DOCUMENTATION OF COMPUTER PROGRAM FOR

PREDICTING LONG WAVE RUNUP

BY

NOBUHISA KOBAYASHI AND ENTIN A. KARJADI

Sponsored by
National Science Foundation

RESEARCH REPORT NO. CACR-93-03
March 1993

CENTER FOR APPLIED COASTAL RESEARCH
OCEAN ENGINEERING LABORATORY
UNIVERSITY OF DELAWARE
NEWARK, DELAWARE
19716

ABSTRACT

A computer program called SBREAK has been developed by expanding the computer program IBREAK which was developed in 1989 for the design of rough or smooth impermeable coastal structures of arbitrary geometry against normally incident monochromatic waves as well as for predicting the wave transformation in the surf and swash zones on impermeable beaches. An additional option has been provided in SBREAK to allow the specification of an incident solitary wave as input at the seaward boundary of the computation domain. SBREAK computes the reflected wave train at the seaward boundary for a specified incident wave train. For a subaerial structure or beach, SBREAK computes wave runup on its seaward slope or wave overtopping over the crest of the structure or beach if it is not high enough to prevent flow over its crest. For a submerged structure or nearshore bar, SBREAK computes the transmitted wave train at the landward boundary of the computation domain. In addition to the conservation equations of mass and momentum used to compute the one-dimensional, time-dependent flow field, an equation of energy is used to estimate the rates of energy dissipation due to wave breaking and bottom friction. Moreover, SBREAK 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.

First, the related numerical models published before SBREAK are introduced to provide an overall perspective. Second, the equations and numerical procedures used in SBREAK are summarized concisely. Third, the essential parts of SBREAK are explained to facilitate the effective use of

SBREAK. The computer program SBREAK 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 so that users may be able to modify SBREAK if necessary. The input parameters and variables together with various options are detailed so as to reduce input errors. The output parameters and variables are also explained in detail so that users may be able to make the best use of the output of SBREAK.

Finally, SBREAK is calibrated and evaluated using available data on breaking or broken solitary wave runup on smooth uniform slopes. For an efficient comparison of SBREAK with a large number of tests, the dimensionless parameters involved in the problem are identified using the normalized incident solitary wave profile and governing equations. The representative solitary wave period and associated surf similarity parameter are introduced so as to examine the similarity and difference between solitary and monochromatic (regular) waves on smooth uniform slopes. The breaking, runup and reflection of solitary and monochromatic waves are qualitatively similar in terms of the surf similarity parameter. For given surf similarity parameter, breaking solitary wave runup is definitely larger than breaking monochromatic wave runup affected by the interaction between wave uprush and downrush on the slope. The present numerical model is shown to be in good agreement with the data of Synolakis (1987a) on breaking or broken solitary wave runup with a limited calibration of the bottom friction factor employed in SBREAK. Solitary wave overtopping and transmission could also be predicted using SBREAK, although only solitary wave runup is examined in this report.

ACKNOWLEDGEMENT

This report is a result of research sponsored by the National Science Foundation under Grant No. BCS 9111827. The authors would like to thank Dr. Clifford J. Astill for his support of our efforts to document this computer program.

TABLE OF CONTENTS

DOCUMENTATION OF COMPUTER PROGRAM FOR PREDICTING LONG WAVE RUNUP

1. INTRODUCTION

1.1 Background

Kobayashi and Wurjanto (1989c) developed a computer program called IBREAK for the design of rough or smooth impermeable coastal structures of arbitrary geometry against normally incident monochromatic and transient waves. The previous work based on IBREAK is summarized in Section 1.2. In order to make successful computations for incident random waves of long durations in an efficient manner, Wurjanto and Kobayashi (1991) developed a computer program called RBREAK by expanding IBREAK with an automated adjustment procedure of the time step size for the explicit finite difference method used in IBREAK. The previous work based on RBREAK is summarized in Section 1.3. Furthermore, Wurjanto and Kobayashi (1992) developed a computer program called PBREAK by extending RBREAK to simulate the flow inside a permeable underlayer of arbitrary thickness as well as the flow above a rough permeable slope of arbitrary geometry. The previous work related to PBREAK is explained in Section 1.4. These numerical models have been calibrated and evaluated for monochromatic and random waves only. Consequently, it is not certain whether these models can also be applied to predict the runup of very long waves such as tsunamis on beaches.

A computer program called SBREAK is presented in this report by expanding IBREAK to predict the runup of solitary waves on beaches. SBREAK is documented in such detail that other researchers will be able to apply or modify SBREAK for their research projects. Examples are presented on the basis of comparison between SBREAK and the laboratory data on solitary wave runup presented by Synolakis (1987a, 1987b) and Synolakis and Skjelbreia (1993).

1.2 Previous Work Based on IBREAK

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, 1975, 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 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 could 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 (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.

Kobayashi and Wurjanto (1989b) 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 used 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.

Kobayashi and Wurjanto (1989e) showed that IBREAK could be calibrated and applied to predict the hydrodynamic forces and sliding motions of dolos units at the Crescent City Breakwater in California. The calibrated numerical model was used to hindcast the response of the dolos units during a storm. The hindcast results were shown to be consistent with the measured results including the upslope movement of poorly interlocked dolos units and the importance of the static and wave forces with negligible impact forces. The numerical model was then used to predict the response of the poorly and well interlocked dolos units under extreme wave conditions. The predicted results have suggested that the wave forces acting on these dolos units may possibly exceed the static forces, while the poorly interlocked dolos units may move considerably, resulting in possible impact forces.

Other researchers (e.g., Allsop et al. 1988; Thompson 1988; Van der Meer and Breteler 1990; Losada et al. 1992) applied the numerical model IBREAK or a similar numerical model to predict the monochromatic wave motion on coastal structures and evaluated the accuracy of the numerical models using their own laboratory measurements. As a whole, their results are consistent with our experiences with IBREAK for the last several years. The numerical model IBREAK is fairly versatile and robust although it is not as accurate as careful hydraulic model tests.

1.3 Previous Work Based on RBREAK

IBREAK was initially used to simulate irregular waves on the slope of a coastal structure since any incident wave train can be specified as input to IBREAK at the seaward boundary of the computation domain. However, the irregular waterline oscillation on the slope was found to cause numerical difficulties and stoppage during the computation of a sufficient duration for a stationary random sea. The constant time step size Δt for the explicit finite difference method used in IBREAK was reduced to overcome the numerical difficulties. This increased the computation time considerably but did not always work.

To avoid the unnecessary increase in the computation time, it was decided to vary the time step size Δt such that smaller values of Δt should be used for portions of the computation with numerical difficulties. Since the portions with numerical difficulties are not known in advance, the time-marching computation needs to be reversed to an earlier time level before the initiation of the current numerical difficulty and then resumed from the reversed time level using a smaller value of Δt. To reduce the computation time, the value of Δt needs to be increased after overcoming the current numerical difficulty. This adjustment procedure was automated. The computer program RBREAK was hence an expanded version of IBREAK with the automated adjustment procedure which is really essential for making successful computations for incident random waves of sufficient durations in an efficient manner.

Kobayashi, Cox and Wurjanto (1990) conducted three irregular wave test runs to obtain detailed data on irregular wave reflection and run-up on a 1:3 rough impermeable slope. The test results were also used to evaluate the capabilities and limitations of RBREAK for predicting the time series and spectral characteristics of the reflected wave and waterline oscillations on the slope. The numerical model was shown to predict the measured time series and spectra reasonably well, including the selective nature of wave reflection and dissipation as well as the appearance of low-frequency components in the waterline oscillations on the 1:3 slope.

Kobayashi and Wurjanto (1992) derived the one-dimensional equations of mass, momentum, and energy from the two-dimensional continuity and Reynolds equations in order to elucidate the approximations involved in the one-dimensional equations employed in the numerical model. The numerical model RBREAK based on these equations was then compared qualitatively with the set-up and swash statistics on a moderately steep beach with a nearshore bar. The numerical model was shown to predict the essential features of the irregular wave transformation and swash oscillation on the barred beach. The computed set-up and swash heights were found to follow the lower bound of the scattered data points partly because of the neglect of low frequency components in the specified incident wave train. A more quantitative comparison was also made with the spectrum of the shoreline oscillation measured on a 1:20 plane beach for which the corresponding wave spectrum was given. RBREAK was shown to predict the dominant low frequency components of the measured spectrum fairly well.

1.4 Previous Work Related to PBREAK

Kobayashi and Wurjanto (1990) developed a numerical model to predict the flow and armor response on a rough permeable slope as well as the flow in a thin permeable underlayer for a normally-incident wave train. This model based on the assumption of a thin permeable underlayer neglected the region landward of the waterline on the rough slope and the inertia terms in the horizontal momentum equation for the flow in the thin permeable underlayer. Computation was made for six test runs to examine the accuracy and capability of the numerical model for simulating the fairly detailed hydrodynamics and armor response under the action of regular waves. The computed critical stability number for initiation of armor movement was compared with the measured stability number corresponding to the start of the damage under irregular wave action to quantify the limitations of the regular wave approximation. The computed wave runup, run-down, and reflection coefficients were shown to be in qualitative agreement with available empirical formulas based on regular wave tests. Kobayashi and Wurjanto (1989d) applied the developed numerical model to hypothetical permeable slopes corresponding to available impermeable slope tests. The computed results with and without a permeable underlayer indicated that the permeability effects would increase the hydraulic stability of armor units noticeably and decrease wave runup and reflection slightly. The computed results were qualitatively consistent with available data although they were not extensive and limited to regular waves only.

Kobayashi, Wurjanto and Cox (1990a) applied the developed numerical model to compute the irregular wave motion on a rough permeable slope. The normally-incident irregular wave train characterized by its spectral density at the toe of the slope was generated numerically for six test runs. The computed critical stability number for initiation of armor movement under the computed irregular wave motion was shown to be in fair agreement with the measured stability number corresponding to the start of the damage (Van der Meer 1988). The comparison of the computed armor stability for the incident regular and irregular waves indicated that the armor stability would be reduced appreciably and vary less along the slope under the irregular wave action. On the other hand, the comparison between the computed reflected wave spectrum and the specified incident wave spectrum indicated the reflection of Fourier components with longer period and the dissipation of Fourier components with shorter periods, while the average reflection coefficient increased with the increase of the surf similarity parameter. The computed waterline oscillations were examined using spectral and time series analyses. The computed spectra of the waterline oscillations showed noticeable low-frequency components, which increased with the decrease of the surf similarity parameter. The statistical analysis of individual wave runup heights indicated that the computed runup distribution followed the Rayleigh distribution fairly well for some of the six test runs. The computed maximum wave runup was in agreement with the empirical formula based on irregular wave runup tests.

Furthermore, Kobayashi, Wurjanto and Cox (1990b) analyzed the computed results for the six test runs to examine the critical incident wave profile associated with the minimum rock stability for each run. The minimum rock stability computed for the runs with dominant plunging waves on gentle slopes was caused by the large wave with the maximum crest elevation during its uprush on the slope. The minimum rock stability computed for the runs with dominant surging waves on steeper slopes was caused by the downrushing water with high velocities resulted from a large zero-upcrossing wave with a high crest followed by a deep trough. These computed results may eventually allow one to quantify incident design wave conditions more specifically than the simple approach based on the representative wave height and period. In addition, a simplified model was proposed to predict the eroded area due to the movement and dislodgement of rock units using the probability of armor movement computed by the numerical model. This model was shown to be in qualitative agreement with the empirical formula for the damage level proposed by Van der Meer (1987, 1988).

The numerical model based on the assumption of a thin permeable underlayer was found to be inapplicable to three test runs conducted for a 1:3 rough permeable slope with a thick permeable underlayer (Kobayashi, Cox and Wurjanto 1991). The computed results did not satisfy the time-averaged equation of mass conservation mainly because the earlier model did not account for water storage in the region landward of the waterline on the slope. These three test runs corresponded to the three test runs for the 1:3 rough impermeable slope conducted by Kobayashi, Cox and Wurjanto (1990) except for the presence of the thick permeable underlayer.

Wurjanto and Kobayashi (1993) developed a one-dimensional, time-dependent numerical model to simulate the flow over a rough permeable slope as well as the flow inside the permeable underlayer of arbitrary thickness for specified normally-incident irregular waves. The derivation of the one-dimensional continuity, momentum and energy equations employed in the numerical model was presented to clarify the basic assumptions made in these equations. The comparison of the numerical model with the three test runs conducted by Kobayashi, Cox and Wurjanto (1991) has shown that the numerical model can predict the time series and spectral characteristics of the reflected waves and waterline oscillations on the 1:3 rough slope with the thick permeable underlayer. The computed results for the three runs have indicated that the wave propagation, attenuation and setup inside the permeable underlayer reduce the intensity of wave breaking and resulting energy dissipation on the slope but increase the energy influx and dissipation inside the thick permeable underlayer. The permeability effects also result in the time-averaged landward and seaward mass fluxes above and inside the permeable underlayer, respectively. Furthermore, Kobayashi and Wurjanto (1993) compared the computed results for the rough permeable and impermeable slopes to quantify the differences caused by the thick permeable underlayer.

1.5 Scope and Outline

The numerical model SBREAK is based on the one-dimensional finite-amplitude shallow-water equations including the effects of bottom friction. SBREAK is suited for predicting nonlinear long wave runup on an impermeable slope of arbitrary geometry and roughness characteristics. Since various options included in IBREAK may be useful to users of SBREAK, these options are kept in SBREAK. The specification of an incident solitary wave as input to SBREAK is added herein to compute solitary wave runup and reflection. The automated adjustment procedure of RBREAK is omitted herein since this procedure is lengthy and may not be necessary for solitary waves of relatively short durations. The permeable underlayer included in PBREAK is not considered herein partly because of the compactness of the resulting computer program and partly because the permeability effects are likely to be less important than the roughness effects for tsunamis.

The numerical model is explained in Section 2 together with the options for specifying the three types of incident waves as well as for computing wave reflection, runup, overtopping, transmission, energy dissipation, and armor stability and movement. The computer program SBREAK is documented in Section 3. SBREAK 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 since a large number of parameters and variables are involved in SBREAK. The input parameters and variables together with various options are detailed so as to minimize input errors. The output parameters and variables are also explained in detail since the proper interpretation of the computed results is essential. SBREAK is compared with available data on solitary wave runup on a smooth uniform slope in Section 4. An example of the input and output is also provided. The compared results are elucidated to evaluate the capabilities and limitations of SBREAK for predicting solitary wave runup as well as to quantify differences between solitary and monochromatic wave runup.

2. NUMERICAL MODEL

2.1 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 finite-amplitude shallow-water waves over the gentle impermeable slope, the vertically-integrated equations for mass and x'-momentum may be expressed as (Kobayashi et al., 1987; Kobayashi and Wurjanto, 1992).

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

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

where t' = time; g = gravitational acceleration; and f' = friction factor related to the shear stress acting on the slope. The friction factor f' is assumed constant, although it could be varied spatially. In this simplified analysis, the constant friction factor f' accounts for the roughness characteristics of the impermeable slope surface. The range of f'

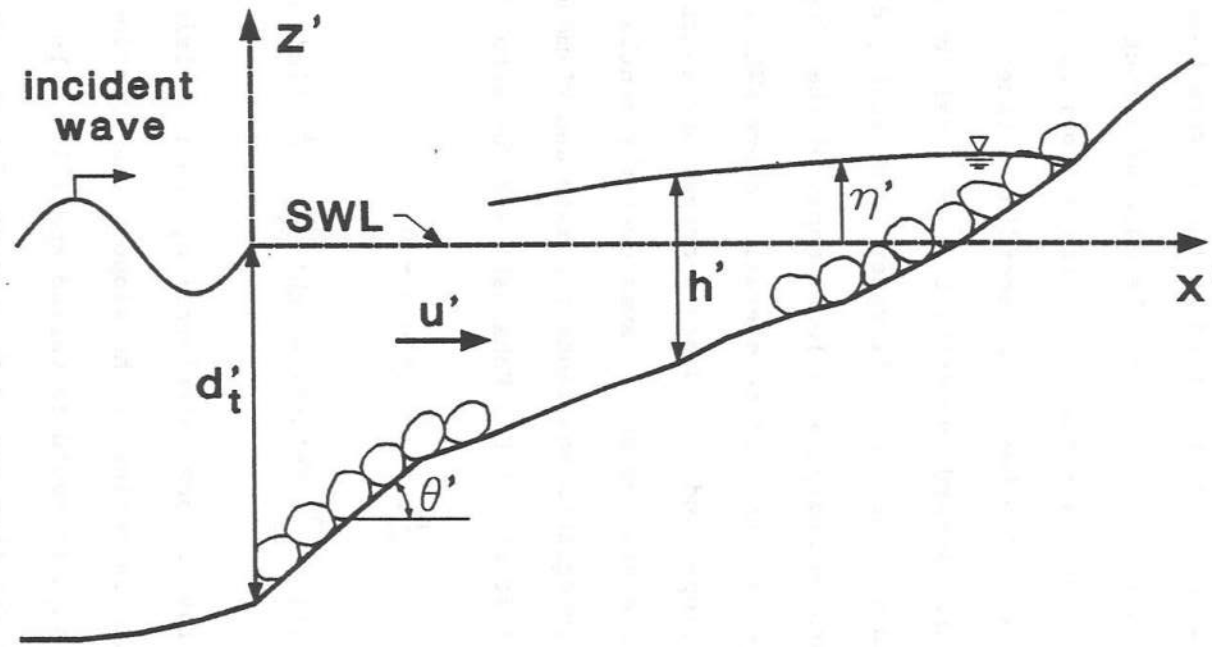

FIGURE 1. Wave Runup on a Rough Slope.

used in the previous applications was $f' = 0.05$-0.3 for rough slopes and $f' = 0.01$-0.05 for smooth slopes. The computed results except for wave runup and overtopping were not sensitive to f'. Relatedly, the theoretical bed level for the flow over the rough impermeable slope is difficult to pinpoint as is the case with oscillatory rough turbulent boundary layers (Jonsson, 1980).

The following dimensionless variables and parameters based on the assumption of finite-amplitude shallow-water waves 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 expressing the ratio between the characteristic horizontal and vertical length scales; θ = dimensionless gradient of the slope; and f = normalized friction factor. The present numerical model assumes that $\sigma^2 \gg 1$ and $(\cot \theta')^2 \gg 1$ in the computation domain (Kobayashi and Wurjanto, 1992). 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, $\xi = \theta/(2\pi)^{1/2}$ (Battjes, 1974). 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 $x \geq 0$. 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

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

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.

2.2 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

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

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 U_1^* and U_j^* for $j \geq (s-1)$ are computed using the seaward and landward boundary conditions, respectively. The values of U_j^* for $j=2,3,\ldots,(s-2)$ are computed using the known values of U_{j-1}, U_j and U_{j+1} at the time $t=(n-1)\Delta t$ (Kobayashi et al., 1987)

$$U_j^* = U_j - \lambda\left[\frac{1}{2}(F_{j+1} - F_{j-1}) + \Delta x G_j \right] + \frac{\lambda^2}{2} (g_j - g_{j-1} - \Delta x\ S_j) + D_j \tag{17}$$

where $\lambda = \Delta t/\Delta x$. The vector g_j in Eq. 17 is given by

$$g_j = \frac{1}{2} \left(A_{j+1} + A_j\right) \left[F_{j+1} - F_j + \frac{\Delta x}{2} \left(G_{j+1} + G_j\right)\right] \tag{18}$$

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 $\mathbf{D}_j$ 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

$$\mathbf{D}_j = \frac{\lambda}{2}\left[\mathbf{Q}_j\left(\mathbf{U}_{j+1} - \mathbf{U}_j\right) - \mathbf{Q}_{j-1}\left(\mathbf{U}_j - \mathbf{U}_{j-1}\right)\right] \tag{22}$$

with

$$\mathbf{Q}_j = p_j\ \mathbf{I} + \frac{1}{2}\ q_j\left(\mathbf{A}_j + \mathbf{A}_{j+1}\right) \tag{23}$$

where $\mathbf{I}$ = 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 values of $\epsilon_1 = \epsilon_2 = 1$ or $\epsilon_1 = \epsilon_2 = 2$ have been used in previous computations. The increase of ϵ_1 *and* ϵ_2 tends to improve numerical stability with negligible effects on computed results (Kobayashi and Wurjanto, 1992).

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

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

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 Section 4 where the numerical model is compared with data on solitary wave runup.

2.3 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 $t = 0$.

The computer program SBREAK provides three options using the integer IWAVE specified as input. For IWAVE = 1, an incident monochromatic wave train is computed using Stokes second-order or cnoidal wave theory depending on the value of the Ursell parameter. For IWAVE = 2, the incident wave profile $\eta_i(t)$ is read from a file. This option has been used for the cases where the incident wave profiles were measured in a wave flume or generated numerically using separate subroutines developed for random waves (Cox et al., 1991). For IWAVE = 3, an incident solitary wave train is computed as explained in Section 2.3.2. The computed results presented in this report are limited to solitary waves with IWAVE = 3.

2.3.1 Monochromatic waves

For the case of IWAVE = 1, the computer program SBREAK 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, 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., Shore Protection Manual, 1984), the normalized incident wave profile $\eta_i(t)$ at x = 0 is given by

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

with

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

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

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 $0 \le (t+t_0) \le 0.5$. 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 $U_r \ge 26$, cnoidal wave theory (e.g., Svendsen and Brink-Kjaer, 1972) is used to compute the normalized incident wave profile $\eta_i(t)$ at $x = 0$

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

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).

2.3.2 Solitary wave

For the case of IWAVE = 3, the computer program SBREAK uses solitary wave theory to specify the normalized incident wave profile $\eta_i(t)$ at $x = 0$. Solitary wave theory corresponds to cnoidal wave theory as the Ursell parameter U_r approaches infinity (e.g., Dean and Dalrymple, 1984). As U_r

approaches infinity, the cnoidal wave parameters m and E approach unity, while the parameter K approaches infinity. Substitution of m = 1 and E/K = 0 into Eq. 32 yields $\eta_{min} = 0$. Moreover, for m = 1 and E/K = 0, Eq. 34 with L being given by Eq. 33 results in $2K = K_2$ where K_2 is given by

$$K_2 = \frac{\sqrt{3}}{2} \frac{\sigma}{d_t} \left(K_s + \frac{K_s^2}{d_t}\right)^{1/2} \tag{35}$$

Correspondingly, the normalized solitary wave profile may be expressed as

$$\eta_i(t) = K_s \operatorname{sech}^2\left[K_2(t - t_c)\right] \quad \text{for } t \geq 0 \tag{36}$$

where t_c = normalized arrival time of the solitary wave crest such that $\eta_i = K_s$ at $t = t_c$. It is noted that Eqs. 35 and 36 can also be derived by normalizing the dimensional solitary wave profile using Eqs. 3-5 together with $K_s = H'/H'_r$ and $t_c = t'_c/T'_r$.

In order to compute $\eta_i(t)$ as a function of $t \geq 0$ using Eqs. 35 and 36, the parameter t_c and the reference wave period T'_r included in $\sigma = T'_r(g/H'_r)^{1/2}$ for the solitary wave need to be specified. Since the time t' is normalized as $t = t'/T'_r$, the unit duration $(t_c - 0.5) \leq t \leq (t_c + 0.5)$ about the crest arrival time t_c may be selected such that $\eta_i \geq \delta_i$ in this unit duration where δ_i needs to be very small and is given by

$$\delta_i = K_s \operatorname{sech}^2 \left(\frac{K_2}{2}\right) \tag{37}$$

which can be rewritten as

$$K_2 = 2 \ln \left(\sqrt{\frac{K_s}{\delta_i}} + \sqrt{\frac{K_s}{\delta_i} - 1} \right) \tag{38}$$

In the computer program SBREAK, a small value of $\delta_i = 0.05$ is specified on the basis of the sensitivity analysis performed in Section 4.1. Then, Eq. 38

yields the value of K_2 for given K_s. Eq. 35 is rearranged to compute the value of σ

$$\sigma = \frac{2d_t}{\sqrt{3}} \left(K_s + \frac{K_s^2}{d_t} \right)^{-1/2} K_2 \tag{39}$$

The reference wave period $T_r' = \sigma(H_r'/g)^{1/2}$ can thus be estimated for the specified small value of δ_i. On the other hand, the initial value of η_i at t=0 is given by

$$\eta_i \ (t=0) = K_s \ \text{sech}^2 \ (K_2 \ t_c) \tag{40}$$

Since the initial conditions for the computation are taken to be $\eta=0$ at t=0 in the region $x \geq 0$, the value of t_c needs to be selected such that $\eta_i(t=0)$ is essentially zero. In the computer program SBREAK, $t_c = 1$ is specified on the basis of the sensitivity analysis performed in Section 4.1 as well as to make it easier to interpret the computed temporal variations relative to the crest arrival time $t_c=1$.

2.4 Wave Reflection

The normalized reflected wave train $\eta_r(t)$ in Eq. 12 at the seaward boundary is computed using Eq. 13. It is also required to find the unknown value of the vector $\dot{U}_1^*$ at x = 0 at 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 $t = n\Delta t$

$$\beta_1^* = \beta_1 - \frac{\Delta t}{\Delta x} (u_1 - c_1)(\beta_2 - \beta_1) + \Delta t \theta_1 \tag{41}$$

where $\beta_1 = (-u_1 + 2c_1)$ and $\beta_2 = (-u_2 + 2c_2)$. The right hand side of Eq. 41 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., 1989)

$$C_t = K_s^2 \, (16 d_t)^{-1} \qquad \text{for gentle slopes} \tag{42}$$

where the assumptions of linear long wave and negligible wave reflection were made in Eq. 14 to derive Eq. 42. 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. 42 as a rough guideline.

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

$$r_1 = \left[(\eta_r)_{max} - (\eta_r)_{min} \right] K_s^{-1} \tag{43}$$

$$r_2 = \left[\overline{\eta_r^2} \left(\overline{\eta_i^2} \right)^{-1} \right]^{1/2} \tag{44}$$

$$r_3 = \left[\overline{(\eta_r - \overline{\eta_r})^2} \left(\overline{\eta_i^2} \right)^{-1} \right]^{1/2} \tag{45}$$

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. 43 is based on the normalized height of the reflected wave train as compared to that of the incident wave train. Eqs. 44 and 45 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. 45 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. 43 and 44 yield $r_1 = r_2$. It may be noted that Eqs. 44 and 45 may also be used for incident random waves if the irregular variations of $\eta_i(t)$ and $\eta_r(t)$ for a long duration are used to compute the time-averaged values. For an incident solitary wave, the temporal variations of $\eta_i(t)$ and $\eta_r(t)$ should be compared, although Eqs. 43-45 may still be used to estimate the degree of wave reflection over a specified duration qualitatively.

2.5 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. This case corresponds to the integer IJOB=1 specified as input in the computer program SBREAK. 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 where δ = 0.001-0.003 has been used and the increase of δ tends to improve numerical stability near the moving waterline.

