

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

The designations employed and the presentation of the material in this publication do not imply the expression of any opinion whatsoever on the part of the Secretariats of UNESCO and IOC concerning the legal status of any country, territory, or its authorities, or concerning the delimitations of the frontiers of any country or territory.

**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
© UNESCO 1997
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

	Page
1.1 EQUATIONS AND DIFFERENCE SCHEMES	1
(1) Shallow Water Theory	1
(2) Difference Scheme and Stability	2
1.2 INITIAL CONDITIONS AND BOUNDARY CONDITIONS	9
(1) Initial Conditions	9
(2) Conditions for a Simple Harmonic Wave Train at an Offshore Open Boundary	9
(3) Open Boundary Conditions for Forced Inputs	12
(4) Open Boundary Conditions for Free Transmission	12
(5) Boundary Conditions at Run up Front	12
(6) Boundary Conditions at Water Overflows Structures	12
1.3 CONTINUATION OF REGIONS	12
(1) Necessity of Continuation of Regions in Numerical Computation	12
(2) Continuation of Regions of Different Δx	13
(3) Continuation of Regions of Different Δt	16

CHAPTER 2: TYPICAL PROGRAMMES

2.1 ASSUMPTIONS	1
2.2 CONSTRUCTION OF MAIN PROGRAMMES	1
(1) Specification Statement	1
(2) Input of Setting Value	3
(3) Setting of Initial Value	3
(4) Computation Repeated with Respect to Time	4
2.3 EXPLANATION OF SUB-ROUTINES	10
(1) DEPTH (Computation of Water Depth at Points for Discharge)	10
(2) SETZRO (Setting of Initial Conditions)	11
(3) CONTIN (Computation of the Equation of Continuity)	12
(4) JOINTZ (Connection of the Water Level in Space and Time)	14
(5) MOTION (Computation of the Equation of Motion)	16
(6) JOINTQ (Connection of the Water Discharge in Space and Time)	20
(7) CHANGE (Change of Data)	23

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.

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

where D is the total water depth given by $h + \eta$ 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

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

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.

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

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.

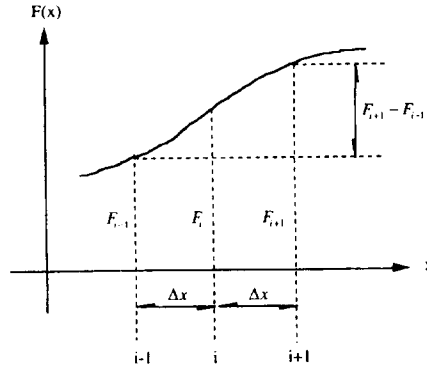


Figure 1-1. Central difference

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

The Taylor series expressions for F_{i-1} and F_{i+1} are

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

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

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

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

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

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

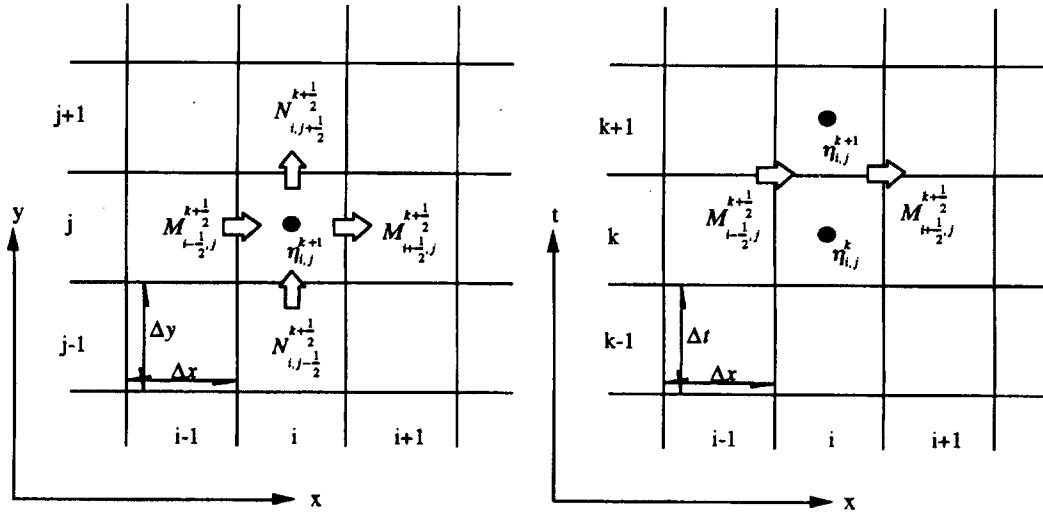


Figure 1-2. Arrangement of points for computation in the leap-frog method.

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

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

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

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

by

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 $M(I+1/2, j, k-1/2)$

$$M_{i+\frac{1}{2},j}^{k+\frac{1}{2}} = M_{i+\frac{1}{2},j}^{k-\frac{1}{2}} - gD_{i+\frac{1}{2},j}^k \frac{\Delta t}{\Delta x} [\eta_{i+1,j}^k - \eta_{i,j}^k]$$

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

$$D_{i+\frac{1}{2},j}^k = h_{i+\frac{1}{2},j} - \eta_{i+\frac{1}{2},j}^k = h_{i+\frac{1}{2},j} + \frac{1}{2} [\eta_{i+1,j}^k + \eta_{i,j}^k]$$

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

$$N_{i,j+\frac{1}{2}}^{k+\frac{1}{2}} = N_{i,j+\frac{1}{2}}^{k-\frac{1}{2}} - gD_{i,j+\frac{1}{2}}^k \frac{\Delta t}{\Delta y} [\eta_{i,j+1}^k + \eta_{i,j}^k]$$

$$D_{i,j+\frac{1}{2}}^k = h_{i,j+\frac{1}{2}} + \eta_{i,j+\frac{1}{2}}^k = h_{i,j+\frac{1}{2}} + \frac{1}{2} [\eta_{i,j+1}^k + \eta_{i,j}^k]$$

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 than 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

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

The central difference is applied to the space derivative.

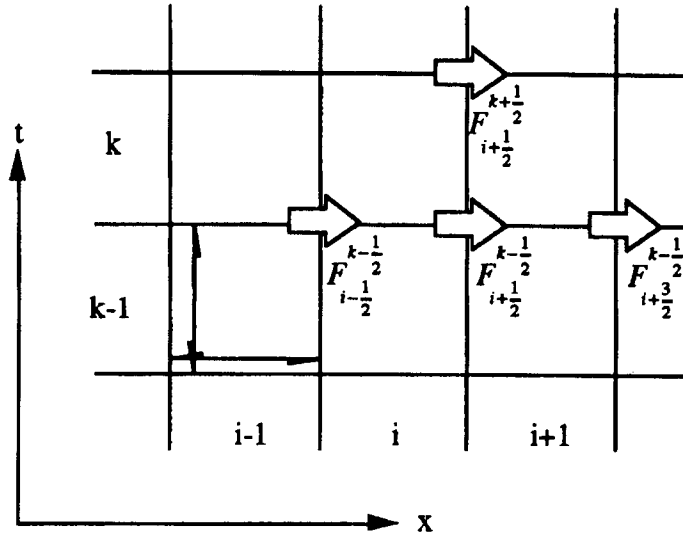


Figure 1-3. Arrangement of points for convection terms.

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

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

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

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.

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

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

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

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.

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

$$\frac{\partial}{\partial y} \left(\frac{MN}{D} \right) = \frac{1}{\Delta y} \left[v_{11} \frac{M_{i+\frac{1}{2},j+1}^{k-\frac{1}{2}} N_{i+\frac{1}{2},j+1}^{k-\frac{1}{2}}}{D_{i+\frac{1}{2},j+1}^{k-\frac{1}{2}}} + v_{21} \frac{M_{i+\frac{1}{2},j}^{k-\frac{1}{2}} N_{i+\frac{1}{2},j}^{k-\frac{1}{2}}}{D_{i+\frac{1}{2},j}^{k-\frac{1}{2}}} + v_{31} \frac{M_{i+\frac{1}{2},j-1}^{k-\frac{1}{2}} N_{i+\frac{1}{2},j-1}^{k-\frac{1}{2}}}{D_{i+\frac{1}{2},j-1}^{k-\frac{1}{2}}} \right]$$

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

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

where

$$\begin{aligned} M_{i+\frac{1}{2},j}^{k-\frac{1}{2}} &\geq 0, \quad \lambda_{11}=0, \quad \lambda_{21}=1, \quad \lambda_{31}=-1 \\ &< 0, \quad \lambda_{11}=1, \quad \lambda_{21}=-1, \quad \lambda_{31}=0 \end{aligned}$$

$$\begin{aligned} N_{i+\frac{1}{2},j}^{k-\frac{1}{2}} &\geq 0, \quad v_{11}=0, \quad v_{21}=1, \quad v_{31}=-1 \\ &< 0, \quad v_{11}=1, \quad v_{21}=-1, \quad v_{31}=0 \end{aligned}$$

$$\begin{aligned} M_{i,j+\frac{1}{2}}^{k-\frac{1}{2}} &\geq 0, \quad \lambda_{12}=0, \quad \lambda_{22}=1, \quad \lambda_{32}=-1 \\ &< 0, \quad \lambda_{12}=1, \quad \lambda_{22}=-1, \quad \lambda_{32}=0 \end{aligned}$$

$$\begin{aligned} N_{i,j+\frac{1}{2}}^{k-\frac{1}{2}} &\geq 0, \quad v_{12}=0, \quad v_{22}=1, \quad v_{32}=-1 \\ &< 0, \quad v_{12}=1, \quad v_{22}=-1, \quad v_{32}=0 \end{aligned}$$

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

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

$$\frac{gn^2}{D^{7/3}} N \sqrt{M^2 + N^2} = \frac{gn^2}{\left(D_{ij+\frac{1}{2}}\right)^{7/3}} \frac{1}{2} \left(N_{ij+\frac{1}{2}}^{k+\frac{1}{2}} + N_{ij+\frac{1}{2}}^{k-\frac{1}{2}} \right) \sqrt{\left(M_{ij+\frac{1}{2}}^{k-\frac{1}{2}} \right)^2 + \left(N_{ij+\frac{1}{2}}^{k-\frac{1}{2}} \right)^2}$$

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

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

$$\begin{aligned} & - \frac{1}{1+x\mu_{i+\frac{1}{2}j}^{k-\frac{1}{2}}} \left[\left(1 - x\mu_{i+\frac{1}{2}j}^{k-\frac{1}{2}} \right) M_{i+\frac{1}{2}j}^{k-\frac{1}{2}} - \frac{\Delta t}{\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{M}{L} \right. \right. \\ & \left. \left. + v_{11} \frac{M_{i+\frac{1}{2}j+1}^{k-\frac{1}{2}} N_{i+\frac{1}{2}j+1}^{k-\frac{1}{2}}}{D_{i+\frac{1}{2}j+1}^{k-\frac{1}{2}}} + v_{21} \frac{M_{i+\frac{1}{2}j}^{k-\frac{1}{2}} N_{i+\frac{1}{2}j}^{k-\frac{1}{2}}}{D_{i+\frac{1}{2}j}^{k-\frac{1}{2}}} + v_{31} \frac{M_{i+\frac{1}{2}j-1}^{k-\frac{1}{2}} N_{i+\frac{1}{2}j-1}^{k-\frac{1}{2}}}{D_{i+\frac{1}{2}j-1}^{k-\frac{1}{2}}} \right\} g D_{i+\frac{1}{2}j}^k \frac{\Delta t}{\Delta x} \left\{ \eta_{i+1}^k \right. \right. \\ & \left. \left. - \frac{1}{y\mu_{ij+\frac{1}{2}}^{k-\frac{1}{2}}} \left[\left(1 - y\mu_{ij+\frac{1}{2}}^{k-\frac{1}{2}} \right) N_{ij+\frac{1}{2}}^{k-\frac{1}{2}} - \frac{\Delta t}{\Delta y} \left\{ \lambda_{12} \frac{M_{i+1,j+\frac{1}{2}}^{k-\frac{1}{2}} N_{i+1,j+\frac{1}{2}}^{k-\frac{1}{2}}}{D_{i+1,j+\frac{1}{2}}^{k-\frac{1}{2}}} + \lambda_{22} \frac{M_{ij+\frac{1}{2}}^{k-\frac{1}{2}} N_{ij+\frac{1}{2}}^{k-\frac{1}{2}}}{D_{ij+\frac{1}{2}}^{k-\frac{1}{2}}} + \lambda_{32} \frac{M_i^k}{D_{ij+\frac{1}{2}}^{k-\frac{1}{2}}} \right. \right. \right. \\ & \left. \left. \left. - \frac{\Delta t}{\Delta y} \left\{ v_{12} \frac{\left(N_{ij+\frac{3}{2}}^{k-\frac{1}{2}} \right)^2}{D_{ij+\frac{3}{2}}^{k-\frac{1}{2}}} + v_{22} \frac{\left(N_{ij+\frac{1}{2}}^{k-\frac{1}{2}} \right)^2}{D_{ij+\frac{1}{2}}^{k-\frac{1}{2}}} + v_{32} \frac{\left(N_{ij-\frac{1}{2}}^{k-\frac{1}{2}} \right)^2}{D_{ij-\frac{1}{2}}^{k-\frac{1}{2}}} \right\} g D_{ij+\frac{1}{2}}^k \frac{\Delta t}{\Delta y} \left\{ \eta_{i,j+1}^k - \eta_{i,j}^k \right\} \right] \right\} \end{aligned}$$

where

$$\mu_{i+\frac{1}{2}j}^{k-\frac{1}{2}} = \frac{1}{2} \frac{gn^2}{\left(D_{i+\frac{1}{2}j}\right)^2} \sqrt{\left(M_{i+\frac{1}{2}j}^{k-\frac{1}{2}} \right)^2 + \left(N_{i+\frac{1}{2}j}^{k-\frac{1}{2}} \right)^2}$$

$$\mu_{ij+\frac{1}{2}}^{k-\frac{1}{2}} = \frac{1}{2} \frac{gn^2}{\left(D_{ij+\frac{1}{2}}^{k-\frac{1}{2}}\right)^2} \sqrt{\left(M_{ij+\frac{1}{2}}^{k-\frac{1}{2}}\right)^2 + \left(N_{ij+\frac{1}{2}}^{k-\frac{1}{2}}\right)^2}$$

$$D_{i+\frac{1}{2},j}^k = \frac{1}{2} (D_{i+1,j}^k + D_{i,j}^k) = \frac{1}{2} (\eta_{i+1,j}^k + \eta_{i,j}^k) + h_{i+\frac{1}{2},j}$$

$$\begin{aligned} D_{i+\frac{1}{2},j}^{k-\frac{1}{2}} &= \frac{1}{4} (D_{i+1,j}^k + D_{i+1,j}^{k-1} + D_{i,j}^k + D_{i,j}^{k-1}) \\ &= \frac{1}{4} (\eta_{i+1,j}^k + \eta_{i+1,j}^{k-1} + \eta_{i,j}^k + \eta_{i,j}^{k-1}) + h_{i+\frac{1}{2},j} \end{aligned}$$

$$D_{ij+\frac{1}{2}}^k = \frac{1}{2} (D_{i,j+1}^k + D_{i,j}^k) = \frac{1}{2} (\eta_{i,j+1}^k + \eta_{i,j}^k) + h_{ij+\frac{1}{2}}$$

$$\begin{aligned} D_{ij+\frac{1}{2}}^{k-\frac{1}{2}} &= \frac{1}{4} (D_{i,j+1}^k + D_{i,j+1}^{k-1} + D_{i,j}^k + D_{i,j}^{k-1}) \\ &= \frac{1}{4} (\eta_{i,j+1}^k + \eta_{i,j+1}^{k-1} + \eta_{i,j}^k + \eta_{i,j}^{k-1}) + h_{ij+\frac{1}{2}} \end{aligned}$$

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,

$$\eta_{ij}^{k-1}, \quad M_{i+\frac{1}{2},j}^{k-\frac{1}{2}}, \quad N_{ij+\frac{1}{2}}^{k-\frac{1}{2}} = 0$$

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

$$\eta_{i,j}^{k-1} = -h_{ij}$$

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

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

and Eq.(37) is reduced to

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

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

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

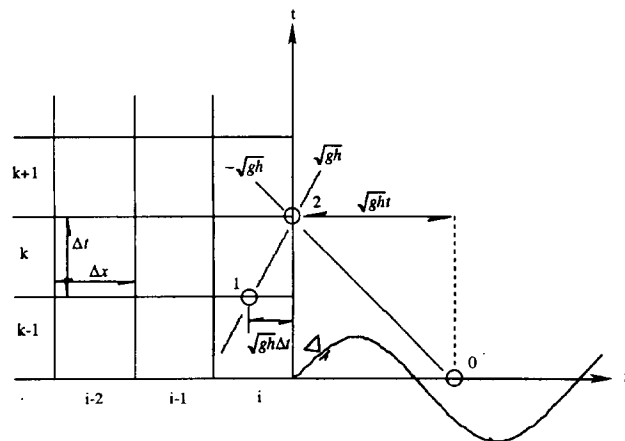


Figure 4. An offshore open boundary and characteristics when a simple harmonic wave train is propagating in the negative x direction.

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

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

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

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

where M_1 and η_1 are calculated by

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

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

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 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

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

This yields

$$u_2 = u_1 \quad \text{or} \quad M_{i+\frac{1}{2}, j}^{k+\frac{1}{2}} = M_1$$

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.

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

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

$$Q = \mu h_1 \sqrt{2gh_1} \quad , \quad \text{if} \quad h_2 \leq \frac{2}{3} h_1 \\ = \mu' h_1 \sqrt{2g(h_1 - h_2)} \quad , \quad \text{if} \quad h_2 > \frac{2}{3} h_1$$

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.

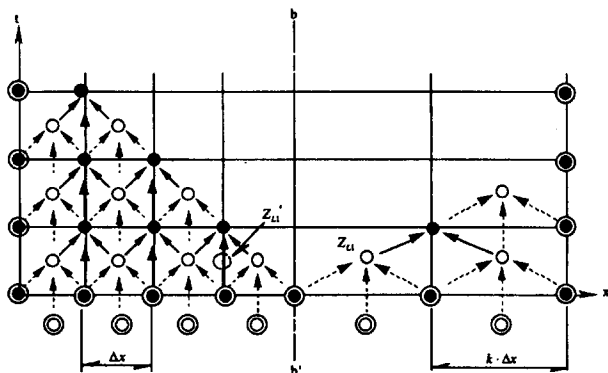
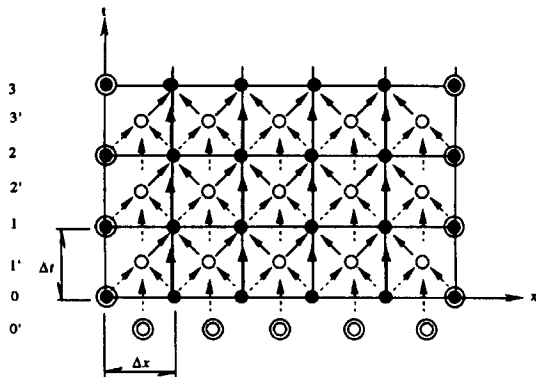


Figure 1-5. Computation procedure in the x-t plane, (A) if the spatial grid length is constant in the whole region, and (B) if two regions of different grid length are connected at the boundary b-b'.

(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.

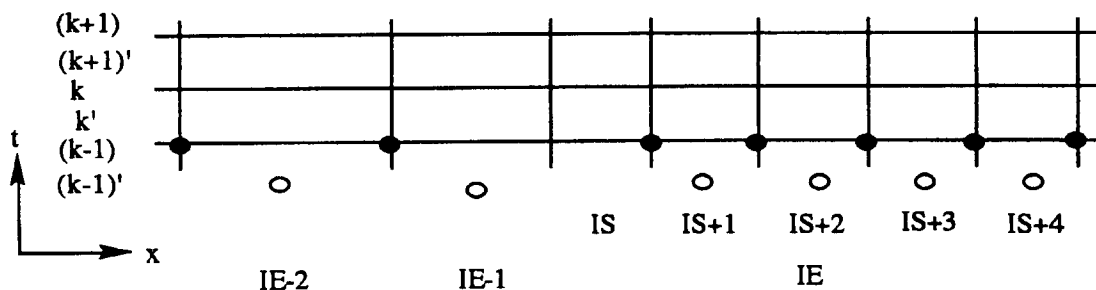


Figure 1-6(a). Details of computation continuation in the x-t plane in case of $k = 3$ and $\Delta t = \text{const}$. Open (filled) circles are points where the water level (discharge) is computed.

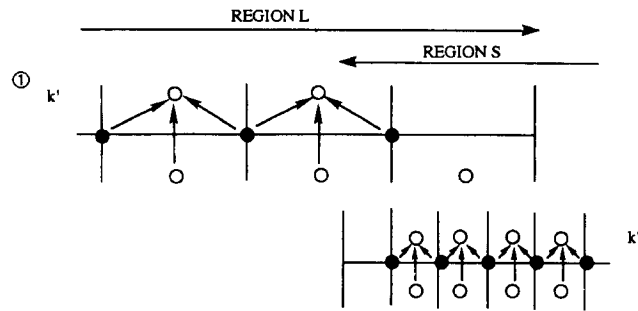


Figure 1-6(b). With the equation of continuity, $Z_S(I, K')$ and $Z_L(I', K')$ are computed. Suffixes $_S$ and $_L$ denote values in the region S of Δx and in the region L of $3\Delta x$

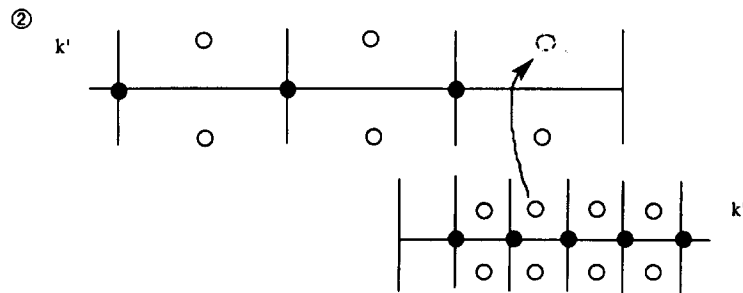


Figure 1-6(c). The water level $Z_L(IE, K')$ is set equal to $Z_S(IS'', K')$

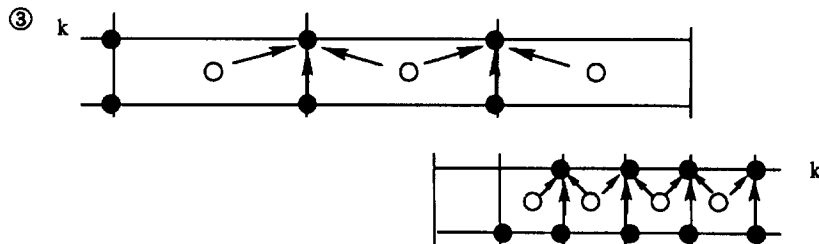


Figure 1-6(d). With the equation of motion, $M_S(I, K)$ and $M_L(I', K)$ are computed.

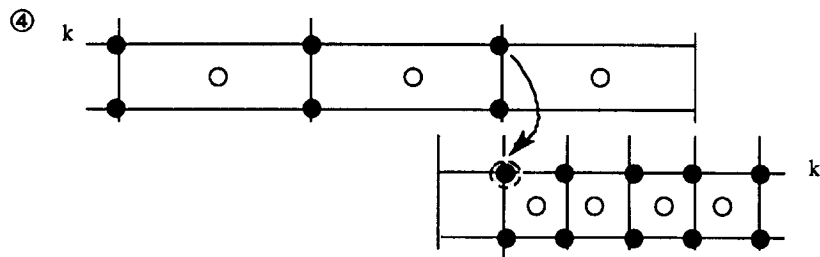


Figure 1-6(e). The water discharge $M_S(IS, K)$ is set to $M_L(IE-1, K)$

(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.

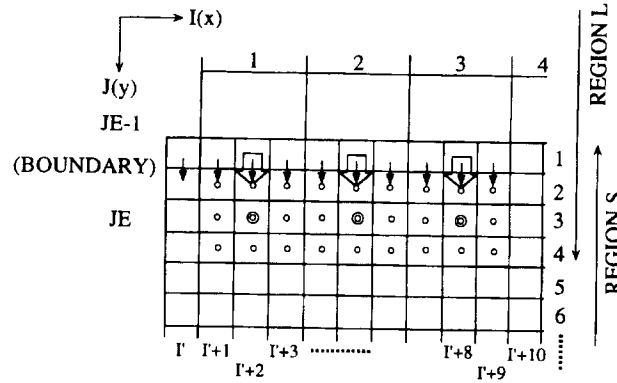


Figure 1-7. Continuation between two regions S and L in the $(x,y-t)$ space, where $k = 3$ and $\Delta t = \text{const.}$

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 $3\Delta t$ to Δt . Figure 1-8 shows the computation procedure in the x-t plane.

