NUMERICAL MODEL FOR DESIGN OF IMPERMEABLE COASTAL STRUCTURES

BY

NOBUHISA KOBAYASHI AND ANDOJO WURJANTO

Sponsored by
U.S. Army Coastal Engineering Research Center

RESEARCH REPORT NO. CE-89-75
JUNE, 1989

CENTER FOR APPLIED COASTAL RESEARCH
DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF DELAWARE
NEWARK, DELAWARE
19716

NUMERICAL MODEL FOR DESIGN OF IMPERMEABLE COASTAL STRUCTURES

BY

NOBUHISA KOBAYASHI AND ANDOJO WURJANTO

Sponsored by
U. S. Army Coastal Engineering Research Center

RESEARCH REPORT NO. CE-89-75
JUNE, 1989

CENTER FOR APPLIED COASTAL RESEARCH
DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF DELAWARE
NEWARK, DELAWARE
19716

ABSTRACT

A computer program called IBREAK is presented in this report. IBREAK may be used for the design of rough or smooth impermeable coastal structures of arbitrary geometry against normally incident waves. Any incident wave train can be specified as input at the seaward boundary of the computation domain, although the computation has been limited to monochromatic and transient waves. IBREAK computes the reflected wave train at the seaward boundary from which the reflection coefficient is predicted. For a subaerial structure, IBREAK computes wave runup on the seaward slope of the structure or wave overtopping over the crest of the structure if it is not high enough to prevent flow over its crest. For a submerged structure, IBREAK computes the transmitted wave train at the landward boundary of the computation domain from which the transmission coefficient is predicted. In addition to the equations of mass and momentum used to compute the flow field for any structure, an equation of energy is used in IBREAK to estimate the spatial variations of energy dissipation rates due to wave breaking and bottom friction. Furthermore, IBREAK computes the hydraulic stability and sliding motion of individual armor units under the action of the computed flow if the structure is protected with armor units. In addition to coastal structures, IBREAK can be used to predict the wave transformation in the surf zone and the resulting swash oscillation on a beach.

In this report, the equations and numerical procedures used in IBREAK are summarized by synthesizing the results published in various papers and reports. The essential parts of IBREAK are then explained to facilitate the use of IBREAK. The computer program IBREAK consists of the main program, 37

subroutines and one function, which are written in self-explanatory manners. The common parameters and variables are listed and explained herein since a large number of parameters and variables are involved in IBREAK. The input parameters and variables together with various options are detailed so as to minimize the misuse of IBREAK. The output parameters and variables are also explained in detail since the proper interpretation of the computed results is essential. Several examples are presented herein, although the computed results were already published in various papers and reports. These examples may be used to get familiarized with IBREAK. The limitations and capabilities of IBREAK are discussed throughout this report. Users of IBREAK are strongly recommended to read the related papers quoted in the references of this report so that the degree of accuracy of the computed results may be inferred. It is also recommended to calibrate and verify IBREAK if it is to be applied to new problems. Our experiences with IBREAK for the last few years suggest that IBREAK is a fairly versatile computer program although it does not describe the detailed flow field associated with plunging breakers.

ACKNOWLEDGEMENT

This is the third and final report resulting from research sponsored by the Coastal Engineering Research Center, U. S. Army Engineer Waterways Experiment Station under Contract No. DACW 3988P0628. The authors would like to thank J. P. Ahrens and D. D. Davidson for their enthusiastic support of our continued efforts to develop a computer program for rubble structure design.

TABLE OF CONTENTS

NUMERICAL MODEL FOR DESIGN OF IMPERMEABLE COASTAL STRUCTURES

PART I: INTRODUCTION

Background

In the planning and design of rubble mound breakwaters and riprap revetments in the coastal environment, the effects of incident wind waves on these structures must be considered. The stability and damage of a rubble mound breakwater or a riprap revetment under wave attack is obviously an important aspect in its design. Wave runup and overtopping on a coastal structure are other important factors since they normally determine the required height of the coastal structure. Wave reflection and transmission are other important aspects for designing rubble mound breakwaters which protect harbors or coastlines from wave attack.

The present design practices of rubble mound breakwaters and riprap revetments are essentially based on hydraulic model tests and empirical formulas in spite of the fact that a very large number of papers were published and reviewed in coastal engineering journals and breakwater conferences (e.g., Bruun, 1985). The empirical formula proposed by Hudson (1959) for estimating the required weight of an individual armor unit in the primary cover layer of a rubble structure is still widely used in spite of its serious limitations as discussed in the Shore Protection Manual (1984), hereafter referred to as SPM. Various attempts were made to improve the Hudson formula by empirically including the neglected effects such as wave period, slope permeability, slope geometry and randomness of wind waves (Ahrens and McCartney, 1975; Bruun and Johannesson, 1976; Kobayashi and Jacobs, 1985; Van der Meer, 1987, 1988). Wave reflection and runup on coastal structures were normally expressed as simple empirical functions of the surf similarity parameter (Battjes, 1974) which is useful for describing the gross characteristics of waves breaking or surging on uniform slopes (e.g., Ahrens and McCartney, 1975; Seelig, 1983; Ahrens and Martin, 1985). Empirical formulas for wave overtopping and transmission were also proposed (e.g., Cross and Sollitt, 1972; Weggel, 1976; Seelig, 1980; Jensen, 1983, 1984). Mathematical models appear to have been used only for linear wave transmission through porous rubble-mound breakwaters (Madsen and White, 1975; Sulisz, 1985).

Empirical formulas are limited to the specific structure geometry and wave conditions examined in the experiments and are not versatile enough to deal with various combinations of different coastal structures and incident waves. Hydraulic model tests are time-consuming and scale effects such as viscous effects make it difficult to interpret the model test results unless expensive large-scale tests are performed. Furthermore, measurements are normally limited to free surface oscillations and slope profiles because of the difficulties of measuring fluid velocities and hydrodynamic forces acting on armor units. Generation of the specified incident waves is difficult because of wave reflection at the wavemaker and generation of parasitic waves (e.g., Mansard, 1988).

The need for numerical models for the rubble structure design may become obvious if one considers extensive use of numerical models for the design of offshore structures. Numerical models attempt to simulate the complicated wave motion and armor response in an approximate manner. Numerical models can hence be used to improve our quantitative understanding of the detailed mechanics as well as to interpret the model test results. Furthermore, a hybrid approach based on empirical formulas, numerical models and hydraulic model tests will improve the efficiency and reliability of the design procedure.

Scope

This report synthesizes several computer programs developed previously for the projects funded by the National Science Foundation, the Delaware Sea Grant College Program, and the U.S. Army Coastal Engineering Research Center. A unified, user-friendly computer program called IBREAK is presented so that the single computer program with various computational options can be used for various coastal engineering problems. The capabilities and limitations of IBREAK are the same as those associated with the computer programs developed for the previous studies summarized in the following.

Kobayashi et al. (1986,1987) developed a numerical flow model to predict the flow characteristics on a rough impermeable slope for specified normally-incident monochromatic waves. The numerical flow model was based on the finite-amplitude shallow-water equations including the effects of bottom friction (Madsen and White, 1976) which were solved numerically in the time domain using an explicit dissipative Lax-Wendroff finite-difference method (Richtmyer and Morton, 1967; Hibberd and Peregrine, 1979; Packwood, 1980). A recent review of numerical methods developed for flows with shocks was given by Moretti (1987). The adopted numerical method is a shock-capturing method for which a separate treatment of a wave front (shock) is not required, although it can not describe the detailed behavior of waves plunging on the slope (e.g., Peregrine, 1983). The numerical flow model was developed in such a way that any incident wave train could be specified at the toe of the slope. The reflected wave train at the toe of the slope was computed from the characteristics advancing seaward. Wave runup and run-down were predicted from the computed oscillation of the instantaneous waterline on the slope. Comparison was made with available monochromatic wave test data for large-scale uniform riprap slopes (Ahrens, 1975; Ahrens and McCartney, 1975) and small-scale composite riprap slopes (Kobayashi and Jacobs, 1985). The numerical model was shown to predict wave runup, run-down and reflection well except for some uncertainties associated with the friction factor for the rough impermeable slopes, the quantitative definition of visually-measured wave runup and the seaward boundary condition used in the model.

Kobayashi and Otta (1987) developed a numerical stability model to predict the hydraulic stability and sliding motion of armor units on a rough impermeable slope under the action of specified normally-incident monochromatic waves. The drag, lift and inertia forces acting on an armor unit were expressed in terms of the fluid velocity and acceleration predicted separately using the numerical flow model. The numerical stability model predicts the variation of the local stability number along the slope whose minimum value corresponds to the critical stability number for initiation of armor movement. The critical stability number computed for available riprap tests was shown to be in good agreement with the observed zero-damage stability number (Ahrens, 1975; Kobayashi and Jacobs, 1985), although the lift coefficient used in the model was calibrated within a reasonable range (Sleath, 1984). The critical hydrodynamic conditions for the minimum armor stability were shown to be different for plunging, collapsing and surging waves.

Kobayashi and Greenwald (1986,1988) performed an experiment to calibrate and evaluate the developed numerical models in more detail. Eight test runs were conducted in a wave tank using a 1:3 glued gravel slope with an impermeable base. For each run with the specified monochromatic wave train generated in a burst, measurements were made of the free surface oscillation at the toe of the slope, the waterline oscillation on the slope, the temporal variations of dynamic pressure on the base of the slope and the displacements of loose gravel units placed on the glued gravel slope. The calibrated numerical models were shown to be capable of predicting the measured temporal variations of the hydrodynamic quantities and the measured spatial variations of the amount of the gravel movement.

Kobayashi and Watson (1987) applied the developed numerical flow model to predict wave reflection and runup on smooth impermeable slopes by adjusting the friction factor and the water depth specifying visually observed wave runup. Comparison with available empirical formulas (Seelig, 1983; Ahrens and Martin, 1985) indicated that the numerical flow model could also predict monochromatic wave reflection and runup on smooth slopes. Furthermore, the experiment conducted using the 1:3 glued gravel slope was repeated using a 1:3 plywood slope. The adjusted numerical flow model was shown to predict the measured temporal variations of the hydrodynamic quantities on the smooth slope as well. This comparison suggested that the numerical flow model developed for coastal structures might also be used to predict the flow characteristics in the swash zone on a beach. The applications of the numerical flow model for predicting the wave transformation and swash oscillation on beaches were presented by Kobayashi et al. (1988,1989), whereas the prediction of the sliding motion of individual sand particles was attempted by Kobayashi and DeSilva (1987).

Kobayashi and Wurjanto (1988,1989a) predicted the monochromatic wave overtopping over the crest of an impermeable coastal structure located on a sloping beach by modifying the numerical flow model. The modified model accounted for wave shoaling on the sloping beach in front of the structure located in relatively shallow water. The average overtopping rate per unit width was computed from the predicted temporal variations of the velocity and depth of the flow over the crest of the structure. The computed average overtopping rates were shown to be in agreement with the extensive small-scale test data of Saville (1955) for which smooth impermeable structures were fronted by a 1:10 slope. The numerical model also predicted the decrease of wave reflection due to the increase of wave overtopping.

Kobayashi and Wurjanto (1989b,1989c) and Wurjanto and Kobayashi (1989) predicted the monochromatic wave reflection and transmission over a submerged impermeable breakwater by modifying the numerical flow model. The modification was related to the landward boundary condition required for the transmitted wave propagating landward. In addition to the equations of mass and momentum used to compute the flow field, an equation of energy was derived to estimate the rate of energy dissipation due to wave breaking. The computed reflection and transmission coefficients were shown to be in agreement with the small-scale test data of Seelig (1980). The numerical model also predicted the spatial variation of the energy dissipation, the mean water level difference, and the time-averaged volume flux per unit width, although available measurements were not sufficient for evaluating the capabilities and limitations of the numerical model for predicting these quantities.

The numerical models discussed above are more versatile than empirical formulas but will need to be improved and expanded. Additional efforts will be required to include the effects of incident random waves since wind waves are random (e.g., Battjes, 1984; Goda, 1985) and wave grouping may affect the design of coastal structures (e.g., Johnson et al., 1978; Burcharth, 1979). Oblique and directional waves will also need to be considered especially for the head of a mound breakwater (e.g., Losada and Gimenez-Curto, 1982; Christensen et al., 1984). The permeability of core materials will need to be included for the wave transmission through a porous breakwater (Madsen and White, 1975; Sulisz, 1985) and for the analysis of the geotechnical stability of such a breakwater (Barends et al., 1983; Hannoura and McCorquodale, 1985). The stability of the toe of a breakwater (e.g., Eckert, 1983) may also be affected by the permeability of the breakwater and its foundation. Furthermore, the adjustment of the geometry of a mound structure will need to be predicted for the design of berm and reef breakwaters (e.g., Baird and Hall, 1984; Ahrens, 1984). For large slender concrete armor units, their structural strength and breakage will become important design considerations (e.g., Port Sines Investigating Panel, 1982). Finally, an economical analysis for the design of a mound structure (e.g., Smith, 1986) will need to be improved on the basis of the improved quantitative understanding of the hydrodynamics and armor response.

It should be stated that the limitations of the computer program IBREAK presented herein will eventually be overcome in view of the rapid progress of the numerical models discussed above. Furthermore, more researchers are expected to apply and expand numerical models (Allsop et al., 1988; Thompson, 1988). This report will make the computer program IBREAK easily accessible to other researchers and engineers.

PART II: NUMERICAL MODEL

Governing Equations

The wave motion on a rough or smooth impermeable slope is computed for the normally incident wave train specified at the seaward boundary of the computation domain as shown in Fig. 1 for the case of a rough slope. The prime indicates the dimensional variables in the following. The symbols shown in Fig. 1 are as follows: x' = horizontal coordinate taken to be positive landward with $x'=0$ at the seaward boundary; z' = vertical coordinate taken to be positive upward with $z'=0$ at the still water level (SWL); d_t' = water depth below SWL at the seaward boundary; θ' = local angle of the slope which may vary along the slope; η' = free surface elevation above SWL; h' = water depth above the impermeable slope; and u' = depth-averaged horizontal velocity.

For the flow over the impermeable slope, the vertically-integrated equations for mass and x'-momentum may be expressed as (Kobayashi et al., 1987)

$$\frac{\partial h'}{\partial t'} + \frac{\partial}{\partial x'} (h'u') = 0 \qquad (1)$$

$$\frac{\partial}{\partial t'} (h'u') + \frac{\partial}{\partial x'} (hu'^2) = -gh' \frac{\partial \eta'}{\partial x'} - \frac{1}{2} f'|u'|u' \qquad (2)$$

where t' = time; g = gravitational acceleration; and f' = constant friction factor related to the shear stress acting on the slope. In this simplified analysis, the constant friction factor f' accounts for the roughness characteristics of the impermeable slope surface. Moreover, the theoretical bed level for the flow over the rough impermeable slope is difficult to

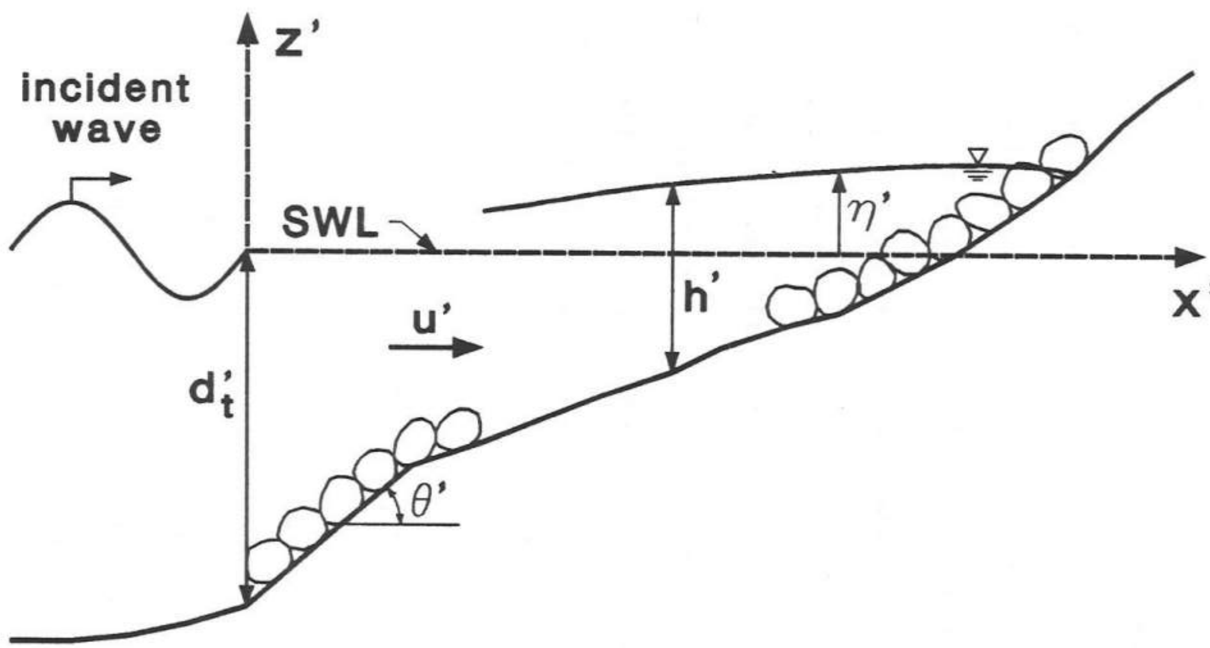


Figure 1. Wave Runup on a Rough Slope

pinpoint as is the case with oscillatory rough turbulent boundary layers (Jonsson, 1980).

The following dimensionless variables and parameters are introduced to normalize Eqs. 1 and 2:

$$t = t'/T_r' \quad ; \quad x = x'/[T_r'(gH_r')^{1/2}] \quad ; \quad u = u'/(gH_r')^{1/2} \tag{3}$$

$$z = z'/H_r' \quad ; \quad h = h'/H_r' \quad ; \quad \eta = \eta'/H_r' \quad ; \quad d_t = d_t'/H_r' \tag{4}$$

$$\sigma = T_r'(g/H_r')^{1/2} \quad ; \quad \theta = \sigma \tan\theta' \quad ; \quad f = \sigma f'/2 \tag{5}$$

where T_r' = representative wave period; H_r' = representative wave height; σ = dimensionless parameter related to wave steepness; θ = dimensionless gradient of the slope; and f = normalized friction factor. The representative wave period and height used for the normalization can be taken as the period and height used to characterize the incident wave for a particular problem. Substitution of Eqs. 3-5 into Eqs. 1 and 2 yields

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) = 0 \tag{6}$$

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}h^2\right) = -\theta h - f|u|u \tag{7}$$

where θ and f express the effects of the slope and friction, respectively. For a uniform slope, θ in Eq. 7 can be replaced by the surf similarity parameter, $\xi = \theta/(2\pi)^{1/2}$. In terms of the normalized coordinate system, the slope is located at

$$z = \int_0^x \theta\, dx - d_t \quad ; \quad x \geq 0 \tag{8}$$

which reduces to $z = (\theta x - d_t)$ for a uniform slope.

The initial time t=0 for the computation marching forward in time is taken to be the time when the specified incident wave train arrives at the seaward boundary located at x=0 as shown in Fig. 1. The initial conditions for the computation are thus given by $\eta=0$ and u=0 at t=0 in the region $x\geq0$. It is noted that h and η are uniquely related for given slope geometry expressed by Eq. 8.

In order to derive appropriate seaward and landward boundary conditions, Eqs. 6 and 7 are expressed in the following characteristic forms

$$\frac{\partial\alpha}{\partial t} + (u+c)\frac{\partial\alpha}{\partial x} = -\theta - \frac{f|u|u}{h} \quad ; \quad \frac{dx}{dt} = u + c \tag{9}$$

$$\frac{\partial\beta}{\partial t} + (u-c)\frac{\partial\beta}{\partial x} = \theta + \frac{f|u|u}{h} \quad ; \quad \frac{dx}{dt} = u - c \tag{10}$$

with

$$c = h^{1/2} \quad ; \quad \alpha = u + 2c \quad ; \quad \beta = -u + 2c \tag{11}$$

where α and β are the characteristic variables.

Assuming that u < c 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)

$$h = d_t + \eta_i(t) + \eta_r(t) \quad \text{at} \quad x = 0 \tag{12}$$

where η_i and η_r are the free surface variations normalized by H_r' at x=0 due to the incident and reflected waves, respectively. The incident wave train is specified by prescribing the variation of η_i 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

$$\eta_r(t) \simeq \frac{1}{2} d_t^{1/2} \beta(t) - d_t - C_t \qquad \text{at} \quad x = 0 \tag{13}$$

where β is given by Eq. 10. The correction term C_t in Eq. 13 introduced by Kobayashi et al. (1989) to predict wave set-down and set-up on a beach may be expressed as

$$C_t = \frac{1}{2} d_t^{1/2} \overline{(\eta - \bar{\eta})(u - \bar{u})} (\bar{h})^{-1} \qquad \text{at} \quad x = 0 \tag{14}$$

where the overbar denotes time averaging. For coastal structures, the nonlinear correction term C_t expressed by Eq. 14 is normally negligible and use may be made of $C_t = 0$.

The landward boundary condition of the numerical model depends on the crest height of a structure as will be explained in relation to the numerical procedures for wave runup, overtopping and transmission.

Numerical Method

Eqs. 6 and 7 are combined and expressed in the following vector form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \mathbf{G} = 0 \tag{15}$$

with

$$\mathbf{U} = \begin{bmatrix} m \\ h \end{bmatrix} \quad ; \quad \mathbf{F} = \begin{bmatrix} mu + 0.5h^2 \\ m \end{bmatrix} \quad ; \quad \mathbf{G} = \begin{bmatrix} \theta h + f|u|u \\ 0 \end{bmatrix} \tag{16}$$

where $m = uh$ is the normalized volume flux per unit width. The vectors **F** and **G** depend on the vector **U** for given θ and f.

Eq. 15 is discretized using a finite difference grid of constant space size Δx and constant time step Δt based on an explicit dissipative Lax-Wendroff method (e.g., Richtmeyer and Morton, 1967). In the following, the known quantities at the node located at $x=(j-1)\Delta x$ $(j=1,2,\ldots,s)$ and at the time $t=(n-1)\Delta t$ are indicated by the subscript j without a superscript. The integer s indicates the wet node next to the moving waterline at $t=(n-1)\Delta t$ for the case of wave runup and the node at the specified landward boundary for the case of wave overtopping or transmission. The unknown quantities at the node j and at the time $t=n\Delta t$ are denoted by the subscript j with the superscript * where the asterisk indicates the quantities at the next time level. The values of $\mathbf{U}_1^*$ and $\mathbf{U}_j^*$ for $j \geq (s-1)$ are computed using the seaward and landward boundary conditions, respectively. The values of $\mathbf{U}_j^*$ for $j=2,3,\ldots,(s-2)$ are computed using the known values of $\mathbf{U}_{j-1}$, $\mathbf{U}_j$ and $\mathbf{U}_{j+1}$ at the time $t=(n-1)\Delta t$ (Kobayashi et al., 1987)

$$\mathbf{U}_j^* = \mathbf{U}_j - \lambda\left[\frac{1}{2}(\mathbf{F}_{j+1} - \mathbf{F}_{j-1}) + \Delta x \mathbf{G}_j \right] + \frac{\lambda^2}{2} (\mathbf{g}_j - \mathbf{g}_{j-1} - \Delta x\, \mathbf{S}_j) + \mathbf{D}_j \tag{17}$$

where $\lambda = \Delta t/\Delta x$. The vector $\mathbf{g}_j$ in Eq. 17 is given by

$$\mathbf{g}_j = \frac{1}{2}\left(\mathbf{A}_{j+1} + \mathbf{A}_j\right)\left[\mathbf{F}_{j+1} - \mathbf{F}_j + \frac{\Delta x}{2}\left(\mathbf{G}_{j+1} + \mathbf{G}_j\right)\right] \tag{18}$$

with

$$\mathbf{A} = \begin{bmatrix} 2u & ; & (h - u^2) \\ 1 & ; & 0 \end{bmatrix} \tag{19}$$

The vector $\mathbf{S}_j$ in Eq. 17 is defined as

$$\mathbf{S}_j = \begin{bmatrix} \Delta x e_j - 0.5\, \theta_j (m_{j+1} - m_{j-1}) \\ 0 \end{bmatrix} \tag{20}$$

with

$$e_j = 2f|u_j|h_j^{-1} \left[\left(u_j^2 - h_j\right)\left(h_{j+1} - h_{j-1}\right)\left(2\Delta x\right)^{-1} - u_j\left(m_{j+1} - m_{j-1}\right)\left(2\Delta x\right)^{-1} - \theta_j h_j - f|u_j|u_j\right] \tag{21}$$

The vector D_j in Eq. 17 represents the additional term for damping high frequency parasitic waves, which tend to appear at the rear of a breaking wave, and is given by

$$D_j = \frac{\lambda}{2} \left[Q_j \left(U_{j+1} - U_j \right) - Q_{j-1} \left(U_j - U_{j-1} \right) \right] \tag{22}$$

with

$$Q_j = p_j \, I + \frac{1}{2} q_j \left(A_j + A_{j+1} \right) \tag{23}$$

where I = unit matrix; and the coefficients p_j and q_j are given by

$$p_j = \frac{1}{2} \left(c_j + c_{j+1} \right)^{-1} \left[\epsilon_2 |w_{j+1} - w_j| \left(v_j + v_{j+1} \right) - \epsilon_1 |v_{j+1} - v_j| \left(w_j + w_{j+1} \right) \right] \tag{24}$$

$$q_j = \left(c_j + c_{j+1} \right)^{-1} \left(\epsilon_1 |v_{j+1} - v_j| - \epsilon_2 |w_{j+1} - w_j| \right) \tag{25}$$

with

$$c = h^{1/2} \quad ; \quad v = u + c \quad ; \quad w = u - c \tag{26}$$

where ϵ_1 and ϵ_2 are the positive damping coefficients determining the amount of numerical damping of high frequency parasitic waves at the rear of a breaking wave.

The numerical stability criterion for this explicit finite difference method is given by (Packwood, 1980)

$$\frac{\Delta t}{\Delta x} < \left(|u_m| + c_m \right)^{-1} \left[\left(1 + \frac{\epsilon^2}{4} \right)^{1/2} - \frac{\epsilon}{2} \right] \tag{27}$$

where u_m = maximum value of u expected to be encountered in the flow field; c_m = maximum expected value of $h^{1/2}$; and ϵ = greatest coefficient of ϵ_1 and ϵ_2. The values of Δt, Δx, ϵ_1 and ϵ_2 need to be specified, considering the numerical stability criterion and desirable spatial and temporal accuracy as will be discussed in relation to the computer program IBREAK.

Incident Wave Profile

The normalized incident wave profile, $\eta_i(t) = \eta_i'(t')/H_r'$, with $t = t'/T_r'$ at the seaward boundary of the computation domain needs to be specified as input where H_r' and T_r' are the representative wave height and period used for the normalization in Eqs. 3-5. The temporal variation of $\eta_i(t)$ can be the measured incident wave profile at the seaward boundary in the absence of a coastal structure (Kobayashi and Greenwald, 1986,1988). If no data on the incident wave profile is available, an appropriate wave theory may be used to specify $\eta_i(t)$ for $t \geq 0$ such that $\eta_i = 0$ at $t = 0$ to be consistent with the assumed initial conditions of no wave action in the region of $x \geq 0$ at $t = 0$.

For the case of incident monochromatic waves, the computer program IBREAK uses cnoidal or Stokes second-order wave theory to specify the periodic variation of $\eta_i(t)$. The height and period of the incident monochromatic wave at the seaward boundary located at $x = 0$ are denoted by H' and T'. The reference wave period T_r' is taken as $T_r' = T'$ for the incident monochromatic wave. The reference wave height H_r' specified as input may be referred to deep water (Kobayashi and Wurjanto, 1988 and 1989a) or the location of wave measurement (Kobayashi et al, 1988). Since the numerical model is based on the finite-amplitude shallow-water equations given by Eqs. 1 and 2, the seaward boundary should be located in relatively shallow water. As a result, it is not always possible to take $H_r' = H'$. Defining $K_s = H'/H_r'$, the height and period of the monochromatic wave profile $\eta_i(t)$ at $x = 0$ is K_s and unity, respectively.

For Stokes second-order wave theory (e.g., SPM, 1984), the incident wave profile $\eta_i(t)$ at $x = 0$ is given by

$$\eta_i(t) = K_s \{0.5 \cos[2\pi(t+t_0)] + a_2 \cos[4\pi(t+t_0)]\} \qquad \text{for } t \geq 0 \tag{28}$$

with

$$a_2 = \frac{2\pi}{L} \cosh\left(\frac{2\pi}{L}\right)\left[2 + \cosh\left(\frac{4\pi}{L}\right)\right]\left[16\frac{d_t}{K_s} \sinh^3\left(\frac{2\pi}{L}\right)\right]^{-1} \tag{29}$$

$$L = L_0 \tanh\left(\frac{2\pi}{L}\right) \tag{30}$$

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; $L = L'/d_t'$ with L' = dimensional wavelength at $x = 0$; and $L_o = L_o'/d_t'$ with $L_o' = gT'^2/2\pi$ being the wavelength in deep water. The normalized wavelength L satisfying Eq. 30 for given L_o is computed using a Newton-Raphson iteration method. Eq. 29 yields the value of a_2 for given $d_t = d_t'/H_r'$, K_s and L. Since Eq. 28 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 $0 \leq (t+t_0) \leq 0.5$. Eq. 28 may be appropriate if the Ursell parameter $U_r < 26$ where $U_r = (H'L'^2/d_t'^3) = (K_sL^2/d_t)$ at $x = 0$. It is noted that the value of U_r based on the normalized wavelength L computed from Eq. 30 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 (e.g., Svendsen and Brink-Kjaer, 1972) is used to compute the incident wave profile $\eta_i(t)$ at $x = 0$

$$\eta_i(t) = \eta_{min} + K_s \, cn^2[2K(t+t_0)] \qquad \text{for } t \geq 0 \tag{31}$$

with

$$\eta_{min} = \frac{K_s}{m}\left(1 - \frac{E}{K}\right) - K_s \tag{32}$$

where η_{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; and m = parameter determining the complete elliptic integrals K(m) and E(m). The parameter m is related to the Ursell parameter U_r

$$U_r = \frac{K_s L^2}{d_t} = \frac{16}{3} mK^2 \tag{33}$$

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 is computed from

$$\frac{\sigma}{L d_t^{1/2}} \left[1 + \frac{K_s}{m d_t} \left(- m + 2 - 3 \frac{E}{K}\right)\right]^{1/2} - 1 = 0 \tag{34}$$

where the normalized wavelength L is given by Eq. 33 as a function of m for given d_t and K_s. The left hand side of Eq. 34 is a reasonably simple function of m in the range $0.8 < m < 1$. As a result, Eq. 34 can be solved using an iteration method which successively narrows down the range of m bracketing the root of Eq. 34. 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 Eq. 33, while Eq. 32 yields the value of η_{min}. The incident wave profile $\eta_i(t)$ is computed using Eq. 31 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 Eq. 28. 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).

Wave Reflection

The normalized reflected wave train $\eta_r(t)$ at the seaward boundary is computed using Eq. 13. It is also required to find the unknown value of the vector U_1^* at $x = 0$ and the time $t = n\Delta t$ which can not be computed using Eq. 17.

A simple first-order finite difference equation corresponding to Eq. 10 with $f = 0$ is used to find the value of β_1^* at $x = 0$ and the time $t = n\Delta t$

$$\beta_1^* = \beta_1 - \frac{\Delta t}{\Delta x} (u_1 - c_1)(\beta_2 - \beta_1) + \Delta t \theta_1 \qquad (35)$$

where $\beta_1 = (-u_1 + 2c_1)$ and $\beta_2 = (-u_2 + 2c_2)$. The right hand side of Eq. 35 can be computed for the known values of U_j with $j = 1$ and 2 at the time $t = (n-1)\Delta t$ where the spatial nodes are located at $x = (j-1)\Delta x$. The value of η_r at the time $t = n\Delta t$ is calculated using Eq. 13. Eq. 12 yields the value of h_1^*, while $u_1^* = [2(h_1^*)^{1/2} - \beta_1^*]$ using the definition of β given in Eq. 11. Thus, the values of h_1^*, u_1^* and $m_1^* = u_1^* h_1^*$ at $x = 0$ and $t = n\Delta t$ are obtained.

The nonlinear correction term C_t given by Eq. 14 needs to be estimated to compute $\eta_r(t)$ using Eq. 13. For incident monochromatic waves on gentle slopes, C_t may be estimated by (Kobayashi et al., 1988)

$$C_t = K_s^2 \, (16 d_t)^{-1} \qquad \text{for gentle slopes} \qquad (36)$$

where the assumptions of linear long wave and negligible wave reflection were made in Eq. 14 to derive Eq. 36. For coastal structures, wave reflection may not be negligible. It is hence suggested to choose the location of the seaward boundary so that C_t may be assumed to be $C_t \simeq 0$ for coastal structures. This assumption may be checked using Eq. 36 as a rough guideline.

The reflection coefficient for incident monochromatic waves may be estimated using the following equations (Kobayashi and Wurjanto, 1989b,1989c)

$$r_1 = \left[(\eta_r)_{max} - (\eta_r)_{min} \right] K_s^{-1} \tag{37}$$

$$r_2 = \left[\overline{\eta_r^2} \left(\overline{\eta_i^2} \right)^{-1} \right]^{1/2} \tag{38}$$

$$r_3 = \left[\overline{(\eta_r - \overline{\eta_r})^2} \left(\overline{\eta_i^2} \right)^{-1} \right]^{1/2} \tag{39}$$

where the subscripts max and min indicate the maximum and minimum values of $\eta_r(t)$ after the periodicity of $\eta_r(t)$ is established, whereas the overbar indicates the time averaging of the periodic variation. The normalized height of the periodic variation of $\eta_i(t)$ is equal to K_s. Eq. 37 is based on the normalized height of the reflected wave train as compared to that of the incident wave train. Eqs. 38 and 39 are based on the time-averaged reflected wave energy as compared to the time-averaged incident wave energy, $\overline{\eta_i^2} = K_s^2/8$, where the energy is estimated using linear wave theory. Eq. 39 accounts for the difference $\overline{\eta_r}$ between the still water level and the mean water level at x = 0 where $\eta_i(t)$ is specified such that $\overline{\eta_i} = 0$. The method used to compute the reflection coefficient should be consistent with the method used to estimate the reflection coefficient from measured free surface oscillations. If the temporal variations of $\eta_r(t)$ and $\eta_i(t)$ are sinusoidal, Eqs. 37 and 38 yield $r_1 = r_2$. It may be noted that Eqs. 38 and 39 may also be used for incident random waves if the irregular variations of $\eta_i(t)$ and $\eta_r(t)$ are used to compute the time-averaged values.

Wave Runup

For the case of no wave overtopping on a subaerial coastal structure as shown in Fig. 1, the landward boundary of the numerical model is located at the moving waterline on the slope where the water depth is essentially zero.

The kinematic boundary condition requires that the horizontal waterline velocity is the same as the horizontal fluid velocity. In reality, it is difficult to pinpoint the exact location of the moving waterline on the slope. For the computation, the waterline is defined as the location where the normalized instantaneous water depth equals a small value δ such as $\delta = 10^{-3}$.

The following numerical procedure dealing with the moving waterline located at $h = \delta$ is used to compute the values of U_j^* at the time $t = n\Delta t$ for the nodes $j \geq (s-1)$ which are not computed by Eq. 17. It is noted that the procedure is somewhat intuitive and may be improved since the moving waterline tends to cause numerical instability.

1. Compute $h_{s+1}=(2h_s-h_{s-1})$, $u_{s+1}=(2u_s-u_{s-1})$, and $m_{s+1}=h_{s+1}u_{s+1}$ at the time $t = (n-1)\Delta t$ where the integer s indicates the wet node next to the moving waterline at $t=(n-1)\Delta t$ such that $h_s > \delta$ and $h_{s+1} \leq \delta$.
2. Compute h_j^* and m_j^* at $t=n\Delta t$ for the nodes $j=(s-1)$ and s, using Eq. 17 without the damping term D_j since the water depth h can be very small at these nodes.
3. If $h_{s-1}^* \leq \delta$, the computation is aborted since the waterline should not move more than Δx because of the numerical stability criterion of the adopted explicit method given by Eq. 27. It is suggested to reduce Δt to avoid the numerical instability.
4. If $h_s^* > h_{s-1}^*$, 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 waterline decreases landward. The following adjustments are made: if $|u_s^*| > |u_{s-1}^*|$, set $u_s^* = 0.9\ u_{s-1}^*$; if $h_s^* < 0$, set $h_s^* = 0.5\ h_{s-1}^*$; and if $h_s^* > h_{s-1}^*$, set $h_s^* = 0.9\ h_{s-1}^*$. Then, $m_s^* = h_s^* u_s^*$ based on the adjusted values of h_s^* and u_s^*.

5. If $h_s^* \le \delta$, set $s^* = (s-1)$ and return where the integer s^* indicates the wet node next to the waterline at $t=n\Delta t$.

6. If $h_s^* > \delta$, compute $h_{s+1}^* = (2h_s^* - h_{s-1}^*)$, $u_{s+1}^* = (2u_s^* - u_{s-1}^*)$, and $m_{s+1}^* = h_{s+1}^* u_{s+1}^*$.

7. If $h_{s+1}^* \le \delta$, set $s^* = s$ and return.

8. If $h_{s+1}^* > \delta$, compute $\mathbf{U}_s^{**}$ at the time $t=(n+1)\Delta t$ using Eq. 17 without the damping term where $\mathbf{U}_j^*$ and $\mathbf{U}_j$ in Eq. 17 are replaced by $\mathbf{U}_s^{**}$ and $\mathbf{U}_s^*$, respectively. Improve the linearly extrapolated values in Step 6 using the following finite difference equations derived from Eqs. 6 and 7 with $f=0$:

$$m_{s+1}^* = m_{s-1}^* - \frac{\Delta x}{\Delta t} (h_s^{**} - h_s) \tag{40}$$

$$u_{s+1}^* = u_{s-1}^* - (u_s^*)^{-1} \left[\frac{\Delta x}{\Delta t} (u_s^{**} - u_s) + h_{s+1}^* - h_{s-1}^* + 2\Delta x \theta_s\right] \tag{41}$$

The upper limit of the absolute value of $(u_s^*)^{-1}$ in Eq. 41 is taken as δ^{-1} to avoid the division by the very small value. Calculate $h_{s+1}^* = m_{s+1}^* / u_{s+1}^*$

9. If $|u_{s+1}^*| \le \delta$, set $s^* = s$ and return.

10. If $h_{s+1}^* \le h_s^*$ and $h_{s+1}^* \le \delta$, set $s^* = s$ and return.

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

12. If $h_{s+1}^* > h_s^*$, the linearly extrapolated values of h_{s+1}^*, u_{s+1}^* and m_{s+1}^* in step 6 are adopted in the following instead of those computed in step 8. Furthermore, set $s^* = s$ if $h_{s+1}^* > h_s^*$ and $s^* = (s+1)$ if $h_{s+1}^* \le h_s^*$ where h_{s+1}^* is the adopted value given by $h_{s+1}^* = (2h_s^* - h_{s-1}^*)$.

13. Set $h_j^* = 0$, $u_j^* = 0$ and $m_j^* = 0$ for $j \ge (s^*+1)$ since no water is present above the computational waterline.

Once the normalized water depth h at the given time is known as a function of x, the normalized free surface elevation, $Z_r = Z_r'/H_r'$, where the physical water depth equals a specified value δ_r', can be computed as long as $\delta_r = (\delta_r'/H_r') > \delta$. The use of the physical depth δ_r' is related to the use of a waterline meter to measure the waterline oscillation on the slope (e.g., Kobayashi and Greenwald, 1986,1988). The specified depth δ_r' can be regarded as the vertical distance between the waterline meter and the slope, while the corresponding elevation Z_r' is the elevation above SWL of the intersection between the waterline meter 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. For incident monochromatic waves, the normalized runup R, run-down R_d and setup $\overline{Z_r}$ for given δ_r' are obtained from the computed periodic oscillation of $Z_r(t)$. The computed results such as those presented by Kobayashi et al. (1989) indicate that wave runup is insensitive to δ_r' but wave run-down is very sensitive to δ_r' since a thin layer of water remains on the slope during wave downrush. This implies that wave run-down is difficult to measure visually or using a waterline meter.

Wave Overtopping

Wave overtopping will occur if uprushing water reaches the landward edge of the crest of a subaerial structure as shown in Fig. 2 where $x_e' = x'$-coordinate of the landward edge of the crest. If wave overtopping occurs, the computation domain for the numerical model is limited to the region $0 \leq x \leq x_e$ where $x_e = x_e'/[T_r'(gH_r')^{1/2}]$ and the dimensionless variables defined in Eqs. 3-5

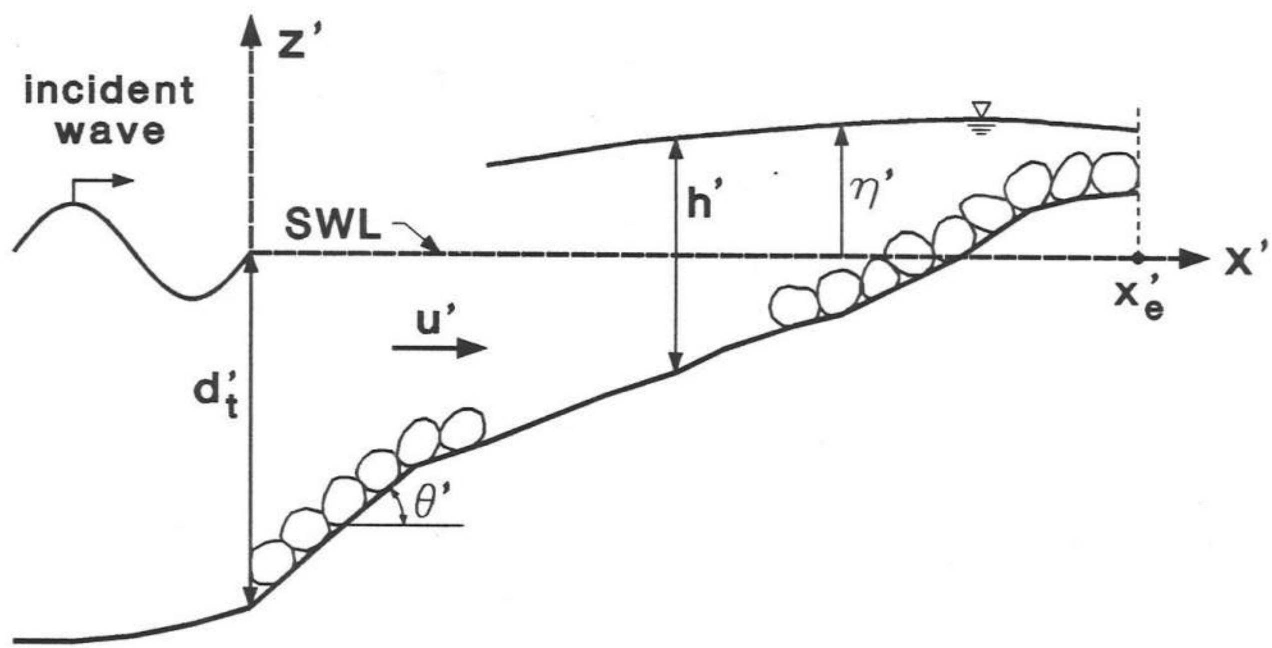


Figure 2. Wave Overtopping over a Subaerial Structure

remain the same. For the computation, wave overtopping is assumed to occur when the computed water depth h at $x = x_e$ becomes greater than the small value δ used for the location of the computational waterline on the slope. It is assumed that water flows over the landward edge freely. The flow approaching the landward edge can be supercritical as well as subcritical since the water depth at $x = x_e$ is relatively small.

For the computation starting from the initial conditions of no wave action in the computation domain, wave overtopping occurs when the integer s indicating the wet node next to the computational waterline at $t = (n-1)\Delta t$ becomes equal to the given integer j_e indicating the most landward node for the case of wave overtopping. When $s = j_e$ at $t = (n-1)\Delta t$, the values of U_j^* at $t = n\Delta t$ for the nodes $j = (s-1)$ and s are computed as follows:

1. Compute U_{s-1}^* using Eq. 17 with $j = (s-1)$ without the damping term D_{s-1} since the water depth h can be very small at this node.
2. If $u_{s-1} > c_{s-1}$ where $c_{s-1} = h_{s-1}^{1/2}$, the flow approaching the landward edge is supercritical, and both characteristics given by Eqs. 9 and 10 advance to the landward edge from the computation domain. Since Eqs. 9 and 10 are equivalent to Eqs. 6 and 7, use is made of the following finite difference equations derived from Eqs. 6 and 7 with $f = 0$:

$$h_s^* = h_s - \frac{\Delta t}{\Delta x} (m_s - m_{s-1}) \tag{42}$$

$$m_s^* = m_s - \frac{\Delta t}{\Delta x} \left[\left(m_s u_s + \frac{1}{2} h_s^2\right) - \left(m_{s-1} u_{s-1} + \frac{1}{2} h_{s-1}^2\right)\right] - \Delta t \theta_s h_s \tag{43}$$

   Then, $u_s^* = m_s^*/h_s^*$ and U_s^* is obtained.
3. If $u_{s-1} \le c_{s-1}$, the flow approaching the landward edge is subcritical or critical, and only the characteristics α given by Eq. 9 advances to the

landward edge from the computation domain. For this case, the flow at the landward edge node is assumed to be critical, that is, $u_s^* = c_s^*$ at $t = n\Delta t$. Use is made of the following finite difference equation derived from Eq. 9 with $f = 0$:

$$\alpha_s^* = \alpha_s - \frac{\Delta t}{\Delta x} (u_s + c_s) (\alpha_s - \alpha_{s-1}) - \Delta t \theta_s \qquad (44)$$

where $\alpha_s = (u_s + 2c_s)$ and $\alpha_{s-1} = (u_{s-1} + 2c_{s-1})$. Since $u_s^* = c_s^* = (h_s^*)^{1/2}$, $\alpha_s^* = (u_s^* + 2c_s^*) = 3u_s^*$. Thus, $u_s^* = \alpha_s^*/3$, $h_s^* = (u_s^*)^2$, $m_s^* = u_s^* h_s^*$ and U_s^* is obtained.

4. If $h_s^* \leq \delta$, wave overtopping is assumed to cease. Set $s* = (s-1)$ and return where the integer $s*$ indicates the wet node next to the waterline at $t = n\Delta t$.
5. If $h_s^* > \delta$, wave overtopping continues and $s* = j_e$.

For incident monochromatic waves, the normalized average overtopping rate per unit width, Q, is obtained from the computed temporal variation of $m = uh$ at $x = x_e$.

$$Q = Q'/[H_r'(gH_r')^{1/2}] = \overline{m} \qquad \text{at } x = x_e \qquad (45)$$

where the overbar indicates the time averaging of $m(t,x)$ at $x = x_e$ after its periodicity is established. Eq. 45 does not include the volume flux during the interval when $h \leq \delta$ at $x = x_e$ since the values of m at the nodes landward of the computational waterline are set to be zero during the computation. Eq. 45 can also be used to predict the value of Q for incident random waves for which the temporal variation of m at $x = x_e$ is irregular.

Wave Transmission

For wave transmission over a submerged breakwater, the landward boundary is always located at $x' = x'_e$ as shown in Fig. 3 where x'_e = x'-coordinate of the landward boundary which can be taken to be any convenient location such as the landward toe of the submerged breakwater. The computation domain for the numerical model is the fixed region $0 \le x \le x_e$ where $x_e = x'_e/[T'_r(gH'_r)^{1/2}]$ and the dimensionless variables defined in Eqs. 3-5 remain the same. To avoid the appearance of the waterline in the region $0 \le x \le x_e$, the normalized water depth h in the computation domain is taken as $h = \delta$ if the computed value of h becomes less than δ. It is assumed that the transmitted waves propagate landward without being reflected from the shoreline and the transmitted water flows landward without a return current. If the effects of the shoreline and return current need to be included, it will be required to extend the computation domain to the shoreline in a manner similar to the computations made by Kobayashi et al. (1988,1989) for the wave transformation over a shore-parallel bar and resulting swash oscillation on a beach.

Assuming that $u < c$ in the vicinity of the landward boundary located at $x = x_e$ where the normalized water depth below SWL is $d_e = d'_e/H'_r$, α and β represent the characteristics advancing landward and seaward, respectively, in the vicinity of the landward boundary. The boundary conditions at $x = x_e$ may then be expressed as (Kobayashi and Wurjanto, 1989b,1989c)

$$h = d_e + \eta_t(t) \qquad \text{at } x = x_e \qquad (46)$$

$$\eta_t(t) = \frac{1}{2} d_e^{1/2} \alpha(t) - d_e \qquad \text{at } x = x_e \qquad (47)$$

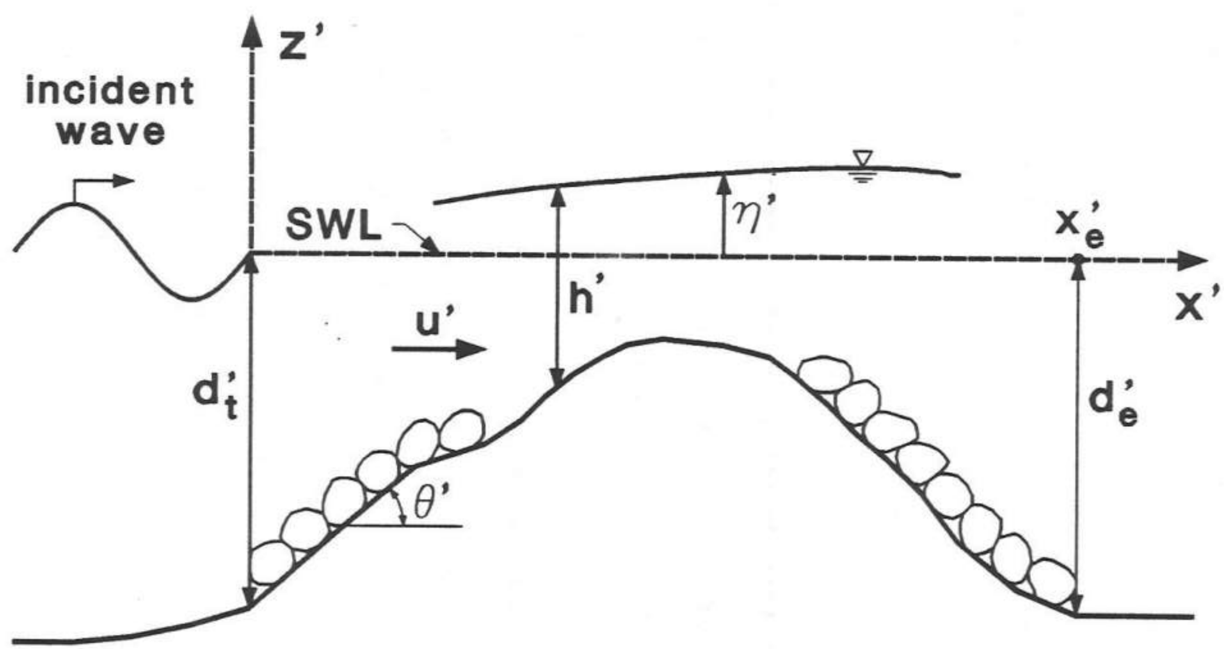


Figure 3. Wave Transmission over a Submerged Structure

where η_t is the free surface oscillation at $x = x_e$ normalized by H_r' due to the transmitted wave, provided that no wave propagates seaward from the region $x > x_e$. Eq. 47 expresses the transmitted wave train η_t in terms of the landward-advancing characteristic α given by Eq. 9 in a manner similar to Eq. 13 for the reflected wave train except that the nonlinear correction term is neglected in Eq. 47.

The following numerical procedure is used to compute the values of U_j^* at $t = n\Delta t$ for the nodes $j = (s-1)$ and s where the integer s for this case is the landward boundary node j_e located at $x = x_e$:

1. Compute U_{s-1}^* using Eq. 17 with $j = (s-1)$ with the damping term D_{s-1}since the water depth h is large at this node.
2. Compute α_s^* at $t = n\Delta t$ using Eq. 44.
3. Compute η_t^* at $t = n\Delta t$ using Eq. 47 with $\alpha = \alpha_s^*$. Then, $h_s^* = (d_e + \eta_t^*)$ from Eq. 46, while $u_s^* = [\alpha_s^* - 2(h_s^*)^{1/2}]$. Thus, $m_s^* = h_s^* u_s^*$ and U_s^* is obtained.