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 $h_s^* \leq \delta$, 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 U_s^{**} at the time $t=(n+1)\Delta t$ using Eq. 17 without the damping term where U_j^* and U_j in Eq. 17 are replaced by U_s^{**} and 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:

$$m_{s+1}^* = m_{s-1}^* - \frac{\Delta x}{\Delta t} (h_s^{**} - h_s) \tag{46}$$

$$u_{s+1}^* = u_{s-1}^* - (u_s^*)^{-1} \left[\frac{\Delta x}{\Delta t} (u_s^{**} - u_s) + h_{s+1}^* - h_{s-1}^* + 2\Delta x \theta_s\right] \tag{47}$$

The upper limit of the absolute value of $(u_s^*)^{-1}$ in Eq. 47 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.

2.6 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. This case corresponds to the integer IJOB=2 specified as input. If wave overtopping occurs, the computation domain for the numerical model is limited to the region $0 \le x \le x_e$

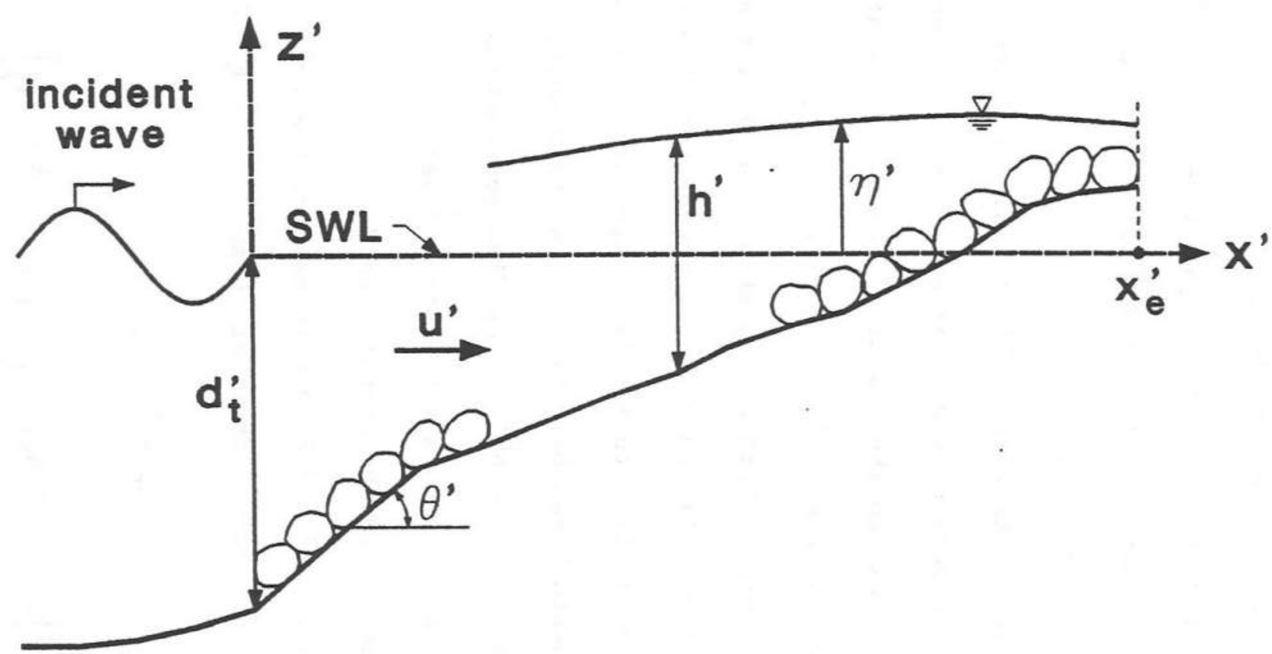

FIGURE 2. Wave Overtopping over a Subaerial Structure.

where $x_e = x_e'/[T_r'(gH_r')^{1/2}]$ and the dimensionless variables defined in Eqs. 3-5 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 $f = 0$:

$$h_s^* = h_s - \frac{\Delta t}{\Delta x} (m_s - m_{s-1}) \tag{48}$$

$$m_s^* = m_s - \frac{\Delta t}{\Delta x} \left[\left(m_s u_s + \frac{1}{2} h_s^2\right) - \left(m_{s-1} u_{s-1} + \frac{1}{2} h_{s-1}^2\right)\right] - \Delta t \theta_s h_s \tag{49}$$

   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 $f = 0$:

$$\alpha_s^* = \alpha_s - \frac{\Delta t}{\Delta x} (u_s + c_s) (\alpha_s - \alpha_{s-1}) - \Delta t \theta_s \tag{50}$$

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 $h_s^* \le \delta$, 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 $x = x_e$.

$$Q = Q'/[H_r'(gH_r')^{1/2}] = \overline{m} \qquad \text{at } x = x_e \tag{51}$$

where the overbar indicates the time averaging of $m(t,x)$ at $x = x_e$ after its periodicity is established. Eq. 51 does not include the volume flux during the interval when $h \le \delta$ 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. 51 can also be used to predict the value of Q for incident random waves if the temporal variation of m at $x = x_e$ is averaged over a long duration. For an incident solitary wave, the temporal variation of m at $x=x_e$ should be examined, although Eq. 51 may still be used to estimate the degree of wave overtopping over a specified duration qualitatively.

2.7 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. This case corresponds to the integer IJOB=3 specified as input. 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 and specify IJOB=1 as input 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)

$$h = d_e + \eta_t(t) \qquad \text{at } x = x_e \tag{52}$$

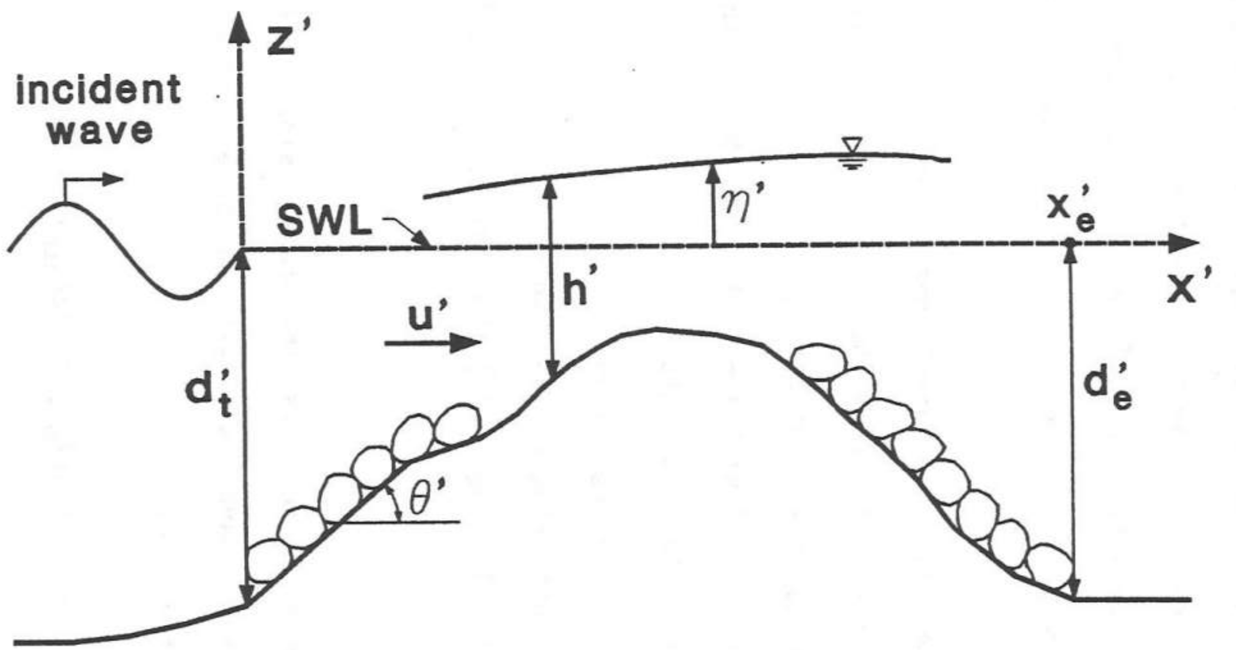

FIGURE 3. Wave Transmission over a Submerged Structure.

$$\eta_t(t) = \frac{1}{2} d_e^{1/2} \alpha(t) - d_e \qquad \text{at } x = x_e \tag{53}$$

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. 53 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. 53.

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. 50.
3. Compute η_t^* at $t = n\Delta t$ using Eq. 53 with $\alpha = \alpha_s^*$. Then, $h_s^* = (d_e + \eta_t^*)$ from Eq. 52, 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 $d_e = d_t$.

$$T_1 = [(\eta_t)_{max} - (\eta_t)_{min}] K_s^{-1} \tag{54}$$

$$T_2 = \left[\overline{\eta_t^2} \left(\overline{\eta_i^2} \right)^{-1} \right]^{1/2} \tag{55}$$

$$T_3 = \left[\overline{(\eta_t - \overline{\eta_t})^2} \left(\overline{\eta_i^2}\right)^{-1} \right]^{1/2} \tag{56}$$

Eqs. 54, 55 and 56 correspond to Eqs. 43, 44 and 45, respectively. Eq. 54 is based on the normalized height of the transmitted wave train as compared to that of the incident wave train. Eqs. 55 and 56 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. 56 accounts for the difference $\overline{\eta_t}$ between the still water level and the mean water level at $x = x_e$. For incident random and solitary waves, the computed temporal variation of $\eta_t(t)$ should be examined, although Eqs. 54-56 may still be used to estimate the degree of wave transmission over a specified duration.

2.8 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, 1992)

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x} (E_F) = - D_f - D_B \tag{57}$$

with

$$E = \frac{1}{2} (hu^2 + \eta^2) \qquad \text{for } h > \eta \tag{58a}$$

$$E = \frac{1}{2} [hu^2 + \eta^2 - (h-\eta)^2] \qquad \text{for } h < \eta \tag{58b}$$

$$E_F = uh \left(\frac{u^2}{2} + \eta\right) \tag{59}$$

$$D_f = f|u|u^2 \tag{60}$$

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. 58a and 58b 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. 57

$$\overline{D_B} = -\frac{d}{dx}(\overline{E_F}) - \overline{D_f} \tag{61}$$

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. 60 is positive. It should be noted that Eq. 61 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. 61 from the seaward boundary to the landward boundary yields the time-averaged energy equation for the region $0 \leq x \leq x_e$

$$\overline{E_F}\,(x=0) - \overline{E_F}\,(x = x_e) = \int_0^{x_e} (\overline{D_f} + \overline{D_B})\, dx \tag{62}$$

where the first and second terms on the left hand side of Eq. 62 are the values of $\overline{E_F}$ at $x = 0$ and $x = x_e$, respectively. Eq. 62 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. 62 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)

$$\overline{E} \simeq \overline{\eta_i^2} + \overline{(\eta_r - \overline{\eta_r})^2} \qquad \text{at} \quad x = 0 \tag{63}$$

$$\overline{E_F} \simeq d_t^{1/2} \left[\overline{\eta_i^2} - \overline{(\eta_r - \overline{\eta_r})^2} \right] \qquad \text{at} \quad x = 0 \tag{64}$$

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. 45 including the effect of $\overline{\eta_r}$ is based on Eqs. 63 and 64. The reflection coefficient r_2 given by Eq. 44 corresponds to Eqs. 63 and 64 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

$$\overline{E} \simeq \overline{(\eta_t - \overline{\eta_t})^2} \qquad \text{at} \quad x = x_e \tag{65}$$

$$\overline{E_F} \simeq d_e^{1/2} \overline{(\eta_t - \overline{\eta_t})^2} \qquad \text{at} \quad x = x_e \tag{66}$$

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. 56 for the case of $d_e = d_t$ is based on Eqs. 65 and 66, whereas the transmission coefficient T_2 given by Eq. 55 does not include the wave setup $\overline{\eta_t}$ at $x = x_e$.

2.9 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

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -\frac{\partial h}{\partial x} - \theta - \frac{f|u|u}{h} \tag{67}$$

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:

$$|N_S + E_1| + E_2 N_S \leq E_3 \tag{68}$$

which is applicable for the case of $u \neq 0$. The stability number N_S in Eq. 68 is defined as

$$N_S = H_r' (s_g - 1)^{-1} \left(\frac{W'}{\rho s_g}\right)^{-1/3} \tag{69}$$

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. 69 for given H_r'. E_1, E_2 and E_3 in Eq. 68 are defined as

$$E_1 = \frac{2C_3^{2/3}}{C_2 C_D |u| u} \left[\frac{C_M}{(s_g - 1)\sigma} \frac{du}{dt} - \sin\theta'\right] \tag{70}$$

$$E_2 = C_L \tan\phi / C_D \tag{71}$$

$$E_3 = \frac{2C_3^{2/3}}{C_2 C_D u^2} \cos\theta' \tan\phi \tag{72}$$

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. 68 can be solved in terms of N_s

$$N_s \le N_R(t,x) = (E_1 + E_3)/(E_2 - 1); \quad \text{if } E_1 < 0,\ E_2 > 1 \text{ and } E_3 < (-E_1 E_2) \tag{73a}$$

$$N_s \le N_R(t,x) = (E_3 - E_1)/(E_2 + 1); \quad \text{otherwise} \tag{73b}$$

where N_R = dimensionless function expressing the degree of the armor unit stability as a function of t and x. For the computation, Eqs. 73a and 73b are used if $|u| \ge 10^{-3}$ 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

$$\left| \frac{C_M}{(s_g-1)\sigma} \frac{du}{dt} - \sin\theta' \right| \le \cos\theta' \tan\phi \tag{74}$$

The condition given by Eq. 74 is satisfied if the normalized fluid acceleration remain within the following lower and upper bounds

$$\sigma\, a_{min} \le du/dt \le \sigma\, a_{max} \tag{75}$$

with

$$a_{min} \ge -\frac{s_g-1}{C_M} \frac{\sin(\phi-\theta')}{\cos\phi} \quad ; \quad a_{max} \le \frac{s_g-1}{C_M} \frac{\sin(\phi+\theta')}{\cos\phi} \tag{76}$$

In terms of the dimensional variables, Eq. 75 can be rewritten as $ga_{min} \le (du'/dt') \le ga_{max}$ 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. 76 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.

2.10 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 SBREAK attached in Appendix may be understood without difficulties.

The normalized forces acting on an armor unit are separated into

$$F_D = \frac{C_D C_2}{2C_3 d} \sigma \; |u - u_a| \; (u - u_a) \tag{77}$$

$$F_L = \frac{C_L C_2}{2C_3 d} \sigma (u - u_a)^2 \tag{78}$$

$$F_I = C_M \frac{du}{dt} \tag{79}$$

$$W_c = \sigma \; (s_g - 1) \; \cos\theta' \tag{80}$$

$$W_s = \sigma \; (s_g - 1) \; \sin\theta' \tag{81}$$

with

$$d = \frac{d'}{H_r'} = (H_r')^{-1} \left(\frac{W'}{C_3 \rho s_g}\right)^{1/3} \quad ; \quad u_a = \frac{u_a'}{(gH_r')^{1/2}} \tag{82}$$

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. 82 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. 77-82 are based on the normalization by $(gW'/\sigma s_g)$. It is noted that the condition given by Eq. 75 is not imposed on the value of du/dt in Eq. 79 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

$$|F_D + F_I - W_s| > F_R \tag{83}$$

with

$$F_R = (W_c - F_L) \tan\phi \geq 0 \tag{84}$$

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. 83 and 84, F_D and F_L are given by Eqs 77 and 78 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

$$(s_g + C_m) \frac{du_a}{dt} = F_D + F_I - W_c - JF_R \tag{85}$$

with

$$J = u_a/|u_a| \tag{86}$$

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:

$$X_a(t) = \frac{X_a'}{d'} = \frac{\sigma}{d} \int_{t_0}^{t} u_a \, dt \tag{87a}$$

$$X_{aa}(t) = \frac{X_a'}{T_r'(gH_r')^{1/2}} = \int_{t_0}^{t} u_a dt \tag{87b}$$

where t_0 = normalized time when the armor unit starts moving. Eq. 87a is used to estimate the degree of the armor movement relative its characteristic length d', whereas Eq. 87b is used to find the x-coordinate of the moving unit since the values of u and du/dt in Eqs. 77-79 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. Alternatively, the locations of individual armor units placed on the slope could be specified but these locations are generally unknown before the actual placement. The movement of the armor unit located at the node j starts when the condition given by Eq. 83 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. 85 is used to find the normalized velocity u_a of the identified unit whose instantaneous location is computed using Eq. 87b. The values of u and du/dt in Eq. 77-79 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. 83 is not satisfied. The stopped unit resumes its movement when the condition given by Eq. 83 is satisfied. The temporal variation of X_a defined by Eq. 87a is also computed for the armor unit identified by its initial location on the slope.

3. COMPUTER PROGRAM SBREAK

3.1 Main Program

The computer program SBREAK attached in Appendix consists of the main program, 37 subroutines and one function, which are written in self-explanatory manners. Double precision is used throughout the program. SBREAK has been written for a mainframe computer, IBM 3090-180E as well as a Sun workstation, which operates diskless in conjunction with a central file server. Consequently, SBREAK may not have to be modified much for other computers. The computation time of SBREAK 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.

3.2 Subroutines and Function

The 37 subroutines and one function arranged in numerical order in the computer program SBREAK are listed in Table 1. The page numbers for the subroutines and function listed in Table 1 correspond to the page numbers used for SBREAK attached in Appendix. 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 SBREAK.
3. INPUT2: this subroutine reads the incident wave profile $\eta_i(t)$ at the seaward boundary for the case of IWAVE = 2 where the incident 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 SBREAK

No.	Subroutine or Function	Page No.		Subroutine or Function	Page No.
1	OPENER	184-187	20	MOVE	221-223
2	INPUT1	187-194	21	FORCES	223-224
3	INPUT2	194-195	22	ACCEL	224
4	BOTTOM	195-198	23	STAT2	224-226
5	PARAM	198-201	24	COEF	226-227
6	INIT1	201-203	25	BALANE	227-229
7	INIT2	203-204	26	MATAFG	229
8	INWAV	204-206	27	MATGJR	229-230
9	FINDM	206-207	28	MATS	230
10	CEL	207-208	29	MATD	230-231
11	SNCNDN	208-209	30	MATU	231-232
12	MARCH	209-211	31	ASSIGN	232
13	LANDBC	211-213	32	DERIV	232
14	RUNUP	213-215	33	DOC1	232-237
15	OVERT	215-216	34	DOC2	237-240
16	SEABC	216-218	35	DOC3	240-244
17	ENERGY	218-219	36	CHEPAR	244
18	STAT1	219	37	CHEOPT	244-245
19	STABNO	219-221	38	STOPP	245

8. INWAV: this subroutine computes the incident monochromatic wave profile at the seaward boundary using Eq. 28 or 31 for the case of IWAVE = 1.

9. FINDM: this subroutine computes the value of the cnoidal wave 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. 41 and the reflected wave train $\eta_r(t)$ using Eq. 13. For the case of IWAVE = 3, the incident solitary wave profile $\eta_i(t)$ given by Eq. 36 is also computed in this subroutine.

17. ENERGY: this subroutine computes the values of E, E_F and D_f defined by Eqs. 58, 59 and 60, 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. 70-73 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. 83-87.

21. FORCES: this subroutine computes the normalized forces given by Eqs. 77-81.

22. ACCEL: this subroutine computes the value of du/dt using Eq. 67.

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. 43-45 as well as the wave transmission coefficients given by Eqs. 54-56 for the case of a submerged structure.

25. BALANE: this subroutine checks the balance of wave energy using Eqs. 61 and 62 as well as the approximate expressions given by Eqs. 64 and 66 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 S_j at the node j defined by Eq. 20.

29. MATD: this subroutine computes the values of the elements of the vector 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 SBREAK.

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

3.3 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 SBREAK 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 SBREAK.
- N2R = N2 = maximum number of time steps allowed for the time-marching computation where N2R = N2 = 30000 in SBREAK.
- N3R = N3 = maximum number of different values of the physical water depth δ_r' allowed for wave runup where N3R = N3 = 3 in SBREAK.
- N4R = N4 = maximum number of points allowed to specify the structure geometry consisting of linear segments where N4R = N4 = 100 in SBREAK.
- 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 SBREAK.

/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. 42.
- 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. 57-66.

- IWAVE = integer indicating the type of the incident wave profile specified at the seaward boundary, where IWAVE = 1 for the incident monochromatic 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 incident solitary wave profile $\eta_i(t)$ computed in the subroutine SEABC.
- 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. 87a 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. 67 at specified time levels if IDUDT = 1.
- ISNR = integer indicating the storage of the spatial variation of N_R given by Eq. 73 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 computed temporal variations at the seaward and landward boundaries every NJUM1 time steps as well as 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. 73 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. This computation must be made after the establishment of the periodicity of the computed temporal variations for the case of IWAVE = 1.

- 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 $h_{s+1} \le \delta$ except that S = JE for IJOB = 3 as discussed below. It is required that $1 \le S \le N1$.
- JE = integer indicating the most landward node j_e of the structure geometry specified as input, where it is required that JE $\le$ 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 SBREAK requires only KS = KSSEA/KSREF. If $H_r' = H'$, KS = 1 since KSSEA = KSREF. It is hence sufficient to set KSSEA=KSREF=1 for $H_r' = H'$.
- 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 $U_r = (H' L'^2/d_t'^3) = (K_s L^2/d_t)$ 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 or beach.

• 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 $n \leq N2$ of the incident wave profile $\eta_i(t)$ at the seaward boundary which is specified in the subroutine INPUT2 if IWAVE = 2 or computed if IWAVE = 1 or 3.

• ETAIS(n_s) = time series of the incident wave train η_i at x=0 stored such that $n_s = [(t/\Delta t) - NSTAT + 1]$ where t = normalized time and NSTAT = time level at the beginning of the statistical computation. 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) where ETAMAX = K_s for the solitary wave profile given by Eq. 36.
- ETAMIN = minimum value of the specified or computed time series ETA(n) where ETAMIN = 0 for the solitary wave profile given by Eq. 36.

/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.

/WAVE6/ solitary wave parameters:

- TCSOL = normalized crest arrival time t_c introduced in Eq. 36.
- KTWO = parameter K_2 computed using Eq. 38.

/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). A sufficient number of linear segments will be required for a complicated bottom geometry partly because sudden changes of slopes may cause numerical difficulties.

/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. 67 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 $\mathbf{g}_j$ defined by Eq. 18 at the node j.
- S1(j) = value of the non-zero element of the vector $\mathbf{S}_j$ 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 $\mathbf{D}_j$ 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 $1 \le i \le$ 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 $\delta_r \ge \delta$ 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. 43, 44 and 45 for k=1, 2 and 3, respectively.
- TCOEF(k) = wave transmission coefficients defined by Eqs. 54, 55 and 56 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. 61 which are computed during the time levels between NSTAT and NTOP, where ENER(i,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. 58, 59 and 60, respectively, whereas $\overline{D_B}$ is computed using Eq. 61.
- ENERB(k) = quantities related to the time-averaged energy balance in the computation domain expressed by Eq. 62, 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. 62; ENERB(6) = right hand side of Eq. 62, ENERB(7) = difference between the right and left hand sides of Eq. 62; 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. 64; ENERB(13) = percentage error in the approximate expression given by Eq. 64; and ENERB(14) = percentage error in the approximate expression given by Eq. 66. These percentage errors may be used to estimate the uncertainties associated with the computed reflection and transmission coefficients as discussed in relation with Eqs. 64 and 66.

/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. 77.
- CL = lift coefficient C_L used in Eq. 78.
- CM = inertia coefficient C_M used in Eq. 79.
- SG = specific gravity s_g of the armor unit.
- TANPH1 = tanϕ with ϕ = frictional angle of the armor unit.
- AMIN = parameter a_{min} specified as input. The condition for a_{min} given in Eq. 76 needs to be satisfied if ISTAB = 1.
- AMAX = parameter a_{max} specified as input. The condition for a_{max} given by Eq. 76 needs to be satisfied if ISTAB = 1.
- DAP = characteristic length d' of the armor unit defined in Eq. 82 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. 80 and 81.

- 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. 60.
- CSTAB2 = $C_M/[(s_g - 1)\ \sigma]$ used in Eq. 60.
- AMAXS = $\sigma\ a_{max}$ used in Eq. 75.
- AMINS = $\sigma\ a_{min}$ used in Eq. 75.
- E2 = $C_L \tan\phi/C_D$ used in Eq. 71.
- 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. 72.

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

- CSTAB3 = $C_2 C_D/(2C_3 d)$ used in Eq. 77.
- CSTAB4 = $C_2 C_L/(2C_3 d)$ used in Eq. 78.
- CM1 = $(C_M - 1)$ = added mass coefficient C_m used in Eq. 85.
- DA = normalized characteristic length d of the armor unit defined in Eq. 82.
- SIGDA = σ/d used in Eq. 87a.
- WEIG = $\sigma(s_g - 1)$ used in Eqs. 80 and 81.

/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. 85.

• XAA(j) = normalized displacement X_{aa} defined by Eq. 87b of the armor unit from its initial location at the node j.

• XA(j) = normalized displacement X_a defined by Eq. 87a 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 $t = (n-1)\Delta t$ stored at the beginning of each time step:

- VSN = u_s used in the landward boundary computation.
- USN(i) = U_s such that USN(1) = m_s and USN(2) = h_s used in the landward boundary computation.
- VMN = u_{s-1} used in the landward boundary computation.
- UMN(1) = m_{s-1} used in the landward boundary computation.
- V1N = u_1 used in Eq. 41 in the seaward boundary computation.
- V2N = u_2 used in Eq. 41 in the seaward boundary computation.