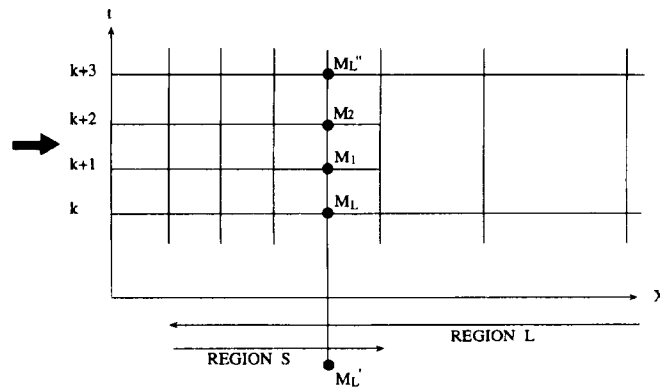


Figure 1-8. Computation continuation between two regions S and L in the x-t plane when Δt varies at the ratio of 1:3.

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, M_1 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.

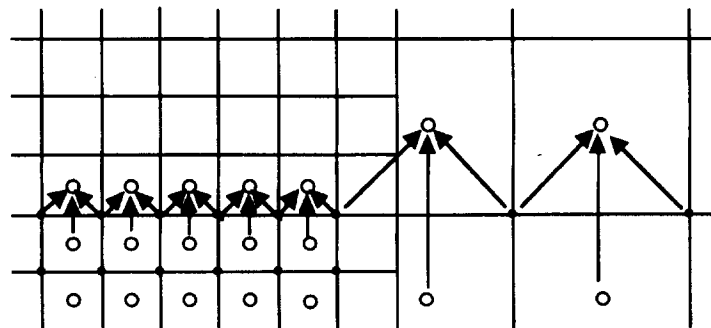


Figure 1-9(a). Details of computation continuation in the x-t plane, when Δt varies at the ratio of 1:3. Computation of the water level Z in the regions S and L, with the equation of continuity.

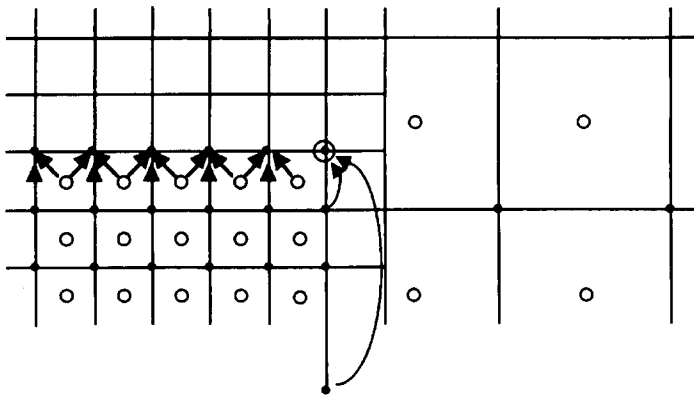


Figure 1-9(b). Computation of the water discharge M in the region S, with the equation of motion. An extrapolation in the region L to determine value of M at the boundary. Connection of the water discharge from the region L to the region S.

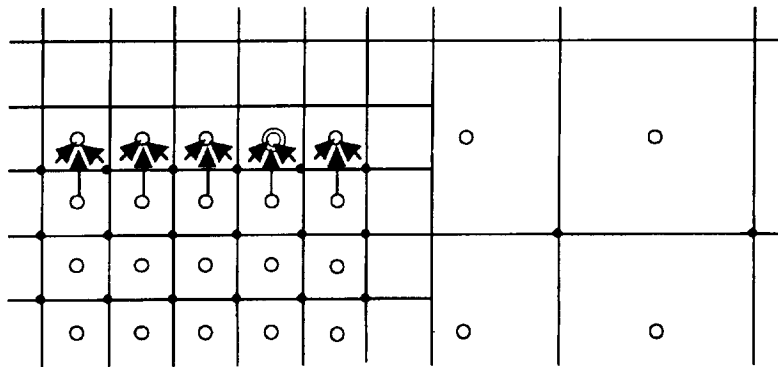


Figure 1-9(c). Computation of the water level Z in the region S, with the equation of continuity. Connection of the water level from the region S to the region L.

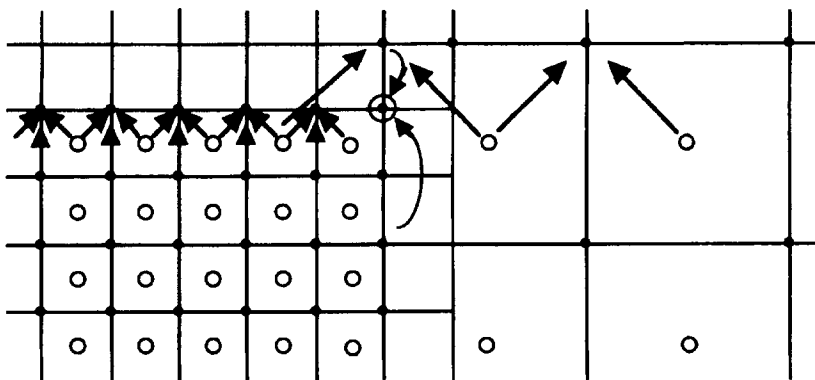


Figure 1-9(d). Computation of the water discharge M in the region S and L, with the equation of motion. An interpolation in the region L to determine values of M at the boundary. Connection of the water discharge from the region L to the region S.

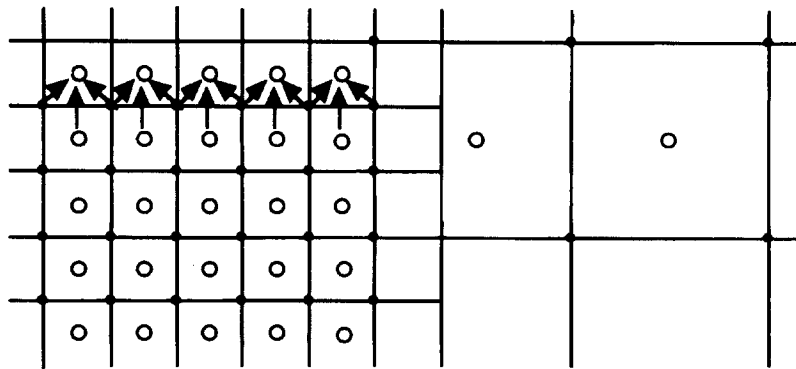


Figure 1-9(e). Computation of the water level Z in the region S , with the equation of continuity

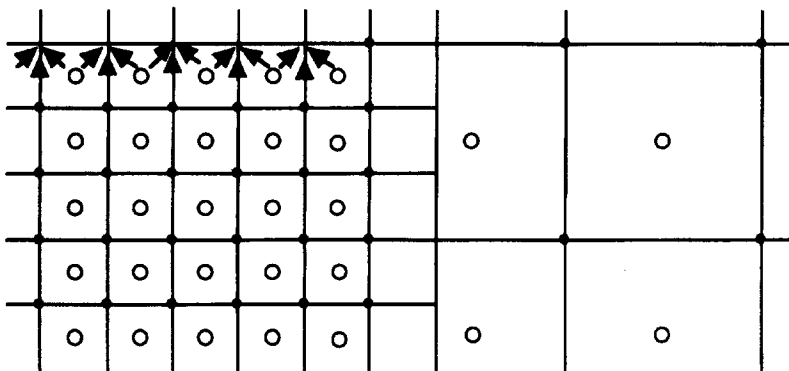


Figure 1-9(f). Computation of the water discharge M in the region S , with the equation of motion.

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.

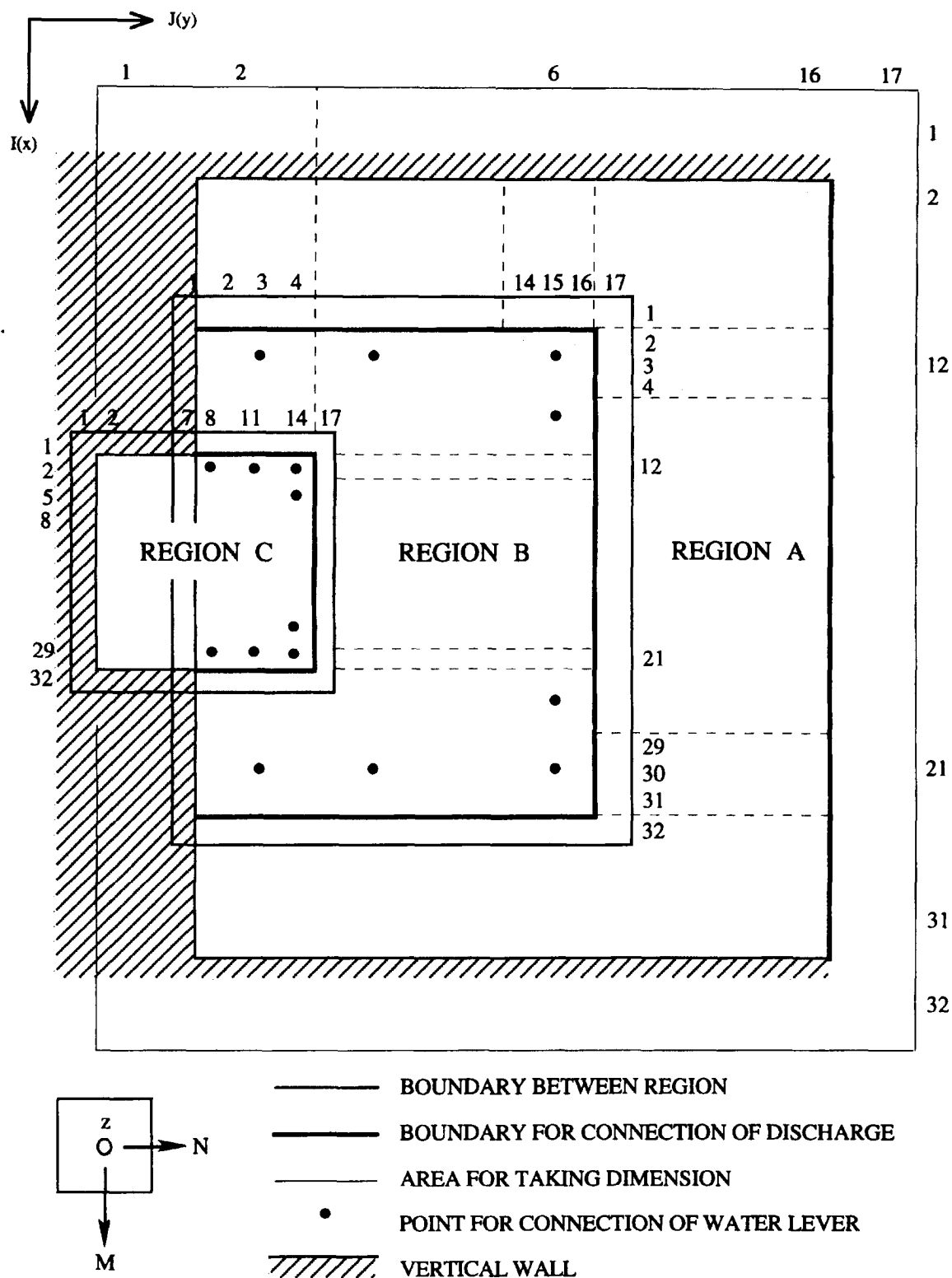


Figure 2-1: A model basin for explanation of the programme

(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.

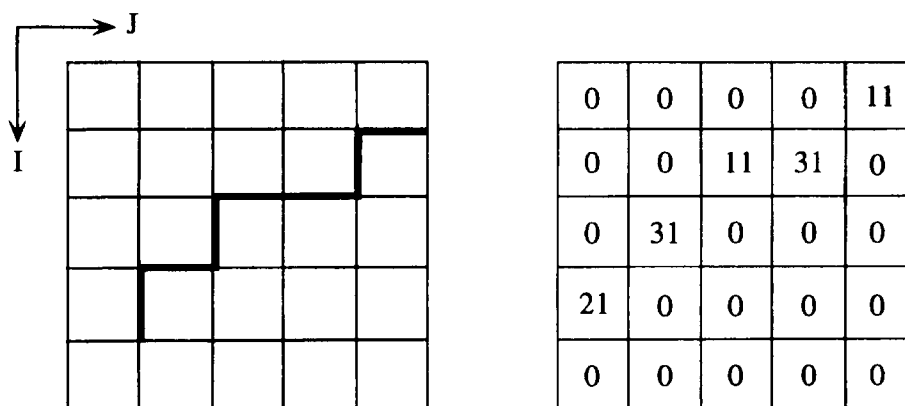


Figure 2-2: An example of the IR map.

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 or 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	ratio 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
C

MAIN PROGRAMME

```

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),HXC(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 DEPTH(32,17,HZA,IRA,IBA,WD,1)
CALL DEPTH(32,17,HZB,IRB,IBB,WD,2)
CALL DEPTH(32,17,HXC,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,HXC,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,HXC)
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,HXC,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+1))
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(IG,JG,2)
WRITE(6,100) KK,(J,J=2,16)
100 FORMAT(1H,2X,'K=',I4,/,6X,15I4)
IG1=IG-1
JG1=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+1))
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	No change
JS, JE	I	(IS, JS) and of end (IE, JE)	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

```

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

      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
DZ1	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 DZ1(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	


```

ROUTINE MOTION(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
DM1=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)
DN1=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.OR.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(IBM.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(IBM.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(IBM.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

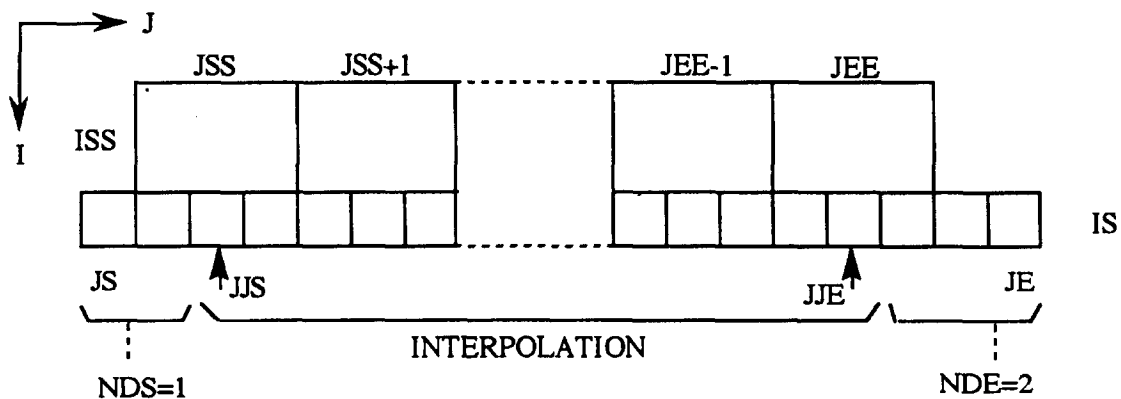
Connection of the water level between computation domains of different Δx and Δt .

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) GO 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

	Page
1. INTRODUCTION	1
2. METHODS OF COMPUTATION	1
2.1 CO-ORDINATES AND AREA FOR COMPUTATION	1
2.2 GOVERNING EQUATIONS AND GRID LENGTH	2
2.3 INITIAL AND BOUNDARY CONDITIONS	4
3. TSUNAMI SIMULATION PROGRAMMES	5
3.1 AN EXAMPLE OF PROGRAMME	5
3.2 VARIABLES AND CONSTANTS IN TSUNAMI SIMULATION	6
3.3 SUBROUTINES IN TSUNAMI SIMULATION	6
3.4 VARIABLES AND CONSTANTS IN INITIAL PROFILE COMPUTATION	7
4. CONCLUDING REMARKS	8

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).

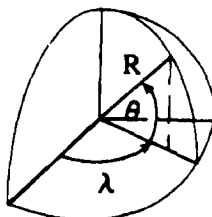


Figure 1. Spherical Co-ordinates

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

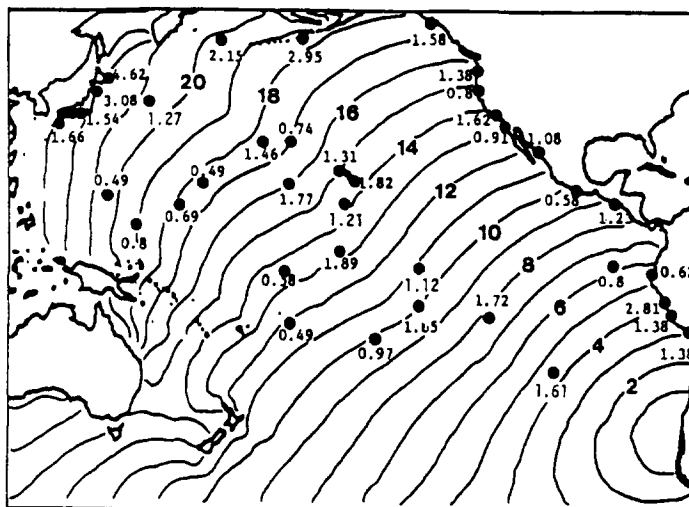


Figure 2. Area for Computation, and Propagation of the Chilean tsunami

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.

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

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

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

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

$$\frac{\eta_{j,m}^{n+\frac{1}{2}} - \eta_{j,m}^{n-\frac{1}{2}}}{\Delta t} + \frac{1}{R \cos \theta_m} \left[\frac{M_{j-\frac{1}{2},m}^n - M_{j+\frac{1}{2},m}^n}{\Delta \lambda} + \frac{N_{j,m}^n \cos \theta_{m+\frac{1}{2}} - N_{j,m}^n \cos \theta_{m-\frac{1}{2}}}{\Delta \theta} \right] = 0 \quad (4)$$

$$\frac{M_{j-\frac{1}{2},m}^{n+1} - M_{j-\frac{1}{2},m}^n}{\Delta t} + \frac{gh_{j+\frac{1}{2},m}}{R \cos \theta_m} \frac{\eta_{j+1,m}^{n+\frac{1}{2}} - \eta_{j,m}^{n+\frac{1}{2}}}{\Delta \lambda} = fN \quad (5)$$

$$\frac{N_{j+\frac{1}{2},m}^{n+1} - N_{j+\frac{1}{2},m}^n}{\Delta t} + \frac{gh_{j+\frac{1}{2},m} \eta_{j,m+1}^{\frac{n+1}{2}} - \eta_{j,m}^{\frac{n+1}{2}}}{R \sin \theta_m} = -fM \quad (6)$$

where

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

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

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

$$\eta_{j,m}^{\frac{n+1}{2}} = \eta_{j,m}^{\frac{n-1}{2}} - R_1 \left[M_{j+\frac{1}{2},m}^n - M_{j-\frac{1}{2},m}^n + N_{j,m+\frac{1}{2}}^n \cos \theta_{m+\frac{1}{2}} - N_{j,m-\frac{1}{2}}^n \cos \theta_{m-\frac{1}{2}} \right] \quad (8)$$

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

$$N_{j,m+\frac{1}{2}}^{n+1} = M_{j,m+\frac{1}{2}}^n - R_4 \cdot h_{j,m+\frac{1}{2}} \left[\eta_{j+1,m}^{\frac{n+1}{2}} - \eta_{j,m}^{\frac{n+1}{2}} \right] + R_5 M' \quad (10)$$

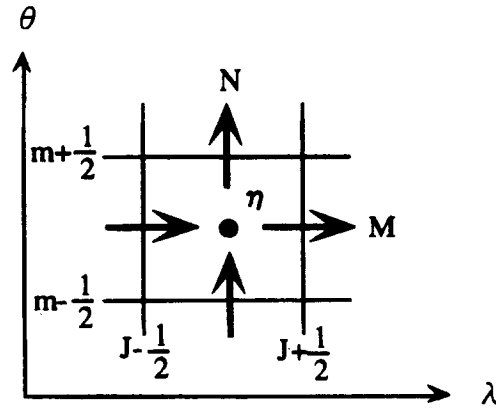


Figure 3. Computation Points for Water Level and Discharge

where R'_n 's are given as follows.

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

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

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

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

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

A point of computation is numbered as (j, m, n) in the (θ , λ , t) directions. The grid lengths are ($\Delta\theta$, $\Delta\lambda$, Δt). In the present computation grid lengths in the latitude and longitude directions are taken equal: i.e. $\Delta s = \Delta\theta = \Delta\lambda$. 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.

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

where K is the Courant number ($= (gh)^{1/2} \Delta t / \Delta x$) 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,

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

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 $\eta \ll h$.

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

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

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

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

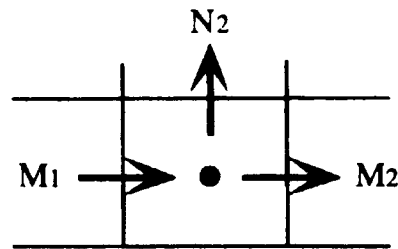


Figure 4. Distinction of Advancing and Receding Waves by the sign of N_2

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.

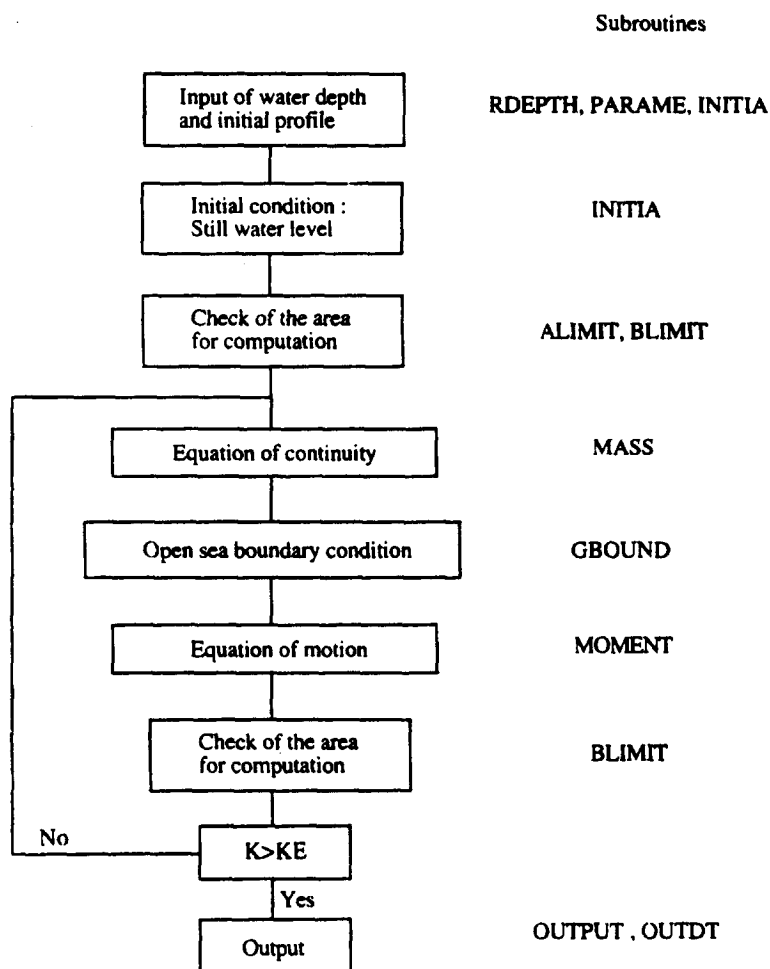


Figure 5. Flow Chart of the Computation

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.
 $C2 = \theta_m$ 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.
INITLA: 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 (δ) in degree
 TH: Strike angle (θ) 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.
 - - -
 Y0, X0: 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.

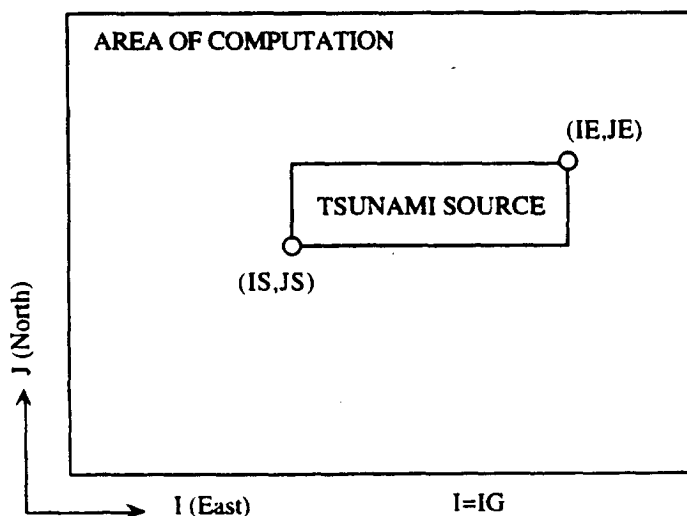


Figure 6. Co-ordinates in the Areas for Computation of Tsunami and Initial Profile

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.

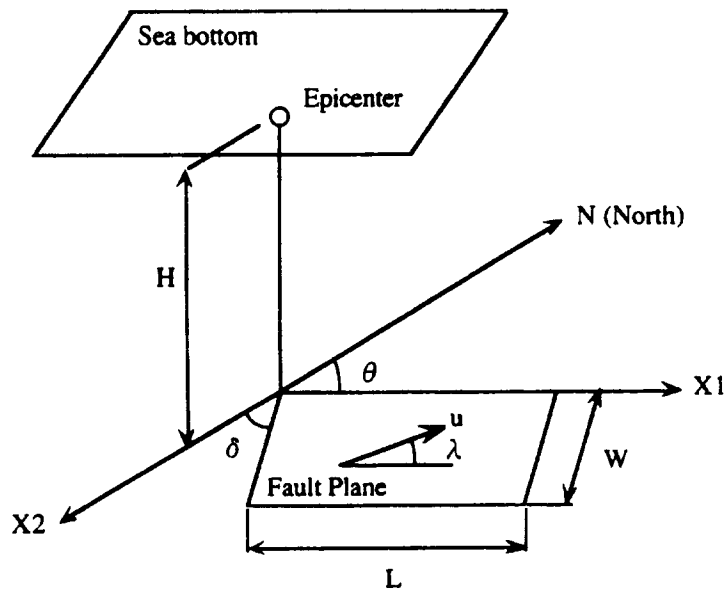


Figure 7. Fault Parameter

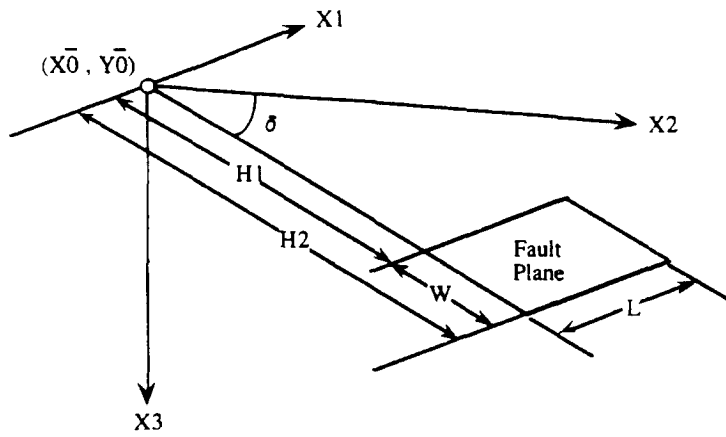


Figure 8 Plan of Fault Plane

Figure 9. Fault Plane in Three Dimensions

