJ,2)=0.0  ; ENDIF
9640    JJ=JJ+1 ; J1=J1+2
9650    IF(MOD(JJ,2).EQ.0)J1=J1+1
9660    ENDDO
9670    RETURN ; END
9680C
9690    SUBROUTINE JNZ (IX,JX,IY,JY,ZX,ZY,HY
9700            &,L0,BCHK,LP,LT)
9710C
9720    INTEGER BCHK,CHK
9730    DIMENSION ZX(IX,JX,2),ZY(IY,JY,2)
9740    DIMENSION HY(IY,JY),L0(4)
9750    IF(MOD(LP,LT).NE.0)RETURN
9760    ISS=2    ; JSS=2    ; IES=IY    ; JES=JY
9770    ISL=L0(1) ; JSL=L0(2) ; IEL=L0(3) ; JEL=L0(4)
9780    CHK=BCHK
9790    KB=CHK/1000
9800    IF(KB.EQ.1)THEN
9810    CHK=CHK-1000
9820    II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
9830    DO WHILE (I1+2.LE.IES)
9840    S=0.0 ; L=0
9850    DO 10 I=I1,I1+2
9860    DO 10 J=J1,J1+2
9870    IF(HY(I,J).GT.0.0)THEN
9880    S=S+0.5*(ZY(I,J,1)+ZY(I,J,2))
9890    L=L+1 ; ENDIF
9900  10 CONTINUE
9910    IF(L.GE.5)THEN
9920    ZX(II,JJ,2)=S/L  ; ELSE
9930    ZX(II,JJ,2)=0.0  ; ENDIF
9940    II=II+1 ; I1=I1+3
9950    ENDDO   ; ENDIF
9960C
9970    KB=CHK/100
9980    IF(KB.EQ.1)THEN
9990    CHK=CHK-100
10000    II=IEL ; JJ=JSL ; I1=IES-2 ; J1=JSS
10010    DO WHILE (J1+1.LE.JES)
10020    S=0.0 ; L=0
10030    DO 20 I=I1,I1+2
10040    DO 20 J=J1,J1+2
10050    IF(HY(I,J).GT.0.0)THEN
10060    S=S+0.5*(ZY(I,J,1)+ZY(I,J,2))
10070    L=L+1 ; ENDIF
10080  20 CONTINUE
10090    IF(L.GE.5)THEN
10100    ZX(II,JJ,2)=S/L  ; ELSE
10110    ZX(II,JJ,2)=0.0  ; ENDIF
10120    JJ=JJ+1 ; J1=J1+3
10130    ENDDO   ; ENDIF
10140C
10150    KB=CHK/10
10160    IF(KB.EQ.1)THEN
10170    CHK=CHK-10
10180    II=ISL ; JJ=JEL ; I1=ISS ; J1=JES-2
10190    DO WHILE (I1+2.LE.IES)
10200    S=0.0 ; L=0
10210    DO 30 I=I1,I1+2
10220    DO 30 J=J1,J1+2
10230    IF(HY(I,J).GT.0.0)THEN
10240    S=S+0.5*(ZY(I,J,1)+ZY(I,J,2))
```

```
      L=L+1 ; ENDIF
  30 CONTINUE
      IF(L.GE.5)THEN
      ZX(II,JJ,2)=S/L  ; ELSE
      ZX(II,JJ,2)=0.0  ; ENDIF
      II=II+1 ; I1=I1+3
      ENDDO   ; ENDIF
C
      IF(CHK.EQ.1)THEN
      II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
      DO WHILE (J1+2.LE.JES)
      S=0.0 ; L=0
      DO 40 I=I1,I1+2
      DO 40 J=J1,J1+2
      IF(HY(I,J).GT.0.0)THEN
      S=S+0.5*(ZY(I,J,1)+ZY(I,J,2))
      L=L+1 ; ENDIF
  40 CONTINUE
      IF(L.GE.5)THEN
      ZX(II,JJ,2)=S/L  ; ELSE
      ZX(II,JJ,2)=0.0  ; ENDIF
      JJ=JJ+1 ; J1=J1+3
      ENDDO  ; ENDIF
      RETURN ; END
C
      SUBROUTINE J2Z (IX,JX,IY,JY,ZX,ZY,HY