For incident monochromatic waves, the transmission coefficient associated with the computed periodic wave train $\eta_t(t)$ may be estimated using the following equations if $d_e = d_t$.

$$T_1 = [(\eta_t)_{max} - (\eta_t)_{min}] \, K_s^{-1} \tag{48}$$

$$T_2 = \left[\overline{\eta_t^2} \left(\overline{\eta_i^2} \right)^{-1} \right]^{1/2} \tag{49}$$

$$T_3 = \left[\overline{(\eta_t - \overline{\eta_t})^2} \left(\overline{\eta_i^2}\right)^{-1} \right]^{1/2} \tag{50}$$

Eqs. 48, 49 and 50 correspond to Eqs. 37, 38 and 39, respectively. Eq. 48 is based on the normalized height of the transmitted wave train as compared to that of the incident wave train. Eqs. 49 and 50 are based on the time-averaged transmitted wave energy as compared to the time-averaged incident wave energy, $\overline{\eta_i^2} = K_s^2/8$, based on linear wave theory. Eq. 50 accounts for the difference $\overline{\eta_t}$ between the still water level and the mean water level at $x = x_e$.

Wave Energy Balance

The normalized equations of mass and x-momentum given by Eqs. 6 and 7 are used to compute the flow field. The normalized energy equation corresponding to Eqs. 6 and 7 may be expressed as (Kobayashi and Wurjanto, 1989b,1989c)

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}(E_F) = -D_f - D_B \qquad (51)$$

with

$$E = \frac{1}{2}(hu^2 + \eta^2) \qquad \text{for } h > \eta \qquad (52a)$$

$$E = \frac{1}{2}[hu^2 + \eta^2 - (h-\eta)^2] \qquad \text{for } h < \eta \qquad (52b)$$

$$E_F = uh\left(\frac{u^2}{2} + \eta\right) \qquad (53)$$

$$D_f = f|u|u^2 \qquad (54)$$

where E = normalized specific energy defined as the sum of kinetic and potential energy per unit horizontal area; E_F = normalized energy flux per unit width; D_f = normalized rate of energy dissipation per unit horizontal area due to bottom friction; and D_B = normalized rate of energy dissipation per unit horizontal area due to wave breaking. The dimensional rate D_B' of energy dissipation due to wave breaking is given by $D_B' = (\rho g H_r'^2/T_r')D_B$ where ρ

= fluid density, which is assumed to be constant neglecting air bubbles. The normalized potential energy is taken to be relative to the normalized potential energy at t = 0 when the incident wave train arrives at x = 0 as shown in Figs. 1-3. Eqs. 52a and 52b are applicable for the portion of the structure below and above SWL, respectively.

Since the wave energy balance is normally analyzed in terms of the time-averaged quantities, the time-averaged dissipation rate, $\overline{D_B}$, due to wave breaking is computed using the time-averaged energy equation derived from Eq. 51

$$\overline{D_B} = - \frac{d}{dx} (\overline{E_F}) - \overline{D_f} \tag{55}$$

The present numerical model needs to predict that $\overline{D_B}$ is positive or zero depending on whether wave breaking occurs or not. The energy flux $\overline{E_F}$ should decrease with the increase of x, while $\overline{D_f} > 0$ since D_f defined in Eq. 54 is positive. It should be noted that Eq. 55 may be used even for the region which is not always exposed to water since h = 0 and u = 0 in the absence of water.

For the case of wave overtopping or transmission, integration of Eq. 55 from the seaward boundary to the landward boundary yields the time-averaged energy equation for the region $0 \leq x \leq x_e$

$$\overline{E_F}\ (x=0) - \overline{E_F}\ (x = x_e) = \int_0^{x_e} (\overline{D_f} + \overline{D_B})\ dx \tag{56}$$

where the first and second terms on the left hand side of Eq. 56 are the values of $\overline{E_F}$ at x = 0 and x = x_e, respectively. Eq. 56 implies that the difference between the net energy fluxes at the seaward and landward boundaries equals the rate of energy dissipation between the two boundaries.

For the case of wave runup on a slope, Eq. 56 needs to be modified such that $\overline{E_F}$ $(x=x_e) = 0$ and x_e should be interpreted as the maximum value of x reached by the waterline on the slope.

The specific energy $\overline{E}$ and the energy flux $\overline{E_F}$ at the seaward boundary where $\eta = (\eta_i + \eta_r)$ at $x = 0$ from Eq. 12 may approximately be given by (Kobayashi and Wurjanto, 1989b, 1989c)

$$\overline{E} \simeq \overline{\eta_i^2} + \overline{(\eta_r - \overline{\eta_r})^2} \qquad \text{at} \quad x = 0 \qquad (57)$$

$$\overline{E_F} \simeq d_t^{1/2} \left[\overline{\eta_i^2} - \overline{(\eta_r - \overline{\eta_r})^2} \right] \qquad \text{at} \quad x = 0 \qquad (58)$$

where $d_t^{1/2}$ is the normalized group velocity at $x = 0$ based on linear long wave theory. The reflection coefficient r_3 given by Eq. 39 including the effect of $\overline{\eta_r}$ is based on Eqs. 57 and 58. The reflection coefficient r_2 given by Eq. 38 corresponds to Eqs. 57 and 58 with $\overline{\eta_r} = 0$.

For the case of wave transmission over a submerged breakwater, the specific energy $\overline{E}$ and the energy flux $\overline{E_F}$ at the landward boundary where $\eta = \eta_t$ at $x = x_e$ may be approximated by

$$\overline{E} \simeq \overline{(\eta_t - \overline{\eta_t})^2} \qquad \text{at} \quad x = x_e \qquad (59)$$

$$\overline{E_F} \simeq d_e^{1/2}\, \overline{(\eta_t - \overline{\eta_t})^2} \qquad \text{at} \quad x = x_e \qquad (60)$$

where $d_e^{1/2}$ is the normalized group velocity at $x = x_e$ based on linear long wave theory. The transmission coefficient T_3 given by Eq. 50 for the case of $d_e = d_t$ is based on Eqs. 59 and 60, whereas the transmission coefficient T_2 given by Eq. 49 does not include the wave setup $\overline{\eta_t}$ at $x = x_e$.

Hydraulic Stability of Armor Units

The hydraulic stability of armor units is analyzed using the computed flow field on a rough impermeable slope. The drag, lift and inertia forces acting on individual armor units may be expressed in terms of the fluid velocity and acceleration on the rough impermeable slope. The normalized fluid acceleration, du/dt, is given by

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -\frac{\partial h}{\partial x} - \theta - \frac{f|u|u}{h} \tag{61}$$

where use is made of Eqs. 6 and 7.

Kobayashi and Otta (1987) expressed the stability condition against sliding or rolling of an armor unit located on the slope with its local slope angle θ' as shown in Fig. 1 in the following form:

$$|N_S + E_1| + E_2 N_S \le E_3 \tag{62}$$

which is applicable for the case of $u \neq 0$. The stability number N_S in Eq. 62 is defined as

$$N_S = H_r' \; (s_g - 1)^{-1} \left(\frac{W'}{\rho s_g}\right)^{-1/3} \tag{63}$$

where H_r' = reference wave height used for the normalization; s_g = specific gravity of the armor unit whose unit mass is given by ρs_g; and W' = median mass of the armor unit. If the stability number N_S required for the hydraulic stability of armor units is known, the required mass W' can be found using Eq. 63 for given H_r'. E_1, E_2 and E_3 in Eq. 62 are defined as

$$E_1 = \frac{2C_3^{2/3}}{C_2 C_D |u| u} \left[\frac{C_M}{(s_g - 1)\sigma} \frac{du}{dt} - \sin\theta'\right] \tag{64}$$

$$E_2 = C_L \tan\phi / C_D \tag{65}$$

$$E_3 = \frac{2C_3^{2/3}}{C_2 C_D u^2} \cos\theta' \tan\phi \tag{66}$$

where C_D, C_L and C_M are the drag, lift and inertia coefficients, respectively, while C_2, C_3 and ϕ are the area coefficient, volume coefficient and frictional angle of the armor unit, respectively. Eq. 62 can be solved in terms of N_s

$$N_s \leq N_R(t,x) = (E_1 + E_3)/(E_2 - 1); \quad \text{if } E_1 < 0,\ E_2 > 1 \text{ and } E_3 < (-E_1 E_2) \tag{67a}$$

$$N_s \leq N_R(t,x) = (E_3 - E_1)/(E_2 + 1); \quad \text{otherwise} \tag{67b}$$

where N_R = dimensionless function expressing the degree of the armor unit stability as a function of t and x. For the computation, Eqs. 67a and 67b are used if $|u| \geq 10^{-3}$ and N_R is set to be $N_R = 1000$ if $|u| < 10^{-3}$.

For the case of u = 0, Kobayashi and Otta (1987) expressed the stability condition in the form

$$\left| \frac{C_M}{(s_g - 1)\sigma} \frac{du}{dt} - \sin\theta' \right| \leq \cos\theta' \tan\phi \tag{68}$$

The condition given by Eq. 68 is satisfied if the normalized fluid acceleration remain within the following lower and upper bounds

$$\sigma\, a_{min} \leq du/dt \leq \sigma\, a_{max} \tag{69}$$

with

$$a_{min} \geq -\frac{s_g - 1}{C_M} \frac{\sin(\phi - \theta')}{\cos\phi} \quad ; \quad a_{max} \leq \frac{s_g - 1}{C_M} \frac{\sin(\phi + \theta')}{\cos\phi} \tag{70}$$

In terms of the dimensional variables, Eq. 69 can be rewritten as $g a_{min} \leq du'/dt' \leq g a_{max}$ where g = gravitational acceleration. The dimensionless parameters a_{min} and a_{max} need to be chosen so as to satisfy the conditions given by Eq. 70 as discussed by Kobayashi and Otta (1987).

The local stability number $N_{sx}(x)$ for initiation of armor movement at given location x is defined as the minimum value of $N_R(t,x)$ at the same location for a specified duration. For incident monochromatic waves, this duration can be taken as one wave period after the establishment of the periodicity of $N_R(t,x)$ with respect to t. If $N_s \le N_{sx}(x)$, the armor unit located at given x will not move during the specified duration. The critical stability number N_{sc} for initiation of armor movement is defined as the minimum value of $N_{sx}(x)$ with respect to x in the computation domain. If $N_s \le N_{sc}$, no armor units in the computation domain will move during the specified duration.

Movement of Armor Units

Kobayashi and Otta (1987) also performed a simplified analysis to predict the sliding motion of armor units when the criterion for initiation of armor movement is exceeded. In the following, the results presented by Kobayashi and Otta (1987) are rearranged so that the computer program attached in Appendix A may be understood without difficulties.

The normalized forces acting on an armor unit are separated into

$$F_D = \frac{C_D C_2}{2C_3 d} \sigma \; |u - u_a| \; (u - u_a) \tag{71}$$

$$F_L = \frac{C_L C_2}{2C_3 d} \sigma (u - u_a)^2 \tag{72}$$

$$F_I = C_M \frac{du}{dt} \tag{73}$$

$$W_c = \sigma\ (s_g - 1)\ \cos\theta' \qquad (74)$$

$$W_s = \sigma\ (s_g - 1)\ \sin\theta' \qquad (75)$$

with

$$d = \frac{d'}{H_r'} = (H_r')^{-1} \left(\frac{W'}{C_3 \rho s_g}\right)^{1/3} \quad ; \quad u_a = \frac{u_a'}{(gH_r')^{1/2}} \qquad (76)$$

where F_D = normalized drag force; F_L = normalized lift force; F_I = normalized inertia force due to the fluid acceleration only; W_c = component of the normalized submerged weight downward normal to the slope; W_s = component of the normalized submerged weight downward parallel to the slope; d = normalized length of the armor unit; and u_a = normalized velocity of the armor unit along the slope. The prime in Eq. 76 indicates the corresponding physical variable. It is simply assumed that the drag and inertia forces act upward or downward parallel to the slope, whereas the lift force acts upward normal to the slope. The normalized forces expressed by Eqs. 71-75 are based on the normalization by $(gW'/\sigma s_g)$. It is noted that the condition given by Eq. 69 is not imposed on the value of du/dt in Eq. 73 to account for possibly large fluid accelerations at the point of wave breaking.

The sliding motion of an armor unit starts if the following condition is satisfied

$$|F_D + F_I - W_s| > F_R \qquad (77)$$

with

$$F_R = (W_c - F_L)\ \tan\phi \geq 0 \qquad (78)$$

where F_R = magnitude of the normalized frictional force acting on the armor unit which is zero if $F_L \geq W_c$. In Eqs. 77 and 78, F_D and F_L are given by Eqs 71 and 72 with $u_a = 0$, respectively, where $u_a = 0$ for a stationary armor unit.

The normalized equation of the sliding motion of the armor unit moving with the normalized velocity u_a along the slope is given by

$$(s_g + C_m) \frac{du_a}{dt} = F_D + F_I - W_c - JF_R \tag{79}$$

with

$$J = u_a/|u_a| \tag{80}$$

where C_m = added mass coefficient given by $C_m = (C_M - 1)$ and F_R is assumed to act in the direction opposite to that of the armor movement. The displacement X_a' of the sliding unit along the slope from its initial location is normalized in the following two different ways:

$$X_a(t) = \frac{X_a'}{d'} = \frac{\sigma}{d} \int_{t_0}^{t} u_a \, dt \tag{81a}$$

$$X_{aa}(t) = \frac{X_a'}{T_r'(gH_r')^{1/2}} = \int_{t_0}^{t} u_a dt \tag{81b}$$

where t_0 = normalized time when the armor unit starts moving. Eq. 81a is used to estimate the degree of the armor movement relative its characteristic length d', whereas Eq. 81b is used to find the x-coordinate of the moving unit since the values of u and du/dt in Eqs. 71-73 should be those at the instantaneous location of the unit.

For the computation of the movement of individual armor units, the grid points used for the computation of the flow fields are used to specify the locations of the units before the armor movement computation. The movement of the armor unit located at the node j starts when the condition given by Eq. 77 is satisfied at the node j. If the armor unit located initially at the node j starts moving, a Lagrangian approach is used by tracking the location of the moving unit identified by its node number j. A forward difference equation in the time t derived from Eq. 79 is used to find the normalized velocity u_a of the identified unit whose instantaneous location is computed using Eq. 81b. The values of u and du/dt in Eq. 71-73 are evaluated at the node closest to the instantaneous location of the moving unit. The moving unit is assumed to stop when the condition given by Eq. 77 is not satisfied. The stopped unit resumes its movement when the condition given by Eq. 77 is satisfied. The temporal variation of X_a defined by Eq. 81a is also computed for the armor unit identified by its initial location on the slope.

PART III: COMPUTER PROGRAM IBREAK

Main Program

The computer program IBREAK attached in Appendix A consists of the main program, 37 subroutines and one function, which are written in self-explanatory manners. Double precision is used throughout the program. IBREAK has been written for a mainframe computer, IBM 3090-180E as well as Sun workstations (Sun 3/60), which operates diskless in conjunction with a central file server, Sun 4/280. Consequently, IBREAK may not have to be modified much for other computers. The computation time of IBREAK for one wave period is on the order of a minute for the mainframe computer and on the order of ten minutes for the Sun workstation.

The main program lists all the important variables and parameters in the common statements. Before the time-marching computation based on Eq. 17, the main program performs the following tasks:

- Open files and read input data using the subroutines OPENER, INPUT1 and INPUT2.
- Process the input data for the time-marching computation using the subroutines BOTTOM, PARAM, INIT1 and INWAV.
- Document the input and processed data using the subroutine DOC1

During the time-marching computation, the unknown quantities at the time $t = n\Delta t$ are computed from the known quantities at the time $t = (n-1)\Delta t$. The computational procedure during the time-marching computation is as follows:

- Estimate h_{s+1}, u_{s+1} and m_{s+1} at the time $t = (n-1)\Delta t$ by linear extrapolation for the case of wave runup where the integer s indicates the wet node next to the moving waterline at $t = (n-1)\Delta t$.

- Retain the values of the quantities at the time $t = (n-1)\Delta t$ which are required for the seaward and landward boundary computations.
- Compute $c_j = h_j^{1/2}$ for $j = 1,2,\ldots,s$ used in the characteristic equations given by Eqs. 9 and 10 with Eq. 11.
- Compute the unknown quantities at the time $t = n\Delta t$ using the subroutines MARCH, LANDBC and SEABC.
- Check the simplified condition of $|u| < (\Delta x/\Delta t)$ that will be satisfied if the numerical stability criterion given by Eq. 27 is satisfied.
- Compute the quantities related to wave energy balance at the request of a user.
- Compute the statistics of η, u and m=uh so that the mean, maximum and minimum values can be found after the time-marching computation.
- Compute the hydraulic stability of armor units or the movement of armor units at the request of a user.
- Store the computed results during the time-marching computation using the subroutine DOC2 at the request of a user.
- Write the time level n every 500 time steps and the value of the normalized time, $t = t'/T_r'$, whenever t is an integer.

After the time-marching computation, the following tasks are performed:

- Compute the statistics of the quantities related to the flow field and armor stability using the subroutine STAT2.
- Compute the overall balance of wave energy using the subroutine BALANE at the request of a user.
- Document the computed results using the subroutine DOC3.

Subroutines and Function

The 37 subroutines and one function arranged in numerical order in the computer program IBREAK are listed in Table 1. The page numbers for the subroutines and function listed in Table 1 correspond to the page numbers used for IBREAK attached in Appendix A. Each of the subroutines and function are explained concisely in the following:

1. OPENER: this subroutine opens input and output files. Some of the files are opened on the basis of options selected by a user.
2. INPUT1: this subroutine reads input data from the primary input data file and checks whether the options selected by a user are within the ranges available in IBREAK.
3. INPUT2: this subroutine reads the wave profile at the seaward boundary if the wave profile measured in a wave flume or generated numerically is specified as input.
4. BOTTOM: this subroutine computes the normalized structure geometry and the value of Δx from the dimensional structure geometry specified as input.
5. PARAM: this subroutine calculates the dimensionless parameters used in the other subroutines.
6. INIT1: this subroutine specifies the initial conditions given by $\eta = 0$ and $u = 0$ at $t = 0$ as well as the initial values of various quantities used for the subsequent computation.
7. INIT2: this subroutine facilitates the assignment of the initial values in the subroutine INIT1.

TABLE 1 - 37 Subroutines and One Function in Computer Program IBREAK

No.	Subroutine or Function	Page No. in IBREAK	No.	Subroutine or Function	Page No. in IBREAK
1	OPENER	7-11	20	MOVE	44-46
2	INPUT1	11-17	21	FORCES	46-47
3	INPUT2	17-18	22	ACCEL	47
4	BOTTOM	18-21	23	STAT2	47-49
5	PARAM	21-24	24	COEF	49-50
6	INIT1	24-26	25	BALANE	50-52
7	INIT2	26	26	MATAFG	52
8	INWAV	26-29	27	MATGJR	52-53
9	FINDM	29-30	28	MATS	53
10	CEL	30-31	29	MATD	53-54
11	SNCNDN	31-32	30	MATU	54-55
12	MARCH	32-34	31	ASSIGN	55
13	LANDBC	34-37	32	DERIV	55
14	RUNUP	37-38	33	DOC1	55-60
15	OVERT	38-39	34	DOC2	60-63
16	SEABC	39-41	35	DOC3	63-67
17	ENERGY	41-42	36	CHEPAR	67-68
18	STAT1	42	37	CHEOPT	68
19	STABNO	42-44	38	STOPP	68-69

8. INWAV: this subroutine computes the incident wave profile at the seaward boundary using Eq. 28 or 31 if the wave profile is not specified as input.

9. FINDM: this subroutine computes the value of the parameter m which satisfies Eq. 34.

10. CEL: this function computes the values of the complete elliptic integrals K and E used in Eqs. 31-34 for given m.

11. SNCNDN: this subroutine computes the Jacobian elliptic function cn used in Eq. 31.

12. MARCH: this subroutine performs the time-marching computation on the basis of Eq. 17.

13. LANDBC: this subroutine manages the landward boundary conditions for wave runup, overtopping or transmission as well as the computation of the normalized free surface elevation Z_r for given δ_r' as discussed in relation to wave runup.

14. RUNUP: this subroutine computes the waterline movement on the slope of a subaerial structure on the basis of the procedure discussed in relation to wave runup.

15. OVERT: this subroutine computes the overtopping flow at the landward edge of the crest of a subaerial structure on the basis of the procedure discussed in relation to wave overtopping.

16. SEABC: this subroutine computes the flow at the seaward boundary using Eq. 35 and the reflected wave train $\eta_r(t)$ using Eq. 13.

17. ENERGY: this subroutine computes the values of E, E_F and D_f defined by Eqs. 52, 53 and 54, respectively, in relation to the normalized equation of wave energy.

18. STAT1: this subroutine is used to calculate the sum, maximum and minimum values of quantities varying with time.

19. STABNO: this subroutine computes the armor stability function $N_R(t,x)$ using Eqs. 64-67 and the local stability number $N_{sx}(x)$ defined as the minimum value of $N_R(t,x)$ at given location.

20. MOVE: this subroutine computes the displacement of armor units using Eqs. 77-81.

21. FORCES: this subroutine computes the normalized forces given by Eqs. 71-75.

22. ACCEL: this subroutine computes the value of du/dt using Eq. 61.

23. STAT2: this subroutine finds the statistical values of the computed variables after the time-marching computation.

24. COEF: this subroutine computes the wave reflection coefficients given by Eqs. 37-39 as well as the wave transmission coefficients given by Eqs. 48-50 for the case of a submerged structure.

25. BALANE: this subroutine checks the balance of wave energy using Eqs. 55 and 56 as well as the approximate expressions given by Eqs. 58 and 60 based on linear long wave theory.

26. MATAFG: this subroutine computes the values of the elements of the matrix **A** defined by Eq. 19 and the vectors **F** and **G** defined in Eq. 16.

27. MATGJR: this subroutine computes the values of the elements of the vector **g** defined by Eq. 18.

28. MATS: this subroutine computes the values of the elements of the vector $\mathbf{S}_j$ at the node j defined by Eq. 20.

29. MATD: this subroutine computes the values of the elements of the vector $\mathbf{D}_j$ at the node j defined by Eq. 22.

30. MATU: this subroutine computes the values of the elements of the vector U_j^* at the node j and at the time $t = n\Delta t$ using Eq. 17.

31. ASSIGN: this subroutine changes a matrix to a vector or a vector to a matrix.

32. DERIV: this subroutine computes the first derivative of a function using a finite difference method.

33. DOC1: this subroutine documents the input data and dimensionless parameters before the time-marching computation.

34. DOC2: this subroutine stores some of the computed results at designated time levels during the time-marching computation.

35. DOC3: this subroutine documents the computed results after the time-marching computation.

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

37. CHEOPT: this subroutine checks whether the options selected by a user are within the ranges available in IBREAK.

38. STOPP: this subroutine executes a programmed stop if some of the input requirements for IBREAK are not satisfied.

Common Parameters and Variables

The parameters and variables included in the common statements in the main program are explained so that a user may be able to comprehend the computer program IBREAK and modify it if required. In the following, the common statements are explained one by one.

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

- N1R = N1= maximum number of grid points allowed in the computation domain where N1R = N1 = 500 in IBREAK.
- N2R = N2 = maximum number of time steps allowed for the time-marching computation where N2R = N2 = 30000 in IBREAK.
- N3R = N3 = maximum number of different values of the physical water depth δ'_r allowed for wave runup where N3R = N3 = 3 in IBREAK.
- N4R = N4 = maximum number of points allowed to specify the structure geometry consisting of linear segments where N4R = N4 = 100 in IBREAK.
- N5R = N5 = maximum number of time levels allowed for storing the spatial variations of the computed quantities as well as the maximum number of nodes allowed for storing the temporal variations of the computed quantities. N5R = N5 = 25 in IBREAK.

/CONSTA/ constants and input to the numerical model:

- PI = π = 3.141592
- GRAV = gravitational acceleration g = 9.81 m/s^2 or 32.2 ft/s^2.
- DELTA = normalized water depth δ used to define the computational waterline as discussed in relation to wave runup.
- X1 = damping coefficient ϵ_1 included in Eqs. 24 and 25.
- X2 = damping coefficient ϵ_2 included in Eqs. 24 and 25.

/ID/ integers used to specify the options of a user:

- IJOB = integer indicating the type of the landward boundary condition, where IJOB = 1 for wave runup on the seaward slope of a subaerial structure; IJOB = 2 for wave overtopping over a subaerial structure; and IJOB = 3 for wave transmission over a submerged structure.

- ISTAB = integer indicating the type of the armor analysis, where ISTAB = 0 for no computation of armor stability or movement; ISTAB = 1 for the computation of armor stability; and ISTAB = 2 for the computation of armor movement.
- ISYST = integer indicating the system of units, where ISYST = 1 for the International System of Units; and ISYST = 2 for the U.S. Customary System of Units.
- IBOT = integer indicating the type of input data for the structure geometry, where IBOT = 1 for the width and slope of linear segments; and IBOT = 2 for the locations of end points of linear segments.
- INONCT = integer indicating whether the nonlinear correction term C_t is included in Eq. 13 for $\eta_r(t)$, where INONCT = 0 for $C_t = 0$; and INONCT = 1 for C_t given by Eq. 36.
- IENERG = integer indicating whether the quantities related to wave energy are computed or not, where IENERG = 0 for no computation; and IENERG = 1 for the computation of the energy quantities discussed in relation to Eqs. 51-60.
- IWAVE = integer indicating the type of the wave profile specified at the seaward boundary, where IWAVE = 1 for the incident wave profile $\eta_i(t)$ computed using the subroutine INWAV; IWAVE = 2 for the incident wave profile $\eta_i(t)$ read in the subroutine INPUT2; and IWAVE = 3 for the total wave profile $[\eta_i(t) + \eta_r(t)]$ read in the

subroutine 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. However, this seaward boundary condition has been found to produce spurious long-period oscillations in the computation domain (Packwood, 1980).

- ISAVA = integer indicating the storage of the spatial variations of η and u in the computation domain at specified time levels if ISAVA = 1.
- ISAVB = integer indicating the storage of the temporal variation of h at specified nodes if ISAVB = 1.
- ISAVC = integer indicating the storage of the temporal variation of the displacement X_a given by Eq. 81a from specified initial nodal locations if ISAVC = 1.

/IDREQ/ **integers used for the special storage of the spatial variations of specified quantities:**

- IREQ = integer indicating the option of the special storage, where IREQ = 0 for no special storage; and IREQ = 1 for the special storage.
- IELEV = integer indicating the storage of the spatial variation of η at specified time levels if IELEV = 1.
- IV = integer indicating the storage of the spatial variation of u at specified time levels if IV = 1.
- IDUDT = integer indicating the storage of the spatial variation of du/dt given by Eq. 61 at specified time levels if IDUDT = 1.
- ISNR = integer indicating the storage of the spatial variation of N_R given by Eq. 67 at specified time levels if ISNR = 1.

- NNREQ = number of the specified time levels at which the spatial variations of the requested quantities are stored. It is required that $1 \leq$ NNREQ $\leq$ N5.
- NREQ(i) = specified time levels with i = 1,2,...,NNREQ at which the spatial variations of the requested quantities are stored.

/TLEVEL/ **integers indicating specific time levels and time steps:**

- NTOP = last time level for the time-marching computation performed for the duration $0 \leq t \leq \Delta t$ NTOP.
- NONE = even number of time steps in one wave period where NONE = $1/\Delta t$.
- NJUM1 = integer indicating the storage of the temporal variations of h and X_a every NJUM1 time steps if ISAVB = 1 and ISAVC = 1, respectively. NONE/NJUM1 must be an integer.
- NJUM2 = integer indicating the computation of N_R given by Eq. 67 every NJUM2 time steps if ISTAB = 1. NONE/NJUM2 must be an integer.
- NSAVA = time level at the beginning of the storage of the spatial variations of η and u if ISAVA = 1.
- NSTAB = time level when the computation of armor stability or movement begins if ISTAB = 1 or 2.
- NSTAT = time level at the beginning of the computation of the statistics such as the mean, maximum and minimum values after the establishment of the periodicity of the computed temporal variations.
- NTIMES = integer indicating the storage of the spatial variations of η and u NTIMES times at equal intervals from the time level NSAVA to the time level NTOP if ISAVA = 1.

/NODES/ **integers used to indicate specific nodal locations:**

- S = integer s indicating the wet node next to the moving waterline at $t = (n-1)\Delta t$ such that $h_s > \delta$ and $h_{s+1} \leq \delta$ except that S = JE for IJOB = 3 as discussed below. It is required that $1 \leq S \leq N1$.
- JE = integer indicating the most landward node j_e of the structure geometry specified as input, where it is required that JE ≤ N1. If IJOB = 1, S < JE all the time. If IJOB = 2, wave overtopping occurs when S = JE. If IJOB = 3, S is taken to be equal to the node JE at the landward boundary of the computation.
- JE1 = (JE−1)
- JSTAB = largest node number at each time level for which the computation of armor stability or movement is performed. JSTAB is taken as the largest node number based on DELTAR(1) in the subroutine LANDBC, although this can be modified easily.
- JMAX = largest value of S during the time levels from NSTAT to NTOP where JMAX < JE if IJOB = 1; JMAX = JE if IJOB = 2; and JMAX = JE = S if IJOB = 3.

/GRID/ **time step; grid size and related quantities:**

- T = constant time step Δt used for the discretization of Eq. 15.
- X = constant space size Δx used for the discretization of Eq. 15.
- TX = $\Delta t/\Delta x$
- XT = $\Delta x/\Delta t$
- TTX = $(\Delta t)^2/\Delta x$
- TTXX = $(\Delta t)^2/(\Delta x)^2$
- TWOX = $2\Delta x$

/WAVE1/ dimensional incident wave characteristics:

- HREFP = representative wave height H_r' introduced in Eqs. 3-5 for the normalization of the governing equations.
- TP = representative wave period T_r' introduced in Eqs. 3-5 for the normalization of the governing equations.
- WLOP = dimensional deep water wavelength L_o' defined as $L_o' = gT_r'^2/2\pi$.

/WAVE2/ dimensionless incident wave characteristics:

- KS = normalized wave height, $K_s = H'/H_r'$, at the seaward boundary located at $x = 0$ where H' = wave height at $x = 0$.
- KSREF = shoaling coefficient, H_r'/H_o', at the location where H_r' and T_r' are specified. It is noted that H_o' is the corresponding deep water wave height.
- KSSEA = shoaling coefficient, H'/H_o', at the seaward boundary. It is noted that IBREAK requires only KS = KSSEA/KSREF. If $H_r' = H'$, KS = 1 since KSSEA = KSREF.
- WLO = normalized deep water wavelength L_o given by $L_o = L_o'/d_t'$ where d_t' = water depth below SWL at the seaward boundary.
- WL = normalized wavelength L defined as $L = L'/d_t'$ with L' = dimensional wavelength at the seaward boundary based on linear wave theory except that the computed value of L is subsequently replaced by the value based on cnoidal wave theory if IWAVE = 1 and $U_r \geq 26$.
- UR = Ursell parameter U_r at the seaward boundary defined as $U_r = (H'L'^2/d_t'^3) = (K_sL^2/d_t)$ based on linear wave theory except that the computed value of U_r is subsequently replaced by the value based on cnoidal wave theory if IWAVE = 1 and $U_r \geq 26$.

• URPRE = value of U_r based on linear wave theory used to decide whether cnoidal or Stokes second-order wave theory is applied if IWAVE = 1.

• KSI = surf similarity parameter ξ defined as $\xi = \sigma \tan\theta'_\xi / \sqrt{2\pi} = \tan\theta'_\xi / (H'_r/L'_o)^{1/2}$ where $\tan\theta'_\xi$ = slope specified as input to define the surf similarity parameter for a specific structure.

• SIGMA = dimensionless parameter σ defined as $\sigma = T'_r(g/H'_r)^{1/2}$.

/WAVE3/ normalized free surface elevations as a function of t:

• ETA(n) = time series with n = time level and n ≤ N2 of the free surface profile at the seaward boundary which is specified in the subroutine INPUT2 if IWAVE ≥ 2 or computed in the subroutine INWAV if IWAVE = 1.

• ETAIS(n_s) = time series of the incident wave train η_i at x=0 stored such that $n_s = [(t/\Delta t) - \text{NSTAT} + 1]$ where t = normalized time. It is required that $n_s \leq$ N2.

• ETARS(n_s) = time series of the reflected wave train η_r at x=0 stored in the same way as η_i at x=0.

• ETATS (n_s) = time series of the transmitted wave train η_t at $x=x_e$ stored in the same way as η_i at x=0 if IJOB = 3.

/WAVE4/ normalized maximum and minimum free surface elevations:

• ETAMAX = maximum value of the specified or computed time series ETA(n).

• ETAMIN = minimum value of the specified or computed time series ETA(n).

/WAVE5/ parameters related to cnoidal wave theory:

- KCNO = complete elliptic integral of the first kind, K(m), used in Eqs. 31-34.
- ECNO = complete elliptic integral of the second kind, E(m), used in Eqs. 32-34.
- MCNO = parameter m computed from Eq. 34.
- KC2 = value of (1–m) used to compute the values of K(m) and E(m) using the function CEL.

/BOT1/ dimensional parameters related to the structure:

- DSEAP = water depth d_t' below SWL at the seaward boundary.
- DLANDP = water depth d_e' below SWL at the landward boundary used for IJOB = 3 only.
- FWP = constant friction factor f′ used in Eq. 2.

/BOT2/ normalized parameters related to the structure:

- DSEA = normalized water depth, $d_t = d_t'/H_r'$, at the seaward boundary.
- DSEAKS = d_t/K_s corresponding to the value of d_t'/H'.
- DSEA2 = normalized group velocity $d_t^{1/2}$ at x = 0 based on linear long wave theory.
- DLAND = normalized water depth, $d_e = d_e'/H_r'$, at the landward boundary for IJOB = 3.
- DLAND2 = normalized group velocity $d_e^{1/2}$ at $x=x_e$ based on linear long wave theory.
- FW = normalized friction factor f given by $f = \sigma f'/2$ and introduced in Eq. 7.
- TSLOPS = slope $\tan\theta_\xi'$ used to define the surf similarity parameter ξ.

• WTOT = normalized horizontal width, $(j_e-1)\Delta x$, of the computation domain.

/BOT3/ vectors related to the normalized structure geometry:

• U2INIT(j) = normalized water depth below SWL at the node j with j=1,2,...,JE located on the structure surface, corresponding to the value of $-z$ where z is given by Eq. 8.

• THETA(j) = dimensionless gradient of the slope, θ_j, at the node j where θ is defined in Eq. 5.

• SSLOPE(j) = $\sin\theta'_j$ where θ'_j = local angle of the slope at the node j required for the computation of armor stability and movement.

• XB(j) = normalized x-coordinate of the node j located on the structure surface.

• ZB(j) = normalized z-coordinate of the node j, corresponding to the normalized structure geometry given by Eq. 8.

/BOT4/ input parameter for the structure geometry:

• NBSEG = number of linear segments of different inclinations used to specify the structure geometry. It is required that $1 \leq$ NBSEG $\leq$ (N4−1).

/BOT5/ dimensional quantities associated with linear segments of the structure:

• WBSEG(i) = horizontal width of the segment i with i=1,2,...,NBSEG where the segment number i increases landward.

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

- XBSEG(i) = horizontal distance from the seaward boundary located at x'=0 to the seaward end of the segment i.
- ZBSEG(i) = vertical distance below SWL at the seaward end of the segment i which is negative if the end point is located above SWL.

/HYDRO/ hydrodynamic quantities computed by the numerical model:

- U(k,j) =values of the components of the vector $\mathbf{U}_j$ defined in Eq. 16 at the node j such that U(1,j) = m_j and U(2,j) = h_j where m = normalized volume flux per unit width; and h = normalized water depth below the instantaneous free surface.
- V(j) = value of the normalized depth-averaged velocity, $u_j = m_j/h_j$, at the node j.
- ELEV(j) = value of the normalized free surface elevation η_j above SWL at the node j.
- C(j) = value of $c_j = h_j^{1/2}$ defined in Eq. 11 at the node j.
- DUDT(j) = value of the normalized fluid acceleration du/dt given by Eq. 61 at the node j.

/MATRIX/ elements of matrices used in the numerical model:

- A1(k,j) = values of the elements of the first row of the matrix $\mathbf{A}_j$ defined by Eq. 19 at the node j such that A1(1,j) = $2u_j$ and A1(2,j) = $(h_j - u_j^2)$.
- F(k,j) = values of the elements of the vector $\mathbf{F}_j$ defined in Eq. 16 at the node j such that F(1,j) = $(m_j u_j + 0.5h_j^2)$ and F(2,j) = m_j.
- G1(j) = value of the non-zero element of the vector $\mathbf{G}_j$ defined in Eq. 16 at the node j such that G1(j) = $\theta_j h_j + f|u_j|u_j$.

• GJR(k,j) = values of the elements with k=1 and 2 of the vector g_j defined by Eq. 18 at the node j.

• S1(j) = value of the non-zero element of the vector S_j given by Eq. 20 at the node j such that S1(j) = $\Delta x e_j - 0.5\ \theta_j\ (m_{j+1} - m_{j-1})$ where e_j is given by Eq. 21.

• D(k,j) = values of the elements with k=1 and 2 of the vector D_j defined by Eq. 22 at the node j.

/RUNP1/ input parameter for wave runup computation:

• NDELR = number of different values of the physical water depth δ_r' associated with the measured or visual waterline for which the normalized free surface elevation Z_r is computed as discussed in relation to wave runup. It is required that 1 ≤ NDELR ≤ N3.

/RUNP2/ quantities related to wave runup:

• DELRP(i) = different values of δ_r' with 1 ≤ i ≤ NDELR specified as input such that DELRP(i) < DELRP(i+1).

• DELTAR(i) = normalized water depth, $\delta_r = \delta_r'/H_r'$, corresponding to the different values of δ_r'. It is required that $\delta_r \geq \delta$ where the small value δ is used to define the computational waterline.

• RUNUPS(i) = normalized free surface elevation Z_r above SWL at the location of $h=\delta_r$ at each time level, where RUNUPS(i) corresponds to DELTAR(i).

• RSTAT(k,i) = mean, maximum and minimum values of RUNUPS(i) during the time levels between NSTAT and NTOP indicated by k=1,2 and 3, respectively.

/OVER/ quantities related to wave overtopping:

- OV(k) = quantities calculated from the computed values of m at $x=x_e$ during the time levels between NSTAT and NTOP if IJOB=2, where OV(1) = Q given by Eq. 45; OV(2) = normalized time when the value of m at $x=x_e$ is the maximum; OV(3) = normalized duration of wave overtopping; and OV(4) = maximum value of m at $x=x_e$. The normalized time and duration are taken to be relative to the duration from the time level NSTAT to the time level NTOP.

/COEFS/ reflection and transmission coefficients:

- RCOEF(k) = wave reflection coefficients defined by Eqs. 37, 38 and 39 for k=1, 2 and 3, respectively.
- TCOEF(k) = wave transmission coefficients defined by Eqs. 48, 49 and 50 for k=1, 2 and 3, respectively.

/STAT/ statistics of the hydrodynamic quantities computed during the time levels between NSTAT and NTOP:

- ELSTAT(i) = time-averaged values $\overline{\eta_i}$, $\overline{\eta_r}$ and $\overline{\eta_t}$ calculated from ETAIS(n_s), ETARS(n_s) and ETATS(n_s) for i=1, 2 and 3, respectively.
- U1STAT(j) = mean value of m_j at the node j.
- ESTAT(k,j) = mean, maximum and minimum values of η_j at the node j indicated by k=1, 2 and 3, respectively.
- VSTAT(k,j) = mean, maximum and minimum values of u_j at the node j indicated by k=1, 2 and 3, respectively.

/ENERG/ **quantities related to wave energy:**

- ENER(i,j) = quantities related to the time-averaged energy equation expressed by Eq. 55 which are computed during the time levels between NSTAT and NTOP, where ENER(k,j) with i=1, 2, 3 and 4 correspond to the values of $\overline{E}$, $\overline{E_F}$, $\overline{D_f}$ and $\overline{D_B}$ at the node j, respectively. $\overline{E}$, $\overline{E_F}$ and $\overline{D_f}$ are the time-averaged values of E, E_F and D_f given by Eqs. 52, 53 and 54, respectively, whereas $\overline{D_B}$ is computed using Eq. 55.

- ENERB(k) = quantities related to the time-averaged energy balance in the computation domain expressed by Eq. 56, where ENERB(1) = $\overline{E_F}(x=0)$; ENERB(2) = $\overline{E_F}(x=x_e)$ which is zero if IJOB = 1; ENERB(3) = $\int_o^{x_e} \overline{D_f}$ dx; ENERB(4) = $\int_o^{x_e} \overline{D_B}$ dx; ENERB(5) = left hand side of Eq. 56; ENERB(6) = right hand side of Eq. 56, ENERB(7) = difference between the right and left hand sides of Eq. 56; ENERB (8) = percentage error defined as 100 × ENERB(7)/ENERB(5); ENERB(9) = $d_t^{1/2}\ \overline{\eta_i^2}$; ENERB(10) = $d_t^{1/2}\ \overline{(\eta_r - \overline{\eta_r})^2}$; ENERB(11) = $d_e^{1/2}\ \overline{(\eta_t - \overline{\eta_t})^2}$ only for IJOB = 3; ENERB(12) = right hand side of Eq. 58; ENERB(13) = percentage error in the approximate expression given by Eq. 58; and ENERB(14) = percentage error in the approximate expression given by Eq. 60. These percentage errors may be used to estimate the uncertainties associated with the computed reflection and transmission coefficients.

/STAB1/ **input parameters related to armor stability and movement:**

- C2 = area coefficient C_2 of the armor unit.
- C3 = volume coefficient C_3 of the armor unit.

- CD = drag coefficient C_D used in Eq. 71.
- CL = lift coefficient C_L used in Eq. 72.
- CM = inertia coefficient C_M used in Eq. 73.
- SG = specific gravity s_g of the armor unit.
- TANPH1 = $\tan\phi$ with ϕ = frictional angle of the armor unit.
- AMIN = parameter a_{min} specified as input. The condition for a_{min} given in Eq. 70 needs to be satisfied if ISTAB = 1.
- AMAX = parameter a_{max} specified as input. The condition for a_{max} given by Eq. 70 needs to be satisfied if ISTAB = 1.
- DAP = characteristic length d' of the armor unit defined in Eq. 76 which needs to be specified as input if ISTAB = 2.

/STAB2/ computed parameters related to armor stability and movement:

- SG1 = $(s_g - 1)$ used in Eqs. 74 and 75.
- CTAN(j) = value of $\cos\theta_j' \tan\phi$ at the node j where θ_j' = local angle of the slope at the node j.

/STAB3/ armor stability parameters used in the subroutine STABNO:

- CSTAB1 = $2\ C_3^{2/3}/(C_2 C_D)$ used in Eq. 64.
- CSTAB2 = $C_M/[(s_g - 1)\ \sigma]$ used in Eq. 64.
- AMAXS = $\sigma\ a_{max}$ used in Eq. 69.
- AMINS = $\sigma\ a_{min}$ used in Eq. 69.
- E2 = $C_L \tan\phi/C_D$ used in Eq. 65.
- E3PRE(j) = value of $2\ C_3^{2/3} \cos\theta_j' \tan\phi/(C_2 C_D)$ at the node j used in Eq. 66.

/STAB4/ armor movement parameters used in the subroutine MOVE:

- CSTAB3 = $C_2 C_D/(2C_3 d)$ used in Eq. 71.
- CSTAB4 = $C_2 C_L/(2C_3 d)$ used in Eq. 72.
- CM1 = $(C_M - 1)$ = added mass coefficient C_m used in Eq. 79.
- DA = normalized characteristic length d of the armor unit defined in Eq. 76.
- SIGDA = σ/d used in Eq. 81a.
- WEIG = $\sigma(s_g - 1)$ used in Eqs. 74 and 75.

/STAB5/ node number and time levels related to armor stability:

- JSNSC = node number j where the critical stability number N_{sc} for initiation of armor movement occurs.
- NSNSC = time level n when the critical stability number N_{sc} for initiation of armor movement occurs.
- NSNSX(j) = time level n when the local stability number $N_{sx}(x)$ at the node j occurs.

/STAB6/ stability numbers for initiation of armor movement:

- SNSC = critical stability number N_{sc} defined as the minimum value of $N_{sx}(x)$ with respect to x.
- SNR(j) = value of armor stability function $N_R(t,x)$ at the node j and at the time $t = n\Delta t$ computed during the time levels from n = NSTAB to n = NTOP if ISTAB = 1.
- SNSX(j) = value of the local stability number $N_{sx}(x)$ at the node j defined as the minimum value of $N_R(t,x)$ at the node j.

/STAB7/ **integers related to armor movement:**

- NMOVE = number of armor units moved from their initial locations where armor movement is computed during the time levels between NSTAB and NTOP if ISTAB = 2.
- NSTOP = number of armor units stopped after their movement.
- ISTATE(j) = integer indicating the state of the armor unit initially located at the node j, where ISTATE(j) = 0, 1 or 2 depending on whether the armor unit is stationary, moving or stopped, respectively.
- NODIN(j) = node number at the initial location of the armor unit which must be equal to the node number j.
- NODFI(j) = node number closest to the armor unit at the end of each time step where each armor unit is identified by the node number j at the initial location of the armor unit.
- NDIS(j) = time level n when the armor unit identified by the node number j has started moving.

/STAB8/ **normalized velocity and displacement of moving or stopped armor units:**

- VA(j) = normalized velocity u_a of the armor unit located initially at the node j which is computed using Eq. 79.
- XAA(j) = normalized displacement X_{aa} defined by Eq. 81b of the armor unit from its initial location at the node j.
- XA(j) = normalized displacement X_a defined by Eq. 81a of the armor unit from its initial location at the node j.

/FILES/ file names and associated node numbers for ISAVB = 1 and ISAVC = 1:

- NNOD1 = number of nodes where the temporal variation of h is stored if ISAVB = 1. It is required that 1 ≤ NNOD1 ≤ N5.
- NNOD2 = number of nodes for which the temporal variation of the normalized displacement X_a is stored if ISAVC = 1. It is required that 1 ≤ NNOD2 ≤ N5.
- NODNO1(i) = i-the node number with i=1,2,...,NNOD1 where the temporal variation of h is stored.
- NODNO2(k) = k-the node number with k=1,2,...,NNOD2 for which the temporal variation of X_a is stored.
- FNAME1(i) = file name associated with NODNO1(i).
- FNAME2(k) = file name associated with NODNO2(k).

/VALUEN/ values at the time t = (n−1)Δt stored at the beginning of each time step:

- VSN = u_s used at the landward boundary computation.
- USN(i) = U_s such that USN(1) = m_s and USN(2) = h_s used at the landward boundary computation.
- VMN = u_{s-1} used at the landward boundary computation.
- UMN(1) = m_{s-1} used at the landward boundary computation.
- V1N = u_1 used in Eq. 35.
- V2N = u_2 used in Eq. 35.

Input Parameters and Variables

The input parameters and variables are summaries in the sequence of the data input in IBREAK. First, the name of the primary input data file, FINP1, is read in the Main Program as follows:

```
      WRITE(*,*) 'Name of Primary Input-Data-File?'
      READ(*,5000) FINP1
5000  FORMAT(A20)
```

where some of the write statements are discussed here for convenience.

Almost all the input data are read in the subroutine INPUT1 from the input data file with its unit number = 11 and its file name = FINP1 as explained in the subroutine OPENER. The data input in the subroutine INPUT1 begins with the following comment lines:

```
      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
```

where NLINES = number of the comment lines proceeding the input data. The unit numbers 28 and 29 correspond to the file names ODOC and OMSG in the subroutine OPENER. The file ODOC stores the essential output for the concise documentation, while the file OMSG stores the messages written under special circumstances during the computation. The format statements used in the subroutine INPUT1 are listed in Table 2.

TABLE 2 - Format Statements Used in Subroutine INPUT1.

```
1110  FORMAT  (I8)
1120  FORMAT  (14A5)
1130  FORMAT  (2I1, I8)
1140  FORMAT  (I1)
1150  FORMAT  (I1, 2X, A20)
1160  FORMAT  (3I1, I8, 3I4)
1170  FORMAT  (5I1, I6)
1180  FORMAT  (3F13.6)
1190  FORMAT  (I6, 2X, A20)
```

The integers indicating the user's options and the related time levels and node locations are then read in the subroutine INPUT1.

```
READ(11,1130)  IJOB, ISTAB, NSTAB
READ(11,1140)  ISYST
READ(11,1140)  IBOT
READ(11,1140)  INONCT
READ(11,1140)  IENERG
READ(11,1150)  IWAVE, FINP2
READ(11,1160)  ISAVA, ISAVB, ISAVC, NSAVA, NTIMES, NNOD1, NNOD2
READ(11,1170)  IREQ, IELEV, IV, IDUDT, ISNR, NNREQ
```

where IJOB, ISTAB, ISYST, IBOT, INONCT, IENERG, IWAVE, ISAVA, ISAVB and ISAVC are explained in the common /ID/, whereas NSTAB, NSAVA and NTIMES are discussed in the common /TLEVEL/. On the other hand, NNOD1 and NNOD2 are explained in the common /FILES/, while IREQ, IELEV, IV, IDUDT, ISNR and NNREQ are discussed in the common /IDREQ/. If IWAVE=2 or 3, the file name FINP2 containing the data on the wave profile at the seaward boundary needs to be specified as input. The specified options are checked in the subroutine CHEOPT which writes the appropriate error message and correction instruction for each input parameter. If IREQ=1, at least one of IELEV, IV, IDUDT and ISNR must be unity. Furthermore, IDUDT=0 if ISTAB=0 and ISNR=0 if ISTAB $\neq$ 1. If IREQ=1, the following input is required:

```
READ(11,1110)  (NREQ(I), I=1, NNREQ)
```

where NREQ is explained in the common /IDREQ/.

Some of the integers included in the common /TLEVEL/ are read as input

```
READ(11,1110)  NTOP
READ(11,1110)  NONE
READ(11,1110)  NJUM1
```

The value of NSTAT is taken as NSTAT = (NTOP − NONE + 1) if IWAVE=1 corresponding to incident monochromatic waves, and NSTAT = NSAVA if IWAVE = 2 or 3. Furthermore, if ISAVA=0, NSAVA = (NTOP + 1) which has the same effect as ISAVA=0. Likewise, NSTAB = (NTOP + 1) if ISTAB=0. For incident monochromatic waves with IWAVE=1, use has been made of NTOP = NONE × (t_p + 1), NSTAB = (NTOP − NONE + 1), NSAVA = (NTOP − NONE), NTIMES = 5 and NJUM1 ~ (NONE/100) where t_p = normalized time when the periodicity is established. For coastal structures, t_p = 4 and NONE = 2000 have been used typically. The computation of the armor stability has been made during the last wave period after the establishment of the periodicity. The spatial variations of η and u have been stored at the normalized time t = t_p, (t_p + 1/4), (t_p + 1/2) , (t_p + 3/4) and t = (t_p + 1) where the spatial variations of η and u must be the same at t = t_p and (t_p + 1) after the establishment of the periodicity.

Next, the following parameters are read:

```
        READ(11,1110)  S
        READ(11,1180)  FWP
        READ(11,1180)  X1, X2
        READ(11,1180)  DELTA
        READ(11,1110)  NDELR
        DO  120  L=1,  NDELR
        READ(11,1180)  DELRP(L) (units:  mm or in)
120     CONTINUE
```

where DELTA, X1 and X2 are explained in the common /CONSTA/, whereas FWP is discussed in the common /BOT1/. The integer S included in the common /NODES/

is specified as input. The value of S specified as input corresponds to the number of spatial nodes from the seaward boundary to the wet node next to the initial waterline at SWL if IJOB = 1 or 2, whereas the input value of S is the number of spatial nodes from the seaward boundary to the landward boundary if IJOB = 3. On the other hand, NDELR and DELRP are explained in the common /RUNP1/ and /RUNP2/, respectively. If IJOB=3, NDELR=0 indicating no computation of wave runup. The physical values of DELRP(L) with $1 \leq L \leq$ NDELR need to be specified in millimeters if ISYST = 1 and in inches if ISYST = 2 as indicated above.