```

0010C
0020C ***** TUNAMI-F1 *****
0030C      Tohoku University's Numerical-Analysis Model
0040C      for Investigation of tsunami
0050C      Far-field Tsunami version
0060C
0070C      1990 MARIANA TSUNAMI (DX=5')
0080C      --- LINEAR LONG WAVE THEORY ---
0090C      NUMERICAL EXPERIMENT
0100C      91.4.11
0110C
0120      PARAMETER (IG=480,JG=600)
0130      PARAMETER (DS=5.,DT=10.)
0140      PARAMETER (KS=1,KE=360*10,KD=360*2,KC=1)
0150C
0160      REAL M,N
0170      DOUBLE PRECISION TS,TE,TD
0180      DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG),H(IG,JG)
0190      DIMENSION R1(IG,JG),R2(IG,JG),R3(IG,JG)
0200      DIMENSION R4(IG,JG),R5(IG,JG),R6(JG)
0210      DIMENSION C1(IG),C2(JG),C3(IG),C4(JG)
0220      DIMENSION V1(JG),V2(JG),V3(JG),V4(JG),V5(JG)
0230      DIMENSION ZM(IG,JG),ZN(IG,JG),LF(IG,JG)
0240C
0250      CALL RDEPTH (IG,JG,H)
0260      WRITE(6,255)
0270 255 FORMAT(1H1,5X,"READ DEPTH OK!")
0280      CALL PARAME (IG,JG,IR,JR,ID,JD,DS,DT,
0290      &          R1,R2,R3,R4,R5,R6,C1,C2,C3,C4,
0300      &          V1,V2,V3,V4,V5,H)
0310      CALL INITIA (IG,JG,IR,JR,ID,JD,
0320      &          Z,M,N,H)
0330      CALL ALIMIT (IG,JG,IS,JS,IE,JE,IR,JR,ID,JD)
0340      CALL BLIMIT (IG,JG,IS,JS,IE,JE,Z)
0350C      CALL OUTPUT (IG,JG,IS,JS,IE,JE,0,KD,Z,H)
0360      CALL CLOCK (TS)
0370      DO 20 K=KS,KE
0380      KK=K ; IF(MOD(KK,50).EQ.0)WRITE(6,*)KK
0390      CALL MASS (IG,JG,IS,JS,IE,JE,
0400      &          Z,M,N,R1,R6)
0410      CALL GBOUND (IG,JG,IS,JS,IE,JE,
0420      &          Z,M,N,C1,C2,C3,C4)
0430      CALL MOMENT (IG,JG,IS,JS,IE,JE,
0440      &          Z,M,N,R2,R3,R4,R5,
0450      &          V1,V2)
0460      CALL OUTPUT (IG,JG,IS,JS,IE,JE,KK,KD,Z,H)
0470      CALL BLIMIT (IG,JG,IS,JS,IE,JE,Z)
0480C
0490      CALL MAX (IG,JG,IS,JS,IE,JE,Z,H,ZM,ZN)
0500      CALL POINT (IG,JG,Z,KK,KC)
0510      CALL PROPA (IG,JG,IS,JS,IE,JE,Z,H,LF,KK)
0520C
0530 20 CONTINUE
0540      CALL OUTDT (IG,JG,ZM,ZN,LF)
0550      CALL CLOCK (TE) ; TD=TE-TS
0560      WRITE(6,*)KK,TD
0570      STOP
0580      END
0590C

```

```

graph TD
    Start(( )) --> Input[Data and initial condition input]
    Input --> MainComp[Main computation]
    MainComp --> Output[Output of computed results]
    Output --> End[Main program]

```

```

0600      SUBROUTINE RDEPTH (IG,JG,H) -----
0610C
0620      DIMENSION H(IG,JG)
0630C
0640      DO 10 J=1,JG
0650      READ (20,100) (H(I,J),I=1,IG)
0660 100  FORMAT(20F6.1)
0670      10 CONTINUE
0680      RETURN
0690      END
0700C
0710      SUBROUTINE PARAM (IG,JG,IR,JR,ID,JD,DS,DT, -----
0720      &      R1,R2,R3,R4,R5,R6,C1,C2,C3,C4, -----
0730      &      V1,V2,V3,V4,V5,H)
0740C
0750      PARAMETER (GG=9.8)
0760      PARAMETER (FL=0.0,PP=3.1415926)
0770      PARAMETER (R=6.37E+6,WW=7.2722E-5)
0780      PARAMETER (IS=240,JS=120,IE=360,JE=240)
0790C
0800      DIMENSION R1(IG,JG),R2(IG,JG),R3(IG,JG),R4(IG,JG),R5(IG,JG)
0810      DIMENSION R6(JG),C1(IG),C2(JG),C3(IG),C4(JG)
0820      DIMENSION V1(JG),V2(JG),V3(JG),V4(JG),V5(JG)
0830      DIMENSION H(IG,JG)
0840C
0850      IR=IS
0860      JR=JS
0870      ID=IE
0880      JD=JE
0890      PX=PP/180.
0900      DX=R*(DS/60.)*PX
0910      RR=DT/DX
0920      RS=0.5*GG*RR
0930      RT=0.5*DT*WW
0940      DO 10 J=1,JG
0950      C2(J)=(FL+(DS*J-0.5*DS)/60.)*PX
0960      C4(J)=(FL+DS*J/60.)*PX
0970 10 CONTINUE
0980      DO 11 J=1,JG
0990      R6(J)=COS(C2(J))
1000      V1(J)=RR/R6(J)
1010      V2(J)=RS/R6(J)
1020      V3(J)=RT*SIN(C2(J))
1030      V4(J)=RS
1040      V5(J)=RT*SIN(C4(J))
1050 11 CONTINUE
1060      DO 12 J=1,JG
1070      C4(J)=0.
1080      DO 12 I=1,IG
1090      IF(H(I,J).GT.C4(J))C4(J)=H(I,J)
1100 12 CONTINUE

```

Input of water depth

Setting of parameters

Setting of C1 and C2

Setting of C3 and R6

Setting of C4

```

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

```

2080  SUBROUTINE BLIMIT (IG,JG,IS,JS,IE,JE,Z) -----
2090C
2100  PARAMETER (GX=1.E-4)
2110  DIMENSION Z(IG,JG)
2120C
2130  IF(IS.EQ.2)GOTO 61
2140  L=0
2150  DO 10 J=JS,JE
2160  IF(ABS(Z(IS+2,J)).GT.GX)L=1
2170  10 CONTINUE
2180  IF(L.EQ.1)THEN
2190  IS=IS-1
2200  IF(IS.LE.2)IS=2
2210  ENDIF
2220  61 IF(IE.EQ.IG-1)GOTO 62
2230  L=0
2240  DO 20 J=JS,JE
2250  IF(ABS(Z(IE-2,J)).GT.GX)L=1
2260  20 CONTINUE
2270  IF(L.EQ.1)THEN
2280  IE=IE+1
2290  IF(IE.GE.IG-1)IE=IG-1
2300  ENDIF
2310  62 IF(JS.EQ.2)GOTO 63
2320  L=0
2330  DO 30 I=IS,IE
2340  IF(ABS(Z(I,JS+2)).GT.GX)L=1
2350  30 CONTINUE
2360  IF(L.EQ.1)THEN
2370  JS=JS-1
2380  IF(JS.LE.2)JS=2
2390  ENDIF
2400  63 IF(JE.EQ.JG-1)GOTO 64
2410  L=0
2420  DO 40 I=IS,IE
2430  IF(ABS(Z(I,JE-2)).GT.GX)L=1
2440  40 CONTINUE
2450  IF(L.EQ.1)THEN
2460  JE=JE+1
2470  IF(JE.GE.JG-1)JE=JG-1
2480  ENDIF
2490  64 RETURN
2500  END
2510C

```

Setting of the area for computation.

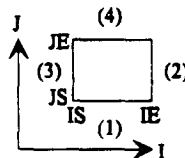
(Enlarge the area if water level is higher than GX(1.0E-4).)

Enlargement in (1) direction.

Enlargement in (2) direction.

Enlargement in (3) direction.

Enlargement in (4) direction.



```

1710     SUBROUTINE INITIA (IG,JG,IS,JS,IE,JE, -----
1720     &                  Z,M,N,H)
1730C
1740     REAL M,N
1750     DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG),H(IG,JG)
1760C
1770     DO 10 J=1,JG
1780     DO 10 I=1,IG
1790     Z(I,J)=0.
1800     M(I,J)=0.
1810     N(I,J)=0.
1820 10 CONTINUE
1830C
1840     CALL DEFORM (IG,JG,IS,JS,IE,JE,Z)
1850C
1860     DO 30 J=JS,JE
1870     DO 30 I=IS,IE
1880     IF(H(I,J).LT.0.) Z(I,J)=0.
1890 30 CONTINUE
1900     RETURN
1910     END
1920C
1930     SUBROUTINE ALIMIT (IG,JG,IS,JS,IE,JE,IR,JR,ID,JD) ---
1940C
1950     IS=IR-1
1960     JS=JR-1
1970     IE=ID+1
1980     JE=JD+1
1990     IF(IS.LE.2) IS=2
2000     IF(JS.LE.2) JS=2
2010     IF(IE.GE.IG-1) IE=IG-1
2020     IF(JE.GE.JG-1) JE=JG-1
2030     WRITE(6,100) IS,JS,IE,JE
2040     RETURN
2050 100 FORMAT(1H , "IS=", I5, 2X, "JS=", I5, 2X, "IE=", I5, 2X, "JE=", I5)
2060     END
2070C

```

Setting of the initial condition

Still water level is assumed.

Input of the initial profile.

Water level is set zero on land.

The area for computation is limited within the area of concern.

(ALIMIT is not necessary.
BLIMIT given below is sufficient)

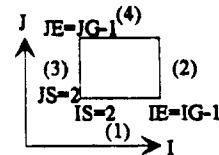
```

2520 SUBROUTINE MASS (IG,JG,IS,JS,IE,JE,Z,M,N,R1,R6) --
2530C
2540 REAL M,N
2550 DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG)
2560 DIMENSION R1(IG,JG),R6(JG)
2570C
2580 DO 10 J=JS,JE
2590 DO 10 I=IS,IE
2600 Z(I,J)=Z(I,J)-R1(I,J)*(M(I,J)-M(I-1,J))
2610 & -R1(I,J)*(N(I,J)*R6(J)-N(I,J-1)*R6(J-1))
2620 10 CONTINUE
2630 DO 20 J=JS,JE
2640 DO 20 I=IS,IE
2650 IF(ABS(Z(I,J)).LT.1.0E-5)Z(I,J)=0.0
2660 20 CONTINUE
2670 RETURN
2680 END
2690C
2700 SUBROUTINE GBOUND (IG,JG,IS,JS,IE,JE, --
2710 & Z,M,N,C1,C2,C3,C4)
2720C
2730 REAL M,N
2740 DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG)
2750 DIMENSION C1(IG),C2(JG),C3(IG),C4(JG)
2760C
2770 IF(JS.LE.2)THEN
2780 DO 10 I=2,IG-1
2790 Z(I,1)=SQRT(N(I,1)**2
2800 & +0.25*(M(I,1)+M(I-1,1))**2)*C1(I)
2810 10 IF(N(I,1).GT.0.)Z(I,1)=-Z(I,1)
2820 ENDIF
2830 IF(IS.LE.2)THEN
2840 DO 20 J=2,JG-1
2850 Z(1,J)=SQRT(M(1,J)**2
2860 & +0.25*(N(1,J)+N(1,J-1))**2)*C2(J)
2870C IF(Z(1,J).GT.1.0)Z(1,J)=1.0
2880C IF(Z(1,J).LT.-1.0)Z(1,J)=-1.0
2890 20 IF(M(1,J).GT.0.)Z(1,J)=-Z(1,J)
2900 ENDIF
2910 IF(JE.GE.JG-1)THEN
2920 DO 30 I=2,IG-1
2930 Z(I,JG)=SQRT(N(I,JG-1)**2
2940 & +0.25*(M(I,JG)+M(I,JG-1))**2)*C3(I)
2950 IF(N(I,JG-1).LT.0.)Z(I,JG)=-Z(I,JG)
2960 30 CONTINUE
2970C
2980 ENDIF
2990 IF(IE.GE.IG-1)THEN
3000 DO 40 J=2,JG-1
3010 Z(IG,J)=SQRT(M(IG-1,J)**2
3020 & +0.25*(N(IG,J)+N(IG,J-1))**2)*C4(J)
3030 40 IF(M(IG-1,J).LT.0.)Z(IG,J)=-Z(IG,J)
3040 ENDIF
3050 Z(1,1)=SQRT(M(1,1)**2+N(1,1)**2)*C1(1)
3060 IF(N(1,1).GT.0.0)Z(1,1)=-Z(1,1)
3070 Z(IG,1)=SQRT(M(IG-1,1)**2+N(IG,1)**2)*C1(IG)
3080 IF(N(IG,1).GT.0.0)Z(IG,1)=-Z(IG,1)
3090 Z(1,JG)=SQRT(M(1,JG)**2+N(1,JG-1)**2)*C3(1)
3100 IF(N(1,JG-1).LT.0.0)Z(1,JG)=-Z(1,JG)
3110 Z(IG,JG)=SQRT(M(IG-1,JG)**2+N(IG,JG-1)**2)*C3(IG)
3120 IF(N(IG,JG-1).LT.0.0)Z(IG,JG)=-Z(IG,JG)
3130 RETURN
3140 END

```

Computation of the equation of continuity.

Setting of the open sea boundary condition.



Setting on the line (1).

Setting on the line (2).

Setting on the line (3).

Setting on the line (4).


```

3150C
3160      SUBROUTINE MOMENT (IG,JG,IS,JS,IE,JE, --
3170      &                  Z,M,N,R2,R3,R4,R5,
3180      &                  V1,V2)
3190C
3200      REAL M,N
3210      DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG)
3220      DIMENSION R2(IG,JG),R3(IG,JG),R4(IG,JG),R5(IG,JG)
3230      DIMENSION V1(JG),V2(JG)
3240C
3250      DO 10 J=JS,JE
3260      DO 10 I=IS,IE
3270      V1(I)=Z(I+1,J)-Z(I,J)
3280      V2(I)=N(I,J-1)+N(I,J)+N(I+1,J-1)+N(I+1,J)
3290      M(I,J)=M(I,J)-R2(I,J)*V1(I)+R3(I,J)*V2(I)
3300  10 CONTINUE
3310      IF(JS.LE.2) THEN
3320      DO 15 I=1,IG-1
3330      15 M(I,1)=M(I,1)-R2(I,1)*(Z(I+1,1)-Z(I,1))
3340      ENDIF
3350      IF(JE.GE.JG-1) THEN
3360      DO 16 I=1,IG-1
3370      16 M(I,JG)=M(I,JG)-R2(I,JG)*(Z(I+1,JG)-Z(I,JG))
3380      ENDIF
3390      IF(IS.LE.2) THEN
3400      DO 17 J=1,JG
3410      17 M(1,J)=M(1,J)-R2(1,J)*(Z(2,J)-Z(1,J))
3420      ENDIF
3430C
3440      DO 20 J=JS,JE
3450      DO 20 I=IS,IE
3460      V1(I)=Z(I,J+1)-Z(I,J)
3470      V2(I)=M(I-1,J)+M(I,J)+M(I-1,J+1)+M(I,J+1)
3480      N(I,J)=N(I,J)-R4(I,J)*V1(I)-R5(I,J)*V2(I)
3490  20 CONTINUE
3500      IF(IS.LE.2) THEN
3510      DO 25 J=1,JG-1
3520      25 N(1,J)=N(1,J)-R4(1,J)*(Z(1,J+1)-Z(1,J))
3530      ENDIF
3540      IF(IE.GE.IG-1) THEN
3550      DO 26 J=1,JG-1
3560      26 N(IG,J)=N(IG,J)-R4(IG,J)*(Z(IG,J+1)-Z(IG,J))
3570      ENDIF
3580      IF(JS.LE.2) THEN
3590      DO 27 I=1,IG
3600      27 N(I,1)=N(I,1)-R4(I,1)*(Z(I,2)-Z(I,1))
3610      ENDIF
3620C
3630      RETURN
3640      END
3650C

```

Computation of the equation of motion.

Computation of M.

Computation of M along the boundary line.

Computation of N.

Computation of N along the boundary line.

```

3660 SUBROUTINE OUTPUT (IG,JG,IS,JS,IE,JE,KK,KD,Z,H) ---
3670C
3680 PARAMETER (GX=1.E-4)
3690 DIMENSION Z(IG,JG),H(IG,JG)
3700 DIMENSION KW(100),KM(23)
3710C
3720 DATA KM/1HJ,1HI,1HH,1HG,1HF,1HE,1HD,1HC,
3730 & 1HB,1HA,1H.,1H1,1H2,1H3,1H4,1H5,
3740 & 1H6,1H7,1H8,1H9,1H0,1H*,1H /
3750 IF(KK.EQ.0)GOTO 120
3760C IF(MOD(KK,180).EQ.0.AND.KK.LE.360*4)THEN
3770C WRITE(26,102)IS,JS,IE,JE
3780C 102 FORMAT(4I10)
3790C DO 15 J=JS,JE
3800C WRITE(26,101)(Z(I,J),I=IS,IE)
3810C 101 FORMAT(20F6.3)
3820C 15 CONTINUE
3830C ENDF
3840 120 IF(MOD(KK,KD).NE.0)RETURN
3850 WRITE(6,410)KK,(KI,KI=120,160,10)
3860 DO 30 KY=1,100
3870 KJ=100-KY+1
3880 J1=6*(KJ-1)+1
3890 J2=J1+5
3900 DO 40 KI=1,80
3910 I1=6*(KI-1)+1
3920 I2=I1+5
3930 L=0
3940 S=0.
3950 DO 41 J=J1,J2
3960 DO 41 I=I1,I2
3970 IF(H(I,J).GT.0.)THEN
3980 L=L+1
3990 S=S+Z(I,J)
4000 ENDF
4010 41 CONTINUE
4020 IF(L.GE.20)THEN
4030 S=S/L
4040 KMM=11+INT(S/0.02)
4050 IF(KMM.GT.21)KMM=21
4060 IF(KMM.LT.1)KMM=1
4070 IF(ABS(S).LT.GX)KMM=23
4080 ELSE
4090 KMM=22
4100 ENDF
4110 KW(KI)=KM(KMM)
4120 40 CONTINUE
4130 JK=(KY+1)/2
4140 WRITE(6,400)JK,(KW(I),I=1,80)
4150 30 CONTINUE
4160 WRITE(6,100)KK,IS,JS,IE,JE
4170 RETURN
4180 100 FORMAT(1H,"K=",I5,2X,"IS,JS,IE,JE=",4I5)
4190 400 FORMAT(1H,I3,2X,80A1)
4200 410 FORMAT(1H1,/,2X,"MAP AT K=",I5,/,3X,4I20)
4210 END
4220C

```

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.

```

4230      SUBROUTINE FILEOT (IG,JG,Z,M,N) ---
4240C
4250      REAL M,N
4260      DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG)
4270C
4280      DO 10 J=1,JG
4290      WRITE(25,100) (Z(I,J),I=1,IG)
4300      WRITE(25,100) (M(I,J),I=1,IG)
4310      WRITE(25,100) (N(I,J),I=1,IG)
4320      10 CONTINUE
4330      100 FORMAT(10E12.5)
4340      RETURN
4350      END
4360C
4370      SUBROUTINE MAX (IG,JG,IS,JS,IE,JE,Z,H,ZM,ZN) ---
4380C
4390      DIMENSION Z(IG,JG),H(IG,JG),ZM(IG,JG),ZN(IG,JG)
4400      DO 10 I=IS,IE
4410      DO 10 J=JS,JE
4420      IF(H(I,J).LT.0.0)GOTO 10
4430      IF(ZM(I,J).LT.Z(I,J))ZM(I,J)=Z(I,J)
4440      IF(ZN(I,J).GT.Z(I,J))ZN(I,J)=Z(I,J)
4450      10 CONTINUE
4460      RETURN
4470      END
4480C
4490      SUBROUTINE POINT (IG,JG,Z,KK,KC) ---
4500C
4510      DIMENSION Z(IG,JG)
4520      IF(MOD(KK,KC).NE.0)RETURN
4530      WRITE(24,100)Z(211,405),Z(252,417),
4540      &Z(249,417),Z(246,418),
4550      &Z(219,415),Z(238,419),
4560      &Z(237,397),Z(266,325),
4570      &Z(133,377),Z(92,315),
4580      &Z(50,292),
4590      &Z(480,339),Z(382,89),
4600      &Z(480,231),Z(269,161)
4610      100 FORMAT(15F9.5)
4620      RETURN
4630      END
4640C

```

Output of the water level and discharge.

Computation of the highest and lowest water levels.

