Numerical Model VBREAK for Vertically Two-Dimensional Breaking Waves on Impermeable Slopes

by

Nobuhisa Kobayashi and Bradley D. Johnson

Research Report No. CACR-95-06
September, 1995

Center for Applied Coastal Research
Ocean Engineering Laboratory
University of Delaware
Newark, DE 19716

Abstract

The computer program called VBREAK is developed to predict the time-dependent, two-dimensional velocity field under normally incident breaking waves on beaches and coastal structures. To reduce computation time considerably, use is made of the depth-integrated continuity and horizontal momentum equations. The momentum equation includes the momentum flux correction due to the vertical variation of the horizontal velocity. The bottom shear stress is expressed in terms of the near-bottom horizontal velocity immediately outside the thin wave boundary layer. The third equation for the momentum flux correction is derived from the depth-integrated wave energy equation. In order to express these three one-dimensional, time-dependent equations in terms of the three unknown variables of the water depth, depth-averaged horizontal velocity and near-bottom horizontal velocity, the normalized vertical profile of the horizontal velocity is assumed to be cubic on the analogy between turbulent bores and hydraulic jumps. Furthermore, the turbulent shear stress is assumed to be expressed using the turbulent eddy viscosity whose mixing length is proportional to the water depth.

The three governing equations are solved using the MacCormack finite difference method for its simplicity and success in the computation of hydraulic jumps. The seaward and landward boundary algorithms are extensions of those used in the previous one-dimensional models such as RBREAK2. The computer program VBREAK attached to this report consists of the main program, 22 subroutines and one function. The parameters and variables used in the program as well as the input and output are explained in detail so that a user will be able to modify and expand VBREAK. This first version of VBREAK does not allow wave overtopping and transmission. The armor stability and movement are not computed either. The developed numerical model has been compared with only two data sets of regular waves spilling on gentle uniform slopes. This user's manual will hopefully encourage other researchers to improve and expand VBREAK and apply it to various practical coastal engineering problems that require the time-dependent, two-dimensional horizontal and vertical velocities under breaking waves on beaches and coastal structures. VBREAK is tested using a workstation and is computationally as efficient as RBREAK2.

ACKNOWLEDGEMENT

This is the final report resulting from research sponsored by the Coastal Engineering Research Center, U. S. Army Engineer Waterways Experiment Station under contract No. DACW 39-93-C-0074. The authors would like to thank J. A. Melby, D. L. Ward, and D. D. Davidson for their enthusiastic support of our continued efforts to improve the capabilities of numerical models for the design of coastal structures.

Contents

PART I
INTRODUCTION

• 1.1 •
BACKGROUND

Available time-dependent, one-dimensional and other numerical models for breaking and nonbreaking waves on inclined structures and beaches were reviewed by Kobayashi and Poff (1994). The one-dimensional shallow-water models (e.g., Kobayashi and Wurjanto 1989; Kobayashi and Poff 1994) are relatively simple and robust. Generally, these models predict the free surface elevation fairly accurately, within about 20% errors. Raubenheimer et al. (1995) showed that the one-dimensional, shallow-water model was in good agreement with the variations of wave spectra and shapes (e.g., wave skewness) measured across the inner surf and swash zones on a gently sloping natural beach.

The one-dimensional models predict only the depth-averaged horizontal velocity. The vertical velocity may be estimated using the two-dimensional continuity equation together with the computed depth-averaged velocity, while the bottom shear stress may be expressed by a quadratic friction equation based on the depth-averaged velocity. The comparisons with the experiment for regular waves spilling on a rough, impermeable 1:35 slope conducted by Cox et al. (1995) indicated that the horizontal velocity measured below the wave trough level was represented by the computed depth-averaged velocity reasonably well. The computed vertical velocity represented the measured vertical velocity at least qualitatively except under the wave crest. The temporal variation of the bottom shear stress was predicted poorly because errors in the computed horizontal velocity were magnified in the computed bottom shear stress and because the bottom friction factor was not really constant. These limited comparisons suggest that a vertically two-dimensional model will be required to predict the detailed vertical variations of the fluid velocities and shear stress which are essential for predicting cross-shore sediment transport on beaches and hydrodynamic forces acting on armor units on coastal structures (e.g., Tørum 1994).

A simplified two-dimensional model is developed in this report. To reduce computational efforts considerably, the normalized vertical variation of the horizontal velocity outside the wave boundary layer is assumed to be cubic. The vertically two-dimensional problem is then reduced to a depth-integrated one-dimensional problem in which the three time-dependent, one-dimensional differential equations for the water depth h, depth-averaged horizontal velocity U and near-bottom horizontal velocity u_b need to be solved numerically. The simplified two-dimensional model called VBREAK is computationally as efficient as the previous one-dimensional models such as RBREAK2 (Kobayashi and Poff 1994). As a result, VBREAK can be applied easily and routinely using workstations.

It may also be noted that Kobayashi and Karjadi (1994, 1995) developed a horizontally two-dimensional, time-dependent model to predict the free surface elevation and the depth-averaged cross-shore and alongshore velocities in the swash and surf zones under obliquely incident regular and irregular waves. An effort is being made to combine the vertically and horizontally two-dimensional models and to develop a simplified three-dimensional, time-dependent model.

• 1.2 •

Outline of Report

The approximate governing equations adopted for VBREAK are derived in Part II. First, approximate two-dimensional equations for shallow-water waves on relatively gentle slopes are derived from the continuity and Reynolds equations. The approximate two-dimensional equations are then integrated vertically to obtain the depth-integrated continuity and horizontal momentum equations. This momentum equation includes the unknown momentum flux correction m due to the vertical variation of the horizontal velocity u. An equation for the momentum flux correction m is derived from the depth-integrated wave energy equation (Kobayashi and Wurjanto 1992). The bottom shear stress and wave energy dissipation inside the thin wave boundary layer are expressed in terms of the near-bottom horizontal velocity u_b and the wave friction factor (Jonsson 1966; Cox et al. 1995). The vertical variation of the horizontal velocity u outside the wave boundary layer normalized by the water depth h, the depth-averaged velocity U, and the near-bottom horizontal velocity u_b is assumed to be cubic on the analogy between turbulent bores and hydraulic jumps (Madsen and Svendsen 1983; Svendsen and Madsen 1984). The momentum flux correction m and the wave energy dissipation rate outside the wave boundary layer due to wave breaking are then expressed in terms of h, U and u_b. The three depth-integrated continuity, horizontal momentum, and momentum flux correction equations may thus be solved numerically to obtain the temporal and cross-shore variations of h, U and u_b. The previous one-dimensional models (e.g., Kobayashi and Wurjanto 1992) correspond to the special case of $u_b = U$ and zero momentum flux correction.

The numerical procedures adopted to solve the three governing equations with appropriate initial and boundary conditions are explained in detail in Part III of this report. The MacCormack finite difference method (MacCormack 1969) is selected because of its simplicity and success in the computation of unsteady open channel flows with hydraulic jumps (Chaudhry 1993). The computation is initiated at the time $t = 0$ when the specified incident wave train arrives at the seaward boundary and no wave action exists in the computation domain. The interval Δt of each time step for the time-marching computation is calculated using an approximate stability criterion of the adopted explicit finite difference method. Approximate seaward boundary conditions are used to compute the boundary values of h, U and u_b as well as the reflected wave train using the method of characteristics (Kobayashi et al. 1987, 1989). The landward boundary algorithm used in RBREAK2 (Kobayashi and Poff 1994) is modified to compute wave runup on the slope which is assumed to be impermeable. The options of wave overtopping and transmission included in RBREAK2 are not allowed in this first version of VBREAK . The options for computing armor stability and movement in RBREAK2 are not included either. The one-dimensional analyses of armor stability and movement by Kobayashi and Otta (1987) will need to be extended to the two-dimensional velocity field predicted by VBREAK .

The computer program VBREAK attached in Appendix A is explained in detail in Part IV. The main program, 22 subroutines and one function are described to an extent that a user will be able to comprehend the overall structure of VBREAK. The parameters and variables included in the COMMON blocks are explained such that a user will be able to follow the entire computer program line by line using the results presented in Parts II and III. The warning and error messages issued in VBREAK are listed in such a way that a user will be able to locate the origin of each message in VBREAK. All the input and output are described so meticulously that a user will be able to prepare the input files and retrieve the output files without difficulty.

The numerical model VBREAK has been compared with only two sets of regular wave data. One data set is the comprehensive measurements of test 1 presented by Stive (1980) and Stive and Wind (1982) in which the incident regular waves broke as spilling breakers on a concrete 1:40 beach. The other data set is the detailed velocity, bottom shear stress and free surface measurements by Cox et al. (1995) for the case of regular waves spilling on a rough, impermeable 1:35 slope. The one-dimensional models corresponding to VBREAK were compared with Stive's test 1 by Kobayashi et al. (1989) and with the test of Cox et al. (1995) by themselves. The comparisons of VBREAK with these tests are presented in a separate report by Johnson et al. (1995). A summary of these two reports is given in a paper by Kobayashi et al. (1995). The input and output used for the comparison of VBREAK with Stive's test 1 are presented as an example in Part V.

The summary and conclusions of this report is given in Part VI. It is obvious that VBREAK will need to be compared with irregular wave tests and coastal structure tests with much steeper slopes. Appendix A lists the computer program VBREAK . Appendix B explains the contents of the disk accompanying this report.

Part II
Mathematical Formulation

• 2.1 •
Two-Dimensional Equations in Shallow Water

The approximate governing equations adopted in the numerical model VBREAK are derived from the two-dimensional continuity and Reynolds equations (e.g., Rodi 1980)

$$\frac{\partial u'_j}{\partial x'_j} = 0 \tag{1}$$

$$\frac{\partial u'_i}{\partial t'} + u'_j \frac{\partial u'_i}{\partial x'_j} = -\frac{1}{\rho}\frac{\partial p'}{\partial x'_i} - g\delta_{i2} + \frac{1}{\rho}\frac{\partial \tau'_{ij}}{\partial x'_j} \tag{2}$$

in which the prime indicates the physical variables and the summention convention is used with respect to repeated indexes. The symbols used in (1) and (2) are depicted in Fig. 1 where t' = time; x'_1 = horizontal coordinate taken to be positive landward; x'_2 = vertical coordinate taken to be positive upward with $x'_2 = 0$ at the still water level (SWL); u'_1 = horizontal velocity; u'_2 = vertical velocity; ρ = fluid density which is assumed constant; p' = pressure; g = gravitational acceleration; δ_{i2} = Kronecker delta; and τ'_{ij} = sum of turbulent and viscous stresses. Assuming that the viscous stresses are negligible, τ'_{ij} may be expressed as (e.g., Rodi 1980)

$$\tau'_{ij} = \rho\left[\nu'_t\left(\frac{\partial u'_i}{\partial x'_j} + \frac{\partial u'_j}{\partial x'_i}\right) - \frac{2}{3}k'\delta_{ij}\right] \tag{3}$$

in which ν'_t = turbulent eddy viscosity; and k' = turbulent kinetic energy per unit mass.

To simplify (1) and (2) with (3) in shallow water, the dimensional variables may be normalized as

$$t = \frac{t'}{T'} \quad ; \quad x_1 = \frac{x'_1}{T'\sqrt{gH'}} \quad ; \quad x_2 = \frac{x'_2}{H'} \tag{4}$$

$$u_1 = \frac{u'_1}{\sqrt{gH'}} \quad ; \quad u_2 = \frac{u'_2}{H'/T'} \quad ; \quad p = \frac{p'}{\rho g H'} \tag{5}$$

$$\nu_t = \frac{\nu'_t}{H'^2/T'} \quad ; \quad k = \frac{k'}{\sqrt{gH'}H'/T'} \quad ; \quad \sigma = T'\sqrt{\frac{g}{H'}} \tag{6}$$

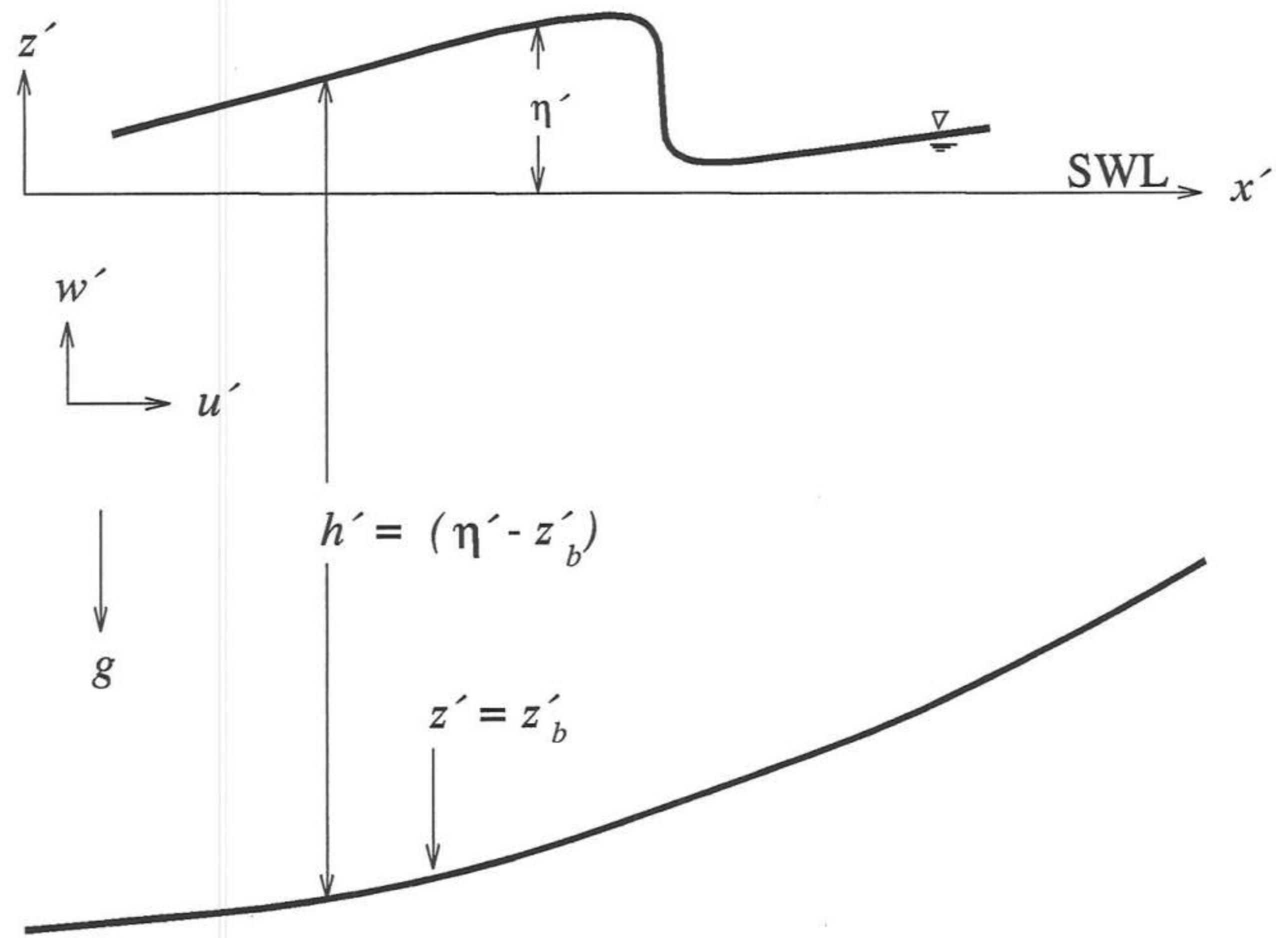

Figure 1: Definition Sketch for Vertically Two-Dimensional Waves

in which T' and H' are the reference wave period and height used for the normalization. The parameter σ defined in (6) is the ratio between the horizontal and vertical length scales. The normalized variables in (4) and (5) are assumed to be on the order of unity in shallow water. The normalization of ν_t' and k' in (6) is based on the turbulence measurements in a wave flume by Cox *et al.* (1994) which have indicated that ν_t and k are on the order of unity or less inside and immediately outside the surf zone, respectively.

Substituting (4)–(6) into (1)–(3), the normalized continuity and momentum equations are obtained. The conventional notations of $x = x_1$, $z = x_2$, $u = u_1$ and $w = u_2$ are used in the following. The normalized continuity equation is given by

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \tag{7}$$

The momentum equations are simplified under the assumption of $\sigma^2 \gg 1$ for shallow water waves. The approximate horizontal momentum equation is expressed as

$$\frac{\partial u}{\partial t} + u\,\frac{\partial u}{\partial x} + w\,\frac{\partial u}{\partial z} = -\frac{\partial}{\partial x}\left(p + \frac{2k}{3\sigma}\right) + \frac{\partial \tau}{\partial z} \tag{8}$$

with

$$\tau = \nu_t\,\frac{\partial u}{\partial z} \tag{9}$$

The approximate vertical momentum equation is written as

$$0 = -\frac{\partial}{\partial z}\left(p + z + \frac{2k}{3\sigma}\right) \tag{10}$$

The free surface and bottom are located at $z' = \eta'$ and $z' = z_b'$ as shown in Fig. 1 where the bottom is assumed to be fixed and impermeable. The water depth h' is given by $h' = (\eta' - z_b')$. The dimensional variables η', z_b' and h' are normalized by the vertical length scale H'

$$\eta = \frac{\eta'}{H'} \quad ; \quad z_b = \frac{z_b'}{H'} \quad ; \quad h = \frac{h'}{H'} \tag{11}$$

The kinematic boundary conditions at the free surface and bottom are expressed as

$$\frac{\partial \eta}{\partial t} + u\,\frac{\partial \eta}{\partial x} - w = 0 \qquad \text{at } z = \eta \tag{12}$$

$$u\,\frac{\partial z_b}{\partial x} - w = 0 \qquad \text{at } z = z_b \tag{13}$$

The normal and tangential stresses at the free surface are assumed to be zero. These boundary conditions for $\sigma^2 \gg 1$ can be shown to yield

$$p + \frac{2k}{3\sigma} = 0 \qquad \text{at } z = \eta \tag{14}$$

$$\tau = 0 \qquad \text{at } z = \eta \tag{15}$$

Integration of (10) with respect to z using (14) gives

$$p = \eta - z - \frac{2k}{3\sigma} \tag{16}$$

The pressure is approximately hydrostatic in shallow water where k is on the order of unity or less and σ is relatively large to satisfy $\sigma^2 \gg 1$. Substituting (16) into (8), the horizontal momentum equation is rewritten as

$$\frac{\partial u}{\partial t} + u\,\frac{\partial u}{\partial x} + w\,\frac{\partial u}{\partial z} = -\frac{\partial \eta}{\partial x} + \frac{\partial \tau}{\partial z} \tag{17}$$

Eqs. (7) and (17) together with (9), (12), (13) and (15) may be solved numerically but considerable numerical difficulties are expected because the unknown free surface elevation η varies rapidly in space and time. In addition, such a numerical model will be too time-consuming to compute breaking wave motions of long duration.

• 2.2 •

Depth-Integrated Equations

To reduce computational efforts significantly, (7) and (17) are integrated from $z = z_b$ to $z = \eta$ using (12), (13) and (15). No additional approximation is introduced in this integration. The depth-integrated continuity equation is expressed as

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{18}$$

where h = water depth given by $h = (\eta - z_b)$; and q = volume flux per unit width defined as

$$q = \int_{z_b}^{\eta} u\, dz \tag{19}$$

The depth-integrated horizontal momentum equation is written as

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x}\left(qU + m + \frac{1}{2}h^2\right) = -\theta h - \tau_b \tag{20}$$

with

$$m = \int_{z_b}^{\eta} (u - U)^2\, dz \tag{21}$$

in which U = depth-averaged horizontal velocity defined as $U = q/h$; θ = normalized bottom slope defined as $\theta = dz_b/dx$; τ_b = bottom shear stress; and m = momentum flux correction due to the vertical variation of the horizontal velocity u where $m = 0$ if $u = U$.

The previous one-dimensional models IBREAK (Kobayashi and Wurjanto 1989), RBREAK (Wurjanto and Kobayashi 1991), and RBREAK2 (Kobayashi and Poff 1994) assumed $m = 0$ and expressed τ_b in terms of U. Eqs. (18) and (20) with $m = 0$ were solved using the dissipative Lax-Wendroff finite difference method to compute h and q as a function of t and x. These one-dimensional models do not predict the vertical variations of the fluid velocities u and w. Furthermore, these models do not account for energy dissipation due to wave breaking explicitly.

Boussinesq equations have been extended to predict breaking waves on gentle slopes (Zelt 1991; Schäffer *et al.* 1992). Boussinesq equations without the dispersive terms correspond to (18) and (20) if the bottom friction is included in Boussinesq equations (Zelt 1991). Gharangik and Chaudhry (1991) computed hydraulic jumps using Boussinesq equations with and without the dispersive terms and found that the dispersive terms had little effect on the computed hydraulic jumps. This indicates that the dispersive terms may be negligible for breaking waves inside the surf zone. Moreover, the dispersive terms derived under the assumption of potential flow may not be valid for breaking waves. To include energy dissipation due to wave breaking in Boussinesq equations, Zelt (1991) and Schäffer *et al.* (1992) added a term corresponding to the term for the momentum flux correction m in (20). Zelt (1991) expressed this additional term in the form of horizontal momentum diffusion with an artificial viscosity proposed by Heitner and Housner (1970). The artificial viscosity was calibrated for breaking solitary waves where the diffusion term was activated using a semi-empirical criterion for solitary wave breaking. On the other hand, Schäffer *et al.* (1992) expressed the additional momentum flux using a simple approach based on a surface roller that represented a passive bulk of water riding on the front of a breaking wave. An empirical geometric method was used to determine the shape and location of the surface rollers during the computation. These models do not predict the vertical variations of the fluid velocities. It is also not certain whether the computed energy dissipation was truly caused by the term added to the momentum equation because they did not check whether the computed results satisfied the energy equation as will be elaborated in the following.

In this report, the equation for the momentum flux correction m is derived from the depth-integrated instantaneous wave energy equation (Kobayashi and Wurjanto 1992)

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}(E_F) = -D \tag{22}$$

which is obtained by integrating (17) multiplied by u from $z = z_b$ to $z = \eta$ by use of (12), (13), (15) and (18). The specific energy E defined as the sum of kinetic and potential energy per unit horizontal area is given by

$$E = \frac{1}{2}\left(qU + m + \eta^2\right) \qquad \text{for } z_b < 0 \tag{23a}$$

$$E = \frac{1}{2}\left(qU + m + \eta^2 - z_b^2\right) \qquad \text{for } z_b > 0 \tag{23b}$$

in which the potential energy is taken to be relative to the potential energy in the absence of wave action with SWL at $z = 0$. The energy flux E_F per unit width is expressed as

$$E_F = \eta q + \frac{1}{2}\left(qU^2 + 3mU + m_3\right) \tag{24}$$

with

$$m_3 = \int_{z_b}^{\eta} (u - U)^3 \, dz \tag{25}$$

in which m_3 = kinetic energy flux correction due to the third moment of the velocity deviation $(u - U)$ over the depth where $m_3 = 0$ if $u = U$. The energy dissipation rate D per unit horizontal area in (22) is given by

$$D = \int_{z_b}^{\eta} \tau \frac{\partial u}{\partial z} \, dz \tag{26}$$

where use is made of the no slip condition $u = 0$ at $z = z_b$.

The wave boundary layer is not analyzed explicitly in this numerical model. The energy dissipation rate D_f inside the wave boundary layer may be estimated by (Jonsson and Carlsen 1976)

$$D_f = \tau_b \, u_b \tag{27}$$

where u_b = near-bottom horizontal velocity immediately outside the wave boundary layer. The normalized bottom shear stress τ_b may be expressed as

$$\tau_b = f_w \mid u_b \mid u_b \quad ; \quad f_w = \frac{1}{2}\,\sigma f'_w \tag{28}$$

in which f'_w = wave friction factor (Jonsson 1966). The value of f'_w specified as input is allowed to vary spacially to accommodate the spacial variation of bottom roughness (Kobayashi and Raichle 1994). The previous one-dimensional models (e.g., Kobayashi and Wurjanto 1992) employed (27) and (28) in which the depth-averaged velocity U and the corresponding friction factor f' were used instead of the near-bottom velocity u_b and the wave friction factor f'_w. Cox *et al.* (1995) showed that the bottom shear stress and near-bottom velocity measured inside the surf zone could be related fairly well by the quadratic friction equation (28) with the wave friction factor f'_w estimated using the formula of Jonsson (1966).

The energy dissipation rate D given by (26) may be expressed as

$$D = D_f + D_B \tag{29}$$

in which D_B = energy dissipation rate outside the wave boundary layer due to wave breaking. Assuming that the thickness of the wave boundary layer is much smaller than the water depth, D_B may be estimated using (26) together with (9)

$$D_B = \int_{z_b}^{\eta} \nu_t \left(\frac{\partial u}{\partial z}\right)^2 dz \quad \text{outside boundary layer} \tag{30}$$

where the vertical variations of u and ν_t outside the wave boundary layer will be assumed in the following.

Rearranging the instantaneous wave energy equation (22) with (27) and (29) by use of (18) and (20), the equation for the momentum flux correction m is derived

$$\frac{\partial m}{\partial t} + \frac{\partial}{\partial x}(3mU + m_3) = 2U\,\frac{\partial m}{\partial x} - 2\,(\tau_b \tilde{u}_b + D_B) \tag{31}$$

with

$$\tilde{u}_b = u_b - U \tag{32}$$

in which $\tilde{u}_b$ = near-bottom horizontal velocity correction due to the vertical variation of the horizontal velocity u outside the wave boundary layer. If $u = U$, $\tilde{u}_b = 0$, $m = 0$ and $m_3 = 0$. As a result, (31) yields $D_B = 0$ if $u = U$, whereas D_B given by (30) outside the wave boundary layer is also zero if u is independent of z. This proves that the energy dissipation due to wave breaking in the previous one-dimensional models based on the assumptions of $\tilde{u}_b = 0$, $m = 0$ and $m_3 = 0$ is solely numerical (Kobayashi and Wurjanto 1992).

In order to express m, m_3 and D_B in terms of $\tilde{u}_b$, the horizontal velocity u outside the wave boundary layer is assumed to be expressible in the form

$$u(t,x,z) = U(t,x) + \tilde{u}_b(t,x)F(\zeta) \tag{33}$$

with

$$\zeta = [z - z_b(x)]/h(t,x) \quad \text{for } 0 \le \zeta \le 1 \tag{34}$$

in which F = normalized function expressing the vertical variation of the velocity deviation $(u - U)$ from $\zeta = 0$ immediately outside the wave boundary layer to $\zeta = 1$ at the free surface. Furthermore, the dimensional turbulent eddy viscosity ν_t' outside the wave boundary layer is assumed to be given by

$$\nu_t' = (C_\ell h')^2 \left| \frac{\partial u'}{\partial z'} \right| \tag{35}$$

in which C_ℓ = mixing length parameter. The turbulence measurements inside the surf zone by Cox *et al.* (1994) have indicated that (35) is a reasonable first approximation outside the wave boundary layer and that C_ℓ is on the order of 0.1. Using (4)–(6) and (11), the normalized turbulent eddy viscosity ν_t corresponding to (35) is expressed as

$$\nu_t = C_\ell^2 \sigma\, h^2 \left| \frac{\partial u}{\partial z} \right| \tag{36}$$

Substitution of (33) with (34) and (36) into (21), (25) and (30) yields

$$m = C_2 h \tilde{u}_b^2 \quad ; \quad C_2 = \int_0^1 F^2 d\zeta \tag{37}$$

$$m_3 = C_3 h \tilde{u}_b^3 \quad ; \quad C_3 = \int_0^1 F^3 d\zeta \tag{38}$$

$$D_B = C_B C_\ell^2 \sigma \left| \tilde{u}_b \right|^3 \quad ; \quad C_B = \int_0^1 \left| \frac{dF}{d\zeta} \right|^3 d\zeta \tag{39}$$

in which m and D_B are positive or zero.

Madsen and Svendsen (1983) and Svendsen and Madsen (1984) assumed a cubic velocity profile for their analyses of a hydraulic jump and a turbulent bore on a beach. Accordingly, the function F in (33) outside the wave boundary layer is assumed to be cubic and expressed as

$$F = 1 - (3 + 0.75a)\zeta^2 + a\zeta^3 \quad \text{for } 0 \le \zeta \le 1 \tag{40}$$

in which a = cubic velocity profile parameter. The function F given by (40) satisfies (19) with $q = Uh$ and (32). The shear stress τ given by (9) with (36) must satisfy (15). However, (40) yields $\tau = 0$ at $\zeta = 1$ only if $a = 4$. Moreover, (40) results in $\tau = 0$ at $\zeta = 0$ immediately outside the wave boundary layer in contradiction with the turbulence measurements inside the surf zone by Cox *et al.* (1994). Consequently, (40) with the single empirical parameter a may not predict the shear stress accurately in the vicinity of the free surface and bottom. Comparison of (40) with the cubic profile assumed by Svendsen and Madsen (1984) suggests that the parameter a is approximately 3. The range of a = 3–4 is considered in the following. Substitution of (40) into the equations for C_2, C_3 and C_B in (37)–(39) yields

$$C_2 = 1 + \frac{2b}{3} + \frac{a}{2} + \frac{b^2}{5} + \frac{ab}{3} + \frac{a^2}{7} \tag{41}$$

$$C_3 = 1 + b + \frac{3a}{4} + \frac{3b^2}{5} + ab + \frac{3a^2 + b^3}{7} + \frac{3ab^2}{8} + \frac{a^2 b}{3} + \frac{a^3}{10} \tag{42}$$

$$C_B = -\left(2b^3 + \frac{36ab^2}{5} + 9a^2 b + \frac{27a^3}{7} \right) \tag{43}$$

in which $b = -(3 + 0.75a)$.

Fig. 2 shows the cubic velocity profile function F given by (40) as a function of ζ for a = 3.0, 3.5 and 4.0. The abscissa in Fig. 2 is the value of $-F$ because $\tilde{u}_b$ in (33) is expected to be negative under the wave crest. Fig. 2 hence depicts the normalizes vertical variation of the horizontal velocity deviation $(u - U)$ under the wave crest. The assumed cubic profile is not sensitive to the parameter a in the range of a = 3–4 except in the vicinity of the free surface where no velocity data is available inside the surf zone. Fig. 3 shows the parameters C_2, C_3 and C_B as a function of the cubic profile parameter a. These parameters vary little for a = 3–4. Fig. 3 indicates that $C_2 \simeq 0.5$, $C_3 \simeq -0.03$ and $C_B \simeq 13$. In short, Figs. 2 and 3 imply that the computed results will not be sensitive to the empirical parameter a. The mixing length parameter C_ℓ affects only D_B given by (39) but will modify the computed magnitude of D_B more than the cubic profile parameter a because D_B is proportional to C_ℓ^2.

Eqs. (18), (20) and (31) together with (28), (32) and (37)–(39) will be solved numerically in the next section to compute h, q and m as a function of t and x. To obtain $\tilde{u}_b$ using (37) for the computed h and m, it is assumed that

$$\tilde{u}_b = -\left(\frac{m}{C_2 h}\right)^{1/2} \qquad \text{for } U \geq 0 \tag{44a}$$

$$\tilde{u}_b = \left(\frac{m}{C_2 h}\right)^{1/2} \qquad \text{for } U < 0 \tag{44b}$$

which ensures that $|u_b| \leq |U|$ with $u_b = (U + \tilde{u}_b)$. It is required in (44) that $m \geq 0$. For the computed h, $U = q/h$ and $\tilde{u}_b$, the horizontal velocity u can be obtained using (33). The vertical velocity w can be found using the continuity equation (7).

$$w = -(z - z_b)\frac{\partial U}{\partial x} - h\frac{\partial \tilde{u}_b}{\partial x}\left[\zeta - \left(1 + \frac{a}{4}\right)\zeta^3 + \frac{a}{4}\zeta^4\right] \tag{45}$$

in which ζ is given by (34) and use is made of $w = 0$ at $z = z_b$.

To examine the degree of numerical dissipation hidden in the computed results, the instantaneous energy equation (22) with (29) is averaged from $t = t_{\text{stat}}$ to $t = t_{\text{max}}$

$$\Delta E + \frac{\partial}{\partial x}\left(\overline{E_F}\right) = -\overline{D_f} - \overline{D_B} \tag{46}$$

with

$$\Delta E = \frac{E(t = t_{\text{max}}) - E(t = t_{\text{stat}})}{t_{\text{max}} - t_{\text{stat}}} \tag{47}$$

in which the overbar denotes the time averaging from the starting time t_{stat} of the statistical calculations to the ending time t_{max} of the computation as will be explained in Section 3.4. For the computed h, q and m, E, E_F, D_f and D_B are computed using (23), (24), (27) and (39), respectively, during $t_{\text{stat}} \leq t \leq t_{\text{max}}$. The computed E, E_F, D_f and D_B will satisfy the time-averaged energy equation (46) in the absence of numerical dissipation in the adopted numerical procedures. In the previous one-dimensional models, $\overline{D_B}$ was calculated using (46) because these models did not include any physical dissipation mechanism associated with wave breaking.

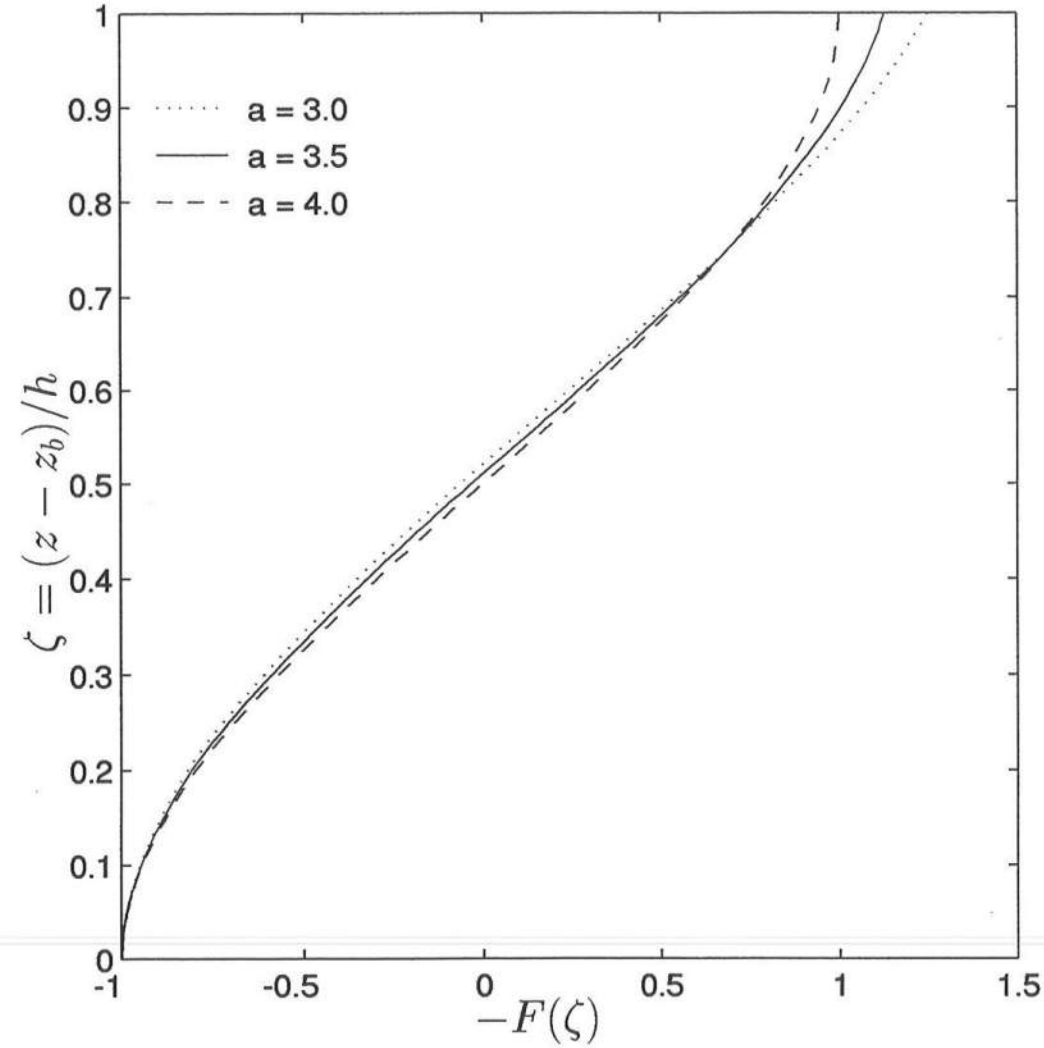

Figure 2: Cubic Velocity Profile Function $-F$ as a Function of ζ with $\zeta = 0$ at Bottom and $\zeta = 1$ at Free Surface for $a = 3.0$, 3.5 and 4.0.