3.4 Input Parameters and Variables

The input parameters and variables are summarized in the sequence of the data input in SBREAK. 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 below.

```
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, the file name FINP2 containing the data on the incident 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. It is noted that the increase of NONE tends to reduce numerical instability as long as the numerical stability indicator ALPHAS described at the end of this section is less than about ten. 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. Use has been made of S = 100-400 to provide adequate spatial resolution. 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 $1 \le L \le$ 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
```

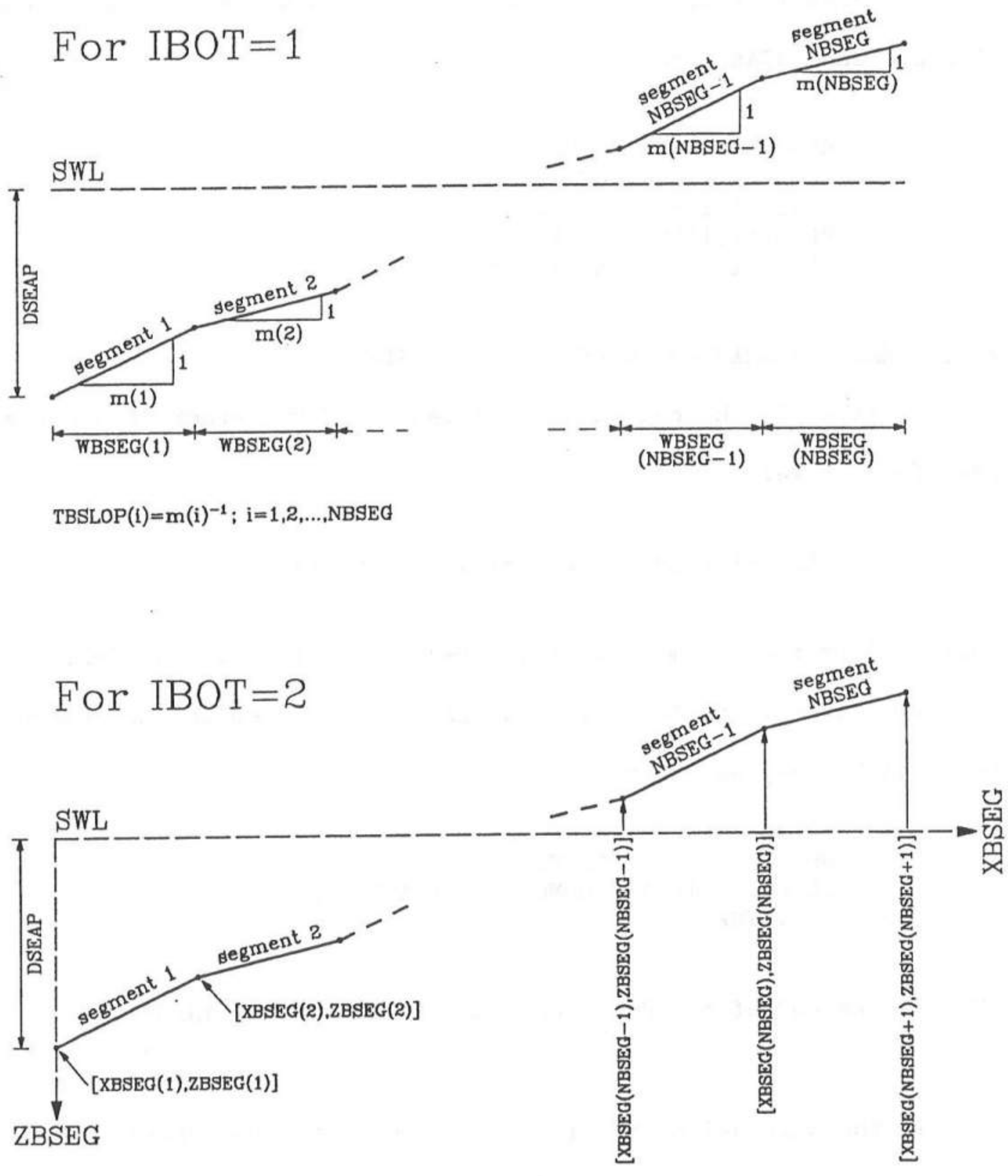

FIGURE 4. Input of Dimensional Structure Geometry for IBOT = 1 and 2.

```
      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.

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.

In the subroutine INPUT1, the following two parameters associated with the incident solitary wave profile are specified if IWAVE=3:

t_c = TCSOL = 1.0

δ_i = DELTAI = 0.05

where t_c = normalized arrival time of the solitary wave crest introduced in Eq. 36; and δ_i = small value used to estimate the reference wave period T_r' associated with the solitary wave on the basis of Eqs. 37-39. These values of t_c and δ_i can be changed by modifying only the two lines in the subroutine INPUT1.

If IWAVE = 2, the time series associated with the normalized incident 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

$$\text{ALPHAS} = \frac{\Delta x}{\Delta t}\,(1 + d_t^{1/2})^{-1}\left[\left(1 + \frac{\epsilon^2}{4}\right)^{1/2} - \frac{\epsilon}{2}\right] \qquad (88)$$

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. On the other hand, our experiences have indicated that the use of ALPHAS greater than about ten does not improve numerical stability in spite of the increase of computation time.

3.5 Error and Warning Statements

The computer program SBREAK 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 SBREAK, 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 SBREAK, 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 500 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 $h^*_{s-1} \leq \delta$, corresponding to the third step in the numerical procedure dealing the moving waterline in Section 2.5. 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} or the value of δ 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 in Section 2.5 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 SEABC for the case of IWAVE=3, the incident solitary wave profile $\eta_i(t)$ given by Eq. 36 is computed. The maximum and minimum values of $\eta_i(t)$ denoted by ETAMAX and ETAMIN are set to be equal to K_s and zero, respectively, in the subroutine INPUT1. If the computed value of $\eta_i(t)$ is greater than ETAMAX, the following statement is written and the computation stops:

```
          WRITE(*,2910)
          WRITE(29,2910)
    2910  FORMAT(/'From Subr. 16 SEABC:'/
                 'ETAMAX exceeds KS')
```

If the computed value of $\eta_i(t)$ is less than ETAMIN, the computation stops after writing the following statement:

```
          WRITE(*,2920)
          WRITE(29,2920)
    2920  FORMAT(/'From Subr. 16 SEABC:'/
                 'ETAMIN is less than zero')
```

In the subroutine STABNO, the following statement is written and the computation stops if the condition given by Eq. 74 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. 76.

In the subroutine DOC1, the following statement is written if the values of ALPHAS given by Eq. 88 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 ≤ NNOD1 ≤ N5, 1 ≤ NNOD2 ≤ N5, 1 ≤ NNREQ ≤ N5, 1 ≤ S ≤ N1, 1 ≤ NDELR ≤ N3 and 1 ≤ 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.

3.6 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 monochromatic 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 $t = (I-1)\Delta t$ and NONE1=(NONE+1). The unit numbers and file names used in SBREAK 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. If IWAVE = 3, the following solitary wave parameters are printed:

```
WRITE(28,2815) TCSOL, KTWO
```

where TCSOL and KTWO are explained in the common /WAVE6/. The following quantities explained in the commons /WAVE1/, /WAVE2/, /WAVE4/, /BOT1/ and /BOT2/ are then 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)
```