              &,L0,BCHK,LP,LT)
C
      INTEGER BCHK,CHK
      DIMENSION ZX(IX,JX,2),ZY(IY,JY,2)
      DIMENSION HY(IY,JY),L0(4)
      IF(MOD(LP,LT).NE.0)RETURN
      ISS=2    ; JSS=2    ; IES=IY   ; JES=JY
      ISL=L0(1) ; JSL=L0(2) ; IEL=L0(3) ; JEL=L0(4)
      CHK=BCHK
      KB=CHK/1000
      IF(KB.EQ.1)THEN
      CHK=CHK-1000
      II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
      DO WHILE (I1+2.LE.IES)
      S=0.0 ; L=0
      DO 10 I=I1,I1+2
      DO 10 J=J1,J1+2
      IF(HY(I,J).GT.0.0)THEN
      S=S+ZY(I,J,2)
      L=L+1 ; ENDIF
  10 CONTINUE
      IF(L.GE.5)THEN
      ZX(II,JJ,2)=S/L  ; ELSE
      ZX(II,JJ,2)=0.0  ; ENDIF
      II=II+1 ; I1=I1+3
      ENDDO   ; ENDIF
C
      KB=CHK/100
      IF(KB.EQ.1)THEN
      CHK=CHK-100
      II=IEL ; JJ=JSL ; I1=IES-2 ; J1=JSS
      DO WHILE (J1+1.LE.JES)
      S=0.0 ; L=0
      DO 20 I=I1,I1+2
      DO 20 J=J1,J1+2
      IF(HY(I,J).GT.0.0)THEN
      S=S+ZY(I,J,2)
      L=L+1 ; ENDIF
  20 CONTINUE
      IF(L.GE.5)THEN
      ZX(II,JJ,2)=S/L  ; ELSE
      ZX(II,JJ,2)=0.0  ; ENDIF
      JJ=JJ+1 ; J1=J1+3
      ENDDO   ; ENDIF
C
      KB=CHK/10
      IF(KB.EQ.1)THEN
      CHK=CHK-10
```

```
      II=ISL ; JJ=JEL ; I1=ISS ; J1=JES-2
      DO WHILE (I1+2.LE.IES)
      S=0.0 ; L=0
      DO 30 I=I1,I1+2
      DO 30 J=J1,J1+2
      IF(HY(I,J).GT.0.0)THEN
      S=S+ZY(I,J,2)
      L=L+1 ; ENDIF
   30 CONTINUE
      IF(L.GE.5)THEN
      ZX(II,JJ,2)=S/L  ; ELSE
      ZX(II,JJ,2)=0.0  ; ENDIF
      II=II+1 ; I1=I1+3
      ENDDO  ; ENDIF
C
      IF(CHK.EQ.1)THEN
      II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
      DO WHILE (J1+2.LE.JES)
      S=0.0 ; L=0
      DO 40 I=I1,I1+2
      DO 40 J=J1,J1+2
      IF(HY(I,J).GT.0.0)THEN
      S=S+ZY(I,J,2)
      L=L+1 ; ENDIF
   40 CONTINUE
      IF(L.GE.5)THEN
      ZX(II,JJ,2)=S/L  ; ELSE
      ZX(II,JJ,2)=0.0  ; ENDIF
      JJ=JJ+1 ; J1=J1+3
      ENDDO  ; ENDIF
      RETURN ; END
C
      SUBROUTINE JNQOA
C
      REAL M,MA,N,NA
      PARAMETER (IO=1020,JO=738,IA=76,JA=76)
      PARAMETER (DO=10.0,DA=4.0)
      PARAMETER (ISS=2,JSS=2,IES=76,JES=76)
      PARAMETER (ISL=121,JSL=592,IEL=150,JEL=621)
      COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
      COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
      I=ISS ; J=JSS-1 ; II=ISL-1 ; JJ=JSL-1
      DO WHILE(I.LE.IES)