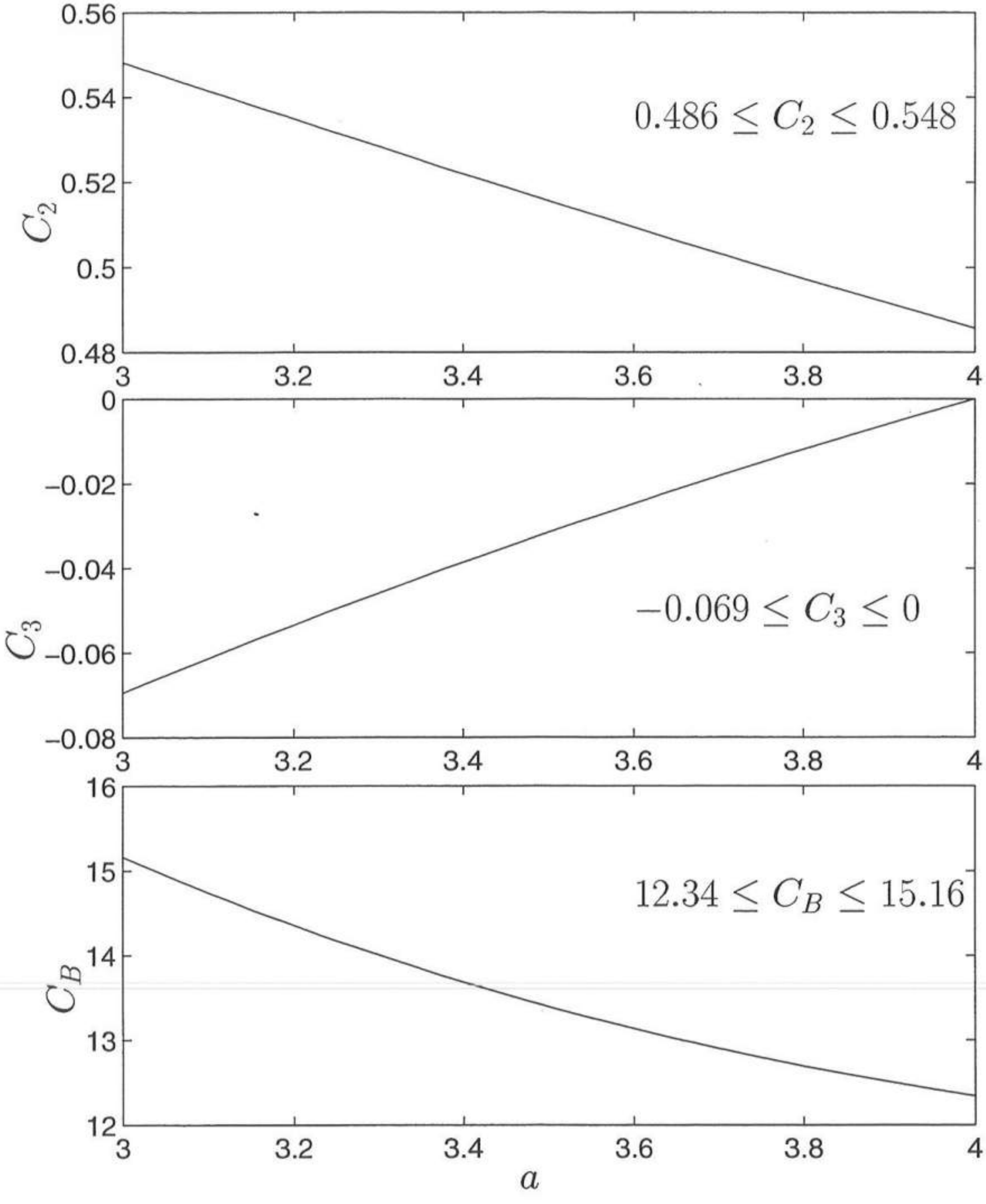

Figure 3: Parameters C_2, C_3 and C_B as a Function of Cubic Profile Parameter a.

PART III
NUMERICAL MODEL

• 3.1 •
MACCORMACK METHOD

To solve (18), (20) and (31) for h, q and m, these equations are combined and expressed in the following vector form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \mathbf{G} = 0 \tag{48}$$

with

$$\mathbf{U} = \begin{bmatrix} h \\ q \\ m \end{bmatrix} \quad ; \quad \mathbf{F} = \begin{bmatrix} q \\ F_2 \\ F_3 \end{bmatrix} \quad ; \quad \mathbf{G} = \begin{bmatrix} 0 \\ G_2 \\ G_3 \end{bmatrix} \tag{49}$$

and

$$F_2 = qU + m + \frac{1}{2}h^2 \quad ; \quad G_2 = \theta h + \tau_b \tag{50}$$

$$F_3 = 3mU + m_3 \quad ; \quad G_3 = 2\left(\tau_b \tilde{u}_b + D_B - U\frac{\partial m}{\partial x}\right) \tag{51}$$

in which $U = q/h$. τ_b is given by (28) with $u_b = (U + \tilde{u}_b)$ and $\tilde{u}_b$ is calculated using (44). m_3 and D_B are obtained from (38) and (39), respectively. Eq. (48) is solved numerically using the MacCormack method (MacCormack 1969) which is a simplified variation of the two-step Lax-Wendroff method (e.g., Anderson et al. 1984) and has been applied successfully for the computation of unsteady open channel flows with hydraulic jumps (e.g., Fennema and Chaudhry 1986; Gharangik and Chaudhry 1991).

The initial time $t = 0$ for the computation marching forward in time is taken to be the time when the incident wave arrives at the seaward boundary located at $x = 0$ and there is no wave action in the computation domain $x \geq 0$. The initial conditions for the computation are thus given by

$$h = -z_b \quad \text{at } t = 0 \quad \text{for } z_b < 0 \quad \text{(below SWL)} \tag{52a}$$

$$h = 0 \quad \text{at } t = 0 \quad \text{for } z_b \geq 0 \quad \text{(above SWL)} \tag{52b}$$

$$q = 0 \quad ; \quad m = 0 \quad \text{at } t = 0 \tag{53}$$

in which z_b = normalized bottom elevation taken to be negative below SWL.

A finite difference grid of constant nodal interval Δx and variable time step Δt is used in the numerical model VBREAK . The spacial nodes are located at $x = (j-1)\Delta x$ with $j = 1, 2, \ldots, j_{\max}$ where $j_{\max}$ = number of the spacial nodes in the computation domain. The computational shoreline is defined as the location where the normalized instantaneous water depth h equals a small value δ such as $\delta = 10^{-3}$ as in the previous one-dimensional models (e.g., Kobayashi and Poff 1994). The integer s is used to indicate the wet node next to the moving shoreline such that $h_{s+1} \leq \delta < h_s$ where h_s and h_{s+1} are the values of h_j at the node $j = s$ and $(s+1)$, respectively. It is noted that no wave overtopping and transmission are allowed in the present form of VBREAK unlike the previous one-dimensional models. It is hence required that $s < j_{\max}$.

The values of $\mathbf{U}_j$ at the node j with $j = 1, 2, \ldots, s$ and at the present time t are known in the following where $\mathbf{U}_j = \mathbf{0}$ with $j = (s+1), (s+2), \ldots, j_{\max}$ landward of the shoreline node s. The unknown values of $\mathbf{U}_j^*$ at the node j and at the next time level $t^* = (t + \Delta t)$ are denoted by the superscript asterisk. The predictor, corrector and final steps of the MacCormack method are expressed as

$$\dot{\mathbf{U}}_j = \mathbf{U}_j - \frac{\Delta t}{\Delta x}(\mathbf{F}_{j+1} - \mathbf{F}_j) - \Delta t \mathbf{G}_j \qquad \text{for } j = 1, 2, \ldots, s \tag{54}$$

$$\ddot{\mathbf{U}}_j = \dot{\mathbf{U}}_j - \frac{\Delta t}{\Delta x}\left(\dot{\mathbf{F}}_j - \dot{\mathbf{F}}_{j-1}\right) - \Delta t \dot{\mathbf{G}}_j \qquad \text{for } j = 2, 3, \ldots, s \tag{55}$$

$$\mathbf{U}_j^* = \frac{1}{2}\left(\mathbf{U}_j + \ddot{\mathbf{U}}_j\right) \qquad \text{for } j = 2, 3, \ldots, s \tag{56}$$

in which a forward spacial difference is used for the second term on the right hand side of the predictor equation (54), and a backward spacial difference is used for this term in the corrector equation (55). This is because the spacial difference in the predictor equation is recommended to be in the direction of propagation of wave fronts (Anderson et al. 1984). Accordingly, the term $\partial m/\partial x$ in the function G_3 defined in (51) is expressed by the forward and backward spacial differences in (54) and (55), respectively. The values of $\dot{\mathbf{U}}_j$ computed by (54) and the corresponding values of $\dot{\mathbf{F}}_j$ and $\dot{\mathbf{G}}_j$ in (55) are temporary values at the next time level t^*. In the computer program VBREAK , the three equations corresponding to each of (54), (55) and (56) are used for clarity. The values of $\mathbf{U}_1^*$ are computed using the seaward boundary conditions in Section 3.5. The numerical procedures for the moving shoreline in Section 3.6 are used to improve the computed values of $\mathbf{U}_s^*$ and find the shoreline node s^* at the next time level t^*.

• 3.2 •
NUMERICAL STABILITY AND SMOOTHING

The constant nodal interval Δx needs to be small enough to resolve steep wave fronts in the surf zone. The variable time step size Δt for numerically stable computation is estimated at the beginning of each time step using the following approximate equation:

$$\Delta t = \frac{C_n \Delta x}{\max\left(|U_j| + \sqrt{h_j}\right)} \qquad \text{for } j = 1, 2, \ldots, s \tag{57}$$

in which C_n is the Courant number and the denominator in (57) is the maximum value of $(|U_j| + \sqrt{h_j})$ at all the wet nodes and at the present time t. The value of Δt for each time step is selected using (57) except for the last time step that is chosen such that $t^* = (t + \Delta t) = t_{\max}$ for given t and $t_{\max}$. The numerical stability of the MacCormack method applied to (48) with $m = 0$ and $G_2 = 0$ requires that $C_n \leq 1$ (e.g., Anderson et al. 1984). Eq. (57) is approximate because the characteristic equations corresponding to (48) with $m \neq 0$ can not be expressed in simple analytical forms. Moreover, (57) does not account for the shorline algorithm which tends to suffer numerical difficulties. Consequently, the value of C_n is specified as input to adjust Δt for the successful computation of each case. This manual adjustment of Δt appears to be sufficient because the computation time of VBREAK is relatively short as summarized in Appendix B.

Use of the MacCormack method results in numerical high-frequency oscillations which tend to appear at the rear of a breaking wave, especially on a gentle slope. For open-channel flows, Chaudhry (1993) summarized a procedure presented by Jameson et al. (1981) to smooth these high-frequency oscillations without disturbing the rest of the computed variations. To apply this procedure for breaking waves on slopes, the computed water depth h_j^* at the node j and at the next time level t^* is used to calculate the parameter ν_j at the node j defined as

$$\nu_j = \frac{|h_{j+1}^* - 2h_j^* + h_{j-1}^*|}{|h_{j+1}^*| + 2|h_j^*| + |h_{j-1}^*|} \qquad \text{for } j = 2,\ 3,\ \ldots,\ (s^* - 1) \tag{58}$$

The parameter $\epsilon_{j+0.5}$ at the midpoint of the nodes j and $(j+1)$ is given by

$$\epsilon_{j+0.5} = \kappa \left(\frac{h_j^* + h_{j+1}^*}{2}\right)^{0.5} \max(\nu_j,\ \nu_{j+1}) \qquad \text{for } j = 2,\ 3,\ \ldots,\ (s^* - 2) \tag{59}$$

in which κ = numerical damping coefficient for regulating the amount of damping the high-frequency oscillations. The computed water depth h_j^* is modified as

$$h_j^* = h_j^* + \epsilon_{j+0.5}\left(h_{j+1}^* - h_j^*\right) - \epsilon_{j-0.5}\left(h_j^* - h_{j-1}^*\right) \quad \text{for } j = 3,\ 4,\ \ldots,\ (s^* - 2) \tag{60}$$

which should be considered as a FORTRAN replacement statement. Likewise, U_j^* and m_j^* are smoothed using (60) with h_j^* being replaced by U_j^* and m_j^*, respectively, where $\epsilon_{j+0.5}$ is the same. The smoothed h_j^* and U_j^* are used to calculate $q_j^* = h_j^* U_j^*$.

Chaudhry (1993) suggested the expressions of ν_j at the end points $j = 1$ and s^* in (58) for open-channel flows. Addition of these expressions in (59) and (60) is found to produce spurious fluid motions even in the absence of waves on slopes. As a result, the smoothing at the end points is not recommended for breaking waves on slopes.

The numerical damping coefficient κ is specified as input to VBREAK . For breaking waves on gentle beach slopes, the value of κ on the order of unity is found to be necessary to damp the high-frequency oscillations adequately. For waves surging on steep slopes of coastal structures, the value of κ on the order of 0.1 appears to be sufficient. However, the smoothing procedure based on (58) tends to cause more damping near the shoreline where the water depth h is very small. To remedy this uneven damping, the term $[(h_j^* + h_{j+1}^*)/2]^{0.5}$ is added in (59) to reduce the damping near the shoreline.

• 3.3 •

Incident or Measured Wave Profile

The required wave input for the numerical model VBREAK is the normalized incident or measured wave profile at the seaward boundary of the computation domain, that is, $\eta_i(t) = \eta_i'(t')/H'$ or $\eta(t) = \eta'(t')/H'$ at $x = 0$ with $t = t'/T'$ where H' and T' are the reference wave height and period used for the normalization in (4)–(6). The specification of $\eta_i(t)$ or $\eta(t)$ at $x = 0$ for $t \geq 0$ needs to satisfy the condition $\eta_i = 0$ or $\eta = 0$ at $x = 0$ at the initial time $t = 0$ to be consistent with the assumed initial conditions of no wave action in the region of $x \geq 0$ at $t = 0$.

The temporal variation of the incident wave profile $\eta_i(t)$ at $x = 0$ can be

1. the incident monochromatic wave profile computed (by the computer program VBREAK) using an appropriate wave theory, or

2. a user-specified incident irregular or transient wave train including the incident wave profile measured in the absence of wave reflection, measured in the presence of wave reflection but separated from the reflected wave using linear wave theory, or generated numerically for given frequency spectrum.

For convenience, the former will be referred to as the case of *regular* waves, while the latter is simply called the case of *irregular* waves, although this case can also include monochromatic transient waves.

The temporal variation of the normalized free surface elevation $\eta(t)$ at $x = 0$ can be specified as input if $\eta(t)$ at $x = 0$ is measured in the presence of wave reflection. This option eliminates uncertainties associated with the separation of incident and reflected waves using linear wave theory in laboratory and field measurements.

3.3.1 Incident Regular Wave (IWAVE=1)

For the case of incident regular waves identified by the integer IWAVE=1 in VBREAK , the periodic variation of $\eta_i(t)$ is computed by the computer program using either cnoidal or Stokes second-order wave theory. The height and period of the incident regular waves at the seaward boundary located at $x = 0$ are denoted by H_i' and T_i'. The reference wave period T' is taken as $T' = T_i'$ for the incident regular waves. The reference wave height H' specified as input may be in deeper water depth. Since the numerical model is based on the assumption of shallow water waves, the seaward boundary should be located in relatively shallow water. As a result, it is not always possible to take $H' = H_i'$. Defining $K_s = H_i'/H'$, the height and period of the regular wave profile $\eta_i(t)$ at $x = 0$ is K_s and unity, respectively.

For Stokes second-order wave theory, the incident wave profile $\eta_i(t)$ at $x = 0$ is given by (e.g., Kobayashi and Poff 1994)

$$\eta_i(t) = K_s \left\{ \frac{1}{2} cos\left[2\pi(t + t_0)\right] + a_2 cos\left[4\pi(t + t_0)\right] \right\} \quad \text{for } t \geq 0 \tag{61}$$

with

$$a_2 = \frac{2\pi}{L} cosh\left(\frac{2\pi}{L}\right)\left[2 + cosh\left(\frac{4\pi}{L}\right)\right]\left[16\frac{d_t}{K_s} sinh^3\left(\frac{2\pi}{L}\right)\right]^{-1} \tag{62}$$

$$L = L_0 \, tanh\frac{2\pi}{L} \tag{63}$$

where t_0 = time shift computed to satisfy the conditions that $\eta_i = 0$ at $t = 0$ and η_i decreases initially; a_2 = normalized amplitude of the second-order harmonic; $L = L'/d_t'$; $d_t = d_t'/H'$; $L_0 = L_0'/d_t'$; d_t' = water depth below SWL at $x = 0$; L' = dimensional wavelength at $x = 0$; and L_0' = dimensional linear wavelength in deep water. The normalized wavelength L satisfying (63) for given L_0 is computed using a Newton-Raphson iteration method. Eq. (62) yields the value of a_2 for given $d_t = d_t'/H'$, K_s, and L. Since (61) satisfies $\eta_i(t+1) = \eta_i(t)$ and $\eta_i(-t-t_0) = \eta_i(t+t_0)$, it is sufficient to compute the profile $\eta_i(t)$ for $0 \leq (t+t_0) \leq 0.5$ to obtain $\eta_i(t)$ for $0 \leq t \leq t_{\max}$ where $t_{\max}$ = specified computation duration. Eq. (61) may be appropriate if the Ursell parameter $U_r < 26$ where $U_r = [H_i'(L')^2/(d_t')^3] = (K_s L^2/d_t)$ at $x = 0$. It is noted that the value of U_r based on the normalized wavelength L computed from (63) is simply used to decide whether cnoidal or Stokes second-order wave theory is applied.

For the case of $U_r \geq 26$, cnoidal wave theory is used to compute the incident wave profile $\eta_i(t)$ at $x = 0$ (e.g., Kobayashi and Poff 1994)

$$\eta_i(t) = \eta_{min} + K_s \, cn^2 \left[2K(t+t_0)\right] \quad \text{for } t \geq 0 \tag{64}$$

with

$$\eta_{min} = \frac{K_s}{m}\left(1 - \frac{E}{K}\right) - K_s \tag{65}$$

where $\eta_{\min}$ = normalized trough elevation below SWL; cn = Jacobian elliptic function; K = complete elliptic integral of the first kind; E = complete elliptic integral of the second kind, which should be differentiated from the specific wave energy E given by (23); and m = parameter determining the complete elliptic integrals $K(m)$ and $E(m)$. It is noted that this parameter m is different from the momentum flux correction m used in the previous sections. The notation of m for cnoidal wave theory is standard and used here. The parameter m for cnoidal wave theory is related to the Ursell parameter U_r

$$U_r = \frac{K_s L^2}{d_t} = \frac{16}{3} m K^2 \tag{66}$$

For $U_r \geq 26$, the parameter m is in the range $0.8 < m < 1$. The parameter m for given σ, d_t, and K_s, where σ is defined in (6), is computed from

$$\frac{\sigma}{L\sqrt{d_t}} \sqrt{\left[1 + \frac{K_s}{m \, d_t}\left(-m + 2 - 3\frac{E}{K}\right)\right]} - 1 = 0 \tag{67}$$

where the normalized wavelength L is given by (66) as a function of m for given d_t and K_s. The left hand side of (67) is a reasonably simple function of m in the range $0.8 < m < 1$. As a result, (67) can be solved using an iteration method which successively narrows down the range of m bracketing the root of (67). After the value of m is computed for given σ, d_t, and K_s, the values of U_r and L are computed using (66), while (65) yields the value of η_{min}. The incident wave profile $\eta_i(t)$ is computed using (64) for $0 \leq (t+t_0) \leq 0.5$ where the time shift t_0

and the periodicity and symmetry of the cnoidal wave profile are used in the same manner as the Stokes second-order wave profile given by (61). It should be mentioned that the Jacobian elliptic function and the complete elliptic integrals of the first and second kinds are computed using the subroutines given by Press *et al.* (1986).

3.3.2 Incident Irregular Wave (IWAVE=2)

The specification of incident irregular waves as input to VBREAK is identified by IWAVE=2 and is the same as in the previous one-dimensional model RBREAK2 (Kobayashi and Poff 1994). Various examples of user-specified irregular wave trains were given by Wurjanto and Kobayashi (1991).

The incident wave train $\eta_i(t)$ normalized by the reference wave height H' is read at the following sampling rate

$$\delta t_i = \frac{t_{\max}}{\text{NPINP}-1} \tag{68}$$

in which NPINP = specified number of points in the input wave train for $0 \leq t \leq t_{\max}$ with t and $t_{\max}$ being normalized by the reference wave period T'. The reference wave height and period can be chosen as any height and period that are convenient for the analysis of computed normalized results. The normalized wave height K_s specified as input for IWAVE=1 is not required for IWAVE=2. It is noted that the normalized incident regular wave train given by (61) and (64) is also computed at the rate δt_i given by (68).

The sampling rate δt_i must be small enough to resolve the temporal variation of $\eta_i(t)$ but is normally much larger than the finite difference time step Δt calculated by (57) for the numerically stable computation. A simple linear interpolation of $\eta_i(t)$ sampled at the rate δt_i is performed to find the value of $\eta_i(t^*)$ at the time level $t^* = (t + \Delta t)$ during the time-marching computation.

3.3.3 Measured Wave Profile (IWAVE=3)

If the free surface oscillation is measured at the seaward boundary of the computation, it is more direct and straightforward to specify the measured free surface oscillation as input to VBREAK. This option identified by IWAVE=3 eliminates the uncertainty associated with the separation of incident and reflected waves using linear wave theory, which is required for the option of IWAVE=2.

The measured free surface elevation above SWL at the seaward boundary is normalized by the values of H' and T' specified as input. The normalized input time series of $\eta(t)$ at $x = 0$ for IWAVE=3 is read at the sampling rate δt_i given by (68) and interpolated linearly in the same way as $\eta_i(t)$ at $x = 0$ for IWAVE=2.

• 3.4 •
Statistical Calculations

The *statistical calculations* in this report imply the calculations of the mean, root-mean-square (rms), maximum and minimum values of computed time-series for the duration of $t_{\text{stat}} \le t \le t_{\max}$. For example, the rms value of η is defined as

$$\eta_{\text{rms}} = \left[\ \overline{(\eta - \bar{\eta})^2} \ \right]^{1/2} = \left[\ \overline{\eta^2} - (\bar{\eta})^2 \right]^{1/2} \tag{69}$$

in which the overbar denotes the time averaging for $t_{\text{stat}} \le t \le t_{\max}$.

For the case of *regular* waves, the statistical calculations should be executed over the last wave period, assuming that the computation duration $t_{\max}$ is large enough to reach the periodicity of time-varying quantities during $t_{\text{stat}} \le t \le t_{\max}$ with $t_{\text{stat}} = (t_{\max} - 1)$. This duration ranges from approximately five wave periods for coastal structures (Kobayashi and Wurjanto 1989) to about 30 wave periods for beaches (Kobayashi et al. 1989).

For *irregular* wave computations, the statistical calculations are conducted over most or all of the computation duration. The initial transient waves may be excluded by specifying an appropriate value of t_{stat} estimated from the corresponding regular wave computation.

• 3.5 •
WAVE REFLECTION

The normalized reflected wave train $\eta_r(t)$ at the seaward boundary needs to be computed to estimate the degree of wave reflection from the computation domain. It is also required to find the unknown value of the vector $\mathbf{U}_1^*$ at $x = 0$ and at the next time level $t^* = (t + \Delta t)$ which can not be computed using (56). The seaward boundary algorithm needs to be developed for the cases of IWAVE=1 and 2 where the incident wave profile $\eta_i(t)$ at $x = 0$ is specified as input and for IWAVE=3 where the total free surface profile $\eta(t)$ at $x = 0$ is specified as input.

In order to derive approximate seaward boundary conditions for h and q, (18) and (20) are expressed in the following characteristic forms:

$$\frac{d\alpha}{dt} = \frac{\partial \alpha}{\partial t} + (U + c)\frac{\partial \alpha}{\partial x} = -\theta - \frac{\tau_b}{h} - \frac{1}{h}\frac{\partial m}{\partial x} \quad \text{along} \quad \frac{dx}{dt} = U + c \tag{70}$$

$$\frac{d\beta}{dt} = \frac{\partial \beta}{\partial t} + (U - c)\frac{\partial \beta}{\partial x} = \theta + \frac{\tau_b}{h} + \frac{1}{h}\frac{\partial m}{\partial x} \quad \text{along} \quad \frac{dx}{dt} = U - c \tag{71}$$

with

$$c = \sqrt{h} \quad ; \quad \alpha = U + 2c \quad ; \quad \beta = -U + 2c \tag{72}$$

in which c is the normalized phase velocity, whereas α and β are the characteristic variables.

Assuming that $U < c$ and the flow is subcritical in the vicinity of the seaward boundary where the normalized water depth below SWL is d_t, α and β represent the characteristics advancing landward and seaward, respectively, in the vicinity of the seaward boundary. The total water depth at the seaward boundary is expressed in the form (Kobayashi et al. 1987).

$$h(t) = d_t + \eta(t) \quad \text{at } x = 0 \tag{73}$$

with

$$\eta(t) = \eta_i(t) + \eta_r(t) \quad \text{at } x = 0 \tag{74}$$

where η_i and η_r are the free surface elevations normalized by H' at $x = 0$ due to the incident and reflected waves, respectively. The incident wave train may be specified by prescribing the variation of η_i with respect to $t \geq 0$. Alternatively, the free surface elevation measured at $x = 0$ may be specified as input by prescribing the variation of η with respect to $t \geq 0$. The normalized reflected wave train η_r is approximately expressed in terms of the seaward advancing characteristic β at $x = 0$

$$\eta_r(t) \simeq \frac{1}{2}\sqrt{d_t}\beta(t) - d_t - C_t \quad \text{at} \quad x = 0 \tag{75}$$

where β is obtained using (71) and linear long wave theory is used to derive (75). For $h(t)$ at $x = 0$ calculated using (73), $U = (2\sqrt{h} - \beta)$ using (72) and $q = hU$.

The correction term C_t in (75) introduced by Kobayashi et al. (1989) to predict wave set-down and setup on a beach may be expressed as

$$C_t = \frac{1}{2}\sqrt{d_t}\,\frac{\overline{(\eta - \bar{\eta})(U - \bar{U})}}{\bar{h}} \qquad \text{at} \quad x = 0 \tag{76}$$

For incident regular waves on gentle slopes, C_t may be estimated by (Kobayashi et al. 1989)

$$C_t = \frac{K_s^2}{16 d_t} \qquad \text{for gentle slopes} \tag{77}$$

where the assumptions of linear long wave and negligible wave reflection were made in (76) to derive (77). For coastal structures, wave reflection may not be negligible but the location of the seaward boundary may be chosen such that $C_t \simeq 0$ on the basis of (77). For incident irregular waves, (77) may still be used as a first approximation to improve the prediction of wave set-down and setup on a beach. For IWAVE=3 the measured time series of $\eta(t)$ at $x = 0$ specified as input includes the wave set-down or setup at $x = 0$. Consequently, the reflected wave train $\eta_r(t)$ is computed using (75) with $C_t = 0$.

On the other hand, the value of m at the seaward boundary needs to be found using (31). The initial condition for m is specified as $m = 0$ at $t = 0$ in the computation domain $x \geq 0$. The value of m at $x = 0$ might be taken as $m = 0$ at $x = 0$ if the seaward boundary is located outside the surf zone. This is because the vertical variation of the horizontal velocity assumed in (33) is caused by wave breaking in this numerical model for shallow water waves. However, the boundary condition of $m = 0$ at $x = 0$ will yield $m = 0$ for $t > 0$ and $x > 0$ because $m = 0$ is a trivial solution of (31). It is hence required to introduce $m \geq 0$ at $x = 0$ so that $m \geq 0$ for $t > 0$ and $x > 0$. One option is to rewrite (31) in terms of $\tilde{u}_b$ using (37), (38) and (39)

$$\frac{\partial \tilde{u}_b}{\partial t} + \frac{\partial}{\partial x}(U\tilde{u}_b) = -\frac{C_3\tilde{u}_b}{2C_2}\left(\frac{\tilde{u}_b}{h}\frac{\partial h}{\partial x} + 3\frac{\partial \tilde{u}_b}{\partial x}\right) - \frac{\tau_b + C_{B\ell}|\tilde{u}_b|\tilde{u}_b}{C_2 h} \tag{78}$$

with

$$C_{B\ell} = C_B C_\ell^2 \sigma \tag{79}$$

in which τ_b is given by (28) with $u_b = (U + \tilde{u}_b)$ and $\tilde{u}_b = 0$ is not a trivial solution of (78). The value of $m = C_2 h \tilde{u}_b^2$ at $x = 0$ may be obtained using the value of $\tilde{u}_b$ at $x = 0$ computed using (78) as explained in the following.

3.5.1 Seaward Boundary Algorithm for IWAVE=1 and 2

An explicit first-order finite difference equation corresponding to (71) is used to find the value of β_1^* at $x = 0$ and the next time t^* for the cases of IWAVE=1 and 2

$$\beta_1^* = \beta_1 - \frac{\Delta t}{\Delta x}(U_1 - c_1)(\beta_2 - \beta_1) + \Delta t \left[\theta_1 + \frac{(\tau_b)_1}{h_1}\right] + \frac{\Delta t}{\Delta x}\,\frac{m_2 - m_1}{h_1} \tag{80}$$

where $\beta_1 = (-U_1 + 2c_1)$ and $\beta_2 = (-U_2 + 2c_2)$. The right hand side of (80) can be computed for the known values of $\mathbf{U}_j$ with $j = 1$ and 2 at the present time t where the spatial nodes are located at $x = (j-1)\Delta x$. The value of η_r^* at the time t^* is calculated using (75). The incident wave profile $\eta_i(t)$ specified as input together with (73) and (74) yields the value of h_1^*, while $U_1^* = [2\sqrt{(h_1^*)} - \beta_1^*]$ using the definition of β given in (72). Thus, the values of h_1^*, U_1^*, and $q_1^* = U_1^* h_1^*$ at $x = 0$ and the time t^* are obtained.

As for the value of m_1^* at $x = 0$ and the next time t^*, an explicit first-order finite difference approximation of (78) is used to obtain the value of $(\tilde{u}_b)_1^*$ as follows:

$$(\tilde{u}_b)_1^* = (\tilde{u}_b)_1 - \frac{\Delta t}{\Delta x}\left[U_2(\tilde{u}_b)_2 - U_1(\tilde{u}_b)_1\right] - \frac{\Delta t}{C_2}\left\{\frac{C_3(\tilde{u}_b)_1}{2\Delta x}\left[(\tilde{u}_b)_1\left(\frac{h_2}{h_1} - 4\right)\right.\right.$$
$$\left.\left. + 3(\tilde{u}_b)_2\right] + h_1^{-1}\left[(\tau_b)_1 + C_{B\ell} \mid (\tilde{u}_b)_1 \mid (\tilde{u}_b)_1\right]\right\} \tag{81}$$

The value of m_1^* is then calculated using (37)

$$m_1^* = C_2 h_1^* \left[(\tilde{u}_b)_1^*\right]^2 \tag{82}$$

3.5.2 Seaward Boundary Algorithm for IWAVE=3

For IWAVE=3 where $\eta(t)$ at $x = 0$ in (73) is specified as input, the values of η_1^* and h_1^* at $x = 0$ and the time t^* are known. The value of U_1^* at $x = 0$ and the time t^* is computed using (71) for the characteristic variable β advancing seaward from the computation domain.

A simple first-order finite difference approximation of (71) along the straight line, $dx/dt = (U_1^* - c_1^*) < 0$, originating from the point at node 1 and the time $t^* = (t + \Delta t)$ may be expressed as

$$\beta_1^* = \beta_{12} + \Delta t \left[\theta_1 + \frac{(\tau_b)_1}{h_1}\right] + \frac{\Delta t}{\Delta x}\,\frac{m_2 - m_1}{h_1} \tag{83}$$

where β_{12} is the value of β at the time t and at the location of $x = \delta x$ given by

$$\delta x = -(U_1^* - c_1^*)\,\Delta t > 0 \tag{84}$$

The numerical stability criterion given by (57) requires that $\delta x < \Delta x$. As a result, the point of $x = \delta x$ is located between nodes 1 and 2. The linear interpolation between the known values of β_1 and β_2 at the time t yields

$$\beta_{12} = \beta_1 + \frac{\delta x}{\Delta x}(\beta_2 - \beta_1) \tag{85}$$

Using (84) and (85), (83) may be rewritten as

$$\beta_1^* = \beta_1 - \frac{\Delta t}{\Delta x}(U_1^* - c_1^*)(\beta_2 - \beta_1) + \Delta t \left[\theta_1 + \frac{(\tau_b)_1}{h_1}\right] + \frac{\Delta t}{\Delta x}\,\frac{m_2 - m_1}{h_1} \tag{86}$$

which corresponds to (80) except that $(U_1 - c_1)$ in (80) is replaced by $(U_1^* - c_1^*)$. Eq. (86) for $\beta_1^* = (2c_1^* - U_1^*)$ is an implicit scheme for U_1^* for the known value of $c_1^* = \sqrt{(h_1^*)}$. Solving (86) for U_1^* yields

$$U_1^* = \left[1 - \frac{\Delta t}{\Delta x}(\beta_2 - \beta_1)\right]^{-1} \left\{2c_1^* - \beta_1 - \frac{\Delta t}{\Delta x}\left[c_1^*(\beta_2 - \beta_1) + \frac{m_2 - m_1}{h_1}\right] - \Delta t\left[\theta_1 + \frac{(\tau_b)_1}{h_1}\right]\right\} \tag{87}$$

If the absolute value of the denominator on the right hand side of (87) becomes almost zero, this implicit algorithm may not be appropriate. This problem has never been encountered so far partly because the numerical stability criterion expressed as (57) generally requires a value of $\Delta t/\Delta x$ that is much less than unity.

After U_1^* is computed using (87), the value of β_1^* is obtained from $\beta_1^* = (2c_1^* - U_1^*)$. The value of η_r^* for the reflected wave profile is then calculated using (75) at the time t^*. The value of η_i^* for the incident wave profile is obtained from $\eta_i^* = (\eta_1^* - \eta_r^*)$ based on (74). The value of m_1^* is computed using (82) with (81).

3.5.3 Wave Reflection Coefficient

The average reflection coefficient r for regular and irregular waves may be estimated using the root-mean-square values of the time series of η_r and η_i as defined by (69)

$$r = (\eta_r)_{\text{rms}} \,/\, (\eta_i)_{\text{rms}} \tag{88}$$

which is equal to the square root of the ratio between the time-averaged reflected wave energy as compared to the time-averaged incident wave energy on the basis of linear wave theory. The reflection coefficient as a function of the frequency for irregular waves can be calculated using the reflected and incident wave spectra computed from the time series of η_r and η_i (*e.g.*, Kobayashi et al. 1990).

• 3.6 •
Wave Runup

For the case of no wave overtopping, the landward boundary of the numerical model is located at the moving shoreline on the slope where the water depth is essentially zero. The kinematic boundary condition requires that the horizontal shoreline velocity be the same as the horizontal fluid velocity. In reality, it is difficult to pinpoint the exact location of the moving shoreline on the slope. For the computation, the shoreline is defined as the location where the normalized instantaneous water depth equals a small value δ such as $\delta = 10^{-3}$ as explained in Section 3.1.

The following numerical procedure dealing with the moving shoreline located at $h = \delta$ is used to obtain the values of $\mathbf{U}_j^*$ at the next time $t^* = (t + \Delta t)$ for the nodes $j \geq s$ where s = integer indicating the wet node next to the moving shoreline at the present time t such that $h_{s+1} \leq \delta < h_s$. It is noted that the procedure is somewhat intuitive and may be improved since the moving shoreline tends to cause numerical instability.

1. After computing $\mathbf{U}_j^*$ with $j = 2, 3, \ldots, s$ using (56), it is checked whether $h_{s-1}^* \le \delta$, which may be encountered during a downrush. This is considered a computation failure since the shoreline should not move more than Δx because of the numerical stability criterion of the adopted explicit method given by (57).

2. If $h_s^* \ge h_{s-1}^*$, use $h_s^* = (2h_{s-1}^* - h_{s-2}^*)$, and $U_s^* = (2U_{s-1}^* - U_{s-2}^*)$, so that the water depth near the shoreline decreases landward. The following adjustments are made

   - if $|U_s^*| > |U_{s-1}^*|$, set $U_s^* = 0.9U_{s-1}^*$;
   - if $h_s^* < 0$, set $h_s^* = 0.5h_{s-1}^*$;
   - and if $h_s^* > h_{s-1}^*$, set $h_s^* = 0.9h_{s-1}^*$.

   Then, obtain $q_s^* = h_s^* U_s^*$ based on the *adjusted* values of h_s^* and U_s^*.