Output of the time history of water level.

```

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) --- Computation of the initial profile
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*(XO-XO)*COS(E*YO)/180.0 ; YL=A*RR*(YO-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

```

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*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.

```

5740 SUBROUTINE UDCAL (X1,X2,X3,C,CC,DP,F) --
5750 REAL K
5760C
5770 SN=SIN(DP) ; CS=COS(DP)
5780 C1=C ; C2=CC*CS ; C3=CC*SN
5790 R=SQRT((X1-C1)**2+(X2-C2)**2+(X3-C3)**2)
5800 Q=SQRT((X1-C1)**2+(X2-C2)**2+(X3+C3)**2)
5810 R2=X2*SN-X3*CS
5820 R3=X2*CS+X3*SN
5830 Q2=X2*SN+X3*CS
5840 Q3=-X2*CS+X3*SN
5850 H=SQRT(Q2**2+(Q3+CC)**2)
5860 K=SQRT((X1-C1)**2+Q2**2)
5870 A1=LOG(R+X1-C1)
5880 A2=LOG(Q+X1-C1)
5890 B1=Q*(Q+X1-C1)
5900 B2=R*(R+X1-C1)
5910 B3=Q*(Q+Q3+CC)
5920 D1=X1-C1 ; D2=X2-C2 ; D3=X3-C3
5930 D4=X3+C3 ; D5=R3-CC ; D6=Q3+CC
5940 T1=ATN(D1*D2,(H+D4)*(Q+H))
5950 T2=ATN(D1*D5,R2*R)
5960 T3=ATN(D1*D6,Q2*Q)
5970 F=SN*(D2*(2*D3/B2+4*D3/B1-4*C3*X3*D4*(2*Q+D1)/(B1**2*Q))
5980 & -6*T1+3*T2-6*T3)
5990 & +CS*(A1-A2-2*(D3**2)/B2
6000 & -4*(D4**2-C3*X3)/B1-4*C3*X3*D4**2*(2*Q+X1-C1)/(B1**2*Q))
6010 & +6*X3*(CS*SN*(2*D6/B1+D1/B3)-Q2*(SN**2-C3**2)/B1)
6020 RETURN
6030 END
6040C
6050 REAL FUNCTION ATN (AX,AY)
6060 DATA GX/1.0E-6/
6070 AAX=ABS(AX) ; AAY=ABS(AY)
6080 P=AX*AY
6090 IF(AAX.LE.GX.AND.AAY.LE.GX)GOTO 10
6100 SR=ATAN2(AAX,AAY)
6110 ATN=SIGN(SR,P)
6120 RETURN
6130 10 WRITE(6,100)AX,AY
6140 100 FORMAT(1H,"ATAN -- AX=",E15.7,2X,"AY=",E15.7)
6150 ATN=0.2
6160 RETURN
6170 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

	Page
TUNAMI-N1: LINEAR THEORY WITH CONSTANT GRIDS	1
TUNAMI-N2: LINEAR THEORY IN DEEP SEA, SHALLOW WATER THEORY IN SHALLOW SEA AND RUNUP ON LAND WITH CONSTANT GRIDS	7
TUNAMI-N3: LINEAR THEORY WITH VARYING GRIDS	16

```

0010*#RUN : LAP OPT=2 F=/SNRK/MM2.7(02)
0020C
0030C ***** TUNAMI-N1 *****
0040C   Tohoku University's Numerical-Analysis Model
0050C     for Investigation of tsunami
0060C     Near-field Tsunami version
0070C
0080C   1968 TOKACHI-OKI EARTHQUAKE TSUNAMI
0090C     USING BY LINEAR LONG WAVE THEORY
0100C
0110C     CODED BY
0120C
0130C     F.IMAMURA, TOHOKU UNIV., JAPAN
0140C
0150C           1991.6.25
0160C
0170C   Z; WATER SURFACE LEVEL  M,N; WATER DISCHARGE
0180C   H; STILL WATER DEPTH  ZD; TOTAL WATER DEPTH
0190C IF,JF; DIMENSION      DX,DT; SPATIAL GRID & TIME STEP SIZE
0200C   G; GRAVITATIONAL ACCES.  KL; TOTAL TIME STEP
0210C IP,JP; POSITION OF OUTPUT POINT
0220C IO,J0; ORIGIN OF EPICENTER
0230C
0240   REAL M,N
0250   PARAMETER(IF=260,JF=180,IO=111,J0=188)
0260   PARAMETER(DX=2700.0,DT=5.0,G=9.8,KD=1,KL=360)
0270   DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
0280   DIMENSION ZM(IF,JF),ZD(IF,JF),IH(IF,JF)
0290C   DIMENSION IP(16),JP(16),PZ(16)
0300C   DATA IP/62,76,86,96,104,114,124,132,142,152,162,172,182
0310C   & ,192,202,212/
0320C   DATA JP/60,60,62,66,72,76,80,84,88,86,80,78,78,80,78/
0330C
0340   R=DT/DX ; KK=0
0350C
0360C ***** INITIAL CONDITION *****
0370C
0380   CALL DEPTH(IF,JF,H,IH)
0390   CALL INTL(IF,JF,Z,M,N)
0400   CALL DEFORM(IF,JF,Z,IO,J0,DX)
0410C   CALL OUT6 (IF,JF,Z,H,KK)
0420   CALL PTIME(CT1)
0430C
0440C ***** MAIN CALCULATION *****
0450C
0460   DO 10 K=1,KL
0470     KK=K
0480     IF(MOD(KK,10).EQ.0)WRITE(6,*)KK
0490     CALL MASS(IF,JF,Z,M,N,H,R,ZD)
0500     CALL OPEN(IF,JF,Z,M,N,H)
0510     CALL MOMENT(IF,JF,Z,M,N,H,R,G)
0520     CALL ZMAX(IF,JF,Z,ZM)
0530C     CALL OUT3 (IF,JF,16,Z,IP,JP,PZ)
0540     IF(MOD(K,50).NE.0)GOTO 20
0550     CALL OUT6(IF,JF,Z,H,KK)
0560C     CALL OUT1 (IF,JF,Z)
0570 20 CALL CHANGE(IF,JF,Z,M,N)
0580 10 CONTINUE
0590C
0600C   CALL OUT66 (IF,JF,ZM,H)
0610C   CALL OUT11 (IF,JF,ZM)
0620   CALL PTIME(CT2)
0630   CT=CT2-CT1
0640   WRITE(6,100)CT
0650 100 FORMAT(1H,3X,"CPU TIME =",E12.4)
0660   STOP
0670   END
0680C
0690C ***** READ DEPTH DATA *****
0700C
0710   SUBROUTINE DEPTH(IF,JF,H,IH)
0720C

```

```

0730  DIMENSION H(IF,JF),IH(IF,JF)
0740  L=JF/10+1
0750  IF(MOD(JF,10).EQ.0)L=L-1
0760  DO 10 LL=1,L
0770    J1=10*(LL-1)+1
0780    J2=J1+9
0790    IF(J2.GT.JF)J2=JF
0800    DO 10 I=1,IF
0810      READ(2,100)(IH(I,J),J=J1,J2)
0820 100  FORMAT(5X,10I5)
0830      DO 10 J=J1,J2
0840        H(I,J)=FLOAT(IH(I,J))
0850        IF(IH(I,J).GT.0.0.AND.IH(I,J).LT.500)H(I,J)=500.0
0860 10 CONTINUE
0870  RETURN
0880  END
0890C
0900C ***** INITIAL CONDITION *****
0910C
0920  SUBROUTINE INTL(IF,JF,Z,M,N)
0930C
0940  REAL M,N
0950  DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
0960  DO 10 K=1,2
0970    DO 10 J=1,JF
0980      DO 10 I=1,IF
0990        Z(I,J,K)=0.0
1000        M(I,J,K)=0.0
1010        N(I,J,K)=0.0
1020 10 CONTINUE
1030  RETURN
1040  END
1050C
1060C ***** MASS CONSERVATION *****
1070C
1080  SUBROUTINE MASS(IF,JF,Z,M,N,H,R,ZD)
1090C
1100  REAL M,N
1110  DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
1120  DIMENSION ZD(IF,JF)
1130  DATA GX/1.0E-10/
1140  DO 10 J=2,JF
1150    DO 10 I=2,IF
1160      IF(H(I,J).LT.0.0)GO TO 10
1170      ZZ=Z(I,J,1)-R*(M(I,J,1)-M(I-1,J,1)+N(I,J,1)-N(I,J-1,1))
1180      ZZ=ZZ+ZD(I,J)
1190      IF(ABS(ZZ).LT.GX)ZZ=0.0
1200      Z(I,J,2)=ZZ
1210 10 CONTINUE
1220  RETURN
1230  END
1240C
1250C ***** MOMENTUM CONSERVATION *****
1260C
1270  SUBROUTINE MOMENT(IF,JF,Z,M,N,H,R,G)
1280C
1290  REAL M,N
1300  DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
1310  DATA GX/1.0E-10/
1320  DO 10 J=1,JF
1330    DO 10 I=1,IF
1340      IF(I.EQ.IF)GOTO 20
1350      IF(H(I,J).LT.0.0)GO TO 10
1360      IF(H(I+1,J).LT.0.0)GO TO 20
1370      HM=0.5*(H(I,J)+H(I+1,J))
1380      XM=M(I,J,1)-G*R*HM*(Z(I+1,J,2)-Z(I,J,2))
1390      IF(ABS(XM).LT.GX)XM=0.0
1400      M(I,J,2)=XM
1410 20  IF(J.EQ.JF)GOTO 10
1420      IF(H(I,J+1).LT.0.0)GO TO 10
1430      HN=0.5*(H(I,J)+H(I,J+1))
1440      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

```

```

2170    J=1
2180    IF(KK.EQ.2)J=JF
2190    DO 10 I=2,IF
2200      IF(H(I,J).LT.0.0)GO TO 10
2210      CC=SQRT(9.8*H(I,J))
2220      UU=0.5*ABS(M(I,J,2)+M(I-1,J,2))
2230      IF(J.EQ.1)UU=SQRT(UU**2+N(I,J,2)**2)
2240      IF(J.EQ.JF)UU=SQRT(UU**2+N(I,J-1,2)**2)
2250      ZZ=UU/CC
2260      IF(J.EQ.1.AND.N(I,J,2).GT.0.0)ZZ=-ZZ
2270      IF(J.EQ.JF.AND.N(I,J-1,2).LT.0.0)ZZ=-ZZ
2280      Z(I,J,2)=ZZ
2290  10 CONTINUE
2300    DO 20 KK=1,2
2310      I=1
2320      IF(KK.EQ.2)I=JF
2330      DO 20 J=2,JF
2340        IF(H(I,J).LT.0.0)GO TO 20
2350        CC=SQRT(9.8*H(I,J))
2360        UU=0.5*ABS(N(I,J,2)+N(I,J-1,2))
2370        IF(I.EQ.1)UU=SQRT(UU**2+M(I,J,2)**2)
2380        IF(I.EQ.IF)UU=SQRT(UU**2+M(I-1,J,2)**2)
2390        ZZ=UU/CC
2400        IF(I.EQ.1.AND.M(I,J,2).GT.0.0)ZZ=-ZZ
2410        IF(I.EQ.IF.AND.M(I-1,J,2).LT.0.0)ZZ=-ZZ
2420        Z(I,J,2)=ZZ
2430  20 CONTINUE
2440    RETURN
2450    END
2460C
2470C *** EXCHANGE FOR LAST STEP DATA TO NEXT STEP DATA *****
2480C
2490    SUBROUTINE CHANGE(IF,JF,Z,M,N)
2500C
2510    REAL M,N
2520    DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
2530    DO 10 J=1,JF
2540      DO 10 I=1,IF
2550        Z(I,J,1)=Z(I,J,2)
2560        M(I,J,1)=M(I,J,2)
2570        N(I,J,1)=N(I,J,2)
2580  10 CONTINUE
2590    RETURN
2600    END
2610C
2620C
2630C *** CALCULATOIN FOR INITIAL TSUNAMI SOURCE *****
2640C    (= SEA BOTTOM DEFORMATION)
2650C
2660C    RR; RADIUS OF EARTH
2670C    L ; FAULT LENGTH,    W ; FAULT WIDTH
2680C    TH; DIP DIRECTION,   DL; DIP ANGLE
2690C    RD; SLIP ANGLE      D ; DISLOCATION
2700C    HH; DEPTH
2710C
2720    SUBROUTINE DEFORM(IF,JF,Z,I0,J0,DX)
2730    REAL L
2740    PARAMETER(A=3.141592,B=4.848E-06,RR=6.37E+6,E=1.7453E-2)
2750    PARAMETER(L=188.2E+3,W=92.0E+3,TH=145.0,DL=20.0)
2760    PARAMETER(RD=148.0,D=3.2,HH=1.0E+3)
2770    DIMENSION Z(IF,JF,2)
2780    XL=DX*(I0-1); YL=DX*(J0-1); H1=HH/SIN(E*DL)
2790    H2=HH/SIN(E*DL)+W; DS=D*COS(E*RD); DD=D*SIN(E*RD)
2800    DO 10 I=1,IF
2810      DO 10 J=1,JF
2820        XX=DX*(I-1); YY=DX*(J-1)
2830        X1=(XX-XL)*SIN(E*TH)+(YY-YL)*COS(E*TH)-L/2.0
2840        X2=(XX-XL)*COS(E*TH)-(YY-YL)*SIN(E*TH)+HH/TAN(E*DL)
2850        X3=0.0
2860        CALL USCAL(X1,X2,X3,L/2.,H2,E*DL,F1)
2870        CALL USCAL(X1,X2,X3,L/2.,H1,E*DL,F2)
2880        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)

```

```

3610  RETURN
3620  END
3630C
3640  REAL FUNCTION ATN(AX,AY)
3650  DATA GX/1.0E-6/
3660  AAX=ABS(AX) ; AAY=ABS(AY)
3670  P=AX*AY
3680  IF(AAX.LE.GX.AND.AAY.LE.GX)GOTO 10
3690  SR=ATAN2(AAX,AAY)
3700  ATN=SIGN(SR,P)
3710  RETURN
3720 10 WRITE(6,100)AX,AY
3730 100 FORMAT(1H,"ATAN --  AX=",E15.7,2X,"AY=",E15.7)
3740  ATN=0.2
3750  END
3760C
3770  SUBROUTINE ZMAX(IF,JF,Z,ZM)
3780  DIMENSION Z(IF,JF),ZM(IF,JF)
3790C
3800  DO 10 I=1,IF
3810    DO 10 J=1,JF
3820      IF(ZM(I,J).LT.Z(I,J))ZM(I,J)=Z(I,J)
3830 10 CONTINUE
3840  RETURN
3850  END

```

```

0010*#RUN : IAP OPT=2 F=/SNRK/GD-MK(10) F=/SNRK/GD-BT1(11);
0020*#      F=/SNRK/TBN-MYK(12)
0030C
0040C ***** TUNAMI-N2 *****
0050C   Tohoku University's Numerical-Analysis Model
0060C   for Investigation of tsunami
0070C   Near-field Tsunami version
0080C
0090C   FOR 1896 MEIJI SANTRIKU TSUNAMI
0100C   AT MIYAKO BAY (DX=50M)
0110C   WITH SHALLOW WATER THEORY
0120C   INCLUDING EFFECTS OF RUNUP AND STRUCTURES
0130C
0140C           1991.1.25
0150C   BY
0160C
0170C   F.IMAMURA, TOHOKU UNIV., JAPAN
0180C
0190C   Z; WATER SURFACE LEVEL  M,N; WATER DISCHARGE
0200C   HM,HN; STILL WATER DEPTH AT POINT OF WATER DISCHARGE
0210C   DM,DN; TOTAL WATER DEPTH AT POINT OF WATER DISCHARGE
0220C   HZ; STILL WATER DEPTH  ZD; TOTAL WATER DEPTH
0230C   IF,JF; DIMENSION PARAME  DX,DT; SPATIAL GRID & TIME STEP SIZE
0240C   G; GRAVITATIONAL ACCES.  KL; TOTAL TIME STEP
0250C   IR; MAP OF WAVE-BRAKER  BT; HEIGHT OF WAVE-BRAKER
0260C   IP,JP; POSITION OF OUTPUT POINT
0270C   ZM; MAXIMUM WATER LEVEL
0280C
0290   REAL M,N
0300   PARAMETER(IF=170,JF=150)
0310   DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),HZ(IF,JF)
0320   DIMENSION DZ(IF,JF,2),DM(IF,JF,2),DN(IF,JF,2)
0330   DIMENSION HM(IF,JF),HN(IF,JF),B(1800,14),ZM(IF,JF)
0340   DIMENSION IP(13),JP(13),PZ(13)
0350   DIMENSION IR(IF,JF),BT(10)
0360   DATA IP/14,27,50,62,78,114,123,144,144,132,116,102,85/
0370   DATA JP/73,80,79,78,79,52,40,48,72,71,78,85,103/
0380   DATA BT/5.26,6.70,8.0,5.5,6.0,10.0,10.0,10.0,10.0,10.0/
0390   G=9.8 ; DX=50.0 ; DT=1.0 ; R=DT/DX
0400   KL=5400
0410C
0420C ***** INITIAL CONDITION *****
0430C
0440   CALL READ(10,11,IF,JF,HZ,IR)
0450   CALL DATA(12,B)
0460   CALL HMN(IF,JF,HZ,HM,HN,IR,BT)
0470   CALL INTL(IF,JF,Z,M,N,DZ,HZ)
0480   CALL PTIME(CT1)
0490C
0500C ***** MAIN CALCULATION *****
0510C
0520   DO 10 K=1,KL
0530     KK=K
0540     CALL NLMASS(IF,JF,Z,M,N,DZ,HZ,R)
0550     CALL BNC(IF,JF,Z,B,2,101,150,KK,DT)
0560     CALL NLMMT(IF,JF,Z,M,N,DZ,DM,DN,HZ,HM,HN,R,DT,0.025,IR)
0570     CALL MAX(IF,JF,Z,ZM)
0580     IF(MOD(K,30).NE.0)GOTO 20
0590     CALL POINT(17,IF,JF,13,Z,IP,JP,PZ)
0600     CALL OUT1(15,IF,JF,Z,M,N,DM,DN,HZ)
0610     CALL OUT6(IF,JF,Z,HZ,KK)
0620 20 CALL CHANGE(IF,JF,Z,M,N,DZ)
0630 10 CONTINUE
0640     CALL OUT3(16,IF,JF,ZM)
0650C   CALL OUT4 (18,IF,JF,Z,M,N,HZ,ZM)
0660     CALL PTIME(CT2)
0670     CT=CT2-CT1
0680     WRITE(6,180)CT
0690 180 FORMAT(1H,3X,"CPU TIME=",E12.4)
0700     STOP
0710     END
0720C

```



```

0730C ***** READ DEPTH DATA *****
0740C
0750 SUBROUTINE READ(N1,N2,IF,JF,HZ,IR)
0760 DIMENSION HZ(IF,JF),IR(IF,JF)
0770 L=IF/10
0780 IF(MOD(IF,10).NE.0)L=L+1
0790 DO 10 LL=1,L
0800 I1=10*(LL-1)+1
0810 I2=I1+9
0820 IF(I2.GE.IF)I2=IF
0830 DO 10 J=1,JF
0840 READ(N1,100)(HZ(I,JF-J+1),I=I1,I2)
0850 READ(N2,200)(IR(I,JF-J+1),I=I1,I2)
0860 100 FORMAT(5X,10F5.1)
0870 200 FORMAT(4X,10I5)
0880 10 CONTINUE
0890 RETURN
0900 END
0910C
0920C ***** INITIAL CONDITION *****
0930C
0940 SUBROUTINE INTL(IF,JF,Z,M,N,D,H)
0950 REAL M,N
0960 DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
0970 DIMENSION H(IF,JF),D(IF,JF,2)
0980 DO 10 K=1,2
0990 DO 10 J=1,JF
1000 DO 10 I=1,IF
1010 Z(I,J,K)=0.0
1020 M(I,J,K)=0.0
1030 N(I,J,K)=0.0
1040 D(I,J,K)=H(I,J)
1050 IF(H(I,J).GT.0.0) GOTO 10
1060 D(I,J,K)=0.0
1070 Z(I,J,K)=-H(I,J)
1080 10 CONTINUE
1090 RETURN
1100 END
1110C
1120C ***** CAL. OF WATER DEPTH AT POINT OF DISCHARGE *****
1130C
1140 SUBROUTINE HMN(IF,JF,HZ,HM,HN,IR,BT)
1150 DIMENSION HZ(IF,JF),HM(IF,JF),HN(IF,JF),IR(IF,JF),BT(10)
1160 DO 10 J=1,JF
1170 DO 10 I=1,IF
1180 IF(I.EQ.IF)GOTO 11
1190 HM(I,J)=0.5*(HZ(I,J)+HZ(I+1,J))
1200 GOTO 12
1210 11 HM(I,J)=HZ(I,J)
1220 12 IF(J.EQ.JF)GOTO 13
1230 HN(I,J)=0.5*(HZ(I,J)+HZ(I,J+1))
1240 GOTO 10
1250 13 HN(I,J)=HZ(I,J)
1260 10 CONTINUE
1270 DO 20 I=1,IF
1280 DO 20 J=1,JF
1290 IRR=MOD(IR(I,J),10)
1300 IRM=IR(I,J)/10
1310 IF(IRM.EQ.0) GOTO 20
1320 IF(IRM.EQ.2) GOTO 22
1330 HM(I,J)=-BT(IRR)
1340 IF(IRM.EQ.1) GOTO 23
1350 22 HN(I,J)=-BT(IRR)
1360 23 IF(IRM.EQ.2) GOTO 24
1370 IF(HM(I,J).GT.HZ(I,J))HM(I,J)=HZ(I,J)
1380 IF(HM(I,J).GT.HZ(I+1,J))HM(I,J)=HZ(I+1,J)
1390 IF(IRM.EQ.1) GOTO 20
1400 24 IF(HN(I,J).GT.HZ(I,J))HN(I,J)=HZ(I,J)
1410 IF(HN(I,J).GT.HZ(I,J+1))HN(I,J)=HZ(I,J+1)
1420 20 CONTINUE
1430 RETURN
1440 END

```

```

1450C
1460C ***** MASS CONSERVATION *****
1470C
1480 SUBROUTINE NLMASS(IF,JF,Z,M,N,DZ,HZ,R)
1490 REAL M,N
1500 DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
1510 DIMENSION DZ(IF,JF,2),HZ(IF,JF)
1520 DATA GX,GY/1.0E-5,1.0E-10/
1530 DO 10 J=2,JF
1540   DO 10 I=3,IF
1550     IF(HZ(I,J).LT.-9.9)GOTO 11
1560     XM=0.0
1570     XN=0.0
1580     IF(I.NE.1)XM=M(I-1,J,1)
1590     IF(J.NE.1)XN=N(I,J-1,1)
1600     ZZZ=Z(I,J,1)-R*(M(I,J,1)-XM+N(I,J,1)-XN)
1610     IF(ABS(ZZZ).LT.GY)ZZZ=0.0
1620     DD=ZZZ+HZ(I,J)
1630     IF(DD.LT.GX)GOTO 11
1640     DZ(I,J,2)=DD
1650     Z(I,J,2)=ZZZ
1660     GOTO 10
1670 11 DD=0.0
1680     DZ(I,J,2)=DD
1690     Z(I,J,2)=DD-HZ(I,J)
1700 10 CONTINUE
1710 RETURN
1720 END
1730C
1740C ***** MOMENTUM CONSERVATION *****
1750C
1760 SUBROUTINE NLMMT(IF,JF,Z,M,N,DZ,DM,DN,HZ,HM,HN,R,DT,FM,IR)
1770 REAL M,N
1780 DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
1790 DIMENSION DZ(IF,JF,2),DM(IF,JF,2),DN(IF,JF,2)
1800 DIMENSION HZ(IF,JF),HM(IF,JF),HN(IF,JF),IR(IF,JF)
1810 DATA GG,GX/9.8,1.0E-5/
1820C
1830C ----- CAL. OF TOTAL DEPTH AT POINT OF DISCHARGE -----
1840C
1850 DO 10 I=2,IF
1860   DO 10 J=2,JF
1870     IRR=IR(I,J)/10
1880     IF(IRR.EQ.1.OR.IRR.EQ.3)GO TO 11
1890     DM2=0.5*(DZ(I,J,2)+DZ(I+1,J,2))
1900     DM1=0.25*(DZ(I,J,2)+DZ(I,J,1)+DZ(I+1,J,2)+DZ(I+1,J,1))
1910     GO TO 12
1920 11 IF(DZ(I,J,1))13,13,14
1930 13 IF(DZ(I+1,J,1))15,15,16
1940 14 IF(DZ(I+1,J,1))17,17,18
1950 16 IF(Z(I+1,J,1)+HM(I,J))15,15,19
1960 17 IF(Z(I,J,1)+HM(I,J))15,15,20
1970 18 IF(Z(I,J,1)+HM(I,J))21,21,22
1980 21 IF(Z(I+1,J,1)+HM(I,J))15,15,19
1990 22 IF(Z(I+1,J,1)+HM(I,J))20,20,23
2000 15 DM2=0.0
2010 DM1=0.0
2020 GO TO 12
2030 19 DM2=Z(I+1,J,1)+HM(I,J)
2040 DM1=DM2
2050 GO TO 12
2060 20 DM2=Z(I,J,1)+HM(I,J)
2070 DM1=DM2
2080 GO TO 12
2090 23 DM2=0.5*(Z(I,J,1)+Z(I+1,J,1))+HM(I,J)
2100 DM1=DM2
2110 12 IF(IRR.GE.2)GO TO 112
2120 DN2=0.5*(DZ(I,J,2)+DZ(I,J+1,2))
2130 DN1=0.25*(DZ(I,J,2)+DZ(I,J,1)+DZ(I,J+1,2)+DZ(I,J+1,1))
2140 GO TO 24
2150 112 IF(DZ(I,J,1))113,113,114
2160 113 IF(DZ(I,J+1,1))115,115,116

```