      SI=(I-ISS+1.75)*DA/DO
      IS=IFIX(SI) ; DI=SI-IS ; II=IS+ISL-1
      NA(I,J,2)=(1-DI)*N(II,JJ,2)+DI*N(II+1,JJ,2)
      I=I+1
      ENDDO
C
      I=IES ; J=JSS ; II=IEL ; JJ=JSL-1
      DO WHILE(J.LE.JES)
      SJ=(J-JSS+1.75)*DA/DO
      JS=IFIX(SJ) ; DJ=SJ-JS ; JJ=JS+JSL-1
      MA(I,J,2)=(1-DJ)*M(II,JJ,2)+DJ*M(II,JJ+1,2)
      J=J+1
      ENDDO
C
      I=ISS-1 ; J=JSS ; II=ISL-1 ; JJ=JSL-1
      DO WHILE(J.LE.JES)
      SJ=(J-JSS+1.75)*DA/DO
      JS=IFIX(SJ) ; DJ=SJ-JS ; JJ=JS+JSL-1
      MA(I,J,2)=(1-DJ)*M(II,JJ,2)+DJ*M(II,JJ+1,2)
      J=J+1
      ENDDO
      RETURN ; END
C
      SUBROUTINE JNQ (IX,JX,IY,JY,MX,NX,MY,NY
               &,HY,L0,BCHK,LP,LT)
C
      INTEGER BCHK,CHK
      REAL MX,NX,MY,NY
      DIMENSION MX(IX,JX,2),NX(IX,JX,2),HY(IY,JY)
      DIMENSION MY(IY,JY,2),NY(IY,JY,2),L0(4)
      IF(MOD(LP,LT).NE.0)RETURN
```

```
      ISS=2   ; JSS=2   ; IES=IY   ; JES=JY
      ISL=L0(1) ; JSL=L0(2) ; IEL=L0(3) ; JEL=L0(4)
      CHK=BCHK
      KB=CHK/1000
      IF(KB.EQ.1)THEN
      CHK=CHK-1000
      I=ISS ; J=JSS-1 ; II=ISL-1 ; JJ=JSL-1
      DO WHILE(I.LE.IES)
      SI=(I-ISS+2)/3.0
      IS=IFIX(SI) ; DI=SI-IS ; II=IS+ISL-1
      NY(I,J,2)=(1-DI)*NX(II,JJ,2)+DI*NX(II+1,JJ,2)
      IF(HY(I,J).LT.0.0)NY(I,J,2)=0.0
      I=I+1
      ENDDO ; ENDIF
C
      KB=CHK/100
      IF(KB.EQ.1)THEN
      CHK=CHK-100
      I=IES ; J=JSS ; II=IEL ; JJ=JSL-1
      DO WHILE(J.LE.JES)
      SJ=(J-JSS+2)/3.0
      JS=IFIX(SJ) ; DJ=SJ-JS ; JJ=JS+JSL-1
      MY(I,J,2)=(1-DJ)*MX(II,JJ,2)+DJ*MX(II,JJ+1,2)
      IF(HY(I,J).LT.0.0)MY(I,J,2)=0.0
      J=J+1
      ENDDO ; ENDIF
C
      KB=CHK/10
      IF(KB.EQ.1)THEN
      CHK=CHK-10
      I=ISS ; J=JES ; II=ISL-1 ; JJ=JEL
      DO WHILE(I.LE.IES)
      SI=(I-ISS+2)/3.0
      IS=IFIX(SI) ; DI=SI-IS ; II=IS+ISL-1
      NY(I,J,2)=(1-DI)*NX(II,JJ,2)+DI*NX(II+1,JJ,2)
      IF(HY(I,J).LT.0.0)NY(I,J,2)=0.0
      I=I+1
      ENDDO ; ENDIF
C
      IF(CHK.EQ.1)THEN
      I=ISS-1 ; J=JSS ; II=ISL-1 ; JJ=JSL-1
      DO WHILE(J.LE.JES)
      SJ=(J-JSS+2)/3.0
      JS=IFIX(SJ) ; DJ=SJ-JS ; JJ=JS+JSL-1
      MY(I,J,2)=(1-DJ)*MX(II,JJ,2)+DJ*MX(II,JJ+1,2)
      IF(HY(I,J).LT.0.0)MY(I,J,2)=0.0
      J=J+1
      ENDDO ; ENDIF
      RETURN ; END
C
      SUBROUTINE COUT (KK,KD)
C
      REAL M,N,MA,NA,MB,NB,MC,NC,ME,NE,MF,NF
      REAL MD1,ND1,MD2,ND2,MD3,ND3,MD4,ND4,MD5,ND5