3. If $h_s^* \le \delta$, set $s^* = (s-1)$ and go to Step 11. The integer s^* indicates the wet node next to the shoreline at the next time t^*.

4. If $h_s^* > \delta$, compute $h_{s+1}^* = (2h_s^* - h_{s-1}^*)$, $U_{s+1}^* = (2U_s^* - U_{s-1}^*)$, and $q_{s+1}^* = h_{s+1}^* U_{s+1}^*$.

5. If $h_{s+1}^* \le \delta$, set $s^* = s$ and go to Step 11.

6. If $h_{s+1}^* > \delta$, compute $\mathbf{U}_s^{**}$ at the time $t^{**} = (t^* + \Delta t)$ using (56) with $m = 0$ where $\mathbf{U}_j^*$ and $\mathbf{U}_j$ in (56) are replaced by $\mathbf{U}_s^{**}$ and $\mathbf{U}_s^*$, respectively. The vertical variation of the horizontal velocity may be assumed to be small in the vicinity of the moving shoreline. Improve the linearly extrapolated values in Step 4 using the following finite difference equations derived from (18) and (20) with $m = 0$ at $\tau_b = 0$:

$$q_{s+1}^* = q_{s-1}^* - \frac{\Delta x}{\Delta t}\left(h_s^{**} - h_s\right) \tag{89}$$

$$U_{s+1}^* = U_{s-1}^* - \frac{1}{U_s^*}\left[\frac{\Delta x}{\Delta t}\left(U_s^{**} - U_s\right) + h_{s+1}^* - h_{s-1}^* + 2\Delta x\,\theta_s\right] \tag{90}$$

   The upper limit of the absolute value of $(U_s^*)^{-1}$ in (90) is taken as δ^{-1} to avoid the division by the very small value. Calculate $h_{s+1}^* = q_{s+1}^*/U_{s+1}^*$.

7. If $|U_{s+1}^*| \le \delta$, set $s^* = s$ and go to Step 11.

8. If $h_{s+1}^* \le h_s^*$ and $h_{s+1}^* \le \delta$, set $s^* = s$ and go to Step 11.

9. If $h_{s+1}^* \le h_s^*$ and $h_{s+1}^* > \delta$, set $s^* = (s+1)$ and go to Step 11.

10. If $h_{s+1}^* > h_s^*$, the linearly extrapolated values of h_{s+1}^*, U_{s+1}^*, and q_{s+1}^* in Step 4 are adopted in the following instead of those computed in Step 6. Furthermore, set $s^* = (s+1)$ if $h_{s+1}^* \le h_s^*$ and $U_{s+1}^* \ge \delta$ and set $s^* = s$ otherwise where h_{s+1}^* and U_{s+1}^* are the adopted values.

11. After $s*$ is obtained, set $h_j^* = 0$, $U_j^* = 0$, $q_j^* = 0$ and $m_j^* = 0$ for $j \ge (s^*+1)$ since no water is present above the computational shoreline. If $s^* = (s+1)$, set $m_{s+1}^* = 0$. It is noted that the smoothing procedure given by (60) does not affect this shoreline algorithm.

Once the normalized water depth h at the given time is known as a function of x, the normalized free surface elevation, $Z_r = Z'_r/H'$, where the physical water depth equals a specified value δ'_r, can be computed as long as $\delta_r = (\delta'_r/H') > \delta$. The use of the physical depth δ'_r is related to the use of a runup wire to measure the shoreline oscillation on the slope (*e.g.*, Raubenheimer et al. 1995). The specified depth δ'_r can be regarded as the vertical distance between the runup wire and the slope, while the corresponding elevation Z'_r is the elevation above SWL of the intersection between the runup wire and the free surface. The computed oscillations of $Z_r(t)$ for different values of δ'_r can be used to examine the sensitivity to δ'_r of wave runup and run-down, which are normally defined as the maximum and minimum elevations relative to SWL reached by uprushing and downrushing water on the slope, respectively. The normalized runup R, run-down R_d, and setup $\overline{Z_r}$ as well as the root-mean-square value (standard deviation) of $Z_r(t)$ for given δ'_r are obtained from the computed oscillation of $Z_r(t)$.

PART IV
COMPUTER PROGRAM VBREAK

• 4.1 •
INTRODUCTION

The computer program VBREAK attached in Appendix A consists of the main program, 22 subroutines, and one function. Full double precision mode is used throughout the program to gain maximum numerical accuracy. The program has been tested on a Sun SPARC2 operating under UNIX-based SunOS. Written in standard FORTRAN-77, VBREAK should run on other machines.

The numerical model VBREAK is a major extension of the most recent one-dimensional model RBREAK2 to predict vertically two-dimensional breaking wave motions on impermeable slopes. The computer program VBREAK is written in a very concise manner by streamlining the various subroutines and input requirements of RBREAK2. However, several options such as wave overtopping and wave transmission as well as armor stability and movement included in RBREAK2 are not allowed in the first version of VBREAK . Furthermore, VBREAK has been compared with regular wave data only as will be explained in Section 5.1. Additional efforts will be required to expand VBREAK and make it as versatile as RBREAK2. Detailed two-dimensional data will also be needed to calibrate and verify the expanded numerical model.

• 4.2 •
INPUT DATA FILES

To execute the computer program VBREAK , the following input data files are needed:

1. *Primary input data file* containing all the variables and parameters needed to specify the case being investigated, except the input wave train, which is prescribed in the second input data file.

2. File containing the *input wave profile* for IWAVE=2 or 3 as explained in Sections 3.3.2 and 3.3.3.

These two input data files are prepared by a user. The user has the freedom for selecting the names of the input data files which are to be read by VBREAK as the variables FINP1 and FINP2, respectively. The only limitation imposed by VBREAK is that the name should consist of no more than ten characters. The operating system under which VBREAK is running may dictate a certain convention regarding file naming.

The user enters the name of the primary input data file interactively at the beginning of VBREAK computation. The name of the file containing the input wave train is specified in the primary input data file. The preparation of the primary input data file may be best explained by use of the example given in Section 5.2. In addition, the order of the contents of the primary input data file will be presented in Section 4.6.

The time increment of the input wave train, δt_i, is normally much larger than the finite difference time step Δt for the stable numerical computation. The computer program VBREAK performs a simple linear interpolation of the input wave train to get the appropriate value at each time level during the time-marching computation. The interpolation is carried out in Subr. 14 SEABC.

• 4.3 •

MAIN PROGRAM VBREAK

The main program lists all the important parameters and variables in the COMMON blocks. These parameters and variables are described in Section 4.5. The main program coordinates tasks which are actually executed by subroutines. The tasks can be categorized into five groups.

1. Reading Input Data.
2. Checking FORTRAN PARAMETERs in the Subroutines.
3. Groundwork.
4. Time-marching Computation.
5. Finishing.

4.3.1 READING INPUT DATA

The first variable read by VBREAK is MREP, which is specified interactively by the user. MREP determines the interval between two consecutive progress messages VBREAK displays on a terminal screen. The progress message is displayed every MREP wave periods where the period herein is the reference wave period. For example, if MREP=4, the following message (without the horizontal lines) will appear on the screen when the computation has just finished 56 wave periods.

```
Finished          56 Wave Periods
```

This reporting is useful in keeping track of the progress of a long computation. If not necessary, however, the reporting can be turned off by specifying MREP=0.

The next input is the name of the primary input data file, which is also entered interactively. The two input data files explained in Section 4.2 are then read from the prepared files. Input and output files are opened by calling Subr. 01 OPENER. The contents of the first and the second input data files are read by Subr. 02 INPUT1 and Subr. 03 INPUT2, respectively, at the beginning of the computation before the time-marching computation begins. A list of the READ statements corresponding to the primary input data file will be presented in Section 4.6.

4.3.2 Checking FORTRAN PARAMETERs in the Subroutines

In the computer program VBREAK, the dimension of an array is specified using an integer which is independently declared as a PARAMETER by each program unit where the term *program unit* is used herein to represent the main program, subroutines, and function. The almost all of VBREAK's variables, which are mostly arrays, are passed between the program units using COMMON blocks. Only few variables are passed as arguments of the subroutines and function. This arrangement demands that the dimensions of arrays in the COMMON blocks throughout VBREAK remain the same. Subr. 22 CHEPAR detects possible mismatches in the array dimensions among the program units using the PARAMETERs specified in the main program as reference,

4.3.3 Groundwork

The tasks performed in preparation for the time-marching computation include

- Computation of the normalized bottom geometry using Subr. 04 BOTTOM.
- Computation of the wave and velocity profile parameters using Subr. 05 PARAM.
- Assignment of the initial values using Subr. 06 INIT.
- For IWAVE=1 the incident periodic wave profile explained in Section 3.3.1 is computed using Subr. 07 INCREG.

4.3.4 Time-Marching Computation

The time-marching computation is executed for $0 \leq t \leq t_{\max}$. One wave period in the time-marching computation is unity on the basis of the normalization by the reference wave period. The integer MOWAVE is used to indicate the number of wave periods completed during the time-marching computation.

The *present* time t is denoted by TIME, whereas the *next* time $t^* = (t + \Delta t)$ is indicated by TIMEST with the additional letters ST instead of the superscript asterisk.

During the time-marching computation, the unknown quantities at the time t^* = TIMEST are computed from the known quantities at the time t = TIME.

At the beginning of each time step, the following is performed:

- Compute the time step size Δt denoted by DT using (57) (Subr. 11 COMPDT).
- If $t \leq t_{\text{stat}} < t^*$ and IENERG=1, store the wave energy quantities at the time t for the subsequent interpolations to calculate the values of these quantities at the time t_{stat} for their statistical calculations (Subr. 18 ENERGY).

Time-marching from one time level to the next is done as follows:

- Compute the unknown hydrodynamic quantities at the time $t^* =$ `TIMEST` using Sections 3.1, 3.5 and 3.6 (Subr. 12 `MARCH`, 13 `LANDBC`, and 14 `SEABC`).
- Smooth the computed hydrodynamic quantities using (58)–(60) (Subr. 15 `SMOOTH`).
- Compute the bottom shear stress using (28) and (32) (Subr. 16 `BSTRES`).
- Compute the mean, root-mean-square, maximum, and minimum values of the hydrodynamic quantities if $t_{\text{stat}} < t^*$ (Subr. 17 `STATIS`).
- Compute the wave energy quantities using (23), (24), (27) and (39) if `IENERG=1` and $t_{\text{stat}} < t^*$ (Subr. 18 `ENERGY`).

4.3.5 DOCUMENTATION

The following subroutines are used to document the computed results:

- Subr. 19 `DOC1` to store the input data and related parameters before the time-marching computation.
- Subr. 20 `DOC2` to store the spacial and temporal variations of certain variables during the time-marching computation.
- Subr. 21 `DOC3` to store the computed results after the time-marching computation.

• 4.4 •

SUBROUTINES AND FUNCTION

The 22 subroutines and one function arranged in numerical order in the computer program VBREAK are listed in Table 1. The page numbers for the subroutines and function listed in Table 1 correspond to the page numbers in the VBREAK listing presented in Appendix A. Interdependence among the program units are mapped out in Table 2. Each of the subroutines and function are explained concisely in the following. Explanation is given in the format: [Number] – [`NAME`] – [Description], where the [Number] refers to the numerical order in the computer program VBREAK .

01 `OPENER` opens the input and output files.

02 `INPUT1` reads information from the primary input data file and checks whether the selected options are within the ranges available or recommended in VBREAK .

03 `INPUT2` reads the input wave train, that is, the free surface profile at the seaward boundary prescribed by a user as explained in Sections 3.3.2 and 3.3.3.

04 `BOTTOM` computes the normalized bottom geometry and bottom friction factors as well as the value of Δx from the dimensional structure geometry and bottom friction factors specified as input.

05 `PARAM` calculates the dimensionless wave and velocity profile parameters used in other subroutines.

06 `INIT` specifies the initial conditions of no wave action at $t = 0$.

07 `INCREG` computes the incident periodic wave profile at the seaward boundary using (61) or (64) for the case of `IWAVE=1`.

08 `FINDM` computes the value of the cnoidal wave parameter m which satisfies (67).

09 `CEL` computes the values of the complete elliptic integrals K and E used in (64)–(67) for given m. `CEL` is the only function in the computer program VBREAK .

10 `SNCNDN` computes the Jacobian elliptic function cn used in (64).

11 `COMPDT` computes the time step size Δt using (57).

12 `MARCH` performs the time-marching computation using (54)–(56).

13 `LANDBC` takes care of the landward boundary conditions by computing the shoreline movement and the normalized free surface elevation Z_r for given δ_r' as discussed in Section 3.6.

14 `SEABC` takes care of the seaward boundary conditions and computes the reflected wave train $\eta_r(t)$ as explained in Section 3.5.

15 `SMOOTH` smooths the hydrodynamic quantities computed in Subr. 12 `MARCH`.

16 `BSTRES` computes the near-bottom horizontal velocity and the bottom shear stress τ_b using (28) with (32).

17 `STATIS` computes the mean, root-mean-square, maximum, and minimum values of η_i, η_r, Z_r, U, u_b, η and q during $t_{\text{stat}} \leq t \leq t_{\text{max}}$ as explained in Section 3.4.

18 `ENERGY` computes the values of E, E_F, D_f, and D_B defined by (23), (24), (27) and (39), respectively, and checks whether the time-averaged wave energy equation (46) is satisfied or not.

19 `DOC1` documents the input data and dimensionless parameters before the time-marching computation.

20 `DOC2` stores the temporal variations of η_i, η_r and Z_r as well as the spacial and temporal variations of η, U and $\tilde{u}_b$ at designated time levels during the time-marching computation.

Table 1: List of 22 subroutines and one function in computer program VBREAK.

No.	Subroutine (S) or Function (F)		Page No. in VBREAK
01	S	OPENER	A-8 - A-12
02	S	INPUT1	A-12 - A-18
03	S	INPUT2	A-18 - A-19
04	S	BOTTOM	A-19 - A-22
05	S	PARAM	A-22 - A-23
06	S	INIT	A-23 - A-26
07	S	INCREG	A-26 - A-29
08	S	FINDM	A-29 - A-30
09	F	CEL	A-30 - A-32
10	S	SNCNDN	A-32 - A-33
11	S	COMPDT	A-33 - A-34
12	S	MARCH	A-34 - A-37
13	S	LANDBC	A-37 - A-41
14	S	SEABC	A-42 - A-44
15	S	SMOOTH	A-44 - A-46
16	S	BSTRES	A-46
17	S	STATIS	A-46 - A-51
18	S	ENERGY	A-51 - A-54
19	S	DOC1	A-54 - A-59
20	S	DOC2	A-59 - A-61
21	S	DOC3	A-61 - A-64
22	S	CHEPAR	A-64 - A-65
23	S	CHEOPT	A-65 - A-66

Table 2: Interdependence among the program units of computer program VBREAK.

No.	Program Unit	Called from	Makes Call(s) to
00	MAIN	--	01, 02, 03, 04, 05, 06, 07, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21
01	OPENER	00	22
02	INPUT1	00	22, 23
03	INPUT2	00	22
04	BOTTOM	00	22
05	PARAM	00	--
06	INIT	00	22
07	INCREG	00	08, 09, 10, 22
08	FINDM	07	09
09	CEL	07, 08	--
10	SNCNDN	07	--
11	COMPDT	00	22
12	MARCH	00	22
13	LANDBC	00	22
14	SEABC	00	22
15	SMOOTH	00	22
16	BSTRES	00	22
17	STATIS	00	22
18	ENERGY	00	22
19	DOC1	00	22
20	DOC2	00	22
21	DOC3	00	22
22	CHEPAR	01, 02, 03, 04 06, 07, 11, 12 13, 14, 15, 16 17, 18, 19, 20 21	--
23	CHEOPT	02	--

21 DOC3 documents the computed results after the time-marching computation.

22 CHEPAR checks whether the values of the integers N1R, N2R, N3R, N4R, and N5R used to specify the size of matrices and vectors in the main program are equal to the values of the corresponding integers N1, N2, N3, N4, and N5 used in the subroutines.

23 CHEOPT checks whether the options in Subr. 02 INPUT1 selected by a user are within the ranges available in the present form of VBREAK .

• 4.5 •

PARAMETERS AND VARIABLES IN COMMON BLOCKS

The parameters and variables included in the COMMON blocks in the main program VBREAK are explained in the following so that a user may be able to comprehend the computer program VBREAK and modify it if required.

/DIMENS/ contains the integers used to specify the sizes of matrices and vectors.

N1R = N1 = maximum number of spacial nodes allowed in the computation domain.

N2R = N2 = maximum number of data points allowed in the input wave train and output time series.

N3R = N3 = maximum number of different values of the physical water depth δ'_r for wave runup.

N4R = N4 = maximum number of points allowed to specify the bottom geometry consisting of linear segments.

N5R = N5 = maximum number of spacial nodes where the time series of η, U and $\tilde{u}_b$ are stored as well as the maximum number of specified time levels at which the spacial variations of η, U and $\tilde{u}_b$ are stored.

The present setting for these integers is N1=800, N2=70000, N3=3, N4=100, N5=40. These integers can be changed as long as the changes are made throughout the program.

/CONSTA/ contains basic constants.

PI = π = 3.141592...

GRAV = gravitational acceleration, $g = 9.81 m/s^2$ or $32.2 ft/s^2$.

/ID/ contains the integers used to specify the user's options.

ISYST indicates the system of units. ISYST=1 for the International System of Units (SI), and ISYST=2 for the U.S. Customary System of Units (USCS).

IWAVE indicates the type of the wave profile specified at the seaward boundary. IWAVE=1 for the incident regular wave profile $\eta_i(t)$ computed using Subr. 07 INCREG, IWAVE=2 for the incident irregular wave profile $\eta_i(t)$ read by Subr. 03 INPUT2, and IWAVE=3 for the measured wave profile $[\eta_i(t) + \eta_r(t)]$ read by Subr. 03 INPUT2. It is noted that IWAVE=3 corresponds to the free surface oscillation measured at the seaward boundary in the presence of a coastal structure or beach.

IBOT indicates the type of input data for the bottom geometry divided into linear segments of different slopes and roughness. IBOT=1 for the width and slope of linear segments, and IBOT=2 for the locations of the landward end points of linear segments.

INCLCT indicates whether the nonlinear correction term C_t is included or not in (75) in calculating the reflected wave profile $\eta_r(t)$. INONCT=0 for $C_t = 0$, and INONCT=1 for C_t given (77). $C_t = 0$ for IWAVE=3 because the measured wave profile includes this correction implicitly.

IENERG indicates whether the quantities related to wave energy are computed (IENERG=1) or not (IENERG=0).

ITEMVA: ITEMVA=1 indicates the storage of the time series of certain hydrodynamic quantities during $0 \leq t \leq t_{\text{max}}$. ITEMVA=0 indicates no storage.

ISPAEU: ISPAEU=1 indicates the storage of the spatial variations of η, U and $\tilde{u}_b$ in the computation domain at specified time levels. ISPAEV=0 indicates no storage.

/TLEVEL/ contains the time levels for the time-marching computation.

TIME = present time level t.

TIMEST = next time level $t^* = (t + \Delta t)$.

TSTAT = starting time t_{stat} for the statistical calculations explained in Section 3.4.

TMAX = computation duration t_{max} where the computation is performed for $0 \leq t \leq t_{\text{max}}$ and the statistical calculations are made for $t_{\text{stat}} \leq t \leq t_{\text{max}}$.

/NODES/ contains the locations of spacial nodes.

STILL = number of the nodal intervals between the seaward boundary at $x = 0$ and the still water shoreline located at $x = x_{\text{SWL}}$. The nodes are viewed to be located on the surface of the bottom. STILL = 100–600 has been used where the constant nodal interval $\Delta x = x_{\text{SWL}}/\text{STILL}$. It is required that $100 \leq \text{STILL} \leq (\text{N1} - 1)$.

S = location of the wet node s next to the moving shoreline at the present time t.

SST = location of the wet node s^* next to the moving shoreline at the next time t^* where $s^* = (s - 1)$, s or $(s + 1)$.

SMAX = maximum value of the wet nodal location s during $0 \leq t \leq t_{\text{max}}$.

JMAX = number of the spacial nodes in the computation domain. It is required that $\text{JMAX} \leq \text{N1}$.

/GRID/ contains the nodal interval, time step size and related quantities.

DX = constant nodal interval Δx used in the numerical model.

DT = time step size Δt computed using (57) at the beginning of each time step except that $t^* = t_{\text{max}}$ and $\Delta t = (t_{\text{max}} - t)$ for the last time step of the computation for $0 \leq t \leq t_{\text{max}}$.

DXDT = $\Delta x/\Delta t$

DTDX = $\Delta t/\Delta x$

DTMAX = maximum value of Δt used for the computation.

DTMIN = minimum value of Δt used for the computation.

/CPARA/ contains the computational parameters introduced in the numerical model.

DELTA = normalized water depth δ defining the computational shoreline. The range of δ = 0.001–0.003 has been used and the increase of δ tends to improve numerical stability near the moving shoreline.

COURNO = Courant number C_n introduced in (57) where $C_n \leq 1$ for numerical stability.

DKAPPA = numerical damping coefficient κ introduced in (59).

/WAVREF/ contains the input wave parameters.

HREF = physical reference wave height, H', specified in *meters* when SI is used or in *feet* when USCS is used. H' is used to normalize the dimensional variables and parameters in (4)–(6).

TREF = physical reference wave period. T', in *seconds*. T' is used to normalize the dimensional variables and parameters in (4)–(6).

KS = Normalized regular wave height at the seaward boundary specified as input for IWAVE=1. $K_s = 1$ is set in Subr. 02 INPUT1 for IWAVE=2 and 3.

/VERPAR/ contains the parameters related to the vertical velocity profile.

APROFL = cubic profile parameter a introduced in (40), which needs to be specified as input.

CMIXL = mixing length parameter C_ℓ introduced in (35), which needs to be specified as input.

C2 = parameter C_2 defined in (37) and computed using (41) in Subr. 05 PARAM.

C3 = parameter C_3 defined in (38) and computed using (42) in Subr. 05 PARAM.

CB = parameter C_B defined in (39) and computed using (43) in Subr. 05 PARAM.

CBL = parameter $C_{B\ell} = C_B C_\ell^2 \sigma$ defined in (79) and computed in Subr. 05 PARAM.

/WAVINP/ contains the input wave profile at the seaward boundary.

DELTI = constant sampling rate δt_i given by (68) where the sampling time $t = (n-1)\delta t_i$.

ETAINP(n) = incident regular wave train $\eta_i(t)$ for IWAVE=1 computed by Subr. 07 INCREG, the incident irregular wave train $\eta_i(t)$ for IWAVE=2 read from the input file or the measured total free surface oscillation $\eta(t)$ at $x = 0$ for IWAVE = 3 read from the input file.

NPINP = number of points in the input wave train ETAINP(n) with n=1,2,...,NPINP during $0 \leq t \leq t_{\max}$. It is required that 200 $\leq$ NPINP $\leq$ N2 to resolve the input wave train.

/IRWAVE/ contains the incident and reflected wave profiles at the seaward boundary.

ETAI = incident wave free surface elevation η_i at the present time t.

ETAIST = incident wave free surface elevation η_i^* at the next time t^*.

ETAR = reflected wave free surface elevation η_r at the present time t.

ETARST = reflected wave free surface elevation η_r^* at the next time t^*.

/WAVPAR/ contains the dimensionless wave parameters.

SIGMA = ratio of the horizontal and vertical length scales, σ, defined in (6) where $\sigma^2 \gg 1$ is assumed.

WL = normalized linear wavelength, $L = L'/d_t'$, at the seaward boundary unless cnoidal wave theory is adopted for IWAVE=1.

UR = Ursell parameter U_r at the seaward boundary used in Section 3.3.1.

KSI = surf similarity parameter (Battjes 1974) based on the slope SLSURF specified as input.

/CNOWAV/ contains parameters related to cnoidal wave theory.

KCNO = complete elliptic integral of the first kind, $K(m)$, used in (64)–(67).

ECNO = complete elliptic integral of the second kind, $E(m)$, used in (65) and (67).

MCNO = parameter m computed from (67).

KC2 = value of $(1-m)$ used to compute the values of $K(m)$ and $E(m)$ using the Function 09 CEL.

/BOTPAR/ contains parameters related to the bottom geometry.

DSEAP = water depth d_t' below SWL at the seaward boundary.

DSEA = normalized water depth, $d_t = d_t'/H'$, at the seaward boundary.

SLSURF = tangent of slope, $tan\theta_\xi'$, used to define the surf similarity parameter $\xi = \sigma \tan\theta_\xi' / \sqrt{2\pi}$.

WTOT = normalized horizontal width, (JMAX $-$ 1)Δx, of the computation domain.

/BOTSEG/ contains the dimensional bottom geometry specified as input.

WBSEG(i) = horizontal width of the linear segment i with i = 1,2,...,NBSEG numbered in the landward direction.

TBSLOP(i) = tangent of the slope of the segment i which is negative if the slope is downward in the landward direction.

XBSEG(i+1) = horizontal distance from the seaward boundary located at $x' = 0$ to the landward end of the segment i where XBSEG(1)=0.

ZBSEG(i+1) = elevation relative to SWL at the landward end of the segment i which is *negative* if the end point is located *below* SWL and ZBSEG(1) = $-d_t'$.

BFFSEG(i) = wave friction factor f_w' in (28) of the segment i.

NBSEG = number of linear segments of different inclinations and roughness used to specify the bottom geometry. It is required that $1 \leq$ NBSEG $\leq$ (N4 $-$ 1).

/BOTNOD/ contains vectors related to the normalized bottom geometry. The spatial nodes are viewed to be located on the bottom surface. Index j refers to node number with j=1,2,...,JMAX.

XB(j) = normalized x-coordinate of the node j given by XB(j) $= (j-1)\Delta x$.

ZB(j) = normalized z-coordinate of the node j, corresponding to the normalized bottom elevation z_b defined in (11).

THETA(j) = normalized gradient of the slope, θ_j, at the node j where $\theta = dz_b/dx$.

/WRUNUP/ contains quantities related to wave runup.

DELRP(i) = different values of δ_r' with i = 1,2,..., NDELR being specified in *centimeters* when SI is used and in *inches* when USCS is used. Each value of DELRP is independent of the others.

DELTAR(i) = normalized water depth $\delta_r = \delta_r'/H'$ corresponding to the different values of δ_r'. Derived from DELRP, each value of DELTAR is also independent of the others.

RUNZ(i) = normalized instantaneous free surface elevation Z_r above SWL at the location of $h = \delta_r$ at the present time t, where RUNZ(i) corresponds to DELTAR(i).

RUNZST(i) = value of Z_r^* at the next time t^*.

NDELR = number of different values of the physical water depth δ_r' associated with the measured or visual shoreline for which the normalized free surface elevation Z_r is computed as discussed in relation to wave runup in Section 3.6. It is required that $0 \leq$ NDELR $\leq$ N3 where wave runup Z_r is not computed if NDELR=0.

/HQUETA/ contains the computed normalized hydrodynamic quantities at node j with j = 1,2,..., JMAX.

H(j) = instantaneous water depth h_j at the present time t.

Q(j) = volume flux q_j per unit width at the time t.

U(j) = depth-averaged horizontal velocity, $U_j = q_j/h_j$, at the time t.

ETA(j) = free surface elevation η_j above SWL at the time t.

HST(j) = instantaneous water depth h_j^* at the next time t^*.

QST(j) = volume flux q_j^* per unit width at the time t^*.

UST(j) = depth-averaged horizontal velocity, $U_j^* = q_j^*/h_j^*$, at the time t^*.

ETAST(j) = free surface elevation η_j^* above SWL at the time t^*.

/VERVAR/ contains the normalized hydrodynamic quantities associated with the vertical velocity variations at node j with j = 1,2,...JMAX.

FM(j) = momentum flux correction m_j defined by (21) at the present time t.

UB(j) = near-bottom horizontal velocity correction $(\tilde{u}_b)_j$ defined by (32) at the present time t.

FMST(j) = momentum flux correction m_j^* at the next time t^*.

UBST(j) = near-bottom horizontal velocity correction $(\tilde{u}_b)_j^*$ at the time t^*.

FM3(j) = kinetic energy flux correction $(m_3)_j$ defined by (25) at the time t and $(m_3)_j^*$ at the time t^* at the beginning and end of each time step, respectively.

DB(j) = energy dissipation rate outside the wave boundary layer due to wave breaking, $(D_B)_j$, given by (39) at the time t and $(D_B)_j^*$ at the time t^* at the beginning and end of each time step, respectively.

/TAUBFW/ contains the normalized quantities related to the bottom shear stress at node j with j=1,2,..., JMAX.

UUB(j) = near bottom horizontal velocity $(u_b)_j$ at the time t and $(u_b)_j^*$ at the time t^* at the beginning and end of each time step, respectively.

TAUB(j) = bottom shear stress $(\tau_b)_j$ given by (28) at the time t and $(\tau_b)_j^*$ at the time t^* at the beginning and end of each time step, respectively.

FW(j) = normalized wave friction factor $(f_w)_j$ defined in (28).

/STOTEP/ contains time levels for storing time series if ITEMVA=1.

DELTO = normalized sampling rate, $\delta t_0 = t_{\max}/(\text{NPOUT} - 1)$, for storing time series.

TIMOUT(n) = time level, $(n-1)\delta t_0$, with n=1,2,..., NPOUT for storing time series.

NPOUT = number of time levels for storing time series. It is required that $2 \leq$ NPOUT $\leq$ N2.

/STONOD/ contains the nodal locations for storing the time series of η, U and $\tilde{u}_b$ at the sampling rate δt_0 if ITEMVA=1.

NONODS = number of nodes where the time series of η, U and $\tilde{u}_b$ are stored. It is required that $0 \leq$ NONODS $\leq$ N5 where these time series are not stored if NONODS=0

NODLOC(i) = nodal locations with i = 1,2,..., NONODS for storing these time series.

/STOSPA/ contains the specified time levels for storing the spacial variations of η, U and $\tilde{u}_b$ if ISPAEU=1.

TIMSPA(i) = specified time levels with i = 1,2,..., NOTIML for storing these spacial variations.

NOTIML = number of the specified time levels for storing these spacial variations. It is required that $0 \leq$ NOTIML $\leq$ N5 where NOTIML=0 corresponds to no storage.

/EISTAT/ contains the statistical values of the incident wave train $\eta_i(t)$ where the statistical calculations are performed during $t_{\text{stat}} \leq t \leq t_{\max}$ as explained in Section 3.4.

EIMEAN = mean value $\bar{\eta}_i$.

EIRMS = root-mean-square value $(\eta_i)_{\text{rms}}$ where the rms value is defined in the manner given by (69) and equals the standard deviation.

EIMAX = maximum value $(\eta_i)_{\max}$.

EIMIN = minimum value $(\eta_i)_{\min}$.

/ERSTAT/ contains the statistical values of the reflected wave train $\eta_r(t)$ at the seaward boundary.

ERMEAN = mean value $\bar{\eta}_r$.

ERRMS = rms value $(\eta_r)_{\rm rms}$.

ERMAX = maximum value $(\eta_r)_{\rm max}$.

ERMIN = minimum value $(\eta_r)_{\rm min}$.

REFCOE = average reflection coefficient r defined by (88).

/RZSTAT/ contains the statistical values of the time-varying shoreline elevation Z_r above SWL at the location $h = \delta_r =$ DELTAR(i) with i=1,2,..., NDELR.

RZMEAN(i) = mean value $\bar{Z}_r$, that is, wave setup for the specified δ_r'.

RZRMS(i) = rms value $(Z_r)_{\rm rms}$ indicating the degree of the shoreline oscillation.

RZMAX(i) = maximum value $(Z_r)_{\rm max}$, that is, wave runup R_u.

RZMIN(i) = minimum value $(Z_r)_{\rm min}$, that is, wave run-down R_d.

/ETSTAT/ contains the statistical values of the free surface elevation η_j at node j with j = 1,2,..., SMAX.

EMEAN(j) = mean value $\bar{\eta}_j$.

ERMS(j) = rms value $(\eta_j)_{\rm rms}$.

EMAX(j) = maximum value $(\eta_j)_{\rm max}$.

EMIN(j) = minimum value $(\eta_j)_{\rm min}$.

/USTAT/ contains the statistical values of the depth-averaged horizontal velocity U_j at node j with j=1,2,..., SMAX.

UMEAN(j) = mean value $\bar{U}_j$.

URMS(j) = rms value $(U_j)_{\rm rms}$.

UMAX(j) = maximum value $(U_j)_{\rm max}$.

UMIN(j) = minimum value $(U_j)_{\rm min}$.

/UBSTAT/ contains the statistical values of the near-bottom horizontal velocity $(u_b)_j$ at node j with j=1,2,..., SMAX.

UBMEAN(j) = mean value $\overline{(u_b)_j}$.

UBRMS(j) = rms value $[(u_b)_j]_{\rm rms}$.

UBMAX(j) = maximum value $[(u_b)_j]_{\rm max}$.

UBMIN(j) = minimum value $[(u_b)_j]_{\rm min}$.

/QSTAT/ contains the mean value of the volume flux q_j at node j with j=1,2,..., SMAX.

QMEAN(j) = mean value $\bar{q}_j$ during $t_{\rm stat} \le t \le t_{\rm max}$ where the time-averaged continuity equation derived from (18) for the assumed impermeable bottom requires $\bar{q} = 0$ if $h(t = t_{\rm stat}) = h(t = t_{\rm max})$ or $\bar{q} \simeq 0$ if the computed results are stationary for the sufficiently long duration of $(t_{\rm max} - t_{\rm stat}) \gg 1$.

/WESTAT/ contains the normalized quantities related to the time-averaged wave energy equation (46) for $t_{\rm stat} \le t \le t_{\rm max}$.

ESMEAN(j) = mean specific wave energy $\overline{E_j}$, at the node j with j=1,2,..., SMAX.

EFMEAN(j) = mean energy flux, $\overline{(E_F)_j}$, per unit width.

DFMEAN(j) = mean energy dissipation rate, $\overline{(D_f)_j}$, inside the wave boundary layer.

DBMEAN(j) = mean physical energy dissipation rate, $\overline{(D_B)_j}$, outside the wave boundary layer due to wave breaking.

DBMDIF(j) = difference between $\overline{(D_B)_j}$ computed using the time-averaged wave energy equation (46) and $\overline{(D_B)_j}$ based on (39). This difference will be zero in the absence of numerical energy dissipation but will likely be positive because the former may include numerical energy dissipation and be larger than the latter.

DELES(j) = quantity $(\Delta E)_j$ given by (47) which accounts for the increment of the specific wave energy E from $t = t_{\text{stat}}$ to $t = t_{\max}$. $\Delta E = 0$ if $E(t = t_{\text{stat}}) = E(t = t_{\max})$ or $\Delta E \simeq 0$ for $(t_{\max} - t_{\text{stat}}) \gg 1$.

/ENERG/ contains the normalized wave energy quantities at node j and at the present time t during $t^* = (t + \Delta t) > t_{\text{stat}}$ in Subr. 18 ENERGY.

ESPC(j) = specific wave energy E_j computed using (23).

EFLUX(j) = wave energy flux $(E_F)_j$ computed using (24).

DISF(j) = energy dissipation rate $(D_f)_j$ due to bottom friction computed using (27).

DISB(j) = energy dissipation rate $(D_B)_j$ due to wave breaking computed using (39).

/VECMAC/ contains the vectors at node j introduced in (49) for the MacCormack method.

F2(j) = function $(F_2)_j$ defined in (50).

F3(j) = function $(F_3)_j$ defined in (51).

G2(j) = function $(G_2)_j$ defined in (50).

G3(j) = function $(G_3)_j$ defined in (51).

/DOTMAC/ contains the temporary values at node j used in (54) and (55) for the MacCormack method.

HDOT(j) = $\dot{h}_j$ and then $\ddot{h}_j$ computed using (54) and (55), respectively.

QDOT(j) = $\dot{q}_j$ and then $\ddot{q}_j$ computed using (54) and (55), respectively.

UDOT(j) = temporary value, $\dot{U}_j = \dot{q}_j / \dot{h}_j$.

FMDOT(j) = $\dot{m}_j$ and then $\ddot{m}_j$ computed using (54) and (55), respectively.

UBDOT(j) = $(\dot{\bar{u}}_b)_j$ computed using (44) with $\dot{m}_j$ and $\dot{h}_j$ where it is enforced that $\dot{m}_j \geq 0$ and $\dot{h}_j \geq \delta$.

• 4.6 •

INPUT PARAMETERS AND VARIABLES

The contents of the primary input data file of unit=11 and file=FINP1 is read by Subr. 02 INPUT1. This subroutine provides clear explanations on the input parameters and variables it reads. It is recommended that a user follow the explanations in preparing the primary input data file. The READ statements in the primary input data file are explained in sequence.