Next, 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. 88 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 SBREAK. 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. 87a 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. 43, 44 and 45 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 $\overline{m}$ at the seaward boundary which should be equal to OV(1) = value of $\overline{m}$ at the landward edge if the steady state is really established. It should also be noted that small time-averaged quantities relative to large time-varying quantities are harder to predict very accurately. 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. 54, 55 and 56 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(23,2320) (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
```

4. RUNUP OF BROKEN SOLITARY WAVES

The numerical model described in Section 2 is compared herein with available experimental data on solitary wave runup on a smooth uniform slope. Examples of the input and output of the computer program SBREAK explained in Section 3 are given for a few computed cases.

An emphasis is placed on runup of broken solitary waves since runup of nonbreaking solitary waves is better understood theoretically (e.g., Synolakis, 1987a) and can be predicted well by numerical models based on potential flow theory (e.g., Liu and Cho, 1993). As for runup of breaking or broken solitary waves, no general empirical formula exists unlike Hunt's formula for monochromatic waves (Battjes, 1974; Ahrens and Martin, 1985). Numerical models based on potential flow theory are not applicable to broken waves. Boussinesq wave models such as a Lagrangian finite-element model of Zelt (1991) may be used to predict runup of broken solitary waves if the effects of wave breaking and bottom friction are included empirically. The Boussinesq models include an additional nonhydrostatic pressure correction in the momentum equation given by Eq. 2. However, it is not obvious whether the nonhydrostatic pressure correction based on the potential flow assumption of weakly nonlinear and relatively long waves is really valid for broken solitary waves. In any case, the present numerical model is the simplest time-dependent, one-dimensional model for breaking or broken waves.

4.1 Sensitivity Analysis for Incident Solitary Wave Parameters

The incident solitary wave profile $\eta_i(t)$ has been assumed to be expressed in the normalized form of Eq. 36. In this report, the representative wave height H'_r used for the normalization is taken as the incident solitary wave height H' at the toe of a uniform slope located at $x=0$. As a result, the

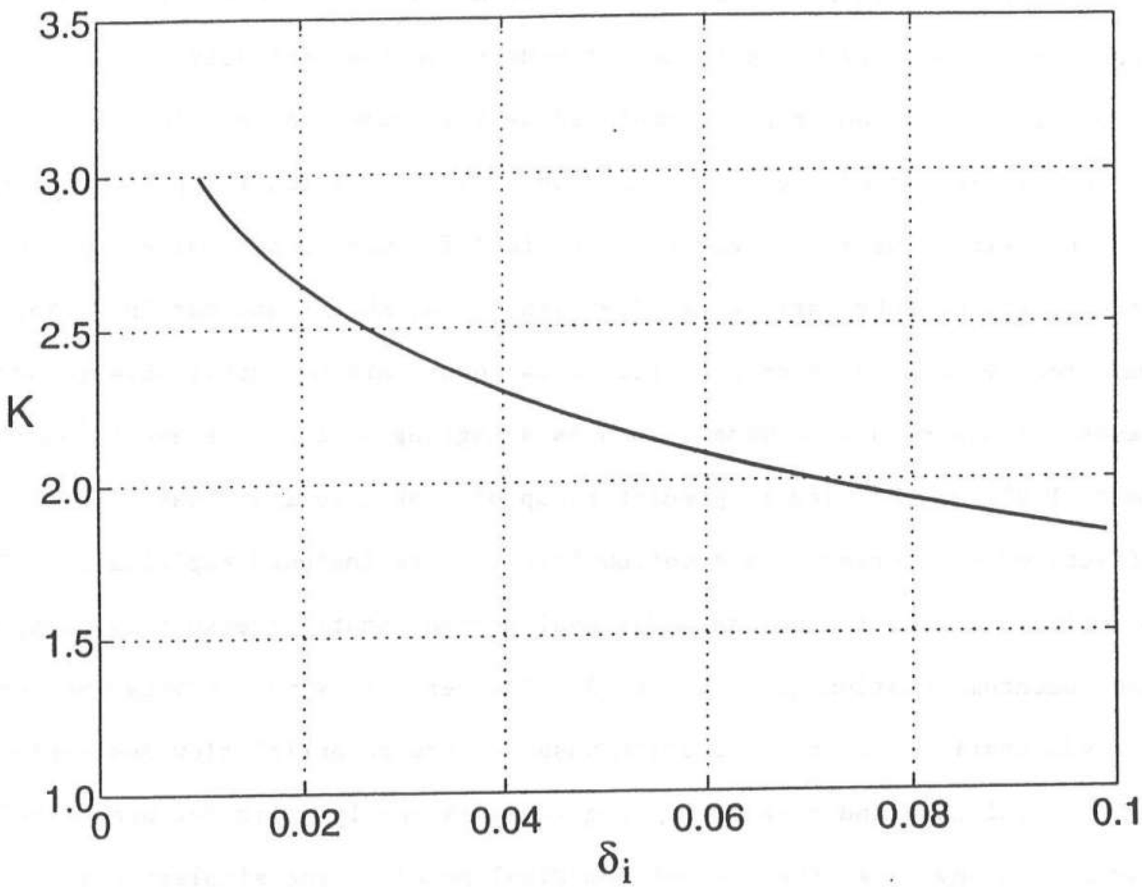

FIGURE 5. $K = K_2/2$ as a function of δ_i.

following computation is limited to the case of $K_s = H'/H_r' = 1$. Then, the variation of $\eta_i(t)$ with respect to the normalized time $t = t'/T_r'$ depends on the two parameters K_2 and t_c. The parameter K_2 is given by Eq. 38 with $K_s = 1$ where $\eta_i(t) \geq \delta_i$ in the unit duration $(t_c - 0.5) \leq t \leq (t_c + 0.5)$ about the normalized crest arrival time t_c.

Fig. 5 shows $K = K_2/2$ as a function of the small parameter δ_i. The parameter K decreases from K = 2.99 at $\delta_i = 0.01$ to K = 1.82 at $\delta_i = 0.1$. The parameter K is related to the complete elliptic integral of the first kind associated with the cnoidal wave profile given by Eq. 31. The range K = 1.82 - 2.99 for cnoidal wave theory corresponds to the Ursell parameter U_r = 8 - 46. Fig. 5 indicates that the parameter $K_2 = 2K$ is not very sensitive to δ_i in the range δ_i = 0.1 - 0.01, although the selection of δ_i is somewhat arbitrary.

Fig. 6 shows $\eta_i(t=0)$ given by Eq. 40 with $K_s=1$ as a function of δ_i for t_c = 0.5, 1.0 and 1.5. The initial value of η_i at t=0 is essentially zero for the range δ_i = 0.01 - 0.1 if the normalized crest arrival time t_c is taken as $t_c = 1$. The use of $t_c = 1$ will ensure a smooth transition from the initial conditions of no wave action in the computation domain $x \geq 0$. Moreover, the computed temporal variations starting from t = 0 can be interpreted easily by selecting the crest arrival time $t_c = 1$.

Fig. 7 shows the parameter $\sigma = T_r'\sqrt{g/H'}$ defined in Eq. 5 and given by Eq. 39 with $K_s = 1$ as a function of the normalized water depth, $d_t = d_t'/H'$, for δ_i = 0.01, 0.05 and 0.1. The present numerical model assumes that $\sigma^2 \gg 1$ and d_t is of the order of unity. Fig. 7 indicates that the assumption of $\sigma^2 \gg 1$ should be valid as long as d_t is larger than about 2, for which the incident solitary waves at the toe of the slope are non-breaking. If d_t becomes much greater than unity, the dispersive effects neglected in the present numerical

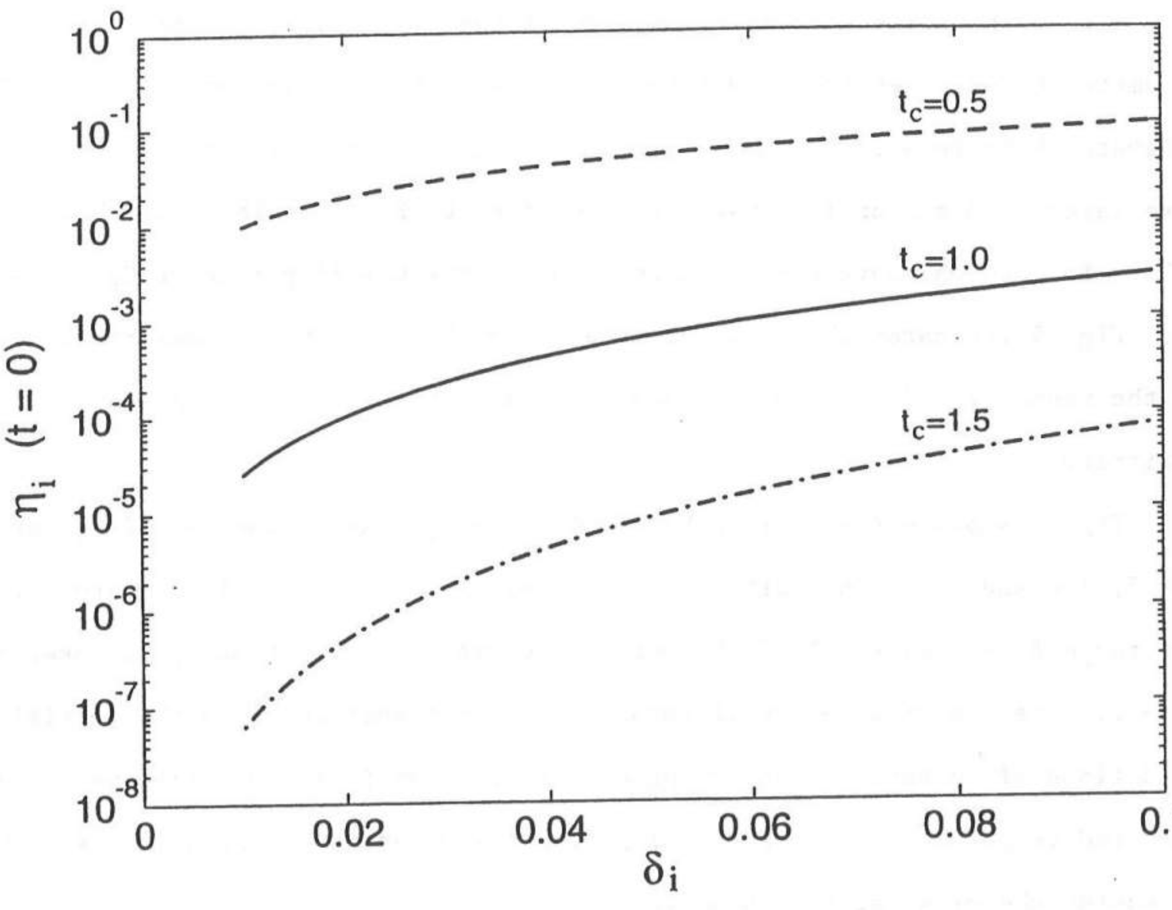

FIGURE 6. Initial Value of η_i at t = 0 as a Function of δ_i for t_c = 0.5, 1.0 and 1.5.

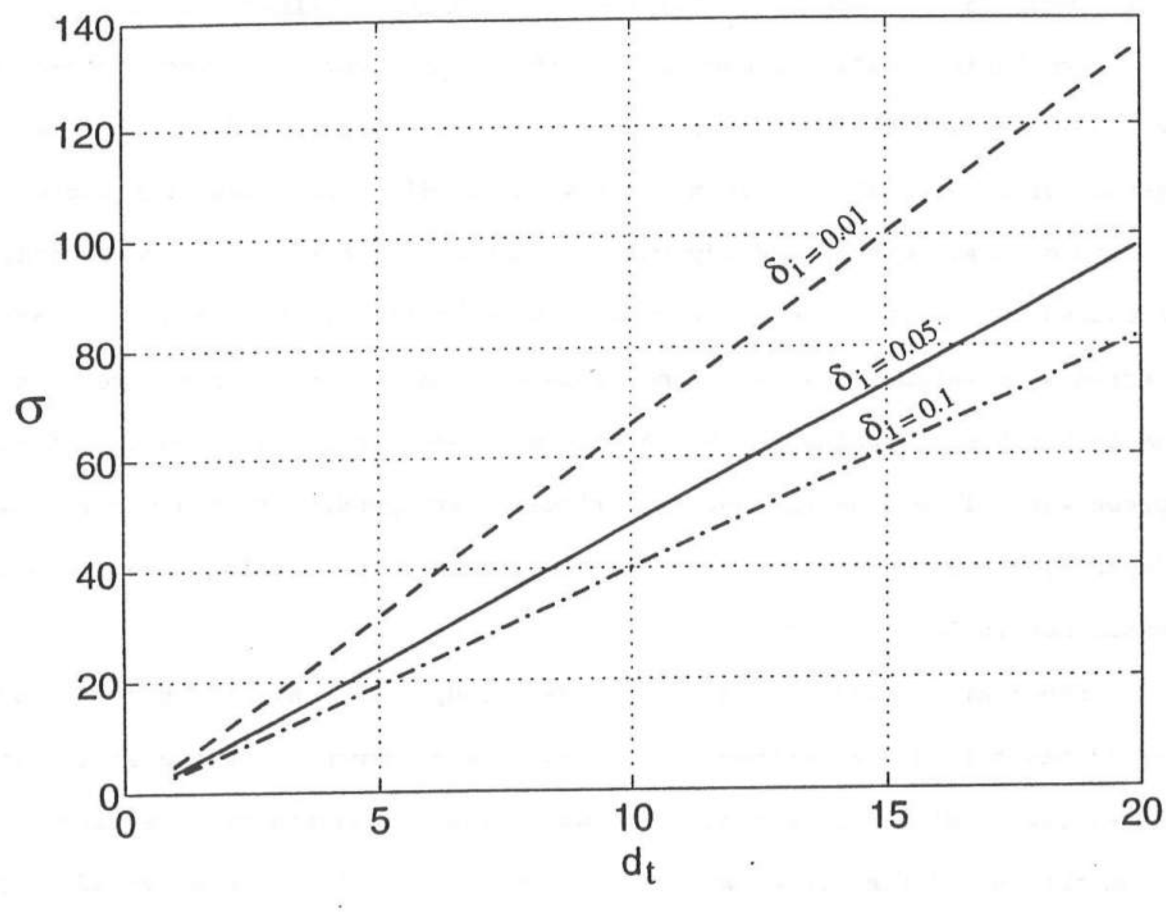

FIGURE 7. Parameter σ as a Function of d_t for δ_i = 0.01, 0.05 and 0.1.

model may not be negligible in the region between the toe of the slope and the point of wave breaking. This is one of the reasons why the numerical model is compared with only breaking solitary wave data of Synolakis (1987a) in the following.

4.2 Runup Data of Broken Solitary Waves on Smooth Uniform Slopes

Synolakis (1987a) conducted experiments in a wave tank whose dimensions were 37.73 m × 0.61 m × 0.39 m. At a distance of 14.68 m from the wave generator a beach of a uniform slope with cot $\theta' = 19.85$ was constructed of aluminum panels with a hydrodynamically smooth surface. The wave generator produced near-perfect solitary waves. Wave heights in the constant depth region were measured by resistance-type wave gages. The runup gage consisted of an array of capacitance wave probes mounted on a frame. The tip of each probe was 0.1 cm from the bottom surface, corresponding to the physical water depth $\delta'_r = 0.1$ cm used in the numerical model for predicting wave runup as explained in Section 2.5.

Synolakis (1987a) listed the values of d'_t, $d_t^{-1} = H'/d'_t$ and R'/d'_t for each of 77 tests in his experiments where d'_t = water depth below the still water level (SWL), H' = incident solitary wave height measured at a certain distance from the toe of the slope, and R' = maximum runup elevation above SWL. The measured wave height is assumed herein to be the same as the wave height H' at the toe of the slope to reduce the computation domain. Breaking on the 1:19.85 slope occurred during backwash when $d_t^{-1} > 0.044$, that is, $d_t < 22.7$ and during uprush when $d_t^{-1} > 0.055$, that is, $d_t < 18.2$. The comparisons between the measured and computed runup in this report are limited to breaking solitary waves during uprush since the present numerical model may not be applicable for very large d_t. Table 2 summarizes the 43 tests for which d_t^{-1} =

TABLE 2. Broken Solitary Wave Runup Data of Synolakis (1987a)

Test No	d'_t (m)	H'/d'_t	R'/d'_t	H' (m)	d_t	R
1	0.0625	0.250	0.506	0.015625	4.0000	2.0240
2	0.0625	0.072	0.233	0.004500	13.8889	3.2361
3	0.0801	0.448	0.723	0.035885	2.2321	1.6138
4	0.0979	0.078	0.251	0.007636	12.8205	3.2179
5	0.0979	0.384	0.621	0.037594	2.6042	1.6172
6	0.0981	0.097	0.274	0.009516	10.3093	2.8247
7	0.0984	0.462	0.659	0.045461	2.1645	1.4264
8	0.0989	0.236	0.467	0.023340	4.2373	1.9788
9	0.1317	0.294	0.542	0.038720	3.4014	1.8435
10	0.1454	0.610	0.780	0.088694	1.6393	1.2787
11	0.1454	0.591	0.790	0.085931	1.6920	1.3367
12	0.1454	0.607	0.805	0.088258	1.6474	1.3262
13	0.1454	0.607	0.780	0.088258	1.6474	1.2850
14	0.1550	0.601	0.801	0.093155	1.6639	1.3328
15	0.1567	0.090	0.270	0.014103	11.1111	3.0000
16	0.1572	0.259	0.519	0.040715	3.8610	2.0039
17	0.1576	0.590	0.810	0.092984	1.6949	1.3729
18	0.1562	0.298	0.551	0.046548	3.3557	1.8490
19	0.1565	0.322	0.591	0.050393	3.1056	1.8354
20	0.1569	0.170	0.407	0.026673	5.8824	2.3941
21	0.1670	0.273	0.487	0.045591	3.6630	1.7839
22	0.1753	0.276	0.495	0.048383	3.6232	1.7935
23	0.1942	0.633	0.842	0.122929	1.5798	1.3302
24	0.1942	0.625	0.825	0.121375	1.6000	1.3200
25	0.1947	0.626	0.862	0.121882	1.5974	1.3770
26	0.1956	0.283	0.527	0.055355	3.5336	1.8622
27	0.1962	0.286	0.513	0.056113	3.4965	1.7937
28	0.2080	0.323	0.555	0.067184	3.0960	1.7183
29	0.2092	0.188	0.409	0.039330	5.3191	2.1755
30	0.2092	0.271	0.513	0.056693	3.6900	1.8930
31	0.2092	0.416	0.686	0.087027	2.4038	1.6490
32	0.2101	0.159	0.384	0.033406	6.2893	2.4151
33	0.2144	0.160	0.384	0.034304	6.2500	2.4000
34	0.2147	0.143	0.366	0.030702	6.9930	2.5594
35	0.2349	0.394	0.641	0.092551	2.5381	1.6269
36	0.2638	0.267	0.507	0.070435	3.7453	1.8989
37	0.2940	0.075	0.258	0.022050	13.3333	3.4400
38	0.2954	0.073	0.248	0.021564	13.6986	3.3973
39	0.2962	0.065	0.228	0.019253	15.3846	3.5077
40	0.2972	0.056	0.207	0.016643	17.8571	3.6964
41	0.3093	0.188	0.425	0.058148	5.3191	2.2606
42	0.3138	0.094	0.288	0.029497	10.6383	3.0638
43	0.3535	0.193	0.426	0.068226	5.1813	2.2073
max	0.3535	0.633	0.8620	0.122929	17.8571	3.6964
min	0.0625	0.056	0.2070	0.004500	1.5798	1.2787

$H'/d_t' \geq 0.056$ where the normalized runup R is defined as $R = R'/H'$ to be consistent with the normalization given by Eq. 4 with $H' = H_r'$. The maximum and minimum values of d_t', H'/d_t', R'/d_t', H', d_t and R for the 43 tests are listed in Table 2 to indicate the ranges of these parameters associated with the broken solitary wave runup data of Synolakis (1987a).

Synolakis (1987a) stated that the shoreline assumed a parabolic shape in plan view, possibly due to sidewall effects. His formula for maximum runup based on the maximum position of the shoreline can be rewritten as

$$R = 1.109\ d_t^{0.418} \tag{89}$$

The average position of the shoreline resulted in the following formula

$$R = 0.918\ d_t^{0.394} \tag{90}$$

Eqs. 89 and 90 are plotted in Fig. 8 together with the 43 tests in Table 2 which lists the value of R based on the maximum shoreline position for each run. The correlation coefficient r between the measured and empirical values of R is calculated to be r = 0.996, indicating the good fit by Eq. 89. However, Fig. 8 also suggests the difficulty and uncertainty in comparing the present one-dimensional numerical model with the data affected by sidewall effects. In the following, the measured runup based on the maximum shoreline position is assumed to correspond to the computed runup based on the assumption of alongshore uniformity. Eq. 89 is limited to breaking or broken solitary waves on the smooth 1:19.85 slope. For non-breaking solitary waves, Synolakis (1987a) also analyzed other available data for different slopes and proposed the following formula including the slope effect

$$R = 2.831 \sqrt{\cot \theta'}\ d_t^{-1/4} \tag{91}$$

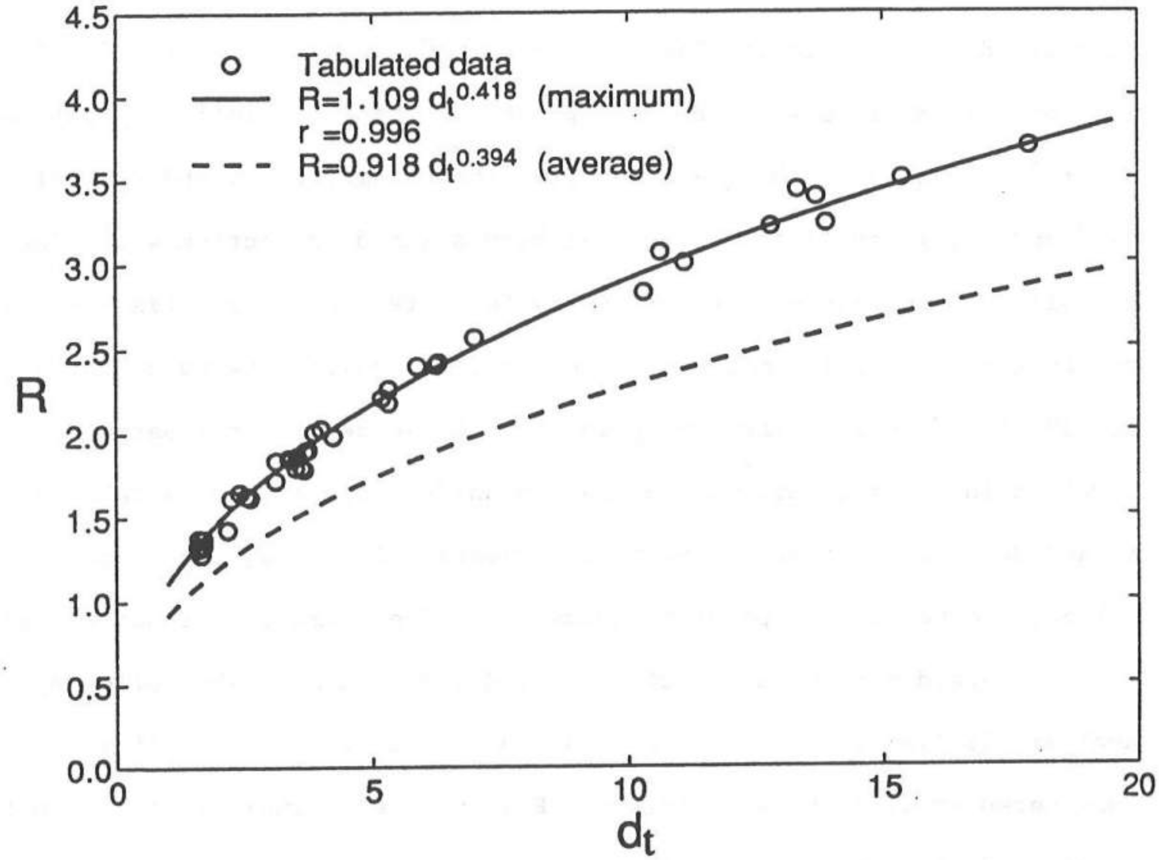

FIGURE 8. Empirical Formulas for Runup R Based on Maximum and Average Shoreline Positions on Smooth 1:19.85 Slope.

Consequently, an attempt is made hereafter to extend Eq. 89 and develop an empirical formula for R including the slope effect.

The continuity and momentum equations normalized as Eqs. 6 and 7 as well as the normalized incident solitary wave profile given by Eq. 36 with $K_s = 1$ are used to identify the dimensionless parameters involved in this problem.

Eqs. 6 and 7 involve the normalized slope gradient, $\theta = \sigma \tan \theta'$ and the normalized bottom friction factor, $f = \sigma f'/2$ where $\sigma = T_r'(g/H')^{1/2}$ for $K_s = 1$. For uniform slopes, θ can be replaced by the surf similarity parameter $\xi = \theta/(2\pi)^{1/2}$. Eq. 36 with $K_s = 1$ requires the parameters K_2 and t_c where K_2 can be found for given δ_i and $t_c = 1$ has been assumed in Section 4.1. The normalized slope geometry described by Eq. 8 reduces to $z = (\theta x - d_t)$ in the region $x \geq 0$ for uniform slopes. For solitary waves, the parameter σ given by Eq. 39 with $K_s = 1$ depends on d_t and K_2. Consequently, the parameters involved in the solitary wave motion on uniform slopes may be taken as θ, f, δ_i and d_t. Alternatively, the four parameters ξ, f', δ_i and d_t may be selected to be the independent parameters. The normalized runup $R = R'/H'$ may thus be regarded to be a function of ξ, d_t, f' and δ_i. The following data analysis is limited to smooth slopes and the friction factor f' is not considered explicitly, assuming that R is not very sensitive to f' in the range of f' expected for smooth slopes. This assumption has been shown to be appropriate for monochromatic waves (Kobayashi and Watson, 1987).

For each of the 43 tests listed in Table 2, the value of σ is computed using Eq. 39 for $\delta_i = 0.01$, 0.05 and 0.1 corresponding to $K_2 = 5.986$, 4.357 and 3.637, respectively. The representative wave period is then computed using $T_r' = \sigma(H'/g)^{1/2}$ and the surf similarity parameter given by $\xi = \sigma \tan \theta' /(2\pi)^{1/2}$ with $\cot \theta' = 19.85$ for these runs. The calculated values of T_r', σ and ξ for each run are listed in Table 3 where these value increase with the decrease of δ_i. The calculated representative wave periods are of the order of 1 sec and may be reasonable for the small-scale experiments. The calculated values of σ are of the order of ten or greater and satisfy the assumption of $\sigma^2 >> 1$ made in the present numerical model. The surf similarity parameter ξ for breaking or broken solitary waves is less than about two, which approximately corresponds to the breaking condition of monochromatic waves (Battjes, 1974). The use of the surf similarity parameter for solitary waves allows the comparisons between solitary and monochromatic waves, although the representative wave period depends on the small parameter δ_i, which is somewhat arbitrary.

The following empirical formula for the normalized runup R for breaking monochromatic waves on smooth uniform slopes is well established (Battjes, 1974; Ahrens and Martin, 1985)

$$R = \xi = \frac{\sigma \tan \theta'}{\sqrt{2\pi}} \qquad \text{for } 0.1 \leq \xi \leq 2.3 \tag{92}$$

The normalized runup R for solitary waves on smooth uniform slopes depends on ξ, d_t and δ_i. The surf similarity parameter ξ includes the effects of both $\tan \theta'$ and σ in the single parameter where σ given by Eq. 39 depends on d_t and δ_i. In view of Eq. 92, the normalized runup of breaking solitary waves on smooth uniform slopes may be assumed to be a function of ξ only for given δ_i, provided that the normalized toe depth d_t is sufficiently large and its direct effect on R is negligible. For the 43 tests listed in Table 2, $1.57 < d_t < 17.86$.

The empirical relationship between R and ξ is estimated using Eq. 89. For the range $1.57 < d_t < 17.86$, Eq. 39 with $K_s = 1$ suggests that σ is approximately proportional to d_t. Then, ξ is approximately proportional to d_t for given $\tan \theta'$. Fig. 9 shows the relationship between ξ and d_t for the 43

TABLE 3. Calculated Values of T_r', σ and ξ for δ_i = 0.1, 0.05 and 0.01 for 43 Tests listed in Table 2.

Test No	T_r' (sec)			σ			ξ		
	$\delta_i = 0.1$	$\delta_i = 0.05$	$\delta_i = 0.01$	$\delta_i = 0.1$	$\delta_i = 0.05$	$\delta_i = 0.01$	$\delta_i = 0.1$	$\delta_i = 0.05$	$\delta_i = 0.01$
1	0.5996	0.7183	0.9870	15.0247	17.9977	24.7311	0.3020	0.3617	0.4970
2	1.2065	1.4453	1.9860	56.3339	67.4810	92.7275	1.1322	1.3562	1.8636
3	0.4711	0.5644	0.7755	7.7900	9.3315	12.8226	0.1566	0.1875	0.2577
4	1.4468	1.7331	2.3814	51.8556	62.1166	85.3561	1.0422	1.2484	1.7155
5	0.5755	0.6893	0.9472	9.2961	11.1356	15.3017	0.1868	0.2238	0.3075
6	1.2874	1.5421	2.1191	41.3357	49.5150	68.0399	0.8308	0.9951	1.3675
7	0.5118	0.6130	0.8424	7.5177	9.0053	12.3744	0.1511	0.1810	0.2487
8	0.7807	0.9352	1.2851	16.0058	19.1730	26.3462	0.3217	0.3853	0.5295
9	0.7889	0.9450	1.2985	12.5570	15.0417	20.6692	0.2524	0.3023	0.4154
10	0.5159	0.6180	0.8492	5.4257	6.4993	8.9309	0.1090	0.1306	0.1795
11	0.5273	0.6316	0.8679	5.6335	6.7482	9.2729	0.1132	0.1356	0.1864
12	0.5177	0.6201	0.8521	5.4576	6.5375	8.9834	0.1097	0.1314	0.1805
13	0.5177	0.6201	0.8521	5.4576	6.5375	8.9834	0.1097	0.1314	0.1805
14	0.5381	0.6446	0.8858	5.5224	6.6152	9.0901	0.1110	0.1330	0.1827
15	1.6946	2.0299	2.7894	44.6935	53.5372	73.5669	0.8982	1.0760	1.4785
16	0.9310	1.1152	1.5324	14.4506	17.3101	23.7862	0.2904	0.3479	0.4781
17	0.5496	0.6583	0.9046	5.6448	6.7618	9.2915	0.1134	0.1359	0.1867
18	0.8520	1.0206	1.4025	12.3693	14.8169	20.3603	0.2486	0.2978	0.4092
19	0.8130	0.9738	1.3382	11.3430	13.5875	18.6710	0.2280	0.2731	0.3752
20	1.1909	1.4265	1.9602	22.8380	27.3571	37.5921	0.4590	0.5498	0.7555
21	0.9295	1.1134	1.5299	13.6340	16.3318	22.4420	0.2740	0.3282	0.4510
22	0.9460	1.1332	1.5571	13.4699	16.1353	22.1720	0.2707	0.3243	0.4456
23	0.5812	0.6962	0.9566	5.1916	6.2189	8.5456	0.1043	0.1250	0.1717
24	0.5863	0.7023	0.9651	5.2710	6.3140	8.6762	0.1059	0.1269	0.1744
25	0.5864	0.7024	0.9652	5.2610	6.3020	8.6597	0.1057	0.1267	0.1740
26	0.9841	1.1788	1.6199	13.1009	15.6932	21.5645	0.2633	0.3154	0.4334
27	0.9793	1.1731	1.6119	12.9483	15.5105	21.3134	0.2602	0.3117	0.4284
28	0.9354	1.1205	1.5398	11.3036	13.5403	18.6061	0.2272	0.2721	0.3739
29	1.2977	1.5544	2.1360	20.4943	24.5497	33.7344	0.4119	0.4934	0.6780
30	1.0449	1.2517	1.7200	13.7454	16.4653	22.6254	0.2763	0.3309	0.4547
31	0.7990	0.9571	1.3152	8.4835	10.1622	13.9641	0.1705	0.2042	0.2806
32	1.4317	1.7149	2.3565	24.5336	29.3882	40.3831	0.4931	0.5906	0.8116
33	1.4411	1.7262	2.3721	24.3697	29.1919	40.1134	0.4898	0.5867	0.8062
34	1.5367	1.8408	2.5295	27.4689	32.9043	45.2147	0.5521	0.6613	0.9087
35	0.8769	1.0504	1.4433	9.0276	10.8139	14.8597	0.1814	0.2173	0.2986
36	1.1840	1.4183	1.9489	13.9733	16.7383	23.0006	0.2808	0.3364	0.4623
37	2.5604	3.0670	4.2145	54.0051	64.6913	88.8941	1.0854	1.3002	1.7866
38	2.6038	3.1190	4.2860	55.5363	66.5256	91.4146	1.1162	1.3370	1.8372
39	2.7735	3.3223	4.5653	62.6054	74.9935	103.0505	1.2582	1.5072	2.0711
40	3.0058	3.6006	4.9477	72.9760	87.4161	120.1209	1.4667	1.7569	2.4142
41	1.5779	1.8901	2.5972	20.4943	24.5497	33.7344	0.4119	0.4934	0.6780
42	2.3422	2.8056	3.8553	42.7133	51.1652	70.3075	0.8584	1.0283	1.4130
43	1.6614	1.9901	2.7346	19.9215	23.8635	32.7915	0.4004	0.4796	0.6590
max	3.0058	3.6006	4.9477	72.9760	87.4161	120.1209	1.4667	1.7569	2.4142
min	0.4711	0.5644	0.7755	5.1916	6.2189	8.5456	0.1043	0.1250	0.1717

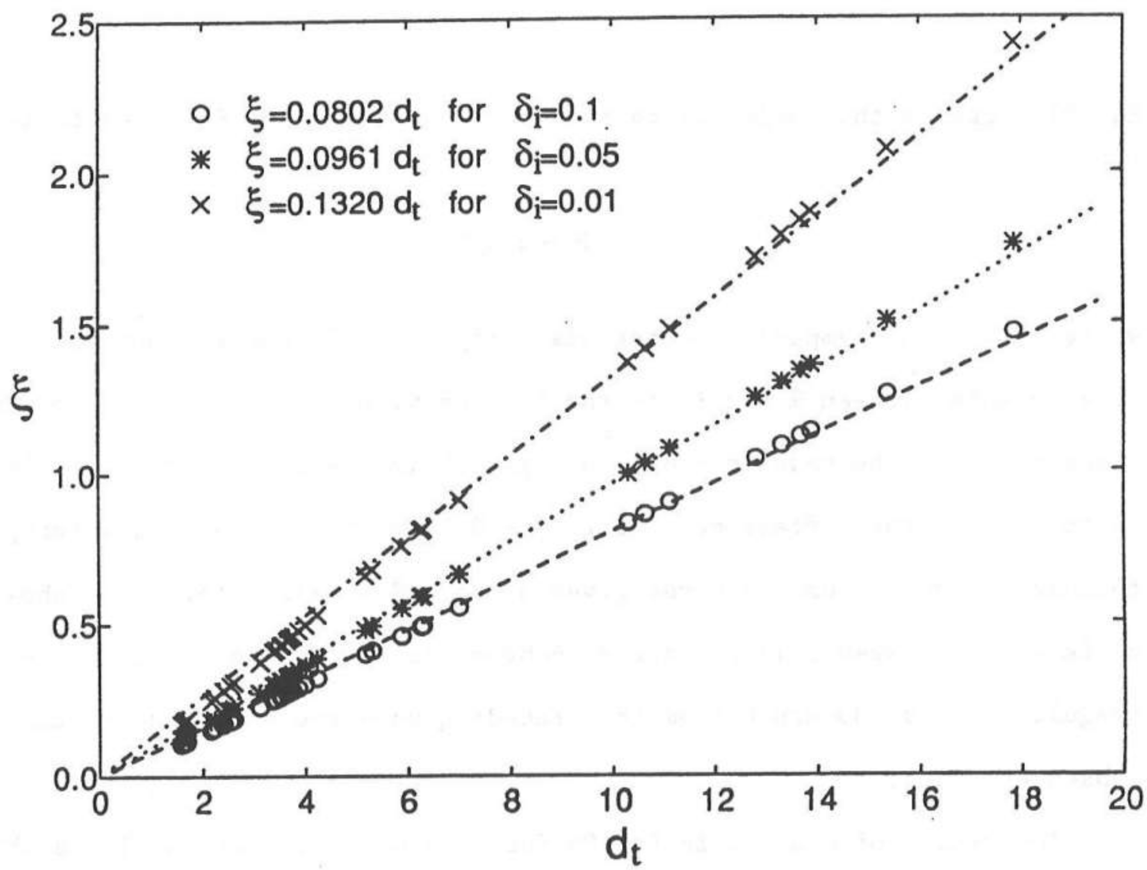

FIGURE 9. Relationship between ξ and d_t with δ_i = 0.1, 0.05 and 0.01 for 43 Tests Listed in Tables 2 and 3.

tests with cot θ' = 19.85 listed in Tables 2 and 3. Linear regression analyses based on $\xi = cd_t$ yield the coefficient c = 0.0802, 0.0961 and 0.1320 for δ_i = 0.1, 0.05 and 0.01, respectively. Substitution of $d_t = \xi/c$ into Eq. 89 results in

$$R = \frac{1.109}{c^{0.418}} \xi^{0.418} \tag{93}$$

Eq. 93 suggests the empirical relationship between R and ξ in the following form:

$$R = a \xi^{b} \tag{94}$$

where a and b are empirical constants. Figs. 10, 11 and 12 show the relationship between R and ξ for the 43 runs with δ_i = 0.1, 0.05 and 0.01, respectively. The results shown in Figs. 10-12 are plotted together in Fig. 13 to clarify the differences among δ_i = 0.1, 0.05 and 0.01. The runup formula for monochromatic waves given by Eq. 92 is also plotted to show the differences between solitary and monochromatic wave runup. For monochromatic (regular) waves, downrush from the preceding wave reduces runup of the subsequent wave.

The values of a and b in Eq. 94 for δ_i = 0.1, 0.05 and 0.01 are obtained for a linear regression analysis based on $\ell og(R) = [\ell og(a) + b\ \ell og(\xi)]$. The fitted values of a and b as well as the correlation coefficient r between the measured and empirical values of R are listed in Table 4. The values of r for δ_i = 0.1, 0.05 and 0.01 are the same because ξ is proportional to K_2 in view of Eqs. 39 and 92, while K_2 given by Eq. 38 with $K_s = 1$ depends on δ_i only. The values of b in Table 4 are very close to b = 0.418 based on Eq. 93. Comparing Eqs. 93 and 94, $a = 1.109/c^{0.418}$, which is a good approximation for the 43 tests. This implies that the runup relationship expressed in the form

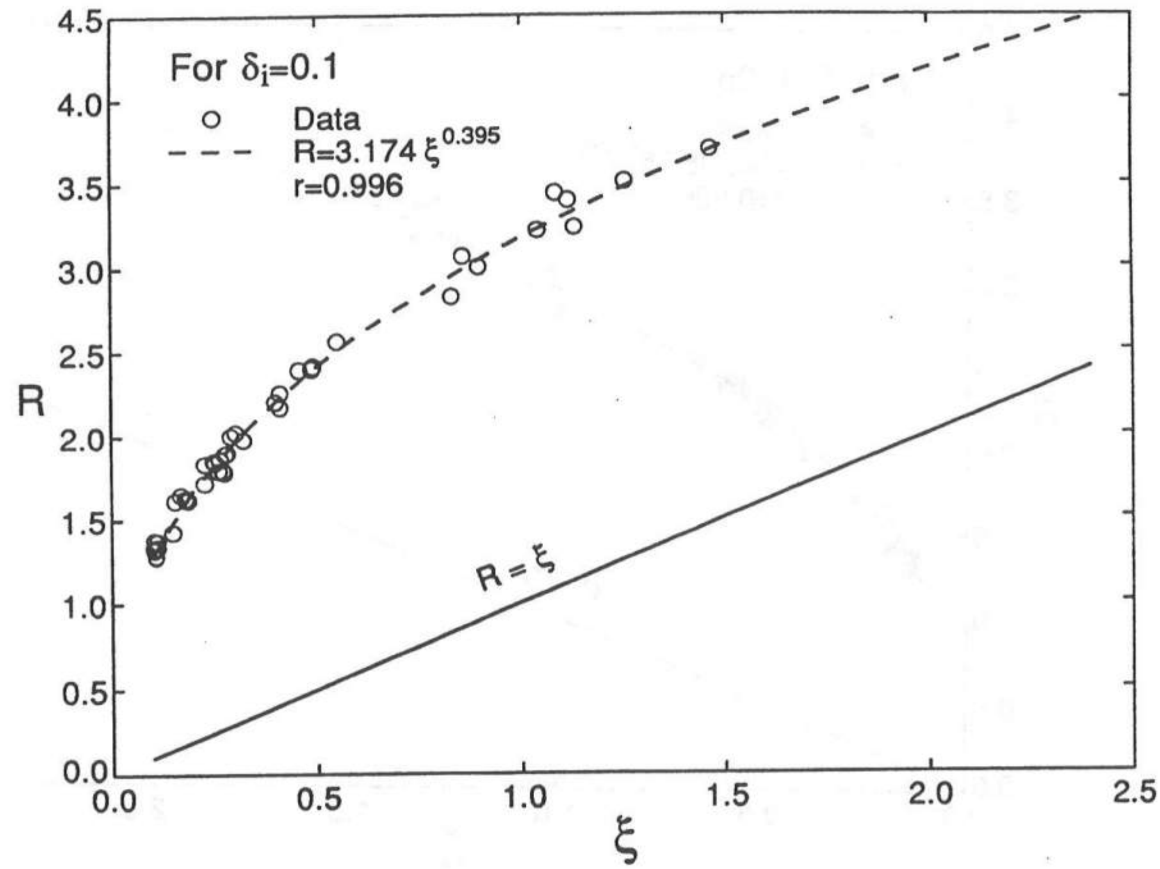

FIGURE 10. Relationship between R and ξ for 43 Tests with δ_i = 0.1.

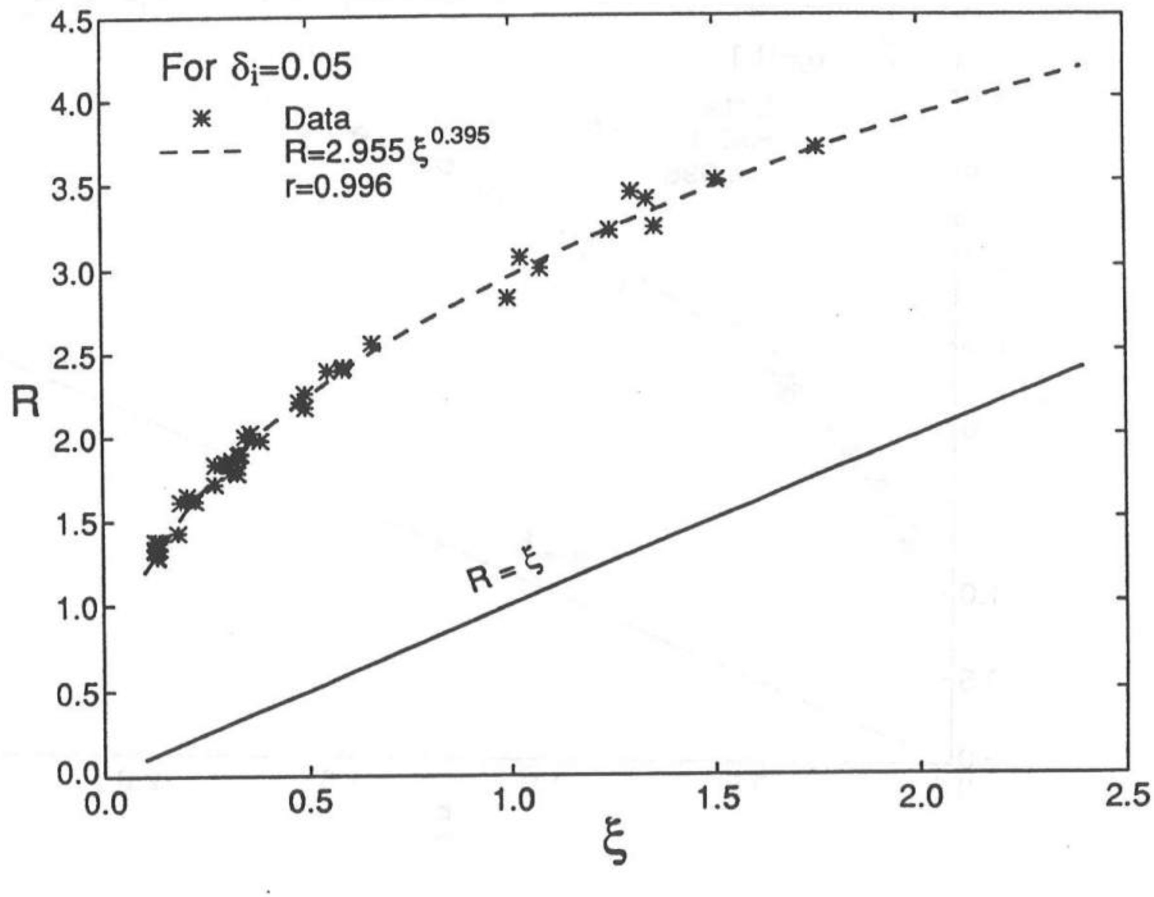

FIGURE 11. Relationship between R and ξ for 43 Tests with $\delta_i = 0.05$.

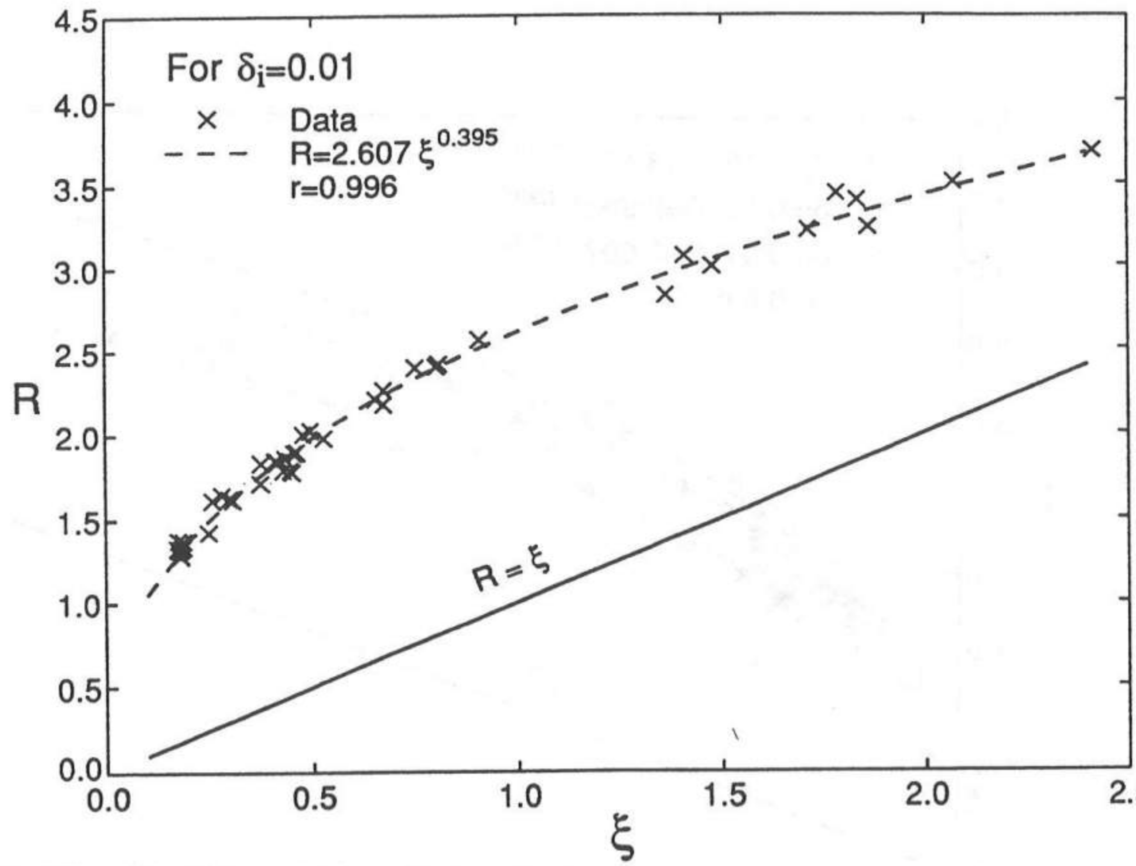

FIGURE 12. Relationship between R and ξ for 43 Tests with δ_i = 0.01.

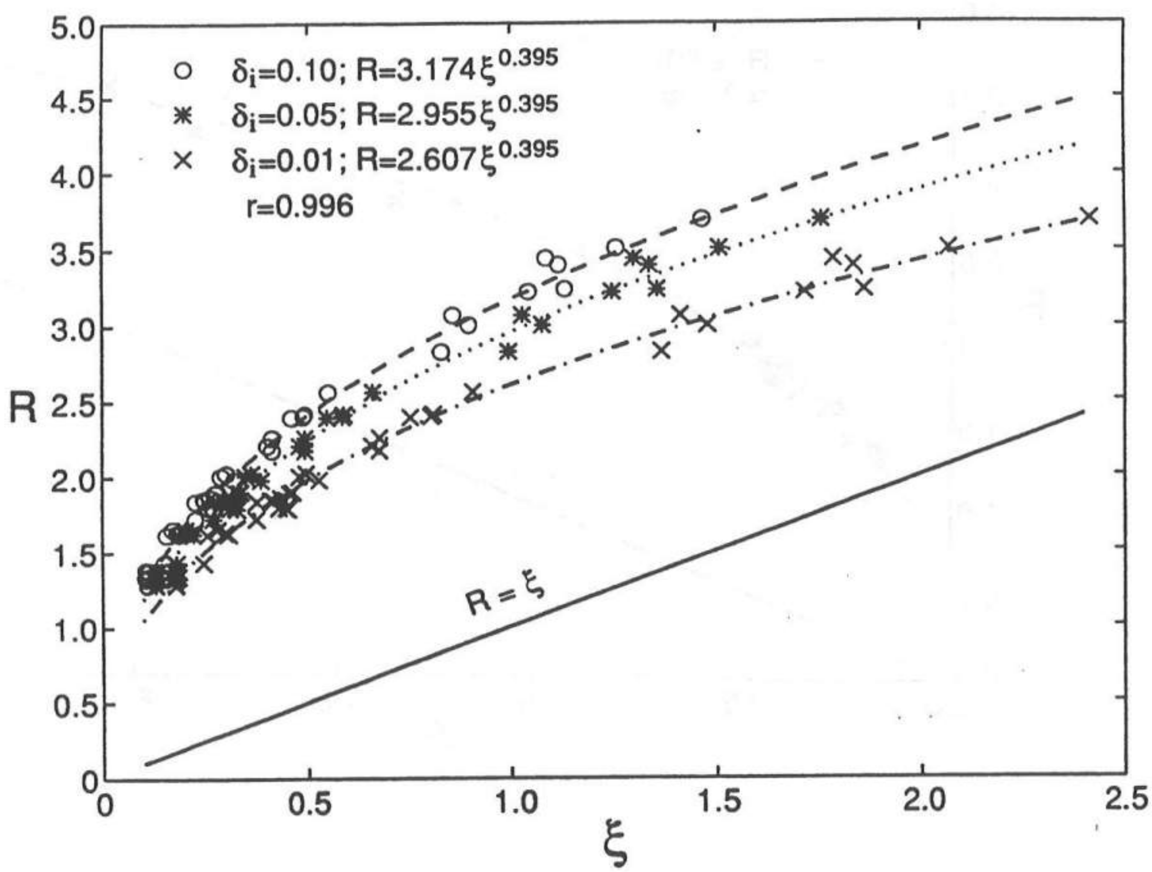

FIGURE 13. Relationship between R and ξ for 43 Tests with δ_i = 0.1, 0.05 and 0.01.

of Eq. 94 is no better than the simpler relationship given by Eq. 89 for these 43 tests with $\cot \theta' = 19.85$. Nevertheless, the use of the surf similarity parameter may make Eq. 94 applicable to smooth slopes with different gradients.

TABLE 4. Fitted Values of a and b in Runup Relationship $R = a\,\xi^b$ for 43 Tests with $\delta_i = 0.1$, 0.05 and 0.01.

δ_i	a	b	r	c	$1.109/c^{0.418}$
0.10	3.174	0.395	0.996	0.0802	3.184
0.05	2.955	0.395	0.996	0.0961	2.952
0.01	2.607	0.395	0.996	0.1320	2.585