      PARAMETER (IO=1020,JO=738,IA=76,JA=76)
      PARAMETER (IB=136,JB=136,IC=154,JC=307)
      PARAMETER (ID1=28,JD1=55,ID2=55,JD2=40)
      PARAMETER (ID3=37,JD3=31,ID4=13,JD4=40)
      PARAMETER (ID5=46,JD5=40)
      PARAMETER (IE1=34,JE1=67,IF1=46,JF1=58)
      COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
      COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
      COMMON /HA/ HA(IA,JA)
      COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
      COMMON /HB/ HB(IB,JB)
      COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
      COMMON /HC/ HC(IC,JC)
      COMMON /XD1/ ZD1(ID1,JD1,2),MD1(ID1,JD1,2),
     &ND1(ID1,JD1,2),HD1(ID1,JD1)
      COMMON /XD2/ ZD2(ID2,JD2,2),MD2(ID2,JD2,2),
     &ND2(ID2,JD2,2),HD2(ID2,JD2)
      COMMON /XD3/ ZD3(ID3,JD3,2),MD3(ID3,JD3,2),
     &ND3(ID3,JD3,2),HD3(ID3,JD3)
      COMMON /XD4/ ZD4(ID4,JD4,2),MD4(ID4,JD4,2),
```

```
          &ND4(ID4,JD4,2),HD4(ID4,JD4)
      COMMON /XD5/ ZD5(ID5,JD5,2),MD5(ID5,JD5,2),
          &ND5(ID5,JD5,2),HD5(ID5,JD5)
      COMMON /XE/  ZE(IE1,JE1,2),ME(IE1,JE1,2),
          &NE(IE1,JE1,2),HE(IE1,JE1)
      COMMON /XF/  ZF(IF1,JF1,2),MF(IF1,JF1,2),
          &NF(IF1,JF1,2),HF(IF1,JF1)
      IF(KK.LT.180*26)RETURN
C       CALL OUTPUT1 (KK,KD)
      CALL OUTPUTQ (KK,KD)
      CALL OUTPUT3 (IA,JA,ZA,HA,KK,KD,1)
      CALL OUTPUT2 (IB,JB,ZB,HB,KK,KD,2)
      CALL OUTPUT2 (IC,JC,ZC,HC,KK,KD,3)
      CALL OUTPUT3 (ID1,JD1,ZD1,HD1,KK,KD,4)
      CALL OUTPUT3 (ID2,JD2,ZD2,HD2,KK,KD,5)
      CALL OUTPUT3 (ID3,JD3,ZD3,HD3,KK,KD,6)
      CALL OUTPUT3 (ID4,JD4,ZD4,HD4,KK,KD,7)
      CALL OUTPUT3 (ID5,JD5,ZD5,HD5,KK,KD,8)
      CALL OUTPUT3 (IE1,JE1,ZE,HE,KK,KD,9)
      CALL OUTPUT3 (IF1,JF1,ZF,HF,KK,KD,10)
      RETURN ; END
C
      SUBROUTINE OUTPUT1 (KK,KD)
C
      REAL M,N
      PARAMETER (IO=1020,JO=738)
      PARAMETER (GX=1.E-2)
      COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
      COMMON /HH/ H(IO,JO)
      DIMENSION KW(100),KM(23)
      DATA KM/1HJ,1HI,1HH,1HG,1HF,1HE,1HD,1HC,
     &        1HB,1HA,1H.,1H1,1H2,1H3,1H4,1H5,
     &        1H6,1H7,1H8,1H9,1H0,1H*,1H /
      IF(MOD(KK,KD).NE.0)RETURN
      WRITE(6,410)KK,(KI,KI=130,289,20)
      DO 30 KY=1,60
      KJ=60-KY+1
      J1=12*(KJ-1)+1
      J2=J1+11
      DO 40 KI=1,85
      I1=12*(KI-1)+1
      I2=I1+11
      L=0  ; S=0.