The comment lines for the header of each input data set are read first.

```
      READ (11,1110) NLINES
1110  FORMAT (I8)
      DO 110 I=1, NLINES
          READ (11,1120) (COMMEN(J), J=1,14)
110   CONTINUE
1120  FORMAT (14A5)
```

where NLINES is the number of lines containing the user's comments.

The following options for the computation are read:

```
      READ(11,1130) ISYST
      READ(11,1130) IWAVE
      READ(11,1130) IBOT
      IF (IWAVE.LE.2) THEN
      READ(11,1130) INCLCT
      ELSE
      INCLCT=0
      ENDIF
      READ(11,1130) IENERG
      READ(11,1130) ITEMVA
      READ(11,1130) ISPAEU
1130  FORMAT(I1)
      IF(IWAVE.GT.1) READ(11,1140) FINP2
1140  FORMAT (A10)
```

in which ISYST=1 (m, cm) or 2 (ft, in); IWAVE=1 (regular η_i computed), 2 (irregular η_i read) or 3 (measured η read); IBOT=1 (width and slope of linear bottom segment) or 2 (coordinates of bottom segment landward end); INCLCT=0 [$C_t = 0$ in computing η_r using (75)] or 1 [C_t computed using (77) only for IWAVE=1 or 2]; IENERG=0 (not computed) or 1 (energy quantities computed); ITEMVA=0 (not stored) or 1 (time series stored); ISPAEU=0 (not stored) or 1 (spacial variations stored); and FINP2 = input data file name containing the input wave train for IWAVE=2 or 3.

The following normalized time levels are then read:

```
      READ(11,1150) TSTAT, TMAX
1150  FORMAT (3F13.6)
```

where TSTAT = starting time t_{stat} for the statistical calculations; and TMAX = computation duration $t_{\max}$.

The following computational parameters are read:

```
      READ(11,1110) STILL
1110  FORMAT(I8)
      READ(11,1150) DELTA, COURNO, DKAPPA
1150  FORMAT(3F13.6)
```

in which STILL = integer used to determine $\Delta x = x_{\text{SWL}}/\text{STILL}$ with x_{SWL} = horizontal distance between the seaward boundary and the still water shoreline ($100 \leq$ STILL $\leq$ (N1-1) required); DELTA = normalized water depth δ used to define the computational shoreline (normally δ = 0.001–01003 and increase δ to overcome numerical difficulties at the moving shoreline); COURNO = Courant number C_n in (57) with $C_n \leq 1$ for numerical stability (normally $C_n = 0.1$–0.9 and reduce C_n to overcome numerical difficulties at the moving shoreline); and DKAPPA = numerical damping coefficient κ in (59) on the order of unity or less (increase κ to reduce numerical high-frequency oscillations at the rear of a breaking wave).

The following wave properties are read next:

```
      READ(11,1150) HREF, TREF
      IF(IWAVE.EQ.1) READ(11,1150) KS
1150  FORMAT (3F13.6)
      READ(11,1110) NPINP
1110  FORMAT(I8)
```

where HREF = reference wave height H' (m or ft); TREF = reference wave period $T'(s)$; KS = incident regular wave height normalized by H' (KS=1 set for IWAVE=2 or 3); and NPINP = number of points in the input wave train ETAINP during $0 \leq t \leq t_{\max}$ sampled at the rate $\delta t_i = t_{\max}/(\text{NPINP} - 1)$. For IWAVE=1, ETAINP is computed using (61) or (64) and $(\delta t_i)^{-1}$ must be an even number to take advantage of the symmetrical profile about the wave crest. It is required that $200 \leq$ NPINP $\leq$ N2 to resolve the input wave train sufficiently.

The input velocity profile parameters are as follows:

```
      READ(11,1150) APROFL, CMIXL
1150  FORMAT(3F13.6)
```

where APROFL = cubic velocity profile parameter a in (40), which is expected to be in the range a = 3–4; and CMIXL = mixing length parameter C_ℓ on the order of 0.1.

The bottom geometry from the seaward boundary to the landward end above wave runup is specified in the following:

```
      READ(11,1150) DSEAP, SLSURF
1150  FORMAT(3F13.6)
      READ(11,1110) NBSEG
1110  FORMAT(I8)
      IF(IBOT.EQ.1) THEN
         DO 130 I=1, NBSEG
            READ(11,1150) WBSEG(I), TBSLOP(I), BFFSEG(I)
130      CONTINUE
      ELSE
         XBSEG(1) = 0.D+00
         ZBSEG(1) = - DSEAP
```

```
            DO 140 I=2, NBSEG+1
               READ(11,1150) XBSEG(I), ZBSEG(I), BFFSEG(I-1)
  140       CONTINUE
         ENDIF
```

in which DSEAP = water depth d'_t (m or ft) below SWL at the seaward boundary located at $x' = 0$; SLSURF = appropriate slope used to calculate the surf similarity parameter (normally, the averaged slope in the computation domain or the slope at the still water shoreline); NBSEG = number of linear bottom segments where it is required that $1 \leq$ NBSEG $\leq$ (N4 $-$ 1); WBSEG(I) = horizontal width (m or ft) of segment I; TBSLOP(I) = slope of segment I (+ upslope and $-$ downslope landward); BFFSEG(I) = bottom friction factor f'_w in (28) of segment I (normally $f'_w = 0.01$–0.05 for smooth slopes and $f'_w = 0.05$–0.3 for rough slopes); XBSEG(I) = horizontal distance (m or ft) from $x' = 0$ to the landward end of segment (I-1); and ZBSEG(I) = elevation (m or ft) above SWL of the landward end of segment (I-1). It is noted that XBSEG(1)=0 and ZBSEG(1) = $-d'_t$ are set above so that the number of the READ lines is NBSEG for IBOT=2 as well.

The time levels and storage of the certain computed time series are specified in the following steps:

```
         IF(ITEMVA.EQ.1) THEN
         READ(11,1110) NPOUT
  1110   FORMAT(I8)
         ENDIF
```

where NPOUT = number of time levels for storing the certain computed time series at the sampling rate, $\delta t_0 = t_{\max}/(\text{NPOUT} - 1)$, from $t = 0$ to $t = t_{\max}$. It is required that $2 \leq$ NPOUT $\leq$ N2. If ITEMVA=1, the incident wave train $\eta_i(t)$ and the reflected wave train $\eta_r(t)$ at $x = 0$ are stored.

In order to store the time series of computed shoreline elevations, the following additional input parameters need to be specified:

```
         IF(ITEMVA.EQ.1) THEN
            READ(11,1110) NDELR
  1110      FORMAT(I8)
            IN(NDELR.GT.0) THEN
               DO 160 L=1, NDELR
               READ(11,1150) DELRP(L)
  1150         FORMAT(3F13.6)
  160          CONTINUE
            ENDIF
         ENDIF
```

in which NDELR = number of the water depths δ'_r (cm or in) used to trace the normalized elevation above SWL of the intersection between the free surface and the real or hypothetical runup wire placed at the vertical distance δ'_r above and parallel to the bottom. It is required that $0 \leq$ NDELR $\leq$ N3. If NDELR > 0, the NDELR values of δ'_r (cm or in) are read and the time series of the corresponding shoreline elevations are computed and stored for $0 \leq t \leq t_{\max}$.

To store the computed time series of the free surface elevation η, the depth-averaged velocity U, and the near-bottom horizontal velocity correction $\tilde{u}_b$ defined by (32) at certain nodes, their nodal locations need to be specified as follows:

```
      IF(ITEMVA.EQ.1) THEN
          READ(11,1110) NONODS
1110      FORMAT(I8)
          IF(NONODS.GT.0) READ(11,1160) (NODLOC(I), I=1, NONODS)
1160      FORMAT(5I6)
      ENDIF
```

where NONODS = number of nodes for storing the time series of η, U and $\tilde{u}_b$ at the sampling rate δt_0 during $0 \leq t \leq t_{\max}$. It is required that $0 \leq$ NONODS $\leq$ N5. If NONODS > 0, the nodal locations NODLOC(I) with I=1,2,..., NONODS are read and these time series at the specified nodes are stored.

At the end of the primary input data file, the time levels for storing the spatial variations of η, U and $\tilde{u}_b$ are read if ISPAEU=1.

```
      IF(ISPAEU.EQ.1) THEN
          READ(11,1110) NOTIML
1110      FORMAT (I8)
          IF(NOTIML.GT.0) READ(11,1170) (TIMSPA(I), I=1, NOTIML)
1170      FORMAT (5F12.5)
      ENDIF
```

where NOTIML = number of time levels for storing these spacial variations. It is required that $0 \leq$ NOTIML $\leq$ N5. If NOTIML > 0, the time levels TIMSPA(I) with i=1,2,..., NOTIML are read and the spacial variations of η, U and $\tilde{u}_b$ in the computation domain are stored at the specified time levels.

Finally, if IWAVE=2 or 3, the input wave train ETAINP is read in Subr. 03 INPUT2 from the file of unit = 12 whose name is read as the variable FINP2 after the options are read.

```
      READ(12,1210) (ETAINP(I), I=1, NPINP)
1210  FORMAT(5D15.6)
```

where the number of points, NPINP, in the input wave train is read as one of the wave properties. The input wave train for $0 \leq t \leq t_{\max}$ is read at the constant normalized sampling rate δt_i given by (68). For IWAVE=2, the input wave train is the incident wave free surface elevation above SWL, $\eta_i(t)$ at $x = 0$, normalized by the reference wave height H' specified as input. For IWAVE=3, the input wave train is the measured wave free surface elevation above SWL, $\eta(t) = [\eta_i(t) + \eta_r(t)]$ at $x = 0$ normalized by H'. The normalization of t and $t_{\max}$ is made using the reference wave period T' specified as input.

• 4.7 •

WARNING AND ERROR MESSAGES

Warning and error messages are written in the file OMSG (unit=29) and displayed on screen. There are warning messages that the computer program VBREAK may issue, that is,

```
From Subr.  08 FINDM:
Criterion for parameter m=MCNO not satisfied
```

and

```
From Subr. 13 LANDBC:
Computed water depth HST(S) > HST(S-1) at S = ···; TIMEST = ···
Adjusted values:  HST(S) = ··· ; HST(S-1) = ···
```

The first warning is related to the iteration scheme to compute the parameter m using (67), and thus corresponds to the case of regular cnoidal waves only. This warning has never been experienced before. The second warning is related to the somewhat arbitrary adjustments made in the second step of the shoreline algorithm in Section 3.6. VBREAK does not automatically cease computation when these warnings are issued.

Computation is terminated immediately following any error message as summarized in the following. The error messages are self-explanatory.

1. If the value of any of the PARAMETERs N1, N2, N3, N4, and N5 explained in COMMON/DIMENS/ in Section 4.5 in a subroutine that utilizes the PARAMETERs does not match with the corresponding value specified in the main program, the following message is written in Subr. 22 CHEPAR:

```
PARAMETER Error:  N· = ··· in Subroutine ···
Correct Value:  N· = ···
```

2. Subr. 23 CHEOPT writes the error message if any of the 7 options explained in COMMON/ID/ in Section 4.5, denoted by ITEM, specified as input by a user is not in the following range: ISYST=1 or 2; IWAVE=1, 2 or 3; IBOT = 1 or 2; INCLCT = 0 or 1; IENERG = 0 or 1; ITEMVA = 0 or 1; and ISPAEU = 0 or 1.

```
Input Error:  ITEM = ···
Specify ITEM in the ranges of [···,···]
```

3. If the requirement $100 \leq$ STILL $\leq$ (N1 $-$ 1) is not satisfied, Subr. 23 CHEOPT writes

```
Input Error:  STILL = ···
Specify STILL in the recommended range of [100,···]
Change PARAMETER N1 if necessary
```

4. If the requirement $200 \leq$ NPINP $\leq$ N2 is not satisfied, Subr. 23 CHEOPT writes

```
Input Error:  NPINP = ···
Specify NPINP in the recommended range of [200,···]
Change PARAMETER N2 if necessary
```

5. If the requirement $1 \leq$ NBSEG $\leq$ (N4 $-$ 1) is not satisfied, Subr. 23 CHEOPT writes

```
Input Error:  NBSEG = ···
Specify NBSEG in the recommended range of [1,···]
Change PARAMETER N4 if necessary
```

6. If the requirement $2 \leq$ NPOUT $\leq$ N2 is not satisfied, Subr. 23 CHEOPT writes

```
Input Error:  NPOUT = ···
Specify NPOUT in the recommended range of [2,···]
Change PARAMETER N2 if necessary
```

7. If the requirement $0 \leq$ NDELR $\leq$ N3 is not satisfied, Subr. 23 CHEOPT writes

```
Input Error:  NDELR = ···
Specify NDLER in the recommended range of [0,···]
Change PARAMETER N3 if necessary
```

8. If the requirement $0 \leq$ NONODS $\leq$ N5 is not satisfied, Subr. 23 CHEOPT writes

```
Input Error:  NONODS = ···
Specify NONODS in the recommended range of [0,···]
Change PARAMETER N5 if necessary
```

9. If the requirement $0 \leq$ NOTIML $\leq$ N5 is not satisfied, Subr. 23 CHEOPT writes

```
Input Error:  NOTIML = ···
Specify NOTIML in the recommended range of [0,···]
Change PARAMETER N5 if necessary
```

10. If the bottom geometry specified as input is submerged, Surb. 04 BOTTOM writes the following message:

```
Bottom is always below SWL.
There is no still water shoreline.
```

11. If the horizontal length of the bottom geometry specified as input is too long for the maximum number N1 of the spacial nodes allowed in the present form of VBREAK , Subr. 04 BOTTOM writes

```
End Node = ··· ; N1 = ···
Bottom length is too long.
Cut it, or change PARAMETER N1.
```

12. If the values of NPINP and TMAX specified as input do not satisfy that NONE = (NPINP - 1)/TMAX is an even number for IWAVE = 1, Subr. 07 INCREG writes

```
Number of input wave points NPINP = ···
Computation duration         TMAX = ···
    NONE = (NPINP - 1)/TMAX = ···
NONE must be an even number for IWAVE=1.
Change input value of NPINP or TMAX
```

13. If the complete elliptic integral for cnoidal wave theory can not be computed, Function 09 CEL writes

```
Failure in Function CEL
```

14. If the water depth h becomes negative at node j, Subr. 11 COMPDT writes

```
From Subr. 11 COMPDT: Negative water depth = ···
J = ··· ; S = ··· ; TIME = ···
```

15. If $h^*_{s-1} \leq \delta$ as explained in the first step of the shoreline algorithm in Section 3.6, the following error message is written in Subr. 12 MARCH:

```
From Subroutine 12 MARCH
Computed water depth HST(S-1) is less than or equal to DELTA
HST(S-1) = ···
DELTA = ···
S = ···
TIME = ···
Program Aborted
```

16. If the moving shoreline reaches the landward end of the computation domain, Subr. 13 LANDBC writes

```
From Subroutine 13 LANDBC:
TIMEST =··· ; SST = ··· ; End Node =···
Slope is not long enough to accommodate shoreline movement
Specify longer slope to avoid wave overtopping
```

17. If the assumption of $U < c = \sqrt{h}$ at the seaward boundary made in Section 3.5 is not satisfied, the flow is not subcritical and Subr. 14 SEABC writes

```
From Subr. 14 SEABC: Seaward Boundary
(Flow at x = 0 is not subcritical)
Time of occurrence          Time = ···
Water velocity at x = 0     U = ···
Phase velocity at x = 0     c = ···
```

The parameters and variables involved in the above error messages are explained in Section 4.5 as indicated below.

VARIABLE	DESCRIPTION IN SECTION 4.5
N1,N2,N3,N4,N5	COMMON /DIMENS/
TIME, TIMEST, TMAX	COMMON /TLEVEL/
STILL, S, SST	COMMON /NODES/
DELTA	COMMON /CPARA/
NPINP	COMMON /WAVINP/
NBSEG	COMMON /BOTSEG/
NDELR	COMMON /WRUNUP/
H, U, HST	COMMON /HQUETA/
NPOUT	COMMON /STOTEP/
NONODS	COMMON /STONOD/
NOTIML	COMMON /STOSPA/

In the computer program attached in Appendix A, N1 = 800, N2 = 70000, N3 = 3, N4 = 100, N5 = 40.

• 4.8 •
OUTPUT PARAMETERS AND VARIABLES

Output from the computer program VBREAK is stored in files whose names start with the letter "O" for easy identification. The number of output files varies depending on the options selected by a user. Table 3 lists the names of all possible output files generated by VBREAK. The files ODOC and OMSG in Table 3 contain information which should be read and checked by the user. The rest contain the computed results that may need further processing to yield useful information for the user. The files OSPACE, OSTAT, and OENERG contain spacial variations. The files OIRWAV and ORUNUP contain time series covering the time interval $0 \leq t \leq$ TMAX and so do the families of files OSTORExx, OSTORUxx, and OSTOUBxx.

4.8.1 GENERAL OUTPUT

The output files OSPACE, OSTAT, ODOC and OMSG are always generated by VBREAK . The file OMSG contains the warning and error messages discussed in Section 4.7 and is not explained further.

The file OSPACE is written in Surb 19 DOC1 as follows:

```
      WRITE(22,2210) JMAX
      WRITE(22,2220) (XB(J), ZB(J), J=1, JMAX)
2210  FORMAT(I8)
2220  FORMAT(6D12.4)
```

in which JMAX ≐ maximum node number in the computation domain; XB(J) = normalized

Table 3: Summary of Output Files

Unit	File Name	Output parameters and variables stored in file
22	OSPACE	Normalized bottom geometry (XB(J), ZB(J), J=1, JMAX). If ISPAEU=1, spacial variations of η, U and $\tilde{u}_b$ at specified time levels TIMSPA(I) with I = 1, 2, $\cdots$, NOTIML.
23	OSTAT	Spacial variations of mean, rms, maximum and minimum values of η, U, u_b and q.
28	ODOC	Essential parameters for concise documentation.
29	OMSG	Error and warning messages during computation.
30	OIRWAV	If ITEMVA=1, incident wave train $\eta_i(t)$ and reflected wave train $\eta_r(t)$ at $x = 0$ sampled at rate δt_0 during $0 \leq t \leq t_{\max}$.
31	ORUNUP	If ITEMVA=1 and NDELR $>$ 0, time series of shoreline elevations Z_r corresponding to specified NDELR values of water depth δ_r', sampled at rate δt_0 during $0 \leq t \leq t_{\max}$.
35	OENERG	If IENERG=1, spacial variations of time-averaged wave energy quantities.
41	OSTORE01 OSTORE02 :	If ITEMVA=1 and NONODS > 0, time series of free surface η at NONODS nodes sampled at rate δt_0 during $0 \leq t \leq t_{\max}$.
42	OSTORU01 OSTORU02 :	If ITEMVA=1 and NONODS > 0, time series of depth-averaged velocity U at NONODS nodes at rate δt_0 during $0 \leq t \leq t_{\max}$.
43	OSTOUB01 OSTOUB02 :	If ITEMVA=1 and NONODS > 0, time series of near-bottom horizontal velocity correction $\tilde{u}_b$ at NONODS nodes at rate δt_0 during $0 \leq t \leq t_{\max}$.

horizontal coordinate of node j given by XB(J) = $(j-1)\Delta x$; and ZB(J) = normalized vertical coordinate of the bottom at node j where ZB(J) is positive above SWL.

The file OSTAT is written in Subr. 21 DOC3 in the following manner:

```
      WRITE(23,9000) SMAX
      WRITE(23,8001) (XB(J), J=1, SMAX)
      WRITE(23,8001) (ZB(J), J=1, SMAX)
      WRITE(23,8001) (EMAX(J), J=1, SMAX)
      WRITE(23,8001) (EMIN(J), J=1, SMAX)
      WRITE(23,8001) (EMEAN(J), J=1, SMAX)
      WRITE(23,8001) (ERMS(J), J=1, SMAX)
      WRITE(23,8001) (UMAX(J), J=1, SMAX)
      WRITE(23,8001) (UMIN(J), J=1, SMAX)
      WRITE(23,8001) (UMEAN(J), J=1, SMAX)
      WRITE(23,8001) (URMS(J), J=1, SMAX)
      WRITE(23,8001) (UBMAX(J), J=1, SMAX)
      WRITE(23,8001) (UBMIN(J), J=1, SMAX)
      WRITE(23,8001) (UBMEAN(J), J=1, SMAX)
      WRITE(23,8001) (UBRMS(J), J=1, SMAX)
      WRITE(23,8001) (QMEAN(J), J=1, SMAX)
9000  FORMAT(I8)
8001  FORMAT (5F15.6)
```

where SMAX = largest node number reached by the computational shoreline; XB(J) = normalized horizontal coordinate of node j added for plotting convenience; ZB(J) = normalized bottom elevation added for plotting convenience; E indicates the normalized free surface elevation η above SWL; U denotes the normalized depth-averaged horizontal velocity U; UB indicates the normalized near-bottom horizontal velocity u_b; Q denotes the normalized volume flux per unit width, q; MAX implies the maximum value; MIN indicates the minimum value; MEAN denotes the mean value; and RMS implies the root-mean-square value defined by (69). The statistical calculations are performed during $t_{\text{stat}} \leq t \leq t_{\max}$ as explained in Section 3.4.

In the following, the quantities written in the file ODOC are explained without the corresponding FORMAT statements for brevity. The contents of this file with the FORMAT statements is given in the output example in Section 5.3.

Before the time-marching computation, the following quantities are written in sequence in Surb. 19 DOC1:

```
      IF (IWAVE.EQ.1) THEN
         IF (MCNO.EQ.O.D+00) THEN
            WRITE(28,2812) KS
         ELSE
            WRITE(28,2813) KS, KC2, ECNO, KCNO
         ENDIF
      ENDIF
```

where MCNO=0 indicates the use of Stokes second-order wave theory in Section 3.3.1; KS = normalized incident regular wave height; KC2 = $(1-m)$ with m computed using (67) for cnoidal wave theory; ECNO = complete elliptic integral E of the second kind; and KCNO = complete elliptic integral K of the first kind.

```
WRITE(28,2816) TREF, HREF, UL, DSEAP, UL
WRITE(28,2817) DSEA, INCLCT, WL, SIGMA, UR, KSI
WRITE(28,2818) DELTI
```

in which TREF = reference wave period $T'(s)$; HREF = reference wave height H' (m or ft); UL = m or ft; DSEAP = water depth d'_t (m or ft) below SWL at the seaward boundary $x = 0$; DSEA = $d_t = d'_t/H'$; INCLCT=0 [$C_t = 0$ in (75)] or 1 [C_t computed using (77)]; WL = L'/d'_t with L' = linear wavelength at $x = 0$ (unless cnoidal wave theory is used); SIGMA = $\sigma = T'\sqrt{g/H'}$ where $\sigma^2 \gg 1$ is assumed; UR = Ursell parameter U_r used in Section 3.3.1; KSI = surf similarity parameter $\xi = \sigma \tan\theta'_\xi/\sqrt{2\pi}$ with the slope $\tan\theta'_\xi$ = SLSURF specified as input; and DELTI = normalized sampling rate δt_i given by (68) for the input wave train.

```
WRITE(28,2819) APROFL, CMIXL, C2, C3, CB, CBL
```

where APROFL = cubic velocity profile parameter a in (40); CMIXL = mixing length parameter C_ℓ in (35); C2 = C_2 in (37); C3 = C_3 in (38); CB = C_B in (39); and CBL = $C_{B\ell} = C_B C_\ell^2 \sigma$. C_2, C_3 and C_B are computed using (41), (42) and (43), respectively.

```
WRITE(28,2821) WTOT, NBSEG
IF(IBOT.EQ.1) THEN
    WRITE(28,2824) (K, WBSEG(K), TBSLOP(K), BFFSEG(K), K=1, NBSEG)
ELSE
    WRITE(28,2825) XBSEG(1), ZBSEG(1)
    WRITE(28,2824) (K-1, XBSEG(K), ZBSEG(K), BFFSEG(K-1), K=2, NBSEG+1)
ENDIF
```

where WTOT = normalized horizontal width of the computation domain; and NBSEG = number of linear bottom segments specified as input for the dimensional bottom geometry starting from XBSEG(1) = $x' = 0$ and ZBSEG(1) = $-d'_t$ (m or ft). For linear segment K, WBSEG(K) = width (m or ft); TBSLOP(K) = slope; BFFSEG(K) = wave friction factor f'_w in (28); XBSEG(K+1) = horizontal distance (m or ft) from $x' = 0$ of its landward end; and ZBSEG(K+1) = vertical distance (m or ft) above SWL of its landward end.

```
WRITE(28,2841) DX, DELTA, COURNO, DKAPPA
WRITE(28,2842) TMAX, TSTAT, JMAX
WRITE(28,2843) STILL
IF(ITEMVA.EQ.1) THEN
WRITE(28,2844) DELTO
IF(NDELR.GT.0) WRITE(28,2845) NDELR
ENDIF
IF(ITEMVA.EQ.1.AND.NONODS.GT.0) WRITE(28,2846) NONODS
IF(ISPAEU.EQ.1) WRITE(28,2847) NOTIML
```

in which DX = normalized nodal interval Δx; DELTA = normalized water depth δ used to define the computational shoreline location; COURNO = Courant number C_n in (57) where $C_n \leq 1$ for numerical stability; DKAPPA = numerical damping coefficient κ in (59); TMAX = normalized computation duration $t_{\max}$; TSTAT = starting time t_{stat} for the statistical calculations; JMAX = maximum nodal number in the computation domain; STILL = location of the wet node next to the still water shoreline at $t = 0$; DELTO = normalized sampling rate, $\delta t_0 = t_{\max}/(\text{NPOUT} - 1)$, of storing time series where $t_{\max}$ and NPOUT are specified as input; NDELR = number of water depths δ_r' for computing the normalized shoreline elevation Z_r above SWL; NONODS = number of nodes for storing the time series of η, U and $\tilde{u}_b$; and NOTIML = number of time levels for storing the spacial variations of η, U and $\tilde{u}_b$.

After the time-marching computation, the following quantities are written in sequence in Surb. 21 DOC3:

```
WRITE(28,2811) DTMAX, DTMIN
```

in which DTMAX and DTMIN are the maximum and minimum values of the time step Δt used during the time-marching computation.

```
WRITE(28,2810) REFCOE
WRITE(28,2813) EIMAX, EIMIN, EIMEAN, EIRMS
WRITE(28,2814) ERMAX, ERMIN, ERMEAN, ERRMS
```

where REFCOE = wave reflection coefficient r defined as (88); EIMAX, EIMIN, EIMEAN and EIRMS = maximum, minimum, mean and root-mean-square values of the normalized incident wave train $\eta_i(t)$ at $x = 0$, respectively; and ERMAX, ERMIN, ERMEAN and ERRMS = maximum, minimum, mean and rms values of the normalized reflected wave train $\eta_r(t)$ at $x = 0$, respectively.

```
    WRITE(28,2821) SMAX
    DO 110 L=1, NDELR
       WRITE(28,2823) L, DELRP(L), RZMAX(L), RZMIN(L), RZMEAN(L), RZRMS(L)
110 CONTINUE
```

where SMAX = largest node number reached by the computational shoreline; DELRP(L) = water depth δ_r' (cm or in) specified as input to compute the normalized elevation Z_r of the intersection between the free surface and a runup wire placed at the vertical distance δ_r' above and parallel to the bottom; and RZMAX(L), RZMIN(L), RZMEAN(L) and RZRMS(L) = maximum (runup), minimum (run-down), mean (setup) and rms values of Z_r for the specified δ_r', respectively.

4.8.2 CONDITIONAL OUTPUT

If ITEMVA=1, the computed time series of certain variables are stored at the time $t = \text{TIMOUT}(n) = (n-1)\delta t_0$ with $n = 1, 2, \cdots$, NPOUT. The initial values of these variables at $t = 0$ with $n = 1$ are stored in Surb. 19 DOC1 before the time-marching computation. When TIME < TIMOUT(n) $\leq$ TIMEST with $n = 2, 3, \cdots$, NPOUT in Main Program VBREAK during the time-marching computation, Subr. 20 DOC2 is called to interpolate the values of these variables at the present time TIME and at the next time TIMEST and store the interpolated values at the time TIMOUT(n) in the same format as in Subr. 19 DOC1. In the following, only the WRITE statements in Subr. 20 DOC2 are presented for brevity.

The file OIRWAV stores the time series of the incident wave profile $\eta_i(t)$ and the reflected wave profile $\eta_r(t)$ at $x = 0$ if ITEMVA=1.

```
      WRITE(30,8001) EIINT, ERINT, TIMINT
8001  FORMAT(5F15.6)
```

where EIINT and ERINT are the interpolated values of η_i and η_r at the time TIMINT = TIMOUT(n).

The file ORUNUP stores the time series of the shoreline elevation Z_r for the specified NDELR values of the water depth δ'_r if NDELR > 0 and ITEMVA=1.

```
      WRITE(31,8001) (DUMR(L), L=1, NDELR), TIMINT
8001  FORMAT(5F15.6)
```

where DUMR(L) is the interpolated value of Z_r at the time TIMINT = TIMOUT(n).

If IENERG=1, the file OENERG stores the spacial variations of the time-averaged wave energy quantities in Subr. 21 DOC3.

```
      WRITE(35,9000) SMAX
      WRITE(35,8001) (XB(J), J=1, SMAX)
      WRITE(35,8001) (ESMEAN(J), J=1, SMAX)
      WRITE(35,8001) (EFMEAN(J), J=1, SMAX)
      WRITE(35,8001) (DFMEAN(J), J=1, SMAX)
      WRITE(35,8001) (DBMEAN(J), J=1, SMAX)
      WRITE(35,8001) (DBMDIF(J), J=1, SMAX)
      WRITE(35,8001) (DELES(J), J=1, SMAX)
9000  FORMAT(I8)
8001  FORMAT(5F15.6)
```

where SMAX = largest shoreline node; XB(J) = normalized horizontal coordinate of node j added for plotting convenience; ESMEAN(J) = mean specific wave energy $\overline{E_j}$; EFMEAN(J) = mean energy flux $\overline{(E_F)_j}$; DFMEAN(J) = mean energy dissipation rate, $\overline{(D_f)_j}$, inside the wave boundary layer; DBMEAN(J) = mean energy dissipation rate, $\overline{(D_b)_j}$, outside the wave boundary layer due to wave breaking; DBMDIF(J) = difference between $\overline{(D_B)_j}$ computed using the time-averaged wave energy equation (46) and that based on (39), which may be regarded as numerical energy dissipation rate; and DELES(J) = quantity $(\Delta E)_j$ given by (47).

If ISPAEU=1, the spacial variation of the free surface elevation η, the depth-averaged velocity U, and the near-bottom horizontal velocity correction $\tilde{u}_b$ defined by (32) are stored at the specified time levels TIMSPA(I) with I = 1, 2, $\cdots$, NOTIML. When TIME $<$ TIMSPA(I) $\leq$ TIMEST in Main Program VBREAK during the time-marching computation, Subr. 20 DOC2 is called to interpolate the values of η, U and $\tilde{u}_b$ at the time = TIME and TIMEST and store the interpolated values at the time TIMINT = TIMSPA(I) as follows:

```
      WRITE(22,9000) JMAX
```

```
      WRITE(22,8002) (EINT(J), UINT(J), UBINT(J), J=1, JMAX), TIMINT
9000  FORMAT(I8)
8002  FORMAT(3F15.6)
```

where JMAX = largest node number in the computation domain; and EINT(J), UINT(J) and UBINT(J) = interpolated values of η, U and $\tilde{u}_b$ at node j.

4.8.3 Time Series at Specified Nodes

IF NONODS > 0 and ITEMVA=1, the time series of η, U and $\tilde{u}_b$ at the specified nodal locations NODLOC(I) with I = 1, 2, $\cdots$, NONODS are stored in a manner similar to the storage of the time series of η_i and η_r in the file OIRWAV. The initial values of these variables at TIME=0 are stored in Subr. 19 DOC1 as follows:

```
      WRITE(41,8001) (DUMZ(I), I=1, NONODS)
      WRITE(41,8001) (DUME(I), I=1, NONODS), TIME
      WRITE(42,8001) (DUMU(I), I=1, NONODS), TIME
      WRITE(43,8001) (DUMUB(I), I=1, NONODS), TIME
8001  FORMAT(5F15.6)
```

where DUMZ(I) = value of ZB(J) $= (z_b)_j$ at node J=NODLOC(I) added for convenience; DUME(I), DUMU(I) and DUMUB(I) = values of η, U and $\tilde{u}_b$ at node J=NODLOG(I) and at TIME=0. The water depth $h = (\eta - z_b)$ can be found from the stored η and z_b. The interpolated values of these variables at TIMINT = TIMOUT(n) with n = 2, 3, $\cdots$, NPOUT are stored in Subr. 20 DOC 2 as follows:

```
      WRITE(41,8001) (DUME(I), I=1, NONODS), TIMINT
      WRITE(42,8001) (DUMU(I), I=1, NONODS), TIMINT
      WRITE(43,8001) (DUMUB(I), I=1, NONODS), TIMINT
8001  FORMAT(5F15.6).
```

where DUME(I), DUMU(I) and DUMUB(I) are the interpolated values of η, U and $\tilde{u}_b$ at node J = NODLOC(I) and at the time TIMINT.

The time series of η, U and $\tilde{u}_b$ at the NONODS nodes are stored in groups of 100 reference wave periods to avoid creating too large output files. The output files are named as follows:

- OSTOREXX for η
- OSTORUXX for U
- OSTOUBXX for $\tilde{u}_b$

where XX = 01 for the file containing the first 100 waves, XX = 02 for the file containing the second 100, and so on.

PART V
BREAKING WAVES ON GENTLE SLOPES

• 5.1 •
COMPARISONS WITH REGULAR WAVE DATA

The numerical model VBREAK has been compared with two data sets of regular waves spilling on gentle uniform slopes. One data set is the comprehensive measurements of test 1 presented by Stive (1980) and Stive and Wind (1982). The other data set is the detailed velocity, bottom shear stress and free surface measurements by Cox et al. (1995). The compared results are presented in the separate report by Johnson et al. (1995). Since the numerical model VBREAK predicts the vertical variations of the horizontal and vertical velocities, the comparisons of the measured and computed velocities can be made without any ambiguity. The previous one-dimensional models such as RBREAK2 (Kobayashi and Poff 1994) predict only the depth-averaged velocity that was assumed to represent the horizontal velocity measured at a certain elevation.

The input and output used for the comparison of VBREAK with Stive's test 1 are presented as an example in the following. Kobayashi et al. (1989) already presented the comparison of the one-dimensional model IBREAK with test 1.

• 5.2 •
EXAMPLE OF INPUT

In Stive's test 1, the incident regular waves with the period $T_i' = 1.79$ s broke as spilling breakers on the 1:40 concrete beach. The seaward boundary for the computation is taken to be at the still water depth $d_t' = 0.2375$ m, where the near-breaking wave profile was shown to be similar to the cnoidal wave profile. The measured wave height at the seaward boundary was H_i' = 0.172 m.

Table 4 lists the primary input file `FINP1` prepared for the computation of Stive's test 1 following the `READ` statements explained in Section 4.6. The input parameters and variables are as follows:

- `NLINES`=3: for the three comment lines listed below the numeral 3.
- `ISYST`=1: for the SI units (m and cm).
- `IWAVE`=1: for the incident regular wave profile $\eta_i(t)$ at the seaward boundary computed by VBREAK .

- IBOT=1: for the input of the width and slope of the 1:40 uniform slope.
- INCLCT=1: to include the nonlinear correction term C_t in (75) in calculating the reflected wave profile $\eta_r(t)$ at the seaward boundary where this term improves the prediction of wave set-down and setup on the gentle slope.
- IENERG=1: for computing the quantities related to wave energy.
- ITEMVA=1: for storing the computed time series.
- ISPAEU=1: for storing the spatial variations of the free surface elevation η, the depth-averaged horizontal velocity U, and the near-bottom horizontal velocity correction $\tilde{u}_b$.
- TSTAT=29.0: for the statistical calculations starting from the normalized time $t = t_{\text{stat}} = 29$ as explained in Section 3.4.
- TMAX=30.0: for the computation duration $t_{\text{max}} = 30$ where the periodicity of the computed time-varying quantities has been checked by Johnson et al. (1995).
- STILL=190: for 190 nodal intervals between the seaward boundary $x' = 0$ at $d'_t = 0.2375$ m and the still water shoreline on the 1:40 slope where 190 nodal intervals over the horizontal distance of 40 $d'_t = 9.5$ m yields the nodal interval $\Delta x' = 0.05$ m.
- DELTA=0.001: for the normalized water depth $\delta = 0.001$ used to define the computational shoreline.
- COURNO=0.4: for the Courant number $C_n = 0.4$ in (57) used to calculate the time step size Δt where $C_n = 0.1$–0.9 has been used, and the decrease of C_n increases Δt and improves the numerical stability.
- DKAPPA=1.0: for the numerical damping coefficient $\kappa = 1.0$ in (59) where the increase of κ increases the damping of high-frequency numerical oscillations at the rear of the breaking wave front with negligible changes for the rest of the wave motion.
- HREF=0.172: for the reference wave height H' taken to be the incident wave height $H'_i = 0.172$ m at the seaward boundary.
- TREF=1.79: for the reference wave period T' taken to be the incident wave period $T'_i = 1.79$ s.
- KS=1.0: for the normalized incident wave height KS $= H'_i/H' = 1.0$.
- NPINP=9001: for the small sampling rate $\delta t_i = t_{\text{max}}/(\text{NPINP} - 1) = 1/300$ used to resolve the incident regular wave train sufficiently where $(\delta t_i)^{-1}$ must be an even number for IWAVE=1.
- APROFL=3.0: for the cubic profile parameter $a = 3.0$ in (40) where the computed velocity profile is found to be insensitive to a in the range $3.0 \leq a \leq 4.0$.
- CMIXL=0.1: for the mixing length parameter $C_\ell = 0.1$ in (35) which may be regarded as a typical value.