The incident wave characteristics are specified as follows:

```
READ(11,1180)  HREFP (units: m or ft), TP (units: sec)
READ(11,1180)  KSREF, KSSEA
```

where HREFP and TP are explained in the common /WAVE1/, whereas KSREF and KSSEA are discussed in the common /WAVE2/. The value of TP is read in seconds, while the value of HREFP is read in meters if ISYST=1 and in feet if ISYST=2.

The input parameters related to the structure geometry are read as follows:

```
READ(11,1180)  DSEAP (units: m or ft)
READ(11,1180)  TSLOPS
READ(11,1110)  NBSEG
```

where DSEAP, TSLOPS and NBSEG are explained in the common /BOT1/, /BOT2/ and /BOT4/, respectively. The dimensional quantities of the linear segments used to describe the structure geometry are explained in the common /BOT5/ as well as in Fig. 4. If IBOT=1, the width and slope of each segment need to be read as input.

```
        DO  130  K=1, NBSEG
        READ(11,1180) WBSEG(K) (units: m or ft), TBSLOP(K)
130     CONTINUE
```

On the other hand, if IBOT=2, the locations of the end points of the segments need to be read as input.

```
        DO  140  K=1, NBSEG+1
        READ(11,1180) XBSEG(K), ZBSEG(K) (units: m or ft)
140     CONTINUE
```

The dimensional quantities DSEAP, WBSEG(K), XBSEG(K) and ZBSEG(K) are read in meters if ISYST=1 and in feet if ISYST=2.

If ISAVB=1, the following quantities explained in the common /FILES/ need to be read as input:

```
        DO  150  I=1,   NNOD1
        READ(11,1190)   NODNO1(I), FNAME1(I)
150     CONTINUE
```

If ISTAB=1 or 2, the following parameters explained in the commons /TLEVEL/ and /STAB1/ need to be read as input:

```
        READ(11,1110)   NJUM2
        READ(11,1180)   C2, C3, SG
        READ(11,1180)   CD, CL, CM
        READ(11,1180)   TANPHI
        READ(11,1180)   AMAX, AMIN
```

where AMAX and AMIN are used only for ISTAB=1.

If ISTAB=2, the characteristic length of the armor unit needs to be specified as well

```
        READ(11,1180)   DAP (units:  m or ft)
```

where DAP is read in meters if ISYST=1 and in feet if ISYST=2.

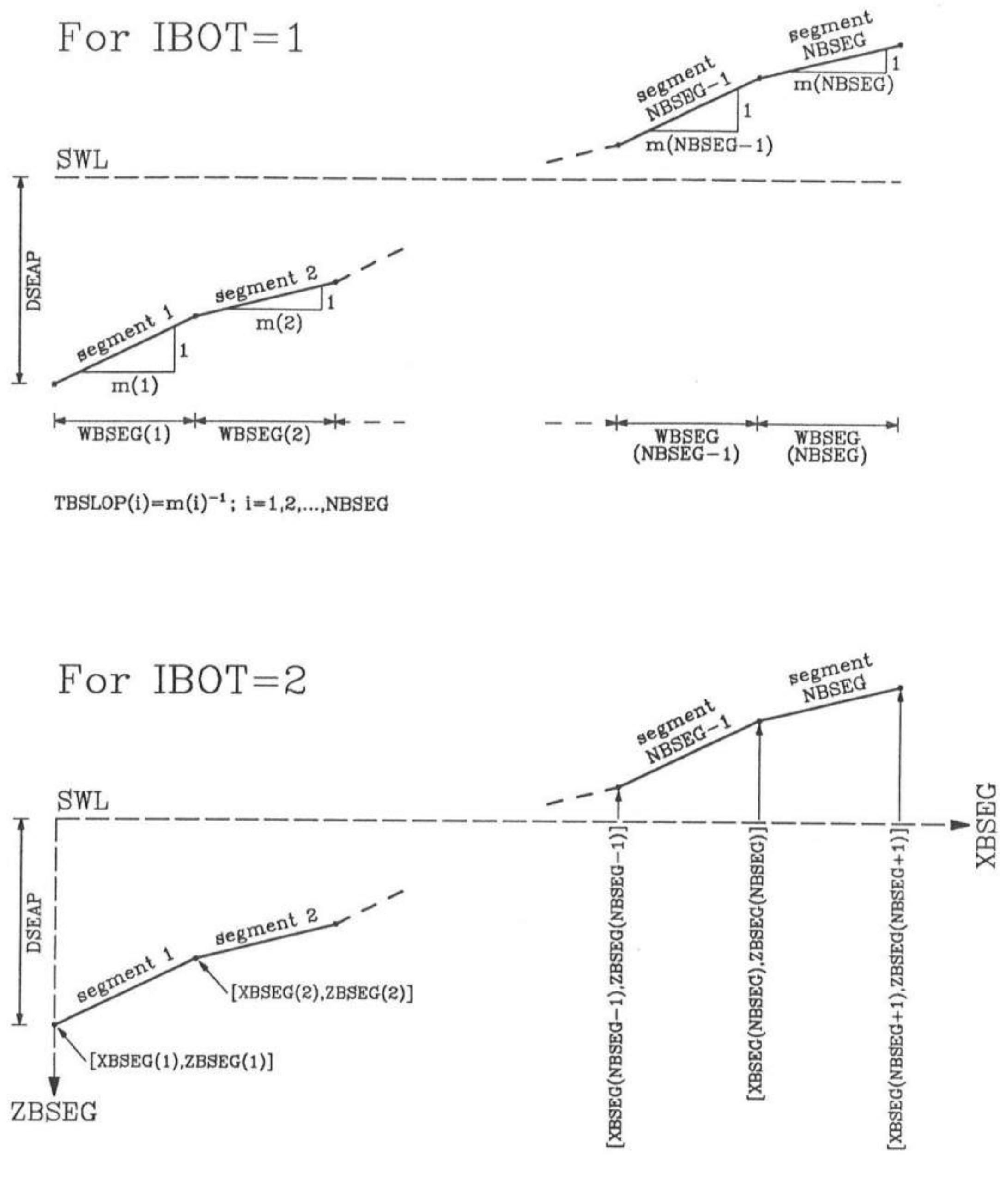


Figure 4. Input of Dimensional Structure Geometry for IBOT = 1 and 2

If ISAVC=1, the following quantities explained in the common /FILES/ needs to be read as input:

```
      DO  160  I=1,  NNOD2
      READ(11,1190)  NODNO2(I), FNAME2(I)
160   CONTINUE
```

This is the end of the data input in the subroutine INPUT1.

If IWAVE=2 or 3, the time series associated with the normalized wave profile at the seaward boundary as explained in the commons /ID/ and /WAVE3/ are read from the file with its unit number = 12 and its file name = FINP2 in the subroutine INPUT2

```
      READ(12,1210)  (IDUM, ETA(I), I=1, N2)
1210  FORMAT(I10, F10.6)
```

The number of the data points read is written on the file OMSG with its unit number = 29 as explained in the subroutine OPENER. The time series read as input are reduced such that ETA(I) with I=1,2,...,NTOP where NTOP/NICE is an integer and NICE=500 is specified in the parameter statement of the subroutine INPUT2.

Finally, the numerical stability indicator ALPHAS is computed in the subroutine DOC1 before the time marching computation. This indicator is defined as

$$\text{ALPHAS} = \frac{\Delta x}{\Delta t}\,(1 + d_t^{1/2})^{-1}\left[\left(1 + \frac{\epsilon^2}{4}\right)^{1/2} - \frac{\epsilon}{2}\right] \qquad (82)$$

The numerical stability criterion given by Eq. 27 requires that the value of ALPHAS should be greater than about unity where $|u_m| = 1$ and $c_m = d_t^{1/2}$ are assumed in Eq. 27. As a result, the conditional stop before the time-marching computation is included in the subroutine DOC1 as follows:

```
      WRITE(*,6010)  ALPHAS
      WRITE(*,6020)
      READ(*,*) ISTOP
      IF (ISTOP.EQ.1) STOP
6010  FORMAT ('Numerical stability indicator =', F7.2)
6020  FORMAT ('Time-marching computation is about to begin'/'1 = stop
             here, else = proceed')
```

If ALPHAS < 1, it is suggested to increase the value of NONE = Δt^{-1} specified as input since the numerical instability is likely to occur during the time-marching computation.

Error and Warning Statements

The computer program IBREAK includes various error and warning statements, some of which have been discussed in relation to the data input for convenience.

In the main program of IBREAK, if the normalized water depth h_j = U(2,J) < 0, the following statement is written and the computation stops:

```
      WRITE(*,2910)  U(2,J), J, S, N
      WRITE(29,2910) U(2,J), J, S, N
2910  FORMAT (/'From Main Program:  Negative water depth
             =',D12.3/'J=',I8,';S=',I8,';N=',I8)
```

where J = node number; S = waterline node number; and N = time level. Furthermore, if $|u_j| > (\Delta x/\Delta t)$, the following warning statement is written

```
      WRITE(*,2920)  V(J),XT,J,S,N
      WRITE(29,2920) V(J),XT,J,S,N
2920  FORMAT(/'From Main Program:  Abs(V(J)) > (X/T):','V(J)=',
             D12.3,';X/T=',D12.3/'J=',I8,';S=',I8,';N=',I8)
```

where V(J) = normalized fluid velocity u_j; and XT = $\Delta x/\Delta t$. It may be shown that the numerical stability criterion given by Eq. 27 is violated if $|u_j|$ >

($\Delta x/\Delta t$). In order to inform the progress of the time-marching computation, the following statement is written whenever the value of the time level N divided by 500 is an integer:

```
WRITE(*,*) 'N', N
```

Moreover, whenever IDUM = N/NONE is an integer, the following statement appears:

```
WRITE(*,*) 'Finished', IDUM, 'Wave Period(s)'
```

In the subroutine INPUT1, if none of IELEV, IV, IDUDT and ISNR are unity for the case of IREQ=1, the subroutine STOPP is called to write, 'Special storage requested, but pertinent identifiers not specified correctly. Check identifiers IREQ, IELEV, IV, IDUDT, ISNR.' Then, the computation stops. Furthermore, the computation stops if the subroutine CHEOPT finds an input error in the options specified by a user. If ISTAB=2 and ISAVC=1, the number of the elements of the vectors NODNO2(I) and FNAME2(I) must be the same as NNOD2. The subroutine STOPP is called to write 'Need more data' if the number of the elements of these vectors is less than NNOD2.

In the subroutine BOTTOM, the computation stops if the structure geometry specified as input is not consistent with the specified value of IJOB. If IJOB=1 or 2, the structure must be subaerial. Otherwise, the subroutine STOPP is called to write 'SWL is always above the structure. RUNUP/OVERTOPPING computation can not be performed.' If IJOB=3, the structure must be submerged. Otherwise, the subroutine STOPP is called to write 'Part of the structure is above SWL. TRANSMISSION computation can not be performed.'

Furthermore, if the number of nodes in the computation domain, JE, becomes greater than N1 = 500 specified in IBREAK, the following statement is written:

```
      WRITE(*,2910)  JE, N1
      WRITE(29,2910) JE, N1
2910  FORMAT (/'End Node=', I8,'; N1=', I8/'Slope/Structure is too
     +        long.'/'Cut it, or change PARAMETER N1.')
```

It is noted that N1=500 should be sufficient for most applications where the values of JE in the range from 100 to 300 have been used.

In the subroutine FINDM, the following statement appears if the parameter m satisfying Eq. 34 is not obtained:

```
      WRITE(*,2910)
      WRITE(29,2910)
2910  FORMAT('From Subr. 9 FINDM: '/'Criterion for parameter M not
     +        satisfied')
```

In the function CEL, the following statement appears if QQC = $(1-m)^{1/2}$ equals zero where the value of m is obtained in the subroutine FINDM:

```
      WRITE(*,*)  'Failure in Function CEL'
      WRITE(29,*) 'Failure in Function CEL'
```

which stops the computation.

In the subroutine MARCH, the computation stops if $h^*_{s-1} \le \delta$, corresponding to the third step in the numerical procedure dealing the moving waterline. The following statement is written:

```
      WRITE(*,2910)  U(2,M), DELTA, S, N
      WRITE(29,2910) U(2,M), DELTA, S, N
2910  FORMAT (/'From Subroutine 12 MARCH'/'U(2,S-1) is less than or equal
     +        to DELTA'/'U(2,S-1) =',D12.3/'DELTA =',D12.3/'S=',I8/'N=',
     +        I8/'Program Aborted')
```

where U(2,M) = h^*_{s-1}; DELTA = δ; S = waterline node number s; and N = time level. It is suggested to increase the value of NONE = Δt^{-1} to avoid the numerical instability which tends to occur near the moving waterline.

In the subroutine LANDBC, the following statement is written and the computation stops if IJOB=1 and S≥JE where S = waterline node number and JE = most landward node number:

```
      WRITE(*,2910)  N,S,JE
      WRITE(29,2910) N,S,JE
2910  FORMAT (/'From Subroutine 13 LANDBC:'/'N=', I8,';S=',I8, ';End
             Node=', I8/'Slope is not long enough to accommodate
             shoreline movement'/'Specify longer slope or choose
             overtopping computation')
```

This statement implies that wave overtopping over the specified structure geometry occurs even though IJOB=1 is specified. It is suggested to use IJOB=2 if the structure geometry is given or increase the crest height of the structure if no overtopping is allowed.

In the subroutine RUNUP, the following statement appears if $h^*_s \geq h^*_{s-1}$ and the adjustment described in the fourth step of the numerical procedure dealing with the moving waterline is made:

```
      WRITE(*,2910)  S,N,U(2,S), U(2,M)
      WRITE(29,2910) S,N,U(2,S), U(2,M)
2910  FORMAT(/'From Subroutine 14 RUNUP:  U(2,S)>U(2,S-1) at', 'S=', I8,
             ';N=', I8/' Adjusted values:','U(2,S)=', E12.3,';U(2,S-1)=',
             E12.3)
```

where U(2,S) = h^*_s; and U(2,M) = h^*_{s-1}. This statement does not stop the computation but suggests the numerical difficulty at the moving waterline which may eventually lead to the numerical instability.

In the subroutine STABNO, the following statement is written and the computation stops if the condition given by Eq. 68 is not satisfied:

```
      WRITE(*,2910)  N,J
      WRITE(29,2910) N,J
2910  FORMAT('From Subr. 19 STABNO'/'Armor Stability impossible'/'N=',
             I8,';J=',I8)
```

where N = time level; and J = node number. This statement implies that the values of AMIN = a_{min} and AMAX = a_{max} specified as input do not satisfy the conditions given by Eq. 70.

In the subroutine DOC1, the following statement is written if the values of ALPHAS given by Eq. 82 is less than unity:

```
      WRITE(*,2910)   ALPHAS
      WRITE(29,2910)  ALPHAS
2910  FORMAT(/'From Subr. 33 DOC1'/'Stability Indicator=', F9.3/'May
              cause numerical instability.  Increase NONE')
```

Furthermore, if NONE/NJUM1 and NONE/NJUM2 are not integers, the written statements instructing the changes of NJUM1 and NJUM2 appear in the manner similar to that for ALPHAS. The computation stops if the requirements for ALPHAS, NJUM1 and NJUM2 are not satisfied.

The subroutine CHEPAR checks whether the values of N1, N2, N3, N4 and N5 used to specify the sizes of matrices and vectors in the main program are equal to the corresponding values used in the subroutines. If this requirement is not satisfied, the computation stops and the instruction to correct the parameter error is written.

The subroutine CHEOPT checks the options selected by a user as well as the requirements of $1 \le$ NNOD1 $\le$ N5, $1 \le$ NNOD2 $\le$ N5, $1 \le$ NNREQ $\le$ N5, $1 \le S \le$ N1, $1 \le$ NDELR $\le$ N3 and $1 \le$ NBSEG $<$ N4. If there is an input error, the computation stops and the instruction to correct the input error is written.

The subroutine STOPP executes a programmed stop. This subroutine has already been explained when it is called in the subroutines INPUT1 and BOTTOM.

Output Parameters and Variables

The output of the input and computed results is made in the subroutines DOC1, DOC2 and DOC3 before, during and after the time-marching computation except that if IWAVE=1, the computed incident wave profile $\eta_i(t)$ over one wave period is written in the subroutine INWAV as follows:

```
      WRITE(34,3410) (ETA(I), I=1, NONE1)
3410  FORMAT (8F9.6)
```

where ETA(I) = $\eta_i(t)$ with t = (I−1)Δt and NONE1=(NONE+1). The unit numbers and file names used in IBREAK are explained in the subroutine OPENER.

The output parameters and variables from the subroutines DOC1, DOC2 and DOC3 are summarized in the following. The format statements in these subroutines are lengthy but self-explanatory. As a result, these format statements are omitted in the following.

The subroutine DOC1 documents the input data and calculated dimensionless parameters before the time-marching computation. First, the wave conditions at the seaward boundary specified as input are written depending on whether IWAVE=1, 2 or 3. If IWAVE=1, cnoidal or Stokes second-order wave theory is used to compute $\eta_i(t)$ in the subroutine INWAV depending on URPRE ≥ 26 or URPRE < 26, respectively, where URPRE is the value of the Ursell parameter based on linear wave theory. If cnoidal wave theory is used, the following output is written:

```
      WRITE(28,2813)  KC2, ECNO, KCNO
```

where KC2, ECNO and KCNO are explained in the common /WAVE5/ as well as in the corresponding format statement. The following quantities explained in the commons /WAVE1/, /WAVE2/, /WAVE4/, /BOT1/ and /BOT2/ are written

```
WRITE(28,2816)  ETAMAX, ETAMIN
WRITE(28,2817)  TP, HREFP, UL, DSEAP, UL, KSREF, KSSEA, KS
WRITE(28,2818)  DSEA, WL, SIGMA, UR, KSI
```

where UL indicates the unit (m or ft) of the quantity in front of UL. If IJOB=3, DLANDP included in the common /BOT1/ is written

```
WRITE(28,2819)  DLANDP, UL
```

The quantities explained in the commons /BOT1/, /BOT2/ and /BOT4/ are then written

```
WRITE(28,2821)  FWP, FW, WTOT, NBSEG
```

If IBOT=1, the width and slope of each segment of the structure as shown in Fig. 4 are written

```
WRITE(28,2824) (K,WBSEG(K),TBSLOP(K),K=1,NBSEG)
```

If IBOT=2, the coordinates of the end points of linear segments of the structure as shown in Fig. 4 are written

```
WRITE(28,2824) (K,XBSEG(K),ZBSEG(K),K=1,NBSEG+1)
```

The quantities explained in the commons /CONSTA/, /TLEVEL/, /NODES/ and /GRID/ are written

```
WRITE(28,2841) X,T,DELTA,X1,X2,ALPHAS
WRITE(28,2842) NTOP,NONE,JE
WRITE(28,2843) S
WRITE(28,2844) NJUM1
WRITE(28,2845) NJUM2
```

where ALPHAS is defined by Eq. 82 and the input value of S is written only if IJOB=1 or 2 since S =JE for IJOB=3. The value of NJUM2 is specified as input if ISTAB=1 or 2, but it is used only for ISTAB=1 in IBREAK. If ISTAB=1 or 2, the quantities explained in the common /STAB1/ are written

```
WRITE(28,2851)  TANPHI,SG,C2,C3,CD,CL,CM
WRITE(28,2851)  AMAX, AMIN
WRITE(28,2853)  DAP,UL
```

where AMAX and AMIN are used and written only for ISTAB=1, while DAP (m or ft) is required only for ISTAB=2. Moreover, the normalized structure geometry is written as follows:

```
WRITE(22,2210)  JE
WRITE(22,2220)  (XB(J),ZB(J),J=1,JE)
```

where JE is discussed in the common /NODES/, whereas XB(J) and ZB(J) are explained in the common /BOT3/.

The subroutine DOC2 stores some of the computed results at designated time levels during the time-marching computation where use is made hereafter of N = current time level, S = most landward wet node at this time level, and J = node number in the range $1 \leq J \leq S$. If ICALL=1, corresponding to the specified time levels for the case of ISAVA=1, the spatial variations of the normalized free surface elevation η and the normalized fluid velocity u are stored

```
WRITE(22,2210)  N,S
WRITE(22,2220)  (ELEV(J),V(J),J=1,S)
```

If ICALL=2 in the subroutine DOC2, the computed temporal variations of certain quantities at specified nodes are stored every NJUM1 time steps throughout the time-marching computation. If ISAVB=1, the temporal variation of the normalized water depth h is stored

```
WRITE(NUNIT,5010)  N, U(2,J)
```

where U(2,J) = value of h at the time level N and at the node J = NODNO1(I) with I=1,2,...,NNOD1, as explained in the common /FILES/, whereas the unit number NUNIT = (49 + I) as explained in the subroutine OPENER. If ISAVC=1, the temporal variation of the normalized displacement of the armor unit, X_a, defined by Eq. 81a from its initial location is stored

```
WRITE(NUNIT,7510)  N,XA(J)
```

where XA(J) = value of X_a at the time level N of the armor unit initially located at the node J = NODNO2(I) with I=1,2,...,NNOD2 as explained in the common /FILES/, while the unit number NUNIT = (74 + I) as explained in the subroutine OPENER. It is noted that XA(J)=0 if ISTATE(J)=0, corresponding to the stationary armor unit at the node J. Furthermore, the computed temporal variations at the landward and seaward boundaries at the time level N are stored every NJUM1 time steps throughout the time-marching computation. If IJOB=1 or 2, the following quantities at the time level N are written

```
WRITE(31,3110)  N,S
WRITE(31,3120)  (RUNUPS(L), L=1, NDELR)
```

where NDELR and RUNUPS(L) are explained in the commons /RUNP1/ and /RUNP2/. If IJOB=2, the hydrodynamic quantities at the most landward node JE are written

```
WRITE(32,3210)  N,U(1,JE),U(2,JE),V(JE),C(JE)
```

where the hydrodynamic quantities are explained in the common /HYDRO/. It is noted that if IJOB=2 and wave overtopping occurs, S=JE and the hydrodynamic quantities at the node JE are non-zero. If IJOB=3, the hydrodynamic quantities at the fixed landward boundary located at the node J=JE are written

```
WRITE(33,3310) N,U(1,JE),V(JE),C(JE),ETAT
```

where ETAT = value of the normalized free surface elevation η_t due to the transmitted wave. On the other hand, the quantities at the seaward boundary located at the node J=1 are written as follows:

```
WRITE(21,2110)  N,ETAI,ETAR,ETATOT,V(1),U(1,1)
```

where ETAI, ETAR and ETATOT are the values of η_i, η_r and $(\eta_i + \eta_r)$ at the time level N, respectively, while η_i and η_r are the normalized free surface elevations due to the incident and reflected waves, respectively.

If ICALL=3 in the subroutine DOC2, the spatial variations of the requested quantities are stored at the specified time levels N=NREQ(I) with I=1,2,...,NNREQ as explained in the common /IDREQ/. If IREQ=1 and the time level N=NREQ(I) in the main program, the following quantities are stored in the subroutine DOC2:

```
WRITE(40,4010)  N,S
WRITE(40,4020)  (ELEV(J),J=1,S)
WRITE(40,4020)  (V(J),J=1,S)
WRITE(40,4020)  (DUDT(J),J=1,S)
WRITE(40,4020)  (SNR(J),J=1,S)
```

where S indicates the most landward wet node at the time level N. It is noted that ELEV(J), V(J), DUDT(J) and SNR(J) are stored only if IELEV, IV,IDUDT and ISNR are unity, respectively, as explained in the common /IDREQ/.

The subroutine DOC3 documents the computed results after the time-marching computation. The reflection coefficients defined by Eqs. 37, 38 and 39 for I=1,2 and 3, respectively, are written

```
      WRITE(28,2811)  (RCOEF(I),I=1,3)
```

If IJOB=1 or 2, the normalized runup, run-down and setup for different values of δ_r' as explained in the common /RUNP2/ are written

```
      WRITE(28,2821)  JMAX
      WRITE(28,2822)  UL
      DO 110  L=1, NDELR
      WRITE(28,2823)  L,DELRP(L),RSTAT(2,L),RSTAT(3,L),RSTAT(1,L)
110   CONTINUE
```

where JMAX is the largest node number reached by the computational waterline based on $h=\delta$ and UL indicates the unit (mm or inches) of DELRP(L) = δ_r'. If IJOB=2, the wave overtopping quantities explained in the common /OVER/ are written

```
      WRITE(28,2831)  OV(1),U1STAT(1),OV(4),OV(2),OV(3)
```

where U1STAT(1) = value of the time-averaged flux $\bar{m}$ at the seaward boundary which should be equal to OV(1) = value of $\bar{m}$ at the landward edge if the steady state is really established. If ISTAB=1, the armor stability quantities explained in the commons /STAB5/ and /STAB6/ are written

```
      WRITE(28,2841)  SNSC,JSNSC,NSNSC
      WRITE(41,4110)  JMAX
      WRITE(41,4120)  (XB(J),ZB(J),SNSX(J),J=1,JMAX)
```

where XB(J) and ZB(J) express the normalized structure geometry as explained in the common /BOT3/. If ISTAB=2, the armor movement quantities explained in the commons /STAB7/ and /STAB8/ are written

```
          WRITE(28,2842)  NMOVE,NSTOP
          WRITE(42,4210)  NMOVE
          DO 120 J=1, JMAX
          IF(ISTATE(J).GE.1) WRITE(42,4220)
                NODIN(J),NODFI(J),NDIS(J),ISTATE(J),XB(J),ZB(J),XA(J)
    120   CONTINUE
```

where the armor unit initially located at the node J is stationary, moving or stopped after its movement depending on ISTATE(J)=0, 1 or 2, respectively. On the other hand, if IJOB=3, the time-averaged values of η_i, η_r and η_t are written as follows:

```
          WRITE(28,2851)  (ELSTAT(I),I=1,3), DELMWL
```

where ELSTAT(I) is explained in the common /STAT/ and DELMWL=($\overline{\eta_t} - \overline{\eta_r}$) is the mean water level difference at the landward and seaward boundaries. If IJOB=1 or 2, the transmitted wave is not present and the following output is made:

```
          WRITE(28,2852)  (ELSTAT(I),I=1,2)
```

Moreover, if IJOB=3, the wave transmission coefficients defined by Eqs. 48, 49 and 50 for I=1, 2 and 3, respectively, are written

```
          WRITE(28,2861)  (TCOEF(I), I=1,3)
          WRITE(28,2861)  U1STAT(1),U1STAT(JE),QAVER
```

where U1STAT(1) and U1STAT(JE) are the values of $\overline{m}$ at the seaward and landward boundaries, respectively, while QAVER is the average of these two values. The statistics of the hydrodynamic quantities explained in the common /STAT/ are written as follows:

```
      WRITE(23,2310) JMAX
      WRITE(23,2320) (U1STAT(J),J=1,JMAX)
      DO 130 I=1,3
      WRITE(23,2320) (ESTAT(I,J),J=1,JMAX)
      WRITE(23,2310) (VSTAT(I,J),J=1,JMAX)
  130 CONTINUE
```

Finally, if IENERG=1, the wave energy quantities explained in the common /ENERG/ are written

```
      WRITE(28,2872)  (ENERB(I),I=1,10)
      IF(IJOB.EQ.3) WRITE(28,2873) ENERB(11)
      WRITE(28,2874) (ENERB(I),I=12,13)
      IF(IJOB.EQ.3) WRITE(28,2875) ENERB(14)
      WRITE(35,3510) JMAX
      DO 140 I=1,4
      WRITE(35,3520)(ENER(I,J),J=1,JMAX)
  140 CONTINUE
```

Examples of Input and Output

In order to illustrate the input and output of the computer program IBREAK, four examples are given in the following. All these examples correspond to the case of IWAVE=1 where the normalized incident wave profile $\eta_i(t)$ is computed using Eq. 28 or 31. Use of IWAVE=2 was made for the comparisons of the numerical model with the experiments where measured incident wave trains were used as input by Kobayashi and Greenwald (1986,1988), Kobayashi and Watson (1987), and Kobayashi et al. (1989). If the free surface oscillation is measured at the seaward boundary of the computation domain in the presence of a coastal structure, use may be made of IWAVE=3, but this seaward boundary condition, which was used by Packwood (1980), has been found to produce spurious long-period oscillations in the computation domain.

The first example is the Test No. 12 of the large-scale riprap tests for uniform slopes conducted by Ahrens (1975). The structure was subaerial with no wave overtopping, corresponding to IJOB=1. Computation is made of the hydraulic stability of armor units by setting ISTAB=1. The tables and figures showing the input and output of the first example are given in Appendix B. Reference should be made to the papers of Kobayashi et al. (1987) and Kobayashi and Otta (1987) for the comparison of the numerical model with the large-scale riprap test data of Ahrens (1975) as well as the limitations and uncertainties of the numerical model.

The second example is the Test No. 32 of the large-scale riprap tests of Ahrens (1975) and is similar to the first example, where IJOB=1 for both examples. The major difference between the first and second examples is that the movement of individual armor units is computed in this example by setting ISTAB=2. The tables and figures showing the input and output of the second example are given in Appendix C.

The third example is one test run from the extensive small-scale test data of Saville (1955) for monochromatic wave overtopping over smooth impermeable structures fronted by a 1:10 slope. The structure was subaerial with wave overtopping, corresponding to IJOB=2. The structure was not protected with armor units. As a result, use is made if ISTAB=0, resulting in no computation of armor stability and movement. The tables and figures showing the input and output of the third example are given in Appendix D. Reference should be made to the report and paper of Kobayashi and Wurjanto (1988,1989a) for the comparison of the numerical model with the extensive small-scale test data of Saville (1955) as well as the difficulty associated with the selection of an appropriate location of the seaward boundary on the 1:10 slope.

The fourth example is one test run from the small-scale test data of Seelig (1980) for monochromatic wave transmission over a submerged smooth impermeable structure, corresponding to IJOB=3 and ISTAB=0. The tables and figures showing the input and output of the fourth example are given in Appendix E. Reference should be made to the report and paper of Kobayashi and Wurjanto (1989b,1989c) for the comparison of the numerical model with one set of the small-scale test data of Seelig (1980) as well as the discussion on the computed wave energy balance, the mean water level difference and the time-averaged volume flux per unit width.

PART IV: CONCLUSIONS

The computer program IBREAK presented herein simulates the interaction of normally incident waves with a rough or smooth impermeable coastal structure in the manner similar to hydraulic model tests in a wave flume. This numerical model is expected to be less accurate than hydraulic model tests performed carefully because of various assumptions and coefficients employed in the numerical model. The advantages of the numerical model are low cost, little start-up time, and high spatial and temporal resolution. During a preliminary design, the numerical model may be used together with empirical formulas, if available, to reduce the number of feasible alternatives. During a detailed design, the numerical model may be used to reduce the number of hydraulic model tests as well as to estimate the quantities which can not be measured directly. Reversely, the hydraulic model test results may be used to calibrate the empirical coefficients included in the numerical model (Kobayashi and Greenwald, 1986,1988). In short, a hybrid approach based on empirical formulas, numerical models and hydraulic model tests will improve the efficiency and reliability of the design of coastal structures.

The numerical model is presently being improved to include the effects of normally incident random waves and permeable underlayers. Additional improvements will include oblique and directional waves, the permeability of core materials, the hydraulic and geotechnical stability of the toe and foundation of a breakwater, the adjustment of the geometry of a mound structure, and the structural strength and breakage of large slender concrete armor units. Numerical models for coastal structures are very new and expected to progress rapidly.

This report is intended to make the computer program IBREAK easily accessible to coastal engineers and researchers. This report emphasizes the usage of IBREAK. Users of IBREAK are strongly recommended to read the related papers which presented more detailed mathematical formulations and physical interpretations. Correct interpretations of the output of IBREAK are essential to avoid misuse of IBREAK. Furthermore, calibration and verification of IBREAK will be required if it is to be applied to situations different from those examined before.

REFERENCES

Ahrens, J.P. (1975). "Large wave tank tests of riprap stability." Tech. Memo. No. 51, U. S. Army Coast. Engrg. Res. Ctr., Ft. Belvoir, Va.

Ahrens, J.P. and McCartney, B.L. (1975). "Wave period effect on the stability of riprap." Proc. Civil Engrg. in Oceans/III, ASCE, 1019-1034.

Ahrens, J.P. (1984). "Reef type breakwaters." Proc. 19th Coastal Engrg. Conf., ASCE, 2648-2662.

Ahrens, J.P. and Martin, F.T. (1985). "Wave runup formulas for smooth slopes." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 111(1), 128-133.

Allsop, N.W.H., Smallman, J.V. and Stephens, R.V. (1988). "Development and application of a mathematical model of wave action on steep slopes." Proc. 21st Coast. Engrg. Conf., ASCE, 281-291.

Baird, W.F. and Hall, K.R. (1984). "The design of breakwaters using quarried stones." Proc. 19th Coast. Engrg. Conf., ASCE, 1024-1031.

Barends, B.J., van der Kogel, H., Uijttewaal, J. and Hagenaar, J. (1983). "West breakwater - Sines, dynamic-geotechnical stability of breakwaters." Proc. Coastal Structures '83, ASCE, 31-44.

Battjes, J.A. (1974). "Surf similarity." Proc. 14th Coast. Engrg. Conf., ASCE, 466-480.

Battjes, J.A. (1984). "A review of methods to establish the wave climate for breakwater design." Coast. Engrg., 8, 141-160.

Bruun, P. and Johannesson, P. (1976). "Parameters affecting stability of rubble mounds." J. Wtrway. Harbors Coast. Engrg. Div., ASCE, 102(WW2), 141-164.

Bruun, P., ed. (1985). The Design and Construction of Mounds for Breakwaters and Coastal Protection. Elsevier, Amsterdam, The Netherlands.

Burcharth, H.F. (1979). "The effect of wave grouping on on-shore structures." Coast. Engrg., 2, 189-199.

Christensen, F.T., Broberg, P.C., Sand, S.E. and Tryde, P. (1984). "Behavior of rubble-mound breakwater in directional and uni-directional waves." Coast. Engrg., 8, 265-278.

Cross, R.H. and Sollitt, C.K. (1972). "Wave transmission by overtopping." J. Wtrway. Harbors Coast. Engrg. Div., ASCE, 98(WW3), 295-309.

Eckert, J. (1983). "Design of toe protection for coastal structures." Proc. Coastal Structures '83, ASCE, 331-341.

Goda, Y. (1985). Random Seas and Design of Maritime Structures, University of Tokyo Press, Tokyo, Japan.

Hannoura, A.A. and McCorquodale, J.A. (1985). "Rubble mounds: Numerical modeling of wave motion." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 111(5), 800-816.

Hibberd, S. and Peregrine, D.H. (1979). "Surf and run-up on a beach: A uniform bore." J. Fluid Mech., 95(2), 323-345.

Hudson, R.Y. (1959). "Laboratory investigation of rubble-mound breakwaters." J. Wtrways. and Harbors Div., ASCE, 85(WW3), 93-120.

Jensen, O.J. (1983). "Breakwater superstructures." Proc. Coastal Structures '83, ASCE, 272-285.

Jensen, O.J. (1984). A Monograph of Rubble-Mound Breakwaters, Danish Hydraulic Institute, Horsholm, Denmark.

Johnson, R.R., Mansard, E.P.D. and Ploeg, J. (1978). "Effects of wave grouping on breakwater stability." Proc. 16th Coast. Engrg. Conf., 2228-2243.

Jonsson, I.G. (1980). "A new approach to oscillatory rough turbulent boundary layers." Ocean Engrg., 7(1), 109-152.

Kobayashi, N. and Jacobs, B.K. (1985). "Stability of armor units on composite slope." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 111(5), 880-894.

Kobayashi, N., Roy, I. and Otta, A.K. (1986). "Numerical simulation of wave runup and armor stability." OTC Paper 5088, 18th Offshore Tech. Conf., 51-56.

Kobayashi, N. and Greenwald, J.H. (1986). "Prediction of wave runup and riprap stability." Proc. 20th Coast. Engrg. Conf., ASCE, 1958-1971.

Kobayashi, N. and DeSilva, G.S. (1987). "Motion of sediment particles in swash zone." Proc. Coastal Hydrodynamics, ASCE, 715-730.

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

Kobayashi, N., Otta, A.K. and Roy, I. (1987). "Wave reflection and runup on rough slopes." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 113(3), 282-298.

Kobayashi, N. and Watson, K.D. (1987). "Wave reflection and runup on smooth slopes." Proc. Coastal Hydrodynamics, ASCE, 548-563.

Kobayashi, N. and Greenwald, J.H. (1988). "Waterline oscillation and riprap movement." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 114(3), 281-296.

Kobayashi, N., Strzelecki, M.S. and Wurjanto, A. (1988). "Swash oscillation and resulting sediment movement." Proc. 21st Coast. Engrg. Conf., ASCE, 1167-1181.

Kobayashi, N. and Wurjanto, A. (1988). "Numerical prediction of wave overtopping on coastal structures." Res. Rept. No. CE-88-72, Ocean Engrg. Program, Univ. of Delaware, Newark, Delaware.

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. (1989a). "Wave overtopping on coastal structures." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 115(2), 235-251.

Kobayashi, N. and Wurjanto, A. (1989b). "Numerical prediction of wave transmission over submerged breakwaters." Res. Rept. No. CE-89-73, Center for Applied Coast. Res., Univ. of Delaware, Newark, Delaware.

Kobayashi, N. and Wurjanto, A. (1989c). "Wave transmission over submerged breakwaters." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 115(5) (in press).

Losada, M.A. and Gimenez-Curto, L.A. (1982). "Mound breakwaters under oblique wave attack: A working hypothesis." Coast. Engrg., 6, 83-92.

Madsen, O.S. and White, S.M. (1975). "Reflection and transmission characteristics of porous rubble-mound breakwaters." Tech. Rept. No. 207, R. M. Parsons Lab., Dept. of Civil Engrg., Mass. Inst. Tech., Cambridge, Mass.

Madsen, O.S. and White, S.M. (1976). "Energy dissipation on a rough slope." J. Wtrways. and Harbors Coast. Engrg. Div., ASCE, 102(WW1), 31-48.

Mansard, E.P.D. (1988). "Towards a better simulation of sea states for modelling of coastal structures." Proc. Berm Breakwaters, ASCE, 1-20.

Moretti, G. (1987). "Computation of flows with shocks." Ann. Rev. Fluid Mech., 19, 313-337.

Packwood, A.R. (1980). "Surf and run-up on beaches." Ph.D. Thesis, School of Math., Univ. of Bristol, U.K.

Peregrine, D.H. (1983). "Breaking waves on beaches." Ann. Rev. Fluid Mech., 15, 149-178.

Port Sines Investigating Panel (1982). Failure of the Breakwater at Port Sines, Portugal. American Society of Civil Engineers, New York, NY.

Press, W.H., et al. (1986). Numerical Recipes: The Art of Scientific Computing. Cambridge University Press, New York, NY.

Richtmyer, R.D. and Morton, K.W. (1967). Difference Methods for Initial-Value Problems. 2nd ed., Interscience, New York, NY.

Saville, T. (1955). "Laboratory data on wave runup and overtopping on shore structures." Tech. Memo. No. 64, U.S. Army Beach Erosion Board, Document Service Center, Dayton, Ohio.

Seelig, W.N. (1980). "Two-dimensional tests of wave transmission and reflection characteristics of laboratory breakwaters." Tech. Rept. 80-1, U.S. Army Coast. Engrg. Res. Ctr., Ft. Belvoir, Va.

Seelig, W.N. (1983). "Wave reflection from coastal structures." Proc. Coastal Structures '83, ASCE, 961-973.

Shore Protection Manual (1984). U.S. Army Engineers Waterways Experiment Station, Coastal Engineering Research Center, U.S. Government Printing Office, Washington, D.C.

Sleath, J.F.A. (1984). Sea Bed Mechanics, Wiley-Interscience, New York, NY.

Smith, O.P. (1986). "Cost-effective optimization of rubble-mound breakwater cross sections." Tech. Rept. CERC-86-2, U.S. Army Coast. Engrg. Res. Ctr., Vicksburg, Miss.

Sulisz, W. (1985). "Wave reflection and transmission of permeable breakwaters of arbitrary cross-section." Coast. Engrg., 9, 371-386.

Svendsen, I.A. and Brink-Kjaer, O. (1972). "Shoaling of cnoidal waves." Proc. 13th Coast. Engrg. Conf., ASCE, 365-383.

Thompson, A.C. (1988). "Numerical model of breakwater wave flows." Proc. 21st Coast. Engrg. Conf., ASCE, 2013-2027.

Van der Meer, J.W. (1987). "Stability of breakwater armor layers - design formulae." Coast. Engrg., 11, 219-239.

Van der Meer, J.W. (1988). "Deterministic and probabilistic design of breakwater armor layers." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 114(1), 66-80.

Weggel, J.R. (1976). "Wave overtopping equation." Proc. 15th Coast. Engrg. Conf., ASCE, 2737-2755.

Wurjanto, A. and Kobayashi, N. (1989). "Wave overtopping and transmission over breakwaters." Proc. PORTS '89, ASCE, 247-256.

APPENDIX A: LISTING OF IBREAK

The computer program IBREAK is listed in the following. The page number is shown on the upper right corner and the number of pages of IBREAK is 69. The page number for the rest of this appendix is omitted since the page A-I with I=2,3,...70 corresponds to the page (I-1) of IBREAK.