```

2170 114 IF(DZ(I,J+1,1))117,117,118
2180 116 IF(Z(I,J+1,1)+HN(I,J))115,115,119
2190 117 IF(Z(I,J,1)+HN(I,J))115,115,120
2200 118 IF(Z(I,J,1)+HN(I,J))121,121,122
2210 121 IF(Z(I,J+1,1)+HN(I,J))115,115,119
2220 122 IF(Z(I,J+1,1)+HN(I,J))120,120,123
2230 115 DN2=0.0
2240 DN1=0.0
2250 GO TO 24
2260 119 DN2=Z(I,J+1,1)+HN(I,J)
2270 DN1=DN2
2280 GO TO 24
2290 120 DN2=Z(I,J,1)+HN(I,J)
2300 DN1=DN2
2310 GO TO 24
2320 123 DN2=0.5*(Z(I,J,1)+DZ(I,J+1,1))+HN(I,J)
2330 DN1=DN2
2340 24 IF(DM1.LT.GX)DM1=0.0
2350 IF(DM2.LT.GX)DM2=0.0
2360 IF(DN1.LT.GX)DN1=0.0
2370 IF(DN2.LT.GX)DN2=0.0
2380 DM(I,J,1)=DM1
2390 DM(I,J,2)=DM2
2400 DN(I,J,1)=DN1
2410 DN(I,J,2)=DN2
2420 10 CONTINUE
2430C
2440C ----- CAL. OF LINEAR TERMS (X-DIRECTION) -----
2450C
2460 FN=0.5*DT*GG*FM**2
2470 DO 200 I=2,IF
2480 DO 200 J=2,JF
2490 IF(HZ(I,J).LT.-8.1)GO TO 200
2500 IF(I.EQ.IF)GO TO 30
2510 IRR=IR(I,J)/10
2520 IF(IRR.EQ.1.OR.IRR.EQ.3)GO TO 60
2530 IF(HM(I,J).LT.-8.1)GO TO 30
2540 IF(DZ(I,J,2))31,31,32
2550 31 IF(DZ(I+1,J,2))30,30,34
2560 32 IF(DZ(I+1,J,2))35,35,36
2570 34 IF(Z(I+1,J,2)+HZ(I,J))30,30,37
2580 35 IF(Z(I,J,2)+HZ(I+1,J))30,30,38
2590 36 DD=DM(I,J,2)
2600 GO TO 39
2610 37 DD=Z(I+1,J,2)+HZ(I,J)
2620 GO TO 39
2630 38 DD=Z(I,J,2)+HZ(I+1,J)
2640 39 XNN=0.25*(N(I,J,1)+N(I+1,J,1)+N(I,J-1,1)+N(I+1,J-1,1))
2650 DF=DD
2660 IF(DF.LT.1.0E-2)DF=1.0E-2
2670 FF=FN*SQRT(M(I,J,1)**2+XNN**2)/DF**(7.0/3.0)
2680 IF(DD.LT.GX)GO TO 30
2690 XM=(1.0-FF)*M(I,J,1)-GG*R*DD*(Z(I+1,J,2)-Z(I,J,2))
2700C
2710C ----- CAL. OF NON-LINEAR TERMS (CONVECTION TERMS) -----
2720C
2730 IF(I.LE.6.OR.J.LE.6)GO TO 40
2740 IF(DM(I,J,1).LT.GX)GO TO 40
2750 IF(M(I,J,1))41,41,42
2760 41 IF(DM(I+1,J,1).LT.GX)GO TO 40
2770 IF(DZ(I+2,J,2).LT.GX)GO TO 40
2780 IF(DZ(I+1,J,2).LT.GX)GO TO 40
2790 IF(IR(I+1,J)/10.EQ.1.OR.IR(I+1,J)/10.EQ.3)GO TO 1001
2800 XM=XM-R*(M(I+1,J,1)**2/DM(I+1,J,1)-M(I,J,1)**2/DM(I,J,1))
2810 GO TO 43
2820 1001 XM=XM-R*(-M(I,J,1)**2/DM(I,J,1))
2830 GO TO 43
2840 42 IF(DM(I-1,J,1).LT.GX)GO TO 40
2850 IF(DZ(I-1,J,2).LT.GX)GO TO 40
2860 IF(DZ(I,J,2).LT.GX)GO TO 40
2870 IF(IR(I-1,J)/10.EQ.1.OR.IR(I-1,J)/10.EQ.3)GO TO 1002
2880 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 LEVEL & 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

```

```

5050  DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),ZZ(IF,JF,3)
5060  DO 10 I=1,IF
5070    DO 10 J=1,JF
5080      ZZ(I,J,1)=Z(I,J,2)
5090      ZZ(I,J,2)=M(I,J,2)
5100      ZZ(I,J,3)=N(I,J,2)
5110 10 CONTINUE
5120    DO 20 L=1,3
5130      DO 20 I=1,IF
5140        WRITE(NN)(ZZ(I,J,L),J=1,JF)
5150 20 CONTINUE
5160  RETURN
5170  END
5180C
5190C ***** OUTPUT OF COMPUTED RESULTS ON THE DISPLAY *****
5200C
5210  SUBROUTINE OUT6(IF,JF,Z,H,K)
5220  DIMENSION KM(47),LW(120),Z(IF,JF,2)
5230  DIMENSION H(IF,JF)
5240  DATA KM/1H,1HW,1HV,1HU,1HT,1HS,1HR,1HP,1HO,1HN,1HM,1HL,1HK,1HJ,
5250  & 1HI,1HH,1HG,1HF,1HE,1HD,1HC,1HB,1HA,1H*,1H1,1H2,1H3,1H4,1H5,1H6,
5260  & 1H7,1H8,1H9,1H0,1H.,1H-,1H+,1H/,1H#,1H$,1H%,1H&,1H=,1H?,1H(,1H),
5270  & 1H./
5280  WRITE(6,110)K
5290 110 FORMAT(//1H,2X,"K=",I5)
5300  LLL=JF/120+1
5310  IF(MOD(JF,120).EQ.0)LLL=LLL-1
5320  DO 100 L=1,LLL
5330    J1=120*(L-1)+1
5340    J2=J1+119
5350    IF(J2.GT.JF)J2=JF
5360    JX=J1+9
5370    WRITE(6,200)(J,J=JX,J2,10)
5380 200 FORMAT(1H,4X,12I10)
5390    DO 90 I=1,IF
5400      DO 80 J=J1,J2
5410        KKM=24
5420        IF(Z(I,J,2).GT.0.0)KKM=IFIX((Z(I,J,2)+0.50)/1.0)+24
5430        IF(Z(I,J,2).LT.0.0)KKM=IFIX((Z(I,J,2)-0.50)/1.0)+24
5440        IF(KKM.LT.2)KKM=2
5450        IF(KKM.GT.46)KKM=46
5460        IF(H(I,J).LT.0.0)KKM=1
5470        LW(J)=KM(KKM)
5480 80 CONTINUE
5490    WRITE(6,101)I,(LW(LL),LL=J1,J2)
5500 101 FORMAT(1H,I3,1X,120A1)
5510 90 CONTINUE
5520 100 CONTINUE
5530  RETURN
5540  END
5550C
5560C ***** CHECK OF MAXIMUM VALUE *****
5570C
5580  SUBROUTINE MAX(IF,JF,Z,ZM)
5590  DIMENSION Z(IF,JF,2),ZM(IF,JF)
5600  DO 10 I=2,IF
5610    DO 10 J=2,JF
5620      IF(ZM(I,J).LT.Z(I,J,2))ZM(I,J)=Z(I,J,2)
5630 10 CONTINUE
5640  RETURN
5650  END
5660C
5670  SUBROUTINE BNC(IF,JF,Z,B,ISS,JSS,JES,KK,DT)
5680  DIMENSION Z(IF,JF,2),B(1800,14),BC(14)
5690  KT=(KK-1)*DT
5700  K1=KT/3+1
5710  IF(K1+1.GE.1800)K1=1799
5720  K2=MOD(KT,3)
5730  DO 10 J=1,14
5740 10 BC(J)=(K2*B(K1+1,J)+(3-K2)*B(K1,J))/3.0
5750  DO 20 J=JSS+2,JES-2
5760    JX=J-(JSS+1)

```

```

5770    JJ=JX/4+1
5780    JY=2*MOD(JX,4)+1
5790    Z(ISS,I,2)=(JY*BC(JJ+1)+(8-JY)*BC(JJ))/8.0
5800  20 CONTINUE
5810    Z(ISS,JSS,2)=BC(1)
5820    Z(ISS,JSS+1,2)=BC(1)
5830    Z(ISS,JES-1,2)=BC(14)
5840    Z(ISS,JES,2)=BC(14)
5850    RETURN
5860    END
5870C
5880C ***** OUTPUT OF WATER LEVEL TIMEHISTORIES *****
5890C
5900    SUBROUTINE POINT(NN,IF,JF,IM,Z,IP,JP,PZ)
5910    DIMENSION Z(IF,JF,2),IP(IM),JP(IM),PZ(IM)
5920    DO 10 I=1,IM
5930        II=IP(I)
5940        JJ=JP(I)
5950        PZ(I)=Z(II,JJ,2)
5960  10 CONTINUE
5970    WRITE(NN)(PZ(I),I=1,IM)
5980    RETURN
5990    END
6000C
6010C ***** OUTPUT OF LAST STEP DATA *****
6020C
6030    SUBROUTINE OUT4(NN,IF,JF,Z,M,N,H,ZM)
6040    REAL M,N
6050    DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
6060    DIMENSION ZM(IF,JF,3)
6070    DO 10 I=1,IF
6080        WRITE(NN)(Z(I,J,2),J=1,JF)
6090        WRITE(NN)(M(I,J,2),J=1,JF)
6100        WRITE(NN)(N(I,J,2),J=1,JF)
6110        WRITE(NN)(H(I,J),J=1,JF)
6120        DO 20 K=1,3
6130  20 WRITE(NN)(ZM(I,J,K),J=1,JF)
6140  10 CONTINUE
6150    RETURN
6160    END
6170C
6180C ***** INPUT OF LAST STEP DATA *****
6190C
6200    SUBROUTINE INDATA(NN,IF,JF,Z,M,N,H,ZM)
6210    REAL M,N
6220    DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2),H(IF,JF)
6230    DIMENSION ZM(IF,JF,3)
6240    DO 10 I=1,IF
6250        READ(NN)(Z(I,J,1),J=1,JF)
6260        READ(NN)(M(I,J,1),J=1,JF)
6270        READ(NN)(N(I,J,1),J=1,JF)
6280        READ(NN)(H(I,J),J=1,JF)
6290        DO 20 K=1,3
6300  20 READ(NN)(ZM(I,J,K),J=1,JF)
6310  10 CONTINUE
6320    RETURN
6330    END

```



```

0010*#RUN : IAP OPT=2 F=/SNRK/JP-MYK(10)
0020C
0030C ***** TUNAMI-N3 *****
0040C   Tohoku University's Numerical-Analysis Model
0050C     for Investigation of tsunami
0060C       Near-field Tsunami version
0070C
0080C   1896 SANRIKU TSUNAMIS NEAR MIYAKO BAY
0090C     --- LINEAR LONG WAVE THEORY ---
0100C       NUMERICAL EXPERIMENT
0110C         91.7.25
0120C         BY
0130C
0140C       F.IMAMURA, TOHOKU UNIV.
0150C
0160C
0170   REAL MA,NA,MB,NB,MC,NC,MD,ND
0180   DOUBLE PRECISION T1,T2,T3
0190   PARAMETER(IA=90,JA=130,IB=37,JB=85)
0200   PARAMETER(IC=43,JC=163,ID=61,JD=73)
0210   PARAMETER(DA=5400,DB=1800,DC=600,DD=200)
0220   PARAMETER(TA=3.0,TB=3.0,TC=3.0,TD=3.0)
0230   PARAMETER(KS=1,KE=1000,KD=200,KC=3)
0240   COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
0250   COMMON /HA/ HA(IA,JA)
0260   COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
0270   COMMON /HB/ HB(IB,JB)
0280   COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
0290   COMMON /HC/ HC(IC,JC)
0300   COMMON /XD/ ZD(ID,JD,2),MD(ID,JD,2),ND(ID,JD,2)
0310   COMMON /HD/ HD(ID,JD)
0320C
0330C ***** INITIAL CONDITION *****
0340C
0350   CALL CDEPTH
0360   WRITE(6,*)"DATA OK"
0370   CALL CINTL
0380   WRITE(6,*)"INITIAL OK"
0390   CALL OUTPUT(IA,JA,ZA,HA,1,1,1)
0400C   CALL COUT (0,KD)
0410   CALL CLOCK(T1)
0420C
0430C ***** MAIN CALCULATION *****
0440C
0450   DO 10 K=KS,KE
0460     KK=K
0470     CALL CMASS
0480     CALL CJNZ
0490     CALL CMMNT
0500     CALL CJNQ
0510     CALL COUT(KK,KD)
0520     CALL CCHNG
0530   10 CONTINUE
0540   CALL CLOCK(T2)
0550   T3=T2-T1
0560   WRITE(6,*)KK,T3
0570   STOP
0580   END
0590C
0600   SUBROUTINE CDEPTH
0610C
0620   PARAMETER(IA=90,JA=130,IB=37,JB=85)
0630   PARAMETER(IC=43,JC=163,ID=61,JD=73)
0640   COMMON /HA/ HA(IA,JA)
0650   COMMON /HB/ HB(IB,JB)
0660   COMMON /HC/ HC(IC,JC)
0670   COMMON /HD/ HD(ID,JD)
0680   DIMENSION LAB(4),LBC(4),LCD(4)
0690   DATA LAB/27,44,38,71/
0700   DATA LBC/12,11,25,64/
0710   DATA LCD/16,79,35,102/
0720C

```

```

0730  NN=10
0740  DO 10 I=1,IA
0750  10 READ(NN,100)(HA(I,J),J=1,JA)
0760  DO 20 I=1,IB
0770  20 READ(NN,100)(HB(I,J),J=1,JB)
0780  DO 30 I=1,IC
0790  30 READ(NN,100)(HC(I,J),J=1,JC)
0800  DO 40 I=1,ID
0810  40 READ(NN,100)(HD(I,J),J=1,JD)
0820  100 FORMAT(10F8.1)
0830C
0840C ***** CAHNGE OF DEPTH DATA *****
0850C
0860  CALL CHH(IA,JA,IB,JB,HA,HB,LAB,1110)
0870  CALL CHH(IB,JB,IC,JC,HB,HC,LBC,1110)
0880  CALL CHH(IC,JC,ID,JD,HC,HD,LCD,1110)
0890  RETURN
0900  END
0910C
0920  SUBROUTINE CINTL
0930C
0940  REAL MA,NA,MB,NB,MC,NC,MD,ND
0950  PARAMETER(IA=90,JA=130,IB=37,JB=85)
0960  PARAMETER(IC=43,JC=163,ID=61,JD=73)
0970  PARAMETER(DA=5400,DB=1800,DC=600,DD=200)
0980  PARAMETER(TA=3.0,TB=3.0,TC=3.0,TD=3.0)
0990  COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
1000  COMMON /HA/ HA(IA,JA)
1010  COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
1020  COMMON /HB/ HB(IB,JB)
1030  COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
1040  COMMON /HC/ HC(IC,JC)
1050  COMMON /XD/ ZD(ID,JD,2),MD(ID,JD,2),ND(ID,JD,2)
1060  COMMON /HD/ HD(ID,JD)
1070C
1080  DO 100 L=1,2
1090  DO 10 J=1,JA
1100  DO 10 I=1,IA
1110  ZA(I,J,L)=0.0 ; MA(I,J,L)=0.0 ; NA(I,J,L)=0.0
1120  10 CONTINUE
1130  DO 20 J=1,JB
1140  DO 20 I=1,IB
1150  ZB(I,J,L)=0.0 ; MB(I,J,L)=0.0 ; NB(I,J,L)=0.0
1160  20 CONTINUE
1170  DO 30 J=1,JC
1180  DO 30 I=1,IC
1190  ZC(I,J,L)=0.0 ; MC(I,J,L)=0.0 ; NC(I,J,L)=0.0
1200  30 CONTINUE
1210  DO 40 J=1,JD
1220  DO 40 I=1,ID
1230  ZD(I,J,L)=0.0 ; MD(I,J,L)=0.0 ; ND(I,J,L)=0.0
1240  40 CONTINUE
1250  100 CONTINUE
1260C
1270  IS=20 ; JS=20 ; IE=90 ; JE=120
1280  CALL DEFORM(IA,JA,IS,JS,IE,JE,ZA,DA)
1290C
1300  RETURN
1310  END
1320C
1330  SUBROUTINE CMASS
1340C
1350  REAL MA,NA,MB,NB,MC,NC,MD,ND
1360  PARAMETER(IA=90,JA=130,IB=37,JB=85)
1370  PARAMETER(IC=43,JC=163,ID=61,JD=73)
1380  PARAMETER(DA=5400,DB=1800,DC=600,DD=200)
1390  PARAMETER(TA=3.0,TB=3.0,TC=3.0,TD=3.0)
1400  COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
1410  COMMON /HA/ HA(IA,JA)
1420  COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
1430  COMMON /HB/ HB(IB,JB)
1440  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

```

```

2890     ENDIF
2900 10 CONTINUE
2910   DO 20 J=2,JF-1
2920     DO 20 I=2,IF-1
2930       HH=0.5*(H(I,J)+H(I,J+1))*GG*DT/DX
2940       IF(HH.GT.0.0)THEN
2950         N(I,J,2)=N(I,J,1)-HH*(Z(I,J+1,2)-Z(I,J,2))
2960       ELSE
2970         N(I,J,2)=0.0
2980       ENDIF
2990 20 CONTINUE
3000C
3010   DO 16 I=1,IF-1
3020     HH=0.5*(H(I,JF)+H(I+1,JF))*GG*DT/DX
3030     IF(HH.GT.0.0)THEN
3040       M(I,JF,2)=M(I,JF,1)-HH*(Z(I+1,JF,2)-Z(I,JF,2))
3050     ELSE
3060       M(I,JF,2)=0.0
3070     ENDIF
3080 16 CONTINUE
3090   DO 26 J=1,JF-1
3100     HH=0.5*(H(IF,J)+H(IF,J+1))*GG*DT/DX
3110     IF(HH.GT.0.0)THEN
3120       N(IF,J,2)=N(IF,J,1)-HH*(Z(IF,J+1,2)-Z(IF,J,2))
3130     ELSE
3140       N(IF,J,2)=0.0
3150     ENDIF
3160 26 CONTINUE
3170   RETURN
3180   END
3190C
3200   SUBROUTINE BOUT
3210C
3220   REAL MA,NA
3230   PARAMETER(IA=90,JA=130)
3240   COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
3250   COMMON /HA/ HA(IA,JA)
3260   DO 10 KK=1,2
3270     J=2
3280     IF(KK.EQ.2)J=JA
3290     DO 10 I=3,IA
3300       IF(HA(I,J).LT.0.0)GOTO 10
3310       CC=SQRT(9.8*HA(I,J))
3320       UU=0.5*ABS(MA(I,J,2)+MA(I-1,J,2))
3330       IF(J.EQ.2)UU=SQRT(UU**2+NA(I,J,2)**2)
3340       IF(J.EQ.JA)UU=SQRT(UU**2+NA(I,J-1,2)**2)
3350       ZZ=UU/CC
3360       IF(J.EQ.2.AND.NA(I,J,2).GT.0.0)ZZ=-ZZ
3370       IF(J.EQ.JA.AND.NA(I,J-1,2).LT.0.0)ZZ=-ZZ
3380       ZA(I,J,2)=ZZ
3390 10 CONTINUE
3400   DO 20 KK=1,2
3410     I=2
3420     IF(KK.EQ.2)I=IA
3430     DO 20 J=3,JA
3440       IF(HA(I,J).LT.0.0)GOTO 20
3450       CC=SQRT(9.8*HA(I,J))
3460       UU=0.5*ABS(NA(I,J,2)+NA(I,J-1,2))
3470       IF(I.EQ.2)UU=SQRT(UU**2+MA(I,J,2)**2)
3480       IF(I.EQ.IA)UU=SQRT(UU**2+MA(I-1,J,2)**2)
3490       ZZ=UU/CC
3500       IF(I.EQ.2.AND.MA(I,J,2).GT.0.0)ZZ=-ZZ
3510       IF(I.EQ.IA.AND.MA(I-1,J,2).LT.0.0)ZZ=-ZZ
3520       ZA(I,J,2)=ZZ
3530 20 CONTINUE
3540   RETURN
3550   END
3560C
3570   SUBROUTINE CHH(IX,JX,IY,JY,HX,HY,L0,BCHK)
3580C
3590   INTEGER BCHK,CHK
3600   DIMENSION HX(IX,JX),HY(IY,JY),L0(4)

```

```

3610C
3620 ISS=2 ; JSS=2 ; IES=IY ; JES=JY ; ISL=L0(1) ; JSL=L0(2)
3630 IEL=L0(3) ; JEL=L0(4)
3640 CHK=BCHK
3650 KB=CHK/1000
3660 IF(KB.EQ.1)THEN
3670   CHK=CHK-1000
3680   II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
3690   DO WHILE(I1+2.LE.IES)
3700     S=0.0 ; L=0
3710     DO 10 I=I1,I1+2
3720       DO 10 J=J1,J1+2
3730         IF(HY(I,J).GT.0.0)THEN
3740           S=S+HY(I,J)
3750           L=L+1
3760         ENDIF
3770 10  CONTINUE
3780       IF(L.GE.5)THEN
3790         HX(II,JJ)=S/L
3800       ENDIF
3810       II=II+1 ; I1=I1+3
3820     ENDDO
3830   ENDIF
3840C
3850   KB=CHK/100
3860   IF(KB.EQ.1)THEN
3870     CHK=CHK-100
3880     II=IEL ; JJ=JSL ; I1=IES-2 ; J1=JSS
3890     DO WHILE(J1+1.LE.JES)
3900       S=0.0 ; L=0
3910       DO 20 I=I1,I1+2
3920         DO 20 J=J1,J1+2
3930           IF(HY(I,J).GT.0.0)THEN
3940             S=S+HY(I,J)
3950             L=L+1
3960           ENDIF
3970 20  CONTINUE
3980         IF(L.GE.5)THEN
3990           HX(II,JJ)=S/L
4000         ENDIF
4010         JJ=JJ+1 ; J1=J1+3
4020       ENDDO
4030     ENDIF
4040C
4050     KB=CHK/10
4060     IF(KB.EQ.1)THEN
4070       CHK=CHK-10
4080       II=ISL ; JJ=JEL ; I1=ISS ; J1=JES-2
4090       DO WHILE(I1+2.LE.IES)
4100         S=0.0 ; L=0
4110         DO 30 I=I1,I1+2
4120           DO 30 J=J1,J1+2
4130             IF(HY(I,J).GT.0.0)THEN
4140               S=S+HY(I,J)
4150               L=L+1
4160             ENDIF
4170 30  CONTINUE
4180           IF(L.GE.5)THEN
4190             HX(II,JJ)=S/L
4200           ENDIF
4210           II=II+1 ; I1=I1+3
4220         ENDDO
4230       ENDIF
4240C
4250       IF(CHK.EQ.1)THEN
4260         II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
4270         DO WHILE(J1+2.LE.JES)
4280           S=0.0 ; L=0
4290           DO 40 I=I1,I1+2
4300             DO 40 J=J1,J1+2
4310               IF(HY(I,J).GT.0.0)THEN
4320                 S=S+HY(I,J)

```