Table 4: Primary input data file FINP1 for Stive's test 1.

```
  3                    = number of comment lines
-----------------------------------------------------------
Stive 1980 Test 1
-----------------------------------------------------------
1                                            <--ISYST
1                                            <--IWAVE
1                                            <--IBOT
1                                            <--INCLCT
1                                            <--IENERG
1                                            <--ITEMVA
1                                            <--ISPAEU
29.             30.                          <--TSTAT,TMAX
190                                          <--STILL
0.001           0.400           1.0          <--DELTA,COURNO,DKAPPA
.172            1.79                         <--HREF,TREF
1.0                                          <--KS
9001                                         <--NPINP
3.0              .10                         <--APROFL,CMIXL
.2375            0.025                       <--DSEAP,SLSURF
1                                            <--NBSEG
  18.000        .025        .05              <--WBSEG(1),TBSLOP(1),BFFSEG(1)
3001                                         <--NPOUT
1                                            <--NDELR
1.                                           <--DELRP(1)
6                                            <--NONODS
1     41    61    81    101                  <--NODLOC(1,2,3,4,5)
141                                          <--NODLOC(6)
5                                            <--NOTIML
29.0       29.25       29.50       29.75       30.00       <--TIMSPA(1,2,3,4,5)
```

- DSEAP=0.2375: for the water depth below SWL, $d_t' = 0.2375$ m, at the seaward boundary.
- SLSURF=0.025: for the slope $\tan\theta_\xi' = 1/40$ used to calculate the surf similarity parameter $\xi = \tan\theta_\xi'/(H'/L_o')^{1/2}$ with $L_o' = gT'^2/2\pi$.
- NBSEG=1: for the smooth uniform slope consisting of one linear segment of constant inclination and roughness.
- WBSEG(1)=18.0: for the horizontal width of the uniform slope taken to be 18 m, which is wide enough to avoid wave overtopping.
- TBSLOP(1)=0.025: for the 1:40 concrete slope in Stive's test 1.
- BFFSEG(1)=0.05: for the wave friction factor $f_w' = 0.05$ associated with the smooth concrete slope where $f_w' = 0.05$ was used in the previous one-dimensional computation by Kobayashi et al. (1989).
- NPOUT=3001: for the sampling rate $\delta t_o = t_{\max}/(\mathtt{NPOUT} - 1) = 1/100$ used to store the computed time series.
- NDELR=1: for one physical water depth δ_r' used to compute wave runup.
- DELRP(1)=1.0: for $\delta_r' = 1$ cm which is the vertical distance of a hypothetical runup wire placed above and parallel to the 1:40 slope.
- NONODS=6: for six nodal locations where the time series of η, U and $\tilde{u}_b$ are stored.
- NODLOC(1)=1: for the node located at $x' = 0$ m where the nodal interval $\Delta x' = 0.05$ m.
- NODLOC(2)=41: for the node located at $x' = (41\text{–}1)\ \Delta x' = 2$m.
- NODLOC(3)=61: for the node located at $x' = (61\text{–}1)\ \Delta x' = 3$m.
- NODLOC(4)=81: for the node located at $x' = (81\text{–}1)\ \Delta x' = 4$m.
- NODLOC(5)=101: for the node located at $x' = (101\text{–}1)\ \Delta x' = 5$m.
- NODLOC(6)=141: for the node located at $x' = (141\text{–}1)\ \Delta x' = 7$m.
- NOTIML=5 for five time levels used to store the spacial variations of η, U and $\tilde{u}_b$.
- TIMSPA(1)=29.0 to store these spacial variations at $t = 29.0$.
- TIMSPA(2)=29.25 to store these spacial variations at $t = 29.25$.
- TIMSPA(3)=29.50 to store these spacial variations at $t = 29.50$.
- TIMSPA(4)=29.75 to store these spacial variations at $t = 29.75$.
- TIMSPA(5)=30.0 to store these spacial variations at $t = 30.0$ which should be identical to those at $t = 29.0$ if the periodicity is established for $t \geq 29$.

It is noted that the input wave profile file FINP2 is not required for IWAVE=1 because the incident regular wave profile $\eta_i(t)$ is computed using cnoidal or Stokes second-order wave theory as explained in Section 3.3.1.

• 5.3 •

EXAMPLE OF OUTPUT

The output files produced by VBREAK have been explained in Section 4.8. For the primary input data file shown in Table 4, all the output files listed in Table 3 are produced by VBREAK . The concise output file ODOC for the computation made for Stive's test 1 is listed in Table 5 which should be self-explanatory with the aid of Section 4.8.1.

The computed results stored in the output files OSPACE, OSTAT, OIRWAV, ORUNUP and OENERG explained in Sections 4.8.1 and 4.8.2 can be opened for plotting appropriate figures as has been done by Johnson et al. (1995) although a user will need to write simple computer programs for plotting these figures.

The output files explained in Section 4.8.3 for the stored time series of η, U and $\tilde{u}_b$ at specified nodes (the six nodes at $x' = 0$, 2, 3, 4, 5 and 7 m for Stive's test 1) can be opened to plot the temporal variations of η, U and $\tilde{u}_b$ at these nodes as well as to compute the vertical variation of the horizontal velocity u using (33), (34) and (40) at given time t at each node. The computation of the vertical velocity w using (45) requires the values of $\partial U/\partial x$ and $\partial \tilde{u}_b/\partial x$ which may be approximated by appropriate finite differences. The time series of U and $\tilde{u}_b$ at the adjacent nodes involved in the finite difference approximations of $\partial U/\partial x$ and $\partial \tilde{u}_b/\partial x$ will need to be stored. For Stive's test 1, no vertical velocity data was available and the time series of U and $\tilde{u}_b$ at the nodes adjacent to the six nodes listed below NONODS=6 in Table 4 are not stored. On the other hand, the data set of Cox et al. (1995) included the measured vertical velocities that were compared with the computed vertical velocities by Johnson et al. (1995).

Table 5: Concise output from file ODOC for Stive's test 1.

```
-----------------------------------------------------------
Stive 1980 Test 1
-----------------------------------------------------------

WAVE CONDITION

Cnoidal Incident Wave at Seaward Boundary
Normalized wave height KS      =       1.000000
1-m =       0.113153458D-02
E   =       0.100242146D+01
K   =       0.477945412D+01
Reference Wave Period           =     1.790000 sec.
Reference Wave Height           =     0.172000 meters
Depth at Seaward Boundary       =     0.237500 meters
Norm. Depth at Seaw. Bdr.       =     1.381
Included Correction Term   CT
0 = no; 1 = yes        INCLCT   =     1
Normalized Wave Length          =    12.963
"Sigma"                         =    13.518
Ursell Number                   =   121.692
Surf Similarity Parameter       =     0.135
Input Wave Train from Time=0 to TMAX
Computed or Read at Normalized Rate DELTI =      0.003333

Parameters of Vertical Velocity Variations
Cubic Profile Parameter      APROFL =       3.000000
Mixing Length Parameter      CMIXL  =       0.100000
Momentum Flux Coefficient    C2     =       0.548214
Kinetic Energy Flux Coeff.   C3     =      -0.069420
Energy Dissipation Coeff.    CB     =      15.163393
Coefficient of DB            CBL    =       2.049839


BOTTOM GEOMETRY

Norm. Horiz. Length of
      Computation Domain    =      7.741422
Number of Segments          =         1
------------------------------------------
 SEGMENT    WBSEG(I)   TBSLOP(I)    BFFSEG(I)
      I       meters
------------------------------------------
      1    18.000000    0.025000     0.050000
```

Table 5: Continued.

```
COMPUTATION PARAMETERS

Normalized DX                                 =     0.215040D-01
Normalized DELTA                              =     0.100000E-02
Courant Number                                =     0.400
 Must not exceed unity
Numerical Damping Coefficient                 =     1.0000
 Must be zero or positive
Normalized Computation Duration  TMAX =     30.000000
Statistical Calculations Start
 when Time is equal to           TSTAT=     29.000000
Total Number of Spatial Nodes    JMAX =       361
Number of Nodes Along Bottom Below SWL
                                 STILL =      190
Storing Temporal Variations from Time = 0
 to TMAX at Normalized Rate      DELTO =     0.010000
Wave Runup Time Series Stored for
                                  NDER =   1 Water Depths
Time Series of ETA, U, and UB
          Stored at             NONODS =        6 Nodes
Spacial Variations of ETA, U, and UB
          Stored at             NOTIML =       5 Time Levels

Maximum time step        =     0.73200E-02
Minimum time step        =     0.27400E-02

REFLECTION COEFFICIENT

ETARRMS/ETAIRMS =       0.008

INCIDENT AND REFLECTED WAVES

             Max           Min           Mean          RMS
Inc.       0.7906       -0.2088        0.0000       0.3096
Ref.      -0.0354       -0.0427       -0.0388       0.0024

SHORELINE OSCILLATIONS

Largest Node Number Reached by Computational Shoreline
                                          SMAX =       204
-------------------------------------------------------------
      I   DELTAR(I)    RUNUP(I)  RUNDOWN(I)  SETUP(I) RMS(I)
            [cm]          Ru         Rd          Zr      Rrms
-------------------------------------------------------------
      1     1.000       0.074      0.047      0.059     0.009
```

PART VI
SUMMARY AND CONCLUSIONS

The numerical model VBREAK is developed to predict the cross- shore and temporal variations of the free surface elevation η, the depth-averaged horizontal velocity U, and the near-bottom horizontal velocity correction $\tilde{u}_b$ associated with the momentum flux correction m due to the vertical variation of the horizontal velocity u under the action of normally incident breaking waves. The three governing equations required for the computation of the three unknown variables are the depth-integrated continuity and horizontal momentum equations together with the new equation for the momentum flux correction m derived from the depth-integrated wave energy equation.

The normalized vertical profile of the horizontal velocity u outside the thin wave boundary layer is assumed to be cubic on the basis of limited available data. The turbulent shear stress outside the wave boundary layer is assumed to be expressed using the turbulent eddy viscosity whose mixing length is proportional to the instantaneous water depth. Although two additional empirical parameters are introduced in relation to these assumptions, the computed vertical profiles of the horizontal velocity are found to be fairly insensitive to these empirical parameters in their ranges expected from limited available data.

The computer program VBREAK is explained in detail so that a user will be able to modify and expand its first version. VBREAK has been compared with only two data sets for regular waves spilling on gentle uniform slopes. VBREAK may have to be modified for irregular waves and steeper coastal structures. The options of wave overtopping and transmission as well as armor stability and movement may be added by modifying the corresponding one-dimensional analyses included in RBREAK2 (Kobayashi and Poff 1994). VBREAK may also be combined with sediment transport analyses to predict cross-shore beach profile changes and toe scour in front of coastal structures.

The detailed and accurate measurements of the time-dependent, two-dimensional velocity fields under various breaking waves on different slopes will be required to calibrate and improve VBREAK. These measurements are very time-consuming and difficult especially near the free surface due to entrained air and near the bottom due to the thin wave boundary layer (Cox et al. 1995). Reversely, the calibrated and verified VBREAK or other numerical models may be used to estimate the quantities that can not be measured easily.

REFERENCES

Anderson, D.A., Tannehill, J.C., and Pletcher, R.H., 1984. *Computational fluid mechanics and heat transfer.* Hemisphere, New York, N.Y.

Battjes, J.A., 1974. Surf similarity. *Proc. 14th Coast. Engrg. Conf.*, ASCE, 466–480.

Chaudhry, M.H., 1993. *Open-channel flow.* Prentice Hall, Englewood Cliffs, N.J.

Cox, D.T., Kobayashi, N., and Okayasu, A., 1994. Vertical variations of fluid velocities and shear stress in surf zones. *Proc. 23rd Coast. Engrg. Conf.*, ASCE, 98–112.

Cox, D.T., Kobayashi, N., and Okayasu, A., 1995. Bottom shear stress in the surf zone. *J. Geophys. Res.* (accepted).

Fennema, R.J., and Chaudhry, M.H., 1986. Explicit numerical schemes for unsteady free-surface flows with shocks. *Water Resources Res.*, **22**(13), 1923–1930.

Gharangik, A.M., and Chaudhry, M.H., 1991. Numerical simulation of hydraulic jump. *J. Hydraulic Engrg.*, ASCE, **117**(9), 1195–1211.

Heitner, K.L., and Housner, G.W., 1970. Numerical model for tsunami run-up. *J. Wtrwy. Port, Coast. and Oc. Engrg.*, ASCE, **96**(3), 701–719.

Jameson, A., Schmidt, W., and Turkel, E., 1981. Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. *Proc. AIAA 14th Fluid and Plasma Dynamics Conf.*, Am. Inst. Aeronaut. and Astronaut., 81–1259.

Johnson, B.D., Kobayashi, N., and Cox, D.T., 1995. Comparisons of numerical model with measured velocity field and bottom shear stress. *Res. Rept. No. CACR-95-09*, Ctr. for Applied Coast. Res., Univ. of Delaware, Newark, Del.

Jonsson, I.G., 1966. Wave boundary layers and friction factors. *Proc. 10th Coast. Engrg. Conf.*, ASCE, **1**, 127–148.

Jonsson, I.G., and Carlsen, N.A., 1976. Experimental and theoretical investigations in an oscillatory turbulent boundary layer. *J. Hydraul. Res.*, **14**, 45–60.

Kobayashi, N., and Otta, A.K., 1987. Hydraulic stability analysis of armor units. *J. Wtrwy. Port, Coast. and Oc. Engrg.*, ASCE, **113**(2), 171–186.

Kobayashi, N., Otta, A.K., and Roy, I., 1987. Wave reflection and run-up on rough slopes. *J. Wtrwy. Port, Coast. and Oc. Engrg.*, ASCE, **113**(3), 282–298.

Kobayashi, N., DeSilva, G.S., and Watson, K.D., 1989. Wave transformation and swash oscillation on gentle and steep slopes. *J. Geophys. Res.*, **94**(C1), 951–966.

Kobayashi, N., and Wurjanto, A., 1989. Numerical model for design of impermeable coastal structures. *Res. Rept. No. CE-89-75*, Ctr. for Applied Coast. Res., Univ. of Delaware, Newark, Del.

Kobayashi, N., Cox, D.T., and Wurjanto, A., 1990. Irregular wave reflection and run-up on rough impermeable slopes. *J. Wtrwy. Port, Coast. and Oc. Engrg.*, ASCE, **116**(6), 708–726.

Kobayashi, N., and Wurjanto, A., 1992. Irregular wave setup and run-up on beaches. *J. Wtrwy. Port, Coast. and Oc. Engrg.*, ASCE, **118**(4), 368–386.

Kobayashi, N., and Raichle, A.W., 1994. Irregular wave overtopping of revetments in surf zones. *J. Wtrwy. Port, Coast. and Oc. Engrg.*, ASCE, **120**(1), 56–73.

Kobayashi, N., and Poff, M.T., 1994. Numerical model RBREAK2 for random waves on impermeable coastal structures and beaches. *Res. Rept. No. CACR-94-12*, Ctr. for Applied Coast. Res., Univ. of Delaware, Newark, Del.

Kobayashi, N., and Karjadi, E.A., 1994. Swash dynamics under obliquely incident waves. *Proc. 24th Coast. Engrg. Conf.*, ASCE, 2155–2169.

Kobayashi, N., Johnson, B.D., and Cox, D.T., 1995. Two-dimensional velocity field of spilling waves on gentle slope. *J. Wtrwy. Port, Coast. and Oc. Engrg.*, ASCE (submitted).

Kobayashi, N., and Karjadi, E.A., 1995. "Obliquely incident irregular waves in surf and swash zones. *J. Geophys. Res.* (submitted).

MacCormack, R.W., 1969. The effect of viscosity in hypervelocity impact cratering. *Paper 69-354*, Am. Inst. of Aeronaut. and Astronaut., New York.

Madsen, P.A., and Svendsen, I.A., 1983. Turbulent bores and hydraulic jumps. *J. Fluid Mech.*, **129**, 1–25.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., 1986. *Numerical recipes: The art of scientific computing.* Cambridge Univ. Press, Cambridge, U.K.

Raubenheimer, B., Guza, R.T., Elgar, S., and Kobayashi, N., 1995. Swash on a gently sloping beach. *J. Geophys. Res.*, **100**(C5), 8751–8760.

Rodi, W., 1980. Turbulence models and their application in hydraulics. *Intl. Assoc. Hydraul. Res.*, Delft, the Netherlands.

Schäffer, H.A., Deigaard, R., and Madsen, P., 1992. A two-dimensional surf zone model based on the Boussinesq equations. *Proc. 23rd Coast. Engrg. Conf.*, ASCE, **1**, 576–589.

Stive, M.J.F., 1980. Velocity and pressure field of spilling breakers. *Proc. 17th Coast. Engrg. Conf.*, ASCE, 547–566.

Stive, M.J.F., and Wind, H.G., 1982. A study of radiation stress and set-up in the nearshore region. *J. Coast. Engrg.*, **6**, 1–25.

Svendsen, I.A., and Madsen, P.A., 1984. A turbulent bore on a beach. *J. Fluid Mech.*, **148**, 73–96.

Tørum, A., 1994. Wave-induced forces on armor unit on berm breakwaters. *J. Wtrwy. Port, Coast. and Oc. Engrg.*, ASCE, **120**(3), 251–268.

Wurjanto, A., and Kobayashi, N., 1991. Numerical model for random waves on impermeable coastal structures and beaches. *Res. Rept. No. CACR-91-05*, Ctr. for Applied Coast. Res., Univ. of Delaware, Newark, Del.

Zelt, J.A., 1991. The run-up of nonbreaking and breaking solitary waves. *J. Coast. Engrg.*, **15**, 205–246.