The listed computer program together with the tabulated input and output for the four examples presented in Appendices B, C, D and E is available on a floppy disk. The computer program IBREAK has been run using a mainframe computer, IBM 3090-180E as well as Sun workstations (Sun 3/60), which operates in conjunction with a central file server, Sun 4/280. Some modifications may be required if other computers are used. The computer time and memory requirements appear to be prohibitive for personal computers.

```
C
C       ##   #######   #######   ########   ######   ##    ##
C       ##   ##    ##  ##    ##  ##        ##    ##  ##   ##
C       ##   ##    ##  ##    ##  ##        ##    ##  ##  ##
C       ##   #######   #######   #######   ########  ## ##
C       ##   ##    ##  ## ##     ##        ##    ##  ######
C       ##   ##    ##  ##   ##   ##        ##    ##  ###  ##
C       ##   #######   ##    ##  ########  ##    ##  ##    ##
C
C           Numerical Simulation for Impermeable Breakwaters
C        . Wave Runup and Overtopping on Subaerial Structures .
C           . Wave Transmission over Submerged Structures .
C              . Stability and Movement of Armor Units .
C
C  Written by Andojo Wurjanto under Supervision of Nobuhisa Kobayashi
C      Ocean Engineering Program, Department of Civil Engineering
C            University of Delaware, Newark, Delaware 19716
C                             June, 1989
C
C ########################  GENERAL NOTES  ##########################
C
C  The purpose of each of 38 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 (Main Program
C  will be referred to as 'Main' hereafter). Description of each
C  COMMON statement is given only in Main.
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 IBREAK
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      DOUBLE PRECISION KCNO,MCNO,KC2
      DIMENSION VDUM(N1)
      CHARACTER*20 FINP1,FINP2,FNAME1,FNAME2
      INTEGER S
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.
C                     Subroutine 36 CHEPAR checks this requirement.
C     /CONSTA/  Basic constants and input to numerical model
```

```
C       /ID/       Identifiers specifying user's options                    IBR00560
C       /IDREQ/    Integers for special storing for armor stability         IBR00570
C                  (see Subroutine 2 INPUT1 for special storing)            IBR00580
C       /TLEVEL/   Integers related to time levels                          IBR00590
C       /NODES/    Integers for spatial nodes                               IBR00600
C       /GRID/     Time step, grid size, and related quantities             IBR00610
C       /WAVE1/    Dimensional wave data                                    IBR00620
C       /WAVE2/    Dimensionless wave parameters                            IBR00630
C       /WAVE3/    Normalized surface elevations                            IBR00640
C       /WAVE4/    Max. and min. of normalized incident wave profile        IBR00650
C       /WAVE5/    Cnoidal wave parameters (K, E, m and 1-m)                IBR00660
C       /BOT1/     Dimensional parameters related to structure              IBR00670
C       /BOT2/     Normalized parameters related to structure               IBR00680
C       /BOT3/     Normalized structure geometry                            IBR00690
C       /BOT4/ and /BOT5/ Dimensional structure geometry                    IBR00700
C       /HYDRO/    Hydrodynamic quantities computed                         IBR00710
C       /MATRIX/   Elements of matrices used in numerical method            IBR00720
C       /RUNP1/ and /RUNP2/ Quantities related to wave runup                IBR00730
C       /OVER/     Quantities related to wave overtopping                   IBR00740
C       /COEFS/    Reflection and transmission coefficients                 IBR00750
C       /STAT/     Mean, max. and min. of hydrodynamic quantities           IBR00760
C       /ENERG/    Quantities related to wave energy                        IBR00770
C       /STAB1/    Armor stability parameters read as input                 IBR00780
C       /STAB2/    Computed armor stability parameters                      IBR00790
C       /STAB3/    Armor stability parameters used in Subr. 19 STABNO       IBR00800
C       /STAB4/    Armor movement parameters used in Subr. 20 MOVE          IBR00810
C       /STAB5/ and /STAB6/ Stability numbers and associated quantities     IBR00820
C                  in Subr. 19 STABNO                                       IBR00830
C       /STAB7/ and /STAB8/ Quantities associated with armor movement       IBR00840
C                  in Subr. 20 MOVE                                         IBR00850
C       /FILES/    File names and associated node numbers related to        IBR00860
C                  options ISAVB=1 and ISAVC=1                              IBR00870
C       /VALUEN/   Values at time level (N-1) stored during computation     IBR00880
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R                                   IBR00890
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2                                   IBR00900
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,            IBR00910
     +                ISAVA,ISAVB,ISAVC                                     IBR00920
      COMMON /IDREQ/  IREQ,IELEV,IV,IDUDT,ISNR,NNREQ,NREQ(N5)               IBR00930
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES        IBR00940
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX                                   IBR00950
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX                               IBR00960
      COMMON /WAVE1/  HREFP,TP,WL0P                                         IBR00970
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA              IBR00980
      COMMON /WAVE3/  ETA(N2),ETAIS(N2),ETARS(N2),ETATS(N2)                 IBR00990
      COMMON /WAVE4/  ETAMAX,ETAMIN                                         IBR01000
      COMMON /WAVE5/  KCNO,ECNO,MCNO,KC2                                    IBR01010
      COMMON /BOT1/   DSEAP,DLANDP,FWP                                      IBR01020
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT         IBR01030
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)         IBR01040
      COMMON /BOT4/   NBSEG                                                 IBR01050
      COMMON /BOT5/   WBSEG(N4),TBSLOP(N4),XBSEG(N4),ZBSEG(N4)              IBR01060
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)                 IBR01070
      COMMON /MATRIX/ A1(2,N1),F(2,N1),G1(N1),GJR(2,N1),S1(N1),D(2,N1)      IBR01080
      COMMON /RUNP1/  NDELR                                                 IBR01090
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3)           IBR01100
```

```
      COMMON /OVER/   OV(4)
      COMMON /COEFS/  RCOEF(3),TCOEF(3)
      COMMON /STAT/   ELSTAT(3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1)
      COMMON /ENERG/  ENER(4,N1),ENERB(14)
      COMMON /STAB1/  C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /STAB2/  SG1,CTAN(N1)
      COMMON /STAB3/  CSTAB1,CSTAB2,AMAXS,AMINS,E2,E3PRE(N1)
      COMMON /STAB4/  CSTAB3,CSTAB4,CM1,DA,SIGDA,WEIG
      COMMON /STAB5/  JSNSC,NSNSC,NSNSX(N1)
      COMMON /STAB6/  SNSC,SNR(N1),SNSX(N1)
      COMMON /STAB7/  NMOVE,NSTOP,
     +                ISTATE(N1),NODIN(N1),NODFI(N1),NDIS(N1)
      COMMON /STAB8/  VA(N1),XAA(N1),XA(N1)
      COMMON /FILES/  NNOD1,NNOD2,NODNO1(N5),NODNO2(N5),
     +                FNAME1(N5),FNAME2(N5)
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
C
      SAVE K,M,N
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 wave profile at seaward boundary if IWAVE>1
C
      WRITE (*,*) 'Name of Primary Input-Data-File?'
      READ  (*,5000) FINP1
      CALL OPENER (1,FINP1,FINP2)
      CALL INPUT1 (FINP2)
      CALL OPENER (2,FINP1,FINP2)
      IF (IWAVE.GT.1) CALL INPUT2
C
C ... PRE-PROCESSING
C
C     Subr. 4 BOTTOM computes normalized structure geometry
C     Subr. 5 PARAM  calculates important parameters
C     Subr. 6 INIT1  specifies initial conditions
C     Subr. 8 INWAV  computes incident wave profile if IWAVE=1
C
      CALL BOTTOM
      CALL PARAM
      CALL INIT1
```

```
      IF (IWAVE.EQ.1) CALL INWAV
C
C ... PRE-LOOP DOCUMENTATION
C
C     Subr. 33 DOC1 documents input data and related parameters
C       before time-marching computation
C     Subr. 34 DOC2 is checked using ICALL=0 before computation
C
      CALL DOC1
      CALL DOC2 (0,0,DUM,DUM)
      IF (IJOB.EQ.3) M=S-1
C
C ----------------------- DO LOOP 500 BEGINS -----------------------
C
C     For known hydrodynamic quantities at time level (N-1) compute
C     values of U(i,j) with i=1,2 and V(j) at node j for next time
C     level N where normalized time t=N*(time step size delta t)
C     with N=1,2,...,NTOP
C
      DO 500 N = 1,NTOP
C
C ..... ESTIMATE U(2,K) AND V(K) WITH K=(S+1) BY EXTRAPOLATION
C
C       S = most landward node at time level (N-1)
C       The following values at node j are known at time level (N-1)
C         U(1,j) = volume flux
C         U(2,j) = total water depth
C         V(j)   = depth-averaged velocity
C
        IF (IJOB.LT.3) THEN
          M = S-1
          IF (S.LT.JE) THEN
            K = S+1
            V(K)   = 2.D+00*V(S) - V(M)
            U(2,K) = 2.D+00*U(2,S) - U(2,M)
            U(1,K) = U(2,K)*V(K)
            IF (U(2,K).GT.0.D+00) THEN
              C(K) = DSQRT(U(2,K))
            ELSE
              C(K) = 0.D+00
            ENDIF
          ENDIF
        ENDIF
C
C ..... RETAIN SOME VALUES AT TIME LEVEL (N-1) AT LANDWARD AND
C       SEAWARD BOUNDARIES
C
        VSN    = V(S)
        USN(1) = U(1,S)
        USN(2) = U(2,S)
        VMN    = V(M)
        UMN(1) = U(1,M)
        V1N    = V(1)
        V2N    = V(2)
C
```

```
C ..... CRITICAL VELOCITIES USED IN CHARACTERISTIC VARIABLES
C
        DO 110 J = 1,S
          IF (U(2,J).LT.0.D+00) THEN
            WRITE  (*,2910) U(2,J),J,S,N
            WRITE (29,2910) U(2,J),J,S,N
            STOP
          ELSE
            C(J) = DSQRT(U(2,J))
          ENDIF
  110   CONTINUE
C
C ..... MARCH FROM TIME LEVEL (N-1) TO TIME LEVEL N
C
C       Subr. 12 MARCH marches computation from time level (N-1) to N
C         excluding landward and seaward boundaries
C       Landward B.C. is in Subr. 13 LANDBC
C       Seaward  B.C. is in Subr. 16 SEABC
C
        CALL MARCH  (N,M)
        CALL LANDBC (N,K,M,ETAT)
        CALL SEABC  (N,ETAR)
C
C ..... CHECK IF STABILITY CRITERION IS NOT VIOLATED
C
C       T = time step; X = spatial grid size; XT = X/T
C
        DO 120 J = 1,S
          IF (DABS(V(J)).GT.XT) THEN
            WRITE  (*,2920) V(J),XT,J,S,N
            WRITE (29,2920) V(J),XT,J,S,N
          ENDIF
  120   CONTINUE
C
C ..... WAVE ENERGY FLUX AND DISSIPATION
C       Computed in Subr. 17 ENERGY
C
        IF (IENERG.EQ.1.AND.N.GE.NSTAT) CALL ENERGY (N)
C
C ..... STATISTICS OF HYDRODYNAMIC QUANTITIES
C
C       Subr. 31 ASSIGN changes notions from matrix to vector or from
C         vector to matrix
C       Subr. 18 STAT1 finds mean, max. and min. values
C       NSTAT = time level when statistical calculations begin
C       At node j:
C         U1STAT(j) = mean volume flux
C         ELEV(j)   = surface elevation above SWL
C         V(j)      = depth-averaged velocity
C       Mean, maximum, and minimum at node j:
C         ESTAT(1,j),ESTAT(2,j),ESTAT(3,j): for ELEV(j)
C         VSTAT(1,j),VSTAT(2,j),VSTAT(3,j): for V(j)
C       JMAX = the largest node number reached by the computational
C         waterline during N=NSTAT to N=NTOP
C       Note:
```

```
C           For IJOB=3, JMAX=S was specified in Subr. 6 INIT1 as input
C
          IF (N.GE.NSTAT) THEN
            CALL ASSIGN (1,VDUM  ,U   ,2,S,1)
            CALL STAT1  (1,U1STAT,VDUM,1,S)
            CALL STAT1  (2,ESTAT ,ELEV,3,JE)
            CALL STAT1  (2,VSTAT ,V   ,3,S)
            IF (IJOB.LT.3.AND.S.GT.JMAX) JMAX=S
          ENDIF
C
C ..... COMPUTATION OF ARMOR STABILITY OR MOVEMENT
C       From N=NSTAB to N=NTOP
C
C       NSTAB = time level when computation of armor stability or
C               movement begins
C       ISTAB=1: INITIATION OF MOVEMENT OF ARMOR UNITS IN
C                SUBR. 19 STABNO
C                Computing SNR at every node at every NJUM2 time steps
C         SNR(j) = stability number against rolling/sliding at node j
C       ISTAB=2: SLIDING MOTION OF ARMOR UNITS IN SUBR. 20 MOVE
C                Tracking individual armor units
C         NMOVE = no. of units dislodged from their initial locations
C         NSTOP = no. of units stopped after moving
C         XAA(j),XA(j) = displacement of moving or stopped armor unit
C                 number j from its initial location, normalized by
C                 TP*sqrt(GRAV*HREFP) and DAP, respectively
C
        IF (ISTAB.GT.0.AND.N.GE.NSTAB) THEN
          IF (ISTAB.EQ.1) THEN
            IDUM = MOD((N-NSTAB),NJUM2)
            IF (IDUM.EQ.0) CALL STABNO (N)
          ELSE
            CALL MOVE (N)
          ENDIF
        ENDIF
C
C ..... DOCUMENTATION DURING TIME-MARCHING COMPUTATION
C       Subr. 34 DOC2 documents computed results at designated time
C       levels
C
C       Calling DOC2(1,...) is for storing "A"
C         "A" = spatial variations of hydrodynamic quantities
C         Storing "A" is performed NTIMES (>1) times at equal
C           intervals from N=NSAVA to N=NTOP
C         NTOP  = final time level
C         NSAVA = time level when storing "A" begins
C       Calling DOC2(2,...) is for storing temporal variations at
C         specified nodes every NJUM1 time steps during N=1 to N=NTOP
C       Calling DOC2(3,...) is for storing spatial variations at
C         specified time levels N=NREQ(i) with i=1,2,...,NNREQ
C
        IF (N.GE.NSAVA) THEN
          IDUM1 = (N-NSAVA)*(NTIMES-1)
          IDUM2 = NTOP-NSAVA
          IDUM3 = MOD(IDUM1,IDUM2)
```

```
          IF (IDUM3.EQ.0) CALL DOC2 (1,N,DUM,DUM)
        ENDIF
        IDUM4 = MOD(N,NJUM1)
        IF (IDUM4.EQ.0) CALL DOC2 (2,N,ETAR,ETAT)
        IF (IREQ.EQ.1) THEN
          DO 130 I = 1,NNREQ
            IF (N.EQ.NREQ(I)) CALL DOC2 (3,N,DUM,DUM)
  130     CONTINUE
        ENDIF
C
C ..... HOW FAR THE COMPUTATION HAS BEEN
C
        IDUM = MOD(N,500)
        IF (IDUM.EQ.0) WRITE (*,*) 'N',N
        IDUM = MOD(N,NONE)
        IF (IDUM.EQ.0) THEN
          IDUM = N/NONE
          WRITE (*,*) ' Finished ',IDUM,' Wave Period(s)'
        ENDIF
C
  500 CONTINUE
C
C ------------------------ DO LOOP 500 ENDS ------------------------
C
C ... POST-PROCESSING
C
C     Subr. 23 STAT2 calculates statistical values
C     Subr. 25 BALANE checks overall energy balance
C
      CALL STAT2
      IF (IENERG.EQ.1) CALL BALANE
C
C ... POST-LOOP DOCUMENTATION
C     Subr. 35 DOC3 documents results after time-marching
C     computation
C
      CALL DOC3
C
C ... FORMATS
C
 2910 FORMAT (/' From Main Program: Negative water depth =',D12.3/
     +          ' J =',I8,'; S =',I8,'; N = ',I8)
 2920 FORMAT (/' From Main Program: Abs(V(J))>(X/T):',
     +          ' V(J) =',D12.3,'; X/T =',D12.3/
     +          ' J =',I8,'; S =',I8,'; N = ',I8)
 5000 FORMAT (A20)
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,FINP1,FINP2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      CHARACTER*20 FINP1,FINP2,FNAME1,FNAME2
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /IDREQ/  IREQ,IELEV,IV,IDUDT,ISNR,NNREQ,NREQ(N5)
      COMMON /FILES/  NNOD1,NNOD2,NODNO1(N5),NODNO2(N5),
     +                FNAME1(N5),FNAME2(N5)
C
      IF (ICALL.EQ.1) THEN
C
C        Subr. 36 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   21   OSEAWAV     Stores quantities at seaward boundary at every
C                    NJUM1 time steps
C                    --> N,ETAI,ETAR,ETATOT,V(1),U(1,1)
C   22   OSPACE    . Unconditionally, stores structure geometry
C                    --> JE,(XB(J),ZB(J),J=1,JE)
C                  . Conditionally, i.e., if ISAVA=1, stores spatial
C                    variations of flow quantities at designated time
C                    levels from N=NSAVA to N=NTOP
C                    --> N,S,(ELEV(J),V(J),J=1,S)
C   23   OSTAT       Stores statistics of hydrodynamic quantities
C                    -->  (U1STAT(J), J=1,JMAX)
C                    -->  (ESTAT(i,J),J=1,JMAX)
C                    -->  (VSTAT(i,J),J=1,JMAX)
C                          i=1,2,3
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=21,FILE='OSEAWAV',STATUS='NEW',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')
         RETURN
C
      ELSE
C
```

```
C ..... CONDITIONAL OPENINGS FOR ICALL=2
C
C       Units 31-39 reserved for files containing hydrodynamic and
C                   energy quantities
C       Units 41-49 reserved for files containing armor stability and
C                   movement quantities
C       Units 50-74 reserved for saving "B"
C       Units 75-99 reserved for saving "C"
C       "B" = temporal variations of normalized total water depth
C             at specified nodes
C       "C" = temporal variations of normalized displacement of
C             armor units from specified initial nodal locations
C
C  Unit  Filename    Purpose
C  ----  --------    ---------------------------------------------------
C   12   FINP2       Contains input data precsribing water surface
C                    elevations at seaward boundary if IWAVE>1
C   31   ORUNUP      Stores waterline node and runup elevations
C                    associated with (DELTAR(L),L=1,NDELR), if IJOB<3,
C                    at every NJUM1 time steps
C                    --> N,S,(RUNUPS(L),L=1,NDELR)
C   32   OOVER       Stores quantities at landward edge node, if
C                    IJOB=2, at every NJUM1 time steps
C                    --> N,U(1,JE),U(2,JE),V(JE),C(JE)
C   33   OTRANS      Stores values at landward boundary, if IJOB=3,
C                    at every NJUM1 time steps
C                    --> N,U(1,JE),V(JE),C(JE),ETAT
C   34   OINWAV      Stores incident wave profile at seaward boundary
C                    if IWAVE=1
C   35   OENERG      Stores time-averaged energy quantities if IENERG=1
C                    --> JMAX,(ENER(i,J),J=1,JMAX)
C                        i=1,2,3,4
C   40   OREQ        Stores spatial variation at designated time levels
C                    (e.g., at time of minimum stability) if IREQ=1
C   41   OSTAB1      Stores local stability number at each node
C                    --> JMAX,(XB(J),ZB(J),SNSX(J),J=1,JMAX)
C   42   OSTAB2      Stores quantities related to armor movement
C
C   50 FNAME1(1)  !
C   51 FNAME1(2)  !  Store "B", i.e., temporal variations of normalized
C   .             !  total water depth at specified nodes at every
C   .             !  NJUM1 time steps if ISAVB=1
C   and so on     !
C
C   75 FNAME2(1)  !
C   76 FNAME2(2)  !  Store "C", i.e., temporal variations of normalized
C   .             !  displacement, XA, of armor units from specified
C   .             !  initial nodal locations at every NJUM1 time steps
C   and so on     !  if ISAVC=1
C
C ----- WAVES AT SEAWARD BOUNDARY
C
      IF (IWAVE.EQ.1) THEN
        OPEN (UNIT=34,FILE='OINWAV',STATUS='NEW',ACCESS='SEQUENTIAL')
      ELSE
```

```
          OPEN (UNIT=12,FILE=FINP2,STATUS='OLD',ACCESS='SEQUENTIAL')
        ENDIF
C
C ----- RUNUP-OVERTOPPING-TRANSMISSION
C
        IF (IJOB.LT.3) THEN
          OPEN (UNIT=31,FILE='ORUNUP',STATUS='NEW',ACCESS='SEQUENTIAL')
          IF (IJOB.EQ.2) OPEN
     +        (UNIT=32,FILE='OOVER',STATUS='NEW', ACCESS='SEQUENTIAL')
        ELSE
          OPEN (UNIT=33,FILE='OTRANS',STATUS='NEW',ACCESS='SEQUENTIAL')
        ENDIF
C
C ----- WAVE ENERGY
C
        IF (IENERG.EQ.1)
     +    OPEN (UNIT=35,FILE='OENERG',STATUS='NEW',ACCESS='SEQUENTIAL')
C
C ----- FOR SAVING "B"
C
        IF (ISAVB.EQ.1) THEN
          DO 110 I = 1,NNOD1
            NUNIT = 49 + I
            OPEN (UNIT=NUNIT,FILE=FNAME1(I),STATUS='NEW',
     +               ACCESS='SEQUENTIAL')
  110     CONTINUE
        ENDIF
C
C ----- ARMOR STABILITY
C
        IF (ISTAB.EQ.1) THEN
          OPEN (UNIT=41,FILE='OSTAB1',STATUS='NEW',ACCESS='SEQUENTIAL')
        ELSEIF (ISTAB.EQ.2) THEN
          OPEN (UNIT=42,FILE='OSTAB2',STATUS='NEW',ACCESS='SEQUENTIAL')
C         --------------- The following is for saving "C"
          IF (ISAVC.EQ.1) THEN
            DO 120 I = 1,NNOD2
              NUNIT = 74 + I
              OPEN (UNIT=NUNIT,FILE=FNAME2(I),STATUS='NEW',
     +                 ACCESS='SEQUENTIAL')
  120       CONTINUE
          ENDIF
        ENDIF
C
C ----- SPECIAL STORING AT SPECIFIED TIME LEVELS
C
        IF (IREQ.EQ.1)
     +    OPEN (UNIT=40,FILE='OREQ',STATUS='NEW',ACCESS='SEQUENTIAL')
C
        RETURN
C
      ENDIF
C
      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=500,N2=30000,N3=3,N4=100,N5=25)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      CHARACTER*5 COMMEN(14)
      CHARACTER*20 FINP2,FNAME1,FNAME2
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /IDREQ/  IREQ,IELEV,IV,IDUDT,ISNR,NNREQ,NREQ(N5)
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /WAVE1/  HREFP,TP,WL0P
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /BOT1/   DSEAP,DLANDP,FWP
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /BOT4/   NBSEG
      COMMON /BOT5/   WBSEG(N4),TBSLOP(N4),XBSEG(N4),ZBSEG(N4)
      COMMON /RUNP1/  NDELR
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3)
      COMMON /STAB1/  C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /FILES/  NNOD1,NNOD2,NODNO1(N5),NODNO2(N5),
     +                FNAME1(N5),FNAME2(N5)
      DATA INDIC /0/
      CALL CHEPAR (2,1,N1,N1R)
      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           IJOB=1: RUNUP on impermeable slope
C               =2: OVERTOPPING over subaerial structure
C               =3: TRANSMISSION over submerged structure
C           ISTAB=0: No computation of armor stability or movement
C                =1: Armor stability computation
C                =2: Armor movement computation
C           If ISTAB>0 --> Must specify NSTAB
```

```
C                Armor stability or movement is computed from N=NSTAB
C                to N=NTOP
C             ISYST=1: International System of Units (SI) is used
C                  =2: US Customary System of Units (USCS) is used
C             IBOT=1: "Type 1" bottom data (width-slope)
C                 =2: "Type 2" bottom data (coordinates)
C             INONCT=0: No correction term in computing ETAR
C                   =1: Correction term for ETAR recommended for
C                       beaches
C             IENERG=0: Energy quantities NOT computed
C                   =1: Energy quantities computed
C             IWAVE=1: Incident 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 given wave
C             "A" = Spatial variations of hydrodynamic quantities
C             "B" = Temporal variations of total water depth at
C                   specified nodes at every NJUM1 time steps
C             "C" = Temporal variations of displacement of armor units
C                   from specified initial nodal locations at every
C                   NJUM1 time steps
C             ISAVA,ISAVB,ISAVC are identifiers associated with saving
C                             "A","B","C", respectively (1=save; 0=no)
C             NSAVA AND NTIMES:
C             If ISAVA=1, "A" is saved NTIMES (>1) times at equal
C                            intervals from N=NSAVA to N=NTOP
C             If ISAVB=1 --> Must specify NNOD1, i.e, the number of
C                            nodes for which "B" is to be saved
C             If ISAVC=1 --> Must specify NNOD2,i.e., the number of
C                            nodes for which "C" is to be saved
C             IREQ=0: No special storing
C                 =1: Special storing requested
C             Special storing = storing spatial variations of requested
C                 quantities at time levels N=NREQ(i)
C                 with i=1,2,...,NNREQ
C             Quantities available for request:
C             . ELEV = surface elevation
C             . V    = depth-averaged velocity
C             . DUDT = total fluid acceleration
C             . SNR  = stability number against rolling/sliding
C             --> requested by IELEV=1, IV=1, IDUDT=1, and ISNR=1,
C                 respectively
C             Note: DUDT can be requested only if ISTAB>0
C                   SNR  can be requested only if ISTAB=1
      READ (11,1130) IJOB,ISTAB,NSTAB
      READ (11,1140) ISYST
      READ (11,1140) IBOT
      READ (11,1140) INONCT
      READ (11,1140) IENERG
      READ (11,1150) IWAVE,FINP2
      READ (11,1160) ISAVA,ISAVB,ISAVC,NSAVA,NTIMES,NNOD1,NNOD2
      READ (11,1170) IREQ,IELEV,IV,IDUDT,ISNR,NNREQ
C
C ......... CHECK OPTIONS
```

```
C           Subr. 37 CHEOPT is to check if user's options are within
C           the ranges available
      CALL CHEOPT ( 1,INDIC,IJOB   ,1,3)
      CALL CHEOPT ( 2,INDIC,ISTAB  ,0,2)
      CALL CHEOPT ( 3,INDIC,ISYST  ,1,2)
      CALL CHEOPT ( 4,INDIC,IBOT   ,1,2)
      CALL CHEOPT ( 5,INDIC,INONCT,0,1)
      CALL CHEOPT ( 6,INDIC,IENERG,0,1)
      CALL CHEOPT ( 7,INDIC,IWAVE  ,1,3)
      CALL CHEOPT ( 8,INDIC,ISAVA  ,0,1)
      CALL CHEOPT ( 9,INDIC,ISAVB  ,0,1)
      CALL CHEOPT (10,INDIC,ISAVC  ,0,1)
      CALL CHEOPT (11,INDIC,IREQ   ,0,1)
      CALL CHEOPT (12,INDIC,IELEV  ,0,1)
      CALL CHEOPT (13,INDIC,IV     ,0,1)
      CALL CHEOPT (14,INDIC,IDUDT  ,0,1)
      CALL CHEOPT (15,INDIC,ISNR   ,0,1)
      IF (ISAVB.EQ.1) CALL CHEOPT (16,IDUM,NNOD1,1,N5)
      IF (ISAVC.EQ.1) CALL CHEOPT (17,IDUM,NNOD2,1,N5)
C
C ......... PRE-PROCESS SPECIAL STORING
C           Subr. 38 STOPP stops execution of the computation
      IF (IREQ.EQ.1) THEN
        IF (ISTAB.EQ.0.AND.IDUDT.NE.0) IDUDT=0
        IF (ISTAB.NE.1.AND.ISNR.NE.0) ISNR=0
        NOREQ = IELEV+IV+IDUDT+ISNR
        IF (NOREQ.EQ.0) THEN
          CALL STOPP (1,3)
        ELSE
          CALL CHEOPT (18,IDUM,NNREQ,1,N5)
          READ (11,1110) (NREQ(I),I=1,NNREQ)
        ENDIF
      ENDIF
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 TIME STEPPING
C           NTOP = total number of time steps for computation
C           NONE = even number of time steps in one wave period
C           The wave period is the reference period used for the
C                  normalization of the governing equations
C           NJUM1: Temporal variations at specified nodes are stored
C                  at every NJUM1 time steps
      READ (11,1110) NTOP
      READ (11,1110) NONE
```

```
      READ (11,1110) NJUM1
C
C ......... IMPORTANT TIME LEVELS INCLUDED IN COMMON /TLEVEL/
C           NTOP   = final time level
C           NONE   = even number of time steps in one wave period
C           NSAVA  = time level when saving "A" begins
C           NTIMES = number of time levels when "A" is saved
C           NSTAB  = time level when computation of armor stability
C                    or movement begins
C           NSTAT  = time level when statistical calculations begin
C           Used: NSTAT=(NTOP-NONE+1) for IWAVE=1
C                 NSTAT=NSAVA for IWAVE>1
C           Note: The value of NTOP for IWAVE>1 will be adjusted in
C                 Subr. 3 INPUT2
C           -------------------------------------------------------
C           For monochromatic incident waves with IWAVE=1, use has
C           been made of the following guideline:
C           . Let tp = normalized time when periodicity is established
C           . Specify:
C             tp=integer which is 4 or greater for coastal structures
C             NONE=on the order of 2000
C           . Calculate:
C             NTOP=(tp+1)*NONE for (tp+1) wave periods
C             NSTAB=(NTOP-NONE+1) for armor stability during the last
C                                 one wave period
C             NSAVA=(NTOP-NONE)
C             NTIMES=5 to save "A" at normalized time t=tp, (tp+1/4),
C               (tp+2/4), (tp+3/4) and (tp+1) where "A" at t=tp must
C               be the same as "A" at t=(tp+1) if "A" is periodic
C             Moreover, NJUM1 = on the order of NONE/100 so that
C               temporal variations are stored (NONE/NJUM1) times in
C               one wave period
C           -------------------------------------------------------
      IF (IWAVE.EQ.1) THEN
        NSTAT = NTOP-NONE+1
      ELSE
        NSTAT = NSAVA
      ENDIF
      IF (ISAVA.EQ.0) NSAVA=NTOP+1
      IF (ISTAB.EQ.0) NSTAB=NTOP+1
C
C ......... GENERAL DATA
C           S as input:
C           . for IJOB<3: number of spatial nodes along the bottom
C                         below SWL
C           . for IJOB=3: number of nodes between seaward and
C                         landward boundaries
C           Note: S should be so large that delta x between two
C                 adjacent nodes is sufficiently small.
C                 S=100 to 300 has been used.
C           FWP    = bottom friction factor
C           X1,X2  = damping coefficients
C           DELTA  = normalized water depth defining computational
C                    waterline
C           NDELR  = number of "DELRP"s to be specified
```

```
C           DELRP = physical water depth associated with visual or
C                   measured waterline
C                   . in millimeters if ISYST=1 (SI)
C                   . in inches      if ISYST=2 (USCS)
      READ (11,1110) S
      READ (11,1180) FWP
      READ (11,1180) X1,X2
      READ (11,1180) DELTA
      READ (11,1110) NDELR
      CALL CHEOPT (19,IDUM,S,1,N1-1)
      IF (IJOB.LT.3) THEN
        CALL CHEOPT (20,IDUM,NDELR,1,N3)
      ELSE
        NDELR = 0
      ENDIF
      DO 120 L = 1,NDELR
        READ (11,1180) DELRP(L)
  120 CONTINUE
C
C ......... WAVE PROPERTIES
C           HREFP = physical wave height at "reference" location
C                   . in meters if ISYST=1 (SI)
C                   . in feet   if ISYST=2 (USCS)
C           TP    = physical reference wave period, in seconds
C           HREFP and TP are used to normalize the governing
C                   equations
C           KSREF = shoaling coefficient at "reference" location
C           KSSEA = shoaling coefficient at seaward boundary
C           SIGMA is a measure of wave steepness
      READ  (11,1180) HREFP,TP
      READ  (11,1180) KSREF,KSSEA
      SIGMA = TP*DSQRT(GRAV/HREFP)
C
C ......... STRUCTURE GEOMETRY
C           The structure geometry is divided into segments of
C           different inclination
C           NBSEG  = number of segments
C           DSEAP  = physical water depth below SWL at seaward
C                    boundary
C           TSLOPS = tangent of slope used to define
C                    "surf similarity parameter"
C           For segment i starting from the seaward boundary:
C             WBSEG(i)  = physical horizontal width
C             TBSLOP(i) = tangent of slope (+ upslope, - downslope)
C             XBSEG(i)  = physical horizontal distance from seaward
C                         boundary to the segment's seaward-end
C             ZBSEG(i)  = physical water depth below SWL (+ below SWL)
C                         at the segment's seaward-end
C           DSEAP,WBSEG,XBSEG,ZBSEG are in meters if ISYST=1 (SI),
C                                          feet   if ISYST=2 (USCS)
      READ (11,1180) DSEAP
      READ (11,1180) TSLOPS
      READ (11,1110) NBSEG
      CALL CHEOPT (21,IDUM,NBSEG,1,N4)
      IF (IBOT.EQ.1) THEN
```

```
        DO 130 K = 1,NBSEG
          READ (11,1180) WBSEG(K),TBSLOP(K)
  130   CONTINUE
      ELSE
        DO 140 K = 1,NBSEG+1
          READ (11,1180) XBSEG(K),ZBSEG(K)
  140   CONTINUE
      ENDIF
C
C ......... DATA RELATED TO SAVING "B", i.e., temporal variations of
C              total water depth at specified nodes
C           NNOD1      = no. of nodes for which "B" is to be saved
C           NODNO1(I) = I-th node number for which "B" is to be saved
C           FNAME1(I) = name of file associated with NODNO1(I)
      IF (ISAVB.EQ.1) THEN
        DO 150 I = 1,NNOD1
          READ (11,1190) NODNO1(I),FNAME1(I)
  150   CONTINUE
      ENDIF
C
C ......... DATA RELATED TO ARMOR STABILITY AND MOVEMENT
C           NJUM2:   stability number SNR is computed at every NJUM2
C                    time steps (NJUM2=1 has been used)
C           SG     = specific gravity
C           C2     = area coefficient
C           C3     = volume coefficient
C           CD     = drag coefficient
C           CL     = lift coefficient
C           CM     = inertia coefficient
C           TANPHI = armor friction factor
C           AMAX,AMIN = upper and lower bounds of fluid acceleration,
C                    normalized by gravitational acceleration, used
C                    only for ISTAB=1
C           DAP    = physical armor diameter
C                    . in meters IF ISYST=1 (SI)
C                    . in feet   IF ISYST=2 (USCS)
C           NNOD2      = no. of nodes for which "C" is to be saved
C           NODNO2(I) = I-th node number for which "C" is to be saved
C           FNAME2(I) = name of file associated with NODNO2(I)
C           "C" = temporal variations of displacement of armor units
C                 from specified initial nodal locations
C --- To compute SNR = stability number against rolling/sliding:
      IF (ISTAB.GT.0) THEN
        READ (11,1110) NJUM2
        READ (11,1180) C2,C3,SG
        READ (11,1180) CD,CL,CM
        READ (11,1180) TANPHI
        READ (11,1180) AMAX,AMIN
      ENDIF
C --- To compute movement of armor units, additional input is required:
      IF (ISTAB.EQ.2) THEN
        READ (11,1180) DAP
        IF (ISAVC.EQ.1) THEN
          DO 160 I = 1,NNOD2
            READ (11,1190,END=990) NODNO2(I),FNAME2(I)
```

```
  160       CONTINUE
          ENDIF
        ENDIF
C
        IF (INDIC.GT.0) STOP
        RETURN
  990 CONTINUE
      CALL STOPP (4,4)
C
C ... FORMATS
C
 1110 FORMAT (I8)
 1120 FORMAT (14A5)
 1130 FORMAT (2I1,I8)
 1140 FORMAT (I1)
 1150 FORMAT (I1,2X,A20)
 1160 FORMAT (3I1,I8,3I4)
 1170 FORMAT (5I1,I6)
 1180 FORMAT (3F13.6)
 1190 FORMAT (I6,2X,A20)
C
      END
C
C -02------------------ END OF SUBROUTINE INPUT1 --------------------
C #03#################### SUBROUTINE INPUT2 ########################
C
C     This subroutine reads seaward boundary wave profile data
C     if IWAVE>1
C
      SUBROUTINE INPUT2
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      PARAMETER (NICE=500)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /WAVE3/  ETA(N2),ETAIS(N2),ETARS(N2),ETATS(N2)
      COMMON /WAVE4/  ETAMAX,ETAMIN
      CALL CHEPAR (3,2,N2,N2R)
C
C     ETA = given time series of free surface profile (IWAVE>1)
C     ETAMAX and ETAMIN are its maximum and minimum, respectively
C
C     In order to get 'nice' time levels for storage of computed
C     results, NTOP is taken to be a multiplication of NICE specified
C     in the PARAMETER statement of this subroutine
C
      READ (12,1210,END=910) (IDUM,ETA(I),I=1,N2)
  910 CONTINUE
      IDUM = I-1
      WRITE  (*,2910) IDUM
      WRITE (29,2910) IDUM
      NTOP = IDUM+1
```

```
  920 CONTINUE
        NTOP = NTOP-1
        IDUM = MOD(NTOP,NICE)
      IF (IDUM.NE.0) GOTO 920
      ETAMAX = -1.D+03
      ETAMIN =  1.D+03
      DO 100 I = 1,NTOP
        IF (ETA(I).GT.ETAMAX) ETAMAX=ETA(I)
        IF (ETA(I).LT.ETAMIN) ETAMIN=ETA(I)
  100 CONTINUE
      IF (ISAVA.EQ.0) NSAVA=NTOP+1
      IF (ISTAB.EQ.0) NSTAB=NTOP+1
 1210 FORMAT(I10,F10.6)
 2910 FORMAT (/' From Subroutine 3 INPUT2'/
     +' Seaward boundary wave profile has been read from a data file'/
     +' Number of data points read =',I8)
C
      RETURN
      END
C
C -03----------------- END OF SUBROUTINE INPUT2 ---------------------
C #04##################### SUBROUTINE BOTTOM ########################
C
C     This subroutine calculates normalized structure geometry and
C     delta x between two adjacent nodes
C
      SUBROUTINE BOTTOM
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      DIMENSION TSLOPE(N1)
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WL0P
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /BOT1/   DSEAP,DLANDP,FWP
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /BOT4/   NBSEG
      COMMON /BOT5/   WBSEG(N4),TBSLOP(N4),XBSEG(N4),ZBSEG(N4)
      COMMON /STAB1/  C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /STAB2/  SG1,CTAN(N1)
      CALL CHEPAR (4,1,N1,N1R)
      CALL CHEPAR (4,4,N4,N4R)
C
C ... THE FOLLOWING VARIABLES ARE DIMENSIONAL
C
C     TSLOPS = tangent of slope used to define
```

```
C               "surf similarity parameter"
C      BSWL: . for IJOB<3: physical horizontal distance between
C              seaward boundary and initial waterline on slope
C            . for IJOB=3: physical horizontal distance between
C              seaward and landward boundaries
C      DSEAP = water depth below SWL at seaward boundary
C
C      The structure geometry is divided into segments of different
C      inclination
C      NBSEG = number of segments
C      For segment i starting from the seaward boundary:
C        WBSEG(i)  = physical horizontal width
C        TBSLOP(i) = tangent of slope (+ upslope, - downslope)
C        XBSEG(i)  = physical horizontal distance from seaward boundary
C                    to the segment's seaward-end
C        ZBSEG(i)  = physical water depth below SWL (+ below SWL)
C                    at the segment's seaward-end
C      BSWL,DSEAP,WBSEG,XBSEG,ZBSEG are in meters if ISYST=1 (SI),
C                                       feet   if ISYST=2 (USCS)
C
C ... COMPLETE SEGMENT DATA 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 X BETWEEN TWO ADJACENT NODES
C     (dimensional)
C
C     The value of S specified as input corresponds to
C     . for IJOB<3: number of nodes along the bottom below SWL
C     . for IJOB=3: number of nodes between seaward and landward
C                   boundaries
C
      IF (IJOB.LT.3) THEN
        K = 0
  900   CONTINUE
          IF (K.EQ.NBSEG) CALL STOPP (5,6)
          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)
        X     = BSWL/DBLE(S)
      ELSE
```

```
          BSWL = XBSEG(NBSEG+1)
          X    = BSWL/DBLE(S-1)
          DO 130 K = 1,NBSEG+1
            IF (ZBSEG(K).LT.0.D+00) CALL STOPP (7,8)
  130     CONTINUE
        ENDIF
C
C ... CALCULATE STRUCTURE GEOMETRY AT EACH NODE (dimensional)
C
C     JE = landward edge node (IJOB<3) or landward boundary node
C          (IJOB=3)
C     U2INIT(j) = water depth below SWL at node j (+ below SWL)
C               = total water depth U(2,j) at time t=0
C                 (physical, later normalized under the same name)
C     TSLOPE(j) = tangent of local slope at node j
C
      IF (IJOB.LT.3) THEN
        DUM = XBSEG(NBSEG+1)/X
        JE  = INT(DUM)+1
      ELSE
        JE = S
      ENDIF
      IF (JE.GT.N1) THEN
        WRITE  (*,2910) JE,N1
        WRITE (29,2910) JE,N1
        STOP
      ELSE
        JE1 = JE-1
      ENDIF
 2910 FORMAT (/' End Node =',I8,'; N1 =',I8/
     +         ' Slope/Structure is too long.'/
     +         ' Cut it, or change PARAMETER N1.')
C
      DIST = -X
      K    = 1
      XCUM = XBSEG(K+1)
      DO 140 J = 1,JE
        DIST = DIST + X
        IF (DIST.GT.XCUM.AND.K.LT.NBSEG) THEN
          K    = K+1
          XCUM = XBSEG(K+1)
        ENDIF
        U2INIT(J) = ZBSEG(K) - (DIST-XBSEG(K))*TBSLOP(K)
        TSLOPE(J) = TBSLOP(K)
  140 CONTINUE
C
C ... NORMALIZATION
C
C     WTOT = normalized width of computation domain
C     At node j:
C       U2INIT(j) = normalized water depth below SWL (+ below SWL)
C       THETA(j)  = normalized tangent of local slope
C       (XB(j),ZB(j)) = normalized coordinates of the structure
C
      DUM  = TP*DSQRT(GRAV*HREFP)
```

```
      X    = X/DUM
      DIST = -X
      WTOT = DBLE(JE1)*X
      DO 150 J = 1,JE
        U2INIT(J) = U2INIT(J)/HREFP
        THETA(J)  = TSLOPE(J)*SIGMA
        DIST      = DIST + X
        XB(J)     = DIST
        ZB(J)     = -U2INIT(J)
  150 CONTINUE
C
C ... QUANTITIES NEEDED FOR COMPUTATION OF ARMOR STABILITY AND
C     MOVEMENT
C
C     TSLOPE(j) = tangent of local slope at node j
C     SSLOPE(j) = sine of local slope at node j
C     CSLOPE = cosine of local slope
C     TANPHI = armor friction factor
C
      IF (ISTAB.GT.0) THEN
        DO 160 J = 1,JE
          ANGLE     = DATAN(TSLOPE(J))
          CSLOPE    = DCOS(ANGLE)
          SSLOPE(J) = DSIN(ANGLE)
          CTAN(J)   = CSLOPE*TANPHI
  160   CONTINUE
      ENDIF
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=500,N2=30000,N3=3,N4=100,N5=25)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WL0P
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /BOT1/   DSEAP,DLANDP,FWP
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /RUNP1/  NDELR
```

```
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3)
      COMMON /STAB1/  C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /STAB2/  SG1,CTAN(N1)
      COMMON /STAB3/  CSTAB1,CSTAB2,AMAXS,AMINS,E2,E3PRE(N1)
      COMMON /STAB4/  CSTAB3,CSTAB4,CM1,DA,SIGDA,WEIG
      CALL CHEPAR (5,1,N1,N1R)
      CALL CHEPAR (5,3,N3,N3R)
C
C ... PARAMETERS RELATED TO FINITE DIFFERENCE GRIDDING
C
C     T = delta t = constant time step
C     X = delta x = constant grid spacing between two adjacent nodes
C     NTOP = final time level
C     NONE = number of time steps in one wave period
C
      T    = 1.D+00/DBLE(NONE)
      TX   = T/X
      XT   = X/T
      TTX  = T*T/X
      TTXX = T*T/(X*X)
      TWOX = 2.D+00*X
C
C ... PARAMETERS RELATED TO WAVE AND SLOPE CHARACTERISTICS
C
C     KSI           = surf similarity parameter
C     WL0P,WL0      = deep-water wavelengths, physical and normalized,
C                     respectively
C     DSEAP,DSEA    = water depths below SWL at seaward boundary,
C                     physical and normalized, respectively
C     DLANDP,DLAND = water depths below SWL at landward boundary,
C                     physical and normalized, respectively
C                     (only for IJOB=3)
C     FWP,FW        = slope friction factors, physical and normalized,
C                     respectively
C     DELRP,DELTAR = water depths associated with visual or measured
C                     waterline, physical and normalized, respectively
C
      WL0P   = GRAV*(TP*TP)/(2.D+00*PI)
      WL0    = WL0P/DSEAP
      KS     = KSSEA/KSREF
      KSI    = SIGMA*TSLOPS/DSQRT(2.D+00*PI)
      DSEA   = DSEAP/HREFP
      DSEAKS = DSEA/KS
      DSEA2  = DSQRT(DSEA)
      FW     = .5D+00*FWP*SIGMA
      DO 110 L = 1,NDELR
        IF (ISYST.EQ.1) THEN
          DELTAR(L) = DELRP(L)/(1.D+03*HREFP)
        ELSE
          DELTAR(L) = DELRP(L)/(12.D+00*HREFP)
        ENDIF
  110 CONTINUE
      IF (IJOB.EQ.3) THEN
        DLAND  = U2INIT(S)
        DLANDP = DLAND*HREFP
```

```
          DLAND2 = DSQRT(DLAND)
        ENDIF
C
C ... LINEAR WAVELENGTH AND PRELIMINARY 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    = WL0
      FUN1  = WL - WL0*DTANH(TWOPI/WL)
  900 IF (DABS(FUN1).GT.1.D-04) THEN
        FUN2 = 1.D+00 + WL0*TWOPI/(WL*DCOSH(TWOPI/WL))**2
        WL   = WL - FUN1/FUN2
        FUN1 = WL - WL0*DTANH(TWOPI/WL)
      GOTO 900
      ENDIF
      UR    = WL*WL/DSEAKS
      URPRE = UR
C
C ... PARAMETERS FOR ARMOR STABILITY AND MOVEMENT
C
C     JSTAB  = the largest node number for which computation of armor
C              stability or movement will be performed
C     SG     = specific gravity
C     C2     = area coefficient
C     C3     = volume coefficient
C     CD     = drag coefficient
C     CL     = lift coefficient
C     CM     = inertia coefficient
C     TANPHI = armor friction factor
C     AMAX,AMIN = upper and lower bounds of fluid acceleration,
C              normalized by gravitational acceleration
C     E3PRE is prepared for computing E3 in Subr. 19 STABNO
C     DAP,DA = armor diameters, physical and normalized, respectively,
C              used in Subr. 20 MOVE
C     WEIG   = normalized submerged weight of armor unit in
C              Subr. 21 FORCES
C
      IF (ISTAB.GT.0) THEN
        SG1 = SG-1.D+00
        IF (IJOB.EQ.3) JSTAB=JE
      ENDIF
      IF (ISTAB.EQ.1) THEN
        CSTAB1 = 2.D+00*C3**(2.D+00/3.D+00)/(C2*CD)
        CSTAB2 = CM/(SG1*SIGMA)
        E2     = CL*TANPHI/CD
        AMAXS  = AMAX*SIGMA
        AMINS  = AMIN*SIGMA
        DO 120 J = 1,JE
          E3PRE(J) = CSTAB1*CTAN(J)
  120   CONTINUE
      ELSEIF (ISTAB.EQ.2) THEN
        CM1    = CM - 1.D+00
        DA     = DAP/HREFP
```

```
          SIGDA  = SIGMA/DA
          CSTAB3 = C2*CD/(2.D+00*C3*DA)
          CSTAB4 = C2*CL/(2.D+00*C3*DA)
          WEIG   = SIGMA*SG1
        ENDIF
C
      RETURN
      END
C
C -05----------------- END OF SUBROUTINE PARAM  ---------------------
C #06##################### SUBROUTINE INIT1  #########################
C
C     This subroutine assigns initial values
C
      SUBROUTINE INIT1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /RUNP1/  NDELR
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3)
      COMMON /OVER/   OV(4)
      COMMON /STAT/   ELSTAT(3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1)
      COMMON /ENERG/  ENER(4,N1),ENERB(14)
      COMMON /STAB5/  JSNSC,NSNSC,NSNSX(N1)
      COMMON /STAB6/  SNSC,SNR(N1),SNSX(N1)
      COMMON /STAB7/  NMOVE,NSTOP,
     +                ISTATE(N1),NODIN(N1),NODFI(N1),NDIS(N1)
      CALL CHEPAR (6,1,N1,N1R)
      CALL CHEPAR (6,3,N3,N3R)
C
C ... INSTANTANEOUS HYDRODYNAMIC QUANTITIES
C
C     Hydrodynamic quantities at node j:
C       U(1,j)  = volume flux
C       U(2,j)  = total water depth (not less than DELTA for IJOB=3)
C       V(j)    = depth-averaged velocity
C       ELEV(j) = surface elevation above SWL
C
      DO 110 J = 1,JE
        U(1,J) = 0.D+00
        IF (J.LE.S) THEN
          U(2,J) = U2INIT(J)
          ELEV(J)= 0.D+00
        ELSE
          U(2,J) = 0.D+00
          ELEV(J)= ZB(J)
        ENDIF
```

```
            V(J) = 0.D+00
            IF (IJOB.EQ.3.AND.U(2,J).LT.DELTA) U(2,J)=DELTA
  110 CONTINUE
C
C ... HYDRODYNAMIC QUANTITIES FOR STATISTICAL CALCULATIONS
C
C     Subr. 7 INIT2 is used to specify initial values for statistical
C       quantities
C     ELSTAT(1) = mean surface elev. of incident wave at seaw. bdry.
C     At node j:
C       U1STAT(j) = mean volume flux
C     Mean, maximum, and minimum at node j:
C       ESTAT(1,j),ESTAT(2,j),ESTAT(3,j): for ELEV(j)
C       VSTAT(1,j),VSTAT(2,j),VSTAT(3,j): for V(j)
C
      ELSTAT(1) = 0.D+00
      CALL INIT2 (1,U1STAT,1,JE)
      CALL INIT2 (2,ESTAT, 3,JE)
      CALL INIT2 (2,VSTAT, 3,JE)
C
C ... RUNUP
C
C     JMAX = the largest node number reached by computational
C            waterline
C     RSTAT(1),RSTAT(2),RSTAT(3) = mean, maximum, and minimum RUNUPS
C     (See Subr. 13 LANDBC for RUNUPS)
C
      JMAX = S
      IF (IJOB.LT.3) CALL INIT2 (2,RSTAT,3,NDELR)
C
C ... OVERTOPPING
C     Computed during N=NSTAT to N=NTOP in Subr. 15 OVERT
C
C     OV(1) = normalized average overtopping rate
C     OV(2) = normalized time when OV(4) occurs after time of N=NSTAT
C     OV(3) = normalized overtopping duration
C     OV(4) = normalized maximum overtopping rate
C     OV(2) and OV(3) will eventually be normalized in Subr. 23 STAT2
C
      IF (IJOB.EQ.2) CALL INIT2 (1,OV,1,4)
C
C ... ARMOR STABILITY AND MOVEMENT
C
C     Stability numbers:
C       SNR(j) = stability number against rolling/sliding at node j
C       SNSX(j) = local stability number = minimum of SNR at node j
C       SNSC = critical stab. number = min. of SNSX along the slope
C     Armor units movement:
C       NMOVE = no. of units dislodged from their initial locations
C       NSTOP = no. of units stopped after moving
C       ISTATE(j) indicates the state of armor unit initially located
C                 at node j: 0=stationary, 1=moving, 2=stopped
C
      IF (ISTAB.EQ.1) THEN
        SNSC = 1.D+03
```

```
          DO 120 J = 1,JE
            SNSX(J) = 1.D+03
  120     CONTINUE
        ELSEIF (ISTAB.EQ.2) THEN
          NMOVE = 0
          NSTOP = 0
          DO 130 J = 1,JE
            ISTATE(J) = 0
  130     CONTINUE
        ENDIF
C
C ... WAVE ENERGY (normalized time-averaged quantities)
C
C       At node j:
C         ENER(1,j) = energy per unit surface area
C         ENER(2,j) = energy flux per unit width
C       Rate of energy dissipation, at node j:
C         ENER(3,j): due to bottom friction, per unit bottom area
C         ENER(4,j): due to wave breaking, per unit surface area
C
      IF (IENERG.EQ.1) CALL INIT2 (1,ENER,4,JE)
C
      RETURN
      END
C
C -06------------------ END OF SUBROUTINE INIT1 --------------------
C #07###################### SUBROUTINE INIT2 #######################
C
C       This subroutine facilitates assignment of initial values in
C       Subr. 6 INIT1
C
      SUBROUTINE INIT2 (MODE,VAL,ND1,ND2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VAL(ND1,ND2)
      IF (MODE.EQ.1) THEN
        DO 120 I = 1,ND1
          DO 110 J = 1,ND2
            VAL(I,J) = 0.D+00
  110     CONTINUE
  120   CONTINUE
      ELSE
        DO 130 J = 1,ND2
          VAL(1,J) =  0.D+00
          VAL(2,J) = -1.D+03
          VAL(3,J) =  1.D+03
  130   CONTINUE
      ENDIF
C
      RETURN
      END
C
C -07------------------ END OF SUBROUTINE INIT2 --------------------
C #08###################### SUBROUTINE INWAV #######################
C
```

```
C     This subroutine computes incident wave profile at seaward
C     boundary if IWAVE=1
C     Wave Profile: Stokes II if UR<26
C                   Cnoidal otherwise
C
      SUBROUTINE INWAV
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      DOUBLE PRECISION K,M,KC2,KC,KS,KSREF,KSSEA,KSI
      DIMENSION ETAU(N2)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /WAVE3/  ETA(N2),ETAIS(N2),ETARS(N2),ETATS(N2)
      COMMON /WAVE4/  ETAMAX,ETAMIN
      COMMON /WAVE5/  K,E,M,KC2
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      CALL CHEPAR (8,2,N2,N2R)
C
C ... CONSTANTS
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
C
C ... COMPUTE HALF OF WAVE PROFILE (unadjusted)
C
C     ETAMAX = normalized maximum surface elevation
C     ETAMIN = normalized minimum surface elevation
C     ETAU = unadjusted surface elevation
C     N0 = 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
C
        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
          TIME = DBLE(N-1)*T
          ETAU(N) = .5D+00*DCOS(TWOPI*TIME)+AMP2*DCOS(FOURPI*TIME)
          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) N0=N
          ENDIF
  110   CONTINUE
        ETAMIN = ETAU(NHALF1)
        ETAMAX = ETAU(1)
C
      ELSE
C
C ----- Cnoidal Wave Profile
C
C       FINDM is to find the parameter M of the Jacobian elliptic func.
C       See Func. 10 CEL and Subr. 11 SNCNDN
C
        CALL FINDM (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
        ETAMAX = ETAMIN + KS
        DO 120 N = 1,NHALF1
          TIME = DBLE(N-1)*T
          TETA = 2.D+00*K*TIME
          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) N0=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
C     so that elevation=0 at time=0 and decreases initially with time
C
C     ETAU = unadjusted surface elevation
C     ETA  = adjusted surface elevation
C
      NMARK = NONE-N0+2
      DO 140 N = 1,NONE1
        IF (N.LE.NMARK) THEN
          ETA(N) = ETAU(N+N0-1)
        ELSE
          ETA(N) = ETAU(N-NMARK+1)
        ENDIF
```

```
  140 CONTINUE
C
C ... SAVE COMPUTED WAVE PROFILE
C
      WRITE (34,3410) (ETA(I),I=1,NONE1)
 3410 FORMAT(8F9.6)
C
      RETURN
      END
C
C -08----------------- END OF SUBROUTINE INWAV ---------------------
C #09##################### SUBROUTINE FINDM #########################
C
C     This subroutine computes the parameter M (MLIL<M<MBIG) of the
C     Jacobian elliptic functions
C
      SUBROUTINE FINDM (M)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K,M,KC2,KC,MSAV,MLIL,MBIG
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      DATA SMALL,MLIL /1.D-07,.8D+00/
      DATA INDI,I     /0,0/
      SIGDT = DSQRT(2.D+00*PI*WL0)
      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 not satisfied')
C
      RETURN
      END
C
C -09------------------ END OF SUBROUTINE FINDM --------------------
C #10################ 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 -10------------- END OF DOUBLE PRECISION FUNCTION CEL ------------
C #11######################## 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 -11----------------- END OF SUBROUTINE SNCNDN --------------------
C #12##################### SUBROUTINE MARCH ########################
C
C     This subroutine marches the computation from time level (N-1)
C     to time level N excluding seaward and landward boundaries
C     which are treated separately
C
      SUBROUTINE MARCH (N,M)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
```

```
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      IF (N.EQ.1) CALL CHEPAR (12,1,N1,N1R)
C
C     U(1,j) and U(2,j) at time level N are computed as follows:
C     . at j=2,3,...,JDAM                   : WITH numerical damping term
C     . at j=(JDAM+1),(JDAM+2),...,JLAX : NO numerical damping term
C     JE1=(JE-1) indicates the node next to the landward edge node
C
      IF (IJOB.LT.3) THEN
        JDAM = S-2
        JLAX = S
        IF (IJOB.EQ.2.AND.S.EQ.JE) JLAX=JE1
      ELSE
        JDAM = JE1
        JLAX = JE1
      ENDIF
      JLAX1 = JLAX+1
C
C ... COMPUTE ELEMENTS OF MATRICES
C
C     Subr. 26 MATAFG computes non-constant elements of Matrices A
C           and G, and the elements of Matrix F
C     Subr. 28 MATS computes the first element of Matrix S
C     Subr. 27 MATGJR, Subr. 29 MATD, and Subr. 30 MATU compute
C           the elements of Matrices g, D, and U, respectively
C     Subr. 30 MATU computes values of U at time level N using
C           the results obtained from the other four subroutines
C
      CALL MATAFG (N,1,JLAX1)
      CALL MATGJR (N,1,JLAX)
      CALL MATS   (N,2,JLAX)
      CALL MATD   (N,JDAM,JLAX)
      CALL MATU   (N,2,JLAX)
C
C ... ABORT COMPUTATION IF WATER DEPTH AT (S-1) <or= DELTA
C
      IF (U(2,M).LE.DELTA) THEN
        WRITE  (*,2910) U(2,M),DELTA,S,N
        WRITE (29,2910) U(2,M),DELTA,S,N
        STOP
      ENDIF
 2910 FORMAT (/' From Subroutine 12 MARCH'/
     +' U(2,S-1) is less than or equal to DELTA'/' U(2,S-1) =',D12.3/
     +' DELTA     =',D12.3/' S =',I8/' N =',I8/' Program Aborted')
C
C ... COMPLETE THE COMPUTATION OF HYDRODYNAMIC QUANTITIES
C
C     Water depth h is taken to be not less than DELTA
C     for submerged structures
C
      DO 100 J = 2,JLAX
        IF (IJOB.EQ.3.AND.U(2,J).LT.DELTA) U(2,J)=DELTA
        V(J)     = U(1,J)/U(2,J)
        ELEV(J) = U(2,J)-U2INIT(J)
```

```
  100 CONTINUE
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 conditions
C
      SUBROUTINE LANDBC (N,K,M,ETAT)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE3/  ETA(N2),ETAIS(N2),ETARS(N2),ETATS(N2)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /RUNP1/  NDELR
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3)
      COMMON /OVER/   OV(4)
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      IF (N.EQ.1) THEN
        CALL CHEPAR (13,1,N1,N1R)
        CALL CHEPAR (13,2,N2,N2R)
        CALL CHEPAR (13,3,N3,N3R)
      ENDIF
C
C ... MANAGE LANDWARD B.C.
C
C     Subr. 14 RUNUP computes shoreline movement if computational
C           waterline is on the slope (IJOB<3)
C     Subr. 15 OVERT computes hydrodynamic quantities at landward edge
C           node if overtopping occurs (IJOB=2)
C     ALPHA = landward-advancing characteristics
C     ETAT  = surface elevation due to transmitted wave at
C             landward boundary (IJOB=3)
C     DLAND = norm. water depth below SWL at landw. boundary (IJOB=3)
C     S used in Subr. 14 RUNUP is the value at time level (N-1)
C
      IF (IJOB.EQ.1) THEN
        CALL RUNUP (N,K,M)
        IF (S.GT.JE1) THEN
          WRITE  (*,2910) N,S,JE
          WRITE (29,2910) N,S,JE
          STOP
        ENDIF
```

```
      ELSEIF (IJOB.EQ.2) THEN
        IF (S.LT.JE) THEN
          CALL RUNUP (N,K,M)
        ELSE
          CALL OVERT (N,M)
        ENDIF
      ELSE
C       For IJOB=3: Wave Transmission over Submerged Breakwater
        DUM     = TX*(VSN+C(S))*(VSN-VMN+2.D+00*(C(S)-C(M)))
        ALPHA   = VSN+2.D+00*C(S) - DUM - T*THETA(S)
        ETAT    = ALPHA*DLAND2/2.D+00 - DLAND
        U(2,S)  = DLAND + ETAT
        V(S)    = ALPHA - 2.D+00*DSQRT(U(2,S))
        U(1,S)  = U(2,S)*V(S)
        ELEV(S) = U(2,S) - U2INIT(S)
      ENDIF
 2910 FORMAT (/' From Subroutine 13 LANDBC:'/
     +   ' N =',I8,'; S =',I8,'; End Node =',I8/
     +   ' Slope is not long enough to accomodate shoreline movement'/
     +   ' Specify longer slope or choose overtopping computation')
C
C ... CONDITIONS LANDWARD OF NEW WATERLINE NODE S AT TIME LEVEL N
C
      IF (IJOB.LT.3) THEN
        L = S+1
        U(1,L)  = 0.D+00
        U(2,L)  = 0.D+00
        V(L)    = 0.D+00
        ELEV(L) = ZB(L)
      ENDIF
C
C ... COMPUTE RUNUPS ASSOCIATED WITH DEPTHS (DELTAR(L),L=1,NDELR)
C     (Assume water depth decreases landward and U(2,S+1)=0.)
C     If IJOB<3, NDELR>0, DO 100 performed
C     If IJOB=3, NDELR=0, DO 100 void
C
C     NSTAB  = time level when computation of armor stability or
C              movement begins
C     JSTAB  = the largest node number based on DELTAR(1)
C              for armor stability or movement
C              Note:
C              For IJOB=3, JSTAB=JE was defined in Subr. 5 PARAM
C     DELTAR = water depth associated with visual or measured
C              waterline
C     RUNUPS = free surface elevation where the water depth equals
C              DELTAR
C     NDELR  = number of DELTARs
C
      IF (NDELR.GE.1) THEN
        DO 100 L = 1,NDELR
          IF (IJOB.EQ.2.AND.S.EQ.JE.AND.U(2,S).GE.DELTAR(L)) THEN
            IF (L.EQ.1.AND.N.GE.NSTAB) JSTAB=S
            RUNUPS(L) = ZB(S) + U(2,S)
          ELSE
            INDIC = 0
```

```
            J = -1
  900       CONTINUE
              J = J + 1
              IF  (U(2,S-J).GE.DELTAR(L)) THEN
                INDIC = 1
                NRUN1 = S-J
                NRUN2 = S-J+1
                IF (L.EQ.1.AND.N.GE.NSTAB) JSTAB=NRUN1
                DEL1   = U(2,S-J)
                DEL2   = U(2,S-J+1)
                RUN    = (ZB(NRUN2)-ZB(NRUN1))*(DEL1-DELTAR(L))
                RUN    = RUN/(DEL1-DEL2)
                RUN    = RUN + ZB(NRUN1)
                RUNUPS(L) = RUN + DELTAR(L)
              ENDIF
            IF (INDIC.EQ.0) GOTO 900
          ENDIF
  100   CONTINUE
      ENDIF
C
C ... STATISTICAL CALCULATIONS
C
C     NSTAT = time level when statistical calculations begin
C     IJOB<3:
C       RSTAT(1,L),RSTAT(2,L),RSTAT(3,L) = mean, max. and min.
C         RUNUPS(L), respectively
C     IJOB=2: Overtopping computed during N=NSTAT to N=NTOP
C       OV(1) = normalized average overtopping rate
C       OV(2) = normalized time when OV(4) occurs after time of N=NSTAT
C       OV(3) = normalized overtopping duration
C       OV(4) = normalized maximum overtopping rate
C       OV(2) and OV(3) will be  normalized in Subr 23 STAT2
C     IJOB=3:
C       ETAT = surface elevation due to transmitted wave at
C         landward boundary
C       ETAT is saved in a time-array as ETATS starting from N=NSTAT
C
      IF (N.GE.NSTAT) THEN
        IF (IJOB.LT.3) THEN
          CALL STAT1 (2,RSTAT,RUNUPS,3,NDELR)
          IF (IJOB.EQ.2) THEN
            OV(1) = OV(1) + U(1,JE)
            IF  (U(1,JE).GT.0.D+00) OV(3)=OV(3)+1.D+00
            IF  (U(1,JE).GT.OV(4)) THEN