```

4330      L=L+1
4340      ENDIF
4350 40  CONTINUE
4360      IF(L.GE.5)THEN
4370      HX(II,JJ)=S/L
4380      ENDIF
4390      JJ=JJ+1 ; J1=J1+3
4400      ENDDO
4410      ENDIF
4420      RETURN
4430      END
4440C
4450      SUBROUTINE JNZ(IX,JX,IY,JY,ZX,ZY,HY,L0,BCHK,LP,LT)
4460C
4470      INTEGER BCHK,CHK
4480      DIMENSION ZX(IX,JX,2),ZY(IY,JY,2)
4490      DIMENSION HY(IY,JY),L0(4)
4500      IF(MOD(LP,LT).NE.0)RETURN
4510      ISS=2 ; JSS=2 ; IES=IY ; JES=JY ; ISL=L0(1) ; JSL=L0(2)
4520      IEL=L0(3) ; JEL=L0(4)
4530      CHK=BCHK
4540      KB=CHK/1000
4550      IF(KB.EQ.1)THEN
4560      CHK=CHK-1000
4570      II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
4580      DO WHILE(I1+2.LE.IES)
4590      S=0.0 ; L=0
4600      DO 10 I=I1,I1+2
4610      DO 10 J=J1,J1+2
4620      IF(HY(I,J).GT.0.0)THEN
4630      S=S+ZY(I,J,2)
4640      L=L+1
4650      ENDIF
4660 10  CONTINUE
4670      IF(L.GE.5)THEN
4680      ZX(II,JJ,2)=S/L
4690      ELSE
4700      ZX(II,JJ,2)=0.0
4710      ENDIF
4720      II=II+1 ; I1=I1+3
4730      ENDDO
4740      ENDIF
4750C
4760      KB=CHK/100
4770      IF(KB.EQ.1)THEN
4780      CHK=CHK-100
4790      II=IEL ; JJ=JSL ; I1=IES-2 ; J1=JSS
4800      DO WHILE(J1+1.LE.JES)
4810      S=0.0 ; L=0
4820      DO 20 I=I1,I1+2
4830      DO 20 J=J1,J1+2
4840      IF(HY(I,J).GT.0.0)THEN
4850      S=S+ZY(I,J,2)
4860      L=L+1
4870      ENDIF
4880 20  CONTINUE
4890      IF(L.GE.5)THEN
4900      ZX(II,JJ,2)=S/L
4910      ELSE
4920      ZX(II,JJ,2)=0.0
4930      ENDIF
4940      JJ=JJ+1 ; J1=J1+3
4950      ENDDO
4960      ENDIF
4970C
4980      KB=CHK/10
4990      IF(KB.EQ.1)THEN
5000      CHK=CHK-10
5010      II=ISL ; JJ=JEL ; I1=ISS ; J1=JES-2
5020      DO WHILE(I1+2.LE.IES)
5030      S=0.0 ; L=0
5040      DO 30 I=I1,I1+2

```

```

5050      DO 30 J=J1,J1+2
5060      IF(HY(I,J).GT.0.0)THEN
5070      S=S+ZY(I,J,2)
5080      L=L+1
5090      ENDIF
5100 30   CONTINUE
5110      IF(L.GE.5)THEN
5120      ZX(II,JJ,2)=S/L
5130      ELSE
5140      ZX(II,JJ,2)=0.0
5150      ENDIF
5160      II=II+1 ; I1=I1+3
5170      ENDDO
5180      ENDIF
5190C
5200      IF(CHK.EQ.1)THEN
5210      II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
5220      DO WHILE(J1+2.LE.JES)
5230      S=0.0 ; L=0
5240      DO 40 I=I1,I1+2
5250      DO 40 J=J1,J1+2
5260      IF(HY(I,J).GT.0.0)THEN
5270      S=S+ZY(I,J,2)
5280      L=L+1
5290      ENDIF
5300 40   CONTINUE
5310      IF(L.GE.5)THEN
5320      ZX(II,JJ,2)=S/L
5330      ELSE
5340      ZX(II,JJ,2)=0.0
5350      ENDIF
5360      JJ=JJ+1 ; J1=J1+3
5370      ENDDO
5380      ENDIF
5390      RETURN
5400      END
5410C
5420      SUBROUTINE JNQ(IX,JX,IY,JY,MX,NX,MY,NY,HY,L0,BCHK,LP,LT)
5430C
5440      INTEGER BCHK,CHK
5450      REAL MX,NX,MY,NY
5460      DIMENSION MX(IX,JX,2),NX(IX,JX,2),HY(IY,JY)
5470      DIMENSION MY(IY,JY,2),NY(IY,JY,2),L0(4)
5480      IF(MOD(LP,LT).NE.0)RETURN
5490      ISS=2 ; JSS=2 ; IES=IY ; JES=JY ; ISL=L0(1) ; JSL=L0(2)
5500      IEL=L0(3) ; JEL=L0(4)
5510      CHK=BCHK
5520      KB=CHK/1000
5530      IF(KB.EQ.1)THEN
5540      CHK=CHK-1000
5550      I=ISS ; J=JSS-1 ; II=ISL-1 ; JJ=JSL-1
5560      DO WHILE(I.LE.IES)
5570      SI=(I-ISS+2)/3.0
5580      IS=IFIX(SI) ; DI=SI-IS ; II=IS+ISL-1
5590      NY(I,J,2)=(1-DI)*NX(II,JJ,2)+DI*NX(II+1,JJ,2)
5600      IF(HY(I,J+1).LT.0.0)NY(I,J,2)=0.0
5610      I=I+1
5620      ENDDO
5630      ENDIF
5640C
5650      KB=CHK/100
5660      IF(KB.EQ.1)THEN
5670      CHK=CHK-100
5680      I=IES ; J=JSS ; II=IEL ; JJ=JSL-1
5690      DO WHILE(J.LE.JES)
5700      SJ=(J-JSS+2)/3.0
5710      JS=IFIX(SJ) ; DJ=SJ-JS ; JJ=JS+JSL-1
5720      MY(I,J,2)=(1-DJ)*MX(II,JJ,2)+DJ*MX(II,JJ+1,2)
5730      IF(HY(I,J).LT.0.0)MY(I,J,2)=0.0
5740      J=J+1
5750      ENDDO
5760      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,J,2)+DI*NX(II+1,J,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,J,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

```

```

7210 IF(MOD(KK,KC).NE.0)RETURN
7220 DO 10 K=1,NG
7230 PZ(K)=Z(IP(K),JP(K),2)
7240 10 CONTINUE
7250 WRITE(24,100)(PZ(K),K=1,NG)
7260 100 FORMAT(10F7.3)
7270 RETURN
7280 END
7290C
7300 SUBROUTINE PROPA(IF,JF,IS,JS,IE,JE,Z,H,LF,KK)
7310C
7320 DIMENSION Z(IF,JF,2),H(IF,JF),LF(IF,JF)
7330C
7340 LK=KK/180+1
7350 DO 10 I=IS,IE
7360 DO 10 J=JS,JE
7370 IF(H(I,J).GT.0.0)THEN
7380 IF(LF(I,J).EQ.0)THEN
7390 IF(ABS(Z(I,J,2)).GT.0.01)LF(I,J)=LK
7400 ENDIF
7410 ELSE
7420 LF(I,J)=-9
7430 ENDIF
7440 10 CONTINUE
7450 RETURN
7460 END
7470C
7480 SUBROUTINE OUTDT(IF,JF,ZM,ZN,LK)
7490 DIMENSION ZM(IF,JF),ZN(IF,JF),LK(IF,JF)
7500 DO 20 I=1,IF
7510 DO 20 J=1,JF
7520 ZN(I,J)=ZM(I,J)-ZN(I,J)
7530 IF(ZN(I,J).GT.100.0)ZN(I,J)=100.0
7540 IF(ZM(I,J).GT.100.0)ZM(I,J)=100.0
7550 20 CONTINUE
7560 DO 10 J=1,JF
7570 WRITE(31,100)(ZM(I,J),I=1,IF)
7580 WRITE(32,100)(ZN(I,J),I=1,IF)
7590 WRITE(33,200)(LK(I,J),I=1,IF)
7600 10 CONTINUE
7610 100 FORMAT(20F6.3)
7620 200 FORMAT(40I3)
7630 RETURN
7640 END
7650C
7660 SUBROUTINE DEFORM(IO,JO,IS,JS,IE,JE,Z,DX)
7670 REAL L
7680 PARAMETER(A=3.141592,B=4.848E-06)
7690 PARAMETER(RR=6.37E+6,E=1.7453E-2)
7700 PARAMETER(HH=1.0E+3,D=12.5,DL=20.0,TH=156.0,RD=58.0)
7710 PARAMETER(L=210.0E+3,W=50.0E+3)
7720 PARAMETER(YO=37.0,XO=139.67)
7730 PARAMETER(YO=40.9,XO=143.3)
7740 DIMENSION Z(IO,JO,2)
7750 XL=A*RR*(XO-XO)*COS(E*YO)/180.0 ; YL=A*RR*(YO-YO)/180.0
7760 H1=HH/SIN(E*DL) ; H2=HH/SIN(E*DL)+W ; DS=D*COS(E*RD)
7770 DD=D*SIN(E*RD)
7780 WRITE(6,*)XL,YL,H1,H2,DS,DD
7790 DO 10 I=IS,IE
7800 DO 10 J=JS,JE
7810 XX=DX*(I-1) ; YY=DX*(J-1)
7820C YY=A*RR*DR*(J-1)/(60.0*180)
7830C XX=A*RR*DR*(I-1)*COS(E*(YO+DR*(J-1)/60.0))/(60.0*180)
7840 X1=(XX-XL)*SIN(E*TH)+(YY-YL)*COS(E*TH)
7850 X2=(XX-XL)*COS(E*TH)-(YY-YL)*SIN(E*TH)+HH/TAN(E*DL)
7860 X3=0.0
7870 CALL USCAL(X1,X2,X3,L,H2,E*DL,F1)
7880 CALL USCAL(X1,X2,X3,L,H1,E*DL,F2)
7890 CALL USCAL(X1,X2,X3,0.,H2,E*DL,F3)
7900 CALL USCAL(X1,X2,X3,0.,H1,E*DL,F4)
7910 CALL UDCAL(X1,X2,X3,L,H2,E*DL,G1)
7920 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

```

```
8650 REAL FUNCTION ATN(AX,AY)
8660 DATA GX/1.0E-6/
8670 AAX=ABS(AX) ; AAY=ABS(AY)
8680 P=AX*AY
8690 IF(AAX.LE.GX.AND.AAY.LE.GX)GOTO 10
8700 SR=ATAN2(AAX,AAY)
8710 ATN=SIGN(SR,P)
8720 RETURN
8730 10 WRITE(6,100)AX,AY
8740 100 FORMAT(1H,"ATAN -- AX=",E15.7,2X,"AY=",E15.7)
8750 ATN=0.2
8760 RETURN
8770 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

	Page
TUNAMI-F2: LINEAR THEORY FOR PROPAGATION IN THE OCEAN AND COASTAL WATERS	1-23

```

0010C
0020C ***** TUNAMI-F2 *****
0030C   Tohoku University's Numerical-Analysis Model
0040C     for Investigation of tsunami
0050C     Far-field Tsunami version
0060C
0070C     1960 CHILEAN TSUNAMI (DX=10')
0080C     OHMA VERSION 1.2
0090C     91.7.25
0100C
0110C     BY
0120C
0130C     F.IMAMURA, TOHOKU UNIV.
0140C
0150C     Z; WATER SURFACE LEVEL  M,N;WATER DISCHARGE
0160C     DX; GRID LENGTH
0170C
0180   REAL M,N,MA,NA,MB,NB,MC,NC,ME,NE,MF,NF
0190   REAL MD1,ND1,MD2,ND2,MD3,ND3,MD4,ND4,MD5,ND5
0200   DOUBLE PRECISION T1,T2,T3
0210   PARAMETER (IO=1020,JO=738,IA=76,JA=76)
0220   PARAMETER (IB=136,JB=136,IC=154,JC=307)
0230   PARAMETER (ID1=28,JD1=55,ID2=55,JD2=40)
0240   PARAMETER (ID3=37,JD3=31,ID4=13,JD4=40)
0250   PARAMETER (ID5=46,JD5=40)
0260   PARAMETER (IE1=34,JE1=67,IF1=46,JF1=58)
0270   PARAMETER (DO=10.0,DA=4.0,DB=1.33,DC=833)
0280   PARAMETER (DD=278.0,DE=93.0,DF=31.0)
0290   PARAMETER (TO=20.0,TA=10.0,TB=5.0,TC=5.0)
0300   PARAMETER (TD=5.0,TE=2.5,TF=1.25)
0310   PARAMETER (KS=1,KE=180*32,KD=180*2,KC=3)
0320   DATA IS,JS,IE,JE/961,61,1020,180/
0330   COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
0340   COMMON /HH/ H(IO,JO)
0350   COMMON /RR/ R1(IO,JO),R2(IO,JO),R3(IO,JO),
0360     &R4(IO,JO),R5(IO,JO),R6(IO)
0370   COMMON /CC/ C1(IO),C2(IO),C3(IO),C4(IO)
0380   COMMON /VV/ V1(IO),V2(IO),V3(IO),V4(IO),V5(IO)
0390   COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
0400   COMMON /HA/ HA(IA,JA)
0410   COMMON /RA/ RA1(IA,JA),RA2(IA,JA),RA3(IA,JA),
0420     &RA4(IA,JA),RA5(IA,JA),RA6(IA)
0430   COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
0440   COMMON /HB/ HB(IB,JB)
0450   COMMON /RB/ RB1(IB,JB),RB2(IB,JB),RB3(IB,JB),
0460     &RB4(IB,JB),RB5(IB,JB),RB6(IB)
0470   COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
0480   COMMON /HC/ HC(IC,JC)
0490   COMMON /XD1/ ZD1(ID1,JD1,2),MD1(ID1,JD1,2),
0500     &ND1(ID1,JD1,2),HD1(ID1,JD1)
0510   COMMON /XD2/ ZD2(ID2,JD2,2),MD2(ID2,JD2,2),
0520     &ND2(ID2,JD2,2),HD2(ID2,JD2)
0530   COMMON /XD3/ ZD3(ID3,JD3,2),MD3(ID3,JD3,2),
0540     &ND3(ID3,JD3,2),HD3(ID3,JD3)
0550   COMMON /XD4/ ZD4(ID4,JD4,2),MD4(ID4,JD4,2),
0560     &ND4(ID4,JD4,2),HD4(ID4,JD4)
0570   COMMON /XD5/ ZD5(ID5,JD5,2),MD5(ID5,JD5,2),
0580     &ND5(ID5,JD5,2),HD5(ID5,JD5)
0590   COMMON /XE/ ZE(IE1,JE1,2),ME(IE1,JE1,2),
0600     &NE(IE1,JE1,2),HE(IE1,JE1)
0610   COMMON /XF/ ZF(IF1,JF1,2),MF(IF1,JF1,2),
0620     &NF(IF1,JF1,2),HF(IF1,JF1)

```



```

0630C
0640C ***** INITIAL CONDITION *****
0650C
0660  CALL RDEPTH
0670  CALL PARAME
0680  CALL INITIA (IS,JS,IE,JE)
0690C  CALL COUT (0,KD)
0700  CALL CLOCK (T1)
0710C
0720C ***** MAIN CALCULATION *****
0730C
0740  DO 10 K=KS,KE
0750  KK=K
0760C
0770C ***** TRANSOCEANIC PROPAGATION *****
0780C
0790  CALL MASS1 (IS,JS,IE,JE)
0800  CALL BOUND (IS,JS,IE,JE)
0810  CALL LIMIT (IS,JS,IE,JE)
0820C
0830C ***** CAL. NEAR NORTH JAPAN *****
0840C
0850  IF (IS.LE.121.AND.JE.GE.592) THEN
0860  CALL CAL (IS,JS,IE,JE,KK) ; ELSE
0870  CALL MMNT1 (IS,JS,IE,JE) ; ENDIF
0880  CALL COUT (KK,KD)
0890  CALL CHANGE
0900  10 CONTINUE
0910  CALL CLOCK (T2) ; T3=T2-T1
0920  WRITE(6,*)KK,T3
0930  STOP ; END
0940C
0950C ***** CAL. NEAR NORTH JAPAN *****
0960C
0970C
0980  SUBROUTINE CAL (IS,JS,IE,JE,KK)
0990C
1000  REAL M,N,MA,NA,MB,NB,MC,NC,ME,NE,MF,NF
1010  REAL MD1,ND1,MD2,ND2,MD3,ND3,MD4,ND4,MD5,ND5
1020  PARAMETER (IO=1020,JO=738,IA=76,JA=76)
1030  PARAMETER (IB=136,JB=136,IC=154,JC=307)
1040  PARAMETER (ID1=28,JD1=55,ID2=55,JD2=40)
1050  PARAMETER (ID3=37,JD3=31,ID4=13,JD4=40)
1060  PARAMETER (ID5=46,JD5=40)
1070  PARAMETER (IE1=34,JE1=67,IF1=46,JF1=58)
1080  PARAMETER (DC=833.0,DD=278.0,DE=93.0,DF=31.0)
1090  PARAMETER (TC=5.0,TD=5.0,TE=2.5,TF=1.25)
1100  COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
1110  COMMON /HH/ H(IO,JO)
1120  COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
1130  COMMON /HA/ HA(IA,JA)
1140  COMMON /RA/ RA1(IA,JA),RA2(IA,JA),RA3(IA,JA),
1150  &RA4(IA,JA),RA5(IA,JA),RA6(IA)
1160  COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
1170  COMMON /HB/ HB(IB,JB)
1180  COMMON /RB/ RB1(IB,JB),RB2(IB,JB),RB3(IB,JB),
1190  &RB4(IB,JB),RB5(IB,JB),RB6(IB)
1200  COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
1210  COMMON /HC/ HC(IC,JC)
1220  COMMON /XD1/ ZD1(ID1,JD1,2),MD1(ID1,JD1,2),
1230  &ND1(ID1,JD1,2),HD1(ID1,JD1)
1240  COMMON /XD2/ ZD2(ID2,JD2,2),MD2(ID2,JD2,2),
1250  &ND2(ID2,JD2,2),HD2(ID2,JD2)
1260  COMMON /XD3/ ZD3(ID3,JD3,2),MD3(ID3,JD3,2),
1270  &ND3(ID3,JD3,2),HD3(ID3,JD3)
1280  COMMON /XD4/ ZD4(ID4,JD4,2),MD4(ID4,JD4,2),
1290  &ND4(ID4,JD4,2),HD4(ID4,JD4)
1300  COMMON /XD5/ ZD5(ID5,JD5,2),MD5(ID5,JD5,2),
1310  &ND5(ID5,JD5,2),HD5(ID5,JD5)
1320  COMMON /XE/ ZE(IE1,JE1,2),ME(IE1,JE1,2),
1330  &NE(IE1,JE1,2),HE(IE1,JE1)
1340  COMMON /XF/ ZF(IF1,JF1,2),MF(IF1,JF1,2),
1350  &NF(IF1,JF1,2),HF(IF1,JF1)
1360  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

```

```

2110 PD(1)=ZD1(24,39,2) ; PD(2)=ZD1(22,43,2)
2120 PD(3)=ZD1(19,42,2) ; PD(4)=ZD1(13,14,2)
2130 PD(5)=ZD1(10,7,2) ; PD(6)=ZD2(41,16,2)
2140 PD(7)=ZD2(35,19,2) ; PD(8)=ZD2(39,23,2)
2150 PD(9)=ZD2(18,39,2) ; PD(10)=ZD3(6,25,2)
2160 PD(11)=ZD3(26,7,2) ; PD(12)=ZD4(4,27,2)
2170 PD(13)=ZD5(22,4,2) ; PD(14)=ZD5(32,6,2)
2180 WRITE(30,100)(PD(LL),LL=1,14)
2190 100 FORMAT(7F8.3)
2200 ENDIF
2210C
2220 RETURN ; END
2230C
2240C ***** READ DEPTH DATA *****
2250C
2260 SUBROUTINE RDEPTH
2270C
2280 REAL MD1,ND1,MD2,ND2,MD3,ND3,MD4,ND4,MD5,ND5
2290 REAL ME,NE,MF,NF
2300 PARAMETER (IO=1020,JO=738,IA=76,JA=76)
2310 PARAMETER (IB=136,JB=136,IC=154,JC=307)
2320 PARAMETER (ID1=28,JD1=55,ID2=55,JD2=40)
2330 PARAMETER (ID3=37,JD3=31,ID4=13,JD4=40)
2340 PARAMETER (ID5=46,JD5=40)
2350 PARAMETER (IE1=34,JE1=67,IF1=46,IF1=58)
2360 COMMON /HH/ H(IO,JO)
2370 COMMON /HA/ HA(IA,JA)
2380 COMMON /HB/ HB(IB,JB)
2390 COMMON /HC/ HC(IC,JC)
2400 COMMON /XD1/ ZD1(ID1,JD1,2),MD1(ID1,JD1,2),
2410 &ND1(ID1,JD1,2),HD1(ID1,JD1)
2420 COMMON /XD2/ ZD2(ID2,JD2,2),MD2(ID2,JD2,2),
2430 &ND2(ID2,JD2,2),HD2(ID2,JD2)
2440 COMMON /XD3/ ZD3(ID3,JD3,2),MD3(ID3,JD3,2),
2450 &ND3(ID3,JD3,2),HD3(ID3,JD3)
2460 COMMON /XD4/ ZD4(ID4,JD4,2),MD4(ID4,JD4,2),
2470 &ND4(ID4,JD4,2),HD4(ID4,JD4)
2480 COMMON /XD5/ ZD5(ID5,JD5,2),MD5(ID5,JD5,2),
2490 &ND5(ID5,JD5,2),HD5(ID5,JD5)
2500 COMMON /XE/ ZE(IE1,JE1,2),ME(IE1,JE1,2),
2510 &NE(IE1,JE1,2),HE(IE1,JE1)
2520 COMMON /XF/ ZF(IF1,JF1,2),MF(IF1,JF1,2),
2530 &NF(IF1,JF1,2),HF(IF1,JF1)
2540 DIMENSION LAB(4),LBC(4),LDE(4),LEF(4)
2550 DIMENSION LD1(4),LD2(4),LD3(4),LD4(4),LD5(4)
2560 DATA LAB/5,20,49,64/
2570 DATA LBC/6,33,74,134/
2580 DATA LD1/60,144,68,161/
2590 DATA LD2/35,182,52,194/
2600 DATA LD3/90,133,101,142/
2610 DATA LD4/114,103,117,115/
2620 DATA LD5/122,23,136,35/
2630 DATA LDE/11,19,21,40/
2640 DATA LEF/14,26,28,44/
2650 DO 10 J=1,JO
2660 10 READ(20,900)(H(I,J),I=1,IO)
2670 DO 20 J=1,JA
2680 20 READ(21,900)(HA(I,J),I=1,IA)
2690 DO 30 J=1,JB
2700 30 READ(22,900)(HB(I,J),I=1,IB)
2710 DO 40 J=1,JC
2720 40 READ(23,900)(HC(I,J),I=1,IC)
2730 DO 51 J=1,JD1
2740 51 READ(24,900)(HD1(I,J),I=1,ID1)
2750 DO 52 J=1,JD2
2760 52 READ(24,900)(HD2(I,J),I=1,ID2)
2770 DO 53 J=1,JD3
2780 53 READ(24,900)(HD3(I,J),I=1,ID3)
2790 DO 54 J=1,JD4
2800 54 READ(24,900)(HD4(I,J),I=1,ID4)
2810 DO 55 J=1,JD5
2820 55 READ(24,900)(HD5(I,J),I=1,ID5)
2830 DO 60 J=1,JE1
2840 60 READ(25,900)(HE(I,J),I=1,IE1)