Appendix A

Listing of

Computer Program VBREAK

```
C
C     ##     ##  #######   #######   ########   ######    ##     ##
C     ##     ##  ##    ##  ##    ##  ##        ##    ##   ##    ##
C     ##     ##  ##    ##  ##    ##  ##        ##    ##   ##   ##
C     ##     ##  #######   #######   #######   ########   ## ##
C     ##     ##  ##    ##  ##  ##    ##        ##    ##   ######
C      ##   ##   ##    ##  ##   ##   ##        ##    ##   ###  ##
C        ##      #######   ##    ##  ########  ##    ##   ##     ##
C
C          Numerical Simulation of Vertically Two-Dimensioinal
C             Waves on Impermeable Beaches and Breakwaters;
C
C             Nobuhisa Kobayashi and Bradley D. Johnson
C               Center for Applied Coastal Research
C            University of Delaware, Newark, Delaware 19716
C                            August, 1995
C
C #########################  GENERAL NOTES  ##########################
C
C  The purpose of each of 23 subroutines arranged in numerical order
C  is described in each subroutine and where it is called.
C
C  All COMMON statements appear in the Main Program. Description of
C  each COMMON statement is given only in Main Program.
C
C  DOUBLE PRECISION is used throughout the program.
C
C #00######################  MAIN PROGRAM  ###########################
C
C     Main program performs time-marching computation using
C     subroutines
C
      PROGRAM VBREAK
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      DOUBLE PRECISION KS,KSI
      DOUBLE PRECISION KCNO,MCNO,KC2
      CHARACTER*10 FINP1,FINP2
      INTEGER STILL,S,SST,SMAX
C
C ... COMMONs
C
C       Name    Contents
```

```
C     ----------------------------------------------------------------
C     /DIMENS/  The values of the "PARAMETER"s specified in Main.
C               Note: Most subroutines have their own PARAMETER state-
C                     ments. PARAMETER values specified in subroutines
C                     must be the same as their counterparts in Main Program.
C                     Subroutine 22 CHEPAR checks this requirement.
C     /CONSTA/  Basic constants
C     /ID/      Identifiers specifying user's options
C     /TLEVEL/  Time levels
C     /NODES/   Integers for spacial nodes
C     /GRID/    Grid size, time step, and related quantities
C     /CPARA/   Input computational parameters
C     /WAVREF/  Reference wave height and period
C     /VERPAR/  Parameters of vertical velocity variatioins
C     /WAVINP/  Normalized input wave train
C     /IRWAVE/  Incident and reflected wave trains
C     /WAVPAR/  Dimensionless wave parameters
C     /CNOWAV/  Cnoidal wave parameters (K, E, m and 1-m)
C     /BOTPAR/  Parameters related to bottom geometry
C     /BOTSEG/  Dimensional input bottom geometry
C     /BOTNOD/  Normalized botttom geometry at each node
C     /WRUNUP/  Quantities related to wave runup computation
C     /HQUETA/  Hydrodynamic quantities computed
C     /VERVAR/  Variables for vertical velocity variations
C     /TAUBFW/  Bottom shear stress and friction factor
C     /STOTEP/  Parameters for storing time series
C     /STONOD/  Nodes for storing time series of ETA, U, and UB
C     /STOSPA/  Time levels for storing spacial variations of ETA, U, and UB
C     /EISTAT/  Mean, rms, max, and min of ETAI
C     /ERSTAT/  Mean, rms, max, and min of ETAR
C     /RZSTAT/  Mean, rms, max, and min of RUNZ
C     /ETSTAT/  Mean, rms, max, and min of ETA
C     /USTAT/   Mean, rms, max, and min of U
C     /UBSTAT/  Mean, rms, max, and min of UUB = (U+UB)
C     /QSTAT/   Mean of Q
C     /WESTAT/  Mean wave energy quantities
C     /ENERG/   Quantities related to wave energy
C     /VECMAC/  Vectors used in MacCormack numerical method
C     /DOTMAC/  Hydrodynamic quantities used for MacCormack predictor
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV
      COMMON /ID/     ISYST,IWAVE,IBOT,INCLCT,IENERG,
     +                ITEMVA,ISPAEU
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
```

```
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /CPARA/  DELTA,COURNO,DKAPPA
      COMMON /WAVREF/ HREF,TREF,KS
      COMMON /VERPAR/ APROFL,CMIXL,C2,C3,CB,CBL
      COMMON /WAVINP/ DELTI,ETAINP(N2),NPINP
      COMMON /IRWAVE/ ETAI,ETAIST,ETAR,ETARST
      COMMON /WAVPAR/ SIGMA,WL,UR,KSI
      COMMON /CNOWAV/ KCNO,ECNO,MCNO,KC2
      COMMON /BOTPAR/ DSEAP,DSEA,SLSURF,WTOT
      COMMON /BOTSEG/ WBSEG(N4),TBSLOP(N4),XBSEG(N4),ZBSEG(N4),
     +                BFFSEG(N4),NBSEG
      COMMON /BOTNOD/ XB(N1),ZB(N1),THETA(N1)
      COMMON /WRUNUP/ DELRP(N3),DELTAR(N3),RUNZ(N3),RUNZST(N3),NDELR
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),QST(N1),
     +                UST(N1),ETAST(N1)
      COMMON /VERVAR/ FM(N1),UB(N1),FMST(N1),UBST(N1),FM3(N1),DB(N1)
      COMMON /TAUBFW/ UUB(N1),TAUB(N1),FW(N1)
      COMMON /STOTEP/ DELTO,TIMOUT(N2),NPOUT
      COMMON /STONOD/ NONODS,NODLOC(N5)
      COMMON /STOSPA/ TIMSPA(N5),NOTIML
      COMMON /EISTAT/ EIMEAN,EIRMS,EIMAX,EIMIN
      COMMON /ERSTAT/ ERMEAN,ERRMS,ERMAX,ERMIN,REFCOE
      COMMON /RZSTAT/ RZMEAN(N3),RZRMS(N3),RZMAX(N3),RZMIN(N3)
      COMMON /ETSTAT/ EMEAN(N1),ERMS(N1),EMAX(N1),EMIN(N1)
      COMMON /USTAT/  UMEAN(N1),URMS(N1),UMAX(N1),UMIN(N1)
      COMMON /UBSTAT/ UBMEAN(N1),UBRMS(N1),UBMAX(N1),UBMIN(N1)
      COMMON /QSTAT/  QMEAN(N1)
      COMMON /WESTAT/ ESMEAN(N1),EFMEAN(N1),DFMEAN(N1),DBMEAN(N1),
     +                DBMDIF(N1),DELES(N1)
      COMMON /ENERG/  ESPC(N1),EFLUX(N1),DISF(N1),DISB(N1)
      COMMON /VECMAC/ F2(N1),F3(N1),G2(N1),G3(N1)
      COMMON /DOTMAC/ HDOT(N1),QDOT(N1),UDOT(N1),FMDOT(N1),UBDOT(N1)
C
C
C ... VARIABLES ASSOCIATED WITH THE "PARAMETER"s
C
C     Variables specified in PARAMETER statement cannot be passed
C     through COMMON statement. The following dummy integers are
C     used in COMMON /DIMENS/.
C
      N1R = N1
      N2R = N2
      N3R = N3
```

```
      N4R = N4
      N5R = N5
C
C ... OPEN FILES AND READ DATA
C
C     First  call to Subr. 1 OPENER opens files unconditionally
C     Second call to Subr. 1 OPENER opens files conditionally
C     Subr. 2 INPUT1 reads primary input data
C     Subr. 3 INPUT2 reads input wave at seaward boundary if IWAVE >1
C
      WRITE (*,*) 'VBREAK reports progress on MREP waves'
      WRITE (*,*) 'Enter MREP (0 if no report on screen)'
      READ (*,*) MREP
      WRITE (*,*) 'Name of Primary Input-Data-File?'
      READ  (*,5000) FINP1
 5000 FORMAT (A10)
      CALL OPENER (1,0,FINP1,FINP2)
      CALL INPUT1 (FINP2)
      CALL OPENER (2,0,FINP1,FINP2)
      IF (IWAVE.GT.1) CALL INPUT2
C
C ... PREPARATIONS FOR TIME MARCHING COMPUTATION
C
C     Subr. 4 BOTTOM computes normalized structure geometry
C     Subr. 5 PARAM  calculates important parameters
C     Subr. 6 INIT  specifies initial conditions
C     Subr. 7 INCREG  computes incident periodic wave profile if IWAVE=1
C
      CALL BOTTOM
      CALL PARAM
      CALL INIT
      IF (IWAVE.EQ.1) CALL INCREG
C
C     Subr. 19 DOC1 documents input data and related parameters
C     at TIME = 0 as indicated below
C     Subr. 20 DOC2 is checked using ICALL=0 before computation
      CALL DOC2 (0,DUM)
C
C ---------------------- TIME-MARCHING COMPUTATION -----------------------
C
C     For known H(j), Q(j), U(j), ETA(j), FM(j), and UB(j) at
C     node j with j = 1,2,...,S at time level TIME compute
C     values of HST(j), QST(j), UST(j), EST(j), FMST(j), and
C     UBST(j) with j = 1,2,...,SST at next time level TIMEST.
```

```
C     Integer MOWAVE defined as (TIME).LT.(MOWAVE).LE.(TIME+1)
C     counts number of reference wave periods computed
C
      TIME = 0.D+00
      S = STILL
      MOWAVE = 0
 500  CONTINUE
      IF (TIME.GE.DBLE(MOWAVE)) THEN
      MOWAVE = MOWAVE + 1
      IF (ITEMVA.EQ.1.AND.NONODS.GT.0) CALL OPENER(3,MOWAVE,FINP1,FINP2)
      IF (TIME.EQ.0.D+00) CALL DOC1
      ENDIF
C
C ..... MARCH FROM TIME LEVEL TIME TO TIME LEVEL TIMEST
C
C       Subr. 11 COMPDT computes time step size DT = (TIMEST-TIME)
C         using numerical stability criterion for MacCormack method
C       Subr. 18 ENERGY is called with ICALL = 1 to initialize
C         statistical calculations for time-averaged energy equation.
C       Subr. 12 MARCH marches one time step from TIME to TIMEST
C         excluding landward and seaward boundaries
C       Landward B.C. is in Subr. 13 LANDBC
C       Seaward  B.C. is in Subr. 14 SEABC
C
        CALL COMPDT
C
        IF(IENERG.EQ.1) THEN
         IF(TIME.LE.TSTAT.AND.TSTAT.LT.TIMEST) CALL ENERGY(1)
        ENDIF
C
        CALL MARCH
        CALL LANDBC
        CALL SEABC
C       HST(j), UST(j), and FMST(j) are smoothed, and QST(j), UBST(j)
C       ETAST(j) are recomputed using smoothed HST(j), UST(j), and
C       FMST(j) in Subr. 15 SMOOTH.
C
        CALL SMOOTH
C
C ..... BOTTOM SHEAR  STRESS
C       Computed in Subr. 16 BSTRES
C
        CALL BSTRES
C
```

```
C ..... STATISTICS OF HYDRODYNAMIC QUANTITIES
C
C       Subr. 17 STATIS finds mean, root mean-square, max. and min.
C         values of ETAI,ETAR, and RUNZ(L) with L = 1,2,...,NDELR as
C         well as U(j), UUB(j), ETA(j), and Q(j) with j = 1,2...,JMAX
C         for duration of time=TSTAT to TMAX
        IF (TIMEST.GT.TSTAT) CALL STATIS
C
C ..... WAVE ENERGY FLUX AND DISSIPATION
C       computed in Subr. 18 ENERGY for duration of time = TSTAT to
C         TMAX for ICALL = 2
        IF (IENERG.EQ.1.AND.TSTAT.LT.TIMEST) CALL ENERGY(2)
C
C ..... DOCUMENTATION DURING TIME-MARCHING COMPUTATION
C       Subr. 20 DOC2 documents computed results at designated time
C       levels
C
C       Calling DOC2(1,...) is for storing spatial variations
C         ETA, U, and UB when ISPAEU = 1 and TIME.LT.TIMSPA(i).LE.TIMEST
C         with i = 1,2,...,NOTIML
C       Calling DOC2(2,...) is for storing temporal variations of these
C         three variables at specified nodes when ITEMVA=1 and
C         TIME.LT.TIMOUT(N).LE.TIMEST with N = 2, 3,...,NPOUT
C         where TIMOUT(1) = 0 in Subr. 02 INPUT1 and temporal
C         variations at time=0 have been stored in Subr. 19 DOC1.
C
        IF (ISPAEU.NE.1) GO TO 200
        IF (TIME.EQ.0.D+00) ICOUNT = 1
        IF (ICOUNT.GT.NOTIML) GO TO 200
        IF (TIME.LT.TIMSPA(ICOUNT).AND.TIMSPA(ICOUNT).LE.TIMEST) THEN
          CALL DOC2(1,TIMSPA(ICOUNT))
          ICOUNT = ICOUNT + 1
        ENDIF
C
 200    IF (ITEMVA.NE.1) GOTO 300
        IF (TIME.EQ.0.D+00) NCOUNT = 2
        IF (NCOUNT.GT.NPOUT) GOTO 300
        IF (TIME.LT.TIMOUT(NCOUNT).AND.TIMOUT(NCOUNT).LE.TIMEST) THEN
          CALL DOC2(2,TIMOUT(NCOUNT))
          NCOUNT = NCOUNT + 1
        ENDIF
C
C ..... HOW FAR THE COMPUTATION HAS BEEN
C
```

```
 300    IF (MREP.GT.0.AND.TIMEST.GE.DBLE(MOWAVE)) THEN
        IDUM = MOD(MOWAVE,MREP)
        IF (IDUM.EQ.0)  WRITE (*,*) ' Finished ',MOWAVE,' Wave Periods'
        ENDIF
C
C ..... IF TIMEST = TMAX, end of time-marching computation.  If
C        TIMEST.LT.TMAX, proceed to next time level.
C
        IF (TIMEST.EQ.TMAX) GO TO 600
        IF (TIMEST.LT.TMAX) THEN
          TIME = TIMEST
          S = SST
          ETAI = ETAIST
          ETAR = ETARST
          IF (NDELR.GT.0) THEN
            DO 610 L = 1,NDELR
            RUNZ(L) = RUNZST(L)
 610        CONTINUE
          ENDIF
          DO 620 J = 1,JMAX
          H(J) = HST(J)
          Q(J) = QST(J)
          U(J) = UST(J)
          ETA(J) = ETAST(J)
          FM(J) = FMST(J)
          UB(J) = UBST(J)
 620      CONTINUE
         GOTO 500
        ENDIF
C
C --------------------- END OF 500 CONTINUE -------------------------
C
C ... POST-LOOP DOCUMENTATION
C     Subr. 21 DOC3 documents results after time-marching
C     computation
C
 600  CALL DOC3
C
      STOP
      END
C
C -00------------------  END OF MAIN PROGRAM  ----------------------
C #01###################  SUBROUTINE OPENER  #######################
C
```

```
C     This subroutine opens all input and output files
C
      SUBROUTINE OPENER(ICALL,M,FINP1,FINP2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      CHARACTER*10 FINP1,FINP2,FSTORE(20),FSTORU(20),FSTOUB(20)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     ISYST,IWAVE,IBOT,INCLCT,IENERG,ITEMVA,
     +                ISPAEU
      COMMON /STONOD/ NONODS,NODLOC(N5)
      DATA FSTORE /
     1 'OSTORE01  ','OSTORE02  ','OSTORE03  ','OSTORE04  ',
     2 'OSTORE05  ','OSTORE06  ','OSTORE07  ','OSTORE08  ',
     3 'OSTORE09  ','OSTORE10  ','OSTORE11  ','OSTORE12  ',
     4 'OSTORE13  ','OSTORE14  ','OSTORE15  ','OSTORE16  ',
     5 'OSTORE17  ','OSTORE18  ','OSTORE19  ','OSTORE20  '/
      DATA FSTORU /
     1 'OSTORU01  ','OSTORU02  ','OSTORU03  ','OSTORU04  ',
     2 'OSTORU05  ','OSTORU06  ','OSTORU07  ','OSTORU08  ',
     3 'OSTORU09  ','OSTORU10  ','OSTORU11  ','OSTORU12  ',
     4 'OSTORU13  ','OSTORU14  ','OSTORU15  ','OSTORU16  ',
     5 'OSTORU17  ','OSTORU18  ','OSTORU19  ','OSTORU20  '/
      DATA FSTOUB /
     1 'OSTOUB01  ','OSTOUB02  ','OSTOUB03  ','OSTOUB04  ',
     2 'OSTOUB05  ','OSTOUB06  ','OSTOUB07  ','OSTOUB08  ',
     3 'OSTOUB09  ','OSTOUB10  ','OSTOUB11  ','OSTOUB12  ',
     4 'OSTOUB13  ','OSTOUB14  ','OSTOUB15  ','OSTOUB16  ',
     5 'OSTOUB17  ','OSTOUB18  ','OSTOUB19  ','OSTOUB20  '/
C
      IF (ICALL.EQ.1) THEN
C
C        Subr. 22 CHEPAR (k,i,Ni,NiR) checks Ni=NiR with i = 1,2,3,4 or 5
C        in Subr. k
C
         CALL CHEPAR (1,5,N5,N5R)
C
C ..... UNCONDITIONAL OPENINGS
C
C        Units 11-19 reserved for input data files
C        Units 21-29 reserved for unconditionally-opened files
C
C Unit  Filename   Purpose
C ----  --------   --------------------------------------------------
```

```
C   11   FINP1      Contains primary input data
C   22   OSPACE   . Unconditionally, stores normalized bottom geometry
C                   --> JMAX,(XB(J),ZB(J),J=1,JMAX)
C                 . Conditionally, i.e., if ISPAEU=1, stores spatial
C                   variations of ETA, U, and UB at designated time
C                   levels TIMSPA(i), i = 1, 2, ..., NOTIML
C   23   OSTAT      Stores spatial variation of mean, rms, max, and
C                   min values of ETA, U, UUB, and Q
C   28   ODOC       Stores essential output for concise documentation
C   29   OMSG       Stores messages written under special
C                   circumstances during computation
C
        OPEN (UNIT=11,FILE=FINP1,    STATUS='OLD',ACCESS='SEQUENTIAL')
        OPEN (UNIT=22,FILE='OSPACE', STATUS='NEW',ACCESS='SEQUENTIAL')
        OPEN (UNIT=23,FILE='OSTAT',  STATUS='NEW',ACCESS='SEQUENTIAL')
        OPEN (UNIT=28,FILE='ODOC',   STATUS='NEW',ACCESS='SEQUENTIAL')
        OPEN (UNIT=29,FILE='OMSG',   STATUS='NEW',ACCESS='SEQUENTIAL')
      ENDIF
C
      IF(ICALL.EQ.2) THEN
C
C ..... CONDITIONAL OPENINGS FOR ICALL = 2
C
C       Units 30-39 reserved for files containing hydrodynamic and
C                   energy quantities
C
C  Unit  Filename   Purpose
C  ----  --------   ------------------------------------------------
C   12   FINP2      Contains input data prescribing water surface
C                   elevations at seaward boundary if IWAVE=2 or 3
C   30   OIRWAV     Stores incident and reflected wave trains at
C                   seaward boundary at sampling rate DELTO starting
C                   from TIME = 0
C   31   ORUNUP     Stores shoreline node and runup elevations
C                   associated with (DELTAR(L),L=1,NDELR) at sampling
C                   rate DELTO starting from TIME = 0
C   35   OENERG     Stores time-averaged energy quantities if IENERG = 1
C
C   41   FSTORE     Store time series of normalized free surface
C                   elevation at specified nodes from TIME =0
C   42   FSTORU     Store time series of normalized depth-averaged
C                   velocity at specified nodes from TIME =0
C   43   FSTOUB     Store time series of near-bottom horizontal velocity
C                   correction UB at specified nodes from TIME =0
```

```
C
C ----- INPUT WAVE TRAIN AT SEAWARD BOUNDARY
C
        IF (IWAVE.GT.1) THEN
          OPEN (UNIT=12,FILE=FINP2,STATUS='OLD',ACCESS='SEQUENTIAL')
        ENDIF
C
C ----- INCIDENT & REFLECTED WAVES AND RUNUP
C
        IF (ITEMVA.EQ.1) THEN
          OPEN (UNIT=30,FILE='OIRWAV',STATUS='NEW',ACCESS='SEQUENTIAL')
          OPEN (UNIT=31,FILE='ORUNUP',STATUS='NEW',ACCESS='SEQUENTIAL')
        ENDIF
C
C ----- WAVE ENERGY
C
        IF (IENERG.EQ.1)
     +    OPEN (UNIT=35,FILE='OENERG',STATUS='NEW',ACCESS='SEQUENTIAL')
C
        ENDIF
C
        IF (ICALL.EQ.3) THEN
C ..... CONDITIONAL OPENINGS FOR ICALL = 3
C
C-------------------------------------------------------------------
C Time series of ETA, U, and UB at specified nodes are stored if
C ITEMVA = 1 and NONODS>0.  For computation with long duration,
C a single output file may be too large to store.  Therefore,
C time series are stored in groups of 100 reference wave periods,
C i.e., 100 wave periods to an output file.
C Time series of free surface elevation ETA and corresponding
C       bottom elevation ZB are stored in files 'OSTORE01'
C  (the first 100 waves), 'OSTORE02' (the second 100
C waves), and so on (under variable FSTORE and unit number 41).
C Time series of depth-averaged velocity U are stored in files
C 'OSTORU01' (the first 100 waves), 'OSTORU02' (the second 100
C waves), and so on (under variable FSTORU and unit number 42).
C       Time series of near-bottom horizontal velocity correction UB
C       are stored in the same manner under variable FSTOUB and unit
C       number 43.
C In the following, the opening and closing of applicable output
C files are performed every 100 reference wave periods.
C-------------------------------------------------------------------
        IDUM = MOD(M,100)
```

```
        IF (IDUM.EQ.1) THEN
         MPACK = M/100 + 1
         IF (M.GT.1) THEN
         CLOSE (41)
         CLOSE (42)
         CLOSE (43)
          ENDIF
       OPEN(UNIT=41,FILE=FSTORE(MPACK),STATUS='NEW',ACCESS='SEQUENTIAL')
       OPEN(UNIT=42,FILE=FSTORU(MPACK),STATUS='NEW',ACCESS='SEQUENTIAL')
       OPEN(UNIT=43,FILE=FSTOUB(MPACK),STATUS='NEW',ACCESS='SEQUENTIAL')
        ENDIF
       ENDIF
C
       RETURN
       END
C
C -01----------------- END OF SUBROUTINE OPENER --------------------
C #02##################### SUBROUTINE INPUT1 #######################
C
C     This subroutine reads data from primary input data file and
C     checks some of them
C
      SUBROUTINE INPUT1 (FINP2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      DOUBLE PRECISION KS,KSI
      CHARACTER*5 COMMEN(14)
      CHARACTER*10 FINP2
      INTEGER STILL,S,SST,SMAX
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV
      COMMON /ID/     ISYST,IWAVE,IBOT,INCLCT,IENERG,ITEMVA,ISPAEU
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /CPARA/  DELTA,COURNO,DKAPPA
      COMMON /WAVREF/ HREF,TREF,KS
      COMMON /VERPAR/ APROFL,CMIXL,C2,C3,CB,CBL
      COMMON /WAVINP/ DELTI,ETAINP(N2),NPINP
      COMMON /WAVPAR/ SIGMA,WL,UR,KSI
      COMMON /BOTPAR/ DSEAP,DSEA,SLSURF,WTOT
      COMMON /BOTSEG/ WBSEG(N4),TBSLOP(N4),XBSEG(N4),ZBSEG(N4),
     +                BFFSEG(N4),NBSEG
      COMMON /WRUNUP/ DELRP(N3),DELTAR(N3),RUNZ(N3),RUNZST(N3),NDELR
```

```
      COMMON /STOTEP/ DELTO,TIMOUT(N2),NPOUT
      COMMON /STONOD/ NONODS,NODLOC(N5)
      COMMON /STOSPA/ TIMSPA(N5),NOTIML
      DATA INDIC /0/
      CALL CHEPAR (2,2,N2,N2R)
      CALL CHEPAR (2,3,N3,N3R)
      CALL CHEPAR (2,4,N4,N4R)
      CALL CHEPAR (2,5,N5,N5R)
C
C ......... COMMENT LINES
C           NLINES = number of comment lines preceding input data
      READ (11,1110) NLINES
      DO 110 I = 1,NLINES
        READ  (11,1120) (COMMEN(J),J=1,14)
        WRITE (28,1120) (COMMEN(J),J=1,14)
        WRITE (29,1120) (COMMEN(J),J=1,14)
 110  CONTINUE
C
C ......... OPTIONS
C           ISYST =1: international System of Units (SI) is used
C                 =2: US Customary System of Units (USCS) is used
C           IWAVE =1: incident periodic waves at seaward boundary computed
C                 =2: incident waves at seaward boundary given as input
C                 =3: total waves at seaward boundary given as input
C           If IWAVE>1 --> Must specify FINP2 = name of input data
C                          file containing the input waves
C           IBOT  =1:  width and slope of linear bottom segment
C                 =2:  coordinates of linear bottom segment
C           INCLCT=0: no correction term in computing ETAR
C                 =1: correction term for ETAR recommended for
C                     regular and irregular waves on beaches for IWAVE<3.
C                     For IWAVE = 3, measured total waves include this
C                     correction.
C           IENERG=0: energy quantities NOT computed
C                 =1: energy quantities computed
C           ITEMVA=0: computed time series NOT stored
C                 =1: computed time series stored
C           ISPAEU=0: computed spatial variations NOT stored
C                 =1: computed spatial variations of ETA, U, and UB stored
      READ (11,1130) ISYST
      READ (11,1130) IWAVE
      READ (11,1130) IBOT
      IF(IWAVE.LE.2) THEN
      READ (11,1130) INCLCT
```

```
      ELSE
      INCLCT = 0
      ENDIF
      READ (11,1130) IENERG
      READ (11,1130) ITEMVA
      READ (11,1130) ISPAEU
      IF ( IWAVE.GT.1) READ (11,1140) FINP2
C
C ......... CHECK OPTIONS
C           Subr. 23 CHEOPT is to check if user's options are within
C           the ranges available or recommended
      CALL CHEOPT ( 1,INDIC,ISYST ,1,2)
      CALL CHEOPT ( 2,INDIC,IWAVE ,1,3)
      CALL CHEOPT ( 3,INDIC,IBOT  ,1,2)
      CALL CHEOPT ( 4,INDIC,INCLCT,0,1)
      CALL CHEOPT ( 5,INDIC,IENERG,0,1)
      CALL CHEOPT ( 6,INDIC,ITEMVA,0,1)
      CALL CHEOPT ( 7,INDIC,ISPAEU,0,1)
C
C ......... CONSTANTS
C           PI   = 3.141592...
C           GRAV = gravitational acceleration
C                  . in m/sec**2  if ISYST=1 (SI)
C                  . in ft/sec**2 if ISYST=2 (USCS)
      PI = 4.D+00*DATAN(1.D+00)
      IF (ISYST.EQ.1) THEN
        GRAV = 9.81D+00
      ELSE
        GRAV = 32.2D+00
      ENDIF
C
C ......... DATA RELATED TO NORMALIZED TIME LEVELS
C           TSTAT = starting time of statistical calculations of mean,
C                   root-mean-square, maximum and minimum values
C           TMAX  = computation duration starting from TIME = 0
      READ (11,1150) TSTAT,TMAX
C
C ......... COMPUTATIONAL INPUT DATA
C           STILL = number of spatial nodes along the bottom below
C                   SWL used to determine nodal spacing DX for
C                   given bottom geometry.
C           Note :  STILL should be so large that delta x between two
C                   adjacent nodes is sufficiently small.
C                   STILL = 100 to 600 has been used.
```

```
C           DELTA = normalized water depth defining computational
C                   shoreline
C           COURNO= Courant number, less than or equal to unity, for
C                   stable MacCormack finite difference method.
C                   Decrease of COURNO reduces time step DT
C           DKAPPA= Numerical damping coefficient (zero for no
C                   smoothing and positive for smoothing computed
C                   H, U, and FM)
      READ (11,1110) STILL
      CALL CHEOPT(8,IDUM,STILL,100,N1-1)
      READ (11,1150) DELTA,COURNO,DKAPPA
C
C ......... INPUT WAVE PROPERTIES
C           HREF  = dimensional reference wave height
C                   . in meters if ISYST = 1 (SI)
C                   . in feet   if ISYST = 2 (USCS)
C           TREF  = dimensional reference wave period, in seconds
C                   HREF and TREF are used to normalize the governing
C                   equations
C           KS    = normalized incident regular wave height only for
C                   IWAVE = 1
C           NPINP = number of points in input wave train ETAINP during
C                   TIME = 0 to TMAX sampled at rate of
C                   DELTI = TMAX/(NPINP-1). For IWAVE = 1, ETAINP is
C                   computed in Subr.7 INCREG and (1/DELTI) must
C                   be an even number.  NPINP must be sufficiently
C                   large and the lower limit of 200 is
C                   set below.
C           SIGMA is ratio between horizontal and vertical length
C                   scales which is assumed to be large in shallow
C                   water
      READ (11,1150) HREF,TREF
      IF (IWAVE.EQ.1) READ (11,1150) KS
      IF (IWAVE.GT.1) KS = 1.D+00
      READ (11,1110) NPINP
      CALL CHEOPT(9,IDUM,NPINP,200,N2)
      DELTI = TMAX/DBLE(NPINP-1)
      SIGMA = TREF*DSQRT(GRAV/HREF)
C
C ......... INPUT VELOCITY PROFILE PARAMETERS
C           APROFL = parameter 'a' for assumed cubic velocity profile
C           CMIXL  = mixing length parameter
      READ (11,1150) APROFL,CMIXL
C
```

```
C ......... BOTTOM GEOMETRY
C           The bottom geometry is divided into segments of
C           different inclination and roughness starting from
C           seaward boundary
C           NBSEG  = number of segments
C           DSEAP  = dimensional water depth below SWL at seaward
C                    boundary(a positive number)
C           SLSURF = tangent of slope, used to define
C                    "surf similarity parameter"
C           For segments starting from the seaward boundary:
C             WBSEG(i)  = dimensional horizontal width of segment i
C             TBSLOP(i) = tangent of slope (+ upslope, - downslope)
C             BFFSEG(i) = bottom friction factor
C             XBSEG(i)  = dimensional horizontal distance from seaward
C                         boundary to landward-end of segment (i-1)
C             ZBSEG(i)  = dimensional vertical coordinate (+ above SWL)
C                         of the landward end of segment (i-1)
C             DSEAP, WBSEG, XBSEG, ZBSEG are in meters if ISYST = 1 (SI),
C                          in feet if ISYST = 2 (USCS)
      READ (11,1150) DSEAP,SLSURF
      READ (11,1110) NBSEG
      CALL CHEOPT (10,IDUM,NBSEG,1,N4-1)
      IF (IBOT.EQ.1) THEN
        DO 130 I = 1,NBSEG
          READ (11,1150) WBSEG(I),TBSLOP(I),BFFSEG(I)
  130   CONTINUE
      ELSE
        XBSEG(1) = 0.D+00
        ZBSEG(1) = -DSEAP
        DO 140 I = 2,NBSEG+1
          READ (11,1150) XBSEG(I), ZBSEG(I), BFFSEG(I-1)
  140   CONTINUE
      ENDIF
C
C     DSEA = normalized water depth below SWL at seaward boundary
      DSEA = DSEAP/HREF
C
C ..... STORAGE OF COMPUTED TIME SERIES
C       If ITEMVA = 0, computed time series are not stored and no
C         additional input is required.
C       If ITEMVA = 1, incident and reflected wave trains at seaward
C         boundary are stored during TIME = 0 to TMAX sampled at rate
C         of DELTO = TMAX/(NPOUT-1) with NPOUT being specified as
C         input.  The storage time levels are at TIMOUT(n) =
```

```
C          (n-1)*DELTO
      IF (ITEMVA.EQ.1) THEN
      READ (11,1110) NPOUT
      CALL CHEOPT(11,IDUM,NPOUT,2,N2)
      DELTO = TMAX/DBLE(NPOUT-1)
      DO 150 N=1,NPOUT
      TIMOUT(N)=DBLE(N-1)*DELTO
 150  CONTINUE
      ENDIF
C
C ..... If NDELR>0 in addition to ITEMVA=1, wave runup time series
C        corresponding to NDELR water depths are stored at sampling
C        rate of DELTO.  DELRP(L)= dimensional water depth associated
C        with measured or visual shoreline in centimeters if ISYST
C        = 1 (SI) and in inches if ISYST = 2 (USCS).  Corresponding
C        normalized depths are denoted by DELTAR(L) =
C        DELRP(L)/HREF(m or ft).
      IF (ITEMVA.EQ.1) THEN
       READ (11,1110) NDELR
      CALL CHEOPT(12,IDUM,NDELR,0,N3)
       IF (NDELR.GT.0) THEN
        DO 160 L = 1,NDELR
        READ (11,1150) DELRP(L)
        IF (ISYST.EQ.1) DELTAR(L) = DELRP(L)/(1.D+02 * HREF)
        IF (ISYST.EQ.2) DELTAR(L) = DELRP(L)/(12.D+00 * HREF)
 160    CONTINUE
       ENDIF
      ENDIF
C
C ..... If NONODS > 0 in addition to ITEMVA = 1, computed time series of
C        free surface elevation ETA, depth-averaged velocity U, and near-bottom
C        horizontal velocity correction UB are stored at sampling rate DELTO
C        at specified NONODS nodes NODLOC(I) with I = 1,2,....,NONODS.
      IF (ITEMVA.EQ.1) THEN
       READ (11,1110) NONODS
      CALL CHEOPT(13,IDUM,NONODS,0,N5)
       IF (NONODS.GT.0) READ(11,1160) (NODLOC(I),I = 1,NONODS)
      ENDIF
C
C ..... STORAGE OF SPATIAL VARIATIONS OF ETA , U, AND UB if ISPAEU = 1
C        Spacial variations of free surface elevation ETA , depth
C        averaged velocity U, and near-bottom horizontal velocity correction
C        UB are stored at specified time levels TIMSPA(I) with
C        I = 1,2,...,NOTIML  where NOTIML = number of time levels.
```

```
      IF (ISPAEU.EQ.1) THEN
       READ (11,1110) NOTIML
      CALL CHEOPT(14,IDUM,NOTIML,0,N5)
       IF (NOTIML.GT.0) READ(11,1170) (TIMSPA(I),I=1,NOTIML)
      ENDIF
C
      IF (INDIC.GT.0) STOP
      RETURN
C
C ... FORMATS
C
 1110 FORMAT (I8)
 1120 FORMAT (14A5)
 1130 FORMAT (I1)
 1140 FORMAT (A10)
 1150 FORMAT (3F13.6)
 1160 FORMAT (5I6)
 1170 FORMAT (5F12.5)
C
      END
C
C -02----------------- END OF SUBROUTINE INPUT1 -------------------
C #03#################### SUBROUTINE INPUT2 #######################
C
C     This subroutine reads input wave profile data at
C     seaward boundary if IWAVE = 2 or 3
C
      SUBROUTINE INPUT2
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /WAVINP/ DELTI,ETAINP(N2),NPINP
      CALL CHEPAR (3,2,N2,N2R)
C
C     ETA = given input free surface time series at seaward boundary
C     for IWAVE = 2 (incident wave) or for IWAVE = 3 (sum of incident
C     and reflected waves)
C
      READ (12,1210) (ETAINP(I),I=1,NPINP)
 1210 FORMAT (5D15.6)
C
      RETURN
      END
```

```
C
C -03----------------- END OF SUBROUTINE INPUT2 --------------------
C #04#################### SUBROUTINE BOTTOM ######################
C
C     This subroutine calculates normalized bottom geometry and
C     DX between two adjacent nodes
C
      SUBROUTINE BOTTOM
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      DOUBLE PRECISION KS,KSI
      DIMENSION TSLOPE(N1),BFFNOD(N1)
      INTEGER STILL,S,SST,SMAX
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV
      COMMON /ID/     ISYST,IWAVE,IBOT,INCLCT,IENERG,ITEMVA,ISPAEU
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /WAVREF/ HREF,TREF,KS
      COMMON /WAVPAR/ SIGMA,WL,UR,KSI
      COMMON /BOTPAR/ DSEAP,DSEA,SLSURF,WTOT
      COMMON /BOTSEG/ WBSEG(N4),TBSLOP(N4),XBSEG(N4),ZBSEG(N4),
     +                BFFSEG(N4),NBSEG
      COMMON /BOTNOD/ XB(N1),ZB(N1),THETA(N1)
      COMMON /TAUBFW/ UUB(N1),TAUB(N1),FW(N1)
      CALL CHEPAR (4,1,N1,N1R)
      CALL CHEPAR (4,4,N4,N4R)
C
C ... THE FOLLOWING VARIABLES ARE DIMENSIONAL
C
C     BSWL =  dimensional horizontal distance between
C             seaward boundary and initial shoreline at SWL
C     DSEAP = water depth below SWL at seaward boundary
C
C     The structure geometry is divided into segments of different
C     inclination and roughness
C     NBSEG = number of segments
C     For segments starting from the seaward boundary:
C       WBSEG(i)  = dimensional horizontal width of segment i
C       TBSLOP(i) = tangent of slope (+ upslope, - downslope)
C       BFFSEG(i) = bottom friction factor
C       XBSEG(i)  = dimensional horizontal distance from seaward boundary
C                   to the seaward-end of segment i
```

```
C       ZBSEG(i)  = dimensional vertical coordinate (+ above SWL)
C                   at the seaward-end of segment i
C     BSWL,DSEAP,WBSEG,XBSEG,ZBSEG are in meters if ISYST=1 (SI),
C                                        in feet   if ISYST=2 (USCS)
C
C ... COMPLETE SEGMENT DATA THAT IS  NOT SPECIFIED AS INPUT
C
      IF (IBOT.EQ.1) THEN
        DCUM     = 0.D+00
        XBSEG(1) = 0.D+00
        ZBSEG(1) = -DSEAP
        DO 110 K = 2,NBSEG+1
          DCUM     = DCUM + WBSEG(K-1)*TBSLOP(K-1)
          XBSEG(K) = XBSEG(K-1) + WBSEG(K-1)
          ZBSEG(K) = -DSEAP + DCUM
 110    CONTINUE
      ELSE
        DO 120 K = 1,NBSEG
          TBSLOP(K) = (ZBSEG(K+1)-ZBSEG(K))/(XBSEG(K+1)-XBSEG(K))
 120    CONTINUE
      ENDIF
C
C ... CALCULATE GRID SPACING DX BETWEEN TWO ADJACENT NODES
C     (dimensional)
C
C     The value of STILL specified as input corresponds to
C     number of nodes along the bottom below SWL
C
        K = 0
 900    CONTINUE
         IF (K.EQ.NBSEG) THEN
          WRITE(*,2900)
          WRITE(29,2900)
          STOP
         ENDIF
        K = K+1
        CROSS = ZBSEG(K)*ZBSEG(K+1)
        IF (CROSS.GT.0.D+00) GOTO 900
        BSWL = XBSEG(K+1) - ZBSEG(K+1)/TBSLOP(K)
        DX     = BSWL/DBLE(STILL)
 2900   FORMAT(/'Bottom is always below SWL.'/
     +          'There is no still water shoreline.')

C
```