In order to examine whether Eq. 94 is applicable for different values of $\cot \theta'$, the solitary wave runup data of Hall and Watts (1953) for which $\cot \theta' = 1.00 - 11.43$ is analyzed in the same way as the data of Synolakis (1987a) listed in Tables 2 and 3. The following data analysis is limited to the case of $\delta_i = 0.05$ since the degree of agreement in Fig. 13 is independent of the specific value of δ_i. Table 5 lists the dimensionless parameters for the 177 runs of Hall and Watts (1953) where the term "run" is used herein to differentiate their experiments from those of Synolakis (1987a). In Table 5, solitary waves are simply assumed to be breaking for $\xi < 2$ and nonbreaking for $\xi > 2$. The ranges of the dimensionless parameters listed at the end of Table 5 may be compared with those listed in Tables 2 and 3 as well as in Table 6 which lists the dimensionless parameters for the 34 nonbreaking solitary wave tests of Synolakis (1987a).

Fig. 14 plots the normalized runup R as a function of the surf similarity parameter ξ for all the solitary wave runup data analyzed herein. Fig. 14 indicates that the surf similarity parameter ξ alone is not sufficient for

TABLE 5. Solitary Wave Runup Data of Hall and Watts (1953) with $\delta_i = 0.05$.

Run No	d_t	σ	$\tan\theta'$	ξ	R	Condition
41	13.8889	67.4810	0.1763	4.7469	3.1852	non-breaking
42	20.0893	98.6342	0.1763	6.9383	2.9375	non-breaking
43	6.6225	31.0525	0.1763	2.1844	3.5762	non-breaking
44	8.2645	39.2666	0.1763	2.7622	2.9174	non-breaking
45	10.9890	52.9246	0.1763	3.7229	2.7967	non-breaking
46	11.6279	56.1304	0.1763	3.9485	2.9593	non-breaking
47	17.8571	87.4161	0.1763	6.1492	2.7500	non-breaking
48	5.9932	27.9099	0.1763	1.9633	3.3185	breaking
49	7.5431	35.6557	0.1763	2.5082	3.0603	non-breaking
50	9.1146	43.5254	0.1763	3.0618	4.8802	non-breaking
51	9.6154	46.0356	0.1763	3.2383	2.8407	non-breaking
52	10.1744	48.8386	0.1763	3.4355	2.9884	non-breaking
53	15.6250	76.2010	0.1763	5.3603	2.9464	non-breaking
54	5.6180	26.0387	0.1763	1.8317	3.3296	breaking
55	5.9524	27.7064	0.1763	1.9490	3.5278	breaking
56	7.8125	37.0039	0.1763	2.6030	3.3698	non-breaking
57	10.5634	50.7894	0.1763	3.5727	3.2183	non-breaking
58	11.3636	54.8043	0.1763	3.8552	3.4621	non-breaking
59	14.7059	71.5842	0.1763	5.0355	2.9804	non-breaking
60	4.7710	21.8222	0.1763	1.5351	3.2023	breaking
61	6.1881	28.8830	0.1763	2.0318	3.1733	non-breaking
62	8.8028	41.9632	0.1763	2.9519	3.3592	non-breaking
63	13.5870	65.9648	0.1763	4.6402	3.1848	non-breaking
64	3.8911	17.4587	0.1763	1.2281	3.2529	breaking
65	3.9683	17.8406	0.1763	1.2550	3.2024	breaking
66	7.0423	33.1505	0.1763	2.3319	3.4366	non-breaking
67	7.2993	34.4358	0.1763	2.4224	3.4526	non-breaking
68	9.8039	46.9808	0.1763	3.3048	6.6569	non-breaking
69	12.1951	58.9772	0.1763	4.1487	3.1829	non-breaking
70	3.3784	14.9285	0.1763	1.0501	3.2387	breaking
71	4.1209	18.5962	0.1763	1.3081	3.2143	breaking
72	6.1475	28.6804	0.1763	2.0175	3.4836	non-breaking
73	9.1463	43.6846	0.1763	3.0730	3.0610	non-breaking
74	2.3585	9.9424	0.1763	0.6994	2.9481	breaking
75	3.0864	13.4934	0.1763	0.9492	2.8395	breaking
76	4.9020	22.4734	0.1763	1.5809	3.0588	breaking
77	7.4627	35.2533	0.1763	2.4799	3.6269	non-breaking
78	7.4503	35.1915	0.2679	3.7618	2.8079	non-breaking
79	9.6983	46.4511	0.2679	4.9655	2.7759	non-breaking
80	12.3626	59.8180	0.2679	6.3943	2.6154	non-breaking

TABLE 5. (Continued)

Run No	d_t	σ	$\tan\theta'$	ξ	R	Condition
81	13.0814	63.4264	0.2679	6.7800	2.5988	non-breaking
82	13.8889	67.4810	0.2679	7.2135	2.6852	non-breaking
83	18.4426	90.3583	0.2679	9.6590	3.1803	non-breaking
84	20.0893	98.6342	0.2679	10.5436	3.4643	non-breaking
85	6.4103	29.9922	0.2679	3.2061	2.8878	non-breaking
86	7.9365	37.6246	0.2679	4.0219	2.6706	non-breaking
87	10.9890	52.9246	0.2679	5.6574	2.6319	non-breaking
88	16.3934	80.0616	0.2679	8.5583	2.5492	non-breaking
89	5.4348	25.1257	0.2679	2.6858	2.8727	non-breaking
90	7.2314	34.0964	0.2679	3.6448	2.8347	non-breaking
91	9.8870	47.3974	0.2679	5.0666	2.7797	non-breaking
92	10.1744	48.8386	0.2679	5.2207	2.9360	non-breaking
93	14.3443	69.7679	0.2679	7.4579	2.5492	non-breaking
94	5.3191	24.5497	0.2679	2.6243	2.9823	non-breaking
95	6.7568	31.7233	0.2679	3.3911	2.9910	non-breaking
96	9.2593	44.2505	0.2679	4.7302	2.8333	non-breaking
97	14.7059	71.5842	0.2679	7.6521	2.6275	non-breaking
98	4.4326	20.1418	0.2679	2.1531	3.1206	non-breaking
99	5.6306	26.1018	0.2679	2.7902	2.9730	non-breaking
100	7.2674	34.2766	0.2679	3.6640	2.7093	non-breaking
101	12.2549	59.2773	0.2679	6.3365	2.2353	non-breaking
102	4.1322	18.6524	0.2679	1.9939	3.0537	breaking
103	4.9505	22.7147	0.2679	2.4281	2.9752	non-breaking
104	7.0423	33.1505	0.2679	3.5437	3.0986	non-breaking
105	9.8039	46.9808	0.2679	5.0221	2.6667	non-breaking
106	3.2328	14.2121	0.2679	1.5192	3.2888	breaking
107	4.1209	18.5962	0.2679	1.9879	3.1264	breaking
108	5.2817	24.3631	0.2679	2.6043	2.9648	non-breaking
109	9.7403	46.6616	0.2679	4.9880	2.8571	non-breaking
110	2.1552	8.9603	0.2679	0.9578	3.0991	breaking
111	3.2895	14.4910	0.2679	1.5490	3.1974	breaking
112	4.4643	20.2989	0.2679	2.1699	3.2321	non-breaking
113	4.9020	22.4734	0.2679	2.4023	3.0490	non-breaking
114	8.0645	38.2654	0.2679	4.0904	3.1290	non-breaking
115	6.9876	32.8771	0.4663	6.1161	2.3323	non-breaking
116	8.9286	42.5933	0.4663	7.9236	2.2619	non-breaking
117	12.3626	59.8180	0.4663	11.1279	2.1484	non-breaking
118	18.4426	90.3583	0.4663	16.8093	2.0820	non-breaking
119	6.4103	29.9922	0.4663	5.5794	2.5064	non-breaking
120	8.2645	39.2666	0.4663	7.3048	2.3554	non-breaking

TABLE 5. (Continued)

Run No	d_t	σ	$\tan\theta'$	ξ	R	Condition
121	11.6279	56.1304	0.4663	10.4419	2.3314	non-breaking
122	16.3934	80.0616	0.4663	14.8938	2.0820	non-breaking
123	6.6794	31.3367	0.4663	5.8296	3.0649	non-breaking
124	7.5431	35.6557	0.4663	6.6330	2.4569	non-breaking
125	9.6154	46.0356	0.4663	8.5640	2.2637	non-breaking
126	14.3443	69.7679	0.4663	12.9789	2.0820	non-breaking
127	5.3191	24.5497	0.4663	4.5670	2.6241	non-breaking
128	7.0755	33.3166	0.4663	6.1979	2.4434	non-breaking
129	9.8684	47.3042	0.4663	8.8000	2.3618	non-breaking
130	13.3929	64.9902	0.4663	12.0901	3.1607	non-breaking
131	4.5956	20.9508	0.4663	3.8975	2.5625	non-breaking
132	6.1881	28.8830	0.4663	5.3731	2.5099	non-breaking
133	8.2237	39.0624	0.4663	7.2668	2.2961	non-breaking
134	4.5045	20.4985	0.4663	3.8133	2.7568	non-breaking
135	5.8140	27.0159	0.4663	5.0258	2.7035	non-breaking
136	8.9286	42.5933	0.4663	7.9236	2.6429	non-breaking
137	13.8889	67.4810	0.4663	12.5535	2.6389	non-breaking
138	3.5377	15.7138	0.4663	2.9232	2.8396	non-breaking
139	4.6296	21.1199	0.4663	3.9289	2.7407	non-breaking
140	6.6964	31.4219	0.4663	5.8454	2.6429	non-breaking
141	10.4167	50.0536	0.4663	9.3115	2.0556	non-breaking
142	1.7730	7.1320	0.4663	1.3268	3.2199	breaking
143	2.9070	12.6140	0.4663	2.3466	3.1919	non-breaking
144	4.0984	18.4847	0.4663	3.4387	3.1148	non-breaking
145	6.9444	32.6615	0.4663	6.0760	2.3472	non-breaking
146	6.7771	31.8250	1.0000	12.6964	1.9157	non-breaking
147	9.2975	44.4422	1.0000	17.7299	1.9008	non-breaking
148	13.0814	63.4264	1.0000	25.3035	1.8488	non-breaking
149	18.4426	90.3583	1.0000	36.0478	1.7377	non-breaking
150	6.4103	29.9922	1.0000	11.9651	2.0962	non-breaking
151	8.2645	39.2666	1.0000	15.6651	2.1198	non-breaking
152	11.6279	56.1304	1.0000	22.3928	2.0581	non-breaking
153	16.3934	80.0616	1.0000	31.9399	1.8852	non-breaking
154	5.7947	26.9199	1.0000	10.7395	2.1656	non-breaking
155	7.5431	35.6557	1.0000	14.2246	2.0603	non-breaking
156	10.8025	51.9888	1.0000	20.7405	1.8580	non-breaking
157	15.6250	76.2010	1.0000	30.3998	2.0536	non-breaking
158	5.3191	24.5497	1.0000	9.7939	2.2553	non-breaking
159	7.2464	34.1713	1.0000	13.6324	2.0483	non-breaking
160	9.8684	47.3042	1.0000	18.8716	2.0921	non-breaking

TABLE 5. (Continued)

Run No	d_t	σ	$\tan\theta'$	ξ	R	Condition
161	15.4639	75.3918	1.0000	30.0770	2.0103	non-breaking
162	4.7710	21.8222	1.0000	8.7058	2.3626	non-breaking
163	6.5104	30.4925	1.0000	12.1647	2.2083	non-breaking
164	9.4697	45.3053	1.0000	18.0742	2.2803	non-breaking
165	12.2549	59.2773	1.0000	23.6482	2.0784	non-breaking
166	4.1322	18.6524	1.0000	7.4412	2.4835	non-breaking
167	4.7170	21.5538	1.0000	8.5987	2.1698	non-breaking
168	8.5470	40.6817	1.0000	16.2296	2.2650	non-breaking
169	14.9254	72.6866	1.0000	28.9978	2.3731	non-breaking
170	3.3784	14.9285	1.0000	5.9556	2.7072	non-breaking
171	4.1209	18.5962	1.0000	7.4188	2.5275	non-breaking
172	7.3529	34.7043	1.0000	13.8450	2.4216	non-breaking
173	10.4167	50.0536	1.0000	19.9685	2.4583	non-breaking
174	1.9841	8.1387	1.0000	3.2469	2.5952	non-breaking
175	3.2895	14.4910	1.0000	5.7811	2.5592	non-breaking
176	6.9444	32.6615	1.0000	13.0301	2.9444	non-breaking
177	7.4627	35.2533	1.0000	14.0640	2.3731	non-breaking
max	21.7391	106.9271	1.0000	36.0478	6.6569	
min	1.7730	7.1320	0.0875	0.3291	1.7377	

TABLE 6. Nonbreaking Solitary Wave Runup Data of Synolakis (1987a) with $\delta_i = 0.05$.

Test No	d_t	σ	$\tan\theta'$	ξ	R
44	27.7778	137.2869	0.0504	2.7592	3.4444
45	27.7778	137.2869	0.0504	2.7592	3.3611
46	20.8333	102.3740	0.0504	2.0575	3.7917
47	25.6410	126.5433	0.0504	2.5432	3.8974
48	25.0000	123.3204	0.0504	2.4785	3.9000
49	47.6190	237.0715	0.0504	4.7646	3.6190
50	71.4286	356.8326	0.0504	7.1716	3.5000
51	19.6078	96.2144	0.0504	1.9337	3.7451
52	18.1818	89.0477	0.0504	1.7897	3.7636
55	29.4118	145.5031	0.0504	2.9243	4.2353
54	55.5556	276.9907	0.0504	5.5669	4.1111
55	111.1111	556.4465	0.0504	11.1834	4.0000
56	55.5556	276.9907	0.0504	5.5669	4.1667
57	37.0370	183.8495	0.0504	3.6950	4.0000
58	26.3158	129.9359	0.0504	2.6114	3.8421
59	21.2766	104.6021	0.0504	2.1023	4.1489
60	21.2766	104.6021	0.0504	2.1023	4.1489
61	52.6316	262.2834	0.0504	5.2713	4.1053
62	52.6316	262.2834	0.0504	5.2713	4.0000
63	111.1111	556.4465	0.0504	11.1834	4.5556
64	200.0000	1003.5950	0.0504	20.1701	3.8000
65	166.6667	835.9135	0.0504	16.8001	3.6667
66	142.8571	716.1414	0.0504	14.3929	3.7143
67	35.7143	177.1972	0.0504	3.5613	4.3929
68	125.0000	626.3127	0.0504	12.5875	3.6250
69	43.4783	216.2449	0.0504	4.3461	3.7826
70	58.8235	293.4283	0.0504	5.8973	3.7059
71	41.6667	207.1335	0.0504	4.1629	4.0833
72	83.3333	416.7159	0.0504	8.3751	4.0000
73	71.4286	356.8326	0.0504	7.1716	3.7143
74	111.1111	556.4465	0.0504	11.1834	4.0000
75	22.7273	111.8945	0.0504	2.2488	4.1364
76	45.4545	226.1848	0.0504	4.5458	4.4545
77	25.6410	126.5433	0.0504	2.5432	4.1538
max	200.0000	1003.5950	0.0504	20.1701	4.5556
min	18.1818	89.0477	0.0504	1.7897	3.3611

predicting R for nonbreaking solitary waves in the range of ξ exceeding about two. This may also be shown using Eq. 91 where $\cot \theta' = \sigma/[\xi(2\pi)^{1/2}]$ and σ is approximately proportional to d_t. This finding is also consistent with the empirical formula by Ahrens and Martin for the normalized runup R of nonbreaking monochromatic waves, which is not based on the surf similarity parameter. Fig. 15 shows the data points in Fig. 14 in the range of $\xi < 5$ together with the empirical relationship given by Eq. 94 with $\delta_i = 0.05$ which has been based on the 43 tests of Synolakis (1987a) in the range $0.125 < \xi < 1.757$. The data points of Hall and Watts (1953) in this range of ξ tend to follow the empirical formula, $R = a\,\xi^b$, for breaking or broken solitary wave runup. Consequently, this empirical formula may be applied to predict breaking or broken solitary wave runup on smooth uniform slopes of arbitrary gradient.

4.3 Computed Solitary Wave Runup for Selected Tests

The numerical model described in Section 2 is compared with the broken solitary wave runup data of Synolakis (1987a) listed in Tables 2 and 3. It is noted that the solitary wave generation technique employed by Hall and Watts (1953) may not have produced near-perfect solitary waves. Table 7 lists some of the input and computation parameters for the 43 tests from which 9 tests are selected for the subsequent computation and comparison. In Table 7, $d_t' =$ water depth below SWL at the toe of the 1:19.85 slope where the incident solitary wave profile is specified as input to the numerical model; $H_r' =$ representative wave height taken as the incident solitary wave height H'; $T_r' =$ representative wave period computed using Eq. 39 with $\delta_i = 0.05$; and $R' =$ measured runup height above SWL. The values of these parameters for each test

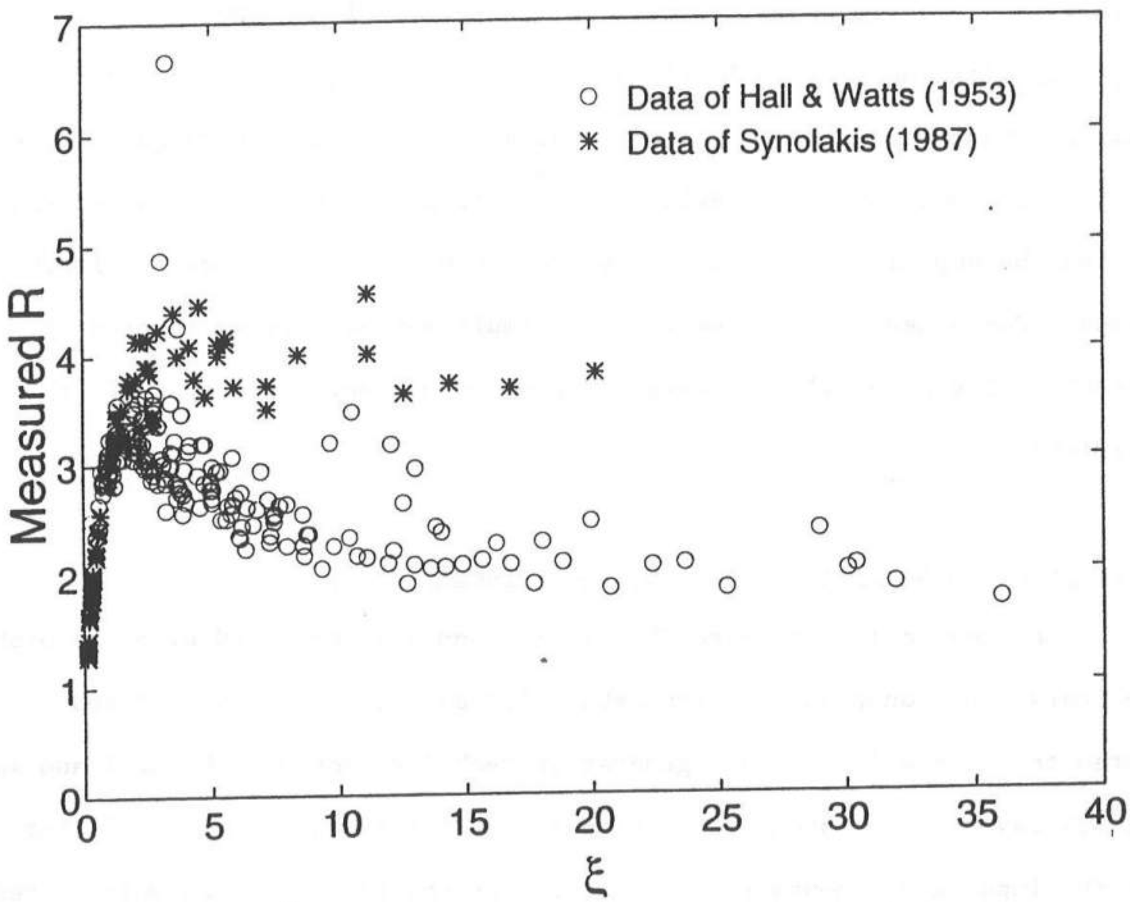

FIGURE 14. Normalized Runup R as a Function of Surf Similarity Parameter ξ for Entire Range of ξ.

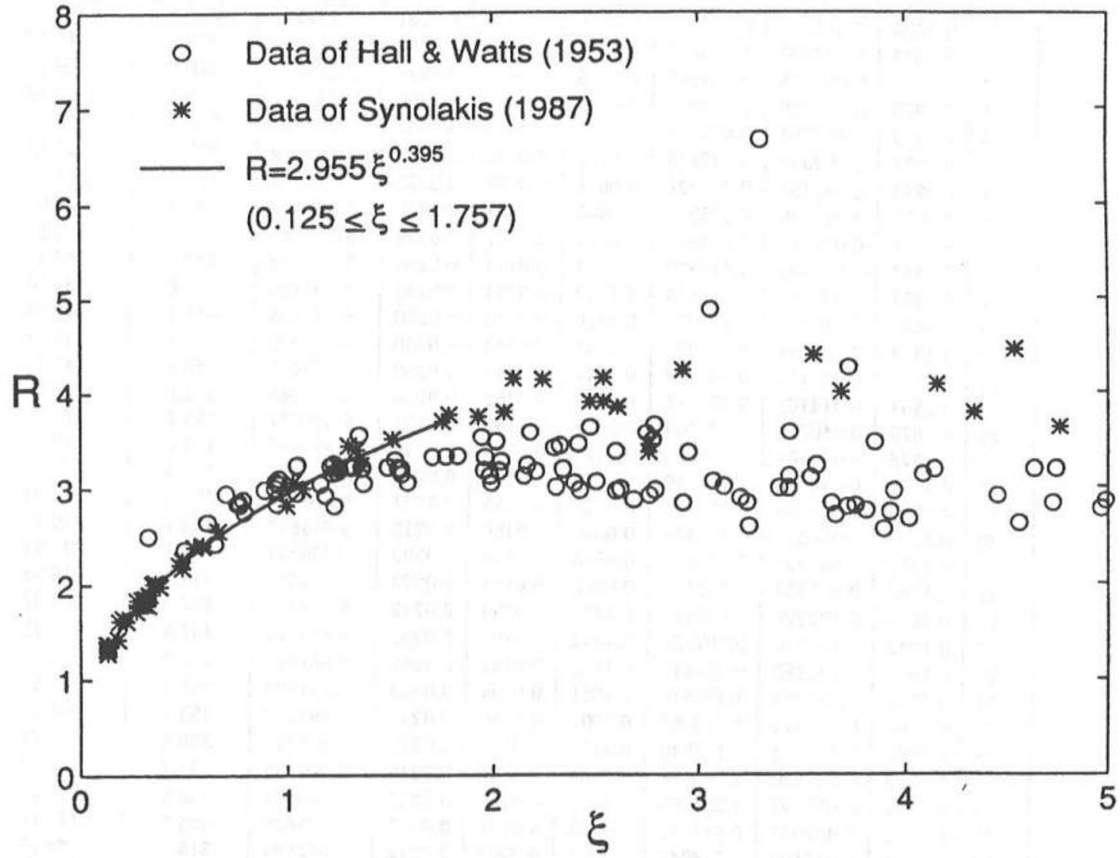

FIGURE 15. Normalized Runup R as a Function of Surf Similarity Parameter ξ for $\xi < 5$.

TABLE 7. Input and Computational Parameters for 43 Tests by Synolakis (1987a).

Test No	d_t' (m)	H_r' (m)	T_r' (sec)	R' (m)	$\Delta x'$ (m)	Δx	WBSEG (m)	WBSEG/ $\Delta x'$	$(\mathrm{NONE})_{min}$	t_{max}
1	0.0625	0.015625	0.718277	0.0316	0.0062	0.0221	2.182259	351.8	220.07	6.4117
2	0.0625	0.004500	1.445285	0.0146	0.0062	0.0204	1.674223	269.9	374.42	4.1925
3	0.0801	0.035885	0.564378	0.0579	0.0079	0.0237	3.314324	416.9	169.99	8.3563
4	0.0979	0.007636	1.733052	0.0246	0.0097	0.0205	2.674973	275.3	361.83	4.2884
5	0.0979	0.037594	0.689342	0.0608	0.0097	0.0232	3.753513	386.3	182.22	7.7532
6	0.0981	0.009516	1.542135	0.0269	0.0097	0.0207	2.747619	282.2	329.73	4.5744
7	0.0984	0.045461	0.613027	0.0648	0.0098	0.0239	3.884018	397.7	167.62	8.4860
8	0.0989	0.023340	0.935212	0.0462	0.0098	0.0219	3.338362	340.1	225.62	6.2623
9	0.1317	0.038720	0.944994	0.0714	0.0131	0.0224	4.739626	362.6	205.07	6.8677
10	0.1454	0.088694	0.617990	0.1134	0.0144	0.0250	6.263032	434.0	147.40	9.8209
11	0.1454	0.085931	0.631583	0.1149	0.0144	0.0249	6.306325	437.0	149.60	9.6526
12	0.1454	0.088258	0.620093	0.1170	0.0144	0.0250	6.371265	441.5	147.74	9.7944
13	0.1454	0.088258	0.620093	0.1134	0.0144	0.0250	6.263032	434.0	147.74	9.7944
14	0.1550	0.093155	0.644629	0.1242	0.0154	0.0250	6.773465	440.3	148.43	9.7413
15	0.1567	0.014103	2.029911	0.0423	0.0156	0.0206	4.370245	281.0	340.41	4.4718
16	0.1572	0.040715	1.115169	0.0816	0.0156	0.0221	5.549667	355.7	216.72	6.5065
17	0.1576	0.092984	0.658310	0.1277	0.0156	0.0249	6.929317	443.0	149.72	9.6437
18	0.1562	0.046548	1.020639	0.0861	0.0155	0.0225	5.663191	365.3	203.86	6.9082
19	0.1565	0.050393	0.973847	0.0925	0.0155	0.0227	5.860459	377.3	197.03	7.1490
20	0.1569	0.026673	1.426498	0.0639	0.0156	0.0213	5.015846	322.1	259.72	5.5196
21	0.1670	0.045591	1.113371	0.0813	0.0166	0.0223	5.736521	346.1	211.81	6.6524
22	0.1753	0.048383	1.133152	0.0868	0.0174	0.0223	6.063386	348.5	210.81	6.6834
23	0.1942	0.122929	0.696156	0.1635	0.0193	0.0252	8.723571	452.6	144.85	10.0237
24	0.1942	0.121375	0.702320	0.1602	0.0193	0.0252	8.625272	447.5	145.72	9.9533
25	0.1947	0.121882	0.702445	0.1678	0.0193	0.0252	8.861976	458.6	145.61	9.9621
26	0.1956	0.055355	1.178840	0.1031	0.0194	0.0223	6.951903	358.1	208.52	6.7554
27	0.1962	0.056113	1.173067	0.1007	0.0195	0.0224	6.891442	353.9	207.56	6.7861
28	0.2080	0.067184	1.120540	0.1154	0.0206	0.0227	7.566026	366.5	196.77	7.1589
29	0.2092	0.039330	1.554428	0.0856	0.0208	0.0215	6.700253	322.7	248.79	5.7296
30	0.2092	0.056693	1.251700	0.1073	0.0208	0.0222	7.348061	353.9	212.49	6.6317
31	0.2092	0.087027	0.957149	0.1435	0.0208	0.0235	8.425666	405.8	175.78	8.0570
32	0.2101	0.033406	1.714943	0.0807	0.0209	0.0212	6.572684	315.2	267.23	5.3878
33	0.2144	0.034304	1.726237	0.0823	0.0213	0.0212	6.707204	315.2	266.52	5.3999
34	0.2147	0.030702	1.840780	0.0786	0.0213	0.0211	6.601521	309.8	279.58	5.1906
35	0.2349	0.092551	1.050362	0.1506	0.0233	0.0233	9.146013	392.3	180.13	7.8487
36	0.2638	0.070435	1.418308	0.1337	0.0262	0.0222	9.218736	352.1	213.87	6.5901
37	0.2940	0.022050	3.067014	0.0759	0.0292	0.0205	8.094394	277.4	367.94	4.2409
38	0.2954	0.021564	3.119040	0.0733	0.0293	0.0204	8.044983	274.4	372.22	4.2087
39	0.2962	0.019253	3.322294	0.0675	0.0294	0.0204	7.890383	268.4	391.19	4.0764
40	0.2972	0.016643	3.600604	0.0615	0.0295	0.0203	7.731190	262.1	417.07	3.9191
41	0.3093	0.058148	1.890078	0.1315	0.0307	0.0215	10.053603	327.5	248.79	5.7296
42	0.3138	0.029497	2.805633	0.0904	0.0311	0.0206	8.919828	286.4	334.16	4.5308
43	0.3535	0.068226	1.990090	0.1506	0.0351	0.0215	11.500822	327.8	246.01	5.7868
max	0.3535	0.122929	3.600604	0.1678	0.0351	0.0252	11.500822	458.6	417.07	10.0237
min	0.0625	0.004500	0.564378	0.0146	0.0062	0.0203	1.674223	262.1	144.85	3.9191

are used to estimate appropriate values of computational parameters as described below.

The computer program SBREAK requires the specification of the integer S as input where S determines the number of spatial nodes from the seaward boundary to the wet node next to the initial shoreline at SWL for the computation of wave runup corresponding to IJOB=1. The physical grid size $\Delta x'$ for the uniform slope with $\cot \theta' = 19.85$ is then given by $\Delta x' = (d_t' \cot \theta')/S$. The corresponding normalized grid size Δx is computed as $\Delta x = \Delta x'/[T_r'(gH_r')^{1/2}]$. The values of $\Delta x'$ and Δx for the case of S = 200 are listed in Table 7. To provide a sufficient spatial resolution for a breaking solitary wave, $\Delta x'$ should be of the order of its height H' or less and Δx should be much smaller than unity. The selection of S = 200 guessed from the range S = 100 - 400 used in previous computations appears to be reasonable. Our previous experiences also indicate that the computed results are essentially independent of S as long as S is sufficiently large.

The horizontal distance of the computation domain denoted by WBSEG in Table 7 needs to be specified as input to SBREAK. In Table 7, WBSEG = $(d_t' + 1.5\,R') \cot \theta'$ for each test where the factor 1.5 for R' is added to ensure that the computed runup will remain in the specified computation domain. This is required for the wave runup computation with IJOB=1. The total number of spational nodes in the computation domain is given by JE=[1+ integer (WBSEG/$\Delta x'$)]. The integer JE must not exceed the integer N1 where N1 = 500 in SBREAK. Table 7 indicates that this requirement is satisfied for all the tests.

The computer program SBREAK requires the specification of NONE $=1/\Delta t$ as input where NONE = even number of time steps in one representation wave period; and Δt = time step size normalized by the representative wave period.

The minimum value of NONE required for the numerical stability may be estimated using Eq. 88 with ALPHAS = 1

$$(\text{NONE})_{min} = \frac{1+\sqrt{d_t}}{\Delta x} \left[\left(1 + \frac{\epsilon^2}{4}\right)^{1/2} - \frac{\epsilon}{2} \right]^{-1} \tag{95}$$

where $d_t = d_t'/H_r'$; and ϵ = greatest coefficient of the numerical damping coefficients ϵ_1 and ϵ_2. The minimum value of NONE given by Eq. 95 with $\epsilon = 1$ is listed for each test in Table 7. Since the numerical stability criterion given by Eq. 27 does not consider the numerical stability at the moving shoreline, which tends to cause more numerical difficulties, the value of NONE is simply taken as NONE = 1000 as a first attempt.

It is required to estimate the duration t_{max} of the computation starting from the normalized time t = 0 when the incident solitary wave arrives at x = 0. The integer NTOP = (NONE) t_{max} is specified as input to SBREAK. The value of t_{max} for each test must be large enough to simulate at least the entire process of wave uprush and resulting runup. The value of t_{max} listed in Table 7 is given by $t_{max} = (2t_c + 2d_t' \cot \theta'/[T_r'\sqrt{gd_t'}\,])$ with the crest arrival time $t_c = 1$. The term $(2t_c)$ is the normalized time required for the passage of the incident solitary wave at x =0, while the other term is the travel time normalized by T_r' where the travel time is crudely estimated as the horizontal distance $(2d_t' \cot \theta')$ divided by the velocity $\sqrt{gd_t'}$. On the basis of the values of t_{max} listed in Table 7, NTOP is simply taken at NTOP = 10 (NONE).

For the calibration and verification of the numerical model, 9 tests are selected to represent the entire range of the surf similarity parameter ξ = 0.125 - 1.757 for the 43 tests plotted in Fig. 11 with $\delta_i = 0.05$. In the following, detailed computed results are presented for tests 23, 32 and 40 corresponding to ξ = 0.125, 0.591 and 1.757, respectively. The input and output for test 32 are described in detail to allow users of SBREAK to get familiarized with the input and output procedures, although the following example is limited to solitary wave runup. Other examples associated with different options provided by SBREAK were given in the report for IBREAK by Kobayashi and Wurjanto (1989c). For each of the selected 9 tests, the bottom friction factor f' = 0.01 and 0.005 are assumed. The value of f' = 0.01 was the lower bound for the plywood slope used by Kobayashi and Watson (1987). The aluminum slope used by Synolakis (1987a) must have been smoother than the plywood slope. As a result, the value of f' = 0.01 has been employed first but resulted in slight underprediction of the normalized runup R. Subsequently, the value of f' = 0.005 has been specified as input in the hope of better agreement between the measured and computed values of R.

Table 8 shows the primary input data file, FINP1, for test 32 with f' = 0.005. The input parameters and variables listed in Table 8 are explained in sequence where Section 3.4 has described the input required for all the options included in SBREAK. The number of the comment lines proceeding the input data is 3 in Table 8.

TABLE 8. Primary Input Data File, FINP1, for Test 32 with Bottom Friction Factor f' = 0.005.