              OV(2) = DBLE(N-NSTAT+1)
              OV(4) = U(1,JE)
            ENDIF
          ENDIF
        ELSE
          ETATS(N-NSTAT+1) = ETAT
        ENDIF
      ENDIF
C
      RETURN
      END
```

```
C
C -13------------------ END OF SUBROUTINE LANDBC -------------------
C #14####################### SUBROUTINE RUNUP #######################
C
C     This subroutine computes waterline movement for IJOB=1 and if
C     no overtopping occurs for IJOB=2
C
      SUBROUTINE RUNUP (N,K,M)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      DIMENSION USN2(2),US1N1(2)
      INTEGER S,SNEW
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /MATRIX/ A1(2,N1),F(2,N1),G1(N1),GJR(2,N1),S1(N1),D(2,N1)
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      IF (N.EQ.1) CALL CHEPAR (14,1,N1,N1R)
C
C ... ADJUST VALUES AT S IF U(2,S)>U(2,S-1)
C
      IF (U(2,S).GE.U(2,M)) THEN
        V(S)     = 2.D+00*V(M) - V(S-2)
        U(2,S)   = 2.D+00*U(2,M) - U(2,S-2)
        IF (ABS(V(S)).GT.ABS(V(M))) V(S)=.9*V(M)
        IF (U(2,S).LT.0.D+00) U(2,S)=.5*U(2,M)
        IF (U(2,S).GT.U(2,M)) U(2,S)=.9*U(2,M)
        U(1,S)   = V(S)*U(2,S)
        ELEV(S) = U(2,S) - U2INIT(S)
        WRITE  (*,2910) S,N,U(2,S),U(2,M)
        WRITE (29,2910) S,N,U(2,S),U(2,M)
      ENDIF
 2910 FORMAT (/' From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at',
     +        ' S =',I8,'; N =',I8/' Adjusted values:',
     +        ' U(2,S) =',E12.3,'; U(2,S-1) =',E12.3)
C
C ... DETERMINE THE NEXT WATERLINE NODE
C
      IF (U(2,S).LE.DELTA) THEN
        SNEW = M
      ELSE
        V(K)    = 2.D+00*V(S) - V(M)
        U(2,K) = 2.D+00*U(2,S) - U(2,M)
        U(1,K) = V(K)*U(2,K)
        IF (U(2,K).LE.DELTA) THEN
          SNEW = S
        ELSE
C         ------------------------------------------------------------
C         *        USN2(i),VSN2 = U(i,S) and V(S), respectively,
C                  at time level (N+1), i=1,2
          CALL MATAFG (2,M,K)
```

```
          CALL MATGJR (2,M,S)
          CALL MATS   (2,S,S)
          DUM1 = TX*((F(1,K)-F(1,M))/2.D+00+X*G1(S))
          DUM2 = TTXX*(GJR(1,S)-GJR(1,M))
          DUM3 = TX*(F(2,K)-F(2,M))
          DUM4 = TTXX*(GJR(2,S)-GJR(2,M))
          USN2(1) = U(1,S)-DUM1+(DUM2-TTX*S1(S))/2.D+00
          USN2(2) = U(2,S)-(DUM3-DUM4)/2.D+00
          VSN2 = USN2(1)/USN2(2)
C         ------------------------------------------------------------
C         *        US1N1(i),VS1N1 = U(i,S+1) and V(S+1), respectively,
C                  at time level N, i=1,2
          VS = V(S)
          IF (DABS(VS).LT.DELTA) VS=DSIGN(DELTA,VS)
          VS1N1 = V(M)-(XT*(VSN2-VSN)+U(2,K)-U(2,M)+TWOX*THETA(S))/VS
          US1N1(1) = U(1,M) - XT*(USN2(2)-USN(2))
          US1N1(2) = US1N1(1)/VS1N1
C         ------------------------------------------------------------
          IF (DABS(VS1N1).LE.DELTA) THEN
            SNEW = S
          ELSE
            IF (US1N1(2).LE.U(2,S)) THEN
              IF (US1N1(2).LE.DELTA) THEN
                SNEW = S
              ELSE
                SNEW   = K
                U(2,K) = US1N1(2)
                U(1,K) = US1N1(1)
                V(K)   = VS1N1
              ENDIF
            ELSE
              IF (U(2,K).LE.U(2,S)) THEN
                SNEW = K
              ELSE
                SNEW = S
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
      IF (SNEW.EQ.K) ELEV(K)=U(2,K)-U2INIT(K)
      S = SNEW
C
C     S at time level N has been found
C
      RETURN
      END
C
C -14------------------  END OF SUBROUTINE RUNUP  -------------------
C #15####################  SUBROUTINE OVERT  ########################
C
C     This subroutine computes quantities at landward-end node for
C     IJOB=2 if overtopping occurs, that is, S=JE and M=(S-1)=JE1
C
      SUBROUTINE OVERT (N,M)
```

```
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /MATRIX/ A1(2,N1),F(2,N1),G1(N1),GJR(2,N1),S1(N1),D(2,N1)
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      DATA INDI /0/
      SAVE INDI
      IF (INDI.EQ.0) THEN
        CALL CHEPAR (15,1,N1,N1R)
        INDI = 1
      ENDIF
      IF (VMN.GT.C(M)) THEN
        U(1,S) = USN(1) - TX*(F(1,S)-F(1,M)) - T*(THETA(S)*USN(2))
        U(2,S) = USN(2) - TX*(USN(1)-UMN(1))
        V(S)   = U(1,S)/U(2,S)
      ELSE
        VCS  = VSN + 2.D+00*C(S)
        VCM  = VMN + 2.D+00*C(M)
        V(S) = (VCS-TX*(VSN+C(S))*(VCS-VCM)-T*(THETA(S)))/3.D+00
        U(2,S) = V(S)*V(S)
        U(1,S) = V(S)*U(2,S)
      ENDIF
      IF (U(2,S).LE.DELTA) THEN
        S = M
      ELSE
        ELEV(S) = U(2,S) - U2INIT(S)
      ENDIF
C
      RETURN
      END
C
C -15----------------- END OF SUBROUTINE OVERT ---------------------
C #16##################### SUBROUTINE SEABC ########################
C
C     This subroutine treats seaward boundary conditions at node j=1
C
      SUBROUTINE SEABC (N,ETAR)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
```

```
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /WAVE3/  ETA(N2),ETAIS(N2),ETARS(N2),ETATS(N2)
      COMMON /BOT1/   DSEAP,DLANDP,FWP
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      IF (N.EQ.1) THEN
        CALL CHEPAR (16,1,N1,N1R)
        CALL CHEPAR (16,2,N2,N2R)
      ENDIF
C
C ... ESTIMATE ETAR
C
C     BETA = seaward-advancing characteristics
C     ETAR = surface elevation due to reflected wave at
C            seaward boundary
C     A correction term included in ETAR if INONCT=1 to improve
C            prediction of wave set-down and setup on beach
C
      VC1 = -V1N+2.D+00*C(1)
      VC2 = -V2N+2.D+00*C(2)
      BETA = VC1 - TX*(V1N-C(1))*(VC2-VC1) + T*THETA(1)
      ETAR = BETA*DSEA2/2.D+00 - DSEA
      IF (INONCT.EQ.1) ETAR=ETAR-KS*KS/(16.D+00*DSEA)
C
C ... VALUES AT NODE ONE
C
      IF (IWAVE.EQ.1) THEN
        NWAVE    = MOD(N,NONE) + 1
        U(2,1)   = DSEA+ETAR+ETA(NWAVE)
        ELEV(1)  = ETA(NWAVE)+ETAR
      ELSEIF (IWAVE.EQ.2) THEN
        U(2,1)   = DSEA+ETAR+ETA(N)
        ELEV(1)  = ETA(N)+ETAR
      ELSE
        U(2,1)   = DSEA+ETA(N)
        ELEV(1)  = ETA(N)
      ENDIF
      V(1)    = 2.D+00*DSQRT(U(2,1))-BETA
      U(1,1)  = U(2,1)*V(1)
C
C ... STATISTICAL CALCULATIONS
C
C     NSTAT = time level when statistical calculations begin
C     ETAI  = surface elevation associated with incident wave at
C             seaward boundary
C     ETAR  = surface elevation associated with reflected wave at
C             seaward boundary
C     ETAI is saved in a time-array as ETAIS starting from N=NSTAT
C     ETAR is saved in a time-array as ETARS starting from N=NSTAT
C
      IF (N.GE.NSTAT) THEN
        IF (IWAVE.EQ.1) THEN
          NWAVE = MOD(N,NONE)+1
```

```
          ETAI  = ETA(NWAVE)
        ELSEIF (IWAVE.EQ.2) THEN
          ETAI  = ETA(N)
        ELSE
          ETAI  = ETA(N)-ETAR
        ENDIF
        ETAIS(N-NSTAT+1) = ETAI
        ETARS(N-NSTAT+1) = ETAR
      ENDIF
C
      RETURN
      END
C
C -16----------------- END OF SUBROUTINE SEABC ---------------------
C #17####################  SUBROUTINE ENERGY  ########################
C
C     This subroutine computes quantities related to wave energy
C
      SUBROUTINE ENERGY (N)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      DIMENSION E(3)
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /ENERG/  ENER(4,N1),ENERB(14)
      IF (N.EQ.NSTAT) CALL CHEPAR (17,1,N1,N1R)
C
C     E's are instantaneous quantities, ENER's time-averaged
C     quantities
C
C     At node j:
C       E(1),ENER(1,j) = norm. energy per unit surface area
C       E(2),ENER(2,j) = norm. energy flux per unit width
C     Normalized rate of energy dissipation at node j:
C       E(3),ENER(3,j): due to bottom friction, per unit bottom area
C            ENER(4,j): due to wave breaking, per unit surface area
C
      DO 120 J = 1,S
        E(1) = (U(1,J)*V(J)+ELEV(J)*ELEV(J))/2.D+00
        IF (U(2,J).LT.ELEV(J)) E(1)=E(1)-(U(2,J)-ELEV(J))**2/2.D+00
        E(2) = U(1,J)*(V(J)*V(J)/2.D+00+ELEV(J))
        E(3) = FW*DABS(V(J))*V(J)*V(J)
        DO 110 I = 1,3
          ENER(I,J) = ENER(I,J) + E(I)
  110   CONTINUE
  120 CONTINUE
C
C     ENER's are time-averaged in Subr. 25 BALANE
```

```
C
      RETURN
      END
C
C -17----------------- END OF SUBROUTINE ENERGY --------------------
C #18##################### SUBROUTINE STAT1 ########################
C
C     For MODE=1, VAL1(1,J) is sum of VAL2(J)
C     For MODE=2, VAL1(1,J) is sum of VAL2(J)
C                 VAL1(2,J) is maximum of VAL2(J)
C                 VAL1(3,J) is minimum of VAL2(J)
C
      SUBROUTINE STAT1 (MODE,VAL1,VAL2,ND1,ND2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VAL1(ND1,ND2),VAL2(ND2)
      IF (MODE.EQ.1) THEN
        DO 120 I = 1,ND1
          DO 110 J = 1,ND2
            VAL1(I,J) = VAL1(I,J)+VAL2(J)
  110     CONTINUE
  120   CONTINUE
      ELSE
        DO 130 J = 1,ND2
          VAL1(1,J) = VAL1(1,J)+VAL2(J)
          IF (VAL2(J).GT.VAL1(2,J)) VAL1(2,J)=VAL2(J)
          IF (VAL2(J).LT.VAL1(3,J)) VAL1(3,J)=VAL2(J)
  130   CONTINUE
      ENDIF
C
      RETURN
      END
C
C -18---------------- END OF SUBROUTINE STAT1 ---------------------
C #19#################### SUBROUTINE STABNO ########################
C
C     This subroutine computes stability number against
C     rolling/sliding, SNR
C
      SUBROUTINE STABNO (N)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /STAB2/  SG1,CTAN(N1)
      COMMON /STAB3/  CSTAB1,CSTAB2,AMAXS,AMINS,E2,E3PRE(N1)
      COMMON /STAB5/  JSNSC,NSNSC,NSNSX(N1)
      COMMON /STAB6/  SNSC,SNR(N1),SNSX(N1)
      IF (N.EQ.NSTAB) CALL CHEPAR (19,1,N1,N1R)
```

```
C
C ... FLUID ACCELERATION
C     Computed in Subr. 22 ACCEL
C
      CALL ACCEL (N)
C
C ... STABILITY NUMBER SNR
C
C     SNR(j) = stability number against rolling/sliding at node j
C     SNR is computed for j=1,2,...,JSTAB where JSTAB is defined
C       in Subr. 13 LANDBC
C
      DO 110 J = 1,JSTAB
C
        IF (DABS(V(J)).LT.1.D-03) THEN
C         ------------ Avoid having very small velocity values
C                      SNR=1000 indicates very stable units
          SNR(J) = 1.D+03
        ELSE
C         ------------ Impose lower and upper bounds of fluid
C                      acceleration
          IF (DUDT(J).GT.AMAXS) DUDT(J)=AMAXS
          IF (DUDT(J).LT.AMINS) DUDT(J)=AMINS
C         ------------ SNR=-1000 indicates that AMAX and AMIN
C                      specified in Subr. 2 INPUT1 needs to be
C                      modified
          VALUE = CSTAB2*DUDT(J)-SSLOPE(J)
          ABSV  = DABS(VALUE)
          IF (ABSV.GT.CTAN(J)) THEN
            SNR(J) = -1.D+03
            WRITE  (*,2910) N,J
            WRITE (29,2910) N,J
            STOP
          ENDIF
C         ------------ Compute SNR
          E1   = VALUE*CSTAB1/(V(J)*DABS(V(J)))
          E3   = E3PRE(J)/(V(J)*V(J))
          E1E2 = -E1*E2
          IF (E1.LT.0.D+00.AND.E2.GT.1.D+00.AND.E3.LT.E1E2) THEN
            SNR(J) = (E3+E1)/(E2-1.D+00)
          ELSE
            SNR(J) = (E3-E1)/(E2+1.D+00)
          ENDIF
        ENDIF
 2910 FORMAT (' From Subr. 19 STABNO'/' Armor stability impossible'/
     +        ' N =',I8,'; J =',I8)
C
  110 CONTINUE
C
C ... FIND SNSX
C
C     SNSX(j)  = local stability number = minimum of SNR at node j
C     NSNSX(j) = time level when SNSX(j) occurs
C
      DO 120 J = 1,JSTAB
```

```
         IF (SNR(J).LT.SNSX(J)) THEN
           SNSX(J)  = SNR(J)
           NSNSX(J) = N
         ENDIF
  120 CONTINUE
C
      RETURN
      END
C
C -19---------------- END OF SUBROUTINE STABNO --------------------
C #20####################### SUBROUTINE MOVE #######################
C
C     This subroutine computes movement of armor units
C
      SUBROUTINE MOVE (N)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /STAB1/  C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /STAB2/  SG1,CTAN(N1)
      COMMON /STAB4/  CSTAB3,CSTAB4,CM1,DA,SIGDA,WEIG
      COMMON /STAB5/  JSNSC,NSNSC,NSNSX(N1)
      COMMON /STAB6/  SNSC,SNR(N1),SNSX(N1)
      COMMON /STAB7/  NMOVE,NSTOP,
     +                ISTATE(N1),NODIN(N1),NODFI(N1),NDIS(N1)
      COMMON /STAB8/  VA(N1),XAA(N1),XA(N1)
      IF (N.EQ.NSTAB) THEN
        CALL CHEPAR (20,1,N1,N1R)
        CALL CHEPAR (20,5,N5,N5R)
        DO 110 J = 1,JE
          NODIN(J) = J
          NODFI(J) = J
          VA(J)    = 0.D+00
          XAA(J)   = 0.D+00
 110    CONTINUE
      ENDIF
C
C ... FLUID ACCELERATION
C     Computed in Subr. 22 ACCEL
C
      CALL ACCEL (N)
C
C ... GENERAL TERMS
C
```

```
C      Counters:
C        NMOVE = number of armor units dislodged from their initial
C                locations
C        NSTOP = number of armor units stopped after moving
C      Node numbers:
C        JSTAB = the largest node number for which armor movement
C          is computed
C        For moving/stopped armor unit number j:
C          NODIN(j) = node number where it was initially located
C          NODFI(j) = node number closest to the armor unit at the end
C                     of each time step
C      Dynamics:
C        FDES = normalized destabilizing force
C        FR   = normalized resistance force
C        FDES and FR are computed in Subr. 21 FORCES
C        ISTATE(j) indicates the state of armor unit initially located
C          at node j: 0=stationary, 1=moving, 2=stopped
C        For moving/stopped armor unit number j:
C          VA(j) = normalized velocity
C          XAA(j), XA(j) = displacement from its initial location,
C            normalized by TP*sqrt(GRAV*HREFP) and DAP, respectively
C          NDIS(j) = time level N when it starts moving the first time
C      It is assumed that once an armor unit is dislodged from a node,
C        no other unit will be dislodged from the same node.
C
      DO 120 J = 1,JSTAB
        IF (ISTATE(J).EQ.0) THEN
C       ......... STATIONARY ARMOR UNIT .............................
C                 Check whether the unit at node j starts moving
          CALL FORCES (V(J),J,FDES,FR,DUDT,SSLOPE,CTAN)
          IF (DABS(FDES).GT.FR) THEN
            IF (FDES.LT.0.D+00) FR=-FR
            NMOVE     = NMOVE+1
            ISTATE(J) = 1
            NDIS(J)   = N
            VA(J)     = (FDES-FR)*T/(SG+CM1)
            XAA(J)    = .5D+00*VA(J)*T
            NODFI(J)  = J + NINT(XAA(J)/X)
          ENDIF
        ELSEIF (ISTATE(J).EQ.1) THEN
C       ......... MOVING ARMOR UNIT .................................
C                 Follow the moving unit initially located at node j
          NOD  = NODFI(J)
          VREL = V(NOD) - VA(J)
          CALL FORCES (VREL,NOD,FDES,FR,DUDT,SSLOPE,CTAN)
          FR        = FR*(VA(J)/DABS(VA(J)))
          DVA       = (FDES-FR)*T/(SG+CM1)
          DXAA      = (VA(J)+.5D+00*DVA)*T
          VA(J)     = VA(J) + DVA
          XAA(J)    = XAA(J) + DXAA
          NODFI(J) = NODIN(J) + NINT(XAA(J)/X)
C                 Check whether the moving unit identified by the
C                 initial node j stops at the end of each time step
          IF (DABS(VA(J)).LT.1.D-06.AND.DABS(FDES).LT.DABS(FR)) THEN
            ISTATE(J) = 2
```

```
            NSTOP      = NSTOP+1
          ENDIF
        ELSE
C       ......... STOPPED ARMOR UNIT ..................................
C                Check whether the stopped armor unit located
C                initially at node j resumes movement
          NOD = NODFI(J)
          CALL FORCES (V(NOD),NOD,FDES,FR,DUDT,SSLOPE,CTAN)
          IF (DABS(FDES).GT.FR) THEN
            IF (FDES.LT.0.D+00) FR=-FR
            VA(J)      = (FDES-FR)*T/(SG+CM1)
            XAA(J)     = XAA(J) + .5D+00*VA(J)*T
            NODFI(J)   = NODIN(J) + NINT(XAA(J)/X)
            NSTOP      = NSTOP-1
            ISTATE(J)  = 1
          ENDIF
        ENDIF
  120 CONTINUE
C
C ... COMPUTE XA
C
      DO 130 J = 1,JSTAB
        IF (ISTATE(J).GE.1) XA(J)=XAA(J)*SIGDA
  130 CONTINUE
C
      RETURN
      END
C
C -20------------------ END OF SUBROUTINE MOVE ---------------------
C #21####################### SUBROUTINE FORCES #######################
C
C     This subroutine computes destabilizing force FDES
C     and resistance force FR used in Subr. 20 MOVE
C
      SUBROUTINE FORCES (VELO,NODE,FDES,FR,DUDT,SSLOPE,CTAN)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      DIMENSION DUDT(N1R),SSLOPE(N1R),CTAN(N1R)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /STAB1/  C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /STAB4/  CSTAB3,CSTAB4,CM1,DA,SIGDA,WEIG
C
C     WEIG = normalized submerged weight of armor unit defined in
C            Subr. 5 PARAM
C     WSIN = component of WEIG parallel to local slope
C     WCOS = component of WEIG normal to local slope
C     FD   = normalized drag force
C     FL   = normalized lift force
C     FI   = normalized inertia force due to fluid only
C     FDES = normalized destabilizing force = FD+FI-WSIN
C     FR   = normalized resistance or friction force
C          = 0 if (WCOS-FL)<0; no contact with other units
C     Note: FR returned to Subr. 20 MOVE is positive or zero
```

```
C
      WSIN = WEIG*SSLOPE(NODE)
      WCOS = WEIG*CTAN(NODE)/TANPHI
      FD   = SIGMA*CSTAB3*VELO*DABS(VELO)
      FL   = SIGMA*CSTAB4*VELO*VELO
      FI   = CM*DUDT(NODE)
      FDES = FD + FI - WSIN
      FR   = (WCOS-FL)*TANPHI
      IF (FR.LE.0.D+00) FR=0.D+00
C
      RETURN
      END
C
C -21----------------- END OF SUBROUTINE FORCES --------------------
C #22####################  SUBROUTINE ACCEL  #########################
C
C     This subroutine computes total fluid acceleration using
C     Subr. 31 ASSIGN and Subr. 32 DERIV
C
      SUBROUTINE ACCEL (N)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      DIMENSION VDUM1(N1),VDUM2(N1)
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      IF (N.EQ.NSTAB) CALL CHEPAR (22,1,N1,N1R)
      CALL ASSIGN (1,VDUM1,U,2,S,2)
      CALL DERIV  (VDUM1,VDUM2,X,S)
      DO 100 J = 1,S
        DUDT(J) = -VDUM2(J)-THETA(J)-FW*V(J)*DABS(V(J))/U(2,J)
  100 CONTINUE
C
      RETURN
      END
C
C -22----------------- END OF SUBROUTINE ACCEL --------------------
C #23####################  SUBROUTINE STAT2  #########################
C
C     This subroutine computes statistical values after time-marching
C     computation
C
      SUBROUTINE STAT2
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
```

```
     +                    ISAVA,ISAVB,ISAVC
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /WAVE3/  ETA(N2),ETAIS(N2),ETARS(N2),ETATS(N2)
      COMMON /RUNP1/  NDELR
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3)
      COMMON /OVER/   OV(4)
      COMMON /COEFS/  RCOEF(3),TCOEF(3)
      COMMON /STAT/   ELSTAT(3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1)
      COMMON /STAB5/  JSNSC,NSNSC,NSNSX(N1)
      COMMON /STAB6/  SNSC,SNR(N1),SNSX(N1)
      CALL CHEPAR (23,1,N1,N1R)
      CALL CHEPAR (23,2,N2,N2R)
      CALL CHEPAR (23,3,N3,N3R)
      NDENOM = NTOP-NSTAT+1
      DENOM  = DBLE(NDENOM)
C
C ... REFLECTION AND TRANSMISSION COEFFICIENTS AND
C     MEAN SURFACE ELEVATIONS AT BOUNDARIES
C     Using Subr. 24 COEF
C
C     RCOEF(i) = reflection coefficient of the i-th kind
C     TCOEF(i) = transmission coefficient of the i-th kind
C                i=1,2,3 for monochromatic incident waves
C     Mean surface elevations:
C       ELSTAT(1): due to incident wave ETAIS (Subr. 16 SEABC)
C       ELSTAT(2): due to reflected wave ETARS (Subr. 16 SEABC)
C       ELSTAT(3): due to transmitted wave ETATS (Subr. 13 LANDBC)
C
      CALL COEF (1,DUM  ,ELSTAT(1),ETAIS,NDENOM)
      CALL COEF (2,RCOEF,ELSTAT(2),ETARS,NDENOM)
      IF (IJOB.EQ.3) CALL COEF (2,TCOEF,ELSTAT(3),ETATS,NDENOM)
C
C ... WAVE SETUP ON SLOPE COMPUTED FROM WATERLINE MOTION
C     (Subr. 13 LANDBC)
C
      IF (IJOB.LT.3) THEN
        DO 110 L = 1,NDELR
          RSTAT(1,L) = RSTAT(1,L)/DENOM
  110   CONTINUE
      ENDIF
C
C ... OVERTOPPING
C     Computed in Subr. 13 LANDBC
C     OV(2) and OV(3) are relative to the interval of N=NSTAT to
C       N=NTOP which is taken to be unity
C
      IF (IJOB.EQ.2) THEN
        DO 120 I = 1,3
          OV(I) = OV(I)/DENOM
  120   CONTINUE
      ENDIF
C
C ... MEAN HYDRODYNAMIC QUANTITIES
C
```

```
C       Mean value at node j (from Main):
C         U1STAT(j):  volume flux
C         ESTAT(1,j): surface elevation above SWL
C         VSTAT(1,j): depth-averaged velocity
C
        DO 130 J = 1,JMAX
          U1STAT(J)   = U1STAT(J)/DENOM
          ESTAT(1,J)  = ESTAT(1,J)/DENOM
          VSTAT(1,J)  = VSTAT(1,J)/DENOM
  130   CONTINUE
C
C ... CRITICAL STABILITY NUMBER, SNSC
C
C       SNSC   = critical stability number = min. of SNSX along the slope
C       NSNSC = time level N when SNSC occurs
C       JSNSC = node number where SNSC occurs
C       SNSX(j)   = local stability number (Subr. 19 STABNO)
C       NSNSX(j) = time level N when SNSX(j) occurs
C       SNR(j)     = stability number against rolling/sliding at node j
C
        IF (ISTAB.EQ.1) THEN
          DO 140 J = 1,JMAX
            IF (SNSX(J).LT.SNSC) THEN
              SNSC   = SNSX(J)
              JSNSC = J
              NSNSC = NSNSX(J)
            ENDIF
  140     CONTINUE
        ENDIF
C
        RETURN
        END
C
C -23----------------- END OF SUBROUTINE STAT2 ---------------------
C #24###################### SUBROUTINE COEF ##########################
C
C       This subroutine computes:
C       . time-averaged value of given quantity, VAL (MODE=1)
C       . reflection or transmission coefficients (three kinds) (MODE=2)
C         for monochromatic incident waves
C
        SUBROUTINE COEF (MODE,COE,AVER1,VAL,ND)
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DOUBLE PRECISION KS,KSREF,KSSEA,KSI
        DIMENSION COE(3),VAL(ND)
        COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
        SUM1 = 0.D+00
        DO 110 I = 1,ND
          SUM1 = SUM1 + VAL(I)
  110   CONTINUE
        AVER1 = SUM1/DBLE(ND)
        IF (MODE.EQ.2) THEN
          VALMAX = -1.D+03
          VALMIN =  1.D+03
```

```
        SUM2   =  0.D+00
        SUM3   =  0.D+00
        DO 120 I = 1,ND
          IF (VAL(I).GT.VALMAX) VALMAX = VAL(I)
          IF (VAL(I).LT.VALMIN) VALMIN = VAL(I)
          SUM2 = SUM2 + VAL(I)*VAL(I)
          SUM3 = SUM3 + (VAL(I)-AVER1)**2
  120   CONTINUE
        AVER2 = SUM2/DBLE(ND)
        AVER3 = SUM3/DBLE(ND)
C             Monochromatic incident wave profile is assumed to be
C             given by ETAI=(KS/2)COS[2*PI*(t+t0)] since linear
C             wave theory is normally used to estimate reflection
C             and transmission coefficients
        COE(1) = (VALMAX-VALMIN)/KS
        COE(2) = DSQRT(8.D+00*AVER2)/KS
        COE(3) = DSQRT(8.D+00*AVER3)/KS
      ENDIF
C
      RETURN
      END
C
C -24----------------- END OF SUBROUTINE COEF ---------------------
C #25#################### SUBROUTINE BALANE ########################
C
C     This subroutine checks overall energy balance
C
      SUBROUTINE BALANE
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      DIMENSION VDUM1(N1),VDUM2(N1)
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /COEFS/  RCOEF(3),TCOEF(3)
      COMMON /ENERG/  ENER(4,N1),ENERB(14)
      CALL CHEPAR (25,1,N1,N1R)
      CALL CHEPAR (25,3,N3,N3R)
C
C     All quantities involved herein are time-averaged quantities
C
C     At node j:
C       ENER(1,j) = norm. energy per unit surface area
C       ENER(2,j) = norm. energy flux per unit width
C     Norm. rate of energy dissipation, at node j:
C       ENER(3,j): due to bottom friction, per unit bottom area
C       ENER(4,j): due to wave breaking, per unit surface area
```

```
C
      DENOM = DBLE(NTOP-NSTAT+1)
      DO 120 J = 1,JMAX
        DO 110 I = 1,3
          ENER(I,J) = ENER(I,J)/DENOM
  110   CONTINUE
  120 CONTINUE
C     ----------- Use Subr. 31 ASSIGN and Subr. 32 DERIV
      CALL ASSIGN (1,VDUM1,ENER,4,JMAX,2)
      CALL DERIV  (VDUM1,VDUM2,X,JMAX)
      DO 130 J = 1,JMAX
        ENER(4,J) = -VDUM2(J)-ENER(3,J)
  130 CONTINUE
C
C     Normalized energy flux at boundaries:
C       ENERB(1): at seaward boundary
C       ENERB(2): at landward boundary
C     Normalized rate of energy dissipation in the computation domain:
C       ENERB(3): due to bottom friction
C       ENERB(4): due to wave breaking
C
      ENERB(1) = ENER(2,1)
      IF (IJOB.EQ.1) THEN
        ENERB(2) = 0.D+00
      ELSE
        ENERB(2) = ENER(2,JMAX)
      ENDIF
      DO 150 I = 3,4
        ENERB(I) = (ENER(I,1)+ENER(I,JMAX))/2.D+00
        DO 140 J = 2,JMAX-1
          ENERB(I) = ENERB(I) + ENER(I,J)
  140   CONTINUE
        ENERB(I) = ENERB(I)*X
  150 CONTINUE
C
      ENERB(5) = ENERB(1) - ENERB(2)
      ENERB(6) = ENERB(3) + ENERB(4)
      ENERB(7) = ENERB(6) - ENERB(5)
      ENERB(8) = 100.D+00*ENERB(7)/ENERB(5)
C
C     Approximate energy flux based on linear long wave:
C       ENERB(9):  due to incident wave at seaward boundary
C       ENERB(10): due to reflected wave at seaward boundary
C       ENERB(11): due to transmitted wave at landward boundary
C
      ENERB(9)  = KS*KS*DSEA2/8.D+00
      ENERB(10) = DSEA2*(KS*RCOEF(3))**2/8.D+00
      ENERB(12) = ENERB(9)-ENERB(10)
      ENERB(13) = 100.D+00*(ENERB(12)-ENERB(1))/ENERB(1)
      IF (IJOB.EQ.3) THEN
        ENERB(11) = DLAND2*(KS*TCOEF(3))**2/8.D+00
        ENERB(14) = 100.D+00*(ENERB(11)-ENERB(2))/ENERB(2)
      ENDIF
C
      RETURN
```

```
      END
C
C -25----------------- END OF SUBROUTINE BALANE --------------------
C #26##################### SUBROUTINE MATAFG ########################
C
C     This subroutine computes, for each node,
C     . the elements of the the first row of Matrix A (2x2)
C                                          --> A1(1,j) and A1(2,j)
C     . the elements of Matrix F (2x1)     --> F(1,j) and F(2,j)
C     . the first element of Matrix G (2x1) --> G1(j)
C     j=node number
C
      SUBROUTINE MATAFG (N,JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /MATRIX/ A1(2,N1),F(2,N1),G1(N1),GJR(2,N1),S1(N1),D(2,N1)
      IF (N.EQ.1) CALL CHEPAR (26,1,N1,N1R)
      DO 100 J = JBEGIN,JEND
        A1(1,J) = 2.D+00*V(J)
        A1(2,J) = U(2,J)-V(J)*V(J)
        F(1,J)  = V(J)*U(1,J) + U(2,J)*U(2,J)/2.D+00
        F(2,J)  = U(1,J)
        G1(J)   = THETA(J)*U(2,J) + FW*DABS(V(J))*V(J)
  100 CONTINUE
C
      RETURN
      END
C
C -26----------------- END OF SUBROUTINE MATAFG --------------------
C #27##################### SUBROUTINE MATGJR ########################
C
C     This subroutine computes, for each node, the elements of
C     Matrix g (2x1) --> GJR(1,j) and GJR(2,j), j=node number
C
      SUBROUTINE MATGJR (N,JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /MATRIX/ A1(2,N1),F(2,N1),G1(N1),GJR(2,N1),S1(N1),D(2,N1)
      IF (N.EQ.1) CALL CHEPAR (27,1,N1,N1R)
      DO 100 J = JBEGIN,JEND
        FG1 = F(1,J+1)-F(1,J) + X*(G1(J+1)+G1(J))/2.D+00
        FG2 = F(2,J+1)-F(2,J)
        DUM = (A1(1,J+1)+A1(1,J))*FG1 + (A1(2,J+1)+A1(2,J))*FG2
        GJR(1,J) = DUM/2.D+00
        GJR(2,J) = FG1
  100 CONTINUE
```

```
C
      RETURN
      END
C
C -27----------------- END OF SUBROUTINE MATGJR --------------------
C #28###################### SUBROUTINE MATS ########################
C
C     This subroutine computes, for each node, the first element of
C     Matrix S (2x1) --> S1(j), j=node number
C
      SUBROUTINE MATS (N,JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /MATRIX/ A1(2,N1),F(2,N1),G1(N1),GJR(2,N1),S1(N1),D(2,N1)
      IF (N.EQ.1) CALL CHEPAR (28,1,N1,N1R)
      DO 100 J = JBEGIN,JEND
        DUM1 = (V(J)*V(J)-U(2,J))*(U(2,J+1)-U(2,J-1))/TWOX
        DUM2 = V(J)*(U(1,J+1)-U(1,J-1))/TWOX
        DUM3 = THETA(J)*U(2,J)
        DUM4 = FW*DABS(V(J))*V(J)
        DUM5 = 2.D+00*FW*DABS(V(J))/U(2,J)
        EJN  = DUM5*(DUM1-DUM2-DUM3-DUM4)
        S1(J) = X*EJN - THETA(J)*(U(1,J+1)-U(1,J-1))/2.D+00
  100 CONTINUE
C
      RETURN
      END
C
C -28------------------ END OF SUBROUTINE MATS --------------------
C #29###################### SUBROUTINE MATD ########################
C
C     This subroutine computes, for each node, the elements of
C     Matrix D (2x1) --> D(1,j) and D(2,j), j=node number
C
      SUBROUTINE MATD (N,JDAM,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      DIMENSION Q(2,2,N1),UU(2,N1)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /MATRIX/ A1(2,N1),F(2,N1),G1(N1),GJR(2,N1),S1(N1),D(2,N1)
      IF (N.EQ.1) CALL CHEPAR (29,1,N1,N1R)
      DO 120 J = 1,JDAM
        CC1 = C(J+1)+C(J)
        CC2 = C(J+1)-C(J)
        VC1 = V(J+1)+V(J)+CC1
```

```
          VC2 = V(J+1)-V(J)+CC2
          VC3 = V(J+1)+V(J)-CC1
          VC4 = V(J+1)-V(J)-CC2
          PPP = (-X1*DABS(VC2)*VC3+X2*DABS(VC4)*VC1)/(2.D+00*CC1)
          QQQ = (X1*DABS(VC2)-X2*DABS(VC4))/CC1
          Q(1,1,J) = QQQ*(A1(1,J+1)+A1(1,J))/2.D+00 + PPP
          Q(1,2,J) = QQQ*(A1(2,J+1)+A1(2,J))/2.D+00
          Q(2,1,J) = QQQ
          Q(2,2,J) = PPP
          DO 110 I = 1,2
            UU(I,J) = U(I,J+1)-U(I,J)
  110     CONTINUE
  120 CONTINUE
      DO 150 I = 1,2
        DO 140 J = 2,JDAM
          D(I,J) = 0.D+00
          DO 130 L = 1,2
            D(I,J) = D(I,J) + Q(I,L,J)*UU(L,J) - Q(I,L,J-1)*UU(L,J-1)
  130     CONTINUE
          D(I,J) = TX*D(I,J)/2.D+00
  140   CONTINUE
  150 CONTINUE
      IF (JEND.GT.JDAM) THEN
        DO 170 I = 1,2
          DO 160 J = JDAM+1,JEND
            D(I,J) = 0.D+00
  160     CONTINUE
  170   CONTINUE
      ENDIF
C
      RETURN
      END
C
C -29----------------- END OF SUBROUTINE MATD --------------------
C #30###################### SUBROUTINE MATU #######################
C
C     This subroutine computes the elements of Matrix U (2x1)
C     --> U(1,j) and U(2,j) with j=node number at next time level
C
      SUBROUTINE MATU (N,JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /MATRIX/ A1(2,N1),F(2,N1),G1(N1),GJR(2,N1),S1(N1),D(2,N1)
      IF (N.EQ.1) CALL CHEPAR (30,1,N1,N1R)
      DO 100 J = JBEGIN,JEND
        DUM1 = TX*((F(1,J+1)-F(1,J-1))/2.D+00+X*G1(J))
        DUM2 = TTXX*(GJR(1,J)-GJR(1,J-1))
        DUM3 = TX*(F(2,J+1)-F(2,J-1))
        DUM4 = TTXX*(GJR(2,J)-GJR(2,J-1))
        U(1,J) = U(1,J) - DUM1 + (DUM2-TTX*S1(J))/2.D+00 + D(1,J)
        U(2,J) = U(2,J) - (DUM3-DUM4)/2.D+00 + D(2,J)
```

```
  100 CONTINUE
C
      RETURN
      END
C
C -30------------------ END OF SUBROUTINE MATU ---------------------
C #31##################### SUBROUTINE ASSIGN ########################
C
C     This subroutine changes notations from matrix to vector or
C     from vector to matrix
C
      SUBROUTINE ASSIGN (MODE,VAL1,VAL2,ND1,ND2,NROW)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VAL1(ND2),VAL2(ND1,ND2)
      IF (MODE.EQ.1) THEN
        DO 110 J = 1,ND2
          VAL1(J) = VAL2(NROW,J)
  110   CONTINUE
      ELSE
        DO 120 J = 1,ND2
          VAL2(NROW,J) = VAL1(J)
  120   CONTINUE
      ENDIF
C
      RETURN
      END
C
C -31----------------- END OF SUBROUTINE ASSIGN --------------------
C #32###################### SUBROUTINE DERIV ########################
C
C     This subroutine computes the first derivative, DER, of given
C     quantity, FUN, with respect to given variable, VAR, for
C     J=1,2,...,ND
C
      SUBROUTINE DERIV (FUN,DER,VAR,ND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION FUN(ND),DER(ND)
      VAR2 = 2.D+00*VAR
      DER(1)  = (FUN(2)-FUN(1))/VAR
      DER(ND) = (FUN(ND)-FUN(ND-1))/VAR
      DO 100 J = 2,ND-1
        DER(J) = (FUN(J+1)-FUN(J-1))/VAR2
  100 CONTINUE
C
      RETURN
      END
C
C -32------------------ END OF SUBROUTINE DERIV --------------------
C #33####################### 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=500,N2=30000,N3=3,N4=100,N5=25)
      DOUBLE PRECISION KCNO,MCNO,KC2
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      CHARACTER*7 UL
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /GRID/   T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WL0P
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /WAVE4/  ETAMAX,ETAMIN
      COMMON /WAVE5/  KCNO,ECNO,MCNO,KC2
      COMMON /BOT1/   DSEAP,DLANDP,FWP
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW,TSLOPS,WTOT
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /BOT4/   NBSEG
      COMMON /BOT5/   WBSEG(N4),TBSLOP(N4),XBSEG(N4),ZBSEG(N4)
      COMMON /STAB1/  C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      CALL CHEPAR (33,1,N1,N1R)
      CALL CHEPAR (33,4,N4,N4R)
      ISTOP = 0
C
C ... SYSTEM OF UNITS
C
      IF (ISYST.EQ.1) THEN
        UL = ' meters'
      ELSE
        UL = ' feet  '
      ENDIF
C
C ... NUMERICAL STABILITY INDICATOR, ALPHAS
C
      EPSI   = DMAX1(X1,X2)
      DUM1   = 1.D+00 + EPSI*EPSI/4.D+00
      DUM2   = DSQRT(DUM1) - EPSI/2.D+00
      ALPHAS = DUM2*X*DBLE(NONE)/(1.D+00+DSQRT(DSEA))
C
C ... WAVE CONDITION
C
      WRITE (28,2811)
      IF (IWAVE.EQ.1) THEN
        IF (URPRE.LT.26) THEN
          WRITE (28,2812)
        ELSE
          WRITE (28,2813) KC2,ECNO,KCNO
        ENDIF
      ELSEIF (IWAVE.EQ.2) THEN
        WRITE (28,2814)
```

```
      ELSE
        WRITE (28,2815)
      ENDIF
      WRITE (28,2816) ETAMAX,ETAMIN
      WRITE (28,2817) TP,HREFP,UL,DSEAP,UL,KSREF,KSSEA,KS
      WRITE (28,2818) DSEA,WL,SIGMA,UR,KSI
      IF (IJOB.EQ.3)  WRITE (28,2819) DLANDP,UL
 2811 FORMAT (/'WAVE CONDITION')
 2812 FORMAT (/'Stokes II Incident Wave at Seaward Boundary')
 2813 FORMAT (/'Cnoidal Incident Wave at Seaward Boundary'/
     +        '1-m = ',D20.9/
     +        'E   = ',D20.9/
     +        'K   = ',D20.9)
 2814 FORMAT (/'Incident Wave at Seaward Boundary Given as Input')
 2815 FORMAT (/'Total Wave at Seaward Boundary Given as Input')
 2816 FORMAT ('Norm. Maximum Surface Elev.      = ',F12.6/
     +        'Norm. Minimum Surface Elev.      = ',F12.6/)
 2817 FORMAT ('Reference Wave Period            = ',F12.6,' sec.'/
     +        'Reference Wave Height            = ',F12.6,A7/
     +        'Depth at Seaward Boundary        = ',F12.6,A7/
     +        'Shoal. Coef. at Reference  Ks1 = ',F9.3/
     +        '              at Seaw. Bdr. Ks2 = ',F9.3/
     +        '              Ks = Ks2/Ks1      = ',F9.3)
 2818 FORMAT ('Norm. Depth at Seaw. Bdr.        = ',F9.3/
     +        'Normalized Wave Length           = ',F9.3/
     +        '"Sigma"                          = ',F9.3/
     +        'Ursell Number                    = ',F9.3/
     +        'Surf Similarity Parameter        = ',F9.3)
 2819 FORMAT ('Depth at Landward Boundary       = ',F12.6,A7)
C
C ... STRUCTURE PROPERTIES
C
      WRITE (28,2821) FWP,FW,WTOT,NBSEG
      IF (IBOT.EQ.1) THEN
        WRITE (28,2822) UL
        WRITE (28,2824) (K,WBSEG(K),TBSLOP(K),K=1,NBSEG)
      ELSE
        WRITE (28,2823) UL,UL
        WRITE (28,2824) (K,XBSEG(K),ZBSEG(K),K=1,NBSEG+1)
      ENDIF
 2821 FORMAT (/'SLOPE PROPERTIES'//
     +        'Friction Factor                = ',F12.6/
     +        'Norm. Friction Factor          = ',F12.6/
     +        'Norm. Horiz. Length of'/
     +        '      Computation Domain       = ',F12.6/
     +        'Number of Segments             = ',I8)
 2822 FORMAT ('--------------------------------'/
     +        ' SEGMENT    WBSEG(I)   TBSLOP(I)'/
     +        '        I        ',A7/
     +        '--------------------------------')
 2823 FORMAT ('--------------------------------'/
     +        ' SEGMENT    XBSEG(I)    ZBSEG(I)'/
     +        '        I        ',A7,'     ',A7/
     +        '--------------------------------')
 2824 FORMAT (I8,2F12.6)
```

```
C
C ... COMPUTATION PARAMETERS
C
      WRITE (28,2841) X,T,DELTA,X1,X2,ALPHAS
      WRITE (28,2842) NTOP,NONE,JE
      IF (IJOB.LT.3)  WRITE (28,2843) S
      WRITE (28,2844) NJUM1
      IF (ISTAB.GT.0) WRITE (28,2845) NJUM2
 2841 FORMAT (/'COMPUTATION PARAMETERS'//
     +  'Normalized Delta x    = ',D14.6/
     +  'Normalized Delta t    = ',D14.6/
     +  'Normalized DELTA      = ',E14.6/
     +  'Damping Coeff.     x1 = ',F9.3/
     +  '                   x2 = ',F9.3/
     +  'Num. Stab. Indicator  = ',F9.3)
 2842 FORMAT (
     +  'Total Number of Time Steps        NTOP = ',I8/
     +  'Number of Time Steps in 1 Wave Period'/
     +  '                                  NONE = ',I8/
     +  'Total Number of Spatial Nodes       JE = ',I8)
 2843 FORMAT (
     +  'Number of Nodes Along Bottom Below SWL'/
     +  '                                     S = ',I8)
 2844 FORMAT (
     +  'Storing Temporal Variations at Every'/
     +  '                                NJUM1 = ',I8,' Time Steps')
 2845 FORMAT (
     +  'Armor Stability Number Computed'/
     +  '                      at Every  NJUM2 = ',I8,' Time Steps')
C
C ... PARAMETERS FOR ARMOR STABILITY AND MOVEMENT
C
      IF (ISTAB.GT.0) WRITE (28,2851) TANPHI,SG,C2,C3,CD,CL,CM
      IF (ISTAB.EQ.1) WRITE (28,2852) AMAX,AMIN
      IF (ISTAB.EQ.2) WRITE (28,2853) DAP,UL
 2851 FORMAT (/'PARAMETERS FOR ARMOR STABILITY AND MOVEMENT'//
     +        'Armor Friction Factor       = ',F9.3/
     +        'Specific Gravity            = ',F9.3/
     +        'Area Coefficient        C2  = ',F9.3/
     +        'Volume Coefficient      C3  = ',F9.3/
     +        'Drag Coefficient        CD  = ',F9.3/
     +        'Lift Coefficient        CL  = ',F9.3/
     +        'Inertia Coefficient     CM  = ',F9.3)
 2852 FORMAT ('Norm. Upper and Lower Bounds of du/dt'/
     +        '                      AMAX  = ',F9.3/
     +        '                      AMIN  = ',F9.3)
 2853 FORMAT ('Armor Diameter              = ',F12.6,A7)
C
C ... NORMALIZED STRUCTURE GEOMETRY
C
C     File 22 = 'OSPACE'
C     (XB(j),ZB(j)) = normalized coordinates of the structure
C                     at node j
C     ZB negative below SWL
C
```

```
      WRITE (22,2210) JE
      WRITE (22,2220) (XB(J),ZB(J),J=1,JE)
 2210 FORMAT (I8)
 2220 FORMAT (6D12.4)
C
C ... SOME CHECKINGS
C
C     ---- Numerical stability criterion requires ALPHAS > about 1
C
      IF (ALPHAS.LE.1.) THEN
        WRITE  (*,2910) ALPHAS
        WRITE (29,2910) ALPHAS
        ISTOP = ISTOP+1
      ENDIF
 2910 FORMAT (/' From Subr. 33 DOC1'/' Stability Indicator =',F9.3/
     +         ' May cause numerical instability. Increase NONE')
C
C     --- Temporal variations are stored at every NJUM1 time steps.
C         Stability number SNR is computed at every NJUM2 time steps
C         (SNR is stability number against rolling/sliding).
C         For plotting the results, the values of NONE/NJUM1 and
C         NONE/NJUM2 should be integers.
C
      VAL1 = DBLE(NONE)/DBLE(NJUM1)
      RES1 = DMOD(VAL1,1.D+00)
      IF (RES1.NE.0.D+00) THEN
        WRITE  (*,2920) NONE,NJUM1,VAL1
        WRITE (29,2920) NONE,NJUM1,VAL1
        ISTOP = ISTOP+1
      ENDIF
      IF (ISTAB.EQ.1) THEN
        VAL2 = DBLE(NONE)/DBLE(NJUM2)
        RES2 = DMOD(VAL2,1.D+00)
        IF (RES2.NE.0.D+00) THEN
          WRITE  (*,2930) NONE,NJUM2,VAL2
          WRITE (29,2930) NONE,NJUM2,VAL2
          ISTOP = ISTOP+1
        ENDIF
      ENDIF
 2920 FORMAT (/' From Subr. 33 DOC1'/' NONE =',I8/' NJUM1 =',I8/
     +         ' NONE/NJUM1 =',F12.3,', not an integer'/
     +         ' Change NJUM1')
 2930 FORMAT (/' From Subr. 33 DOC1'/' NONE =',I8/' NJUM2 =',I8/
     +         ' NONE/NJUM2 =',F12.3,', not an integer'/
     +         ' Change NJUM2')
C
      IF (ISTOP.GT.0) STOP
C
C ... CONDITIONAL STOP BEFORE TIME-MARCHING COMPUTATION
C
      WRITE (*,6010) ALPHAS
      WRITE (*,6020)
      READ  (*,*) ISTOP
      IF (ISTOP.EQ.1)  STOP
 6010 FORMAT (' Numerical stability indicator =',F7.2)
```

```
 6020 FORMAT (' Time-marching computation is about to begin'/
     +         ' 1 = stop here, else = proceed')
C
      RETURN
      END
C
C -33------------------ END OF SUBROUTINE DOC1 ---------------------
C #34####################### SUBROUTINE DOC2 ########################
C
C     This subroutine stores computed results at designated time
C     levels during time-marching computation
C
      SUBROUTINE DOC2 (ICALL,N,ETAR,ETAT)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=500,N2=30000,N3=3,N4=100,N5=25)
      CHARACTER*20 FNAME1,FNAME2
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /IDREQ/  IREQ,IELEV,IV,IDUDT,ISNR,NNREQ,NREQ(N5)
      COMMON /TLEVEL/ NTOP,NONE,NJUM1,NJUM2,NSAVA,NSTAB,NSTAT,NTIMES
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /WAVE3/  ETA(N2),ETAIS(N2),ETARS(N2),ETATS(N2)
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /RUNP1/  NDELR
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3)
      COMMON /STAB5/  JSNSC,NSNSC,NSNSX(N1)
      COMMON /STAB6/  SNSC,SNR(N1),SNSX(N1)
      COMMON /STAB7/  NMOVE,NSTOP,
     +                ISTATE(N1),NODIN(N1),NODFI(N1),NDIS(N1)
      COMMON /STAB8/  VA(N1),XAA(N1),XA(N1)
      COMMON /FILES/  NNOD1,NNOD2,NODNO1(N5),NODNO2(N5),
     +                FNAME1(N5),FNAME2(N5)
      DATA ZERO /0.D+00/
C
      IF (ICALL.EQ.0) THEN
C
C ..... CHECKING PARAMETERs
C
        CALL CHEPAR (34,1,N1,N1R)
        CALL CHEPAR (34,2,N2,N2R)
        CALL CHEPAR (34,3,N3,N3R)
        CALL CHEPAR (34,5,N5,N5R)
C
      ELSEIF (ICALL.EQ.1) THEN
C
C ..... STORING "A"
C       "A" = spatial variations of hydrodynamic quantities
C
C       File 22 = 'OSPACE'
C       N = current time level
C       S = waterline node (IJOB<3) or landward-end node (IJOB=3)
C       At node j:
```

```
C           ELEV(j) = surface elevation above SWL
C           V(j)    = depth-averaged velocity
C
        WRITE (22,2210) N,S
        WRITE (22,2220) (ELEV(J),V(J),J=1,S)
C
      ELSEIF (ICALL.EQ.2) THEN
C
C ..... STORING "B"
C       "B" = temporal variations of total water depth at specified
C             nodes
C
        IF (ISAVB.EQ.1) THEN
          DO 110 I = 1,NNOD1
            NUNIT = 49+I
            J     = NODNO1(I)
            WRITE (NUNIT,5010) N,U(2,J)
  110     CONTINUE
        ENDIF
C
C ..... STORING "C"
C       "C" = temporal variations of displacement of armor units
C             from specified initial nodal locations
C
        IF (ISAVC.EQ.1) THEN
          DO 210 I = 1,NNOD2
            NUNIT = 74+I
            J = NODNO2(I)
            IF (ISTATE(J).EQ.0) THEN
              WRITE (NUNIT,7510) N,ZERO
            ELSE
              WRITE (NUNIT,7510) N,XA(J)
            ENDIF
  210     CONTINUE
        ENDIF
C
C ..... STORING VALUES AT LANDWARD-END NODE
C
C       File 31 = 'ORUNUP'
C       File 32 = 'OOVER'
C       File 33 = 'OTRANS'
C       JE = landward-end node
C       N  = current time level
C       S  = waterline node
C       RUNUPS = free surface elevation where the water depth equals
C                DELTAR
C       DELTAR = water depth associated with visual or measured
C                waterline
C       NDELR  = number of DELTARs
C       ETAT   = surface elevation due to transmitted wave at
C                landward boundary
C       At node j:
C         U(1,j) = volume flux
C         U(2,j) = total water depth
C         V(j)   = depth-averaged velocity
```

```
C         C(j)    = critical velocity                                      IBR33560
C                                                                          IBR33570
        IF (IJOB.GT.1) C(JE)=DSQRT(U(2,JE))                                IBR33580
        IF (IJOB.LT.3) THEN                                                IBR33590
          WRITE (31,3110) N,S                                              IBR33600
          WRITE (31,3120) (RUNUPS(L),L=1,NDELR)                            IBR33610
          IF (IJOB.EQ.2) WRITE (32,3210) N,U(1,JE),U(2,JE),V(JE),C(JE)     IBR33620
        ELSE                                                               IBR33630
          WRITE (33,3310) N,U(1,JE),V(JE),C(JE),ETAT                       IBR33640
        ENDIF                                                              IBR33650
C                                                                          IBR33660
C ..... STORING VALUES AT SEAWARD BOUNDARY                                 IBR33670
C                                                                          IBR33680
C       File 21 = 'OSEAWAV'                                                IBR33690
C       N = current time level                                             IBR33700
C       Surface elevations at seaward boundary:                            IBR33710
C       . ETAI --> due to incident wave                                    IBR33720
C       . ETAR --> due to reflected wave                                   IBR33730
C       . ETATOT = ETAI+ETAR                                               IBR33740
C       At node j:                                                         IBR33750
C         V(j)    = depth-averaged velocity                                IBR33760
C         U(1,j) = volume flux                                             IBR33770
C                                                                          IBR33780
        IF (IWAVE.EQ.1) THEN                                               IBR33790
          NWAVE  = MOD(N,NONE) + 1                                         IBR33800
          ETAI   = ETA(NWAVE)                                              IBR33810
          ETATOT = ETAI+ETAR                                               IBR33820
        ELSEIF (IWAVE.EQ.2) THEN                                           IBR33830
          ETAI   = ETA(N)                                                  IBR33840
          ETATOT = ETAI+ETAR                                               IBR33850
        ELSE                                                               IBR33860
          ETAI   = ETA(N)-ETAR                                             IBR33870
          ETATOT = ETA(N)                                                  IBR33880
        ENDIF                                                              IBR33890
        WRITE (21,2110) N,ETAI,ETAR,ETATOT,V(1),U(1,1)                     IBR33900
C                                                                          IBR33910
      ELSE                                                                 IBR33920
C                                                                          IBR33930
C ..... SPECIAL STORING IF ICALL=3                                         IBR33940
C                                                                          IBR33950
C       File 40 = 'OREQ'                                                   IBR33960
C       S = waterline node (IJOB<3) or landward-end node (IJOB=3)          IBR33970
C       At node j for specified time level N:                              IBR33980
C         ELEV(j) = surface elevation above SWL                            IBR33990
C         V(j)    = depth-averaged velocity                                IBR34000
C         DUDT(j) = total fluid acceleration                               IBR34010
C         SNR(j)  = stability number against rolling/sliding               IBR34020
C                                                                          IBR34030
        WRITE (40,4010) N,S                                                IBR34040
        IF (IELEV.EQ.1) WRITE (40,4020) (ELEV(J),J=1,S)                    IBR34050
        IF (IV.EQ.1)    WRITE (40,4020) (V(J),J=1,S)                       IBR34060
        IF (IDUDT.EQ.1) WRITE (40,4020) (DUDT(J),J=1,S)                    IBR34070
        IF (ISNR.EQ.1)  WRITE (40,4020) (SNR(J),J=1,S)                     IBR34080
C                                                                          IBR34090
      ENDIF                                                                IBR34100
```

```
C
C ... FORMATS
C
 2110 FORMAT (I8,5D12.4)
 2210 FORMAT (2I8)
 2220 FORMAT (6D12.4)
 3110 FORMAT (2I8)
 3120 FORMAT (6D12.4)
 3210 FORMAT (I8,5D12.4)
 3310 FORMAT (I8,5D12.4)
 4010 FORMAT (2I8)
 4020 FORMAT (6D12.4)
 5010 FORMAT (I8,D12.4)
 7510 FORMAT (I8,D12.4)
C
      RETURN
      END
C
C -34------------------  END OF SUBROUTINE DOC2  ---------------------
C #35######################  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=500,N2=30000,N3=3,N4=100,N5=25)
      CHARACTER*7 UL
      INTEGER S
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV,DELTA,X1,X2
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /NODES/  S,JE,JE1,JSTAB,JMAX
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /COEFS/  RCOEF(3),TCOEF(3)
      COMMON /STAT/   ELSTAT(3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1)
      COMMON /RUNP1/  NDELR
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3)
      COMMON /OVER/   OV(4)
      COMMON /STAB5/  JSNSC,NSNSC,NSNSX(N1)
      COMMON /STAB6/  SNSC,SNR(N1),SNSX(N1)
      COMMON /STAB7/  NMOVE,NSTOP,
     +                ISTATE(N1),NODIN(N1),NODFI(N1),NDIS(N1)
      COMMON /STAB8/  VA(N1),XAA(N1),XA(N1)
      COMMON /ENERG/  ENER(4,N1),ENERB(14)
      CALL CHEPAR (35,1,N1,N1R)
      CALL CHEPAR (35,3,N3,N3R)
C
C ... SYSTEM OF UNITS
C
      IF (ISYST.EQ.1) THEN
        UL = '    [mm]'
      ELSE
```

```
          UL = ' [inch]'
        ENDIF
C
C ... REFLECTION COEFFICIENTS
C
        WRITE (28,2811) (RCOEF(I),I=1,3)
 2811 FORMAT (/'REFLECTION COEFFICIENTS'//
     +        'r1 = ',F9.3/
     +        'r2 = ',F9.3/
     +        'r3 = ',F9.3)
C
C ... RUNUP, RUNDOWN, SETUP
C
        IF (IJOB.LT.3) THEN
          WRITE (28,2821) JMAX
          WRITE (28,2822) UL
          DO 110 L = 1,NDELR
            WRITE(28,2823)L,DELRP(L),RSTAT(2,L),RSTAT(3,L),RSTAT(1,L)
  110     CONTINUE
        ENDIF
 2821 FORMAT (/'RUNUP, RUNDOWN, SETUP'//
     +'Largest Node Number Reached by Computational Waterline'/
     +'                                          JMAX = ',I8)
 2822 FORMAT (
     +'-------------------------------------------------------'/
     +'       I   DELTAR(I)     RUNUP(I)    RUNDOWN(I)    SETUP(I)'/
     +'                ',A7,'         R            Rd          Zr'/
     +'-------------------------------------------------------')
 2823 FORMAT (I8,1X,F9.3,3(3X,F9.3))
C
C ... OVERTOPPING
C
        IF (IJOB.EQ.2) THEN
          WRITE (28,2831) OV(1),U1STAT(1),OV(4),OV(2),OV(3)
        ENDIF
 2831 FORMAT (/'OVERTOPPING'//
     +        'Norm. Avg. Overtopping Rate     = ',D14.6/
     +        'Norm. Avg. Flow at Seaw. Bdr.   = ',D14.6/
     +        'Norm. Max. Overtopping Rate     = ',D14.6/
     +'Max. Rate Occurs at     ',F8.6,' Within Interval [NSTAT,NTOP]'/
     +'Overtopping Duration = ',F8.6,' Within Interval [NSTAT,NTOP]'/
     +'The last two quantities are relative to the specified'/
     +'  interval taken as unity')
C
C ... QUANTITIES FOR ARMOR STABILITY AND MOVEMENT
C
C     File 41 = 'OSTAB1'
C     File 42 = 'OSTAB2'
C     (XB(j),ZB(j)) = normalized coordinates of the structure at
C                     node j
C     ISTAB=1:
C       SNSX(j) = local stability number = minimum of SNR at a node j
C       SNR(j)  = stability number against rolling/sliding at node j
C     ISTAB=2:
C       ISTATE(j) indicates the state of armor unit initially located
```

```
C          at node j: 0=stationary, 1=moving, 2=stopped
C        For moving/stopped armor unit number j:
C          NODIN(j) = its initial location (i.e., node number)
C          NODFI(j) = node closest to its final location
C          NDIS(j)  = time level N when it started moving
C          XA(j)    = displacement from its initial location,
C                       normalized by DAP
C
      IF (ISTAB.EQ.1) THEN
        WRITE (28,2841) SNSC,JSNSC,NSNSC
        WRITE (41,4110) JMAX
        WRITE (41,4120) (XB(J),ZB(J),SNSX(J),J=1,JMAX)
      ELSEIF (ISTAB.EQ.2) THEN
        WRITE (28,2842) NMOVE,NSTOP
        WRITE (42,4210) NMOVE
        DO 120 J = 1,JMAX
          IF (ISTATE(J).GE.1) WRITE (42,4220)
     +       NODIN(J),NODFI(J),NDIS(J),ISTATE(J),XB(J),ZB(J),XA(J)
  120   CONTINUE
      ENDIF
 2841 FORMAT (/'STABILITY NUMBER'//
     +          'Critical Stability Number Nsc = ',F9.3/
     +          'At Node Number                J = ',I9/
     +          'At Time Level                 N = ',I9)
 2842 FORMAT (/'ARMOR UNITS MOVEMENT'//
     +          'Number of Units Moved         = ',I8/
     +          'Number of Units Stopped       = ',I8)
 4110 FORMAT (I8)
 4120 FORMAT (6D12.4)
 4210 FORMAT (I8)
 4220 FORMAT (3I8,I3,3D12.4)
C
C ... WAVE SET-DOWN OR SETUP
C
      IF (IJOB.EQ.3) THEN
        DELMWL = ELSTAT(3) - ELSTAT(2)
        WRITE (28,2851) (ELSTAT(I),I=1,3),DELMWL
      ELSE
        WRITE (28,2852) (ELSTAT(I),I=1,2)
      ENDIF
 2851 FORMAT (/'WAVE SET-DOWN OR SETUP'//
     +          'Average value of ETAI = ',F12.6/
     +          '                 ETAR = ',F12.6/
     +          '                 ETAT = ',F12.6/
     +          'MWL Difference        = ',F12.6)
 2852 FORMAT (/'WAVE SET-DOWN OR SETUP'//
     +          'Average value of ETAI = ',F12.6/
     +          '                 ETAR = ',F12.6)
C
C ... TRANSMISSION
C
      IF (IJOB.EQ.3) THEN
        QAVER = .5*(U1STAT(1)+U1STAT(JE))
        WRITE (28,2861) (TCOEF(I),I=1,3)
        WRITE (28,2862) U1STAT(1),U1STAT(JE),QAVER
```

```
      ENDIF
 2861 FORMAT (/'TRANSMISSION'//
     +        'Transmission Coefficient T1 = ',F9.3/
     +        '                         T2 = ',F9.3/
     +        '                         T3 = ',F9.3)
 2862 FORMAT ('Norm. Avg. Flow at Seaw. Bdr.  = ',F12.6/
     +        'Norm. Avg. Flow at Landw. Bdr. = ',F12.6/
     +        'Average of the Above Two       = ',F12.6)
C
C ... STATISTICS OF HYDRODYNAMIC QUANTITIES
C
C     File 23 = 'OSTAT'
C       JMAX = the largest node number reached by computational
C              waterline
C       At node j:
C         ELEV(j)    = surface elevation above SWL
C         V(j)       = depth-averaged velocity
C         U1STAT(j) = mean volume flux
C       Mean, maximum, and minimum at node j:
C         ESTAT(1,j),ESTAT(2,j),ESTAT(3,j): for ELEV(j)
C         VSTAT(1,j),VSTAT(2,j),VSTAT(3,j): for V(j)
C
      WRITE (23,2310) JMAX
      WRITE (23,2320) (U1STAT(J),J=1,JMAX)
      DO 130 I = 1,3
        WRITE (23,2320) (ESTAT(I,J),J=1,JMAX)
        WRITE (23,2320) (VSTAT(I,J),J=1,JMAX)
  130 CONTINUE
 2310 FORMAT (I8)
 2320 FORMAT (6D12.4)
C
C ... QUANTITIES FOR TIME-AVERAGED ENERGY BALANCE
C
C     File 35 = 'OENERG'
C     At node j:
C       ENER(1,j) = norm. energy per unit surface area
C       ENER(2,j) = norm. energy flux per unit width
C     Normalized rate of energy dissipation at node j:
C       ENER(3,j): due to bottom friction, per unit bottom area
C       ENER(4,j): due to wave breaking, per unit surface area
C
      IF (IENERG.EQ.1) THEN
        WRITE (28,2871)
        WRITE (28,2872) (ENERB(I),I=1,10)
        IF (IJOB.EQ.3)  WRITE (28,2873) ENERB(11)
        WRITE (28,2874) (ENERB(I),I=12,13)
        IF (IJOB.EQ.3)  WRITE (28,2875) ENERB(14)
        WRITE (35,3510) JMAX
        DO 140 I = 1,4
          WRITE (35,3520) (ENER(I,J),J=1,JMAX)
  140   CONTINUE
      ENDIF
 2871 FORMAT (/'TIME-AVERAGED ENERGY BALANCE'//
     +        'Normalized Energy Flux:')
 2872 FORMAT ('. at Seaw. Boundary   A =',D14.6/
```

```
     +          '. at Landw. Boundary  B =',D14.6/
     +          'Normalized Rate of Energy Dissipation'/
     +          'in the Computation Domain, Due to:'/
     +          '. bottom friction      C =',D14.6/
     +          '. wave breaking        D =',D14.6/
     +          'Calculation 1:'/
     +          '                E = A-B =',D14.6/
     +          '                F = C+D =',D14.6/
     +          '  Must G=0, but G = F-E =',D14.6/
     +          '  % error  100G/E =',F14.2/
     +          'Approximate Energy Flux, Based on'/
     +          'Linear Long Wave, Due to:'/
     +          '. incident wave at seaw. boundary      P =',D14.6/
     +          '. reflected wave at seaw. boundary     Q =',D14.6)
 2873 FORMAT ('. transmitted wave at landw. bndry.   R =',D14.6)
 2874 FORMAT ('Calculation 2:'/
     +          'Net Energy Flux at Seaw. Bndry. S = P-Q =',D14.6/
     +          '% Error at Seaward Boundary  100(S-A)/A =',F14.2)
 2875 FORMAT ('% Error at Landward Boundary 100(R-B)/B =',F14.2)
 3510 FORMAT (I8)
 3520 FORMAT (5D15.6)
C
      RETURN
      END
C
C -35----------------- END OF SUBROUTINE DOC3 --------------------
C #36#################### 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
      SUBROUTINE CHEPAR (ICALL,NW,NCHEK,NREF)
C
      CHARACTER*2 WHICH(5)
      CHARACTER*6 SUBR(38)
      DATA WHICH /'N1','N2','N3','N4','N5'/
      DATA SUBR  /'OPENER','INPUT1','INPUT2','BOTTOM','PARAM ',
     1           'INIT1 ','INIT2 ','INWAV ','FINDM ','CEL   ',
     2           'SNCNDN','MARCH ','LANDBC','RUNUP ','OVERT ',
     3           'SEABC ','ENERGY','STAT1 ','STABNO','MOVE  ',
     4           'FORCES','ACCEL ','STAT2' ,'COEF  ','BALANE',
     5           'MATAFG','MATGJR','MATS  ','MATD  ','MATU  ',
     6           'ASSIGN','DERIV ','DOC1  ','DOC2  ','DOC3  ',
     7           'CHEPAR','CHEOPT','STOPP '/
      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 -36----------------  END OF SUBROUTINE CHEPAR  --------------------
C #37#####################  SUBROUTINE CHEOPT  ######################
C
C     This subroutine checks user's options
C
      SUBROUTINE CHEOPT (ICALL,INDIC,ITEM,ILOW,IUP)
C
      CHARACTER*2 WHICH(6)
      CHARACTER*6 OPTI(21)
      DATA WHICH /'N5','N5','N5','N1','N3','N4'/
      DATA OPTI  //'IJOB  ','ISTAB ','ISYST ','IBOT  ','INONCT',
     1             'IENERG','IWAVE ','ISAVA ','ISAVB ','ISAVC ',
     2             'IREQ  ','IELEV ','IV    ','IDUDT ','ISNR  ',
     3             'NNOD1 ','NNOD2 ','NNREQ ','S     ','NDELR ',
     4             'NBSEG '/
      IF (ICALL.LE.15) 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-15
          WRITE  (*,2920) OPTI(ICALL),ITEM,OPTI(ICALL),IUP,WHICH(I)
          WRITE (29,2920) OPTI(ICALL),ITEM,OPTI(ICALL),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 range of [1,',I8,']'/
     +        ' Change PARAMETER ',A2,' if necessary')
C
      RETURN
      END
C
C -37----------------  END OF SUBROUTINE CHEOPT  --------------------
C #38#####################  SUBROUTINE STOPP  #######################
C
C     This subroutine executes a programmed stop
C
      SUBROUTINE STOPP (IBEGIN,IEND)
C
      CHARACTER*55 MSG(8)
      DATA MSG /
     1 ' Special storing requested,',
     2 ' but pertinent identifiers not specified correctly.',
     3 ' Check identifiers IREQ,IELEV,IV,IDUDT,ISNR.',
     4 ' Need more data.',
     5 ' SWL is always above the structure.',
```

```
     6 ' RUNUP/OVERTOPPING computation can not be performed.',         IBR37410
     7 ' Part of the structure is above SWL.',                         IBR37420
     8 ' TRANSMISSION computation can not be performed.'/              IBR37430
      DO 100 I = IBEGIN,IEND                                           IBR37440
        WRITE  (*,2910) MSG(I)                                         IBR37450
        WRITE (29,2910) MSG(I)                                         IBR37460
  100 CONTINUE                                                         IBR37470
      WRITE  (*,2920)                                                  IBR37480
      WRITE (29,2920)                                                  IBR37490
 2910 FORMAT (A55)                                                     IBR37500
 2920 FORMAT (' Programmed Stop.')                                     IBR37510
C                                                                      IBR37520
      STOP                                                             IBR37530
      END                                                              IBR37540
C                                                                      IBR37550
C -38-----------------  END OF SUBROUTINE STOPP  --------------------  IBR37560
```