```

```

2850 DO 70 J=1,JF1
2860 70 READ(26,900)(HF(I,J),I=1,IF1)
2870C
2880 900 FORMAT(20F6.1)
2890 DO 101 J=1,JD1
2900 DO 101 I=1,ID1
2910 IF(HD1(I,J).LT.0.0)THEN ; HD1(I,J)=-90.0
2920 ELSE ; IF(HD1(I,J).LT.10.0)HD1(I,J)=10.0
2930 ENDIF
2940 101 CONTINUE
2950 DO 102 J=1,JD2
2960 DO 102 I=1,ID2
2970 IF(HD2(I,J).LT.0.0)THEN ; HD2(I,J)=-90.0
2980 ELSE ; IF(HD2(I,J).LT.10.0)HD2(I,J)=10.0
2990 ENDIF
3000 102 CONTINUE
3010 DO 103 J=1,JD3
3020 DO 103 I=1,ID3
3030 IF(HD3(I,J).LT.0.0)THEN ; HD3(I,J)=-90.0
3040 ELSE ; IF(HD3(I,J).LT.10.0)HD3(I,J)=10.0
3050 ENDIF
3060 103 CONTINUE
3070 DO 104 J=1,JD4
3080 DO 104 I=1,ID4
3090 IF(HD4(I,J).LT.0.0)THEN ; HD4(I,J)=-90.0
3100 ELSE ; IF(HD4(I,J).LT.10.0)HD4(I,J)=10.0
3110 ENDIF
3120 104 CONTINUE
3130 DO 105 J=1,JD5
3140 DO 105 I=1,ID5
3150 IF(HD5(I,J).LT.0.0)THEN ; HD5(I,J)=-90.0
3160 ELSE ; IF(HD5(I,J).LT.10.0)HD5(I,J)=10.0
3170 ENDIF
3180 105 CONTINUE
3190C
3200 DO 110 J=1,JE1
3210 DO 110 I=1,IE1
3220 IF(HE(I,J).LT.0.0)THEN ; HE(I,J)=-90.0
3230 ELSE ; IF(HE(I,J).LT.10.0)HE(I,J)=10.0
3240 ENDIF
3250 110 CONTINUE
3260 DO 120 J=1,JF1
3270 DO 120 I=1,IF1
3280 IF(HF(I,J).LT.0.0)THEN ; HF(I,J)=-90.0
3290 ELSE ; IF(HF(I,J).LT.10.0)HF(I,J)=10.0
3300 ENDIF
3310 120 CONTINUE
3320C
3330C ***** CAHNGE OF DEPTH DATA *****
3340C
3350 CALL CHH (IA,JA,IB,JB,HA,HB,LAB,1101)
3360 CALL CHH (IB,JB,IC,JC,HB,HC,LBC,0101)
3370 CALL CHH (IC,JC,ID1,JD1,HC,HD1,LD1,1111)
3380 CALL CHH (IC,JC,ID2,JD2,HC,HD2,LD2,1101)
3390 CALL CHH (IC,JC,ID3,JD3,HC,HD3,LD3,0110)
3400 CALL CHH (IC,JC,ID4,JD4,HC,HD4,LD4,1110)
3410 CALL CHH (IC,JC,ID5,JD5,HC,HD5,LD5,0110)
3420 CALL CHH (ID1,JD1,IE1,JE1,HD1,HE,LDE,1011)
3430 CALL CHH (IE1,JE1,IF1,JF1,HE,HF,LEF,1011)
3440 RETURN
3450 END
3460C
3470C ***** PARAMETERS FOR VECTOR OPERATION *****
3480C
3490 SUBROUTINE PARAME
3500 PARAMETER (GG=9.8)
3510 PARAMETER (FO=-60.,FA=38.5,FB=39.7,PP=3.1415926)
3520 PARAMETER (R=6.37E+6,WW=7.2722E-5)
3530 PARAMETER (IO=1020,JO=738,IA=76,JA=76)
3540 PARAMETER (IB=136,JB=136,IC=154,JC=307)
3550 PARAMETER (DO=10.0,DA=4.0,DB=1.33)
3560 PARAMETER (TO=20.0,TA=10.0,TB=5.0)
3570 COMMON /HH/ H(IO,JO)
3580 COMMON /RR/ R1(IO,JO),R2(IO,JO),R3(IO,JO),

```

```

3590      &R4(IO,JO),R5(IO,JO),R6(IO)
3600 COMMON /CC/ C1(IO),C2(IO),C3(IO),C4(IO)
3610 COMMON /VV/ V1(IO),V2(IO),V3(IO),V4(IO),V5(IO)
3620 COMMON /HA/ HA(IA,JA)
3630 COMMON /RA/ RA1(IA,JA),RA2(IA,JA),RA3(IA,JA),
3640      &RA4(IA,JA),RA5(IA,JA),RA6(IA)
3650 COMMON /HB/ HB(IB,JB)
3660 COMMON /RB/ RB1(IB,JB),RB2(IB,JB),RB3(IB,JB),
3670      &RB4(IB,JB),RB5(IB,JB),RB6(IB)
3680C
3690      PX=PP/180.
3700      DX=R*(DO/60.)*PX
3710      RR=TO/DX
3720      RS=0.5*GG*RR
3730      RT=0.5*TO*WW
3740      DO 10 J=1,JO
3750      C1(J)=(FO+(DO*J-0.5*DO)/60.)*PX
3760      C2(J)=(FO+DO*J/60.)*PX
3770 10 CONTINUE
3780      DO 11 J=1,JO
3790      C3(J)=COS(C1(J))
3800      V1(J)=RR/C3(J)
3810      V2(J)=RS/C3(J)
3820      V3(J)=RT*SIN(C1(J))
3830      V4(J)=RS
3840      V5(J)=RT*SIN(C2(J))
3850      R6(J)=C3(J)
3860 11 CONTINUE
3870      DO 12 J=1,JO
3880      C4(J)=0.
3890      DO 12 I=1,IO
3900      IF(H(I,J).GT.C4(J))C4(J)=H(I,J)
3910 12 CONTINUE
3920      DO 15 J=1,JO
3930      DO 15 I=1,IO
3940      IF(H(I,J).LT.0.)GOTO 15
3950      R1(I,J)=V1(J)
3960      IF(I.NE.IO)THEN
3970      IF(H(I+1,J).LT.0.)GOTO 16
3980      R2(I,J)=V2(J)*(H(I,J)+H(I+1,J))
3990      R3(I,J)=V3(J)
4000      ELSE
4010      R2(I,J)=V2(J)*H(I,J)*2
4020      R3(I,J)=V3(J)
4030      ENDIF
4040 16 IF(J.NE.JO)THEN
4050      IF(H(I,J+1).LT.0.)GOTO 15
4060      R4(I,J)=V4(J)*(H(I,J)+H(I,J+1))
4070      R5(I,J)=V5(J)
4080      ELSE
4090      R4(I,J)=V4(J)*H(I,J)*2
4100      R5(I,J)=V5(J)
4110      ENDIF
4120 15 CONTINUE
4130      DO 17 I=1,IO
4140 17 C1(I)=0.
4150      DO 18 I=1,IO
4160 18 IF(H(I,1).GT.0.)C1(I)=1./SQRT(GG*H(I,1))
4170      DO 19 J=1,JO
4180 19 C2(J)=0.
4190      DO 20 J=1,JO
4200 20 IF(H(1,J).GT.0.)C2(J)=1./SQRT(GG*H(1,J))
4210      DO 21 I=1,IO
4220 21 C3(I)=0.
4230      DO 22 I=1,IO
4240 22 IF(H(I,JO).GT.0.)C3(I)=1./SQRT(GG*H(I,JO))
4250      DO 23 J=1,JO
4260 23 C4(J)=0.
4270      DO 24 J=1,JO
4280 24 IF(H(IO,J).GT.0.)C4(J)=1./SQRT(GG*H(IO,J))
4290C
4300      DX=R*(DA/60.)*PX
4310      RR=TA/DX
4320      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

```

```

5810  DIMENSION HX(IX,JX),HY(IY,JY),L0(4)
5820C
5830  ISS=2 ; JSS=2 ; IES=IY ; JES=JY
5840  ISL=L0(1) ; JSL=L0(2) ; IEL=L0(3) ; JEL=L0(4)
5850  CHK=BCHK
5860  KB=CHK/1000
5870  IF(KB.EQ.1)THEN
5880  CHK=CHK-1000
5890  II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
5900  DO WHILE (I1+2.LE.IES)
5910  S=0.0 ; L=0
5920  DO 10 I=I1,I1+2
5930  DO 10 J=J1,J1+2
5940  IF(HY(I,J).GT.0.0)THEN
5950  S=S+HY(I,J)
5960  L=L+1 ; ENDIF
5970 10 CONTINUE
5980  IF(L.GE.5)THEN
5990  HX(II,JJ)=S/L ; ENDIF
6000  II=II+1 ; I1=I1+3
6010  ENDDO ; ENDIF
6020C
6030  KB=CHK/100
6040  IF(KB.EQ.1)THEN
6050  CHK=CHK-100
6060  II=IEL ; JJ=JSL ; I1=IES-2 ; J1=JSS
6070  DO WHILE (J1+1.LE.JES)
6080  S=0.0 ; L=0
6090  DO 20 I=I1,I1+2
6100  DO 20 J=J1,J1+2
6110  IF(HY(I,J).GT.0.0)THEN
6120  S=S+HY(I,J)
6130  L=L+1 ; ENDIF
6140 20 CONTINUE
6150  IF(L.GE.5)THEN
6160  HX(II,JJ)=S/L ; ENDIF
6170  JJ=JJ+1 ; J1=J1+3
6180  ENDDO ; ENDIF
6190C
6200  KB=CHK/10
6210  IF(KB.EQ.1)THEN
6220  CHK=CHK-10
6230  II=ISL ; JJ=JEL ; I1=ISS ; J1=JES-2
6240  DO WHILE (I1+2.LE.IES)
6250  S=0.0 ; L=0
6260  DO 30 I=I1,I1+2
6270  DO 30 J=J1,J1+2
6280  IF(HY(I,J).GT.0.0)THEN
6290  S=S+HY(I,J)
6300  L=L+1 ; ENDIF
6310 30 CONTINUE
6320  IF(L.GE.5)THEN
6330  HX(II,JJ)=S/L ; ENDIF
6340  II=II+1 ; I1=I1+3
6350  ENDDO ; ENDIF
6360C
6370  IF(CHK.EQ.1)THEN
6380  II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
6390  DO WHILE (J1+2.LE.JES)
6400  S=0.0 ; L=0
6410  DO 40 I=I1,I1+2
6420  DO 40 J=J1,J1+2
6430  IF(HY(I,J).GT.0.0)THEN
6440  S=S+HY(I,J)
6450  L=L+1 ; ENDIF
6460 40 CONTINUE
6470  IF(L.GE.5)THEN
6480  HX(II,JJ)=S/L ; ENDIF
6490  JJ=JJ+1 ; J1=J1+3
6500  ENDDO ; ENDIF
6510  RETURN ; END
6520C
6530  SUBROUTINE LIMIT (IS,JS,IE,JE)
6540C

```



```

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

```

```

7290 DO 20 J=2,JF
7300 DO 20 I=2,IF
7310 IF(ABS(Z(I,J,2)).LT.1.0E-5)Z(I,J,2)=0.0
7320 20 CONTINUE
7330 RETURN
7340 END
7350C
7360 SUBROUTINE MASS3 (IF,JF,Z,M,N,H,DX,DT,LP,LT)
7370C
7380 REAL M,N
7390 DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
7400 DIMENSION H(IF,JF)
7410 IF(MOD(LP,LT).NE.0)RETURN
7420 DO 10 J=2,JF
7430 DO 10 I=2,IF
7440 IF(H(I,J).GT.0.0)THEN
7450 Z(I,J,2)=Z(I,J,1)-DT/DX*(M(I,J,1)-M(I-1,J,1)
7460 & +N(I,J,1)-N(I,J-1,1))
7470 ELSE
7480 Z(I,J,2)=0.0
7490 ENDIF
7500 10 CONTINUE
7510 DO 20 J=2,JF
7520 DO 20 I=2,IF
7530 IF(ABS(Z(I,J,2)).LT.1.0E-5)Z(I,J,2)=0.0
7540 20 CONTINUE
7550 RETURN ; END
7560C
7570 SUBROUTINE MMNT1 (IS,JS,IE,JE)
7580C
7590 REAL M,N
7600 PARAMETER (IO=1020,JO=738)
7610 COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
7620 COMMON /HH/ H(IO,JO)
7630 COMMON /RR/ R1(IO,JO),R2(IO,JO),R3(IO,JO),
7640 & R4(IO,JO),R5(IO,JO),R6(IO)
7650 COMMON /VV/ V1(IO),V2(IO),V3(IO),V4(IO),V5(IO)
7660 DO 10 J=JS,JE
7670 DO 10 I=IS,IE
7680 V1(I)=Z(I+1,J,2)-Z(I,J,2)
7690 V2(I)=N(I,J-1,1)+N(I,J,1)+N(I+1,J-1,1)+N(I+1,J,1)
7700 M(I,J,2)=M(I,J,1)-R2(I,J)*V1(I)+R3(I,J)*V2(I)
7710 10 CONTINUE
7720 IF(JS.LE.2)THEN
7730 DO 15 I=1,IO-1
7740 15 M(I,1,2)=M(I,1,1)-R2(I,1)*(Z(I+1,1,2)-Z(I,1,2))
7750 ENDIF
7760 IF(JE.GE.JO-1)THEN
7770 DO 16 I=1,IO-1
7780 16 M(I,JO,2)=M(I,JO,1)-R2(I,JO)*(Z(I+1,JO,2)-Z(I,JO,2))
7790 ENDIF
7800 IF(IS.LE.2)THEN
7810 DO 17 J=1,JO
7820 17 M(1,J,2)=M(1,J,1)-R2(1,J)*(Z(2,J,2)-Z(1,J,2))
7830 ENDIF
7840C
7850 DO 20 J=JS,JE
7860 DO 20 I=IS,IE
7870 V1(I)=Z(I,J+1,2)-Z(I,J,2)
7880 V2(I)=M(I-1,J,1)+M(I,J,1)+M(I-1,J+1,1)+M(I,J+1,1)
7890 N(I,J,2)=N(I,J,1)-R4(I,J)*V1(I)-R5(I,J)*V2(I)
7900 20 CONTINUE
7910 IF(IS.LE.2)THEN
7920 DO 25 J=1,JO-1
7930 25 N(1,J,2)=N(1,J,1)-R4(1,J)*(Z(1,J+1,2)-Z(1,J,2))
7940 ENDIF
7950 IF(IE.GE.IO-1)THEN
7960 DO 26 J=1,JO-1
7970 26 N(IO,J,2)=N(IO,J,1)-R4(IO,J)*(Z(IO,J+1,2)-Z(IO,J,2))
7980 ENDIF
7990 IF(JS.LE.2)THEN
8000 DO 27 I=1,IO
8010 27 N(I,1,2)=N(I,1,1)-R4(I,1)*(Z(I,2,2)-Z(I,1,2))
8020 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))

```

```

10250 L=L+1 ; ENDIF
10260 30 CONTINUE
10270 IF(L.GE.5)THEN
10280 ZX(II,JJ,2)=S/L ; ELSE
10290 ZX(II,JJ,2)=0.0 ; ENDIF
10300 II=II+1 ; I1=I1+3
10310 ENDDO ; ENDIF
10320C
10330 IF(CHK.EQ.1)THEN
10340 II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
10350 DO WHILE (J1+2.LE.JES)
10360 S=0.0 ; L=0
10370 DO 40 I=I1,I1+2
10380 DO 40 J=J1,J1+2
10390 IF(HY(I,J).GT.0.0)THEN
10400 S=S+0.5*(ZY(I,J,1)+ZY(I,J,2))
10410 L=L+1 ; ENDIF
10420 40 CONTINUE
10430 IF(L.GE.5)THEN
10440 ZX(II,JJ,2)=S/L ; ELSE
10450 ZX(II,JJ,2)=0.0 ; ENDIF
10460 JJ=JJ+1 ; J1=J1+3
10470 ENDDO ; ENDIF
10480 RETURN ; END
10490C
10500 SUBROUTINE J2Z (IX,JX,IY,JY,ZX,ZY,HY
10510 &,L0,BCHK,LP,LT)
10520C
10530 INTEGER BCHK,CHK
10540 DIMENSION ZX(IX,JX,2),ZY(IY,JY,2)
10550 DIMENSION HY(IY,JY),L0(4)
10560 IF(MOD(LP,LT).NE.0)RETURN
10570 ISS=2 ; JSS=2 ; IES=IY ; JES=JY
10580 ISL=L0(1) ; JSL=L0(2) ; IEL=L0(3) ; JEL=L0(4)
10590 CHK=BCHK
10600 KB=CHK/1000
10610 IF(KB.EQ.1)THEN
10620 CHK=CHK-1000
10630 II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
10640 DO WHILE (I1+2.LE.IES)
10650 S=0.0 ; L=0
10660 DO 10 I=I1,I1+2
10670 DO 10 J=J1,J1+2
10680 IF(HY(I,J).GT.0.0)THEN
10690 S=S+ZY(I,J,2)
10700 L=L+1 ; ENDIF
10710 10 CONTINUE
10720 IF(L.GE.5)THEN
10730 ZX(II,JJ,2)=S/L ; ELSE
10740 ZX(II,JJ,2)=0.0 ; ENDIF
10750 II=II+1 ; I1=I1+3
10760 ENDDO ; ENDIF
10770C
10780 KB=CHK/100
10790 IF(KB.EQ.1)THEN
10800 CHK=CHK-100
10810 II=IEL ; JJ=JSL ; I1=IES-2 ; J1=JSS
10820 DO WHILE (J1+1.LE.JES)
10830 S=0.0 ; L=0
10840 DO 20 I=I1,I1+2
10850 DO 20 J=J1,J1+2
10860 IF(HY(I,J).GT.0.0)THEN
10870 S=S+ZY(I,J,2)
10880 L=L+1 ; ENDIF
10890 20 CONTINUE
10900 IF(L.GE.5)THEN
10910 ZX(II,JJ,2)=S/L ; ELSE
10920 ZX(II,JJ,2)=0.0 ; ENDIF
10930 JJ=JJ+1 ; J1=J1+3
10940 ENDDO ; ENDIF
10950C
10960 KB=CHK/10
10970 IF(KB.EQ.1)THEN
10980 CHK=CHK-10

```

```

10990  II=ISL ; JJ=JEL ; I1=ISS ; J1=JES-2
11000  DO WHILE (I1+2.LE.IES)
11010  S=0.0 ; L=0
11020  DO 30 I=I1,I1+2
11030  DO 30 J=J1,J1+2
11040  IF(HY(I,J).GT.0.0)THEN
11050  S=S+ZY(I,J,2)
11060  L=L+1 ; ENDIF
11070  30 CONTINUE
11080  IF(L.GE.5)THEN
11090  ZX(II,JJ,2)=S/L ; ELSE
11100  ZX(II,JJ,2)=0.0 ; ENDIF
11110  II=II+1 ; I1=I1+3
11120  ENDDO ; ENDIF
11130C
11140  IF(CHK.EQ.1)THEN
11150  II=ISL ; JJ=JSL ; I1=ISS ; J1=JSS
11160  DO WHILE (J1+2.LE.JES)
11170  S=0.0 ; L=0
11180  DO 40 I=I1,I1+2
11190  DO 40 J=J1,J1+2
11200  IF(HY(I,J).GT.0.0)THEN
11210  S=S+ZY(I,J,2)
11220  L=L+1 ; ENDIF
11230  40 CONTINUE
11240  IF(L.GE.5)THEN
11250  ZX(II,JJ,2)=S/L ; ELSE
11260  ZX(II,JJ,2)=0.0 ; ENDIF
11270  JJ=JJ+1 ; J1=J1+3
11280  ENDDO ; ENDIF
11290  RETURN ; END
11300C
11310  SUBROUTINE JNQOA
11320C
11330  REAL M,MA,N,NA
11340  PARAMETER (IO=1020,JO=738,IA=76,JA=76)
11350  PARAMETER (DO=10.0,DA=4.0)
11360  PARAMETER (ISS=2,JSS=2,IES=76,JES=76)
11370  PARAMETER (ISL=121,JSL=592,IEL=150,JEL=621)
11380  COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
11390  COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
11400  I=ISS ; J=JSS-1 ; II=ISL-1 ; JJ=JSL-1
11410  DO WHILE(I.LE.IES)
11420  SI=(I-ISS+1.75)*DA/DO
11430  IS=IFIX(SI) ; DI=SI-IS ; II=IS+ISL-1
11440  NA(I,J,2)=(1-DI)*N(II,JJ,2)+DI*N(II+1,JJ,2)
11450  I=I+1
11460  ENDDO
11470C
11480  I=IES ; J=JSS ; II=IEL ; JJ=JSL-1
11490  DO WHILE(J.LE.JES)
11500  SJ=(J-JSS+1.75)*DA/DO
11510  JS=IFIX(SJ) ; DJ=JS-JS ; JJ=JS+JSL-1
11520  MA(I,J,2)=(1-DJ)*M(II,JJ,2)+DJ*M(II,JJ+1,2)
11530  J=J+1
11540  ENDDO
11550C
11560  I=ISS-1 ; J=JSS ; II=ISL-1 ; JJ=JSL-1
11570  DO WHILE(J.LE.JES)
11580  SJ=(J-JSS+1.75)*DA/DO
11590  JS=IFIX(SJ) ; DJ=JS-JS ; JJ=JS+JSL-1
11600  MA(I,J,2)=(1-DJ)*M(II,JJ,2)+DJ*M(II,JJ+1,2)
11610  J=J+1
11620  ENDDO
11630  RETURN ; END
11640C
11650  SUBROUTINE JNQ (IX,JX,IY,JY,MX,NX,MY,NY
11660  &,HY,L0,BCHK,LP,LT)
11670C
11680  INTEGER BCHK,CHK
11690  REAL MX,NX,MY,NY
11700  DIMENSION MX(IX,JX,2),NX(IX,JX,2),HY(IY,JY)
11710  DIMENSION MY(IY,JY,2),NY(IY,JY,2),L0(4)
11720  IF(MOD(LP,LT).NE.0)RETURN

```

```

11730 ISS=2 ; JSS=2 ; IES=IY ; JES=JY
11740 ISL=L0(1) ; JSL=L0(2) ; IEL=L0(3) ; JEL=L0(4)
11750 CHK=BCHK
11760 KB=CHK/1000
11770 IF(KB.EQ.1)THEN
11780 CHK=CHK-1000
11790 I=ISS ; J=JSS-1 ; II=ISL-1 ; JJ=JSL-1
11800 DO WHILE(I.LE.IES)
11810 SI=(I-ISS+2)/3.0
11820 IS=IFIX(SI) ; DI=SI-IS ; II=IS+ISL-1
11830 NY(I,J,2)=(1-DI)*NX(II,JJ,2)+DI*NX(II+1,JJ,2)
11840 IF(HY(I,J).LT.0.0)NY(I,J,2)=0.0
11850 I=I+1
11860 ENDDO ; ENDIF
11870C
11880 KB=CHK/100
11890 IF(KB.EQ.1)THEN
11900 CHK=CHK-100
11910 I=IES ; J=JSS ; II=IEL ; JJ=JSL-1
11920 DO WHILE(J.LE.JES)
11930 SJ=(J-JSS+2)/3.0
11940 JS=IFIX(SJ) ; DJ=JS-JS ; JJ=JS+JSL-1
11950 MY(I,J,2)=(1-DJ)*MX(II,JJ,2)+DJ*MX(II,JJ+1,2)
11960 IF(HY(I,J).LT.0.0)MY(I,J,2)=0.0
11970 J=J+1
11980 ENDDO ; ENDIF
11990C
12000 KB=CHK/10
12010 IF(KB.EQ.1)THEN
12020 CHK=CHK-10
12030 I=ISS ; J=JES ; II=ISL-1 ; JJ=JEL
12040 DO WHILE(I.LE.IES)
12050 SI=(I-ISS+2)/3.0
12060 IS=IFIX(SI) ; DI=SI-IS ; II=IS+ISL-1
12070 NY(I,J,2)=(1-DI)*NX(II,JJ,2)+DI*NX(II+1,JJ,2)
12080 IF(HY(I,J).LT.0.0)NY(I,J,2)=0.0
12090 I=I+1
12100 ENDDO ; ENDIF
12110C
12120 IF(CHK.EQ.1)THEN
12130 I=ISS-1 ; J=JSS ; II=ISL-1 ; JJ=JSL-1
12140 DO WHILE(J.LE.JES)
12150 SJ=(J-JSS+2)/3.0
12160 JS=IFIX(SJ) ; DJ=JS-JS ; JJ=JS+JSL-1
12170 MY(I,J,2)=(1-DJ)*MX(II,JJ,2)+DJ*MX(II,JJ+1,2)
12180 IF(HY(I,J).LT.0.0)MY(I,J,2)=0.0
12190 J=J+1
12200 ENDDO ; ENDIF
12210 RETURN ; END
12220C
12230 SUBROUTINE COUT (KK,KD)
12240C
12250 REAL M,N,MA,NA,MB,NB,MC,NC,ME,NE,MF,NF
12260 REAL MD1,ND1,MD2,ND2,MD3,ND3,MD4,ND4,MD5,ND5
12270 PARAMETER (IO=1020,JO=738,IA=76,JA=76)
12280 PARAMETER (IB=136,JB=136,IC=154,JC=307)
12290 PARAMETER (ID1=28,JD1=55,ID2=55,JD2=40)
12300 PARAMETER (ID3=37,JD3=31,ID4=13,JD4=40)
12310 PARAMETER (ID5=46,JD5=40)
12320 PARAMETER (IE1=34,JE1=67,IF1=46,IF1=58)
12330 COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
12340 COMMON /XA/ ZA(IA,JA,2),MA(IA,JA,2),NA(IA,JA,2)
12350 COMMON /HA/ HA(IA,JA)
12360 COMMON /XB/ ZB(IB,JB,2),MB(IB,JB,2),NB(IB,JB,2)
12370 COMMON /HB/ HB(IB,JB)
12380 COMMON /XC/ ZC(IC,JC,2),MC(IC,JC,2),NC(IC,JC,2)
12390 COMMON /HC/ HC(IC,JC)
12400 COMMON /XD1/ ZD1(ID1,JD1,2),MD1(ID1,JD1,2),
12410 &ND1(ID1,JD1,2),HD1(ID1,JD1)
12420 COMMON /XD2/ ZD2(ID2,JD2,2),MD2(ID2,JD2,2),
12430 &ND2(ID2,JD2,2),HD2(ID2,JD2)
12440 COMMON /XD3/ ZD3(ID3,JD3,2),MD3(ID3,JD3,2),
12450 &ND3(ID3,JD3,2),HD3(ID3,JD3)
12460 COMMON /XD4/ ZD4(ID4,JD4,2),MD4(ID4,JD4,2),