```
         3
------------------------------------------------------------------------
FILE syno32:  Synolakis (1987)   Test 32  FWP=0.0050
------------------------------------------------------------------------
10   10001                       --> IJOB,ISTAB,NSTAB
1                                --> ISYST (SI units)
1                                --> IBOT (width-slope representation)
0                                --> INONCT
0                                --> IENERG
3     no wave data file          --> IWAVE,FINP2
100    1000  19   0   0          --> ISAVA-B-C,NSAVA,NTIMES,NNOD1,NNOD2
00000     0                      --> IREQ,IELEV,IV,IDUDT,ISNR,NNREQ
   10000                         --> NTOP
    1000                         --> NONE
      10                         --> NJUM1
     200                         --> S
      .005000                    --> FWP
     1.000000      1.000000      --> X1,X2
      .001000                    --> DELTA (normalized)
       3                         --> NDELR
      .5                         --> DELRP(1)  (mm)
     1.0                         --> DELRP(2)
     5.0                         --> DELRP(3)
     0.033406      1.714943      --> HREFP(meters),TP(seconds)  (will change)
     1.000000      1.000000      --> KSREF,KSSEA
     0.210100                    --> DSEAP(meters)
      .050378                    --> TSLOPS
       1                         --> NBSEG
     6.572684       .050378      --> WBSEG(1)  (meters),TBSLOP(1)
```

- IJOB=1 for wave runup computation.
- ISTAB=0 for no computation of armor stability or movement.
- NSTAB=(NTOP+1) set for no computation of armor stability or movement.
- ISYST=1 for the metric system.
- IBOT=1 for specifying the width and slope of a linear segment of the bottom geometry.
- INONCT=0 for solitary waves where the nonlinear correction term C_t in Eq. 13 and given by Eq. 42 may not be necessary for solitary waves.
- IENERG=0 for no computation of the quantities related to wave energy.
- IWAVE=3 for the incident solitary wave profile computed using Eq. 36 with t_c = 1 and δ_i = 0.05. No wave data file is hence required.
- ISAVA=1 for storing the spatial variations of the normalized free surface elevation η and the normalized depth-averaged velocity u at specified time levels.
- ISAVB=0 for no storage of the temporal variation of the normalized water depth h at specified nodes.
- ISAVC=0 for no storage of the armor displacement.
- NSAVA=NONE set for the time level at the beginning of the storage of the spatial variations of η and u where NSTAT=NSAVA for IWAVE=3 and the statistical computation starts at t=1 when the crest of the incident solitary wave arrives at x=0.
- NTIMES = [1 +(NTOP - NONE)/(NONE/2)] set for storing the spatial variations of η and u at t = 1.0, 1.5, 2.0, ..., t_{max} where t_{max} = NTOP/NONE.
- NNOD1=0 for ISAVB=0.
- NNOD2=0 for ISAVC=0.
- IREQ=0 for no special storage as explained in the common /IDREQ/.

- IELEV=0 for IREQ=0.
- IV=0 for IREQ=0.
- IDUDT=0 for IREQ=0.
- ISNR=0 for IREQ=0.
- NNREQ=0 for IREQ=0.
- NTOP=10(NONE) set on the basis of the values of t_{max} listed in Table 7.
- NONE=1000 set on the basis of the values of $(NONE)_{min}$ given by Eq. 95 and listed in Table 7.
- NJUM1=NONE/100 set for storing the computed temporal variations at the seaward and landward boundaries at the rate of 100 points over one representative wave period throughout the time-marching computation.
- S=200 set to provide an adequate spatial resolution as explained in relation to the values of Δx listed in Table 7.
- FWP=0.005 for the bottom friction factor $f' = 0.005$ for the smooth slope used by Synolakis (1987a).
- X1=1.0 for the numerical damping coefficient ϵ_1=1.0.
- X2=1.0 for the numerical damping coefficient ϵ_2=1.0.
- DELTA=0.001 set for the normalized water depth δ used to define the computational shoreline on the slope.
- NDELR=3 set for the number of different values of the physical water depth δ_r' associated with the measured shoreline for which the normalized free surface elevation Z_r is computed as discussed in Section 2.5 where it is required that $\delta_r' > (H_r'\delta)$
- DELRP(1) = 0.5 for δ_r' = 0.5 mm.
- DELRP(2) = 1.0 for δ_r' = 1.0 mm, corresponding to the runup measurement by Synolakis (1987a).
- DELRP(3) = 5.0 for δ_r' = 5.0 mm.

- HREFP = 0.033406 for H_r' = 0.033406 m for test 32 in Table 7.
- TP = 1.714943 for T_r' = 1.714943 sec for test 32 in Table 7 where the value of T_r' computed using Eq. 39 is used in SBREAK, so any value of TP can be specified as input for IWAVE = 3.
- KSREF=1 for $H' = H_r'$ and $K_s = 1$ as explained in the common /WAVE2/.
- KSSEA=1 for $H' = H_r'$ and $K_s = 1$ as explained in the common /WAVE2/.
- DSEAP = 0.2101 for d_t' = 0.2101 m for test 32 in Table 7.
- TSLOPS = 0.050378 for the uniform slope $\tan\theta' = 1/19.85$ used by Synolakis (1987a) where TSLOPS is the slope used to define the surf similarity parameter for composite slopes.
- NBSEG=1 for a uniform slope consisting of one linear segment.
- WBSEG(1) = 6.572684 for WBSEG = 6.572684 m for test 32 listed in Table 7.
- TBSLOP(1) = 0.050378 for $\tan\theta' = 1/19.85$ used by Synolakis (1987a).

As for the output of SBREAK, Section 3.6 has described the output for all the options. Table 9 shows the contents of the essential output for the concise documentation stored in the file ODOC. This file is normally used to check whether there is any error in the input as well as to obtain important quantities such as solitary wave runup for this computation. The runup, rundown and setup for given δ_r' listed in Table 9 are the maximum, minimum and mean of the normalized free surface elevation, $Z_r = Z_r'/H_r'$, during $1 \le t \le t_{max}$ where Z_r' is the elevation above SWL of the intersection between the free surface and the straight line parallel to the uniform slope at the distance of δ_r'. The computed runup is less sensitive to δ_r' than the computed rundown and setup in the vicinity of δ_r' = 1 mm corresponding to the runup measurement of Synolakis (1987a). Consequently, the runup measurement based on the specified

TABLE 9. Concise Documentation File, ODOC, for Test 32 with Bottom Friction Factor f' = 0.005.