```
C ... CALCULATE BOTTOM GEOMETRY AT EACH NODE (dimensional)
C
C     JMAX = landward edge node corresponding to maximum node number
C     ZB= vertical coordinate of bottom at node j (+ above SWL)
C                  (physical, later normalized under the same name)
C     TSLOPE(j) = tangent of local slope at node j
C
        DUM = XBSEG(NBSEG+1)/DX
        JMAX  = INT(DUM)+1
      IF (JMAX.GT.N1) THEN
        WRITE  (*,2910) JMAX,N1
        WRITE (29,2910) JMAX,N1
        STOP
      ENDIF
 2910 FORMAT (/' End Node =',I8,'; N1 =',I8/
     +          ' Bottom length is too long.'/
     +          ' Cut it, or change PARAMETER N1.')
C
      DIST = -DX
      K    = 1
      XCUM = XBSEG(K+1)
      DO 140 J = 1,JMAX
        DIST = DIST + DX
        IF (DIST.GT.XCUM.AND.K.LT.NBSEG) THEN
          K    = K+1
          XCUM = XBSEG(K+1)
        ENDIF
        ZB(J) = ZBSEG(K) + (DIST-XBSEG(K))*TBSLOP(K)
        TSLOPE(J) = TBSLOP(K)
        BFFNOD(J) = BFFSEG(K)
  140 CONTINUE
C
C ... NORMALIZATION BY HREF AND TREF
C
C     WTOT = normalized width of computation domain
C     At node j:
C       THETA(j)       = normalized tangent of local slope
C       FW(j)          = normalized bottom friction factor
C       (XB(j),ZB(j)) = normalized coordinates of bottom with ZB>0
C                        above SWL
C
      DUM  = TREF*DSQRT(GRAV*HREF)
      DX    = DX/DUM
      WTOT = DBLE(JMAX-1)*DX
```

```
      DO 150 J = 1,JMAX
        THETA(J) = TSLOPE(J)*SIGMA
        XB(J)    = DBLE(J-1)*DX
        ZB(J)    = ZB(J)/HREF
        FW(J)    = .5D+00*SIGMA*BFFNOD(J)
  150 CONTINUE
C
      RETURN
      END
C
C -04---------------- END OF SUBROUTINE BOTTOM ---------------------
C #05###################### SUBROUTINE PARAM ########################
C
C     This subroutine calculates parameters used in other subroutines
C
      SUBROUTINE PARAM
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      DOUBLE PRECISION KS,KSI
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV
      COMMON /WAVREF/ HREF,TREF,KS
      COMMON /VERPAR/ APROFL,CMIXL,C2,C3,CB,CBL
      COMMON /WAVPAR/ SIGMA,WL,UR,KSI
      COMMON /BOTPAR/ DSEAP,DSEA,SLSURF,WTOT
C
C
C ... PARAMETERS RELATED TO WAVE AND SLOPE CHARACTERISTICS
C
C     KSI          = surf similarity parameter
C     WLOP,WLO     = deep-water linear wavelengths, dimensional and
C                    normalized,respectively
C     DSEAP,DSEA   = water depths below SWL at seaward boundary,
C                    dimensional and normalized, respectively
C
      WLOP = GRAV*TREF*TREF/(2.D+00*PI)
      WLO = WLOP/DSEAP
      KSI = SIGMA*SLSURF/DSQRT(2.D+00*PI)
C
C ... LINEAR WAVELENGTH AND URSELL NUMBER
C
C     WL = normalized linear wavelength at seaward boundary
C     UR = Ursell number at seaward boundary based on linear wavelength
```

```
C
      TWOPI = 2.D+00*PI
      WL    = WLO
      FUN1  = WL - WLO*DTANH(TWOPI/WL)
 900  IF (DABS(FUN1).GT.1.D-04) THEN
        FUN2 = 1.D+00 + WLO*TWOPI/(WL*DCOSH(TWOPI/WL))**2
        WL   = WL - FUN1/FUN2
        FUN1 = WL - WLO*DTANH(TWOPI/WL)
      GOTO 900
      ENDIF
      UR = KS*WL*WL/DSEA
C
C     Note: WL and UR will be recomputed using cnoidal wave theory
C     if IWAVE = 1 and UR.GE.26
C
C ....PARAMETERS OF VERTICAL VELOCITY VARIATIONS
C     Cubic profile parameter APROFL and mixing length coefficient
C     CMIXL are read as input in Subr. 02 INPUT1
      A = APROFL
      B = -(3.D+00 + .75D+00 * A)
      AA = A*A
      AB = A*B
      BB = B*B
      AAA = AA*A
      AAB = AA*B
      ABB = A*BB
      BBB = BB*B
      C2 = 1.D+00 + 2.D+00 * B/3.D+00 + .5D+00 * A + .2D+00 * BB
      C2 = C2 + AB/3.D+00 + AA/7.D+00
      C3 = 1.D+00 + B + .75D+00 * A + .6D+00 * BB + AB
      C3 = C3 + (3.D+00*AA + BBB)/7.D+00 + .375D+00 * ABB
      C3 = C3 + AAB/3.D+00 + .1D+00 * AAA
      CB = -(2.D+00 * BBB + 7.2D+00 * ABB + 9.D+00 * AAB)
      CB = CB - 27.D+00 * AAA/7.D+00
      CBL = CB * SIGMA * CMIXL * CMIXL
C
      RETURN
      END
C
C -05----------------- END OF SUBROUTINE PARAM ---------------------
C #06#################### SUBROUTINE INIT #######################
C
C     This subroutine assigns initial values at TIME = 0 when no wave
C     action exists in computation domain
```

```
C
      SUBROUTINE INIT
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      INTEGER STILL,S,SST,SMAX
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /IRWAVE/ ETAI,ETAIST,ETAR,ETARST
      COMMON /BOTNOD/ XB(N1),ZB(N1),THETA(N1)
      COMMON /WRUNUP/ DELRP(N3),DELTAR(N3),RUNZ(N3),RUNZST(N3),NDELR
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),QST(N1),
     +                UST(N1),ETAST(N1)
      COMMON /VERVAR/ FM(N1),UB(N1),FMST(N1),UBST(N1),FM3(N1),DB(N1)
      COMMON /TAUBFW/ UUB(N1),TAUB(N1),FW(N1)
      CALL CHEPAR (6,1,N1,N1R)
      CALL CHEPAR (6,3,N3,N3R)
C
C     ZERO is used to initiate statistical calculations
      ZERO = 0.D+00
C
C .... INCIDENT AND REFLECTED WAVES
C
C      TIME = 0 is chosen at time when incident waves arrive at
C      seaward boundary
C      Incident (ETAI) and reflected (ETAR) wave trains
C      start from zero
       ETAI = ZERO
       ETAR = ZERO
C
C ... INSTANTANEOUS HYDRODYNAMIC QUANTITIES
C
C     Computed using variable time step size DT whose maximum and
C     minimum values for entire computation duration are indicated
C     by DTMAX and DTMIN
      DTMAX = ZERO
      DTMIN = 1.D+00
C     Hydrodynamic quantities at node j at time = TIME
C       H(j)  = total water depth
C       Q(j)  = volume flux
C       U(j)  = depth-averaged velocity given by U(j)=Q(j)/H(j)
C       ETA(j)= surface elevation above SWL given by ETA(j)=H(j)+ZB(j)
C
```

```
      DO 110 J = 1,JMAX
        Q(J) = ZERO
        U(J) = ZERO
        IF (J.LE.STILL) THEN
          H(J) = -ZB(J)
        ELSE
          H(J) = ZERO
        ENDIF
        ETA(J) = H(J)+ZB(J)
  110 CONTINUE
C
C .... VARIABLES FOR VERTICAL VELOCITY VARIATIONS
C       FM(j)  = momentum flux correction
C       UB(j)  = near-bottom horizontal velocity correction
C       FM3(j) = kinetic energy flux correction
C       DB(j)  = energy dissipation rate due to wave breaking
       DO 130 J = 1,JMAX
        FM(J) = ZERO
        UB(J) = ZERO
        FM3(J) = ZERO
        DB(J) = ZERO
 130   CONTINUE
C
C .... BOTTOM SHEAR STRESS
C       TAUB(j) = bottom shear stress
C       UUB(j)  = near-bottom horizontal fluid velocity defined
C                 as UUB(j) = U(j) + UB(j)
       DO 140 J = 1,JMAX
        TAUB(J) = ZERO
        UUB(J) = ZERO
 140   CONTINUE
C
C ... WAVE RUNUP
C
C     SMAX = largest node number reached by computational
C     waterline during entire computation duration
C     Mean, root-mean-square, maximum and minimum runup elevations
C     are computed during time = TSTAT to TMAX
C
      SMAX = STILL
      IF (NDELR.GT.0) THEN
      DO 150 L = 1,NDELR
       RUNZ(L) = ZERO
 150  CONTINUE
```

```
      ENDIF
C
      RETURN
      END
C
C -06------------------  END OF SUBROUTINE INIT  -------------------
C #07#####################  SUBROUTINE INCREG  #######################
C
C     This subroutine computes incident regular wave profile ETAINP(n)
C     with n = 1,2,...,NPINP at seaward boundary if IWAVE = 1
C     Wave Profile: Stokes II if UR<26
C                   Cnoidal otherwise
C
      SUBROUTINE INCREG
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      DOUBLE PRECISION K,M,KC2,KC,KS,KSI
      DIMENSION ETAU(N2)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /WAVREF/ HREF,TREF,KS
      COMMON /WAVINP/ DELTI,ETAINP(N2),NPINP
      COMMON /WAVPAR/ SIGMA,WL,UR,KSI
      COMMON /CNOWAV/ K,E,M,KC2
      COMMON /BOTPAR/ DSEAP,DSEA,SLSURF,WTOT
      CALL CHEPAR (7,2,N2,N2R)
C
C     Input wave train ETAINP(n) with n=1,2,...,NPINP(input) for incident
C     regular waves is computed at normalized rate DELTI for entire
C     compuation duration for TIME = 0 to TMAX(input) where DELTI =
C     TMAX/(NPINP-1). The input parameters NPINP and TMAX for IWAVE=1
C     read in Subr. 02 INPUT1 must be selected such that (1.0/DELTI) =
C     (NPINP-1)/TMAX = NONE is a sufficiently large even number.
C
      DUM = DBLE(NPINP-1)/TMAX
      NONE = INT(DUM)
      IDUM = MOD(NONE,2)
      IF(IDUM.NE.0) THEN
       WRITE(*,2910) NPINP,TMAX,NONE
       WRITE(29,2910) NPINP,TMAX,NONE
       STOP
      ENDIF
```

```
 2910 FORMAT(/'Number of input wave points NPINP = ',I8,
     +      /'Computation duration          TMAX = ',F13.6,
     +      /'              NONE = (NPINP-1)/TMAX = 'I8,
     +      /'NONE must be an even number for
     +        IWAVE = 1.'/'Change input value of NPINP or TMAX.')
C
C ... CONSTANTS AND PARAMETERS
C
      TWOPI  = 2.D+00*PI
      FOURPI = 4.D+00*PI
      HALFPI = PI/2.D+00
      NONE1  = NONE+1
      NHALF  = NONE/2
      NHALF1 = NHALF+1
      DSEAKS = DSEA/KS
C
C ... COMPUTE HALF OF WAVE PROFILE (unadjusted)
C
C     Normailized wave height and period of incident regular waves
C     are KS and unity, repectively
C     ETAU = surface elevation before adjustment of time shift TO
C     NO = approximate time level at which surface elevation is zero
C     UR based on linear wave theory is used in the following
C        criterion
C
      IF (UR.LT.26.) THEN
C
C ----- Stokes II Wave Profile indicated by M=0
C
        M = 0.D+00
        ARG  = TWOPI/WL
        ARG2 = 2.D+00*ARG
        DUM  = 16.D+00*DSEAKS*DSINH(ARG)**3.D+00
        AMP2 = ARG*DCOSH(ARG)*(2.D+00+DCOSH(ARG2))/DUM
         DO 110 N = 1,NHALF1
          T = DBLE(N-1)/DBLE(NONE)
          ETAU(N) = .5D+00*DCOS(TWOPI*T)+AMP2*DCOS(FOURPI*T)
          ETAU(N) = KS*ETAU(N)
          IF (N.GT.1) THEN
            IF(ETAU(N).LE.0.D+00.AND.ETAU(N-1).GT.0.D+00) NO=N
          ENDIF
  110   CONTINUE
C
      ELSE
```

```
C
C ----- Cnoidal Wave Profile
C
C        WL and UR are recalculated using cnoidal wave theory
C        FINDM is to find the parameter M of the Jacobian elliptic func.
C        See Func. 09 CEL and Subr. 10 SNCNDN
C
         CALL FINDM (DSEAKS,M)
         KC2 = 1.D+00-M
         KC  = DSQRT(KC2)
         K   = CEL(KC,1.D+00,1.D+00,1.D+00)
         E   = CEL(KC,1.D+00,1.D+00,KC2)
         UR  = 16.D+00*M*K*K/3.D+00
         WL  = DSQRT(UR*DSEAKS)
         ETAMIN = (1.D+00-E/K)/M - 1.D+00
         ETAMIN = KS*ETAMIN
         DO 120 N = 1,NHALF1
           T = DBLE(N-1)/DBLE(NONE)
           TETA = 2.D+00*K*T
           CALL SNCNDN (TETA,KC2,SNU,CNU,DNU)
           ETAU(N) = ETAMIN + KS*CNU*CNU
           IF (N.GT.1) THEN
             IF (ETAU(N).LE.0.D+00.AND.ETAU(N-1).GT.0.D+00) NO=N
           ENDIF
  120    CONTINUE
         ETAU(NHALF1) = ETAMIN
C
       ENDIF
C
C ... THE OTHER HALF OF WAVE PROFILE
C
       DO 130 N = NHALF+2,NONE1
         ETAU(N) = ETAU(NONE+2-N)
  130 CONTINUE
C
C ... ADJUST WAVE PROFILE FOR ONE WAVE PERIOD
C      so that elevation=0 at time=0 and decreases initially with time
C
C      ETAU = unadjusted surface elevation
C      ETAINP  = adjusted surface elevation for one wave period
C
       NMARK = NONE-NO+2
       DO 140 N = 1,NONE1
         IF (N.LE.NMARK) THEN
```

```
          ETAINP(N) = ETAU(N+NO-1)
        ELSE
          ETAINP(N) = ETAU(N-NMARK+1)
        ENDIF
  140 CONTINUE
C
C ... PERIODIC WAVE PROFILE FOR ENTIRE COMPUTATION DURATION
C     Note: ETAINP = 0 at TIME = 0
C
      DO 150 N = NONE1,NPINP
      ETAINP(N) = ETAINP(N-NONE)
 150  CONTINUE
      ETAINP(1) = 0.D+00
C
      RETURN
      END
C
C -07----------------- END OF SUBROUTINE INCREG --------------------
C #08##################### SUBROUTINE FINDM #######################
C
C     This subroutine computes the parameter M (MLIL<M<MBIG) of the
C     Jacobian elliptic functions
C
      SUBROUTINE FINDM (DSEAKS,M)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K,M,KC2,KC,MSAV,MLIL,MBIG
      DOUBLE PRECISION KSI
      COMMON /WAVPAR/  SIGMA,WL,UR,KSI
      COMMON /BOTPAR/  DSEAP,DSEA,SLSURF,WTOT
      DATA SMALL,MLIL /1.D-07,.8D+00/
      DATA INDI,I     /0,0/
      SIGDT = SIGMA/DSQRT(DSEA)
      MBIG  = 1.00D+00 - 1.00D-15
      M     =  .95D+00
  900 CONTINUE
        I    = I+1
        MSAV = M
        KC2  = 1.D+00-M
        KC   = DSQRT(KC2)
        K    = CEL(KC,1.D+00,1.D+00,1.D+00)
        E    = CEL(KC,1.D+00,1.D+00,KC2)
        UR   = 16.D+00*M*K*K/3.D+00
        WL   = DSQRT(UR*DSEAKS)
```

```
        F     = 1.D+00 + (-M+2.D+00-3.D+00*E/K)/(M*DSEAKS)
        F     = SIGDT*DSQRT(F)/WL - 1.D+00
        IF (F.LT.0.D+00) THEN
          MBIG = M
        ELSEIF (F.GT.0.D+00) THEN
          MLIL = M
        ELSE
          RETURN
        ENDIF
        M    = (MLIL+MBIG)/2.D+00
        DIF = DABS(MSAV-M)
        IF (DIF.LT.SMALL) RETURN
        IF (INDI.EQ.0) THEN
          IF (I.EQ.50) THEN
            SMALL = 1.D-13
            INDI  = 1
          ELSE
            IF (M.GT..9999D+00) THEN
              SMALL = 1.D-13
              INDI  = 1
            ENDIF
          ENDIF
        ENDIF
      IF (I.LT.100) GOTO 900
      WRITE  (*,2910)
      WRITE (29,2910)
 2910 FORMAT (/' From Subr. 9 FINDM:'/
     +          ' Criterion for parameter m = MCNO not satisfied')
C
      RETURN
      END
C
C -08------------------ END OF SUBROUTINE FINDM --------------------
C #09################# DOUBLE PRECISION FUNCTION CEL  #################
C
C     This function computes the general complete elliptic integral,
C     and is a double precision version of the "Function CEL" from
C     the book:
C       William H. Press, et. al.
C       Numerical Recipes: The Art of Scientific Computing.
C       Cambridge University Press, New York, 1986.
C       Pages 187-188.
C
      DOUBLE PRECISION FUNCTION CEL (QQC,PP,AA,BB)
```

```
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (CA=1.D-06,PIO2=1.5707963268D+00)
      IF (QQC.EQ.0.D+00) THEN
        WRITE  (*,*) 'Failure in Function CEL'
        WRITE (29,*) 'Failure in Function CEL'
        STOP
      ENDIF
      QC = DABS(QQC)
      A  = AA
      B  = BB
      P  = PP
      E  = QC
      EM = 1.D+00
      IF (P.GT.0.D+00) THEN
        P = DSQRT(P)
        B = B/P
      ELSE
        F = QC*QC
        Q = 1.D+00-F
        G = 1.D+00-P
        F = F-P
        Q = Q*(B-A*P)
        P = DSQRT(F/G)
        A = (A-B)/G
        B = -Q/(G*G*P)+A*P
      ENDIF
  900 F  = A
      A  = A+B/P
      G  = E/P
      B  = B+F*G
      B  = B+B
      P  = G+P
      G  = EM
      EM = QC+EM
      IF (DABS(G-QC).GT.G*CA) THEN
        QC = DSQRT(E)
        QC = QC+QC
        E  = QC*EM
        GOTO 900
      ENDIF
      CEL = PIO2*(B+A*EM)/(EM*(EM+P))
C
      RETURN
```

```
      END
C
C -09-------------  END OF DOUBLE PRECISION FUNCTION CEL  ------------
C #10######################  SUBROUTINE SNCNDN  ######################
C
C     This subroutine computes the Jacobian elliptic functions,
C     and is a double precision version of the "Subroutine SNCNDN"
C     from the book:
C       William H. Press, et. al.
C       Numerical Recipes: The Art of Scientific Computing.
C       Cambridge University Press, New York, 1986.
C       Page 189.
C
      SUBROUTINE SNCNDN (UU,EMMC,SN,CN,DN)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (CA=1.D-06)
      DIMENSION EM(13),EN(13)
      LOGICAL BO
      EMC = EMMC
      U   = UU
      IF (EMC.NE.0.D+00) THEN
        BO = (EMC.LT.0.D+00)
        IF (BO) THEN
          D   = 1.D+00-EMC
          EMC = -EMC/D
          D   = DSQRT(D)
          U   = D*U
        ENDIF
        A  = 1.D+00
        DN = 1.D+00
        DO 110 I = 1,13
          L = I
          EM(I) = A
          EMC   = DSQRT(EMC)
          EN(I) = EMC
          C     = .5D+00*(A+EMC)
          IF (DABS(A-EMC).LE.CA*A) GOTO 910
          EMC = A*EMC
          A = C
  110   CONTINUE
  910   U  = C*U
        SN = DSIN(U)
        CN = DCOS(U)
```

```
          IF (SN.EQ.0.D+00) GOTO 920
          A = CN/SN
          C = A*C
          DO 120 II = L,1,-1
            B  = EM(II)
            A  = C*A
            C  = DN*C
            DN = (EN(II)+A)/(B+A)
            A  = C/B
  120     CONTINUE
          A = 1.D+00/DSQRT(C*C+1.D+00)
          IF (SN.LT.0.D+00) THEN
            SN = -A
          ELSE
            SN = A
          ENDIF
          CN = C*SN
  920     IF (BO) THEN
            A  = DN
            DN = CN
            CN = A
            SN = SN/D
          ENDIF
        ELSE
          CN = 1.D+00/DCOSH(U)
          DN = CN
          SN = DTANH(U)
        ENDIF
C
        RETURN
        END
C
C -10----------------- END OF SUBROUTINE SNCNDN --------------------
C #11##################### SUBROUTINE COMPDT ########################
C
C     This subroutine computes time step size DT on the basis of
C     numerical stability criterion for MacCormack method using H(j)
C     and U(j) at present TIME
C
      SUBROUTINE COMPDT
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      INTEGER STILL,S,SST,SMAX
```

```
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /CPARA/  DELTA,COURNO,DKAPPA
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),QST(N1),
     +                UST(N1),ETAST(N1)
      IF (TIME.EQ.0.D+00) CALL CHEPAR(11,1,N1,N1R)
C
      DUMAX = 0.D+00
      DO 100 J = 1,S
      IF(H(J).LT.0.D+00) THEN
       WRITE(*,2910) H(J),J,S,TIME
       WRITE(29,2910) H(J),J,S,TIME
      STOP
      ENDIF
      DUM = DABS(U(J))+DSQRT(H(J))
      IF(DUM.GT.DUMAX) DUMAX = DUM
  100 CONTINUE
      DT = COURNO*DX/DUMAX
C
C     Adjust DT so that TIMEST does not exceed the last time level
C     TMAX of computation.
C
      TIMEST = TIME+DT
      IF(TIMEST.GT.TMAX) THEN
       TIMEST = TMAX
       DT = TMAX-TIME
      ENDIF
      DTDX = DT/DX
      DXDT = DX/DT
C
C     Compute DTMAX and DTMIN during entire computation duration
      IF(DT.GT.DTMAX) DTMAX = DT
      IF(DT.LT.DTMIN) DTMIN = DT
C
 2910 FORMAT(/'From Subr. 11 COMPDT: Negative water depth =',
     +       D12.3/'J = ',I8,'; S = ',I8,';TIME = ',D12.3)
C
      RETURN
      END
C
C -11----------------- END OF SUBROUTINE COMPDT --------------------
C #12##################### SUBROUTINE MARCH ########################
```

```
C
C     This subroutine marches the computation from time level TIME
C     to next time level TIMEST excluding seaward and landward boundaries
C     which are treated separately
C
      SUBROUTINE MARCH
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      INTEGER STILL,S,SST,SMAX,SP1,SM1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /CPARA/  DELTA,COURNO,DKAPPA
      COMMON /VERPAR/ APROFL,CMIXL,C2,C3,CB,CBL
      COMMON /BOTNOD/ XB(N1),ZB(N1),THETA(N1)
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),QST(N1),
     +                UST(N1),ETAST(N1)
      COMMON /VERVAR/ FM(N1),UB(N1),FMST(N1),UBST(N1),FM3(N1),DB(N1)
      COMMON /TAUBFW/ UUB(N1),TAUB(N1),FW(N1)
      COMMON /VECMAC/ F2(N1),F3(N1),G2(N1),G3(N1)
      COMMON /DOTMAC/ HDOT(N1),QDOT(N1),UDOT(N1),FMDOT(N1),UBDOT(N1)
      IF (TIME.EQ.0.D+00) CALL CHEPAR (12,1,N1,N1R)
C
C     S = most landward node at present time level TIME.
C     The following values at node j are known at TIME
C     H(j)   = total water depth
C     Q(j)   = volume flux per unit width
C     U(j)   = depth-averaged velocity
C     ETA(j) = free surface elevation above SWL
C     FM(j)  = momentum flux correction
C     UB(j)  = near-bottom horizontal velocity correction
C     FM3(j) = kinetic energy flux correction
C     DB(j)  = energy dissipation rate due to wave breaking
C     TAUB(j)= bottom shear stress
C     The unknown quantities at next time level TIMEST are indicated
C     by additional letters ST(*)
      SP1 = S+1
      SM1 = S-1
C
C ... Predictor Step of MacCormack Method
C
      DO 100 J = 1,SP1
```

```
      F2(J) = Q(J)*U(J)+0.5D+00*H(J)*H(J)+FM(J)
      F3(J) = 3.D+00*FM(J)*U(J)+FM3(J)
      IF(J.EQ.SP1) GOTO 100
      G2(J) = THETA(J)*H(J)+TAUB(J)
      G3(J) = 2.D+00*(TAUB(J)*UB(J)+DB(J)-U(J)*(FM(J+1)-FM(J))/DX)
 100  CONTINUE
      DO 110 J = 1,S
      HDOT(J) = H(J)-DTDX*(Q(J+1)-Q(J))
      QDOT(J) = Q(J)-DTDX*(F2(J+1)-F2(J))-DT*G2(J)
      FMDOT(J) = FM(J)-DTDX*(F3(J+1)-F3(J))-DT*G3(J)
C     FMDOT must be positive and HDOT must not be less than DELTA
      IF(FMDOT(J).LT.0.D+00) FMDOT(J) = 0.D+00
      IF(HDOT(J).LT.DELTA) HDOT(J) = DELTA
 110  CONTINUE
C
C ... Compute Intermediate Variables with a Dot
C
      DO 115 J = 1,S
      UDOT(J) = QDOT(J)/HDOT(J)
      UBDOT(J) = DSQRT(FMDOT(J)/C2/HDOT(J))
      IF(UDOT(J).GE.0.D+00) UBDOT(J) = -UBDOT(J)
      FM3(J) = C3*HDOT(J)*UBDOT(J)**3
      UUB(J) = UDOT(J)+UBDOT(J)
      TAUB(J) = FW(J)*DABS(UUB(J))*UUB(J)
      DB(J) = CBL*DABS(UBDOT(J))**3
 115  CONTINUE
C
C ... Corrector Step of MacCormack Method
C
      DO 120 J = 1,S
      F2(J) = QDOT(J)*UDOT(J)+0.5D+00*HDOT(J)*HDOT(J)+FMDOT(J)
      F3(J) = 3.D+00*FMDOT(J)*UDOT(J)+FM3(J)
      IF(J.EQ.1) GOTO 120
      G2(J) = THETA(J)*HDOT(J)+TAUB(J)
      DUM = UDOT(J)*(FMDOT(J)-FMDOT(J-1))/DX
      G3(J) = 2.D+00*(TAUB(J)*UBDOT(J)+DB(J)-DUM)
 120  CONTINUE
      DO 130 JJ = 2,S
      J = (S+2)-JJ
      HDOT(J) = HDOT(J)-DTDX*(QDOT(J)-QDOT(J-1))
      QDOT(J) = QDOT(J)-DTDX*(F2(J)-F2(J-1))-DT*G2(J)
      FMDOT(J) = FMDOT(J)-DTDX*(F3(J)-F3(J-1))-DT*G3(J)
 130  CONTINUE
C
```

```
C ... HST(J), QST(J), UST(J), AND FMST(J) with J = 2,3,....,S at next
C     time level TIMEST
C
      DO 140 J = 2,S
      HST(J) = 0.5D+00*(H(J) + HDOT(J))
      QST(J) = 0.5D+00*(Q(J) + QDOT(J))
      UST(J) = QST(J)/HST(J)
      FMST(J) = .5D+00*(FM(J)+FMDOT(J))
 140  CONTINUE
C
C ... HST(1), QST(1), UST(1), and FMST(1) are computed in Subr. 14 SEABC
C ... HST(S), QST(S), and UST(S) are improved in Subr. 13 LANDBC
C ... HST(j), UST(j), and FMST(j) are smoothed and the rest of the variables
C       at time = TIMEST are computed in Subr. 15 SMOOTH
C
C ... ABORT COMPUTATION IF WATER DEPTH AT (S-1) <or= DELTA
C
      IF (HST(SM1).LE.DELTA) THEN
        WRITE  (*,2910) HST(SM1),DELTA,S,TIME
        WRITE (29,2910) HST(SM1),DELTA,S,TIME
        STOP
      ENDIF
 2910 FORMAT (/' From Subroutine 12 MARCH'/
     + ' Computed water depth HST(S-1) is less than or equal to DELTA'/
     + ' HST(S-1) =',D12.3/
     + ' DELTA    =',D12.3/
     + ' S = ' , I8/
     + ' TIME = ',D12.3/
     + ' Program Aborted')
C
      RETURN
      END
C
C -12----------------- END OF SUBROUTINE MARCH ---------------------
C #13#################### SUBROUTINE LANDBC ########################
C
C     This subroutine manages the computation for
C     landward boundary condition
C
      SUBROUTINE LANDBC
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      INTEGER STILL,S,SST,SMAX,SP1,SM1
```

```
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     ISYST,IWAVE,IBOT,INCLCT,IENERG,
     +                ITEMVA,ISPAEU
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /CPARA/  DELTA,COURNO,DKAPPA
      COMMON /BOTNOD/ XB(N1),ZB(N1),THETA(N1)
      COMMON /WRUNUP/ DELRP(N3),DELTAR(N3),RUNZ(N3),RUNZST(N3),NDELR
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),QST(N1),
     +                UST(N1),ETAST(N1)
      COMMON /VERVAR/ FM(N1),UB(N1),FMST(N1),UBST(N1),FM3(N1),DB(N1)
      COMMON /TAUBFW/ UUB(N1),TAUB(N1),FW(N1)
      COMMON /VECMAC/ F2(N1),F3(N1),G2(N1),G3(N1)
      COMMON /DOTMAC/ HDOT(N1),QDOT(N1),UDOT(N1),FMDOT(N1),UBDOT(N1)
      IF (TIME.EQ.0.D+00) THEN
        CALL CHEPAR (13,1,N1,N1R)
        CALL CHEPAR (13,3,N3,N3R)
      ENDIF
C
      SP1 = S+1
      SM1 = S-1
C
C ... ADJUST VALUES AT S IF HST(S)>HST(S-1)
C
      IF (HST(S).GE.HST(SM1)) THEN
        UST(S)  = 2.D+00*UST(SM1) - UST(S-2)
        HST(S)  = 2.D+00*HST(SM1) - HST(S-2)
        IF (DABS(UST(S)).GT.DABS(UST(SM1))) UST(S)=9.D-01*UST(SM1)
        IF (HST(S).LT.0.D+00) HST(S) = 5.D-01*HST(SM1)
        IF (HST(S).GT.HST(SM1)) HST(S) = 9.D-01*HST(SM1)
        QST(S)  = UST(S)*HST(S)
        WRITE  (*,2910) S,TIMEST,HST(S),HST(SM1)
        WRITE (29,2910) S,TIMEST,HST(S),HST(SM1)
      ENDIF
 2910 FORMAT (/' From Subroutine 13 LANDBC:'/
     +          ' Computed water depth HST(S)>HST(S-1) at',
     +          ' S =',I8,'; TIMEST =',E12.3/' Adjusted values:',
     +          ' HST(S) =',E12.3,'; HST(S-1) =',E12.3)
C
C ... DETERMINE WATERLINE NODE SST AT TIMEST
C
      IF (HST(S).LE.DELTA) THEN
       SST = SM1
```

```
        GO TO 1000
       ENDIF
       UST(SP1) = 2.D+00*UST(S) - UST(SM1)
       HST(SP1) = 2.D+00*HST(S) - HST(SM1)
       QST(SP1) = UST(SP1)*HST(SP1)
       IF (HST(SP1).LE.DELTA) THEN
        SST = S
        GO TO 1000
       ENDIF
C
C      If HST(S+1) > DELTA, compute HSTST and QSTST
C       (H and Q with two stars) at node j = S at time
C      level (TIMEST+DT) using MacCormack method.
C
       DO 100 J = SM1,SP1
        F2(J) = QST(J)*UST(J) + 0.5D+00 * HST(J)*HST(J)
        IF(J.LE.S) G2(J) = THETA(J)*HST(J)+FW(J)*DABS(UST(J))*UST(J)
  100  CONTINUE
       DO 110 J = SM1,S
        HDOT(J) = HST(J)-DTDX*(QST(J+1)-QST(J))
        QDOT(J) = QST(J)-DTDX*(F2(J+1)-F2(J))-DT*G2(J)
        UDOT(J) = QDOT(J)/HDOT(J)
  110  CONTINUE
       DO 120 J = SM1,S
        F2(J)=QDOT(J)*UDOT(J)+0.5D+00*HDOT(J)*HDOT(J)
  120  CONTINUE
       G2(S) = THETA(S)*HDOT(S)+FW(S)*DABS(UDOT(S))*UDOT(S)
       HDOT(S) = HDOT(S)-DTDX*(QDOT(S)-QDOT(SM1))
       QDOT(S) = QDOT(S)-DTDX*(F2(S)-F2(SM1))-DT*G2(S)
       HSTST = 0.5D+00*(HST(S)+HDOT(S))
       QSTST = 0.5D+00*(QST(S)+QDOT(S))
       USTST = QSTST/HSTST
C
C      Improve estimates of QST(S+1) = QSTSP1 and UST(S+1) = USTSP1
C      using HSTST and USTST
C
       QSTSP1 = QST(SM1) - DXDT*(HSTST-H(S))
       USTS = UST(S)
       IF(DABS(USTS).LT.DELTA) USTS = DSIGN(DELTA,USTS)
       USTSP1 = DXDT*(USTST-U(S))+HST(SP1)-HST(SM1)+2.D+00*DX*THETA(S)
       USTSP1 = UST(SM1)-USTSP1/USTS
       HSTSP1 = QSTSP1/USTSP1
C
       IF(DABS(USTSP1).LE.DELTA) THEN
```

```
        SST = S
        GO TO 1000
       ENDIF
       IF(HSTSP1.LE.HST(S).AND.HSTSP1.LE.DELTA) THEN
        SST = S
        GO TO 1000
       ENDIF
       IF(HSTSP1.LE.HST(S).AND.HSTSP1.GT.DELTA) THEN
        SST = SP1
        QST(SP1) = QSTSP1
        HST(SP1) = HSTSP1
        UST(SP1) = USTSP1
        GO TO 1000
       ENDIF
C
C      If HSTSP1 > HST(S), linearly extrapolated and stored values,
C      QST(SP1), HST(SP1), and UST(SP1) are retained instead of
C      QSTSP1, HSTSP1, and USTSP1 computed above
C
       IF(HST(SP1).LE.HST(S).AND.UST(SP1).GE.DELTA) THEN
        SST = SP1
       ELSE
        SST = S
       ENDIF
C
 1000 CONTINUE
C
C      SST = Shoreline node at next time level TIMEST has been found.
C      HST(j), QST(j), and UST(j) with j = 2,3,...,SST have been computed.
C
       IF(SST.GT.SMAX) SMAX = SST
       IF(SST.GE.JMAX) THEN
        WRITE(*,2920) TIMEST,SST,JMAX
        WRITE(29,2920) TIMEST,SST,JMAX
        STOP
       ENDIF
C
 2920 FORMAT (/' From Subroutine 13 LANDBC:'/
     +    ' TIMEST =',E12.3,'; SST =',I8,'; End Node =',I8/
     +    ' Slope is not long enough to accomodate shoreline movement'/
     +    ' Specify longer slope to avoid wave overtopping ')
C
C ... CONDITIONS LANDWARD OF NEW WATERLINE NODE SST AT time TIMEST
C
```

```
      IF(SST.EQ.SP1) THEN
       FMST(SST) = 0.D+00
      ENDIF
      DO 130 J = SST+1,JMAX
       HST(J) = 0.D+00
       QST(J) = 0.D+00
       UST(J) = 0.D+00
       FMST(J) = 0.D+00
 130  CONTINUE
C
C ... COMPUTE RUNUPS ASSOCIATED WITH DEPTHS (DELTAR(L),L=1,NDELR)
C     (Assume water depth decreases landward )
C
C     DELTAR = water depth associated with visual or measured
C              waterline
C     RUNZST = free surface elevation where the water depth equals
C              DELTAR at time level TIMEST
C     NDELR  = number of DELTARs, read if ITEMVA = 1
C
      IF (ITEMVA.EQ.1.AND.NDELR.GE.1) THEN
        DO 140 L = 1,NDELR
            INDIC = 0
            J = -1
  900       CONTINUE
              J = J + 1
              IF (HST(SST-J).GE.DELTAR(L)) THEN
                INDIC = 1
                NRUN1 = SST-J
                NRUN2 = SST-J+1
                DEL1  = HST(NRUN1)
                DEL2  = HST(NRUN2)
                RUN   = (ZB(NRUN2)-ZB(NRUN1))*(DEL1-DELTAR(L))
                RUN   = RUN/(DEL1-DEL2)
                RUN   = RUN + ZB(NRUN1)
                RUNZST(L) = RUN + DELTAR(L)
              ENDIF
            IF (INDIC.EQ.0) GOTO 900
  140   CONTINUE
      ENDIF
C
      RETURN
      END
C
C -13------------------  END OF SUBROUTINE LANDBC  ------------------
```

```
C #14#####################  SUBROUTINE SEABC  ######################
C
C     This subroutine treats seaward boundary conditions at node j = 1
C
      SUBROUTINE SEABC
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      DOUBLE PRECISION KS
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     ISYST,IWAVE,IBOT,INCLCT,IENERG,
     +                ITEMVA,ISPAEU
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /WAVREF/ HREF,TREF,KS
      COMMON /VERPAR/ APROFL,CMIXL,C2,C3,CB,CBL
      COMMON /WAVINP/ DELTI,ETAINP(N2),NPINP
      COMMON /IRWAVE/ ETAI,ETAIST,ETAR,ETARST
      COMMON /BOTPAR/ DSEAP,DSEA,SLSURF,WTOT
      COMMON /BOTNOD/ XB(N1),ZB(N1),THETA(N1)
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),QST(N1),
     +                UST(N1),ETAST(N1)
      COMMON /VERVAR/ FM(N1),UB(N1),FMST(N1),UBST(N1),FM3(N1),DB(N1)
      COMMON /TAUBFW/ UUB(N1),TAUB(N1),FW(N1)
      IF (TIME.EQ.0.D+00) THEN
        CALL CHEPAR (14,1,N1,N1R)
        CALL CHEPAR (14,2,N2,N2R)
      ENDIF
C
C ... ESTIMATE ETARST AT TIMEST
C
C     BETAST = Seaward-advancing characteristics at node 1 and at
C              time = TIMEST for IWAVE = 1 or 2
C     ETARST = Surface elevation associated with reflected wave at
C              seaward boundary at TIMEST
C     A correction term Ct is included in ETARST if INCLCT = 1 and
C     IWAVE = 1 or 2 to improve prediction of wave setdown and setup
C     on beach.
C
      C1 = DSQRT(H(1))
      IF(U(1).GE.C1) THEN
       WRITE(*,2910) TIME,U(1),C1
       WRITE(29,2910) TIME,U(1),C1
       STOP
```

```
      ENDIF
 2910 FORMAT('From Subr.14 SEABC: Seaward Boundary'/
     +         '(Flow at x = 0 is not subcritical)'/
     +         'Time of occurance          Time = ',F18.9/
     +         'Water velocity at x = 0   U    = ',F18.9/
     +         'Phase velocity at x = 0   C    = ',F18.9)
C
      BETA1 = 2.D+00 *C1-U(1)
      BETA2 = 2.D+00 *DSQRT(H(2))-U(2)
C
      IF(IWAVE.LE.2) THEN
C
       BETAST = DT*(THETA(1)+TAUB(1)/H(1))+DTDX*(FM(2)-FM(1))/H(1)
       BETAST = BETA1-DTDX*(U(1)-C1)*(BETA2-BETA1)+BETAST
       ETARST = 0.5D+00*DSQRT(DSEA)*BETAST-DSEA
       IF(INCLCT.EQ.1) ETARST=ETARST-KS*KS/(16.D+00*DSEA)
      ENDIF
C
C     Input wave train ETAINP(n) with n=1,2,...,NPINP has been
C     computed or read at time levels (n-1)*DELTI with DELTI=
C     TMAX/(NPINP-1).  Interpolate the input time series to
C     obtain ETAINT = value of ETAINP at time = TIMEST.
C     ETAIST = surface elevation associated with incident wave at
C     seaward boundary at TIMEST
C
      DJJ = TIMEST/DELTI
      JJ = INT(DJJ)
      ETA1 = ETAINP(JJ+1)
      ETA2 = ETAINP(JJ+2)
      DEL = DJJ - DBLE(JJ)
      ETAINT = ETA1+DEL*(ETA2-ETA1)
      IF(IWAVE.EQ.3) THEN
       HST(1) = DSEA + ETAINT
      ELSE
       HST(1) = DSEA + ETAINT + ETARST
       ETAIST = ETAINT
      ENDIF
      IF(IWAVE.LE.2) THEN
       UST(1) = 2.D+00*DSQRT(HST(1))-BETAST
       QST(1) = UST(1)*HST(1)
      ENDIF
C
      IF(IWAVE.EQ.3) THEN
       C1 = DSQRT(HST(1))
```

```
       DUM = DTDX*(BETA2-BETA1)
       UST(1) = (2.D+00*C1-BETA1-DUM*C1-DTDX*(FM(2)-FM(1))/H(1)-DT*(
     + THETA(1)+TAUB(1)/H(1)))/(1.D+00-DUM)
       ETARST = 0.5D+00*DSQRT(DSEA)*(2.D+00*C1-UST(1))-DSEA
       ETAIST = ETAINT-ETARST
       QST(1) = UST(1)*HST(1)
      ENDIF
C
C     Explicit first-order finite difference approximation of equation
C     for near-bottom horizontal velocity correction UB is used to
C     compute relatively small FMST(1).
      UBST(1) = (0.5D+00*C3*UB(1)/DX)*(UB(1)*(H(2)/H(1)-4.D+00)
     +           +3.D+00*UB(2))
      UBST(1) = (DT/C2)*(UBST(1)+(TAUB(1)+CBL*DABS(UB(1))*UB(1))/H(1))
      UBST(1) = UB(1)-DTDX*(U(2)*UB(2)-U(1)*UB(1))-UBST(1)
      FMST(1) = C2*HST(1)*UBST(1)**2
      RETURN
      END
C
C -14----------------- END OF SUBROUTINE SEABC ---------------------
C #15##################### SUBROUTINE SMOOTH ########################
C
C     This subroutine smooths HST(j), UST(j), and FMST(j) with
C     j=3,4,..,(SST-2) using the procedure described in Chaudhry(1993)
C     where DKAPPA is read in Subr. 02 INPUT.  The rest of the variables
C     at time = TIMEST are computed and used in Subr. 16 BSTRES, Subr. 17
C     STATIS, Subr. 18 ENERGY, and Subr. 20 DOC2.
C
      SUBROUTINE SMOOTH
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      DIMENSION A(N1),B(N1),D(N1)
      INTEGER STILL,S,SST,SMAX,SSTM1,SSTM2