APPENDIX B: EXAMPLE OF WAVE RUNUP AND ARMOR STABILITY

The first example of the input and output of IBREAK is the Test No. 12 of the large-scale riprap tests conducted by Ahrens (1975).

Table B-1 lists the input data written from the file with its unit number = 11 and its file name FINP1 specified by a user. It is noted that IBREAK requires only the value of KS=KSSEA/KSREF as explained in the common /WAVE2/. For this example, the representative wave height H_r' = HREFP is taken as the wave height H' measured at the toe of the 1:2.5 riprap slope. For the case of $H_r' = H'$, KSSEA=KSREF and KS=1. As a result, KSSEA=KSREF=1.0 could also be used as input instead of KSSEA=KSREF=1.0614 calculated using an available table for cnoidal wave shoaling (Svendsen and Brink-Kjaer, 1972).

Table B-2 lists the warning messages written from the file with its unit number = 29 and its file name = OMSG. These messages tend to appear during the transient period starting from the initial conditions of no wave action in the computation domain until the establishment of the periodic wave motion in the computation domain.

Table B-3 lists the concise output written from the file with its unit number = 28 and its file name = ODOC. The concise output is used to check the accuracy of the input data as well as to obtain important design parameters regarding wave reflection, runup, run-down, setup and armor stability.

Fig. B-1 shows the dimensional structure geometry specified as input. Since IJOB=1 and no wave overtopping is allowed, the width of the 1:2.5 slope must be chosen so that wave runup will not reach the landward edge of the slope.

Fig. B-2 shows the variations of η_i = ETAI, η_r = ETAR and $\eta_{tot} = (\eta_i + \eta_r)$ = ETATOT as a function of the normalized time t = N/NONE plotted from the file with its unit number = 21 and its file name = OSEAWAV. It is noted that the incident wave profile $\eta_i(t)$ and the reflected wave profile $\eta_r(t)$ are approximately 180° out of phase for this example.

Fig. B-3 shows the variations of the normalized waterline elevation Z_r = RUNUPS(i) with i=1,2 and 3 corresponding to the physical depth $\delta_r' = 0.157$, 0.787 and 1.575 inches, respectively, as a function of the normalized time t=N/NONE plotted from the file with its unit number = 31 and its file name = ORUNUP. Wave runup is not sensitive to δ_r' but wave run-down is very sensitive to δ_r' as shown in Fig. B-3.

Fig. B-4 shows the variations of the normalized mean water level $\bar{\eta}$ = ESTAT(1,j), the normalized crest elevation η_{max} = ESTAT(2,j) and the normalized trough elevation η_{min} = ESTAT(3,j) as a function of the normalized horizontal coordinate $x = (j-1)\Delta x$ plotted from the file with its unit number = 23 and its file name = OSTAT. Fig. B-4 also shows the normalized structure geometry described by the nodal locations XB(j) and ZB(j) which are stored in the file with its unit number = 22 and its file name = OSPACE.

Fig. B-5 shows the variations of the time-averaged fluid velocity $\bar{u}$ = VSTAT(1,j), the maximum fluid velocity u_{max} = VSTAT(2,j) and the minimum fluid velocity u_{min} = VSTAT(3,j) as a function of the normalized horizontal coordinate $x = (j-1)\Delta x$ plotted from the file with its unit number = 23 and its file name = OSPACE. It is noted that $\bar{u}$ is negative as explained by Kobayashi et al. (1989).

Fig. B-6 shows the variation of the time-averaged volume flux per unit width $\bar{m}$ = U1STAT(j), as a function of the normalized horizontal coordinate $x = (j-1)\Delta x$ plotted from the file with its unit number = 23 and its file name = OSPACE. It is noted that the time-averaged continuity equation requires $\bar{m}=0$

(Kobayashi et al., 1989) while the computed values of $\overline{m}$ are on the order of 10^{-3} or less.

Fig. B-7 shows the variations of the time-averaged wave energy per unit horizontal area, $\overline{E}$ = ENER(1,j), the time-averaged energy flux per unit width, $\overline{E_F}$ = ENER(2,j), the time-averaged rate of energy dissipation per unit horizontal area due to bottom friction, $\overline{D_f}$ = ENER(3,j), and the time-averaged rate of energy dissipation per unit horizontal area due to wave breaking, $\overline{D_B}$ = ENER(4,j), as a function of the normalized horizontal coordinate $x = (j-1)\Delta x$. Fig. B-7 is plotted from the file with its unit number = 35 and its file name = OENERG. Fig. B-7 indicates that the wave energy dissipation due to wave breaking is small for this example with the surf similarity parameter $\xi = 4.40$ as listed in Table B-3.

Fig. B-8 shows the variations of the normalized free surface elevation η = ELEV(j) as a function of the normalized horizontal coordinate $x = (j-1)\Delta x$ at the normalized time t = (N/NONE) = 5, 5.25, 5.5, 5.75 and 6 plotted from the file with its unit number = 22 and its file name = OSPACE. Fig. B-8 also shows the normalized structure geometry in the same way as Fig. B-4. It is noted that the spatial variation of η at t=6 is the same as that at t=5 because of the periodicity.

Fig. B-9 shows the variations of the normalized fluid velocity u=V(j) as a function of the normalized horizontal coordinate $x = (j-1)\Delta x$ at the normalized time t = (N/NONE) = 5, 5.25. 5.5, 5.75 and 6 plotted from the file with its unit number = 22 and its file name = OSPACE. The spatial variation of u at t=6 is the same as that at t=5 because of the periodicity.

Fig. B-10 shows the variations of the normalized horizontal velocity, u = V(j), the normalized fluid acceleration, du/dt = DUDT(j), and the armor stability function N_R = SNR(j) as a function of the normalized vertical location on the slope, z = ZB(j), at the time level N = NSNSC = 21492 corresponding to the time when the minimum value of N_R shown in Fig. B-10 is equal to the critical stability number N_{sc} = 2.401. Fig. B-10 is plotted from the file with its unit number = 40 and its file name = OREQ together with the file containing ZB(j). The values of N_{sc} and NSNSC are listed in Table B-3. In order to plot Fig. B-10, IREQ=1, IELEV=1, IV=1, IDUDT=1, ISNR=1, NNREQ=1 and NREQ(1) = 21492 are specified as input as shown in Table B-1. The critical time level NSNSC=21492 was computed beforehand using IREQ=0. This implies that computation needs to be made twice to plot a figure similar to Fig. B-10. IBREAK could be modified without difficulties so as to store the requested quantities at the time when the critical stability number N_{sc} occurs. The special storage option included in IBREAK allows the storage of the requested quantities at the specified time levels N=NREQ(i) with i=1,2,...,NNREQ. Consequently, this option can be used for other purposes as well.

Fig. B-11 shows the variation of the local stability number, N_{sx} = SNSX(j), as a function of the normalized vertical location on the slope z=ZB(j) plotted from the file with its unit number = 41 and its file name = OSTAB1. The minimum value of N_{sx} shown in Fig. B-11 is equal to N_{sc} = 2.401.

TABLE B-1

```
         3 = number of comment lines
----------------------------------------------------------------------
Ahrens (1975) Test 12 (Zero Damage)
----------------------------------------------------------------------
11    20001                          --> IJOB,ISTAB,NSTAB
2                                    --> ISYST
1                                    --> IBOT
0                                    --> INONCT
1                                    --> IENERG
1  no wave data file                 --> IWAVE,FINP2
100   20000    5    0    0           --> ISAVA-B-C,NSAVA,NTIMES,NNOD1,NNOD2
11111      1                         --> IREQ,IELEV,IV,IDUDT,ISNR,NNREQ
21492                                --> NREQ(1)
   24000                             --> NTOP
    4000                             --> NONE
      20                             --> NJUM1
     100                             --> S
      .300000                        --> FWP
     1.000000        1.000000        --> X1,X2
      .001000                        --> DELTA(normalized)
       3                             --> NDELR
      .157480                        --> DELRP(1)(inches)
      .787402                        --> DELRP(2)
     1.574803                        --> DELRP(3)
     3.060000        8.500000        --> HREFP(feet),TP(seconds)
     1.061400        1.061400        --> KSREF,KSSEA
    15.000000                        --> DSEAP(feet)
      .400000                        --> TSLOPS
       1                             --> NBSEG
    60.000000         .400000        --> WBSEG(1)(feet),TBSLOP(1)
       1                                           --> NJUM2
      .900000         .660000         2.710000     --> C2,C3,SG
      .500000         .180000         1.500000     --> CD,CL,CM
     1.191754                                      --> TANPHI
     1.000000        -.800000                      --> AMAX,AMIN
```

TABLE B-2

```
-----------------------------------------------------------------
Ahrens (1975) Test 12 (Zero Damage)
-----------------------------------------------------------------

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     100; N =    1806
 Adjusted values: U(2,S) =   0.974E-03; U(2,S-1) =  0.195E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     138; N =    5293
 Adjusted values: U(2,S) =   0.809E-03; U(2,S-1) =  0.162E-02
```

TABLE B-3

```
---------------------------------------------------------------
Ahrens (1975) Test 12 (Zero Damage)
---------------------------------------------------------------

WAVE CONDITION

Cnoidal Incident Wave at Seaward Boundary
1-m =       0.112815261D+00
E   =       0.111509461D+01
K   =       0.252149782D+01
Norm. Maximum Surface Elev.      =      0.628691
Norm. Minimum Surface Elev.      =     -0.371309

Reference Wave Period            =      8.500000 sec.
Reference Wave Height            =      3.060000 feet
Depth at Seaward Boundary        =     15.000000 feet
Shoal. Coef. at Reference  Ks1 =      1.061
             at Seaw. Bdr. Ks2 =      1.061
             Ks = Ks2/Ks1        =      1.000
Norm. Depth at Seaw. Bdr.        =      4.902
Normalized Wave Length           =     12.144
"Sigma"                          =     27.573
Ursell Number                    =     30.084
Surf Similarity Parameter        =      4.400

SLOPE PROPERTIES

Friction Factor              =     0.300000
Norm. Friction Factor        =     4.135970
Norm. Horiz. Length of
      Computation Domain     =     0.711121
Number of Segments           =       1
--------------------------------
 SEGMENT    WBSEG(I)   TBSLOP(I)
    I        feet
--------------------------------
      1   60.000000    0.400000

COMPUTATION PARAMETERS

Normalized Delta x    =   0.444451D-02
Normalized Delta t    =   0.250000D-03
Normalized DELTA      =   0.100000E-02
Damping Coeff.     x1 =     1.000
                   x2 =     1.000
Num. Stab. Indicator =     3.419
Total Number of Time Steps        NTOP =     24000
Number of Time Steps in 1 Wave Period
                                  NONE =      4000
Total Number of Spatial Nodes       JE =      161
Number of Nodes Along Bottom Below SWL
                                     S =      100
Storing Temporal Variations at Every
                                 NJUM1 =        20 Time Steps
Armor Stability Number Computed
                      at Every   NJUM2 =         1 Time Steps
```

PARAMETERS FOR ARMOR STABILITY AND MOVEMENT

```
Armor Friction Factor       =     1.192
Specific Gravity            =     2.710
Area Coefficient       C2   =     0.900
Volume Coefficient     C3   =     0.660
Drag Coefficient       CD   =     0.500
Lift Coefficient       CL   =     0.180
Inertia Coefficient    CM   =     1.500
Norm. Upper and Lower Bounds of du/dt
                     AMAX   =     1.000
                     AMIN   =    -0.800
```

REFLECTION COEFFICIENTS

```
r1 =     0.553
r2 =     0.546
r3 =     0.544
```

RUNUP, RUNDOWN, SETUP

Largest Node Number Reached by Computational Waterline
JMAX = 139

I	DELTAR(I) [inch]	RUNUP(I) R	RUNDOWN(I) Rd	SETUP(I) Zr
1	0.157	1.864	0.656	1.318
2	0.787	1.853	-0.283	0.909
3	1.575	1.842	-0.714	0.705

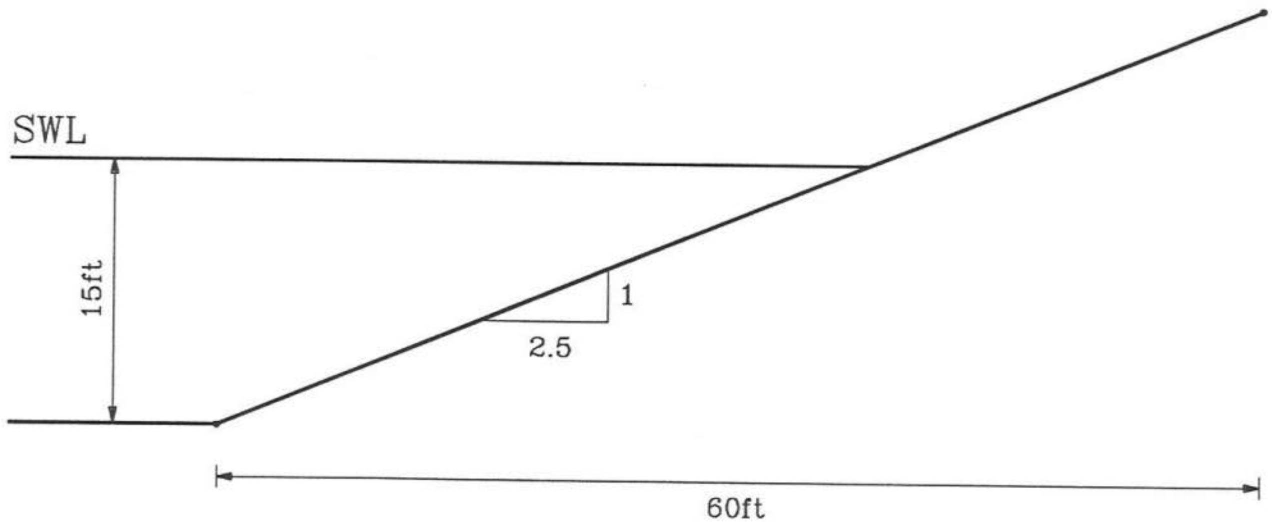
STABILITY NUMBER

```
Critical Stability Number Nsc =    2.401
At Node Number              J =       85
At Time Level               N =    21492
```

WAVE SET-DOWN OR SETUP

```
Average value of ETAI =     0.000000
                 ETAR =     0.015310
```

```
TIME-AVERAGED ENERGY BALANCE

Normalized Energy Flux:
. at Seaw. Boundary    A =  0.202324D+00
. at Landw. Boundary   B =  0.000000D+00
Normalized Rate of Energy Dissipation
in the Computation Domain, Due to:
. bottom friction      C =  0.197916D+00
. wave breaking        D =  0.440842D-02
Calculation 1:
                E = A-B =  0.202324D+00
                F = C+D =  0.202324D+00
  Must G=0, but G = F-E =  0.345043D-06
  % error  100G/E =          0.00
Approximate Energy Flux, Based on
Linear Long Wave, Due to:
. incident wave at seaw. boundary      P =  0.276755D+00
. reflected wave at seaw. boundary     Q =  0.818700D-01
Calculation 2:
Net Energy Flux at Seaw. Bndry. S = P-Q =  0.194885D+00
% Error at Seaward Boundary  100(S-A)/A =          -3.68
```



Ahrens (1975) Test No. 12

Figure B–1

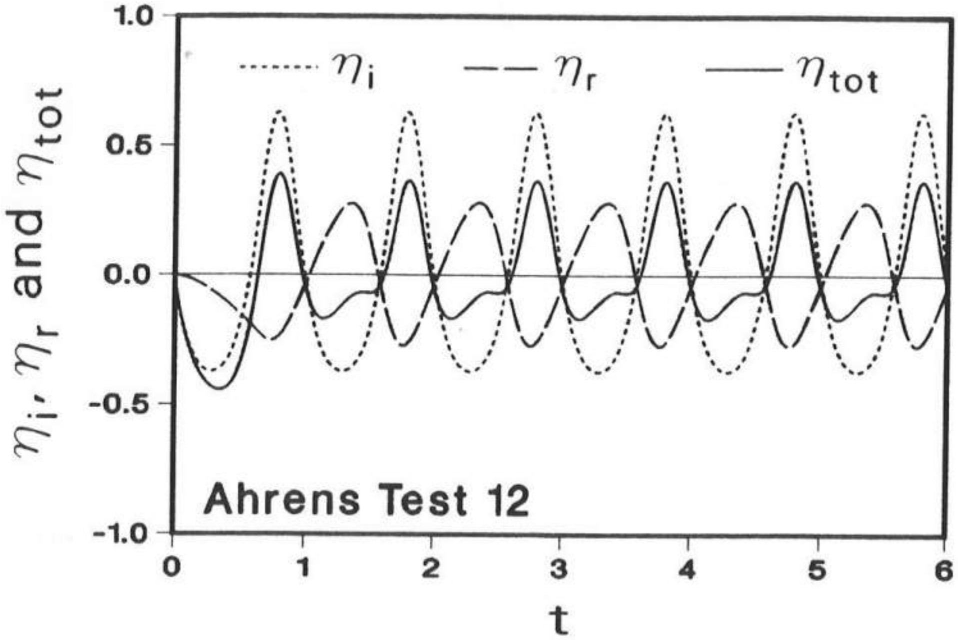


Figure B-2

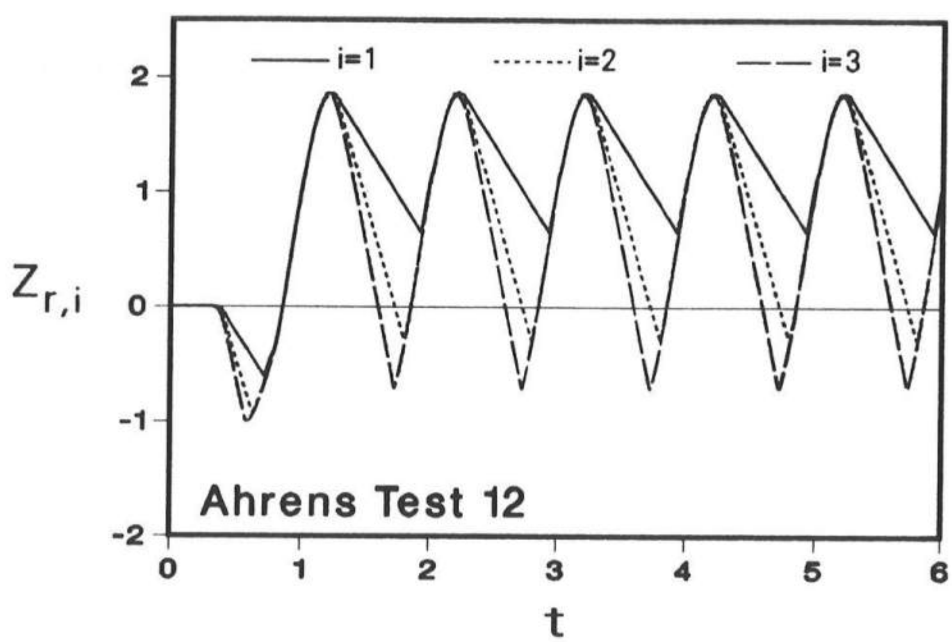


Figure B-3

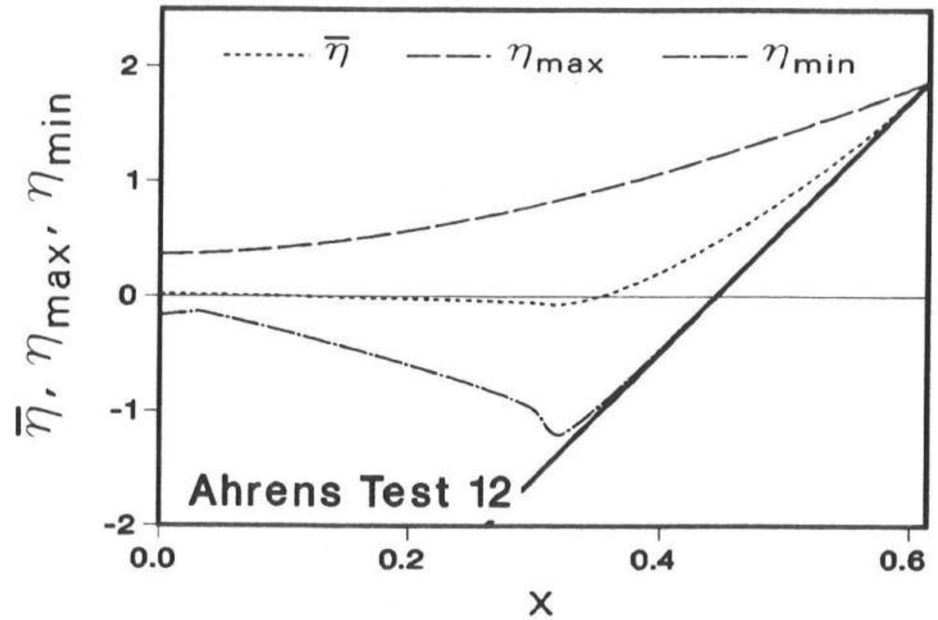


Figure B-4

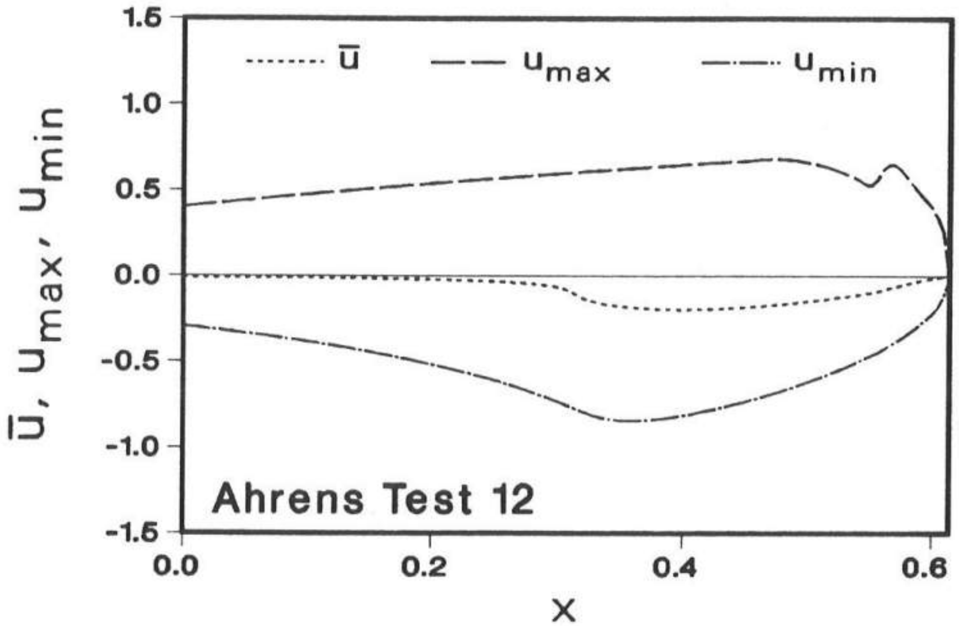


Figure B-5

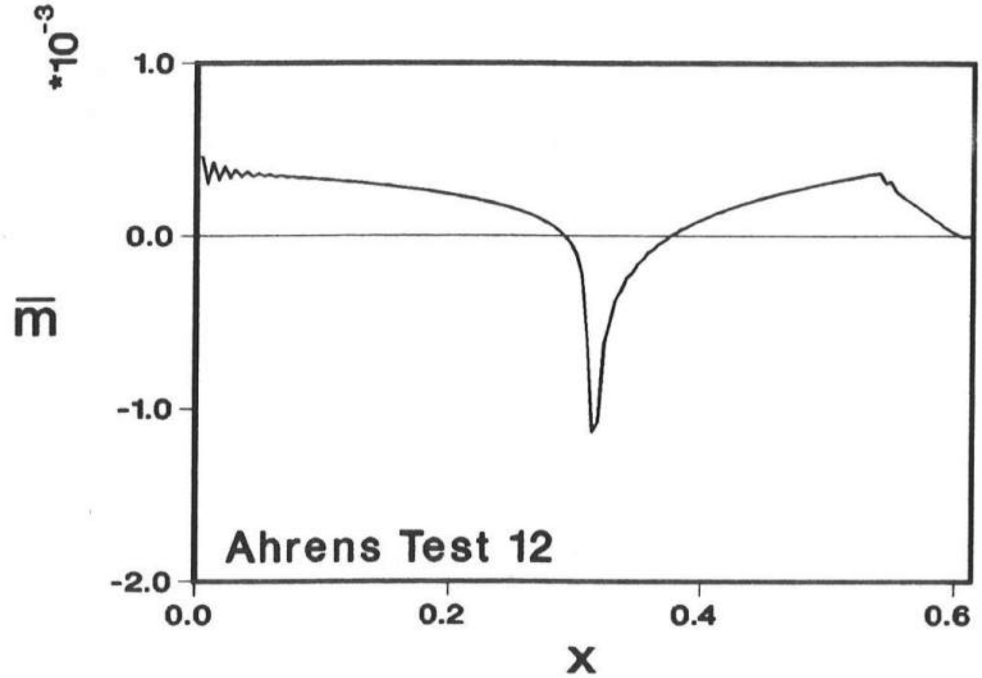


Figure B-6

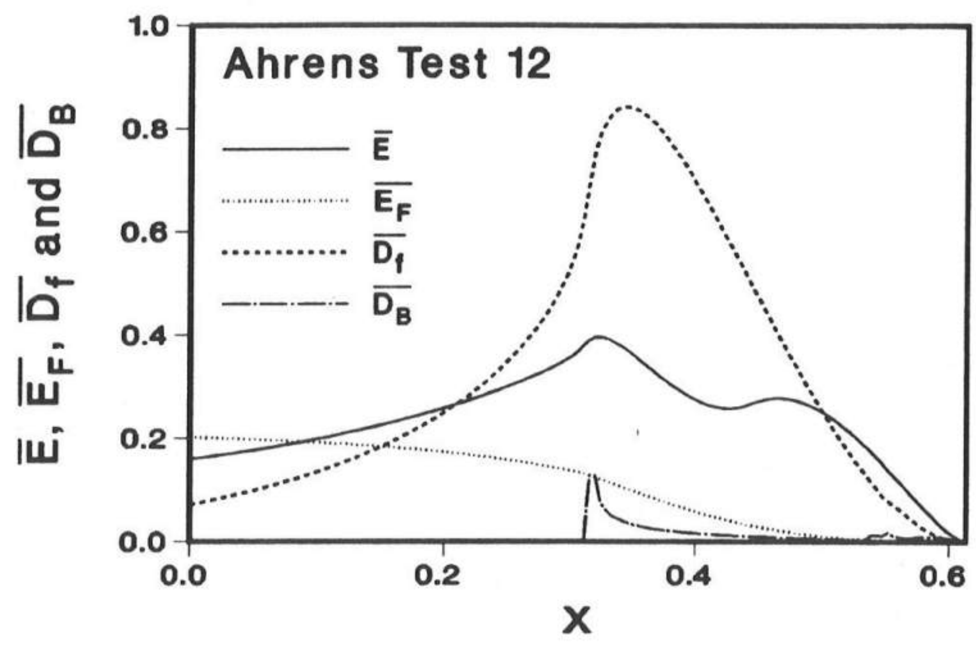


Figure B-7

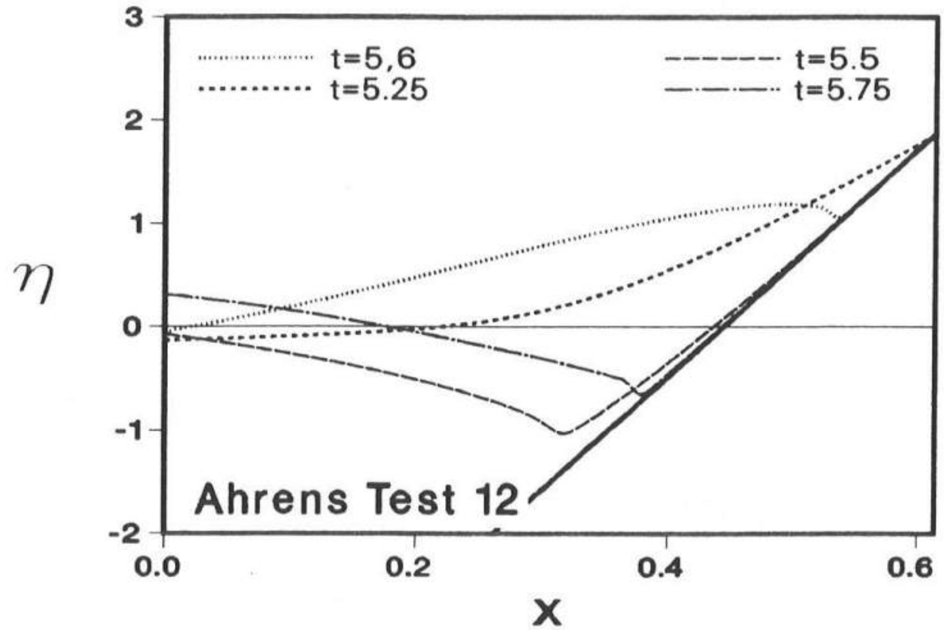


Figure B-8

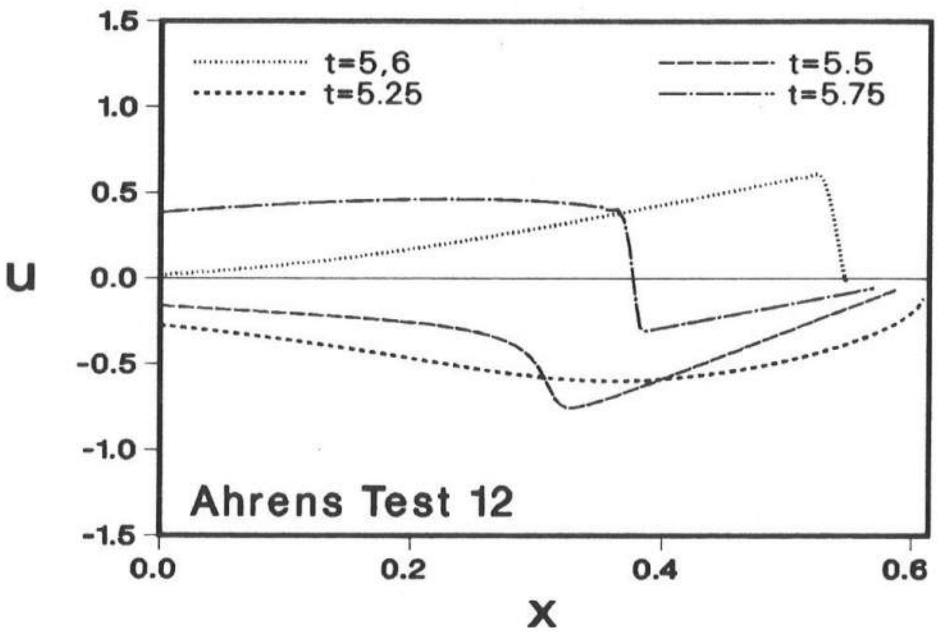


Figure B-9

Ahrens Test 12

Critical Conditions:

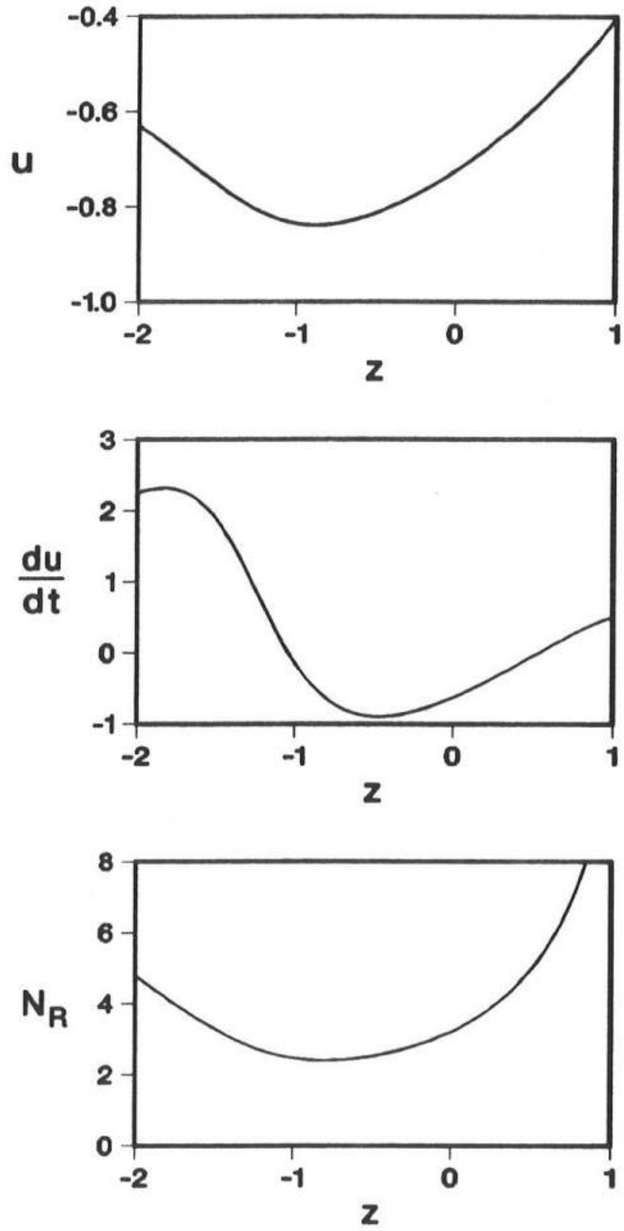


Figure B-10

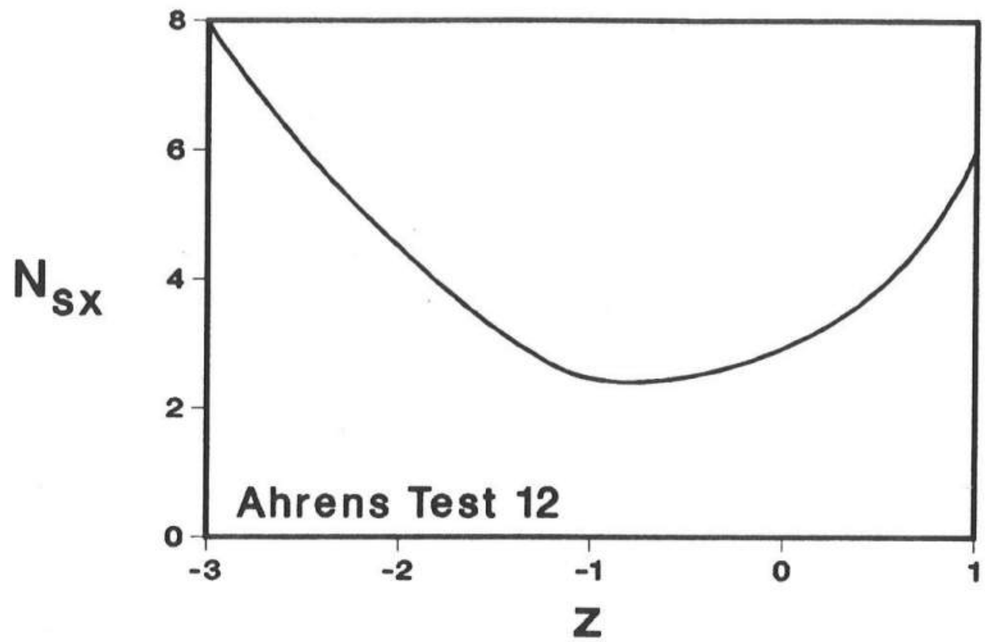


Figure B-11

APPENDIX C: EXAMPLE OF WAVE RUNUP AND ARMOR MOVEMENT

The second example is the Test No. 32 of the large-scale riprap tests of Ahrens (1975) and is similar to the first example, where IJOB=1 for both examples.