      DO 41 J=J1,J2
      DO 41 I=I1,I2
      IF(H(I,J).GT.0.)THEN
      L=L+1  ; S=S+Z(I,J,2)
      ENDIF
   41 CONTINUE
      IF(L.GE.60)THEN
      S=S/L
      KMM=11+INT(S/0.1)
      IF(KMM.GT.21)KMM=21
      IF(KMM.LT.1)KMM=1
      IF(ABS(S).LT.GX)KMM=23
      ELSE
      KMM=22
      ENDIF
      KW(KI)=KM(KMM)
   40 CONTINUE
      JK=2*KJ-60
      WRITE(6,400)JK,(KW(I),I=1,85)
   30 CONTINUE
      RETURN
  400 FORMAT(1H ,I3,2X,85A1)
  410 FORMAT(1H1,/,2X,"MAP AT K=",I5,//,6X,11I10)
      END
C
      SUBROUTINE OUTPUT2 (IF,JF,Z,H,KK,KD,CHR)
C
      INTEGER CHR
      PARAMETER (GX=1.E-2)
      DIMENSION Z(IF,JF,2),H(IF,JF)
      DIMENSION KW(100),KM(23)
```

```
      CHARACTER NAME(10)*10
      DATA KM/1HJ,1HI,1HH,1HG,1HF,1HE,1HD,1HC,
     &       1HB,1HA,1H.,1H1,1H2,1H3,1H4,1H5,
     &       1H6,1H7,1H8,1H9,1H0,1H*,1H /
      NAME(1)="A REGION"
      NAME(2)="B REGION"
      NAME(3)="C REGION"
      NAME(4)="D1 REGION"
      NAME(5)="D2 REGION"
      NAME(6)="D3 REGION"
      NAME(7)="D4 REGION"
      NAME(8)="D5 REGION"
      NAME(9)="E REGION"
      NAME(10)="F REGION"
      IF(MOD(KK,KD).NE.0)RETURN
      WRITE(6,450)NAME(CHR)
  450 FORMAT(1H ,/,10X,A10)
      IN=IF/20
      WRITE(6,410)KK,(KI,KI=20,IN*20,20)
      DO 10 J=JF,1,-2
      JS=J-1 ; JE=J ; IF(JS.LE.0)JS=1
      DO 20 I=1,IF/2
      IS=2*I-1 ; IE=IS+1 ; IF(IE.GE.IF)IE=IF
      S=0.0 ; L=0
      DO 30 II=IS,IE
      DO 30 JJ=JS,JE
      IF(H(II,JJ).GT.0.)THEN
      L=L+1
      S=S+Z(II,JJ,2)
      ENDIF
   30 CONTINUE
      IF(L.GE.2)THEN
      S=S/L
      KMM=11+INT(S/0.5)
      IF(KMM.GT.21)KMM=21
      IF(KMM.LT.1)KMM=1
      IF(ABS(S).LT.GX)KMM=23
      ELSE
      KMM=22
      ENDIF
      KW(I)=KM(KMM)
   20 CONTINUE
      II=IF/2
      WRITE(6,400)J,(KW(I),I=1,II)
   10 CONTINUE
      RETURN
  400 FORMAT(1H ,I3,2X,100A1)
  410 FORMAT(1H ,/,2X,"MAP AT K=",I5,//,6X,11I10)
      END
C
      SUBROUTINE OUTPUT3 (IF,JF,Z,H,KK,KD,CHR)
C
      INTEGER CHR
      PARAMETER (GX=1.E-2)
      DIMENSION Z(IF,JF,2),H(IF,JF)
      DIMENSION KW(100),KM(23)
      CHARACTER NAME(10)*10
      DATA KM/1HJ,1HI,1HH,1HG,1HF,1HE,1HD,1HC,
     &       1HB,1HA,1H.,1H1,1H2,1H3,1H4,1H5,
     &       1H6,1H7,1H8,1H9,1H0,1H*,1H /
      NAME(1)="A REGION"
      NAME(2)="B REGION"
      NAME(3)="C REGION"
      NAME(4)="D1 REGION"
      NAME(5)="D2 REGION"
      NAME(6)="D3 REGION"
      NAME(7)="D4 REGION"
      NAME(8)="D5 REGION"
      NAME(9)="E REGION"
      NAME(10)="F REGION"