```
-----------------------------------------------------------------
FILE syno32:  Synolakis (1987)   Test 32  FWP=0.0050
-----------------------------------------------------------------

WAVE CONDITION

Solitary Incident Wave at Seaward Boundary
Tc      =      0.100000000D+01
K2      =      0.435654442D+01
Norm. Maximum Surface Elev.     =       1.000000
Norm. Minimum Surface Elev.     =       0.000000

Reference Wave Period           =       1.714940 sec.
Reference Wave Height           =       0.033406 meters
Depth at Seaward Boundary       =       0.210100 meters
Shoal. Coef. at Reference  Ks1  =       1.000
             at Seaw. Bdr. Ks2  =       1.000
             Ks = Ks2/Ks1       =       1.000
Norm. Depth at Seaw. Bdr.       =       6.289
Normalized Wave Length          =      11.155
"Sigma"                         =      29.388
Ursell Number                   =      19.785
Surf Similarity Parameter       =       0.591

SLOPE PROPERTIES

Friction Factor               =    0.005000
Norm. Friction Factor         =    0.073470
Norm. Horiz. Length of
      Computation Domain      =    6.690675
Number of Segments            =    1
--------------------------------
 SEGMENT     WBSEG(I)   TBSLOP(I)
    I         meters
--------------------------------
     1       6.572684    0.050378

COMPUTATION PARAMETERS

Normalized Delta x    =    0.212402D-01
Normalized Delta t    =    0.100000D-02
Normalized DELTA      =    0.100000E-02
Damping Coeff.     x1 =    1.000
                   x2 =    1.000
Num. Stab. Indicator  =    3.742
Total Number of Time Steps          NTOP =     10000
Number of Time Steps in 1 Wave Period
                                    NONE =     1000
Total Number of Spatial Nodes         JE =     316
Number of Nodes Along Bottom Below SWL
                                       S =     200
Storing Temporal Variations at Every
                                   NJUM1 =      10 Time Steps

REFLECTION COEFFICIENTS

r1 =     0.328
r2 =     0.283
r3 =     0.240

RUNUP, RUNDOWN, SETUP

Largest Node Number Reached by Computational Waterline
                                            JMAX =      283
----------------------------------------------------------
   I    DELTAR(I)     RUNUP(I)    RUNDOWN(I)    SETUP(I)
          [mm]           R            Rd           Zr
----------------------------------------------------------
   1     0.500        2.577        -0.271        0.714
   2     1.000        2.546        -0.448        0.617
   3     5.000        2.228        -0.534        0.373

WAVE SET-DOWN OR SETUP

Average value of ETAI =     0.025557
                 ETAR =     0.053267
```

value of δ_r' is affected very little by small deviations of δ_r' during the runup measurement. For test 32 with f' = 0.005, the computed value of the normalized runup is R = 2.546 as compared to the measured runup R = 2.415 listed in Table 2.

Table 10 shows the contents of the file OMSG which stores the messages written during the computation as explained in Section 3.5. For test 32 with f' = 0.005, the adjustment for the case of $h_s^* > h_{s-1}^*$ was made seven times in step 4 for the computation of the shoreline movement in Section 2.5. This adjustment does not stop the computation but suggests the numerical difficulty at the moving shoreline.

The computed temporal and spatial variations of the normalized free surface elevation and depth-averaged velocity for test 32 with f' = 0.005 are examined in the following. Fig. 16 shows the incident solitary wave profile $\eta_i(t)$ given by Eq. 36 with $K_s = 1$, $t_c = 1$ and $\delta_i = 0.05$ as a function of the normalized time t. Fig. 16 also shows the normalized reflected wave profile $\eta_r(t)$ at the toe of the slope computed using Eq. 13 with $C_t = 0$. The temporal variation of $\eta_r(t)$ shown in Fig. 16 is similar to the measured variations plotted by Synolakis (1987b). It is noted that $\eta_i(t)$ and $\eta_r(t)$ are stored in the file OSEAWAV with its unit number = 21 where the stored time level N is related to the normalized time t = N/NONE.

Fig. 17 shows the computed temporal variations of the normalized shoreline elevation Z_r above SWL for the water depth $\delta_r' = 0.5$, 1.0 and 5.0 mm. The maximum and minimum values of Z_r are the normalized runup and rundown, respectively, listed in Table 9. During wave uprush, Z_r is not very sensitive to δ_r' because of the steep front of the uprushing wave. During wave downrush, Z_r is sensitive to δ_r' since a thin layer of downrushing water remains on the slope due to bottom friction. It is noted that the temporal variations of

TABLE 10. Contents of Message File, OMSG, for Test 32 with Bottom Friction Factor f′ = 0.005.

```
---------------------------------------------------------------------
FILE syno32:  Synolakis (1987)    Test 32  FWP=0.0050
---------------------------------------------------------------------

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     249; N =     8048
 Adjusted values: U(2,S) =   0.102E-02; U(2,S-1) =   0.105E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     248; N =     8237
 Adjusted values: U(2,S) =   0.983E-03; U(2,S-1) =   0.102E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     243; N =     8665
 Adjusted values: U(2,S) =   0.950E-03; U(2,S-1) =   0.102E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     240; N =     8825
 Adjusted values: U(2,S) =   0.100E-02; U(2,S-1) =   0.107E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     237; N =     9021
 Adjusted values: U(2,S) =   0.985E-03; U(2,S-1) =   0.108E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     233; N =     9307
 Adjusted values: U(2,S) =   0.108E-02; U(2,S-1) =   0.120E-02