Table C-1 lists the input data used for the second example and is similar to Table B-1. Use is made of ISTAB=2 and the movement of individual armor units is computed unlike the first example for which ISTAB=1 as shown in Table B-1. The representative wave height H_r' = HREFP is taken as the wave height H' measured at the toe of the 1:3.5 riprap slope. Since $H_r' = H'$, KSSEA=KSREF=1.0 could also be used instead of KSSEA=KSREF=0.9421 used in Table C-1 on the basis of linear wave shoaling.

Table C-2 lists the warning messages written during the time-marching computation in the same way as Table B-2.

Table C-3 lists the concise output for the second example. Table C-3 includes the quantities related to the armor movement unlike Table B-3 in which the quantities related to the armor stability have been written.

Table C-4 lists the values of ISTATE(j), NODIN(j), NODFI(j), NDIS(j), XB(j), ZB(j) and XA(j) explained in the commons /STAB7/, /STAB8/ and /BOT3/ at the time level N = NTOP = 14000, that is, the normalized time t = (NTOP/NONE) = 7 for the armor units located at the node j at t=0 and moved during $0 < t \leq 7$. Table C-4 is written from the file with its unit number = 42 and its file name = OSTAB2. The normalized armor displacement X_a defined by Eq. 81a is small for this example. This suggests that armor units may move but may not be dislodged from their initial locations.

Fig. C-1 shows the dimensional structure geometry specified as input in Table C-1.

Fig. C-2 shows the temporal variations of the incident wave profile η_i, the reflected wave profile η_r and the total wave profile $\eta_{tot} = (\eta_i + \eta_r)$. Fig. C-2 is similar to Fig. B-2 except that wave reflection for the second example with the surf similarity parameter $\xi = 2.00$ is less than that for the first example with $\xi = 4.40$.

Fig. C-3 shows the temporal variations of the normalized waterline elevation Z_r with i=1,2 and 3 corresponding to the physical depth $\delta_r' = 0.157$, 0.787 and 1.575 inches. The waterline oscillations for the second example with $\xi = 2.00$ are less than those shown in Fig. B-3 for the first example with $\xi = 4.40$.

Fig. C-4 shows the spatial variations of the mean water level $\overline{\eta}$, the crest elevation η_{max}, and the trough elevation η_{min} in the same way as Fig. B-4.

Fig. C-5 shows the spatial variations of the time-averaged velocity $\overline{u}$, the maximum velocity u_{max} and the minimum velocity u_{min} in the same as Fig. B-5.

Fig. C-6 shows the spatial variation of the time-averaged volume flux per unit width $\overline{m}$ in the same way as Fig. B-6. The computed values of $\overline{m}$, which are on the order of 10^{-3} or less, satisfy the time-averaged continuity equation $\overline{m} = 0$ almost exactly.

Fig. C-7 shows the spatial variations of the time-averaged energy quantities $\overline{E}$, $\overline{E_F}$, $\overline{D_f}$ and $\overline{D_B}$ in the same way as Fig. B-7. The wave energy dissipation due to wave breaking for the second example with $\xi = 2.00$ is larger than that for the first example with $\xi = 4.40$.

Fig. C-8 shows the spatial variations of the free surface elevation η at the time t = 6, 6.25, 6.5, 6.75 and 7. Fig. C-8 indicates wave breaking for the second example with ξ = 2.00 unlike Fig. B-8 for the first example with ξ = 4.40. It is noted that the numerical method used for IBREAK produces a steep wave front whenever wave breaking occurs. A two-dimensional model would be required to simulate the wave breaking process in a more realistic manner.

Fig. C-9 shows the spatial variations of the horizontal velocity u at the time t = 6, 6.25, 6.5, 6.75 and 7 in the same way as Fig. B-9 except that the computation duration $0 \leq t \leq 7$ for the second example is taken to be longer because of the computation of the armor movement which will not become periodic even after the establishment of the periodic flow field.

Fig. C-10 plots the values of the normalized displacement X_a at t=7 as a function of the corresponding values of the initial vertical location on the slope z.

Fig. C-11 shows the temporal variations of X_a for the two armor units initially located at z = −1.283 and z = 0.797, corresponding to the node number j = 72 and j = 119, respectively, as listed in Table C-4. The temporal variations of X_a for j = 72 and 119 are plotted from the files with their unit numbers = 75 and 76, respectively. The corresponding file names specified as input are 0072 and 0119 as listed in Table C-1.

TABLE C-1

```
      3 = number of comment lines
-----------------------------------------------------------------
Ahrens (1975) Test 32 (Zero Damage)
-----------------------------------------------------------------
12      1                    --> IJOB,ISTAB,NSTAB
2                            --> ISYST
1                            --> IBOT
0                            --> INONCT
1                            --> IENERG
1  no wave data file         --> IWAVE,FINP2
101   12000   5   0   2      --> ISAVA-B-C,NSAVA,NTIMES,NNOD1,NNOD2
00000     0                  --> IREQ,IELEV,IV,IDUDT,ISNR,NNREQ
   14000                     --> NTOP
    2000                     --> NONE
      10                     --> NJUM1
     100                     --> S
      .300000                --> FWP
     1.000000     1.000000   --> X1,X2
      .001000                --> DELTA(normalized)
      3                      --> NDELR
      .157480                --> DELRP(1) (inches)
      .787402                --> DELRP(2)
     1.574803                --> DELRP(3)
     3.390000     5.700000   --> HREFP(feet),TP(seconds)
      .942100      .942100   --> KSREF,KSSEA
    15.000000                --> DSEAP(feet)
      .285714                --> TSLOPS
      1                      --> NBSEG
    85.000000      .285714   --> WBSEG(1)(feet),TBSLOP(1)
      1                                   --> NJUM2
      .900000      .660000     2.710000   --> C2,C3,SG
      .500000      .400000     1.500000   --> CD,CL,CM
     1.191754                             --> TANPHI
     1.000000     -.800000                --> AMAX,AMIN
     1.030000                             --> DAP(feet)
    72  0072                         --> NODNO2(1),FNAME2(I)
   119  0119                         -->           (2)         (2)
```

TABLE C-2

```
-------------------------------------------------------------------
Ahrens (1975) Test 32 (Zero Damage)
-------------------------------------------------------------------

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =    130; N =    3706
 Adjusted values: U(2,S) =   0.112E-02; U(2,S-1) =  0.224E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =    130; N =    5616
 Adjusted values: U(2,S) =   0.145E-02; U(2,S-1) =  0.290E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =    130; N =    5702
 Adjusted values: U(2,S) =   0.639E-03; U(2,S-1) =  0.128E-02
```

TABLE C-3

```
--------------------------------------------------------------------
Ahrens (1975) Test 32 (Zero Damage)
--------------------------------------------------------------------

WAVE CONDITION

Stokes II Incident Wave at Seaward Boundary
Norm. Maximum Surface Elev.      =     0.594242
Norm. Minimum Surface Elev.      =    -0.405758

Reference Wave Period            =     5.700000 sec.
Reference Wave Height            =     3.390000 feet
Depth at Seaward Boundary        =    15.000000 feet
Shoal. Coef. at Reference  Ks1 =     0.942
             at Seaw. Bdr. Ks2 =     0.942
             Ks = Ks2/Ks1        =     1.000
Norm. Depth at Seaw. Bdr.        =     4.425
Normalized Wave Length           =     7.560
"Sigma"                          =    17.567
Ursell Number                    =    12.917
Surf Similarity Parameter        =     2.002

SLOPE PROPERTIES

Friction Factor              =    0.300000
Norm. Friction Factor        =    2.635082
Norm. Horiz. Length of
      Computation Domain     =    1.419329
Number of Segments           =       1
--------------------------------
 SEGMENT     WBSEG(I)    TBSLOP(I)
    I          feet
--------------------------------
       1   85.000000     0.285714

COMPUTATION PARAMETERS

Normalized Delta x    =    0.881571D-02
Normalized Delta t    =    0.500000D-03
Normalized DELTA      =    0.100000E-02
Damping Coeff.     x1 =      1.000
                   x2 =      1.000
Num. Stab. Indicator =      3.511
Total Number of Time Steps        NTOP =     14000
Number of Time Steps in 1 Wave Period
                                  NONE =      2000
Total Number of Spatial Nodes       JE =       162
Number of Nodes Along Bottom Below SWL
                                     S =       100
Storing Temporal Variations at Every
                                 NJUM1 =        10 Time Steps
Armor Stability Number Computed
                      at Every   NJUM2 =         1 Time Steps
```

PARAMETERS FOR ARMOR STABILITY AND MOVEMENT

Armor Friction Factor		=	1.192
Specific Gravity		=	2.710
Area Coefficient	C2	=	0.900
Volume Coefficient	C3	=	0.660
Drag Coefficient	CD	=	0.500
Lift Coefficient	CL	=	0.400
Inertia Coefficient	CM	=	1.500
Armor Diameter		=	1.030000 feet

REFLECTION COEFFICIENTS

r1 = 0.173
r2 = 0.185
r3 = 0.176

RUNUP, RUNDOWN, SETUP

Largest Node Number Reached by Computational Waterline
JMAX = 130

I	DELTAR(I) [inch]	RUNUP(I) R	RUNDOWN(I) Rd	SETUP(I) Zr
1	0.157	1.284	0.892	1.116
2	0.787	1.275	0.538	0.976
3	1.575	1.265	0.309	0.887

ARMOR UNITS MOVEMENT

Number of Units Moved = 64
Number of Units Stopped = 34

WAVE SET-DOWN OR SETUP

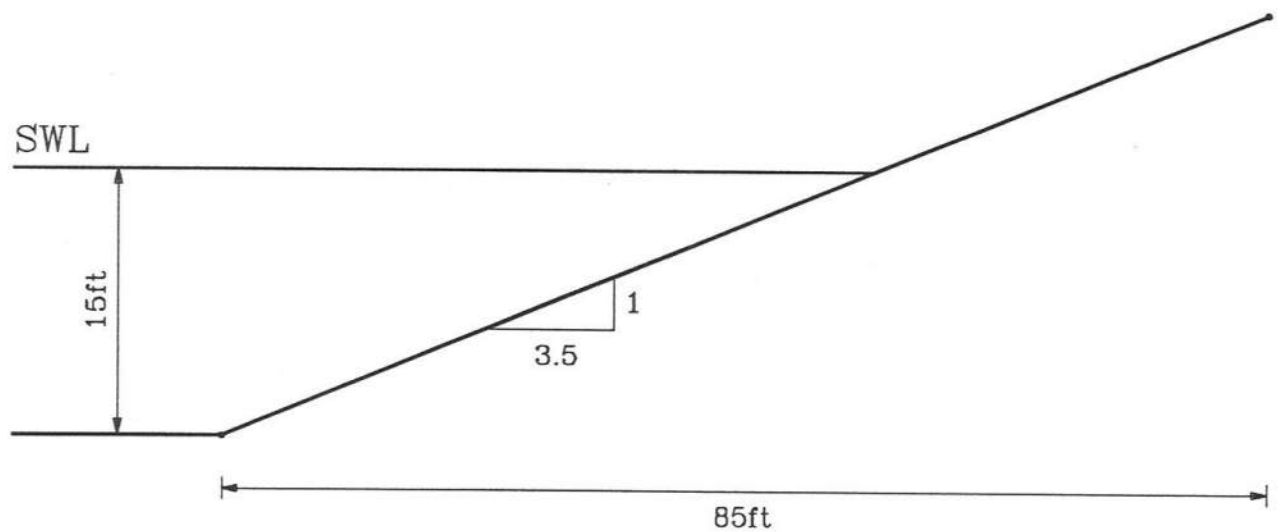
Average value of ETAI = 0.000000
ETAR = 0.019853

```
TIME-AVERAGED ENERGY BALANCE

Normalized Energy Flux:
. at Seaw. Boundary   A =  0.271704D+00
. at Landw. Boundary  B =  0.000000D+00
Normalized Rate of Energy Dissipation
in the Computation Domain, Due to:
. bottom friction      C =  0.205248D+00
. wave breaking        D =  0.664560D-01
Calculation 1:
                E = A-B =  0.271704D+00
                F = C+D =  0.271704D+00
  Must G=0, but G = F-E =  0.512847D-07
  % error  100G/E =          0.00
Approximate Energy Flux, Based on
Linear Long Wave, Due to:
. incident wave at seaw. boundary     P =  0.262939D+00
. reflected wave at seaw. boundary    Q =  0.814467D-02
Calculation 2:
Net Energy Flux at Seaw. Bndry. S = P-Q =  0.254795D+00
% Error at Seaward Boundary  100(S-A)/A =         -6.22
```

TABLE C-4

No.	Status at t=7	Node No. at Initial Location	Node No. at Final Location	Time Step When Start Moving	Coordinates of Initial Location x	Coordinates of Initial Location z	Xa at t=7
1	stopped	67	66	3806	0.582	-1.504	-0.269
2	moving	68	68	3819	0.591	-1.460	-0.090
3	moving	69	69	3835	0.600	-1.416	0.032
4	moving	70	70	1830	0.608	-1.372	0.211
5	moving	71	71	1840	0.617	-1.327	0.141
6	moving	72	72	1856	0.626	-1.283	0.125
7	moving	73	74	1873	0.635	-1.239	0.417
8	moving	74	74	1891	0.643	-1.195	0.185
9	moving	75	76	1909	0.652	-1.150	0.425
10	moving	76	76	1926	0.661	-1.106	-0.037
11	moving	77	77	1943	0.670	-1.062	0.041
12	moving	78	78	1959	0.679	-1.018	-0.161
13	moving	79	79	1975	0.688	-0.974	-0.159
14	stopped	80	80	1991	0.696	-0.929	-0.051
15	stopped	81	81	2006	0.705	-0.885	-0.128
16	moving	82	82	2021	0.714	-0.841	-0.168
17	moving	83	83	2036	0.723	-0.797	-0.182
18	moving	84	84	2051	0.732	-0.752	-0.250
19	moving	85	85	2065	0.740	-0.708	-0.023
20	moving	86	85	2080	0.749	-0.664	-0.302
21	stopped	87	87	2094	0.758	-0.619	-0.173
22	stopped	88	87	2109	0.767	-0.575	-0.271
23	stopped	89	89	2123	0.776	-0.531	-0.245
24	stopped	90	90	2138	0.785	-0.487	-0.119
25	moving	91	90	2153	0.793	-0.442	-0.322
26	moving	92	91	2168	0.802	-0.398	-0.391
27	stopped	93	92	2183	0.811	-0.354	-0.396
28	moving	94	93	2199	0.820	-0.310	-0.308
29	moving	95	94	2215	0.829	-0.266	-0.545
30	moving	96	95	2231	0.837	-0.221	-0.477
31	moving	97	96	2201	0.846	-0.177	-0.598
32	moving	98	97	2217	0.855	-0.133	-0.640
33	stopped	99	98	2255	0.864	-0.089	-0.449
34	moving	100	99	2276	0.873	-0.044	-0.561
35	stopped	101	101	2295	0.882	0.000	-0.179
36	stopped	102	102	2313	0.890	0.044	-0.089
37	stopped	103	103	2330	0.899	0.089	-0.172
38	stopped	104	103	2346	0.908	0.133	-0.286
39	stopped	105	104	2363	0.917	0.177	-0.332
40	stopped	106	105	2380	0.926	0.221	-0.293
41	stopped	107	107	2398	0.934	0.266	-0.241
42	stopped	108	108	2417	0.943	0.310	-0.186
43	stopped	109	108	2437	0.952	0.354	-0.344
44	stopped	110	109	2459	0.961	0.398	-0.359
45	stopped	111	110	2482	0.970	0.442	-0.265
46	stopped	112	111	2506	0.979	0.487	-0.328
47	stopped	113	113	2531	0.987	0.531	-0.201
48	stopped	114	114	2556	0.996	0.575	-0.186
49	stopped	115	114	2581	1.005	0.619	-0.288
50	stopped	116	115	2607	1.014	0.664	-0.474

No.	Status at t=7	Node No. at Initial Location	Node No. at Final Location	Time Step When Start Moving	Coordinates of Initial Location		Xa at t=7
					x	z	
51	stopped	117	117	2633	1.023	0.708	-0.132
52	stopped	118	117	2661	1.031	0.752	-0.282
53	stopped	119	119	2690	1.040	0.797	-0.045
54	stopped	120	119	2720	1.049	0.841	-0.473
55	stopped	121	121	2752	1.058	0.885	-0.047
56	stopped	122	122	2785	1.067	0.929	-0.172
57	stopped	123	123	2821	1.076	0.974	-0.132
58	stopped	124	123	2860	1.084	1.018	-0.267
59	moving	125	124	2901	1.093	1.062	-0.345
60	moving	126	125	2948	1.102	1.106	-0.349
61	stopped	127	126	3000	1.111	1.150	-0.275
62	moving	128	128	3061	1.120	1.195	-0.221
63	moving	129	129	3138	1.128	1.239	-0.135
64	moving	130	130	5471	1.137	1.283	-0.042



Ahrens (1975) Test No. 32

Figure C-1

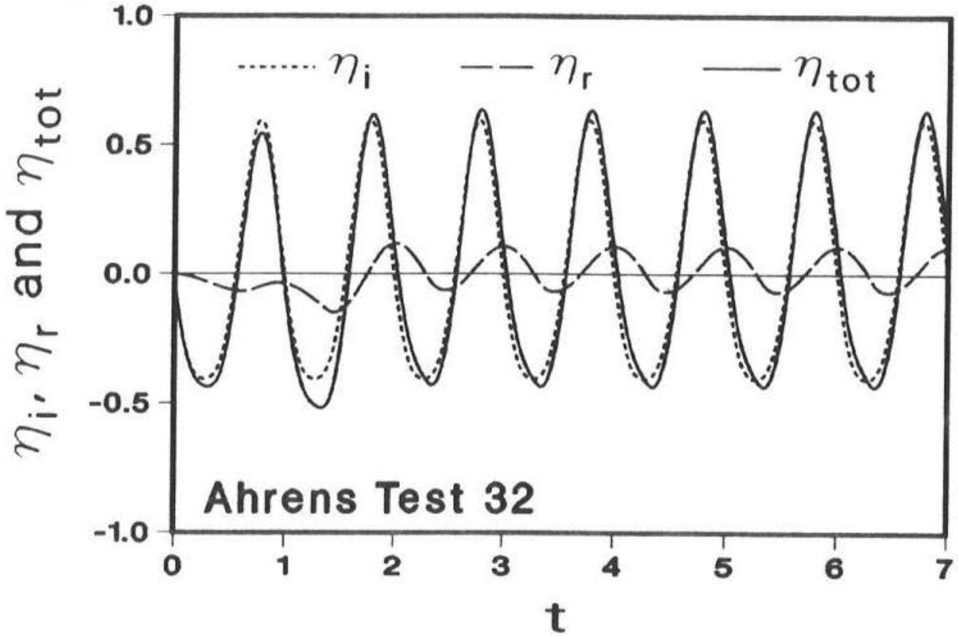


Figure C-2

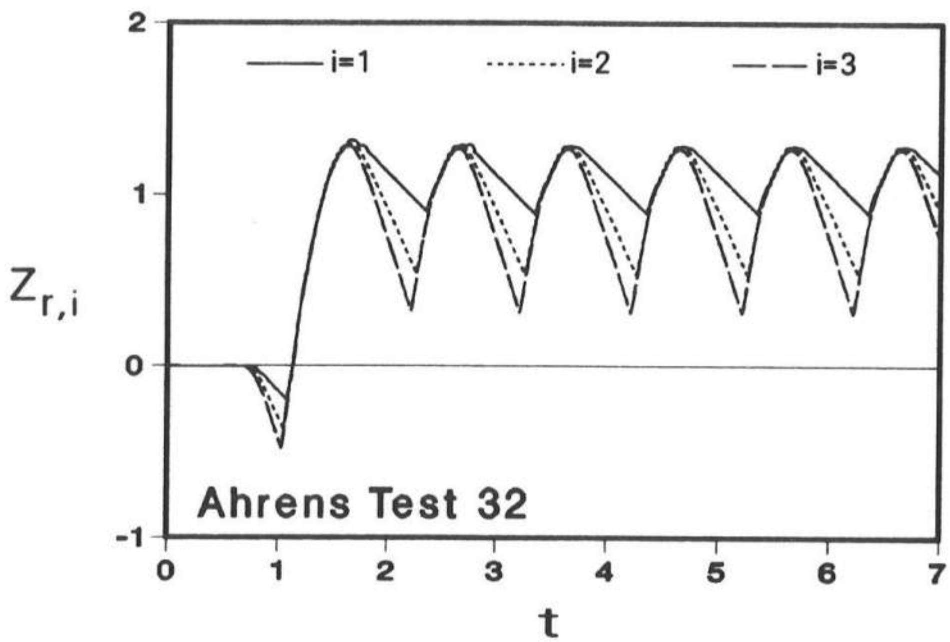


Figure C-3

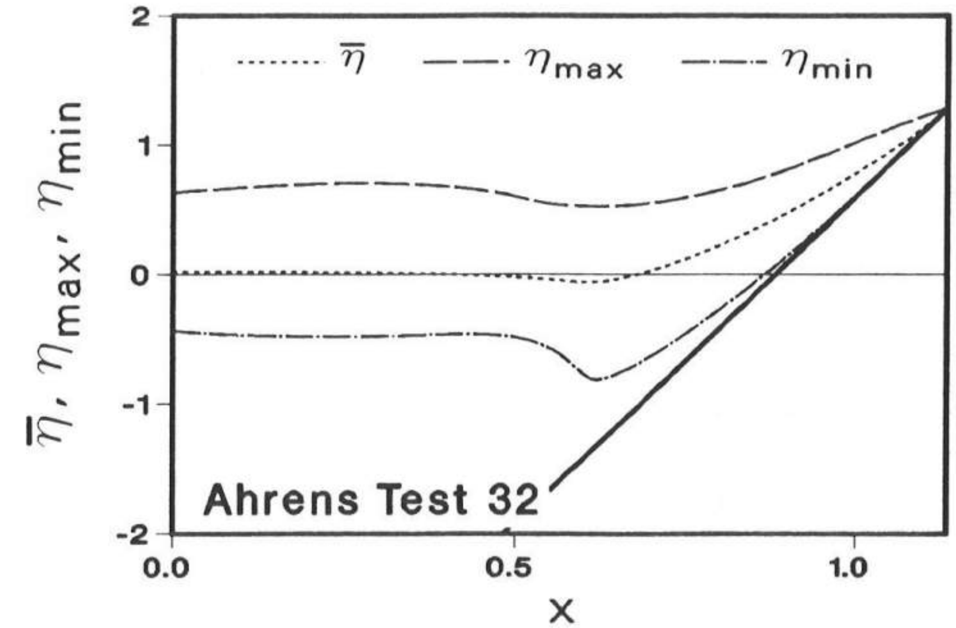


Figure C-4

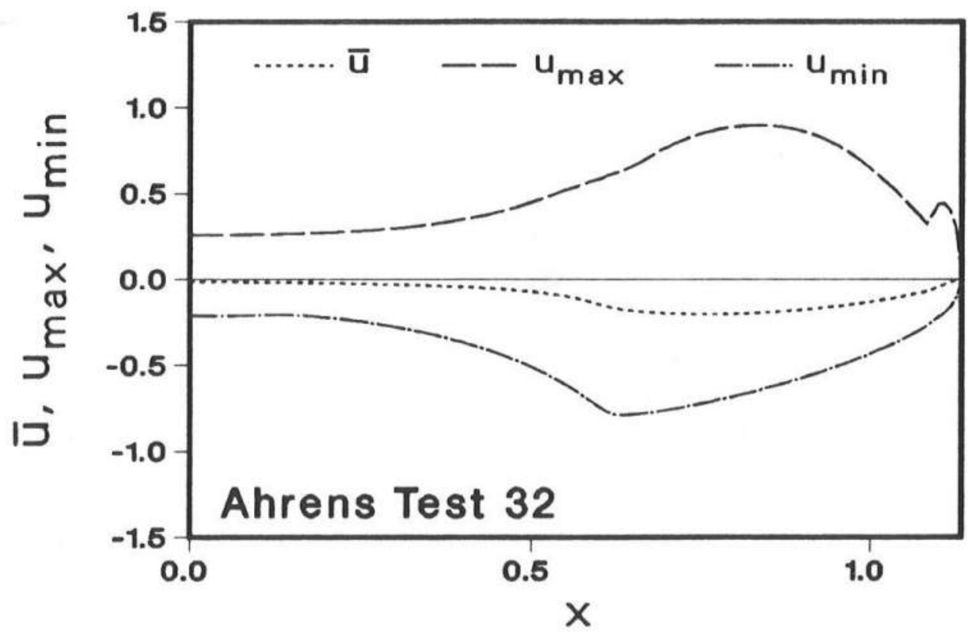


Figure C-5

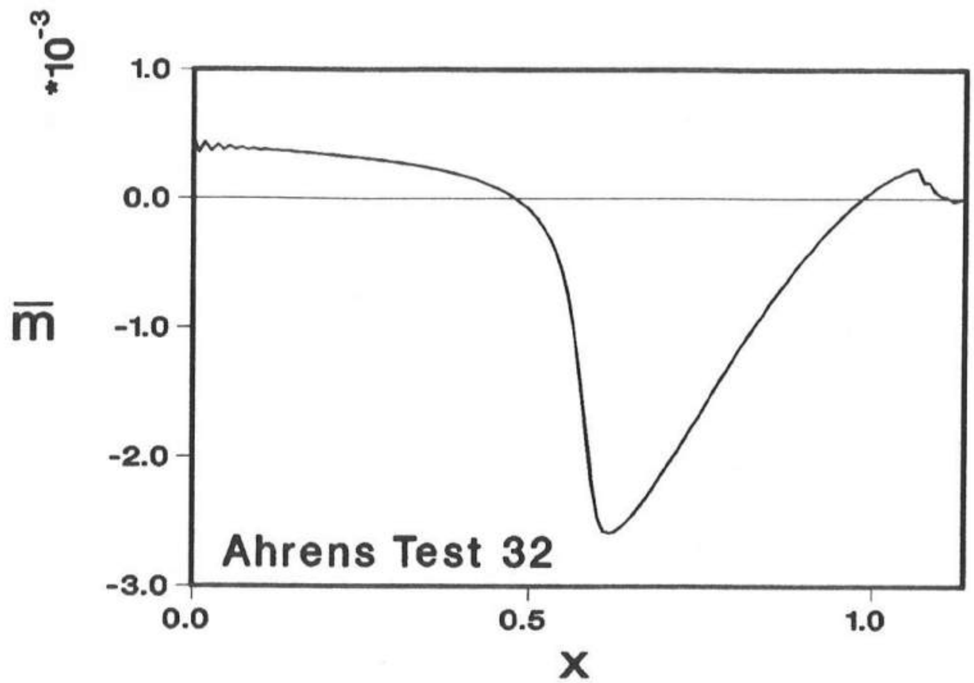


Figure C-6

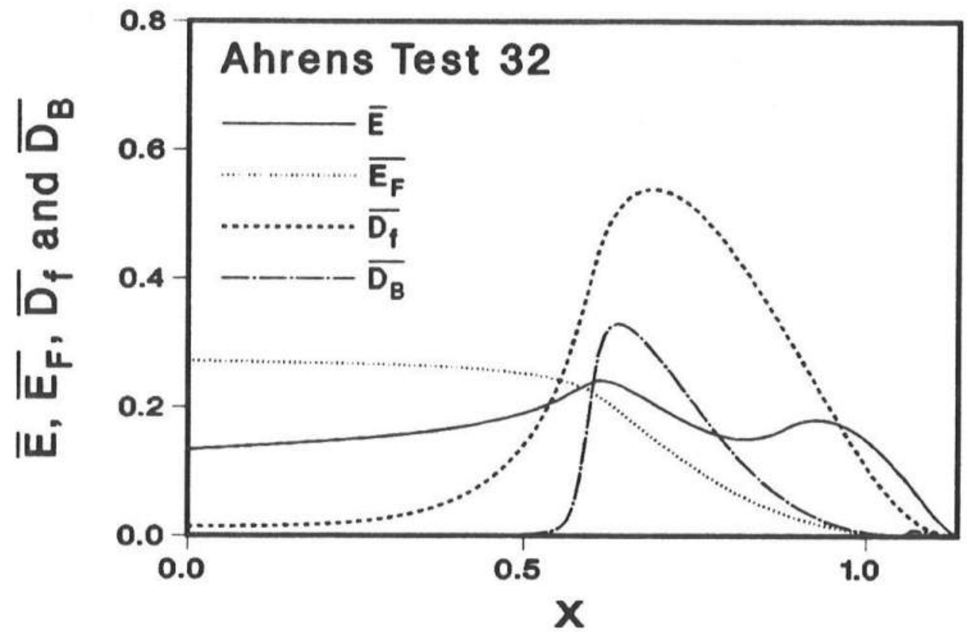


Figure C-7

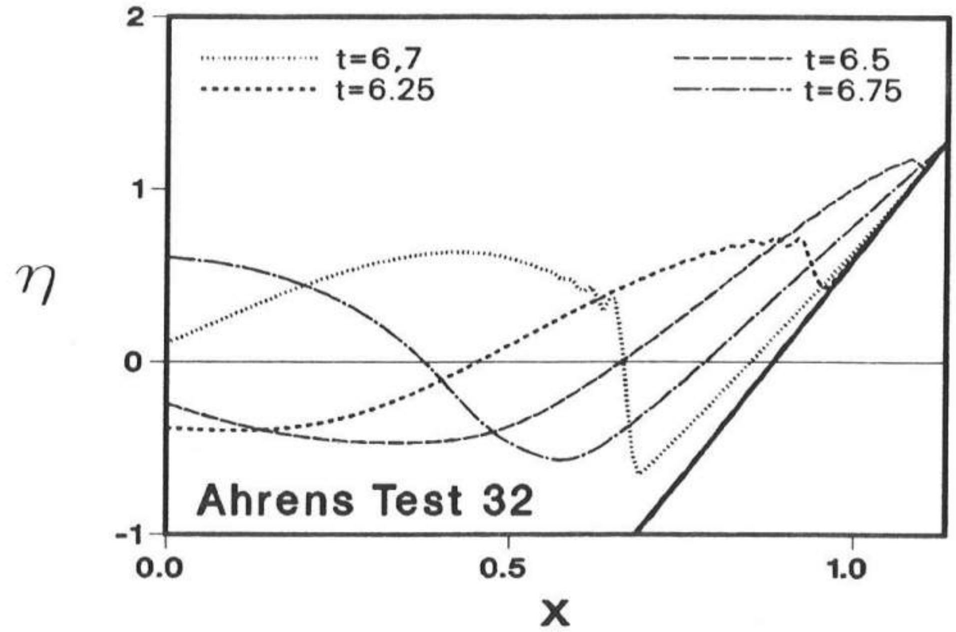


Figure C-8

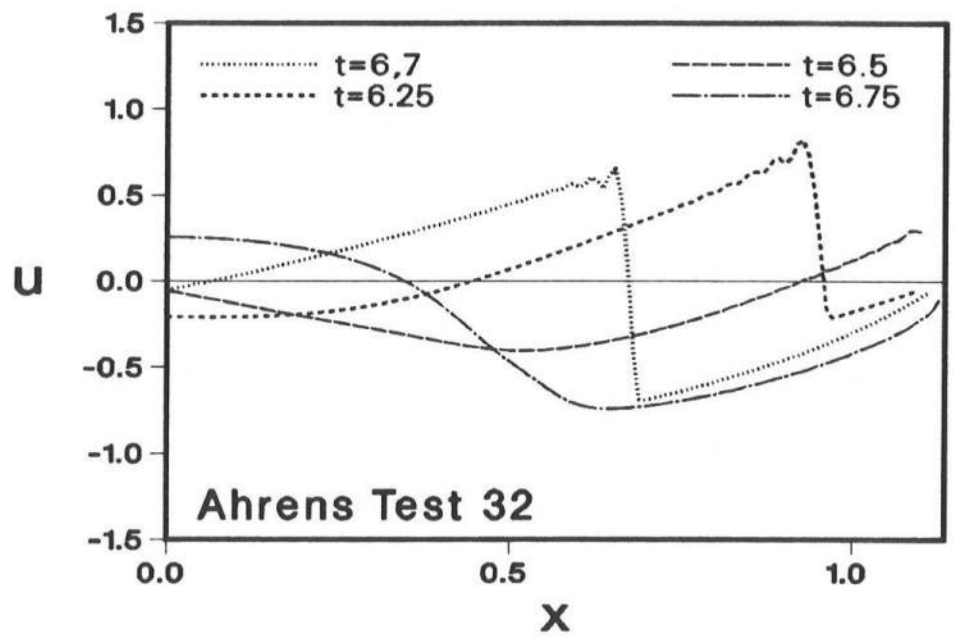


Figure C-9

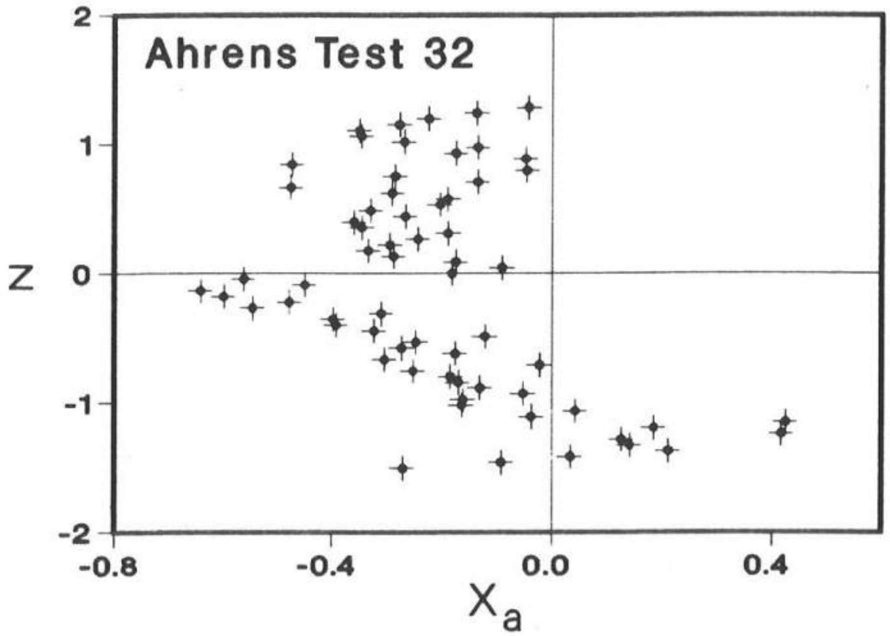


Figure C-10

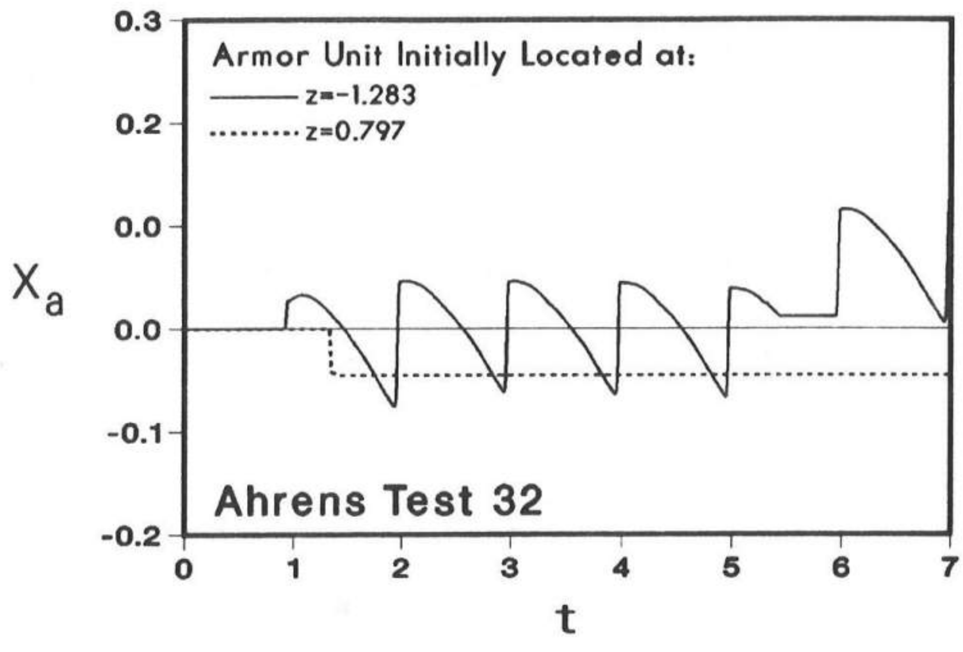


Figure C-11

APPENDIX D: EXAMPLE OF WAVE OVERTOPPING ON SUBAERIAL STRUCTURE

The third example is one test run from the extensive small-scale test data of Saville (1955) for monochromatic wave overtopping over smooth impermeable structures fronted by a 1:10 slope. This test run corresponds to the Run No. 10 in the report of Kobayashi and Wurjanto (1988).

Table D-1 lists the input data for the third example where IJOB=2 for the computation of wave overtopping over a subaerial structure and ISTAB=0 since armor stability and movement are not computed for this smooth impermeable structure. The representative wave height H_r' = HREFP is taken as the deep water wave height H_o' given in the report of Saville (1955). As a result, KSREF = H_r'/H_o' is unity, while KSSEA = H'/H_o' = 1.034 is the shoaling coefficient at the seaward boundary estimated using cnoidal wave theory as discussed in the report of Kobayashi and Wurjanto (1988). It is noted that the friction factor f' = FWP is reduced from 0.3 in Tables B-1 and C-1 for the rough slopes to 0.05 in Table D-1 for the smooth slope.

Table D-2 lists the warning messages written during the time-marching computation in the same way as Tables B-2 and C-2.

Table D-3 lists the concise output for the third example and includes the quantities related to wave overtopping and explained in the common /OVER/ unlike Tables B-3 and C-3.

Fig. D-1 shows the dimensional structure geometry specified as input in Table D-1. The structure consists of three linear segments.

Fig. D-2 shows the temporal variations of the incident wave profile η_i, the reflected wave profile η_r and the total wave profile $\eta_{tot} = (\eta_i + \eta_r)$. The asymmetry of the reflected wave profile in Fig. D-2 appears to be related to wave overtopping since the reflected wave profiles shown in Figs. B-2 and C-2 are more symmetric.

Fig. D-3 shows the temporal variations of the normalized waterline elevation Z_r with i=1, 2 and 3 corresponding to the physical depth $\delta_r' = 1$, 3 and 5 mm. These values of δ_r' for the small-scale test are taken to be smaller than those for the large-scale test where $\delta_r' = 4$,20 and 40mm in Figs. B-3 and C-3. It should be noted that Z_r is taken to be the normalized free surface elevation at the landward edge of the structure when wave overtopping occurs and the normalized water depth h exceeds the value of $\delta_r = \delta_r'/H_r'$. The normalized elevation of the crest of the structure shown in Fig. D-1 corresponds to $Z_r = 1.0$ in Fig. D-3. As a result, the portion of the three curves above $Z_r = 1.0$ in Fig. D-3 indicates the free surface elevation at the landward edge. Fig. D-3 also suggests that a thin layer of water, whose thickness is greater than 1 mm but less than 3 mm, remains on the crest of the structure when the water depth at the landward edge decreases.

Fig. D-4 shows the temporal variations of u and $c = h^{1/2}$ at the landward edge located at $x = x_e$ which have been denoted by V(JE) and C(JE) and stored as a function of the time level N with t = N/NONE in the subroutine DOC2. Fig. D-4 is plotted from the file with its unit number = 32 and its file name = OOVER. Fig. D-4 indicates that wave overtopping starts suddenly as a supercritical flow for which $u > c$ and decreases gradually as a critical flow for which $u = c$.

Fig. D-5 shows the temporal variations of m and h at the landward edge located at $x = x_e$ which have been denoted by U(1,JE) and U(2,JE) and stored together with V(JE) and C(JE) in the subroutine DOC2. Figs. 4 and 5 show that the overtopping flow is highly unsteady. The maximum and time-averaged values of m during $4 \leq t \leq 5$ are 0.2564 and 0.0455, respectively, as listed in Table D-3. The temporal variation of h at $x = x_e$ may be useful for interpreting the temporal variations of Z_r shown in Fig. D-3.

Fig. D-6 shows the temporal variation of u at the seaward boundary located at x = 0 which has been denoted by V(1) and stored as a function of the time level N with t = N/NONE in the subroutine DOC2. Fig. D-6 is plotted from the file with its unit number = 21 and its file name = OSEAWAV. Fig. D-6 is plotted to check the periodicity of u at x=0.

Fig. D-7 shows the spatial variations of the mean water level $\bar{\eta}$, the crest elevation η_{max}, and the trough elevation η_{min}. Formation of the standing wave on the 1:10 slope in front of the structure is apparent in Fig. D-7 whereas this is not apparent in Figs. B-4 and C-4 for which the computation domain has been limited to the region landward of the toe of the structure.

Fig. D-8 shows the spatial variations of the time-averaged velocity $\bar{u}$, the maximum velocity u_{max} and the minimum velocity u_{min}. The maximum and minimum velocity variations on the 1:10 slope appear to be consistent with the standing wave envelope shown in Fig. D-7. Comparison of Fig. D-8 with Figs. B-5 and C-5 indicates that wave overtopping results in unidirectional landward flow on the crest of the structure.

Fig. D-9 shows the spatial variation of the time-averaged volume flux per unit width $\bar{m}$. The time-averaged continuity equation requires $\bar{m}$ = constant. For the case of no wave overtopping, $\bar{m}$=0 as shown in Figs. B-6 and C-6. For the case of wave overtopping shown in Fig. D-9, the computed values of $\bar{m}$ are 0.45 on the average but fluctuates due to the high-frequency numerical oscillations which tend to occur at the locations where the slope is discontinuous. The dimensional structure geometry shown in Fig. D-1 and the normalized structure geometry shown in Fig. D-7 indicate that the slope changes suddenly from 1:10 to 1:3 in the vicinity of x=1.0 where the spatial variation of $\overline{m}$ shown in Fig. D-9 fluctuates greatly. The computed variation of $\overline{m}$ could be smoothed using the smoothing procedure applied by Kobayashi and Wurjanto (1989b,1989c). The smoothed variation of $\overline{m}$ will still yield $\overline{m} \simeq 0.45$. It is noted that the time-averaged quantities are more difficult to compute accurately and more influenced by the numerical high-frequency oscillations than the instantaneous quantities.

Fig. D-10 shows the spatial variations of the time-averaged energy quantities $\overline{E}$, $\overline{E_F}$, $\overline{D_f}$ and $\overline{D_B}$ in the same way as Figs. B-7 and C-7. The numerical high-frequency oscillations are apparent especially for $\overline{D_B}$. This is because the spatial variation of $\overline{D_B}$ computed using Eq. 55 is strongly affected by the apparently small high-frequency oscillations in the spatial variation of $\overline{E_F}$ which would be required to be smoothed before the computation of $\overline{D_B}$ (Kobayashi and Wurjanto, 1989b,1989c).

Fig. D-11 shows the spatial variations of the free surface elevation η at the time t=4, 4.25, 4.5, 4.75 and 5. The numerical high-frequency oscillations are apparent especially in the vicinity of x = 1.0 in Fig. D-11 as compared to Figs. B-8 and C-8.

Fig. D-12 shows the spatial variations of the horizontal velocity u at the time t = 4, 4.25, 4.5, 4.75 and 5. The numerical high-frequency oscillations are apparent in Fig. D-12 as compared to Figs. B-9 and C-9. It is noted that no measurements are available on the detailed spatial and temporal velocity field on a coastal structure. It might be possible that the high-frequency oscillations are related to the actual flow phenomena in the vicinity of the slope discontinuity.