      IF(MOD(KK,KD).NE.0)RETURN
      IN=IF/10 ; JN=JF/10
      WRITE(6,450)NAME(CHR)
  450 FORMAT(1H ,/,10X,A10)
```

```
      WRITE(6,410)KK,(KI,KI=10,100,10)
      DO 10 J=JF,1,-1
      IE=IF ; IF(IE.GE.100)IE=100
      DO 20 I=1,IE
      IF(H(I,J).GT.0.0)THEN
      KMM=11+INT(Z(I,J,2)/0.5)
      IF(KMM.GT.21)KMM=21
      IF(KMM.GT.21)KMM=21
      IF(KMM.LT.1)KMM=1
      IF(ABS(Z(I,J,2)).LT.GX)KMM=23
      ELSE
      KMM=22
      ENDIF
      KW(I)=KM(KMM)
   20 CONTINUE
      WRITE(6,400)J,(KW(I),I=1,IE)
   10 CONTINUE
      RETURN
  400 FORMAT(1H ,I3,2X,100A1)
  410 FORMAT(1H ,/,2X,"MAP AT K=",I5,//,6X,11I10)
      END
C
      SUBROUTINE OUTPUTQ (KK,KD)
C
      REAL M,N
      PARAMETER (GX=1.E-2)
      PARAMETER (IO=1020,JO=738)
      COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
      COMMON /HH/ H(IO,JO)
      DIMENSION KW(100),KM(23)
      DATA KM/1HJ,1HI,1HH,1HG,1HF,1HE,1HD,1HC,
     &        1HB,1HA,1H.,1H1,1H2,1H3,1H4,1H5,
     &        1H6,1H7,1H8,1H9,1H0,1H*,1H /
      IF(MOD(KK,KD).NE.0)RETURN
      IS=101 ; IE=200 ; JS=581 ; JE=630
      WRITE(6,410)KK,(KI,KI=100,200,10)
      DO 10 J=JE,JS,-1
      DO 20 I=1,100
      JJ=J ; II=I+100
      IF(H(II,JJ).GT.0.0)THEN
      KMM=11+INT(Z(II,JJ,2)/0.1)
      IF(KMM.GT.21)KMM=21
      IF(KMM.GT.21)KMM=21
      IF(KMM.LT.1)KMM=1
      IF(ABS(Z(II,JJ,2)).LT.GX)KMM=23
      ELSE
      KMM=22
      ENDIF
      KW(I)=KM(KMM)
   20 CONTINUE
      WRITE(6,400)J,(KW(I),I=1,100)
   10 CONTINUE
      RETURN
  400 FORMAT(1H ,I3,2X,100A1)
  410 FORMAT(1H ,///,2X,"MAP AT K=",I5,/,6X,11I10)
      END
C
      SUBROUTINE CHANGE
C
      REAL M,N
      PARAMETER (IO=1020,JO=738)
      COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
      DO 10 J=1,JO
      DO 10 I=1,IO
      Z(I,J,1)=Z(I,J,2)
      M(I,J,1)=M(I,J,2)
      N(I,J,1)=N(I,J,2)
   10 CONTINUE
      RETURN ; END
C
      SUBROUTINE CHANG (IF,JF,Z,M,N,LP,LT)
C
      REAL M,N
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
```

```
      IF(MOD(LP,LT).NE.0)RETURN
      DO 10 J=1,JF
      DO 10 I=1,IF
      Z(I,J,1)=Z(I,J,2)
      M(I,J,1)=M(I,J,2)
      N(I,J,1)=N(I,J,2)
   10 CONTINUE
      RETURN ; END
C
      SUBROUTINE FILEOT (IF,JF,Z,M,N)
C
      REAL M,N
      DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
C
      DO 10 J=1,JF
      WRITE(25,100)(Z(I,J,2),I=1,IF)
      WRITE(25,100)(M(I,J,2),I=1,IF)
      WRITE(25,100)(N(I,J,2),I=1,IF)
   10 CONTINUE
  100 FORMAT(10E12.5)
      RETURN