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /CPARA/  DELTA,COURNO,DKAPPA
      COMMON /VERPAR/ APROFL,CMIXL,C2,C3,CB,CBL
      COMMON /BOTNOD/ XB(N1),ZB(N1),THETA(N1)
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),QST(N1),
     +                UST(N1),ETAST(N1)
      COMMON /VERVAR/ FM(N1),UB(N1),FMST(N1),UBST(N1),FM3(N1),DB(N1)
      IF(TIME.EQ.0.D+00) THEN
```

```
       CALL CHEPAR(15,1,N1,N1R)
      ENDIF
C
      SSTM1 = SST-1
      SSTM2 = SST-2
      DO 100 J = 1,SST
      A(J) = DABS(HST(J))
  100 CONTINUE
      DO 200  J = 2,SSTM1
      JP1 = J+1
      JM1 = J-1
      D(J) = DABS(HST(JP1)-2.D+00*HST(J)+HST(JM1))
      D(J) = D(J)/(A(JP1)+2.D+00*A(J)+A(JM1))
  200 CONTINUE
      DO 300 J = 2,SSTM2
      D(J)=DKAPPA*DMAX1(D(J),D(J+1))*DSQRT((HST(J)+HST(J+1))/2.D+00)
  300 CONTINUE
      DO 400 J = 2,SSTM2
      A(J) = D(J)*(HST(J+1)-HST(J))
      B(J) = D(J)*(FMST(J+1)-FMST(J))
      D(J) = D(J)*(UST(J+1)-UST(J))
  400 CONTINUE
      DO 500 J = 3,SSTM2
      HST(J) = HST(J)+A(J)-A(J-1)
      FMST(J) = FMST(J)+B(J)-B(J-1)
      UST(J) = UST(J)+D(J)-D(J-1)
      QST(J) = HST(J)*UST(J)
  500 CONTINUE
C
C     Compute variables for vertical velocity variations at
C     time = TIMEST.  Bottom shear stress at time = TIMEST is
C     computed in Subr. 16 BSTRES.
      DO 600 J = 1,SST
      IF(FMST(J).LT.0.D+00) FMST(J) = 0.D+00
      UBST(J) = DSQRT(FMST(J)/C2/HST(J))
      IF(UST(J).GE.0.D+00) UBST(J) = -UBST(J)
      FM3(J) = C3*HST(J)*UBST(J)**3
      DB(J) = CBL*DABS(UBST(J))**3
  600 CONTINUE
      DO 700 J = SST+1,JMAX
      UBST(J) = 0.D+00
      FM3(J) = 0.D+00
      DB(J) = 0.D+00
  700 CONTINUE
```

```
C
C     Compute free surface elevation above SWL at time = TIMEST
      DO 800 J = 1,JMAX
      ETAST(J) = HST(J)+ZB(J)
 800  CONTINUE
C
      RETURN
      END
C
C -15------------------ END OF SUBROUTINE SMOOTH --------------------
C #16##################### SUBROUTINE BSTRES #######################
C
C     This subroutine computes near-bottom horizontal velocity UUB(j)
C     and bottom stress TAUB(j) for j = 1,2,...,JMAX at time = TIMEST
C
      SUBROUTINE BSTRES
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      INTEGER STILL,S,SST,SMAX
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),QST(N1),
     +                UST(N1),ETAST(N1)
      COMMON /VERVAR/ FM(N1),UB(N1),FMST(N1),UBST(N1),FM3(N1),DB(N1)
      COMMON /TAUBFW/ UUB(N1),TAUB(N1),FW(N1)
      IF(TIME.EQ.0.D+00) CALL CHEPAR(16,1,N1,N1R)
C
      DO 100 J=1,SST
      UUB(J) = UST(J)+UBST(J)
      TAUB(J) = FW(J)*DABS(UUB(J))*UUB(J)
 100  CONTINUE
      DO 200 J = SST+1,JMAX
      UUB(J) = 0.D+00
      TAUB(J) = 0.D+00
 200  CONTINUE
C
      RETURN
      END
C
C -16----------------- END OF SUBROUTINE BSTRES ---------------------
C #17##################### SUBROUTINE STATIS #######################
C
```

```
C     This subroutine computes mean,root-mean-square, minimum and maximum
C     values of ETAI, ETAR, RUNZ(L) with L = 1,...,NDELR as well as
C     U(j), UUB(j), ETA(j) and Q(j) with j = 1,2,...,JMAX for duration of
C     time = TSTAT to TMAX.  Near-bottom horizontal velocity UUB(j) is
C     sum of depth-averaged velocity U(j) and near-bottom correction UB(j).
C
      SUBROUTINE STATIS
C
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      INTEGER STILL,S,SST,SMAX
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /IRWAVE/ ETAI,ETAIST,ETAR,ETARST
      COMMON /WRUNUP/ DELRP(N3),DELTAR(N3),RUNZ(N3),RUNZST(N3),NDELR
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),QST(N1),
     +                UST(N1),ETAST(N1)
      COMMON /VERVAR/ FM(N1),UB(N1),FMST(N1),UBST(N1),FM3(N1),DB(N1)
      COMMON /TAUBFW/ UUB(N1),TAUB(N1),FW(N1)
      COMMON /EISTAT/ EIMEAN,EIRMS,EIMAX,EIMIN
      COMMON /ERSTAT/ ERMEAN,ERRMS,ERMAX,ERMIN,REFCOE
      COMMON /RZSTAT/ RZMEAN(N3),RZRMS(N3),RZMAX(N3),RZMIN(N3)
      COMMON /ETSTAT/ EMEAN(N1),ERMS(N1),EMAX(N1),EMIN(N1)
      COMMON /USTAT/  UMEAN(N1),URMS(N1),UMAX(N1),UMIN(N1)
      COMMON /UBSTAT/ UBMEAN(N1),UBRMS(N1),UBMAX(N1),UBMIN(N1)
      COMMON /QSTAT/  QMEAN(N1)
C
C    FOR TIME.LE.TSTAT.LT.TIMEST, linear interpolation between TIME and
C    TIMEST is used to find values at time = TSTAT
      IF(TIME.LE.TSTAT) THEN
      CALL CHEPAR(17,1,N1,N1R)
      CALL CHEPAR(17,3,N3,N3R)
C
      A = (TIMEST-TSTAT)/DT
      B = (TSTAT-TIME)/DT
      D = (TIMEST-TSTAT)/2.D+00
C
      EIINT = A*ETAI+B*ETAIST
      EIMEAN = D*(EIINT+ETAIST)
      EI2INT = A*ETAI**2+B*ETAIST**2
      EIRMS = D*(EI2INT+ETAIST**2)
      EIMAX = DMAX1(EIINT,ETAIST)
```

```
      EIMIN = DMIN1(EIINT,ETAIST)
C
      ERINT = A*ETAR+B*ETARST
      ERMEAN = D*(ERINT+ETARST)
      ER2INT = A*ETAR**2+B*ETARST**2
      ERRMS = D*(ER2INT+ETARST**2)
      ERMAX = DMAX1(ERINT,ETARST)
      ERMIN = DMIN1(ERINT,ETARST)
C
      IF(NDELR.GT.0) THEN
       DO 100 L = 1,NDELR
       RZINT = A*RUNZ(L)+B*RUNZST(L)
       RZMEAN(L) = D*(RZINT+RUNZST(L))
       RZ2INT = A*RUNZ(L)**2+B*RUNZST(L)**2
       RZRMS(L) = D*(RZ2INT+RUNZST(L)**2)
       RZMAX(L) = DMAX1(RZINT,RUNZST(L))
       RZMIN(L) = DMIN1(RZINT,RUNZST(L))
 100   CONTINUE
      ENDIF
C
      DO 110 J = 1,JMAX
C
      ETAINT = A*ETA(J)+B*ETAST(J)
      EMEAN(J) = D*(ETAINT+ETAST(J))
      ET2INT = A*ETA(J)**2+B*ETAST(J)**2
      ERMS(J) = D*(ET2INT+ETAST(J)**2)
      EMAX(J) = DMAX1(ETAINT,ETAST(J))
      EMIN(J) = DMIN1(ETAINT,ETAST(J))
C
      UINT = A*U(J)+B*UST(J)
      UMEAN(J) = D*(UINT+UST(J))
      U2INT = A*U(J)**2+B*UST(J)**2
      URMS(J) = D*(U2INT+UST(J)**2)
      UMAX(J) = DMAX1(UINT,UST(J))
      UMIN(J) = DMIN1(UINT,UST(J))
C
C     UUB(j) at time = TIMEST and V = (U(j)+UB(j)) at time = TIME
      V = U(J)+UB(J)
      VINT = A*V+B*UUB(J)
      UBMEAN(J) = D*(VINT+UUB(J))
      V2INT = A*V**2+B*UUB(J)**2
      UBRMS(J) = D*(V2INT+UUB(J)**2)
      UBMAX(J) = DMAX1(VINT,UUB(J))
      UBMIN(J) = DMIN1(VINT,UUB(J))
```

```
C
      QINT = A*Q(J)+B*QST(J)
      QMEAN(J) = D*(QINT+QST(J))
C
 110  CONTINUE
      ENDIF
C
C     If TIME>TSTAT use a trapezoid method to calculate mean values
C
      IF(TIME.GT.TSTAT) THEN
       D = DT/2.D+00
C
       EIMEAN = EIMEAN+D*(ETAI+ETAIST)
       EIRMS = EIRMS+D*(ETAI**2+ETAIST**2)
       IF(ETAIST.GT.EIMAX) EIMAX = ETAIST
       IF(ETAIST.LT.EIMIN) EIMIN = ETAIST
C
       ERMEAN = ERMEAN+D*(ETAR+ETARST)
       ERRMS = ERRMS+D*(ETAR**2+ETARST**2)
       IF(ETARST.GT.ERMAX) ERMAX = ETARST
       IF(ETARST.LT.ERMIN) ERMIN = ETARST
C
       IF(NDELR.GT.0) THEN
       DO 120 L = 1,NDELR
       RZMEAN(L) = RZMEAN(L)+D*(RUNZ(L)+RUNZST(L))
       RZRMS(L) = RZRMS(L)+D*(RUNZ(L)**2+RUNZST(L)**2)
       IF(RUNZST(L).GT.RZMAX(L)) RZMAX(L) = RUNZST(L)
       IF(RUNZST(L).LT.RZMIN(L)) RZMIN(L) = RUNZST(L)
 120   CONTINUE
       ENDIF
C
       DO 130 J = 1,JMAX
C
       EMEAN(J) = EMEAN(J)+D*(ETA(J)+ETAST(J))
       ERMS(J) = ERMS(J)+D*(ETA(J)**2+ETAST(J)**2)
       IF(ETAST(J).GT.EMAX(J)) EMAX(J) = ETAST(J)
       IF(ETAST(J).LT.EMIN(J)) EMIN(J) = ETAST(J)
C
       UMEAN(J) = UMEAN(J)+D*(U(J)+UST(J))
       URMS(J) = URMS(J)+D*(U(J)**2+UST(J)**2)
       IF(UST(J).GT.UMAX(J)) UMAX(J) = UST(J)
       IF(UST(J).LT.UMIN(J)) UMIN(J) = UST(J)
C
       V = U(J)+UB(J)
```

```
        UBMEAN(J) = UBMEAN(J)+D*(V+UUB(J))
        UBRMS(J) = UBRMS(J) + D*(V**2+UUB(J)**2)
        IF(UUB(J).GT.UBMAX(J)) UBMAX(J) = UUB(J)
        IF(UUB(J).LT.UBMIN(J)) UBMIN(J) = UUB(J)
C
        QMEAN(J)=QMEAN(J)+D*(Q(J)+QST(J))
 130    CONTINUE
       ENDIF
C
C      If TIMEST = TMAX, complete mean caclculations at
C      end of time-marching computation.
C
       IF(TIMEST.EQ.TMAX) THEN
        D = TMAX-TSTAT
        EIMEAN = EIMEAN/D
        EIRMS = DSQRT(EIRMS/D-EIMEAN**2)
        ERMEAN = ERMEAN/D
        ERRMS = DSQRT(ERRMS/D-ERMEAN**2)
        REFCOE = ERRMS/EIRMS
C
        IF(NDELR.GT.0) THEN
        DO 140 L=1,NDELR
        RZMEAN(L) = RZMEAN(L)/D
        RZRMS(L) = DSQRT(RZRMS(L)/D-RZMEAN(L)**2)
 140    CONTINUE
        ENDIF
C
C       SMAX at TIMEST = TMAX is maximum wet node number during
C       entire computation
        DO 150 J = 1,SMAX
        EMEAN(J) = EMEAN(J)/D
        A = ERMS(J)/D - EMEAN(J)**2
        IF(A.LT.0.D+00) THEN
         ERMS(J) = 0.D+00
        ELSE
         ERMS(J) = DSQRT(A)
        ENDIF
        UMEAN(J) = UMEAN(J)/D
        URMS(J) = DSQRT(URMS(J)/D-UMEAN(J)**2)
        UBMEAN(J) = UBMEAN(J)/D
        UBRMS(J) = DSQRT(UBRMS(J)/D-UBMEAN(J)**2)
        QMEAN(J) = QMEAN(J)/D
 150    CONTINUE
       ENDIF
```

```
C
      RETURN
      END
C
C -17----------------- END OF SUBROUTINE STATIS --------------------
C #18#################### SUBROUTINE ENERGY ########################
C
C     This subroutine computes quantities related to wave energy
C
      SUBROUTINE ENERGY(ICALL)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      DIMENSION ES(N1),EF(N1),DF(N1)
      INTEGER STILL,S,SST,SMAX
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /BOTNOD/ XB(N1),ZB(N1),THETA(N1)
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),QST(N1),
     +                UST(N1),ETAST(N1)
      COMMON /VERVAR/ FM(N1),UB(N1),FMST(N1),UBST(N1),FM3(N1),DB(N1)
      COMMON /TAUBFW/ UUB(N1),TAUB(N1),FW(N1)
      COMMON /WESTAT/ ESMEAN(N1),EFMEAN(N1),DFMEAN(N1),DBMEAN(N1),
     +                DBMDIF(N1),DELES(N1)
      COMMON /ENERG/  ESPC(N1),EFLUX(N1),DISF(N1),DISB(N1)
C
C     ICALL = 1 when TIME<TSTAT<TIMEST and wave energy quantities
C     at TIME are computed
      IF(ICALL.EQ.1) THEN
       CALL CHEPAR(18,1,N1,N1R)
       DO 50 J = 1,JMAX
       ESPC(J) = 0.5D+00*(Q(J)*U(J)+FM(J)+ETA(J)**2)
       IF(ZB(J).GT.0.D+00) ESPC(J) = ESPC(J)-0.5D+00*ZB(J)**2
       EFLUX(J) = Q(J)*ETA(J)+0.5D+00*(Q(J)*U(J)**2+3.D+00*U(J)*
     + FM(J)+FM3(J))
       DISF(J) = TAUB(J)*UUB(J)
       DISB(J) = DB(J)
 50    CONTINUE
       GO TO 200
      ENDIF
C
C     For ICALL = 2, instantaneous wave energy quantities at node J,
```

```
C     at time = TIMEST
C       ES(J) = norm. energy per unit surface area
C       EF(J) = norm. energy flux per unit width
C     Normalized rate of energy dissipation at node j:
C       DF(J): due to bottom friction, per unit bottom area
C       DB(J): due to wave breaking, per unit surface area
C              computed in Subr. 15 SMOOTH
C
C      Corresponding values at time = TIME have been stored
C      in ESPC(J), EFLUX(J), DISF(J) and DISB(J), respectively, for
C      ICALL = 1.
C
      IF(ICALL.EQ.2) THEN
       DO 100 J = 1,JMAX
       ES(J) = 0.5D+00*(QST(J)*UST(J)+FMST(J)+ETAST(J)**2)
       IF(ZB(J).GT.0.D+00) ES(J) = ES(J)-0.5D+00*ZB(J)**2
       EF(J) = QST(J)*UST(J)**2+3.D+00*UST(J)*FMST(J)+FM3(J)
       EF(J) = QST(J)*ETAST(J)+0.5D+00*EF(J)
       DF(J) = TAUB(J)*UUB(J)
 100   CONTINUE
      ENDIF
C
C ... If TIMEST>TSTAT, compute mean quantities for duration of
C     time = TSTAT to TMAX in the same manner as in Subr. 17 STATIS
C     DELES(J) stores ES(J) at time = TSTAT
C
      IF( TIME.LE.TSTAT) THEN
       A = (TIMEST-TSTAT)/DT
       B = (TSTAT-TIME)/DT
       D = (TIMEST-TSTAT)/2.D+00
       DO 110 J = 1,JMAX
       DELES(J) = A*ESPC(J)+B*ES(J)
       ESMEAN(J) = D*(DELES(J)+ES(J))
       EFMEAN(J) = D*(A*EFLUX(J)+B*EF(J)+EF(J))
       DFMEAN(J) = D*(A*DISF(J)+B*DF(J)+DF(J))
       DBMEAN(J) = D*(A*DISB(J)+B*DB(J)+DB(J))
 110   CONTINUE
      ENDIF
C
      IF(TIME.GT.TSTAT) THEN
       D = DT/2.D+00
       DO 120 J = 1,JMAX
       ESMEAN(J) = ESMEAN(J)+D*(ESPC(J)+ES(J))
       EFMEAN(J) = EFMEAN(J)+D*(EFLUX(J)+EF(J))
```

```
        DFMEAN(J) = DFMEAN(J)+D*(DISF(J)+DF(J))
        DBMEAN(J) = DBMEAN(J)+D*(DISB(J)+DB(J))
 120    CONTINUE
       ENDIF
C
C      If TIMEST<TMAX, proceed to next time level
C
       IF(TIMEST.LT.TMAX) THEN
        DO 130 J = 1,JMAX
        ESPC(J) = ES(J)
        EFLUX(J) = EF(J)
        DISF(J) = DF(J)
        DISB(J) = DB(J)
 130    CONTINUE
        GO TO 200
       ENDIF
C
C      If TIMEST=TMAX, complete mean calculations at end of time-marching
C      computation. DBMDIF(J) is numerical energy dissipation rate estimated
C      using time-averaged wave energy equation where DELES(J) accounts for
C      the change of ES(J) from time = TSTAT to time = TMAX
C
       IF(TIMEST.EQ.TMAX) THEN
        D = TMAX-TSTAT
        DO 140 J = 1,SMAX
        ESMEAN(J) = ESMEAN(J)/D
        EFMEAN(J) = EFMEAN(J)/D
        DFMEAN(J) = DFMEAN(J)/D
        DBMEAN(J) = DBMEAN(J)/D
        DELES(J) = (ES(J)-DELES(J))/D
 140    CONTINUE
C
        TWODX = 2.D+00*DX
        DO 150 J = 1,SMAX
        IF(J.EQ.1) DEFDX = (EFMEAN(2)-EFMEAN(1))/DX
        IF(J.EQ.SMAX) DEFDX = (EFMEAN(SMAX)-EFMEAN(SMAX-1))/DX
        IF(J.GT.1.AND.J.LT.SMAX) THEN
         DEFDX = (EFMEAN(J+1)-EFMEAN(J-1))/TWODX
        ENDIF
        B = -DEFDX-DFMEAN(J)-DELES(J)
        DBMDIF(J) = B-DBMEAN(J)
 150    CONTINUE
       ENDIF
C
```

```
 200  RETURN
      END
C
C -18---------------- END OF SUBROUTINE ENERGY -------------------
C #19####################### SUBROUTINE DOC1 #######################
C
C     This subroutine documents input data and related parameters
C     before time-marching computation
C
      SUBROUTINE DOC1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      DOUBLE PRECISION KCNO,MCNO,KC2
      DOUBLE PRECISION KS,KSI
      DIMENSION DUME(N5),DUMU(N5),DUMUB(N5),DUMZ(N5)
      CHARACTER*7 UL
      INTEGER STILL,S,SST,SMAX
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     ISYST,IWAVE,IBOT,INCLCT,IENERG,ITEMVA,ISPAEU
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /CPARA/  DELTA,COURNO,DKAPPA
      COMMON /WAVREF/ HREF,TREF,KS
      COMMON /VERPAR/ APROFL,CMIXL,C2,C3,CB,CBL
      COMMON /WAVINP/ DELTI,ETAINP(N2),NPINP
      COMMON /IRWAVE/ ETAI,ETAIST,ETAR,ETARST
      COMMON /WAVPAR/ SIGMA,WL,UR,KSI
      COMMON /CNOWAV/ KCNO,ECNO,MCNO,KC2
      COMMON /BOTPAR/ DSEAP,DSEA,SLSURF,WTOT
      COMMON /BOTSEG/ WBSEG(N4),TBSLOP(N4),XBSEG(N4),ZBSEG(N4),
     +                BFFSEG(N4),NBSEG
      COMMON /BOTNOD/ XB(N1),ZB(N1),THETA(N1)
      COMMON /WRUNUP/ DELRP(N3),DELTAR(N3),RUNZ(N3),RUNZST(N3),NDELR
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),QST(N1),
     +                UST(N1),ETAST(N1)
      COMMON /VERVAR/ FM(N1),UB(N1),FMST(N1),UBST(N1),FM3(N1),DB(N1)
      COMMON /TAUBFW/ UUB(N1),TAUB(N1),FW(N1)
      COMMON /STOTEP/ DELTO,TIMOUT(N2),NPOUT
      COMMON /STONOD/ NONODS,NODLOC(N5)
      COMMON /STOSPA/ TIMSPA(N5),NOTIML
      CALL CHEPAR (19,1,N1,N1R)
      CALL CHEPAR (19,2,N2,N2R)
```

```
      CALL CHEPAR (19,3,N3,N3R)
      CALL CHEPAR (19,4,N4,N4R)
      CALL CHEPAR (19,5,N5,N5R)
C
C ... SYSTEM OF UNITS
C
      IF (ISYST.EQ.1) THEN
        UL = ' meters'
      ELSE
        UL = ' feet   '
      ENDIF
C
C ... WAVE CONDITION
C
      WRITE (28,2811)
      IF (IWAVE.EQ.1) THEN
        IF (MCNO.EQ.0.D+00) THEN
          WRITE (28,2812) KS
        ELSE
          WRITE (28,2813) KS,KC2,ECNO,KCNO
        ENDIF
      ELSEIF (IWAVE.EQ.2) THEN
        WRITE (28,2814)
      ELSE
        WRITE (28,2815)
      ENDIF
      WRITE (28,2816) TREF,HREF,UL,DSEAP,UL
      WRITE (28,2817) DSEA,INCLCT,WL,SIGMA,UR,KSI
      WRITE (28,2818) DELTI
 2811 FORMAT (/'WAVE CONDITION')
 2812 FORMAT (/'Stokes II Incident Wave at Seaward Boundary'/
     +         'Normalized wave height KS = ',F12.6)
 2813 FORMAT (/'Cnoidal Incident Wave at Seaward Boundary'/
     +         'Normalized wave height KS      =   'F12.6/
     +         '1-m = ',D20.9/
     +         'E   = ',D20.9/
     +         'K   = ',D20.9)
 2814 FORMAT (/'Incident Wave at Seaward Boundary Read as Input')
 2815 FORMAT (/'Measured Total Wave Profile at Seaward Boundary'/)
 2816 FORMAT ('Reference Wave Period             = ',F12.6,' sec.'/
     +         'Reference Wave Height            = ',F12.6,A7/
     +         'Depth at Seaward Boundary        = ',F12.6,A7)
 2817 FORMAT ('Norm. Depth at Seaw. Bdr.        = ',F9.3/
     +         'Included Correction Term   CT'/
```

```
     +        '0 = no; 1 = yes        INCLCT  = ',' ',I1/
     +        'Normalized Wave Length         = ',F9.3/
     +        '"Sigma"                        = ',F9.3/
     +        'Ursell Number                  = ',F9.3/
     +        'Surf Similarity Parameter      = ',F9.3)
 2818 FORMAT ('Input Wave Train from Time=0 to TMAX'/
     +        'Computed or Read at Normalized Rate DELTI = ',F12.6)
C
C ... parameters of vertical velocity variations
C
      WRITE(28,2819) APROFL,CMIXL,C2,C3,CB,CBL
 2819 FORMAT(/'Parameters of Vertical Velocity Variations'/
     +        'Cubic Profile Parameter     APROFL = ',F12.6/
     +        'Mixing Length Parameter     CMIXL  = ',F12.6/
     +        'Momentum Flux Coefficient   C2     = ',F12.6/
     +        'Kinetic Energy Flux Coeff.  C3     = ',F12.6/
     +        'Energy Dissipation Coeff.   CB     = ',F12.6/
     +        'Coefficient of DB           CBL    = ',F12.6/)
C
C ... BOTTOM GEOMETRY
C
      WRITE (28,2821) WTOT,NBSEG
      IF (IBOT.EQ.1) THEN
        WRITE (28,2822) UL
        WRITE (28,2824) (K,WBSEG(K),TBSLOP(K),BFFSEG(K),K=1,NBSEG)
      ELSE
        WRITE (28,2823) UL,UL
        WRITE (28,2825) XBSEG(1),ZBSEG(1)
        WRITE (28,2824) (K-1,XBSEG(K),ZBSEG(K),BFFSEG(K-1),
     +                   K=2,NBSEG+1)
      ENDIF
 2821 FORMAT (/'BOTTOM GEOMETRY'//
     +        'Norm. Horiz. Length of'/
     +        '      Computation Domain    = ',F12.6/
     +        'Number of Segments          = ',I8)
 2822 FORMAT ('-------------------------------------------'/
     +        ' SEGMENT    WBSEG(I)   TBSLOP(I)    BFFSEG(I)'/
     +        '     I      ',A7/
     +        '-------------------------------------------')
 2823 FORMAT ('-------------------------------------------'/
     +        ' SEGMENT    XBSEG(I)    ZBSEG(I)    BFFSEG(I)'/
     +        '     I      ',A7,'     ',A7/
     +        '-------------------------------------------')
 2824 FORMAT (I8,3F12.6)
```

```
 2825 FORMAT ('.....X=0',2F12.6)
C
C ... COMPUTATION PARAMETERS
C
      WRITE (28,2841) DX,DELTA,COURNO,DKAPPA
      WRITE (28,2842) TMAX,TSTAT,JMAX
      WRITE (28,2843) STILL
      IF(ITEMVA.EQ.1) THEN
      WRITE (28,2844) DELTO
      IF(NDELR.GT.0) WRITE(28,2845) NDELR
      ENDIF
      IF(ITEMVA.EQ.1.AND.NONODS.GT.0) WRITE(28,2846) NONODS
      IF(ISPAEU.EQ.1) WRITE(28,2847) NOTIML
 2841 FORMAT (/'COMPUTATION PARAMETERS'//
     + 'Normalized DX                        =    ',D14.6/
     + 'Normalized DELTA                     =    ',E14.6/
     + 'Courant Number                       = ',F9.3/
     + ' Must not exceed unity '/
     + 'Numerical Damping Coefficient        =  ',F9.4/
     + ' Must be zero or positive')
 2842 FORMAT (
     + 'Normalized Computation Duration  TMAX = ',F12.6/
     + 'Statistical Calculations Start         '/
     + ' when Time is equal to           TSTAT= ',F12.6/
     + 'Total Number of Spatial Nodes    JMAX = ',I8)
 2843 FORMAT (
     + 'Number of Nodes Along Bottom Below SWL'/
     + '                                STILL = ',I8)
 2844 FORMAT (
     + 'Storing Temporal Variations from Time = 0'/
     + ' to TMAX at Normalized Rate     DELTO = ',F12.6)
 2845 FORMAT (
     + 'Wave Runup Time Series Stored for'/
     + '                                 NDER = ',I3,' Water Depths')
 2846 FORMAT (
     + 'Time Series of ETA, U, and UB'/
     + '           Stored at           NONODS = ',I8,' Nodes')
 2847 FORMAT (
     + 'Spacial Variations of ETA, U, and UB'/
     + '          Stored at            NOTIML =',I8,' Time Levels')
C
C ... NORMALIZED STRUCTURE GEOMETRY
C
C     File 22 = 'OSPACE'
```

```
C     (XB(j),ZB(j)) = normalized coordinates of bottom geometry
C                     at node j
C     ZB negative below SWL
C
      WRITE (22,2210) JMAX
      WRITE (22,2220) (XB(J),ZB(J),J=1,JMAX)
 2210 FORMAT (I8)
 2220 FORMAT (6D12.4)
C
C     If ITEMVA = 1, temporal variations are stored from TIME = 0
C     in the following files:
C     File 30 = 'OIRWAV' for incident wave train
C       ETAI and reflected wave train ETAR
C     File 31 = 'ORUNUP' for wave runup elevations RUNZ(L)
C       with L = 1,...,NDELR
C     File 41 = 'FSTORE' for normalized free surface
C       elevation ETA at NONODS nodes where the bottom elevations ZB(I)
C       at these nodes are stored at the beginning of this file where
C       water depth H(I) = ETA(I)-ZB(I)
C     File 42 = 'FSTORU' for normalized depth averaged
C       velocity U at NONODS nodes
C     File 43 = 'FSTOUB' for near-bottom horizontal velocity
C       correction UB at NONODS nodes
      IF(ITEMVA.EQ.1) THEN
       TIME = 0.D+00
       WRITE(30,8001) ETAI,ETAR,TIME
       IF(NDELR.GT.0) WRITE(31,8001) (RUNZ(L),L=1,NDELR),TIME
       IF(NONODS.GT.0) THEN
        DO 100 I=1,NONODS
        J = NODLOC(I)
        DUMZ(I) = ZB(J)
        DUME(I) = ETA(J)
        DUMU(I) = U(J)
        DUMUB(I) = UB(J)
 100    CONTINUE
        WRITE(41,8001) (DUMZ(I),I=1,NONODS)
        WRITE(41,8001) (DUME(I),I=1,NONODS),TIME
        WRITE(42,8001) (DUMU(I),I=1,NONODS),TIME
        WRITE(43,8001) (DUMUB(I),I=1,NONODS),TIME
       ENDIF
      ENDIF
C
 8001 FORMAT(5F15.6)
C
```

```
      RETURN
      END
C
C -19------------------ END OF SUBROUTINE DOC1 ---------------------
C #20##################### SUBROUTINE DOC2 #######################
C
C     This subroutine stores computed results at designated time
C     levels during time-marching computation
C
      SUBROUTINE DOC2 (ICALL,TIMINT)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      DIMENSION EINT(N1),UINT(N1),UBINT(N1),DUMR(N3),DUME(N5),
     +          DUMU(N5),DUMUB(N5)
      INTEGER STILL,S,SST,SMAX
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /TLEVEL/ TIME,TIMEST,TSTAT,TMAX
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /IRWAVE/ ETAI,ETAIST,ETAR,ETARST
      COMMON /WRUNUP/ DELRP(N3),DELTAR(N3),RUNZ(N3),RUNZST(N3),NDELR
      COMMON /HQUETA/ H(N1),Q(N1),U(N1),ETA(N1),HST(N1),
     +                QST(N1),UST(N1),ETAST(N1)
      COMMON /VERVAR/ FM(N1),UB(N1),FMST(N1),UBST(N1),FM3(N1),DB(N1)
      COMMON /STONOD/ NONODS,NODLOC(N5)
C
      IF (ICALL.EQ.0) THEN
C
C ..... CHECKING PARAMETERs
C
        CALL CHEPAR (20,1,N1,N1R)
        CALL CHEPAR (20,3,N3,N3R)
        CALL CHEPAR (20,5,N5,N5R)
      ENDIF
C
      IF (ICALL.EQ.1) THEN
C
C       Storing spatial variations of ETA, U, and UB
C
C       File 22 = 'OSPACE'
C       TIMINT = time level of storing ETA, U, and UB
C       JMAX = landward-end node
C       At node j:
```

```
C        ETA; ETAST(j) = surface elevation above SWL at TIME and TIMEST
C        U; UST(j)     = depth-averaged velocity at TIME and TIMEST
C        UB; UBST(j)   = near-bottom velocity correction at TIME and TIMEST
C
C       EINT(j)  = interpolated elevation at TIMINT
C       UINT(j)  = interpolated velocity at TIMINT
C       UBINT(j) = interpolated correction at TIMINT
C
      A=(TIMEST-TIMINT)/DT
      B=(TIMINT-TIME)/DT
      DO 100 J=1,JMAX
      EINT(J)=A*ETA(J)+B*ETAST(J)
      UINT(J)=A*U(J)+B*UST(J)
      UBINT(J)=A*UB(J)+B*UBST(J)
 100  CONTINUE
      WRITE(22,9000) JMAX
       WRITE(22,8002) (EINT(J),UINT(J),UBINT(J),J=1,JMAX),TIMINT
      ENDIF
C
      IF (ICALL.EQ.2) THEN
C ... Storing time series at TIMINT corresponding to
C     TIMOUT(n) = (n-1)*DELTO with n=2,3,...,NPOUT
C      where DELTO=TMAX/(NPOUT-1)
C     File 30 = 'OIRWAV' for incident wave train
C      (ETAI,ETAIST) and reflected wave train (ETAR,ETARST)
C     File 31 = 'ORUNUP' for wave runup elevations RUNZ(L)
C      and RUNZST(L) with L=1,...,NDELR
C     File 41 = 'FSTORE' for free surface elevation
C      (ETA,ETAST) at NONODS nodes
C     File 42 = 'FSTORU' for depth-averaged velocity
C      (U,UST) at NONODS nodes
C     File 43 = 'FSTOUB' for near-bottom velocity correction
C      (UB,UBST) at NONODS nodes
C     Values at TIME and TIMEST are interpolated to
C      find value at TIMINT
C
      A = (TIMEST-TIMINT)/DT
      B = (TIMINT-TIME)/DT
C
      EIINT = A*ETAI+B*ETAIST
      ERINT = A*ETAR+B*ETARST
      WRITE(30,8001) EIINT,ERINT,TIMINT
C
      IF(NDELR.GT.0) THEN
```

```
      DO 110 L = 1,NDELR
      DUMR(L) = A*RUNZ(L)+B*RUNZST(L)
 110  CONTINUE
      WRITE(31,8001) (DUMR(L),L=1,NDELR),TIMINT
      ENDIF
C
      IF(NONODS.GT.0) THEN
      DO 120 I = 1,NONODS
      J = NODLOC(I)
      DUME(I) = A*ETA(J)+B*ETAST(J)
      DUMU(I) = A*U(J)+B*UST(J)
      DUMUB(I) = A*UB(J)+B*UBST(J)
 120  CONTINUE
      WRITE(41,8001) (DUME(I),I=1,NONODS),TIMINT
      WRITE(42,8001) (DUMU(I),I=1,NONODS),TIMINT
      WRITE(43,8001) (DUMUB(I),I=1,NONODS),TIMINT
      ENDIF
C
      ENDIF
C
C ... FORMATS
C
 9000 FORMAT (I8)
 8001 FORMAT (5F15.6)
 8002 FORMAT (3F15.6)
C
      RETURN
      END
C
C -20------------------ END OF SUBROUTINE DOC2 ---------------------
C #21###################### SUBROUTINE DOC3 ########################
C
C     This subroutine documents results after time-marching
C     computation
C
      SUBROUTINE DOC3
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=70000,N3=3,N4=100,N5=40)
      CHARACTER*7 UL
      INTEGER STILL,S,SST,SMAX
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     ISYST,IWAVE,IBOT,INCLCT,IENERG,ITEMVA,ISPAEU
      COMMON /NODES/  STILL,S,SST,SMAX,JMAX
```

```
      COMMON /GRID/   DX,DT,DXDT,DTDX,DTMAX,DTMIN
      COMMON /BOTNOD/ XB(N1),ZB(N1),THETA(N1)
      COMMON /WRUNUP/ DELRP(N3),DELTAR(N3),RUNZ(N3),RUNZST(N3),NDELR
      COMMON /EISTAT/ EIMEAN,EIRMS,EIMAX,EIMIN
      COMMON /ERSTAT/ ERMEAN,ERRMS,ERMAX,ERMIN,REFCOE
      COMMON /RZSTAT/ RZMEAN(N3),RZRMS(N3),RZMAX(N3),RZMIN(N3)
      COMMON /ETSTAT/ EMEAN(N1),ERMS(N1),EMAX(N1),EMIN(N1)
      COMMON /USTAT/  UMEAN(N1),URMS(N1),UMAX(N1),UMIN(N1)
      COMMON /UBSTAT/ UBMEAN(N1),UBRMS(N1),UBMAX(N1),UBMIN(N1)
      COMMON /QSTAT/  QMEAN(N1)
      COMMON /WESTAT/ ESMEAN(N1),EFMEAN(N1),DFMEAN(N1),DBMEAN(N1),
     +                DBMDIF(N1),DELES(N1)
      CALL CHEPAR (21,1,N1,N1R)
      CALL CHEPAR (21,3,N3,N3R)
C
C ... SYSTEM OF UNITS
C
      IF (ISYST.EQ.1) THEN
        UL = '   [cm]'
      ELSE
        UL = ' [inch]'
      ENDIF
C
C ... MAXIMUM AND MINIMUM TIME STEPS
C
      WRITE (28,2811) DTMAX,DTMIN
 2811 FORMAT (/'Maximum time step       =',E15.5/
     +         'Minimum time step       =',E15.5)
C
C ... INCIDENT AND REFLECTED WAVES
C
      WRITE (28,2810) REFCOE
 2810 FORMAT (/'REFLECTION COEFFICIENT'//
     +         'ETARRMS/ETAIRMS = ',F9.3)
C
      WRITE(28,2812)
      WRITE(28,2813) EIMAX,EIMIN,EIMEAN,EIRMS
      WRITE(28,2814) ERMAX,ERMIN,ERMEAN,ERRMS
 2812 FORMAT(/'INCIDENT AND REFLECTED WAVES'//
     +'             Max          Min            Mean            RMS')
 2813 FORMAT('Inc.',4(3X,F10.4))
 2814 FORMAT('Ref.',4(3X,F10.4))
C
        WRITE (28,2821) SMAX
```

```
      WRITE (28,2822) UL
      DO 110 L = 1,NDELR
        WRITE(28,2823)L,DELRP(L),RZMAX(L),RZMIN(L),RZMEAN(L),RZRMS(L)
 110  CONTINUE
2821 FORMAT (/'SHORELINE OSCILLATIONS'//
    +'Largest Node Number Reached by Computational Shoreline'/
    +'                                          SMAX = ',I8)
2822 FORMAT (
    +'-------------------------------------------------------------'/
    +'       I   DELTAR(I)    RUNUP(I)  RUNDOWN(I)  SETUP(I) RMS(I)'/
    +'          ',A7,'           Ru          Rd          Zr      Rrms '/
    +'-------------------------------------------------------------')
2823 FORMAT (I8,1X,F9.3,4(2X,F9.3))
C
C ... STATISTICS OF HYDRODYNAMIC QUANTITIES
C
C     File 23 = 'OSTAT'
C       SMAX = the largest node number reached by computational
C              shoreline
C     Mean, rms, max. and min. values at node j of ETA, U, UUB, and Q.
C     For plotting these quantities as a function of x, the normalized
C     bottom geometry ( XB(j), ZB(j)) is stored for plotting convience
C
      WRITE(23,9000) SMAX
      WRITE(23,8001) (XB(J),J=1,SMAX)
      WRITE(23,8001) (ZB(J),J=1,SMAX)
C     Free surface elevation ETA
      WRITE(23,8001) (EMAX(J),J=1,SMAX)
      WRITE(23,8001) (EMIN(J),J=1,SMAX)
      WRITE(23,8001) (EMEAN(J),J=1,SMAX)
      WRITE(23,8001) (ERMS(J),J=1,SMAX)
C     Depth-averaged velocity U
      WRITE(23,8001) (UMAX(J),J=1,SMAX)
      WRITE(23,8001) (UMIN(J),J=1,SMAX)
      WRITE(23,8001) (UMEAN(J),J=1,SMAX)
      WRITE(23,8001) (URMS(J),J=1,SMAX)
C     Near-bottom horizontal velocity UUB
      WRITE(23,8001) (UBMAX(J),J=1,SMAX)
      WRITE(23,8001) (UBMIN(J),J=1,SMAX)
      WRITE(23,8001) (UBMEAN(J),J=1,SMAX)
      WRITE(23,8001) (UBRMS(J),J=1,SMAX)
C     Volume flux Q per unit width
      WRITE(23,8001) (QMEAN(J),J=1,SMAX)
C
```

```
C     TIME-AVERAGED ENERGY QUANTITIES
C
C     FILE 35 = 'OENERG'
C     At node j:
C     ESMEAN(j) = Specific wave energy
C     EFMEAN(j) = Energy flux per unit width
C     DFMEAN(j) = Dissipation rate due to bottom friction
C     DBMEAN(j) = Dissipation rate due to wave breaking
C     DBMDIF(j) = Numerical energy dissipation rate estimated using
C                 time averaged energy equation
C     DELES(j)  = Change of specific wave energy for duration of
C                 time = TSTAT to TMAX
C     Which are plotted as a function of x
C
      IF(IENERG.EQ.1) THEN
      WRITE(35,9000) SMAX
      WRITE(35,8001) (XB(J),J=1,SMAX)
      WRITE(35,8001) (ESMEAN(J),J=1,SMAX)
      WRITE(35,8001) (EFMEAN(J),J=1,SMAX)
      WRITE(35,8001) (DFMEAN(J),J=1,SMAX)
      WRITE(35,8001) (DBMEAN(J),J=1,SMAX)
      WRITE(35,8001) (DBMDIF(J),J=1,SMAX)
      WRITE(35,8001) (DELES(J) ,J=1,SMAX)
      ENDIF
C
 9000 FORMAT (I8)
 8001 FORMAT (5F15.6)
      RETURN
      END
C
C -21------------------ END OF SUBROUTINE DOC3 --------------------
C #22##################### SUBROUTINE CHEPAR ######################
C
C     This subroutine checks PARAMETER NCHEK=N1,N2,N3,N4,N5 specified
C     in given subroutine (ICALL) match NREF=N1R,N2R,N3R,N4R,N5R
C     specified in Main Program
C
      SUBROUTINE CHEPAR (ICALL,NW,NCHEK,NREF)
C
      CHARACTER*2 WHICH(5)
      CHARACTER*6 SUBR(23)
      DATA WHICH /'N1','N2','N3','N4','N5'/
      DATA SUBR  /'OPENER','INPUT1','INPUT2','BOTTOM','PARAM ',
     1            'INIT  ','INCREG','FINDM ','CEL   ',
```

```
     2              'SNCNDN','COMPDT','MARCH ','LANDBC',
     3              'SEABC ','SMOOTH','BSTRES','STATIS','ENERGY',
     4              'DOC1  ','DOC2  ','DOC3  ',
     5              'CHEPAR','CHEOPT'/
      IF (NCHEK.NE.NREF) THEN
        WRITE  (*,2910)
     +         WHICH(NW),NCHEK,ICALL,SUBR(ICALL),WHICH(NW),NREF
        WRITE (29,2910)
     +         WHICH(NW),NCHEK,ICALL,SUBR(ICALL),WHICH(NW),NREF
        STOP
      ENDIF
2910  FORMAT (/
     +' PARAMETER Error: ',A2,' =',I8,' in Subroutine',I3,' ',A6/
     +' Correct Value:   ',A2,' =',I8)
C
      RETURN
      END
C
C -22----------------- END OF SUBROUTINE CHEPAR  --------------------
C #23##################### SUBROUTINE CHEOPT  #######################
C
C     This subroutine checks user's options specified in Subr. 02 INPUT1
C
      SUBROUTINE CHEOPT (ICALL,INDIC,ITEM,ILOW,IUP)
C
      CHARACTER*2 WHICH(7)
      CHARACTER*6 OPTI(14)
      DATA WHICH /'N1','N2','N4','N2','N3','N5','N5'/
      DATA OPTI  /'ISYST ','IWAVE ','IBOT  ','INCLCT',
     1            'IENERG','ITEMVA','ISPAEU','STILL ',
     2            'NPINP ','NBSEG ','NPOUT ','NDELR ','NONODS',
     3            'NOTIML'/
      IF (ICALL.LE.7) THEN
        IF (ITEM.LT.ILOW.OR.ITEM.GT.IUP) THEN
          WRITE  (*,2910) OPTI(ICALL),ITEM,OPTI(ICALL),ILOW,IUP
          WRITE (29,2910) OPTI(ICALL),ITEM,OPTI(ICALL),ILOW,IUP
          INDIC = INDIC + 1
        ENDIF
      ELSE
        IF (ITEM.LT.ILOW.OR.ITEM.GT.IUP) THEN
          I = ICALL-7
          WRITE  (*,2920) OPTI(ICALL),ITEM,OPTI(ICALL),ILOW,IUP,WHICH(I)
          WRITE (29,2920) OPTI(ICALL),ITEM,OPTI(ICALL),ILOW,IUP,WHICH(I)
          STOP
```

```
        ENDIF
      ENDIF
 2910 FORMAT (/' Input Error: ',A6,'=',I1/
     +         ' Specify ',A6,' in the range of [',I1,',',I1,']')
 2920 FORMAT (/' Input Error: ',A6,'=',I8/
     +         ' Specify ',A6,' in the recommended range of [',I3,',',I8,']'/
     +         ' Change PARAMETER ',A2,' if necessary')
C
      RETURN
      END
C
C -23----------------- END OF SUBROUTINE CHEOPT --------------------
```

APPENDIX B

CONTENTS OF THE ACCOMPANYING DISK

This report is accompanied by a 3.5 inch, high-density, IBM-PC-formatted floppy disk containing computer files as listed in the following:

- The Fortran code for program VBREAK is in file **vbreak.f**.

- An example of an input file as well as the associated primary output file are listed in **stive.inp** and **stive.doc**, respectively.

The above input file, **stive.inp**, initiates the simulation of Stive's 1980 test 1. The approximate CPU time on a Sun SPARC2 for thirty waves and a Courant number of 0.4 is 110 seconds.