TABLE D-1

```
        3 = number of comment lines
-----------------------------------------------------------------
Saville (1955) Run 10
-----------------------------------------------------------------
20       0                          --> IJOB,ISTAB,NSTAB
1                                   --> ISYST
1                                   --> IBOT
0                                   --> INONCT
1                                   --> IENERG
1  no wave data file                --> IWAVE,FINP2
100   32000   5   0   0             --> ISAVA-B-C,NSAVA,NTIMES,NNOD1,NNOD2
00000    0                          --> IREQ,IELEV,IV,IDUDT,ISNR,NNREQ
   40000                            --> NTOP
    8000                            --> NONE
      40                            --> NJUM1
     120                            --> S
     .050000                        --> FWP
    1.000000      1.000000          --> X1,X2
     .001000                        --> DELTA(normalized)
      3                             --> NDELR
    1.000000                        --> DELRP(1) (mm)
    3.000000                        -->      (2)
    5.000000                        -->      (3)
     .107576      2.621810          --> HREFP(m),TP(seconds)
    1.000000      1.034000          --> KSREF,KSSEA
     .430000                        --> DSEAP(m)
     .333333                        --> TSLOPS
      3                             --> NBSEG
    2.689000       .100000          --> WBSEG(1) (m),TBSLOP(1)
     .807000       .333333          -->      (2)           (2)
     .090000       .000000          -->      (3)           (3)
```

TABLE D-2

```
-----------------------------------------------------------------
Saville (1955) Run 10
-----------------------------------------------------------------

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     121; N =     7536
 Adjusted values: U(2,S) =    0.221E-02; U(2,S-1) =   0.245E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     121; N =     7575
 Adjusted values: U(2,S) =    0.203E-02; U(2,S-1) =   0.225E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     121; N =     7645
 Adjusted values: U(2,S) =    0.187E-02; U(2,S-1) =   0.207E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     122; N =     8367
 Adjusted values: U(2,S) =    0.127E-02; U(2,S-1) =   0.171E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     120; N =     8834
 Adjusted values: U(2,S) =    0.168E-02; U(2,S-1) =   0.175E-02
```

TABLE D-3

```
-----------------------------------------------------------------
Saville (1955) Run 10
-----------------------------------------------------------------

WAVE CONDITION

Cnoidal Incident Wave at Seaward Boundary
1-m =      0.608412027D-01
E   =      0.107076819D+01
K   =      0.281408330D+01
Norm. Maximum Surface Elev.      =     0.682057
Norm. Minimum Surface Elev.      =    -0.351943

Reference Wave Period            =     2.621810 sec.
Reference Wave Height            =     0.107576 meters
Depth at Seaward Boundary        =     0.430000 meters
Shoal. Coef. at Reference  Ks1 =       1.000
             at Seaw. Bdr. Ks2 =       1.034
             Ks = Ks2/Ks1        =     1.034
Norm. Depth at Seaw. Bdr.        =     3.997
Normalized Wave Length           =    12.383
"Sigma"                          =    25.037
Ursell Number                    =    39.665
Surf Similarity Parameter        =     3.329

SLOPE PROPERTIES

Friction Factor              =     0.050000
Norm. Friction Factor        =     0.625919
Norm. Horiz. Length of
      Computation Domain     =     1.325054
Number of Segments           =        3
--------------------------------
 SEGMENT    WBSEG(I)   TBSLOP(I)
    I        meters
--------------------------------
       1    2.689000    0.100000
       2    0.807000    0.333333
       3    0.090000    0.000000

COMPUTATION PARAMETERS

Normalized Delta x    =   0.981521D-02
Normalized Delta t    =   0.125000D-03
Normalized DELTA      =   0.100000E-02
Damping Coeff.     x1 =     1.000
                   x2 =     1.000
Num. Stab. Indicator =    16.180
Total Number of Time Steps         NTOP =     40000
Number of Time Steps in 1 Wave Period
                                   NONE =      8000
Total Number of Spatial Nodes        JE =       136
Number of Nodes Along Bottom Below SWL
                                      S =       120
Storing Temporal Variations at Every
                                  NJUM1 =        40 Time Steps
```

```
REFLECTION COEFFICIENTS

r1 =      0.606
r2 =      0.579
r3 =      0.579

RUNUP, RUNDOWN, SETUP

Largest Node Number Reached by Computational Waterline
                                          JMAX =      136
---------------------------------------------------------
       I    DELTAR(I)     RUNUP(I)    RUNDOWN(I)    SETUP(I)
              [mm]           R            Rd           Zr
---------------------------------------------------------
       1     1.000        1.297       0.130        1.048
       2     3.000        1.297      -0.937        0.423
       3     5.000        1.297      -0.958        0.327

OVERTOPPING

Norm. Avg. Overtopping Rate      =   0.455369D-01
Norm. Avg. Flow at Seaw. Bdr.    =   0.453231D-01
Norm. Max. Overtopping Rate      =   0.256392D+00
Max. Rate Occurs at     0.439000 Within Interval [NSTAT,NTOP]
Overtopping Duration = 1.000000 Within Interval [NSTAT,NTOP]
The last two quantities are relative to the specified
  interval taken as unity

WAVE SET-DOWN OR SETUP

Average value of ETAI =      0.000000
                 ETAR =     -0.005894

TIME-AVERAGED ENERGY BALANCE

Normalized Energy Flux:
. at Seaw. Boundary   A =  0.186835D+00
. at Landw. Boundary  B =  0.698877D-01
Normalized Rate of Energy Dissipation
in the Computation Domain, Due to:
. bottom friction     C =  0.846857D-01
. wave breaking       D =  0.322616D-01
Calculation 1:
                E = A-B =  0.116947D+00
                F = C+D =  0.116947D+00
  Must G=0, but G = F-E =  0.277556D-16
  % error  100G/E =           0.00
Approximate Energy Flux, Based on
Linear Long Wave, Due to:
. incident wave at seaw. boundary      P =  0.267195D+00
. reflected wave at seaw. boundary     Q =  0.895412D-01
Calculation 2:
Net Energy Flux at Seaw. Bndry. S = P-Q =  0.177653D+00
% Error at Seaward Boundary  100(S-A)/A =          -4.91
```

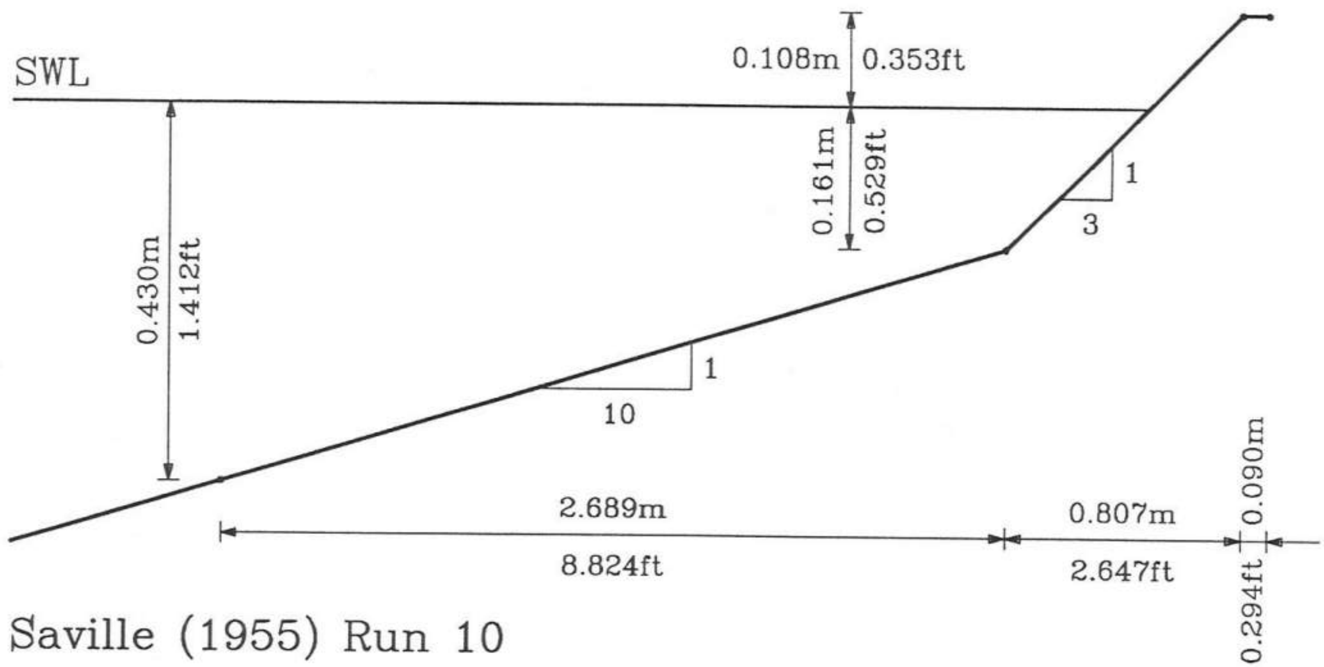

SWL
0.108m
0.353ft
0.161m
0.529ft
0.430m
1.412ft
1
3
1
10
2.689m
8.824ft
0.807m
2.647ft
0.090m
0.294ft
Saville (1955) Run 10

Figure D-1

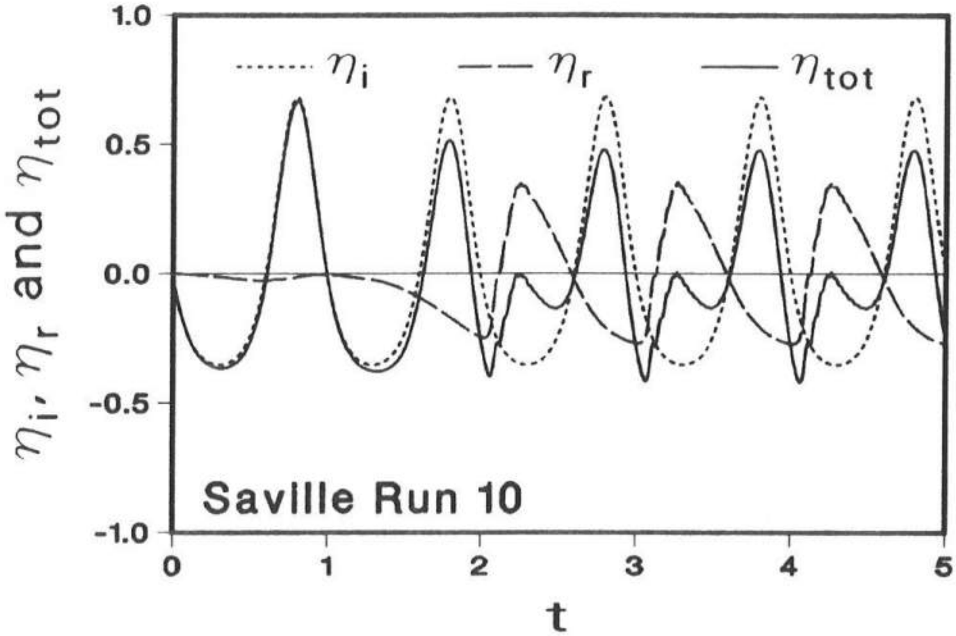


Figure D-2

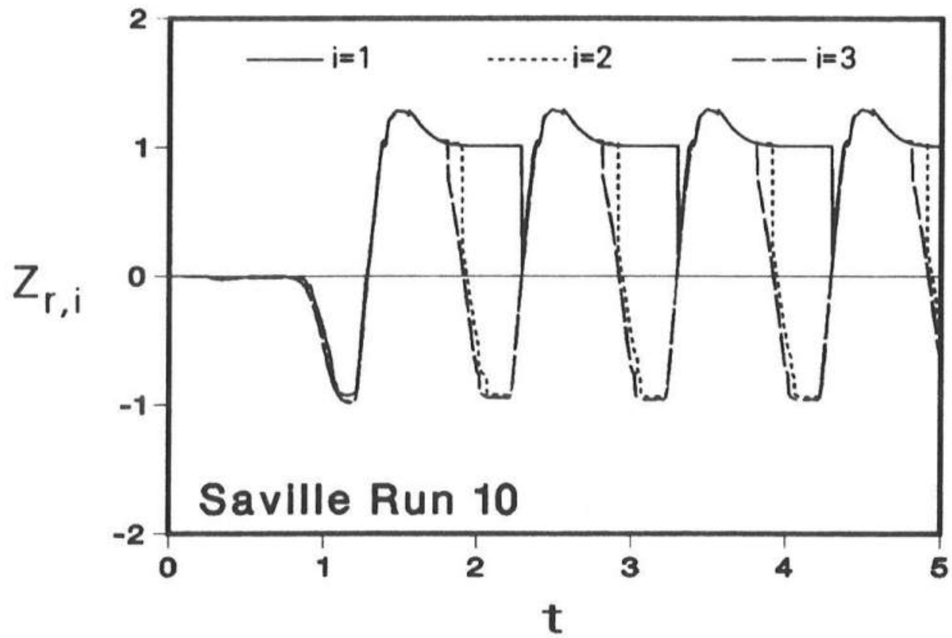


Figure D-3

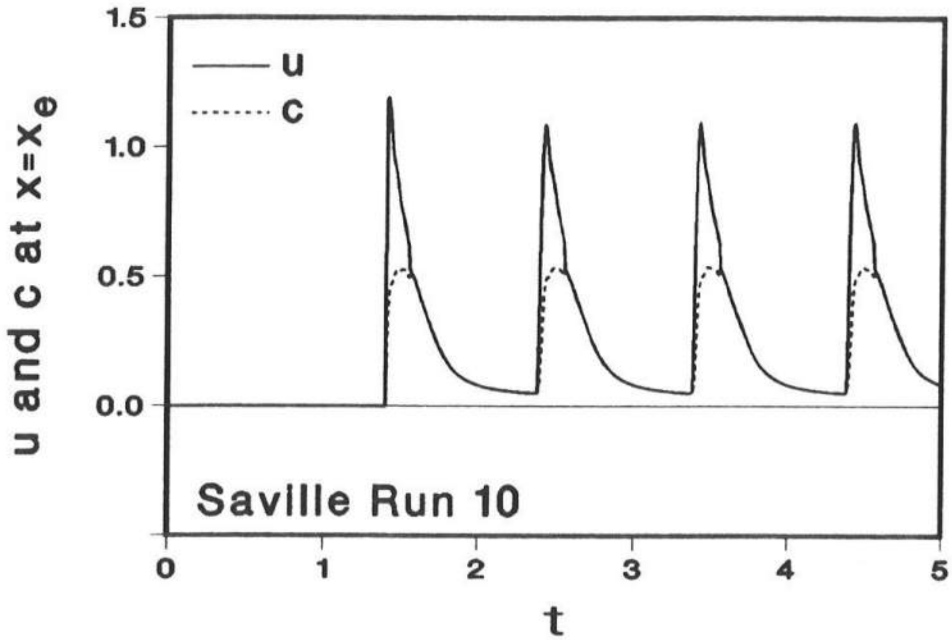


Figure D-4

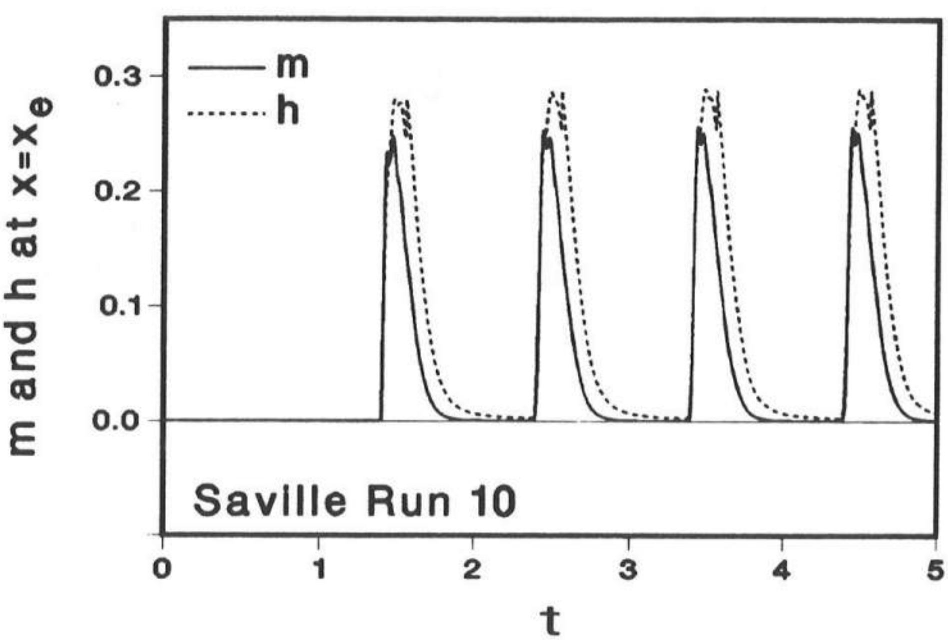


Figure D-5

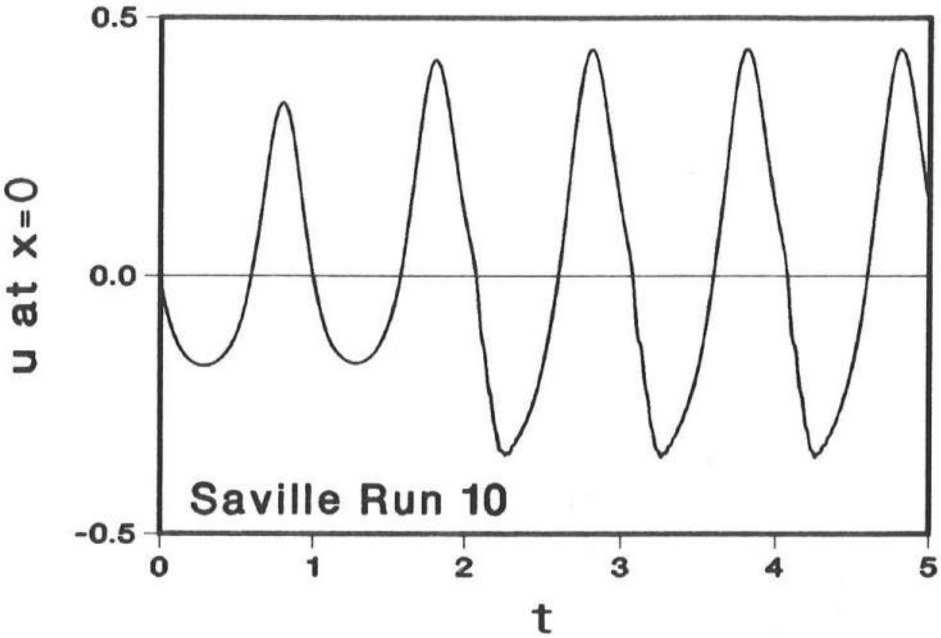


Figure D-6

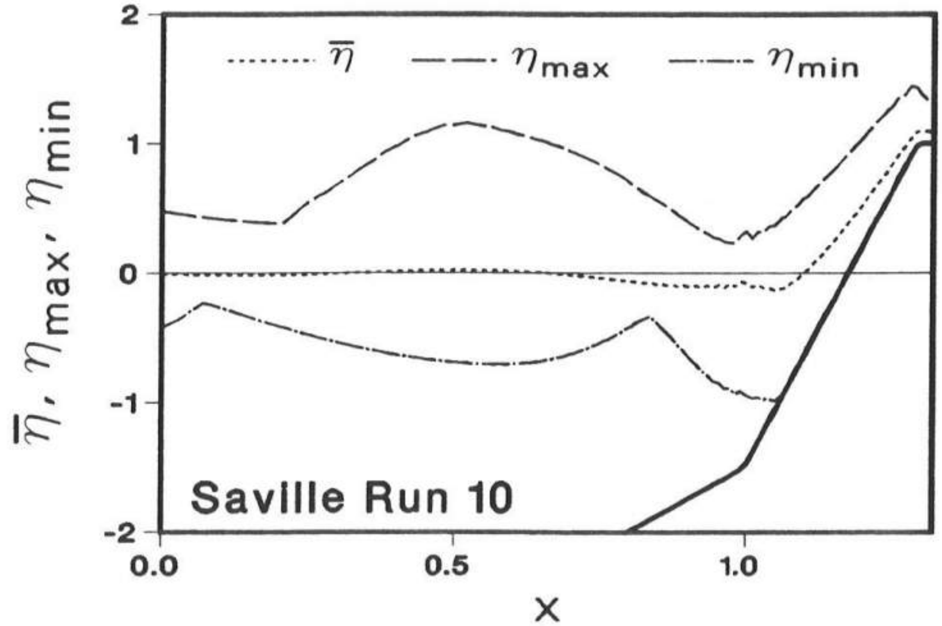


Figure D-7

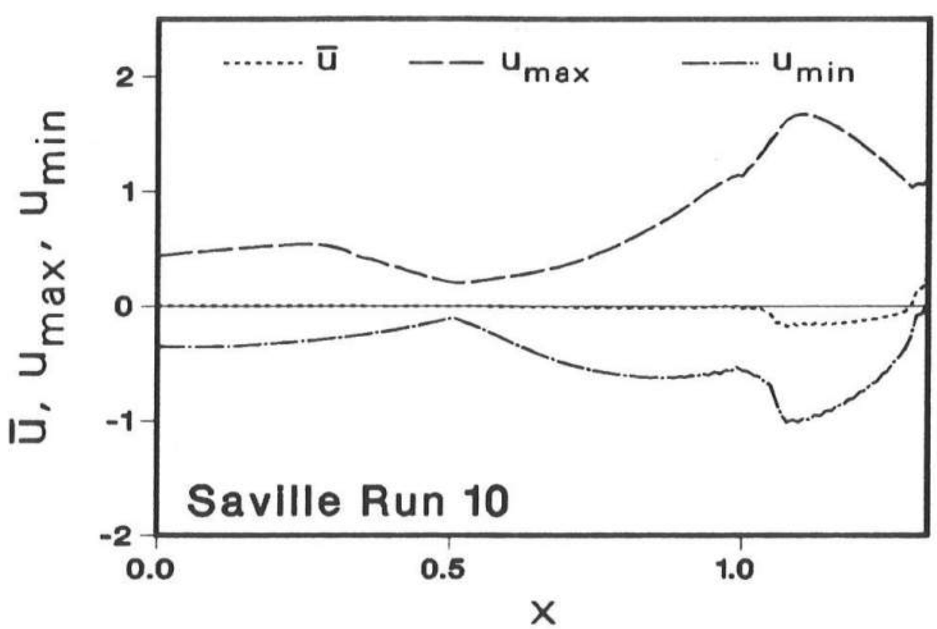


Figure D-8

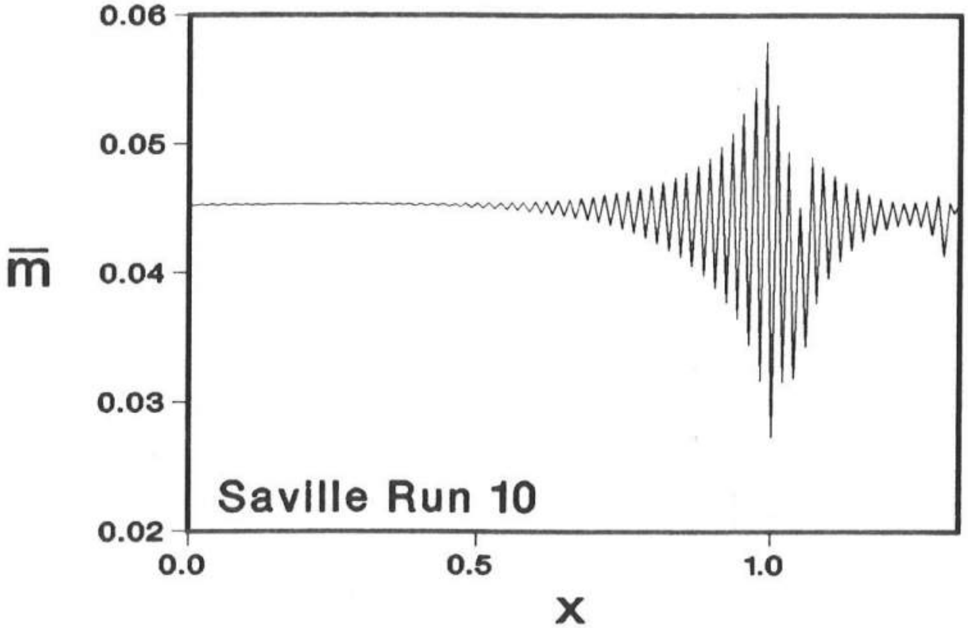


Figure D-9

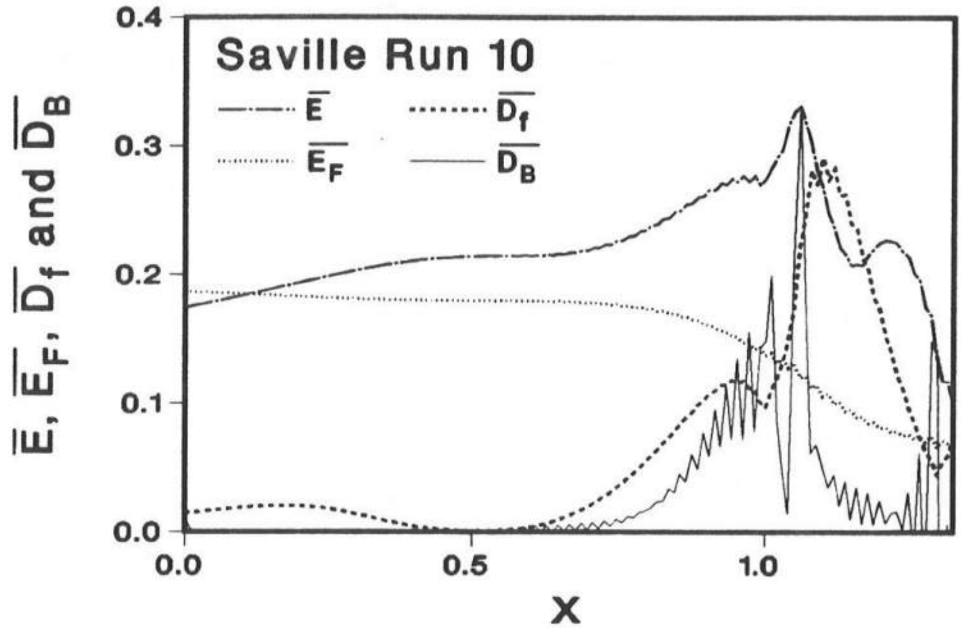


Figure D-10

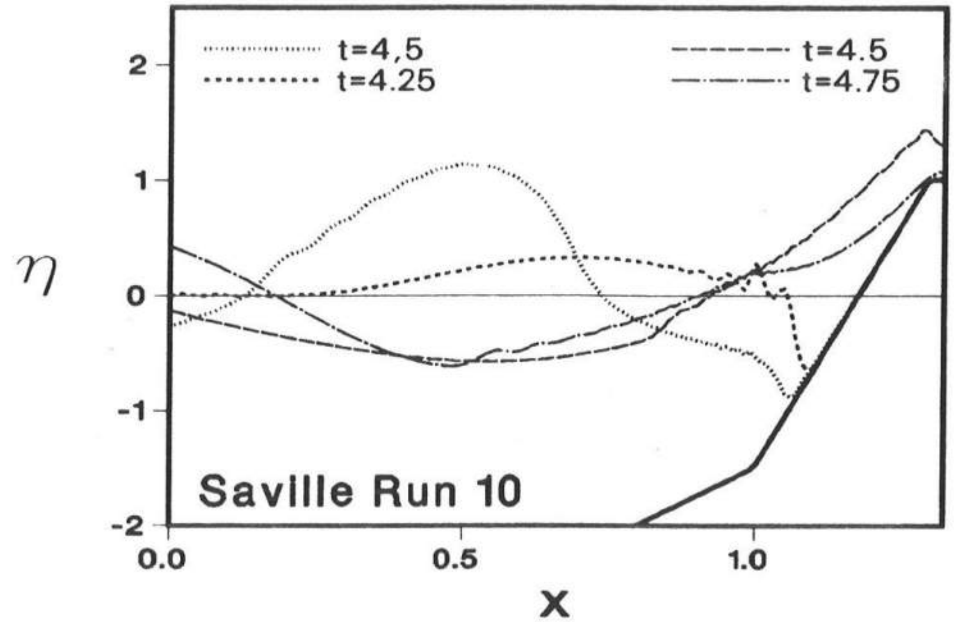


Figure D-11

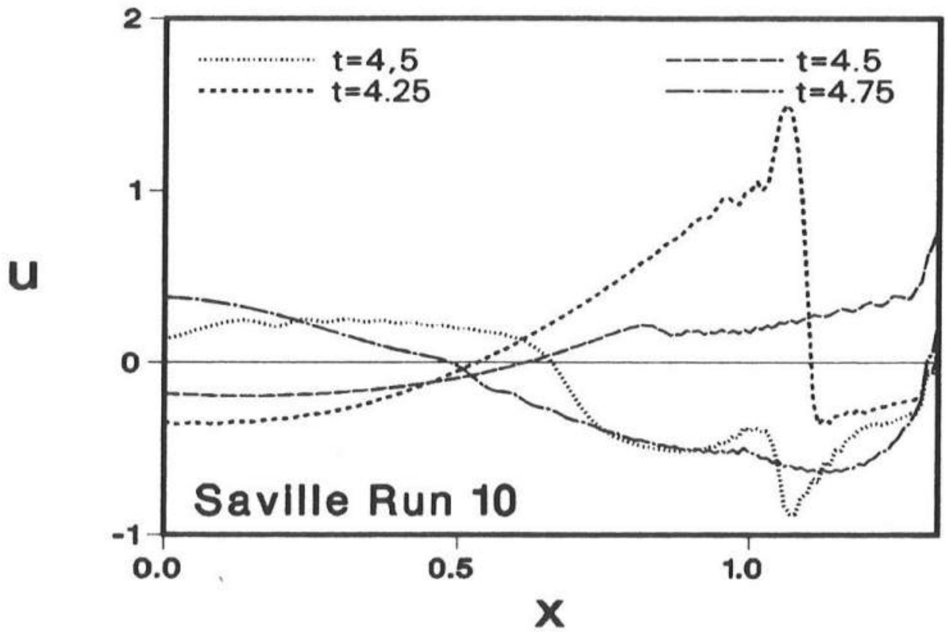


Figure D-12

APPENDIX E: EXAMPLE OF WAVE TRANSMISSION OVER SUBMERGED STRUCTURE

The fourth example is one test run from the small-scale test data of Seelig (1980) for monochromatic wave transmission over a submerged smooth impermeable structure. This test run corresponds to the Run No. 2 in the report of Kobayashi and Wurjanto (1989b).

Table E-1 lists the input data for the fourth example where IJOB=3 for the computation of wave transmission over a submerged breakwater and ISTAB=0 since armor stability and movement are not computed for this smooth impermeable structure. The representative wave height H_r' = HREFP is taken as the wave height H' measured at the seaward toe of the structure. Since H_r' = H', KSSEA = KSREF = 1.0 could also be used in Table E-1 as discussed in relation to Table B-1. The number of nodes used for the computation is S = 300 and greater than those used for the other examples. The damping coefficients X1=2 and X2=2 are specified as input in Table E-1 as compared to X1=1 and X2=1 used for the other examples. These changes of the data input are thought to reduce the numerical high frequency oscillations which tend to occur at the sharp corners of the submerged structure. It is also noted that the input format for the dimensional structure geometry is changed because of the use of IBOT=2 used in Table E-1 instead of IBOT=1 used in Tables B-1, C-1 and D-1. There are no warning messages written during the time-marching computation unlike the other examples as shown in Tables B-2, C-2 and D-2.

Table E-2 lists the concise output for the fourth example and includes the quantities related to wave transmission unlike Tables B-3, C-3 and D-3.

Fig. E-1 shows the dimensional structure geometry consisting of three linear segments specified as input in Table E-1.

Fig. E-2 shows the temporal variations of the incident wave profile η_i, the reflected wave profile η_r, and the total wave profile $\eta_{tot} = (\eta_i + \eta_r)$. The asymmetry of the reflected wave profile shown in Fig. E-2 is similar to that shown in Fig. D-2.

Fig. E-3 shows the temporal variation of the transmitted wave train η_t at the landward toe of the structure which has been denoted by ETAT and stored as a function of the time level N with t = N/NONE in the subroutine DOC2. Fig. E-3 is plotted from the file with its unit number = 33 and its file name = OTRANS. The computed transmitted wave profile exhibits a bore-like profile as discussed by Kobayashi and Wurjanto (1989b).

Fig. E-4 shows the spatial variations of the mean water level $\bar{\eta}$, the crest elevation η_{max}, and the trough elevation η_{min} together with the normalized geometry of the submerged breakwater. The mean water level $\bar{\eta}$ increases landward except in the vicinity of the sharp corners of the submerged structure where the computed results may not be very reliable. The increase of the mean water level from the seaward boundary to the landward boundary is 0.15 as listed in Table E-2.

Fig. E-5 shows the spatial variations of the time-averaged velocity $\bar{u}$, the maximum velocity u_{max} and the minimum velocity u_{min}. Figs. E-4 and E-5 may be compared with Figs. D-7 and D-8 to examine the similarity and difference between wave overtopping over a subaerial structure and wave transmission over a submerged structure.

Fig. E-6 shows the spatial variation of the time averaged volume flux per unit width, $\bar{m}$, which is required to be constant. The oscillations of the computed values of $\bar{m}$ shown in Fig. E-6 are reduced as compared to those shown in Fig. D-9, although the increases of the number of nodes and the damping coefficients do not eliminate this problem. The approximate value of $\overline{m}$ = 0.196 listed in Table E-2 may still be used as the required constant value of $\overline{m}$ caused by wave transmission.

Fig. E-7 shows the spatial variations of the time-averaged energy quantities $\overline{E}$, $\overline{E_F}$ and $\overline{D_f}$ where the spatial variation of $\overline{D_B}$ is not shown because of the numerical high-frequency oscillations similar to those shown in Fig. D-10. The spatial variation of $\overline{D_B}$ computed using the smoothed variation of $\overline{E_F}$ was presented in the report of Kobayashi and Wurjanto (1989b).

Fig. E-8 shows the spatial variations of the free surface elevation η at the time t=4, 4.25, 4.5, 4.75 and 5 in the same way as Fig. D-11.

Fig. E-9 shows the spatial variations of the horizontal velocity u at the time t = 4, 4.25, 4.5, 4.75 and 5 in the same way as Fig. D-12.

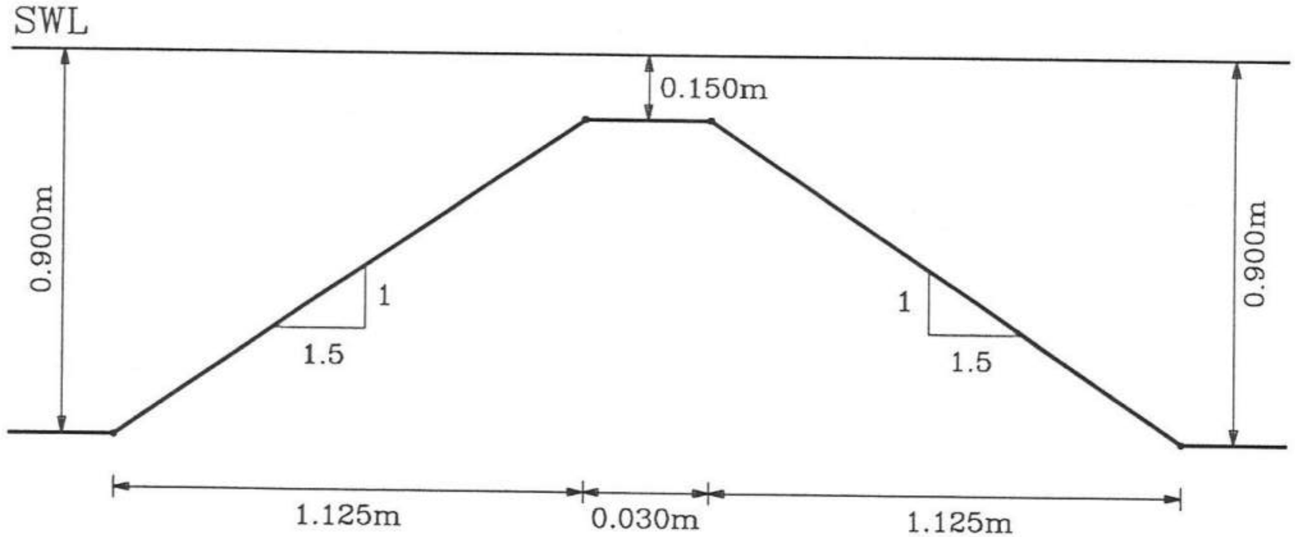
TABLE E-1

```
       3 = number of comment lines
-----------------------------------------------------------------
Seelig (1980) Run 2
-----------------------------------------------------------------
30       0                         --> IJOB,ISTAB,NSTAB
1                                  --> ISYST
2                                  --> IBOT
0                                  --> INONCT
1                                  --> IENERG
1  no wave data file               --> IWAVE,FINP2
100   12000    5    0    0         --> ISAVA-B-C,NSAVA,NTIMES,NNOD1,NNOD2
00000     0                        --> IREQ,IELEV,IV,IDUDT,ISNR,NNREQ
   15000                           --> NTOP
    3000                           --> NONE
      15                           --> NJUM1
     300                           --> S
      .050000                      --> FWP
     2.000000        2.000000      --> X1,X2
      .001000                      --> DELTA(normalized)
      0                            --> NDELR
      .193000        2.390000      --> HREFP(m),TP(seconds)
      .931989         .931989      --> KSREF,KSSEA
      .900000                      --> DSEAP(m)
      .666667                      --> TSLOPS
      3                            --> NBSEG
      .000000         .900000      --> XBSEG(1)(m),ZBSEG(1)(m)
     1.125000         .150000      -->       (2)          (2)
     1.425000         .150000      -->       (3)          (3)
     2.550000         .900000      -->       (4)          (4)
```

TABLE E-2

```
----------------------------------------------------------------
Seelig (1980) Run 2
----------------------------------------------------------------

WAVE CONDITION

Stokes II Incident Wave at Seaward Boundary
Norm. Maximum Surface Elev.       =      0.582547
Norm. Minimum Surface Elev.       =     -0.417453

Reference Wave Period             =      2.390000 sec.
Reference Wave Height             =      0.193000 meters
Depth at Seaward Boundary         =      0.900000 meters
Shoal. Coef. at Reference  Ks1 =      0.932
             at Seaw. Bdr. Ks2 =      0.932
             Ks = Ks2/Ks1       =      1.000
Norm. Depth at Seaw. Bdr.         =      4.663
Normalized Wave Length            =      7.053
"Sigma"                           =     17.039
Ursell Number                     =     10.669
Surf Similarity Parameter         =      4.532
Depth at Landward Boundary        =      0.900000 meters

SLOPE PROPERTIES

Friction Factor               =     0.050000
Norm. Friction Factor         =     0.425984
Norm. Horiz. Length of
      Computation Domain      =     0.775406
Number of Segments            =        3
--------------------------------
 SEGMENT     XBSEG(I)    ZBSEG(I)
    I         meters      meters
--------------------------------
    1       0.000000    0.900000
    2       1.125000    0.150000
    3       1.425000    0.150000
    4       2.550000    0.900000

COMPUTATION PARAMETERS

Normalized Delta x    =   0.259333D-02
Normalized Delta t    =   0.333333D-03
Normalized DELTA      =   0.100000E-02
Damping Coeff.    x1 =     2.000
                  x2 =     2.000
Num. Stab. Indicator =     1.020
Total Number of Time Steps        NTOP =      15000
Number of Time Steps in 1 Wave Period
                                  NONE =       3000
Total Number of Spatial Nodes       JE =        300
Storing Temporal Variations at Every
                                 NJUM1 =         15 Time Steps
```

```
REFLECTION COEFFICIENTS

r1 =      0.494
r2 =      0.550
r3 =      0.504

WAVE SET-DOWN OR SETUP

Average value of ETAI =      0.000000
                 ETAR =     -0.077559
                 ETAT =      0.073361
MWL Difference        =      0.150920

TRANSMISSION

Transmission Coefficient T1 =      0.741
                         T2 =      0.722
                         T3 =      0.691
Norm. Avg. Flow at Seaw. Bdr.  =      0.196176
Norm. Avg. Flow at Landw. Bdr. =      0.196442
Average of the Above Two       =      0.196309

TIME-AVERAGED ENERGY BALANCE

Normalized Energy Flux:
. at Seaw. Boundary   A =  0.219144D+00
. at Landw. Boundary  B =  0.163349D+00
Normalized Rate of Energy Dissipation
in the Computation Domain, Due to:
. bottom friction     C =  0.141936D-01
. wave breaking       D =  0.416010D-01
Calculation 1:
                E = A-B =  0.557946D-01
                F = C+D =  0.557946D-01
  Must G=0, but G = F-E = -0.138778D-16
   % error  100G/E =          0.00
Approximate Energy Flux, Based on
Linear Long Wave, Due to:
. incident wave at seaw. boundary     P =  0.269931D+00
. reflected wave at seaw. boundary    Q =  0.685893D-01
. transmitted wave at landw. bndry.   R =  0.129051D+00
Calculation 2:
Net Energy Flux at Seaw. Bndry. S = P-Q =  0.201342D+00
% Error at Seaward Boundary  100(S-A)/A =          -8.12
% Error at Landward Boundary 100(R-B)/B =         -21.00
```



Seelig (1980) Run 2

Figure E-1

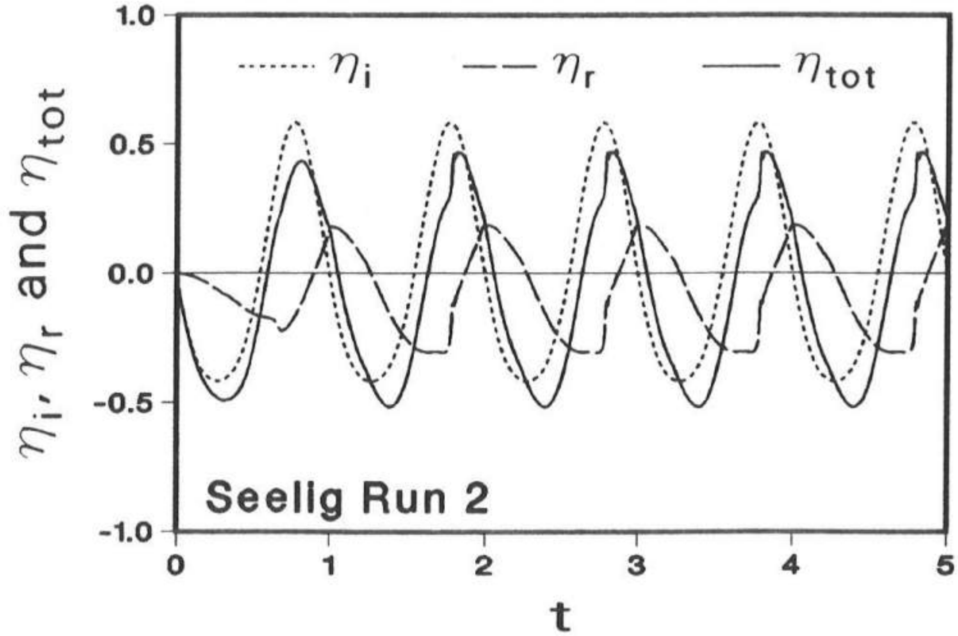


Figure E-2

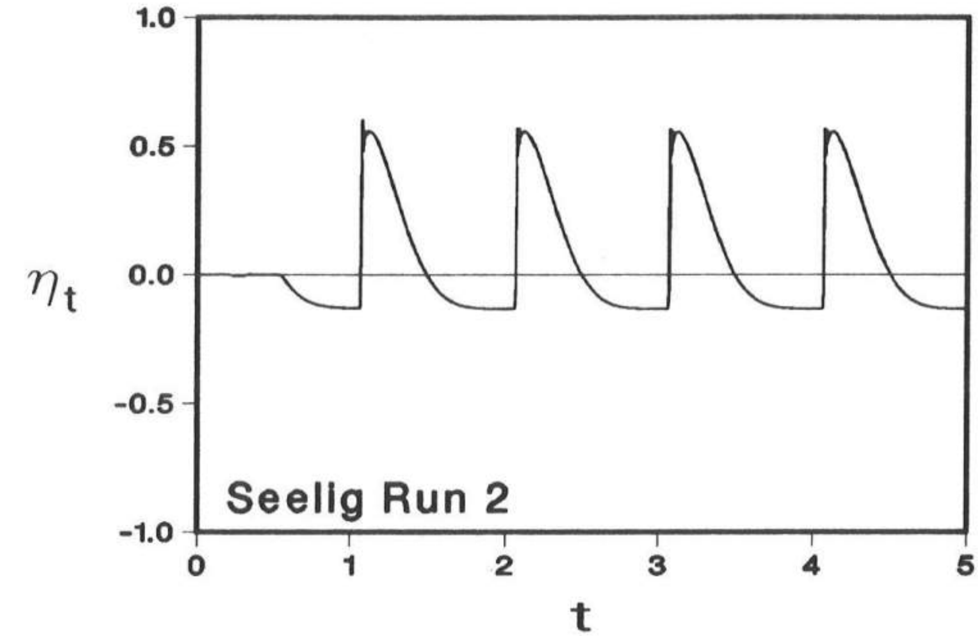


Figure E-3

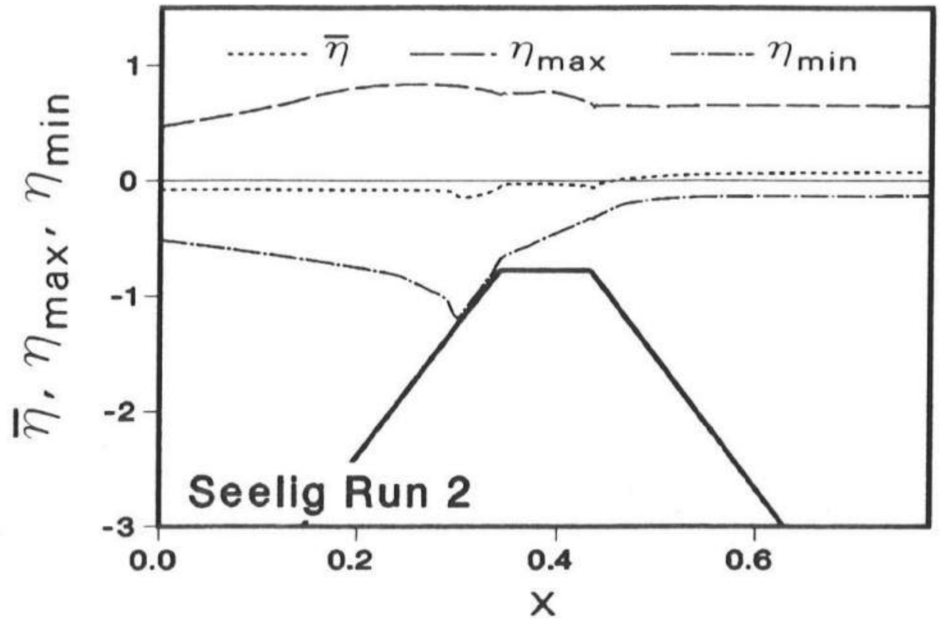


Figure E-4

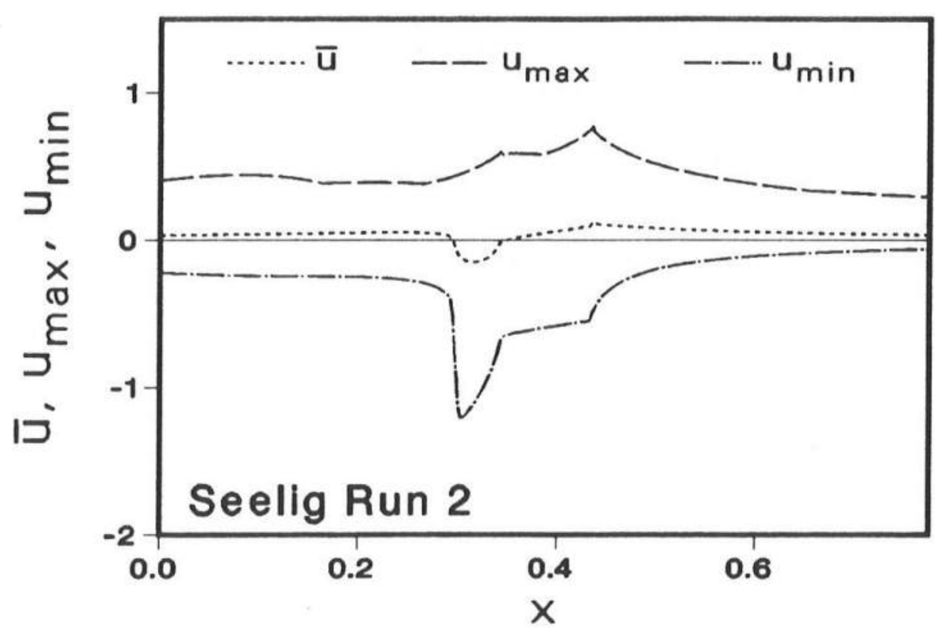


Figure E-5

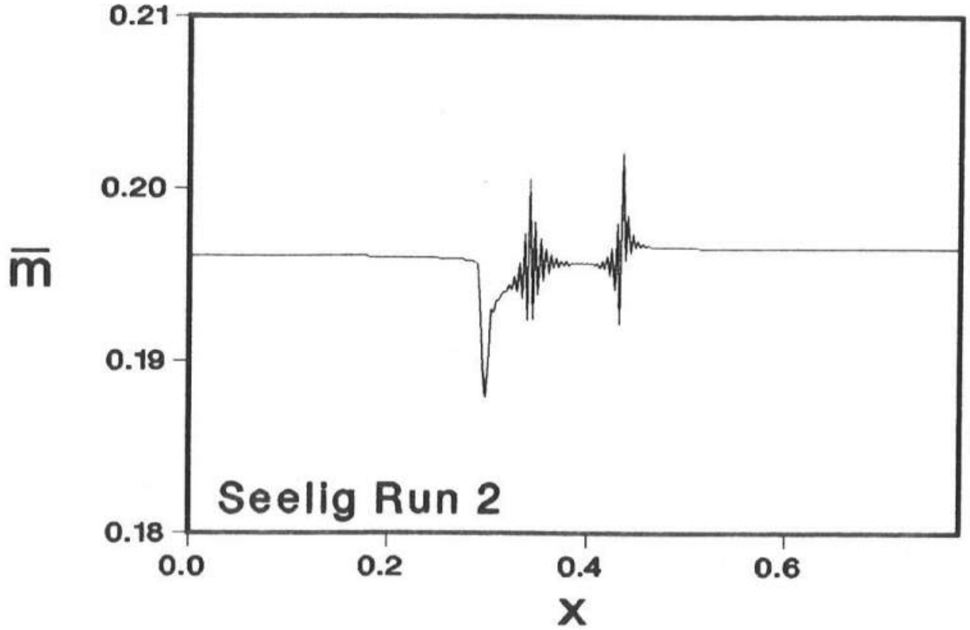


Figure E-6

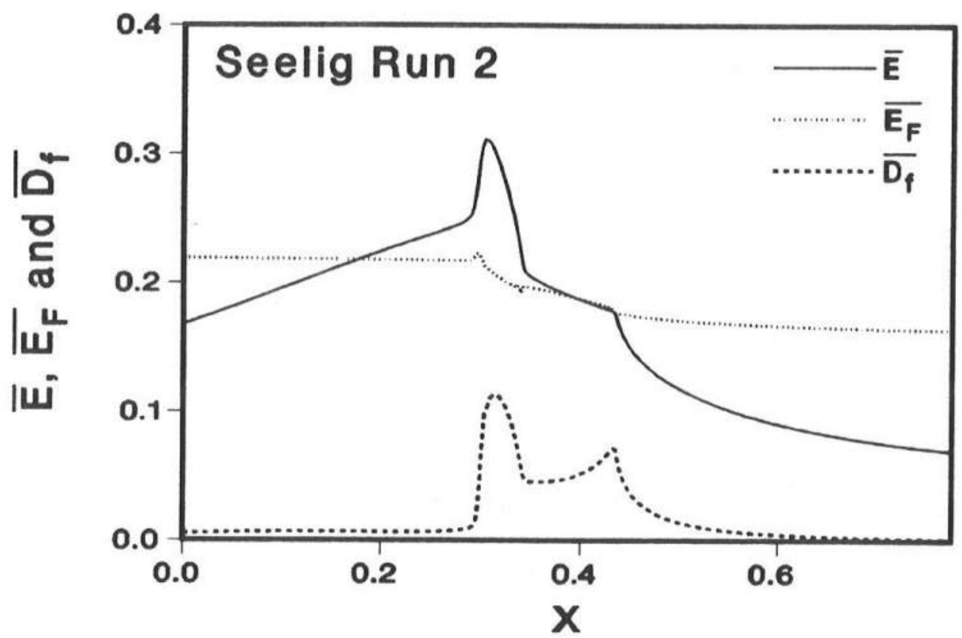


Figure E-7

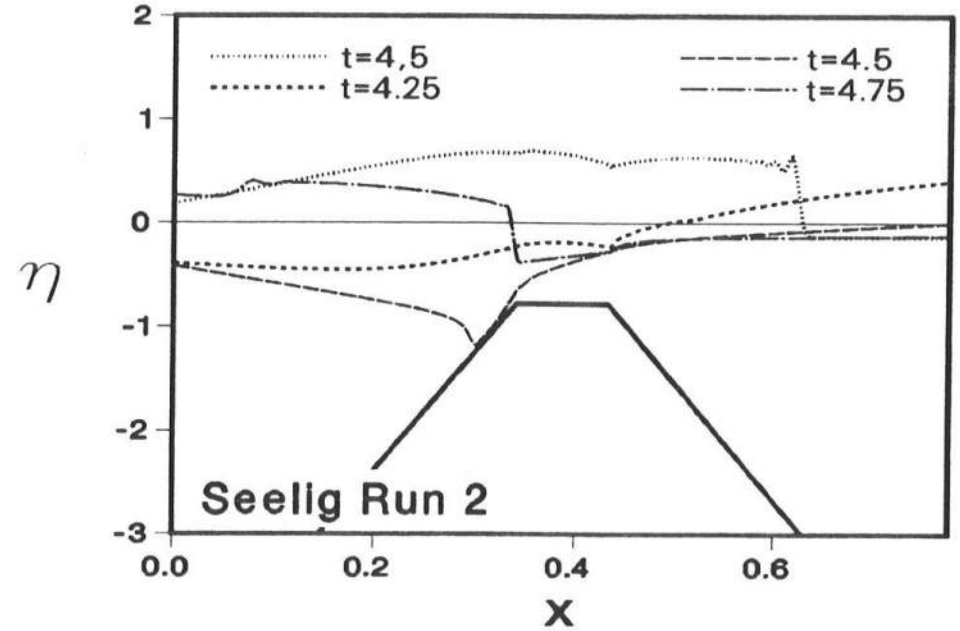


Figure E-8

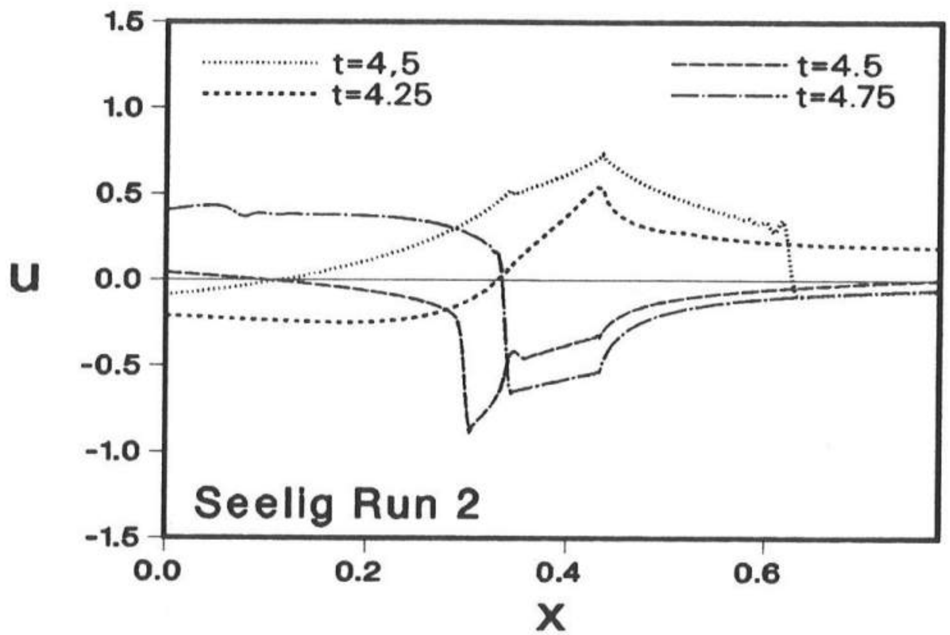


Figure E-9