      END
C
      SUBROUTINE MAX (IF,JF,IS,JS,IE,JE,Z,H,ZM,ZN)
C
      DIMENSION Z(IF,JF,2),H(IF,JF),ZM(IF,JF),ZN(IF,JF)
      DO 10 I=IS,IE
      DO 10 J=JS,JE
      IF(H(I,J).LT.0.0)GOTO 10
      IF(ZM(I,J).LT.Z(I,J,2))ZM(I,J)=Z(I,J,2)
      IF(ZN(I,J).GT.Z(I,J,2))ZN(I,J)=Z(I,J,2)
   10 CONTINUE
      RETURN
      END
C
      SUBROUTINE POINT (IF,JF,Z,NG,PZ,IP,JP,KK,KC)
C
      DIMENSION Z(IF,JF,2),PZ(NG),IP(NG),JP(NG)
C
      IF(MOD(KK,KC).NE.0)RETURN
      DO 10 K=1,NG
      PZ(K)=Z(IP(K),JP(K),2)
   10 CONTINUE
      WRITE(24,100)(PZ(K),K=1,NG)
  100 FORMAT(10F7.3)
      RETURN
      END
C
      SUBROUTINE PROPA (IF,JF,IS,JS,IE,JE,Z,H,LF,KK)
C
      DIMENSION Z(IF,JF,2),H(IF,JF),LF(IF,JF)
C
      LK=KK/180+1
      DO 10 I=IS,IE
      DO 10 J=JS,JE
      IF(H(I,J).GT.0.0)THEN
      IF(LF(I,J).EQ.0)THEN
      IF(ABS(Z(I,J,2)).GT.0.01)LF(I,J)=LK
      ENDIF
      ELSE
      LF(I,J)=-9
      ENDIF
   10 CONTINUE
      RETURN
      END
C
      SUBROUTINE OUTDT (IF,JF,ZM,ZN,LK)
      DIMENSION ZM(IF,JF),ZN(IF,JF),LK(IF,JF)
      DO 20 I=1,IF
      DO 20 J=1,JF
      ZN(I,J)=ZM(I,J)-ZN(I,J)
      IF(ZN(I,J).GT.100.0)ZN(I,J)=100.0
      IF(ZM(I,J).GT.100.0)ZM(I,J)=100.0
   20 CONTINUE
```

```
      DO 10 J=1,JF
      WRITE(31,100)(ZM(I,J),I=1,IF)
      WRITE(32,100)(ZN(I,J),I=1,IF)
      WRITE(33,200)(LK(I,J),I=1,IF)
   10 CONTINUE
  100 FORMAT(20F6.3)
  200 FORMAT(40I3)
      RETURN
      END
C
      SUBROUTINE DEFORM (IO,JO,IS,JS,IE,JE,Z)
      REAL L
      PARAMETER (DR=10.0)
      PARAMETER (A=3.141592,B=4.848E-06)
      PARAMETER (RR=6.37E+6,E=1.7453E-2)
      PARAMETER (HH=53.0E+3,D=24.0,DL=10.0,TH=10.0,RD=90.0)
      PARAMETER (L=800.0E+3,W=200.0E+3)
      PARAMETER (YO=-50.0,XO=280.0)
      PARAMETER (Y0=-46.0,X0=284.0)
      DIMENSION Z(IO,JO,2)
      XL=A*RR*(X0-XO)*COS(E*YO)/180.0 ; YL=A*RR*(Y0-YO)/180.0
      H1=HH/SIN(E*DL)  ; H2=HH/SIN(E*DL)+W
      DS=D*COS(E*RD)   ; DD=D*SIN(E*RD)
      WRITE(6,*)XL,YL,H1,H2,DS,DD
      DO 10 I=1,IE-IS+1
      DO 10 J=1,JE-JS+1
C      XX=DX*(I-1) ; YY=DX*(J-1)
      YY=A*RR*DR*(J-1)/(60.0*180)
      XX=A*RR*DR*(I-1)*COS(E*(YO+DR*(J-1)/60.0))/(60.0*180)
      X1=(XX-XL)*SIN(E*TH)+(YY-YL)*COS(E*TH)-L/2.0
      X2=(XX-XL)*COS(E*TH)-(YY-YL)*SIN(E*TH)+HH/TAN(E*DL)
      X3=0.0
      CALL USCAL (X1,X2,X3, L/2.,H2,E*DL,F1)
      CALL USCAL (X1,X2,X3, L/2.,H1,E*DL,F2)
      CALL USCAL (X1,X2,X3,-L/2.,H2,E*DL,F3)
      CALL USCAL (X1,X2,X3,-L/2.,H1,E*DL,F4)