```



```

12470      &ND4(ID4,JD4,2),HD4(ID4,JD4)
12480      COMMON /XD5/ ZD5(ID5,JD5,2),MD5(ID5,JD5,2),
12490      &ND5(ID5,JD5,2),HD5(ID5,JD5)
12500      COMMON /XE/ ZE(IE1,JE1,2),ME(IE1,JE1,2),
12510      &NE(IE1,JE1,2),HE(IE1,JE1)
12520      COMMON /XF/ ZF(IF1,JF1,2),MF(IF1,JF1,2),
12530      &NF(IF1,JF1,2),HF(IF1,JF1)
12540      IF(KK.LT.180*26)RETURN
12550C     CALL OUTPUT1 (KK,KD)
12560     CALL OUTPUTQ (KK,KD)
12570     CALL OUTPUT3 (IA,JA,ZA,HA,KK,KD,1)
12580     CALL OUTPUT2 (IB,JB,ZB,HB,KK,KD,2)
12590     CALL OUTPUT2 (IC,JC,ZC,HC,KK,KD,3)
12600     CALL OUTPUT3 (ID1,JD1,ZD1,HD1,KK,KD,4)
12610     CALL OUTPUT3 (ID2,JD2,ZD2,HD2,KK,KD,5)
12620     CALL OUTPUT3 (ID3,JD3,ZD3,HD3,KK,KD,6)
12630     CALL OUTPUT3 (ID4,JD4,ZD4,HD4,KK,KD,7)
12640     CALL OUTPUT3 (ID5,JD5,ZD5,HD5,KK,KD,8)
12650     CALL OUTPUT3 (IE1,JE1,ZE,HE,KK,KD,9)
12660     CALL OUTPUT3 (IF1,JF1,ZF,HF,KK,KD,10)
12670     RETURN ; END
12680C
12690     SUBROUTINE OUTPUT1 (KK,KD)
12700C
12710     REAL M,N
12720     PARAMETER (IO=1020,JO=738)
12730     PARAMETER (GX=1.E-2)
12740     COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
12750     COMMON /HH/ H(IO,JO)
12760     DIMENSION KW(100),KM(23)
12770     DATA KM/1HJ,1HI,1HH,1HG,1HF,1HE,1HD,1HC,
12780     & 1HB,1HA,1H.,1H1,1H2,1H3,1H4,1H5,
12790     & 1H6,1H7,1H8,1H9,1H0,1H*,1H /
12800     IF(MOD(KK,KD).NE.0)RETURN
12810     WRITE(6,410)KK,(KI,KI=130,289,20)
12820     DO 30 KY=1,60
12830     KJ=60-KY+1
12840     J1=12*(KJ-1)+1
12850     J2=J1+11
12860     DO 40 KI=1,85
12870     I1=12*(KI-1)+1
12880     I2=I1+11
12890     L=0 ; S=0.
12900     DO 41 J=J1,J2
12910     DO 41 I=I1,I2
12920     IF(H(I,J).GT.0.)THEN
12930     L=L+1 ; S=S+Z(I,J,2)
12940     ENDIF
12950 41 CONTINUE
12960     IF(L.GE.60)THEN
12970     S=S/L
12980     KMM=11+INT(S/0.1)
12990     IF(KMM.GT.21)KMM=21
13000     IF(KMM.LT.1)KMM=1
13010     IF(ABS(S).LT.GX)KMM=23
13020     ELSE
13030     KMM=22
13040     ENDIF
13050     KW(KI)=KM(KMM)
13060 40 CONTINUE
13070     JK=2*KJ-60
13080     WRITE(6,400)JK,(KW(I),I=1,85)
13090 30 CONTINUE
13100     RETURN
13110 400 FORMAT(1H ,I3,2X,85A1)
13120 410 FORMAT(1H1/,2X,"MAP AT K=",I5,/,6X,11I10)
13130     END
13140C
13150     SUBROUTINE OUTPUT2 (IF,JF,Z,H,KK,KD,CHR)
13160C
13170     INTEGER CHR
13180     PARAMETER (GX=1.E-2)
13190     DIMENSION Z(IF,JF,2),H(IF,JF)
13200     DIMENSION KW(100),KM(23)

```

```

13210 CHARACTER NAME(10)*10
13220 DATA KM/1HJ,1HI,1HH,1HG,1HF,1HE,1HD,1HC,
13230 & 1HB,1HA,1H.,1H1,1H2,1H3,1H4,1H5,
13240 & 1H6,1H7,1H8,1H9,1H0,1H*,1H /
13250 NAME(1)="A REGION"
13260 NAME(2)="B REGION"
13270 NAME(3)="C REGION"
13280 NAME(4)="D1 REGION"
13290 NAME(5)="D2 REGION"
13300 NAME(6)="D3 REGION"
13310 NAME(7)="D4 REGION"
13320 NAME(8)="D5 REGION"
13330 NAME(9)="E REGION"
13340 NAME(10)="F REGION"
13350 IF(MOD(KK,KD).NE.0)RETURN
13360 WRITE(6,450)NAME(CHR)
13370 450 FORMAT(1H ./,10X,A10)
13380 IN=IF/20
13390 WRITE(6,410)KK,(KI,KI=20,IN*20,20)
13400 DO 10 J=JF,1,-2
13410 JS=J-1 ; JE=J ; IF(JS.LE.0)JS=1
13420 DO 20 I=1,IF/2
13430 IS=2*I-1 ; IE=IS+1 ; IF(IE.GE.IF)IE=IF
13440 S=0.0 ; L=0
13450 DO 30 II=IS,IE
13460 DO 30 JJ=JS,JE
13470 IF(H(II,JJ).GT.0.)THEN
13480 L=L+1
13490 S=S+Z(II,JJ,2)
13500 ENDIF
13510 30 CONTINUE
13520 IF(L.GE.2)THEN
13530 S=S/L
13540 KMM=11+INT(S/0.5)
13550 IF(KMM.GT.21)KMM=21
13560 IF(KMM.LT.1)KMM=1
13570 IF(ABS(S).LT.GX)KMM=23
13580 ELSE
13590 KMM=22
13600 ENDIF
13610 KW(I)=KM(KMM)
13620 20 CONTINUE
13630 II=IF/2
13640 WRITE(6,400)J,(KW(I),I=1,II)
13650 10 CONTINUE
13660 RETURN
13670 400 FORMAT(1H ,I3,2X,100A1)
13680 410 FORMAT(1H ./,2X,"MAP AT K=",I5,/,6X,11I10)
13690 END
13700C
13710 SUBROUTINE OUTPUT3 (IF,JF,Z,H,KK,KD,CHR)
13720C
13730 INTEGER CHR
13740 PARAMETER (GX=1.E-2)
13750 DIMENSION Z(IF,JF,2),H(IF,JF)
13760 DIMENSION KW(100),KM(23)
13770 CHARACTER NAME(10)*10
13780 DATA KM/1HJ,1HI,1HH,1HG,1HF,1HE,1HD,1HC,
13790 & 1HB,1HA,1H.,1H1,1H2,1H3,1H4,1H5,
13800 & 1H6,1H7,1H8,1H9,1H0,1H*,1H /
13810 NAME(1)="A REGION"
13820 NAME(2)="B REGION"
13830 NAME(3)="C REGION"
13840 NAME(4)="D1 REGION"
13850 NAME(5)="D2 REGION"
13860 NAME(6)="D3 REGION"
13870 NAME(7)="D4 REGION"
13880 NAME(8)="D5 REGION"
13890 NAME(9)="E REGION"
13900 NAME(10)="F REGION"
13910 IF(MOD(KK,KD).NE.0)RETURN
13920 IN=IF/10 ; JN=JF/10
13930 WRITE(6,450)NAME(CHR)
13940 450 FORMAT(1H ./,10X,A10)

```

```

13950  WRITE(6,410)KK,(KI,KI=10,100,10)
13960  DO 10 J=JF,1,-1
13970  IE=IF ; IF(IE.GE.100)IE=100
13980  DO 20 I=1,IE
13990  IF(H(I,J).GT.0.0)THEN
14000  KMM=11+INT(Z(I,J,2)/0.5)
14010  IF(KMM.GT.21)KMM=21
14020  IF(KMM.GT.21)KMM=21
14030  IF(KMM.LT.1)KMM=1
14040  IF(ABS(Z(I,J,2)).LT.GX)KMM=23
14050  ELSE
14060  KMM=22
14070  ENDIF
14080  KW(I)=KM(KMM)
14090  20 CONTINUE
14100  WRITE(6,400)J,(KW(I),I=1,IE)
14110  10 CONTINUE
14120  RETURN
14130  400 FORMAT(1H ,I3,2X,100A1)
14140  410 FORMAT(1H ,/,2X,"MAP AT K=",I5,/,6X,11I10)
14150  END
14160C
14170  SUBROUTINE OUTPUTQ (KK,KD)
14180C
14190  REAL M,N
14200  PARAMETER (GX=1.E-2)
14210  PARAMETER (IO=1020,JO=738)
14220  COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
14230  COMMON /HH/ H(IO,JO)
14240  DIMENSION KW(100),KM(23)
14250  DATA KM/1HJ,1HI,1HH,1HG,1HF,1HE,1HD,1HC,
14260  & 1HB,1HA,1H.,1H1,1H2,1H3,1H4,1H5,
14270  & 1H6,1H7,1H8,1H9,1H0,1H*,1H /
14280  IF(MOD(KK,KD).NE.0)RETURN
14290  IS=101 ; IE=200 ; JS=581 ; JE=630
14300  WRITE(6,410)KK,(KI,KI=100,200,10)
14310  DO 10 J=JE,JS,-1
14320  DO 20 I=1,100
14330  JJ=J ; II=I+100
14340  IF(H(II,JJ).GT.0.0)THEN
14350  KMM=11+INT(Z(II,JJ,2)/0.1)
14360  IF(KMM.GT.21)KMM=21
14370  IF(KMM.GT.21)KMM=21
14380  IF(KMM.LT.1)KMM=1
14390  IF(ABS(Z(II,JJ,2)).LT.GX)KMM=23
14400  ELSE
14410  KMM=22
14420  ENDIF
14430  KW(I)=KM(KMM)
14440  20 CONTINUE
14450  WRITE(6,400)J,(KW(I),I=1,100)
14460  10 CONTINUE
14470  RETURN
14480  400 FORMAT(1H ,I3,2X,100A1)
14490  410 FORMAT(1H ,///,2X,"MAP AT K=",I5,/,6X,11I10)
14500  END
14510C
14520  SUBROUTINE CHANGE
14530C
14540  REAL M,N
14550  PARAMETER (IO=1020,JO=738)
14560  COMMON /XX/ Z(IO,JO,2),M(IO,JO,2),N(IO,JO,2)
14570  DO 10 J=1,JO
14580  DO 10 I=1,IO
14590  Z(I,J,1)=Z(I,J,2)
14600  M(I,J,1)=M(I,J,2)
14610  N(I,J,1)=N(I,J,2)
14620  10 CONTINUE
14630  RETURN ; END
14640C
14650  SUBROUTINE CHANG (IF,JF,Z,M,N,LP,LT)
14660C
14670  REAL M,N
14680  DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)

```

```

14690 IF(MOD(LP,LT).NE.0)RETURN
14700 DO 10 J=1,JF
14710 DO 10 I=1,IF
14720 Z(I,J,1)=Z(I,J,2)
14730 M(I,J,1)=M(I,J,2)
14740 N(I,J,1)=N(I,J,2)
14750 10 CONTINUE
14760 RETURN ; END
14770C
14780 SUBROUTINE FILEOT (IF,JF,Z,M,N)
14790C
14800 REAL M,N
14810 DIMENSION Z(IF,JF,2),M(IF,JF,2),N(IF,JF,2)
14820C
14830 DO 10 J=1,JF
14840 WRITE(25,100)(Z(I,J,2),I=1,IF)
14850 WRITE(25,100)(M(I,J,2),I=1,IF)
14860 WRITE(25,100)(N(I,J,2),I=1,IF)
14870 10 CONTINUE
14880 100 FORMAT(10E12.5)
14890 RETURN
14900 END
14910C
14920 SUBROUTINE MAX (IF,JF,IS,JS,IE,JE,Z,H,ZM,ZN)
14930C
14940 DIMENSION Z(IF,JF,2),H(IF,JF),ZM(IF,JF),ZN(IF,JF)
14950 DO 10 I=IS,IE
14960 DO 10 J=JS,JE
14970 IF(H(I,J).LT.0.0)GOTO 10
14980 IF(ZM(I,J).LT.Z(I,J,2))ZM(I,J)=Z(I,J,2)
14990 IF(ZN(I,J).GT.Z(I,J,2))ZN(I,J)=Z(I,J,2)
15000 10 CONTINUE
15010 RETURN
15020 END
15030C
15040 SUBROUTINE POINT (IF,JF,Z,NG,PZ,IP,JP,KK,KC)
15050C
15060 DIMENSION Z(IF,JF,2),PZ(NG),IP(NG),JP(NG)
15070C
15080 IF(MOD(KK,KC).NE.0)RETURN
15090 DO 10 K=1,NG
15100 PZ(K)=Z(IP(K),JP(K),2)
15110 10 CONTINUE
15120 WRITE(24,100)(PZ(K),K=1,NG)
15130 100 FORMAT(10F7.3)
15140 RETURN
15150 END
15160C
15170 SUBROUTINE PROP A (IF,JF,IS,JS,IE,JE,Z,H,LF,KK)
15180C
15190 DIMENSION Z(IF,JF,2),H(IF,JF),LF(IF,JF)
15200C
15210 LK=KK/180+1
15220 DO 10 I=IS,IE
15230 DO 10 J=JS,JE
15240 IF(H(I,J).GT.0.0)THEN
15250 IF(LF(I,J).EQ.0)THEN
15260 IF(ABS(Z(I,J,2)).GT.0.01)LF(I,J)=LK
15270 ENDIF
15280 ELSE
15290 LF(I,J)=-9
15300 ENDIF
15310 10 CONTINUE
15320 RETURN
15330 END
15340C
15350 SUBROUTINE OUTDT (IF,JF,ZM,ZN,LK)
15360 DIMENSION ZM(IF,JF),ZN(IF,JF),LK(IF,JF)
15370 DO 20 I=1,IF
15380 DO 20 J=1,JF
15390 ZN(I,J)=ZM(I,J)-ZN(I,J)
15400 IF(ZN(I,J).GT.100.0)ZN(I,J)=100.0
15410 IF(ZM(I,J).GT.100.0)ZM(I,J)=100.0
15420 20 CONTINUE

```

```

15430 DO 10 J=1,JF
15440 WRITE(31,100)(ZM(I,J),I=1,IF)
15450 WRITE(32,100)(ZN(I,J),I=1,IF)
15460 WRITE(33,200)(LK(I,J),I=1,IF)
15470 10 CONTINUE
15480 100 FORMAT(20F6.3)
15490 200 FORMAT(40I3)
15500 RETURN
15510 END
15520C
15530 SUBROUTINE DEFORM (IO,JO,IS,JS,IE,JE,Z)
15540 REAL L
15550 PARAMETER (DR=10.0)
15560 PARAMETER (A=3.141592,B=4.848E-06)
15570 PARAMETER (RR=6.37E+6,E=1.7453E-2)
15580 PARAMETER (HH=53.0E+3,D=24.0,DL=10.0,TH=10.0,RD=90.0)
15590 PARAMETER (L=800.0E+3,W=200.0E+3)
15600 PARAMETER (YO=-50.0,XO=280.0)
15610 PARAMETER (YO=-46.0,XO=284.0)
15620 DIMENSION Z(IO,JO,2)
15630 XL=A*RR*(XO-XO)*COS(E*YO)/180.0 ; YL=A*RR*(YO-YO)/180.0
15640 H1=HH/SIN(E*DL) ; H2=HH/SIN(E*DL)+W
15650 DS=D*COS(E*RD) ; DD=D*SIN(E*RD)
15660 WRITE(6,*)XL,YL,H1,H2,DS,DD
15670 DO 10 I=1,IE-IS+1
15680 DO 10 J=1,JE-JS+1
15690C XX=DX*(I-1) ; YY=DX*(J-1)
15700 YY=A*RR*DR*(J-1)/(60.0*180)
15710 XX=A*RR*DR*(I-1)*COS(E*(YO+DR*(J-1)/60.0))/(60.0*180)
15720 X1=(XX-XL)*SIN(E*TH)+(YY-YL)*COS(E*TH)-L/2.0
15730 X2=(XX-XL)*COS(E*TH)-(YY-YL)*SIN(E*TH)+HH/TAN(E*DL)
15740 X3=0.0
15750 CALL USCAL (X1,X2,X3, L/2.,H2,E*DL,F1)
15760 CALL USCAL (X1,X2,X3, L/2.,H1,E*DL,F2)
15770 CALL USCAL (X1,X2,X3,-L/2.,H2,E*DL,F3)
15780 CALL USCAL (X1,X2,X3,-L/2.,H1,E*DL,F4)
15790 CALL UDCAL (X1,X2,X3, L/2.,H2,E*DL,G1)
15800 CALL UDCAL (X1,X2,X3, L/2.,H1,E*DL,G2)
15810 CALL UDCAL (X1,X2,X3,-L/2.,H2,E*DL,G3)
15820 CALL UDCAL (X1,X2,X3,-L/2.,H1,E*DL,G4)
15830 US=(F1-F2-F3+F4)*DS/(12.0*A)
15840 UD=(G1-G2-G3+G4)*DD/(12.0*A)
15850 II=I+IS-1 ; JJ=J+JS-1
15860 Z(II,JJ,1)=US+UD
15870 Z(II,JJ,2)=US+UD
15880 10 CONTINUE
15890 RETURN
15900 END
15910C
15920 SUBROUTINE USCAL (X1,X2,X3,C,CC,DP,F)
15930 REAL K
15940C
15950 SN=SIN(DP) ; CS=COS(DP)
15960 C1=C ; C2=CC*CS ; C3=CC*SN
15970 R=SQRT((X1-C1)**2+(X2-C2)**2+(X3-C3)**2)
15980 Q=SQRT((X1-C1)**2+(X2-C2)**2+(X3+C3)**2)
15990 R2=X2*SN-X3*CS
16000 R3=X2*CS+X3*SN
16010 Q2=X2*SN+X3*CS
16020 Q3=-X2*CS+X3*SN
16030 H=SQRT(Q2**2+(Q3+CC)**2)
16040 K=SQRT((X1-C1)**2+Q2**2)
16050 A1=LOG(R+R3-CC)
16060 A2=LOG(Q+Q3+CC)
16070 A3=LOG(Q+X3+C3)
16080 B1=1+3.0*(TAN(DP))**2
16090 B2=3.0*TAN(DP)/CS
16100 B3=2.0*R2*SN
16110 B4=Q2+X2*SN
16120 B5=2.0*R2**2*CS
16130 B6=R*(R+R3-CC)
16140 B7=4.0*Q2*X3*SN**2
16150 B8=2.0*(Q2+X2*SN)*(X3+Q3*SN)
16160 B9=Q*(Q+Q3+CC)

```

```

16170 B10=4.0*Q2*X3*SN
16180 B11=(X3+C3)-Q3*SN
16190 B12=4.0*Q2**2*Q3*X3*CS*SN
16200 B13=2.0*Q+Q3+CC
16210 B14=Q**3*(Q+Q3+CC)**2
16220 F=CS*(A1+B1*A2-B2*A3)
16230 &+B3/R+2*SN*B4/Q-B5/B6+(B7-B8)/B9+B10*B11/Q**3-B12*B13/B14
16240 RETURN
16250 END
16260C
16270 SUBROUTINE UDCAL (X1,X2,X3,C,CC,DP,F)
16280 REAL K
16290C
16300 SN=SIN(DP) ; CS=COS(DP)
16310 C1=C ; C2=CC*CS ; C3=CC*SN
16320 R=SQRT((X1-C1)**2+(X2-C2)**2+(X3-C3)**2)
16330 Q=SQRT((X1-C1)**2+(X2-C2)**2+(X3+C3)**2)
16340 R2=X2*SN-X3*CS
16350 R3=X2*CS+X3*SN
16360 Q2=X2*SN+X3*CS
16370 Q3=-X2*CS+X3*SN
16380 H=SQRT(Q2**2+(Q3+CC)**2)
16390 K=SQRT((X1-C1)**2+Q2**2)
16400 A1=LOG(R+X1-C1)
16410 A2=LOG(Q+X1-C1)
16420 B1=Q*(Q+X1-C1)
16430 B2=R*(R+X1-C1)
16440 B3=Q*(Q+Q3+CC)
16450 D1=X1-C1 ; D2=X2-C2 ; D3=X3-C3
16460 D4=X3+C3 ; D5=R3-CC ; D6=Q3+CC
16470 T1=ATN(D1*D2,(H+D4)*(Q+H))
16480 T2=ATN(D1*D5,R2*R)
16490 T3=ATN(D1*D6,Q2*Q)
16500 F=SN*(D2*(2*D3/B2+4*D3/B1-4*C3*X3*D4*(2*Q+D1)/(B1**2*Q))
16510 & -6*T1+3*T2-6*T3)
16520 &+CS*(A1-A2-2*(D3**2)/B2
16530 &-4*(D4**2-C3*X3)/B1-4*C3*X3*D4**2*(2*Q+X1-C1)/(B1**2*Q))
16540 &+6*X3*(CS*SN*(2*D6/B1+D1/B3)-Q2*(SN**2-CS**2)/B1)
16550 RETURN
16560 END
16570C
16580 REAL FUNCTION ATN (AX,AY)
16590 DATA GX/1.0E-6/
16600 AAX=ABS(AX) ; AAY=ABS(AY)
16610 P=AX*AY
16620 IF(AAX.LE.GX.AND.AAY.LE.GX)GOTO 10
16630 SR=ATAN2(AAX,AAY)
16640 ATN=SIGN(SR,P)
16650 RETURN
16660 10 WRITE(6,100)AX,AY
16670 100 FORMAT(1H,"ATAN -- AX=",E15.7,2X,"AY=",E15.7)
16680 ATN=0.2
16690 RETURN
16700 END

```