      CALL UDCAL (X1,X2,X3, L/2.,H2,E*DL,G1)
      CALL UDCAL (X1,X2,X3, L/2.,H1,E*DL,G2)
      CALL UDCAL (X1,X2,X3,-L/2.,H2,E*DL,G3)
      CALL UDCAL (X1,X2,X3,-L/2.,H1,E*DL,G4)
      US=(F1-F2-F3+F4)*DS/(12.0*A)
      UD=(G1-G2-G3+G4)*DD/(12.0*A)
      II=I+IS-1  ; JJ=J+JS-1
      Z(II,JJ,1)=US+UD
      Z(II,JJ,2)=US+UD
   10 CONTINUE
      RETURN
      END
C
      SUBROUTINE USCAL (X1,X2,X3,C,CC,DP,F)
      REAL K
C
      SN=SIN(DP)   ; CS=COS(DP)
      C1=C ; C2=CC*CS ; C3=CC*SN
      R=SQRT((X1-C1)**2+(X2-C2)**2+(X3-C3)**2)
      Q=SQRT((X1-C1)**2+(X2-C2)**2+(X3+C3)**2)
      R2=X2*SN-X3*CS
      R3=X2*CS+X3*SN
      Q2=X2*SN+X3*CS
      Q3=-X2*CS+X3*SN
      H=SQRT(Q2**2+(Q3+CC)**2)
      K=SQRT((X1-C1)**2+Q2**2)
      A1=LOG(R+R3-CC)
      A2=LOG(Q+Q3+CC)
      A3=LOG(Q+X3+C3)
      B1=1+3.0*(TAN(DP))**2
      B2=3.0*TAN(DP)/CS
      B3=2.0*R2*SN
      B4=Q2+X2*SN
      B5=2.0*R2**2*CS
      B6=R*(R+R3-CC)
      B7=4.0*Q2*X3*SN**2
      B8=2.0*(Q2+X2*SN)*(X3+Q3*SN)
      B9=Q*(Q+Q3+CC)
```

```
      B10=4.0*Q2*X3*SN
      B11=(X3+C3)-Q3*SN
      B12=4.0*Q2**2*Q3*X3*CS*SN
      B13=2.0*Q+Q3+CC
      B14=Q**3*(Q+Q3+CC)**2
      F=CS*(A1+B1*A2-B2*A3)
     &+B3/R+2*SN*B4/Q-B5/B6+(B7-B8)/B9+B10*B11/Q**3-B12*B13/B14
      RETURN
      END
C
      SUBROUTINE UDCAL (X1,X2,X3,C,CC,DP,F)
      REAL K
C
      SN=SIN(DP)  ; CS=COS(DP)
      C1=C ; C2=CC*CS ; C3=CC*SN
      R=SQRT((X1-C1)**2+(X2-C2)**2+(X3-C3)**2)
      Q=SQRT((X1-C1)**2+(X2-C2)**2+(X3+C3)**2)
      R2=X2*SN-X3*CS
      R3=X2*CS+X3*SN
      Q2=X2*SN+X3*CS
      Q3=-X2*CS+X3*SN
      H=SQRT(Q2**2+(Q3+CC)**2)
      K=SQRT((X1-C1)**2+Q2**2)
      A1=LOG(R+X1-C1)
      A2=LOG(Q+X1-C1)
      B1=Q*(Q+X1-C1)
      B2=R*(R+X1-C1)
      B3=Q*(Q+Q3+CC)
      D1=X1-C1  ; D2=X2-C2  ; D3=X3-C3
      D4=X3+C3  ; D5=R3-CC  ; D6=Q3+CC
      T1=ATN(D1*D2,(H+D4)*(Q+H))
      T2=ATN(D1*D5,R2*R)
      T3=ATN(D1*D6,Q2*Q)
      F=SN*(D2*(2*D3/B2+4*D3/B1-4*C3*X3*D4*(2*Q+D1)/(B1**2*Q))
        &  -6*T1+3*T2-6*T3)
     &+CS*(A1-A2-2*(D3**2)/B2
     &-4*(D4**2-C3*X3)/B1-4*C3*X3*D4**2*(2*Q+X1-C1)/(B1**2*Q))
     &+6*X3*(CS*SN*(2*D6/B1+D1/B3)-Q2*(SN**2-CS**2)/B1)
      RETURN
      END
C
      REAL FUNCTION ATN (AX,AY)
      DATA GX/1.0E-6/
      AAX=ABS(AX) ; AAY=ABS(AY)
      P=AX*AY
      IF(AAX.LE.GX.AND.AAY.LE.GX)GOTO 10
      SR=ATAN2(AAX,AAY)
      ATN=SIGN(SR,P)
      RETURN
   10 WRITE(6,100)AX,AY
  100 FORMAT(1H ,"ATAN --    AX=",E15.7,2X,"AY=",E15.7)
      ATN=0.2
      RETURN
      END
```