 From Subroutine 14 RUNUP: U(2,S)>U(2,S-1) at S =     232; N =     9357
 Adjusted values: U(2,S) =   0.965E-03; U(2,S-1) =   0.109E-02
```

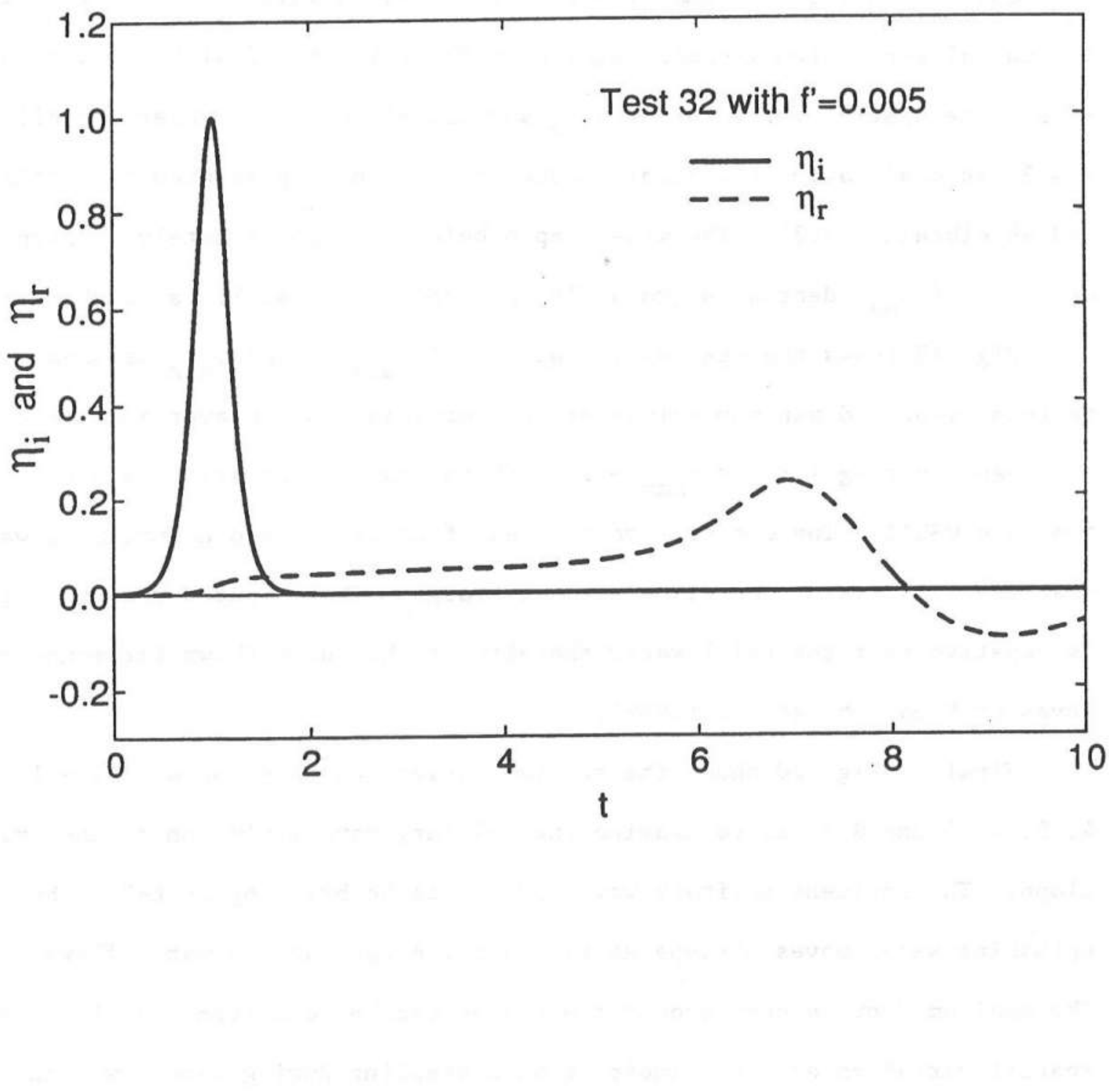

FIGURE 16. Specified Incident Solitary Wave Profile $\eta_i(t)$ and Computed Reflected Wave Profile $\eta_r(t)$ at x = 0 for Test 32 with f' = 0.005.

$Z_r(t)$ for the different values of δ_r' are stored in the file ORUNUP with its unit number = 31.

Fig. 18 shows the spatial variations of η_{max}, $\bar{\eta}$ and η_{min} defined as the maximum, mean and minimum values of the normalized free surface elevation η at given x during $1 \le t \le t_{max} = 10$. These spatial variations are stored in the file OSTAT with its unit number = 23. The solid straight line in Fig. 18 is the normalized bottom geometry stored in the file OSPACE with its unit number = 22. The spatial variation of η_{max} seaward of the still water shoreline at x = 4.25 is qualitatively similar to the analyzed data presented by Synolakis and Skjelbreia (1993). The water depth below the approximately straight envelope of η_{max} decreases gradually landward of the still water shoreline.

Fig. 19 shows the spatial variations of u_{max}, $\bar{u}$ and u_{min} defined as the maximum, mean and minimum values of the normalized depth-averaged velocity u at given x during $1 \le t \le t_{max} = 10$. These spatial variations are stored in the file OSTAT. The computed velocities of uprushing and downrushing water near the still water shoreline are very large. The computed mean velocity $\bar{u}$ is negative near the still water shoreline as has been shown for monochromatic waves by Kobayashi et al. (1989).

Finally, Fig. 20 shows the spatial variations of η and u at t = 1, 2, 3, 4, 5, 6, 7 and 8 so as to examine the solitary wave evolution on the uniform slope. The incident solitary wave appears to be breaking at t=2. The tip of uprushing water moves upslope at t=4 when the rest of the water flows seaward. The maximum runup occurs around t = 4.5 as can be seen from Fig. 17. The spatial variation at t = 6 indicate wave breaking during wave downrush. The wave action on the slope is very small at t = 8 and negligible at t = 10. It is noted that the computed spatial variations of η and u at specified time

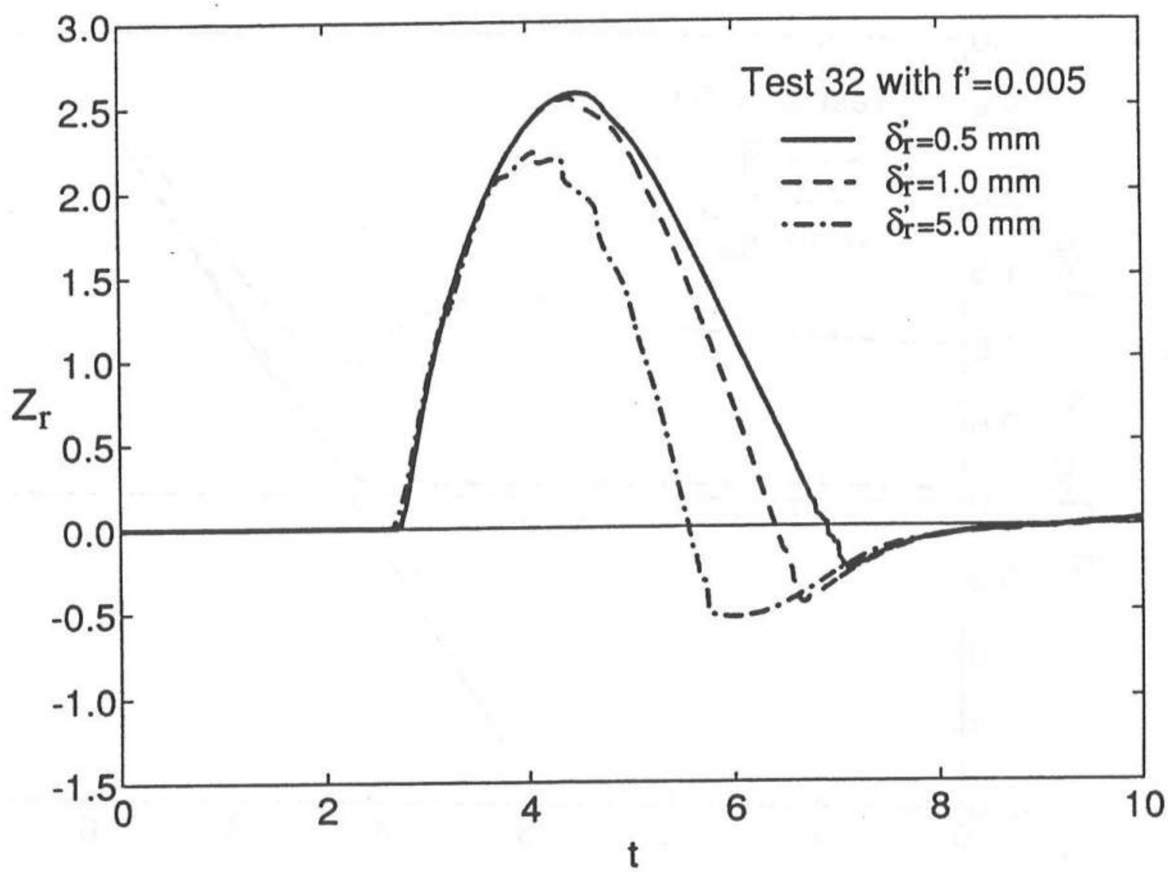

FIGURE 17. Temporal Variations of Normalized Shoreline Elevation Z_r above SWL for δ'_r = 0.5, 1.0 and 5.0 mm for Test 32 with f' = 0.005.

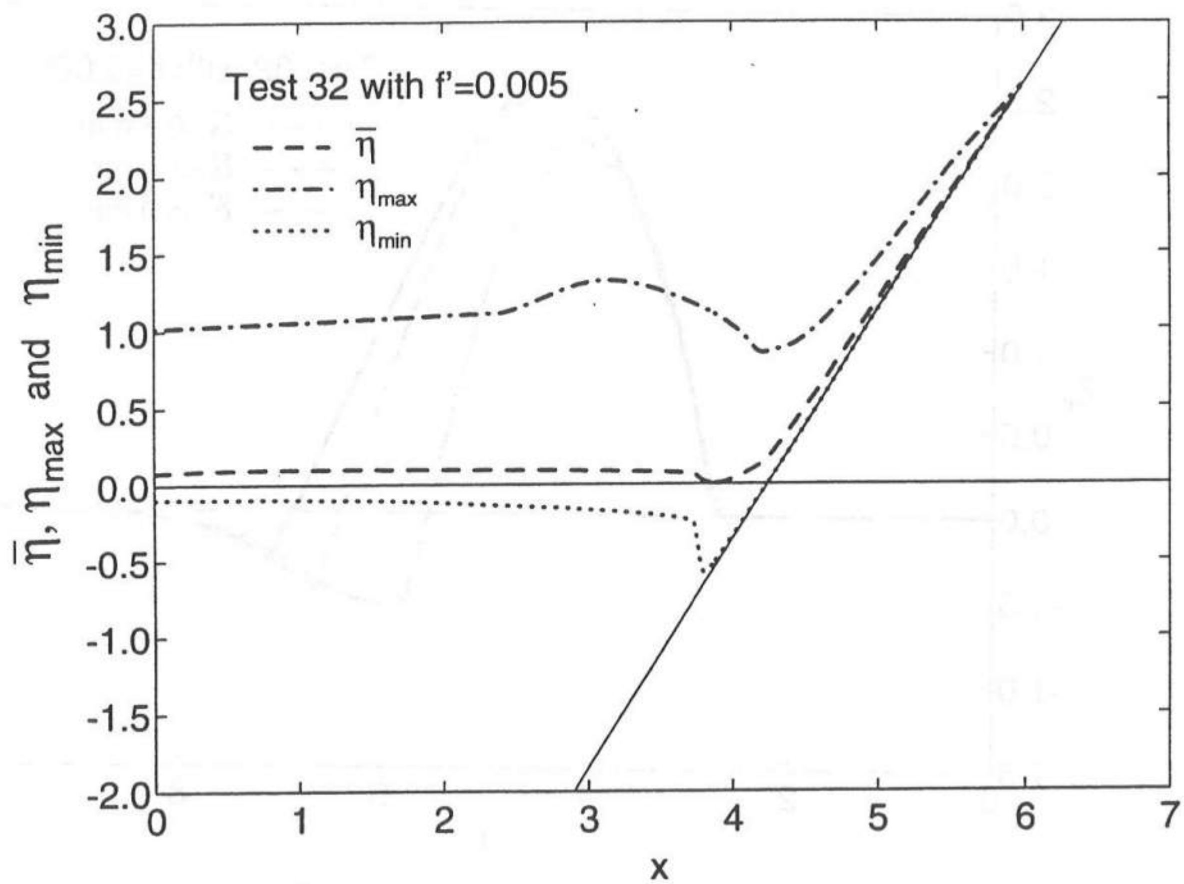

FIGURE 18. Spatial Variations of Maximum, Mean and Minimum Values of Normalized Free Surface Elevation η for Test 32 with $f' = 0.005$.

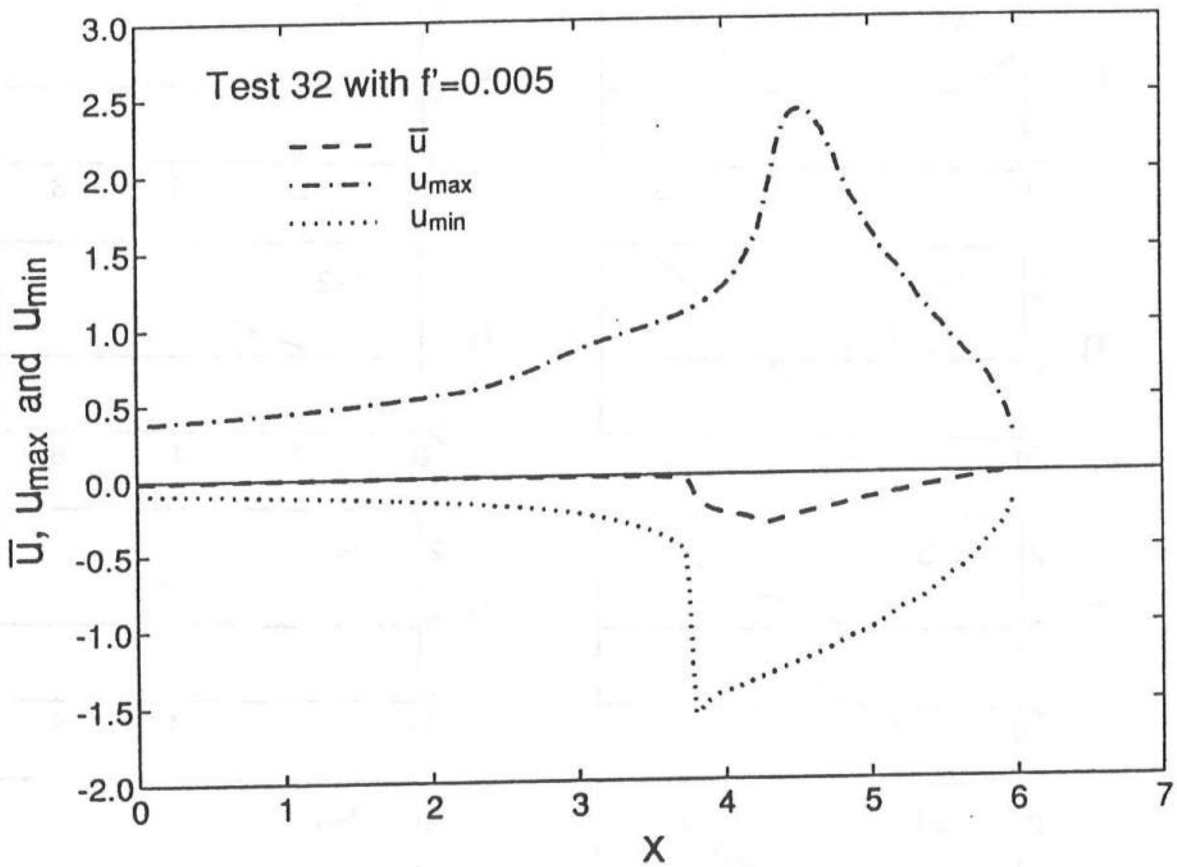

FIGURE 19. Spatial Variations of Maximum, Mean and Minimum Values of Normalized Depth-Averaged Velocity u for Test 32 with f′ = 0.005.

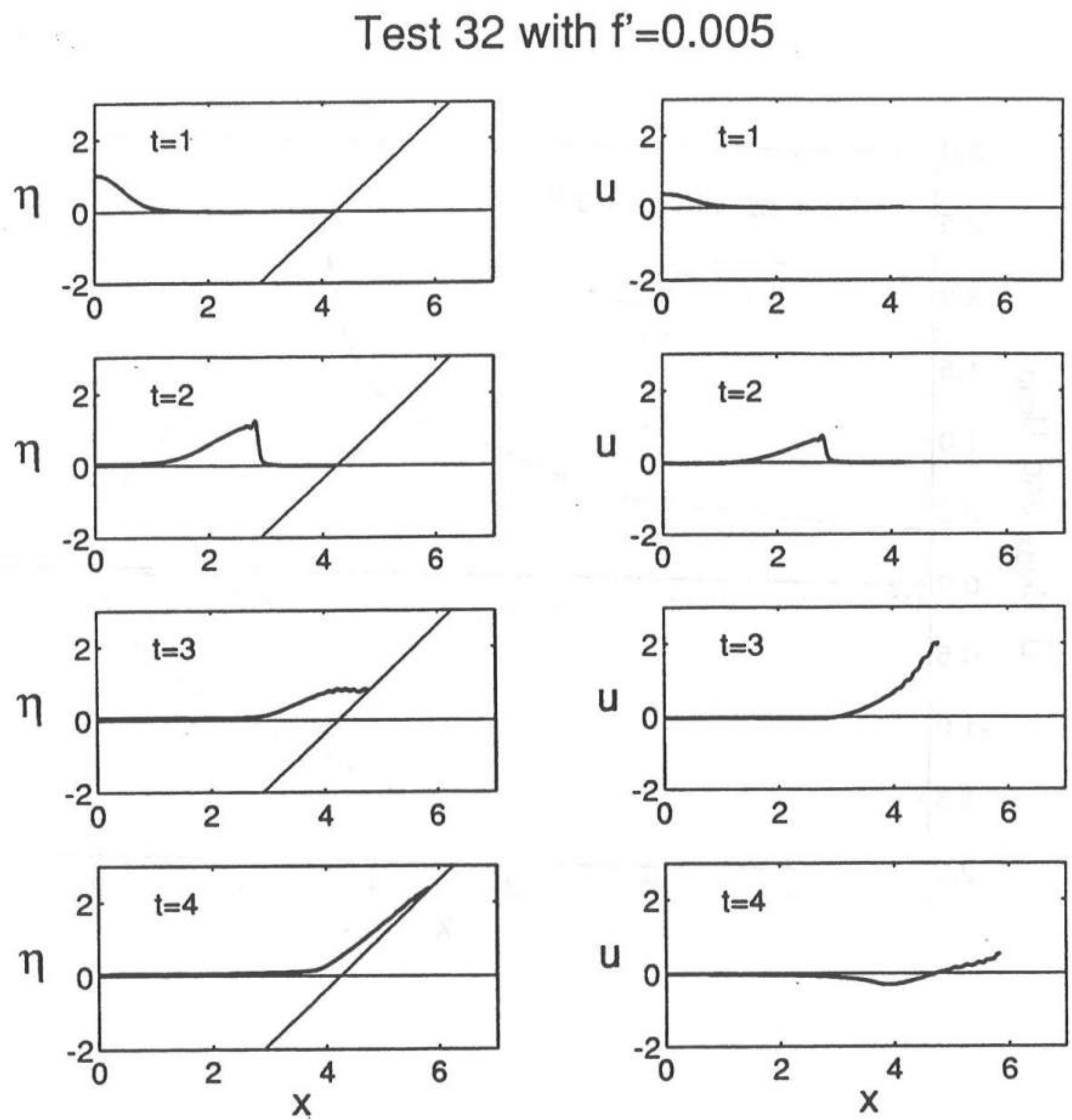

FIGURE 20. Spatial Variations of η and u at t = 1, 2, 3, 4, 5, 6, 7 and 8 for Test 32 with f' = 0.005.

Test 32 with f'=0.005

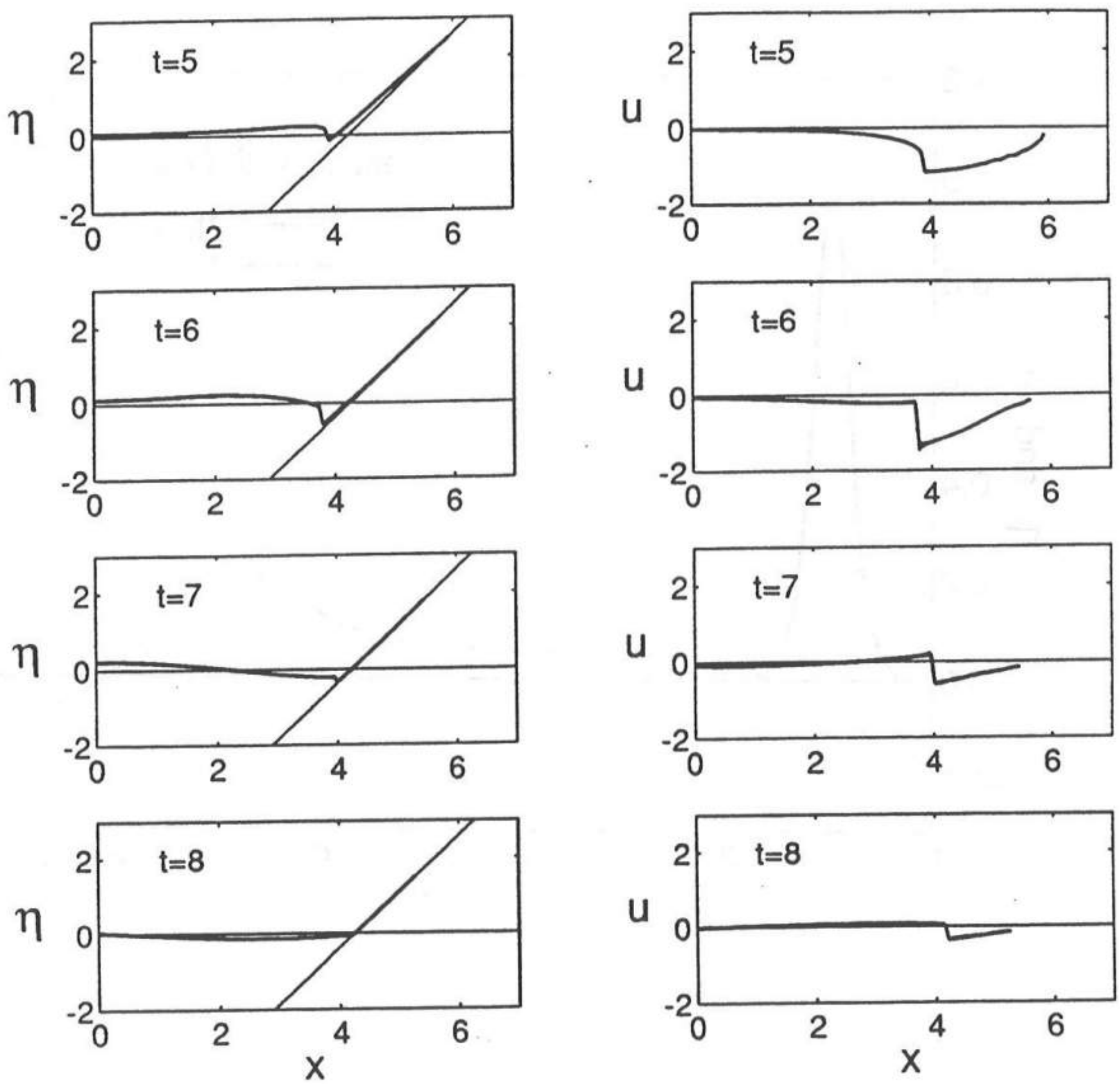

FIGURE 20. (Continued)

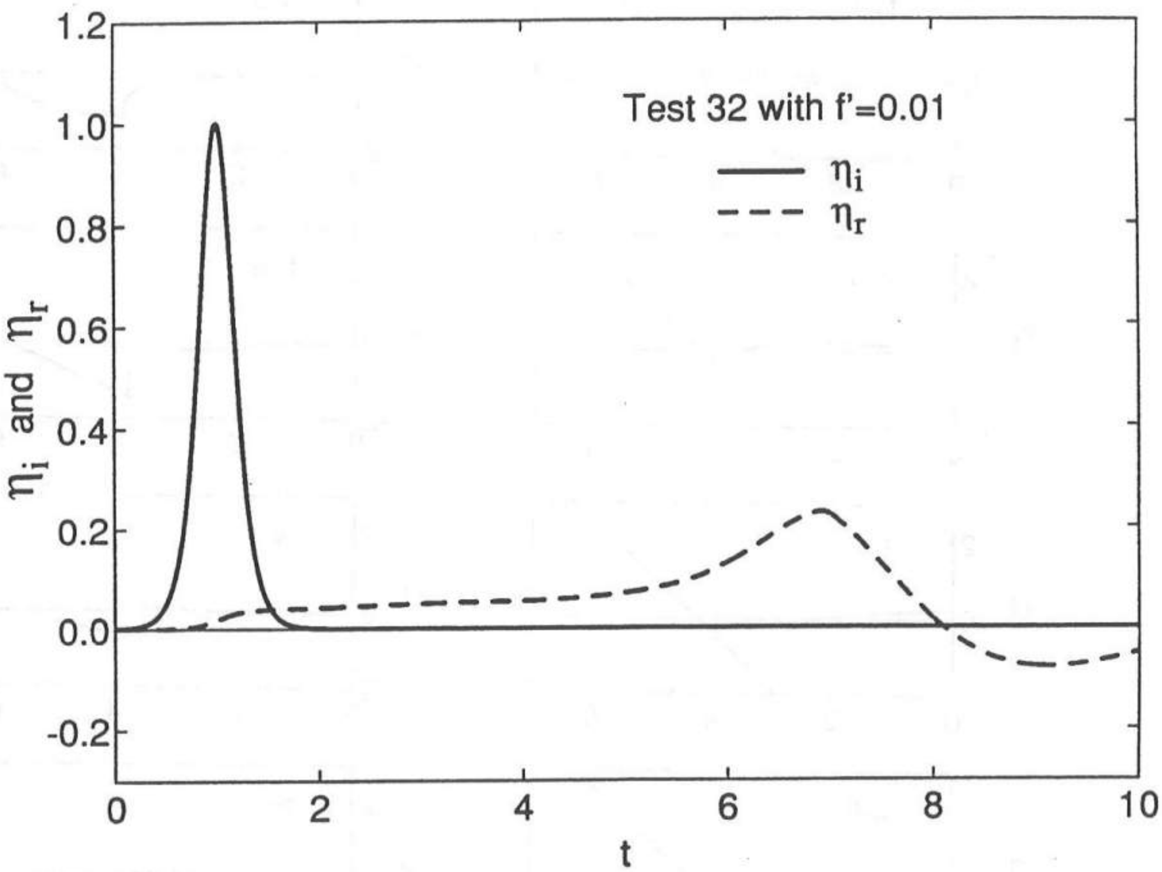

FIGURE 21. Specified Incident Solitary Wave Profile $\eta_i(t)$ and Computed Reflected Wave Profile $\eta_r(t)$ at $x = 0$ for Test 32 with $f' = 0.01$.

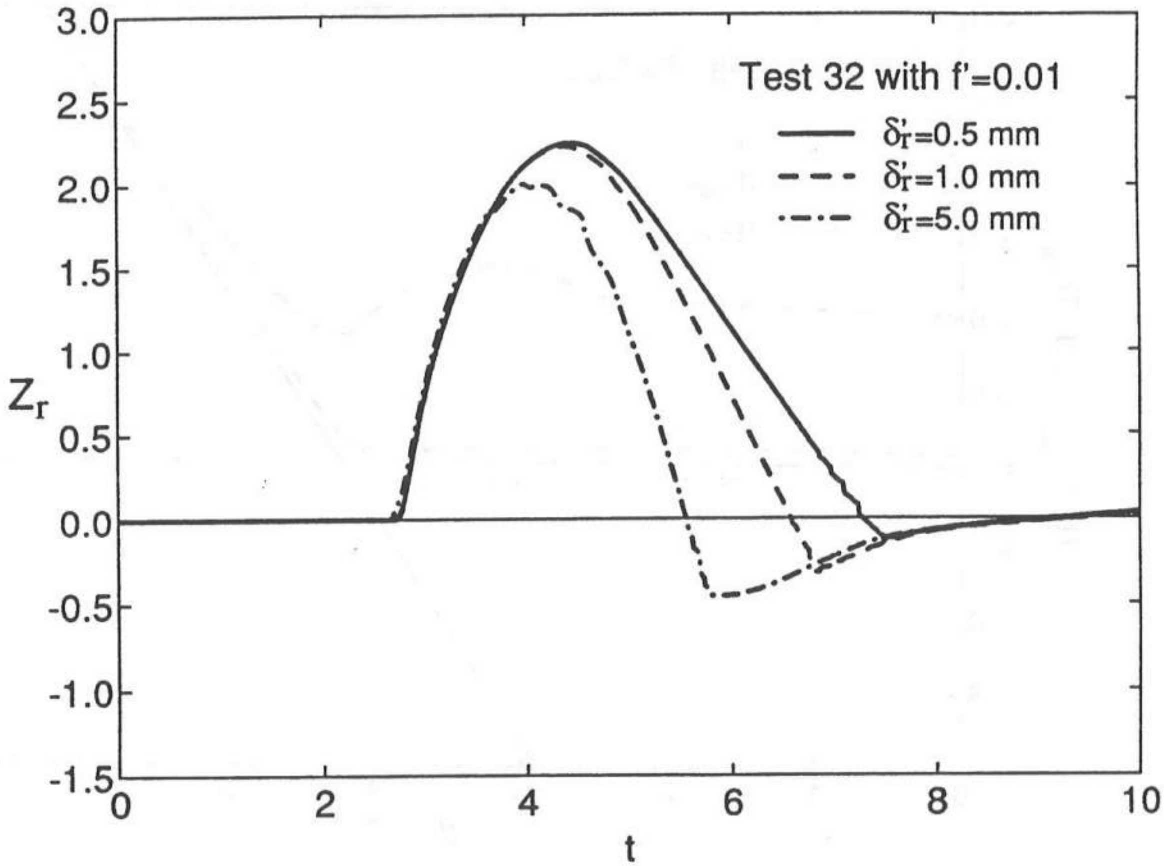

FIGURE 22. Temporal Variations of Normalized Shoreline Elevation Z_r above SWL for $\delta_r' = 0.5$, 1.0 and 5.0 mm for Test 32 with $f' = 0.01$.

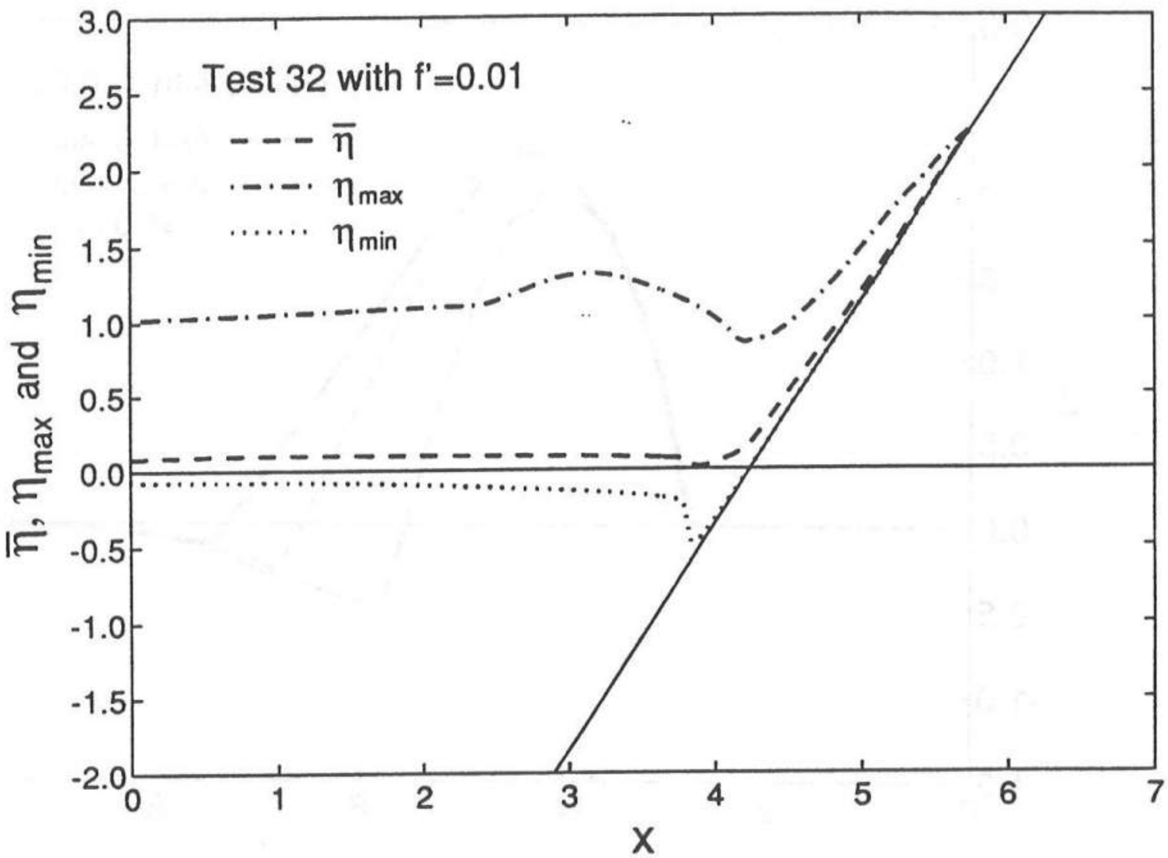

FIGURE 23. Spatial Variations of Maximum, Mean and Minimum Values of Normalized Free Surface Elevation η for Test 32 with $f' = 0.01$.

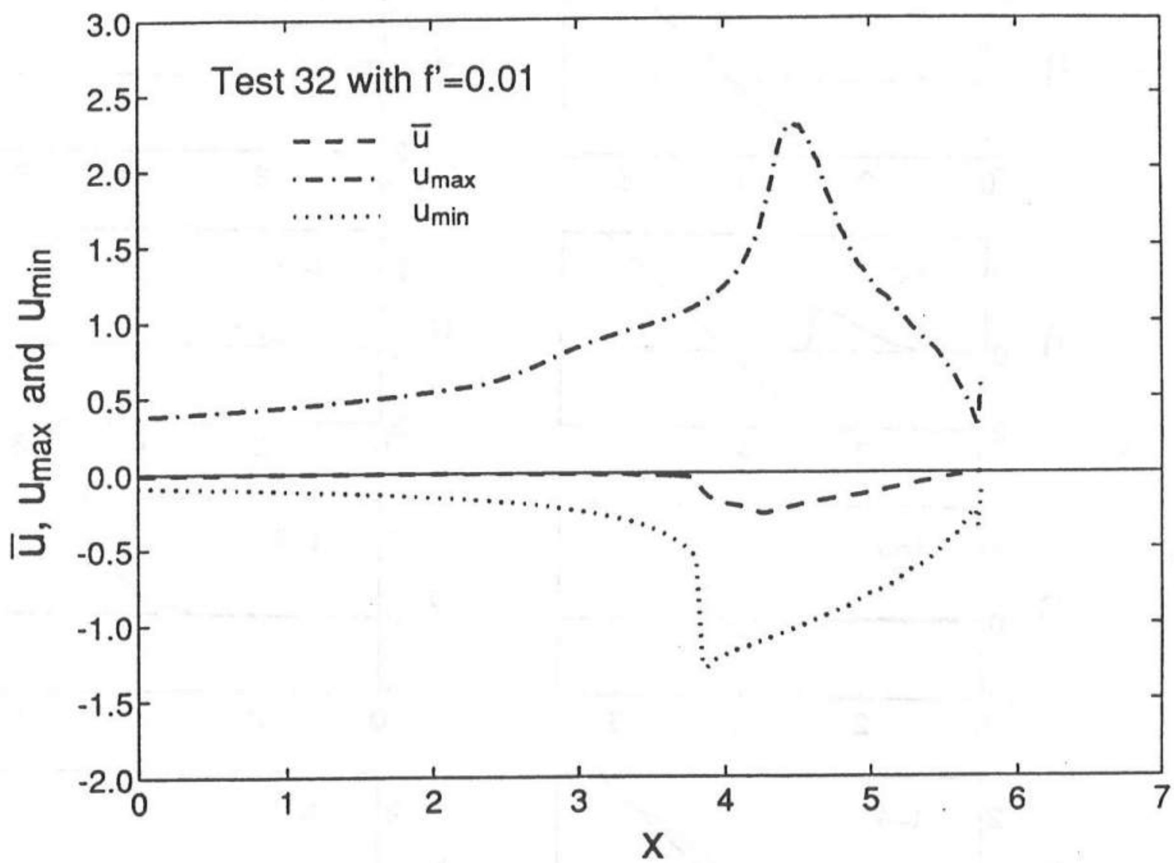

FIGURE 24. Spatial Variations of Maximum, Mean and Minimum Values of Normalized Depth-Averaged Velocity u for Test 32 with f' = 0.01.

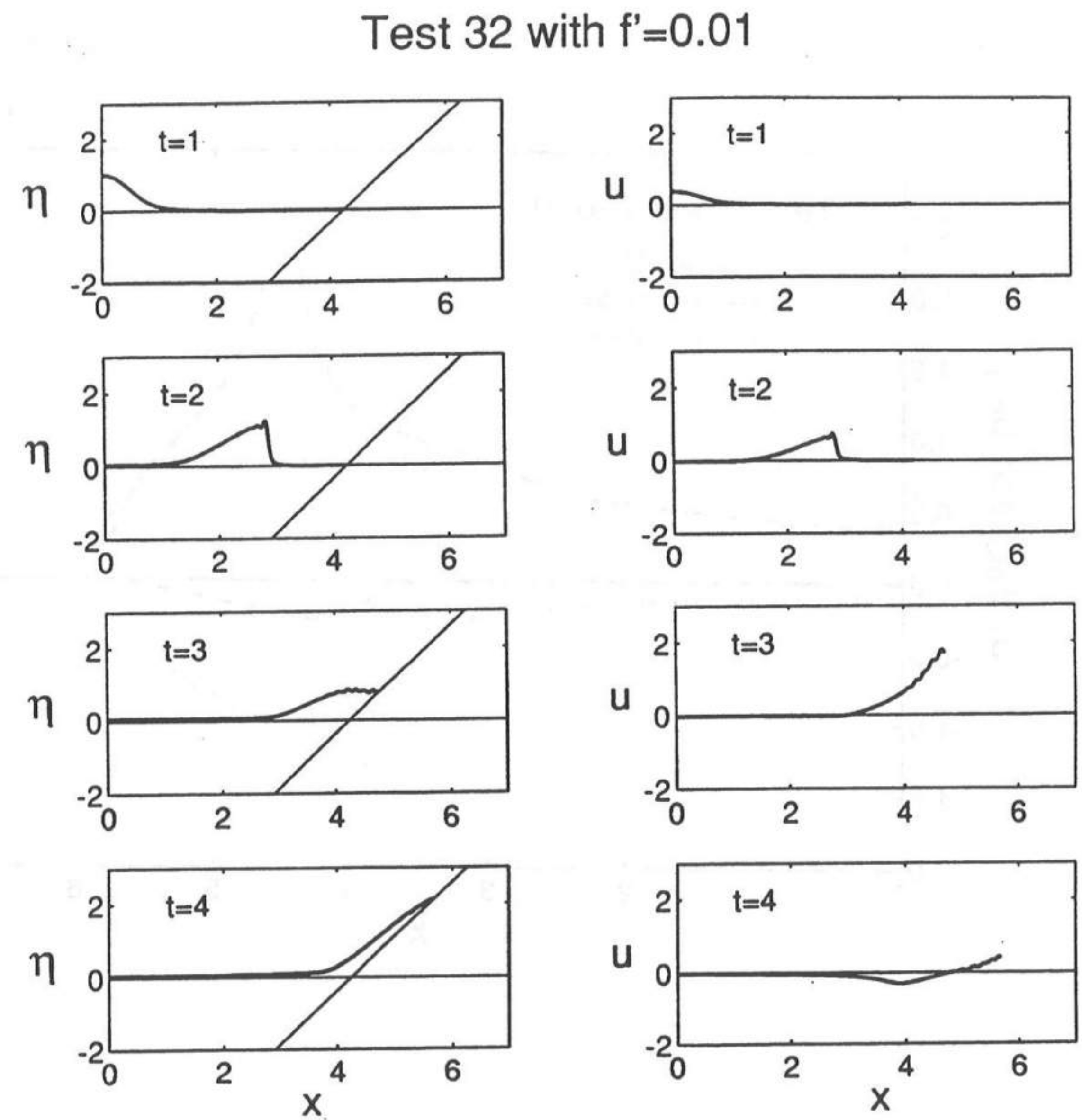

FIGURE 25. Spatial Variations of η and u at t = 1, 2, 3, 4, 5, 6, 7 and 8 for Test 32 with f' = 0.01.

Test 32 with f'=0.01

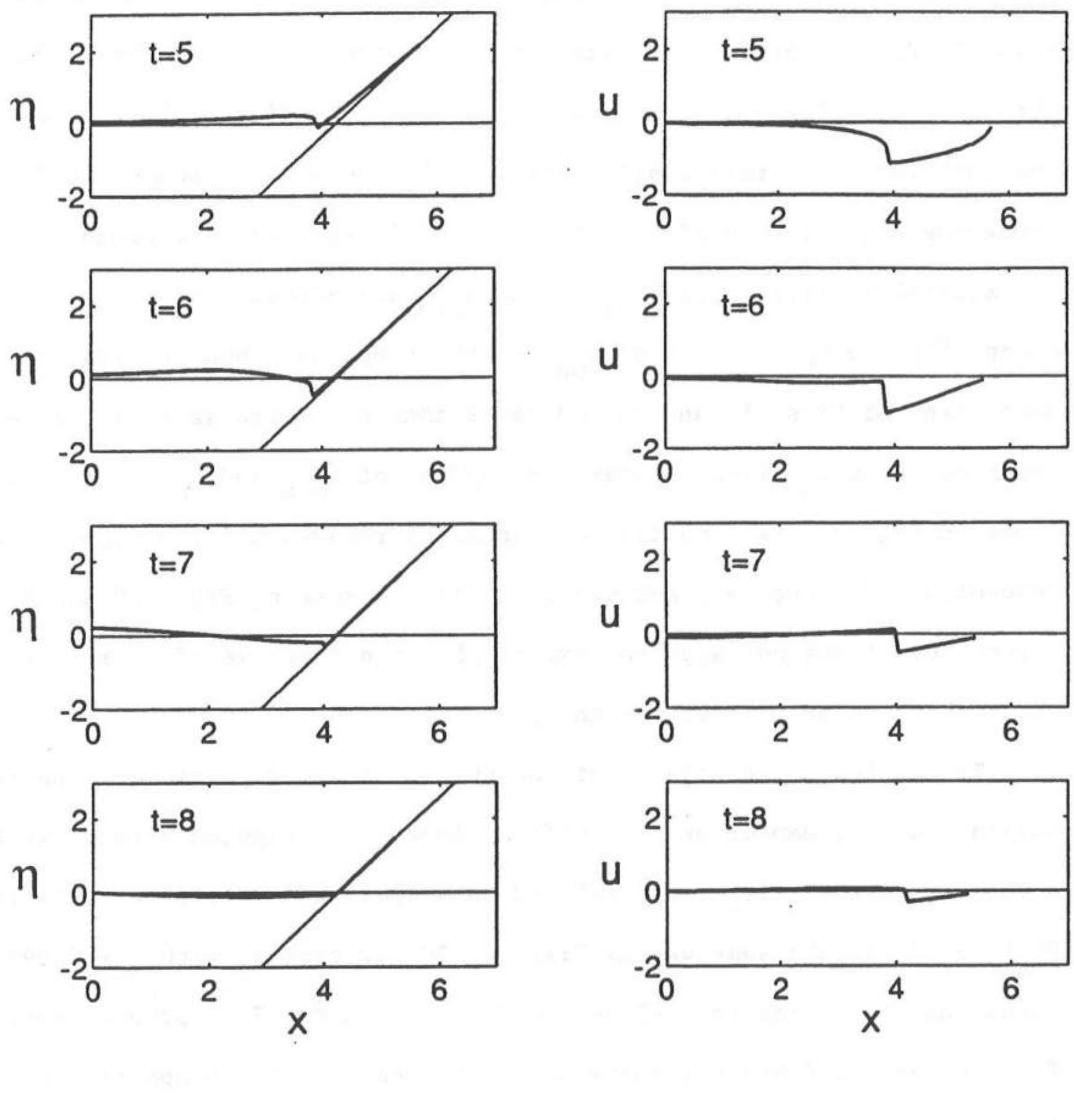

FIGURE 25. (Continued)

levels are stored in the file OSPACE together with the normalized bottom geometry.

The computation for test 32 is repeated using the bottom friction factor $f' = 0.01$ instead of $f' = 0.005$ where the rest of the input listed in Table 8 is the same. The computed results for test 32 with $f' = 0.01$ are plotted in Figs. 21-25. Comparison of Figs. 16 and 21 indicates that the reflected wave profile $\eta_r(t)$ at $x = 0$ is affected very little by $f' = 0.01$ or 0.005, while the incident wave profile $\eta_i(t)$ given by Eq. 36 is independent of f'. Comparing Figs. 17 and 22, the increase of f' reduces wave runup noticeably. The spatial variations of η_{max}, $\bar{\eta}$ and η_{min} are affected very little by f' except for the upper limit of η_{max} on the slope as shown in Figs. 18 and 23. Comparison of Figs. 19 and 24 indicates that the increase of f' reduces the magnitude of u_{min} slightly where the spikes of u_{max} and u_{min} at the landward limit in Fig. 24 are numerical and could be removed using a finer numerical resolution (Wurjanto and Kobayashi, 1991). Comparing Figs. 20 and 25, the effects of f' are not apparent except that the increase of f' reduces the downrushing water velocity slightly.

To elucidate the effects of the surf similarity parameter ξ on the solitary wave dynamics on the uniform slope, the computed results with $f' = 0.005$ for test 23 with $\xi = 0.125$ and test 40 with $\xi = 1.757$ are plotted in Figs. 26-35 in the same way as Figs. 16-20 for test 32 with $\xi = 0.591$. It is noted that the normalized slope $\theta = (2\pi)^{1/2}\xi$ in Eq. 7 is proportional to ξ. The increase of ξ hence results in the increase of θ. Comparison of Figs. 16, 26 and 31 indicates that the increase of ξ leads to the increase in the magnitude of wave reflection and the reduction in the normalized arrival time of the reflected wave at $x = 0$. The increase of the solitary wave reflection with the increase of ξ is qualitatively consistent with available data on the reflection of breaking monochromatic waves (Battjes, 1974). Comparing Figs. 17, 27 and 32, the increase of ξ results in the increase in the normalized wave runup and the reduction in the normalized duration of wave runup and rundown. It is noted that the uprushing water on the slope for test 23 has reached the maximum elevation before $t = 10$ in Fig. 27 since the water flows downslope at $t = 10$ as shown in Fig. 30. Figs. 18, 28 and 33 clearly show the increase of the normalized slope with the increase of ξ. Fig. 28 for test 23 with $\xi = 0.125$ shows the wide zone of wave decay, whereas Fig. 33 for test 40 with $\xi = 1.757$ exhibits gradual shoaling and uprushing above the still water shoreline. Comparison of Figs. 19, 29 and 34 indicates that the increase of ξ results in the increase in the downrushing water velocity. It should be noted that the downrushing water velocity for test 23 may become larger after $t = 10$ since the water still flows downslope at $t = 10$ as shown in Fig. 30. Comparing Figs. 20, 30 and 35, the incident solitary wave breaks more clearly as ξ is reduced, although the present numerical model does not predict the details of wave breaking.

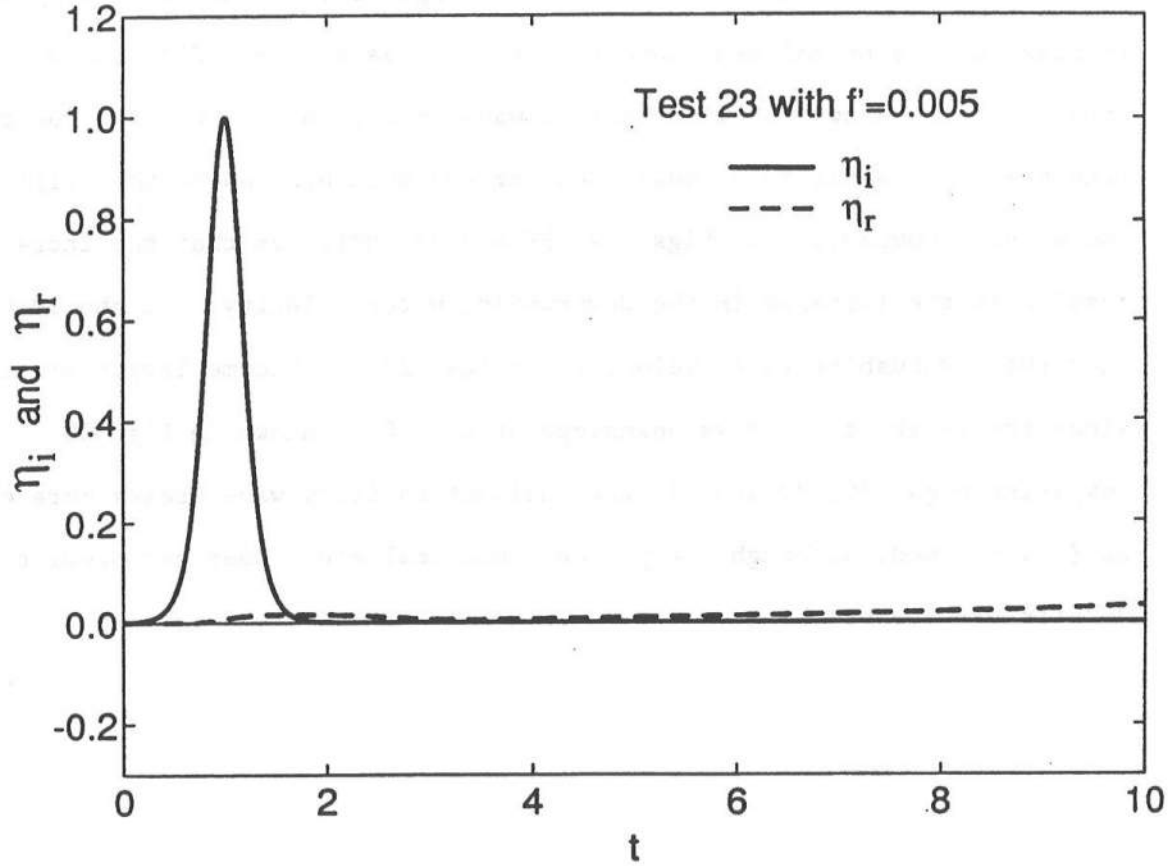

FIGURE 26. Specified Incident Solitary Wave Profile $\eta_i(t)$ and Computed Reflected Wave Profile $\eta_r(t)$ at x = 0 for Test 23 with f' = 0.005.

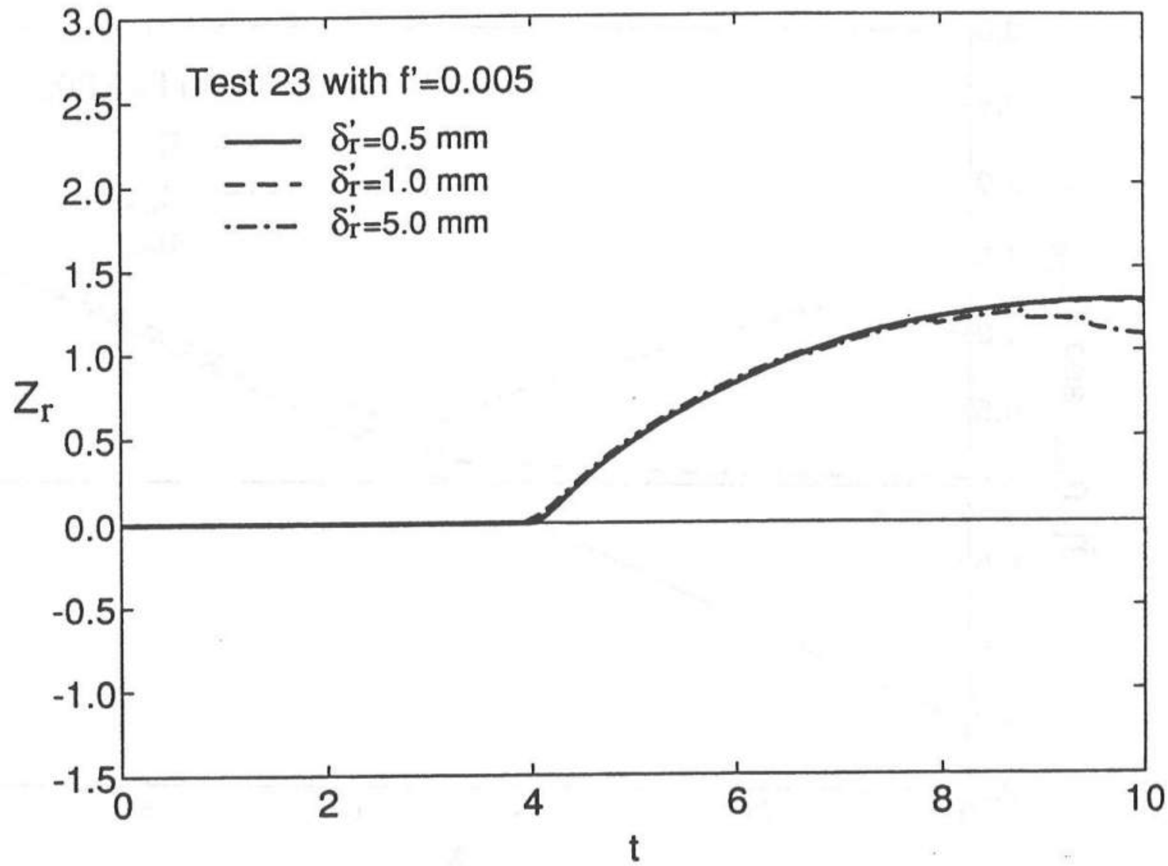

FIGURE 27. Temporal Variations of Normalized Shoreline Elevation Z_r above SWL for $\delta_r' = 0.5$, 1.0 and 5.0 mm for Test 23 with $f' = 0.005$.

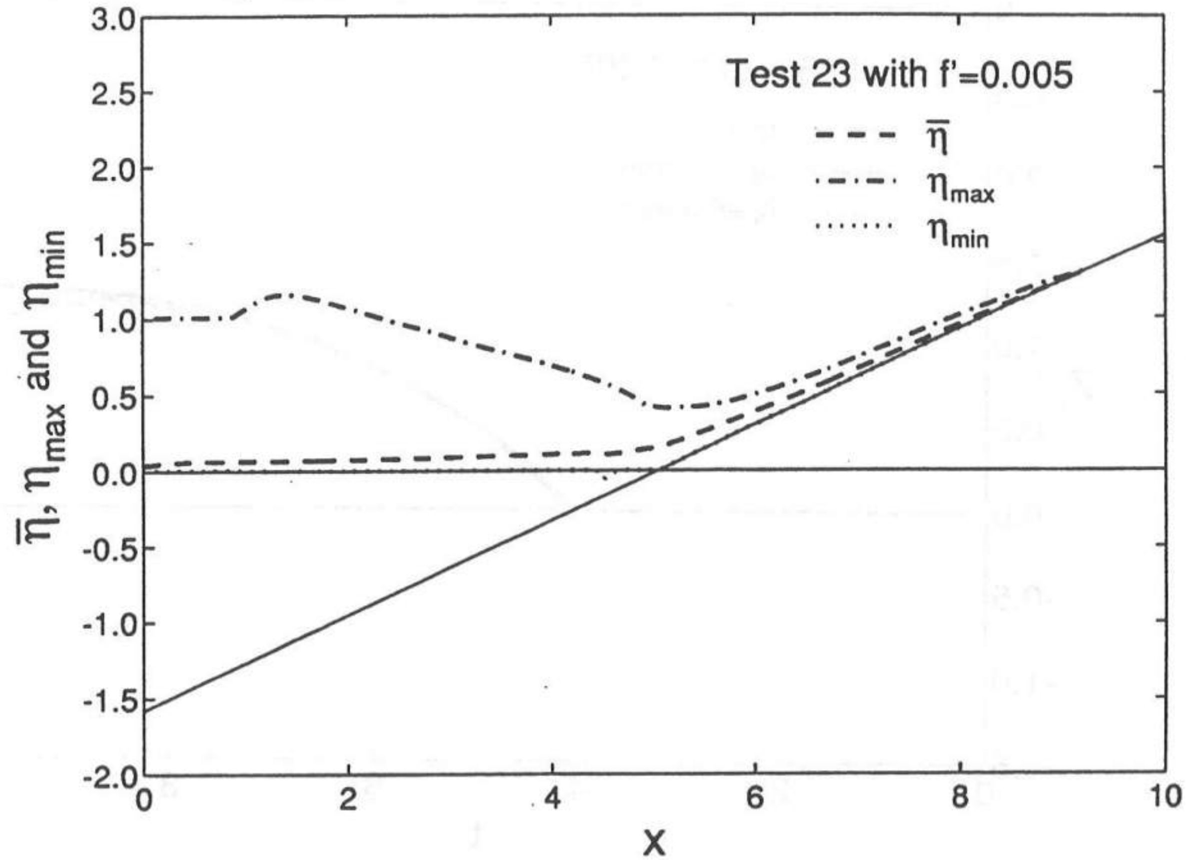

FIGURE 28. Spatial Variations of Maximum, Mean and Minimum Values of Normalized Free Surface Elevation η for Test 23 with f' = 0.005.

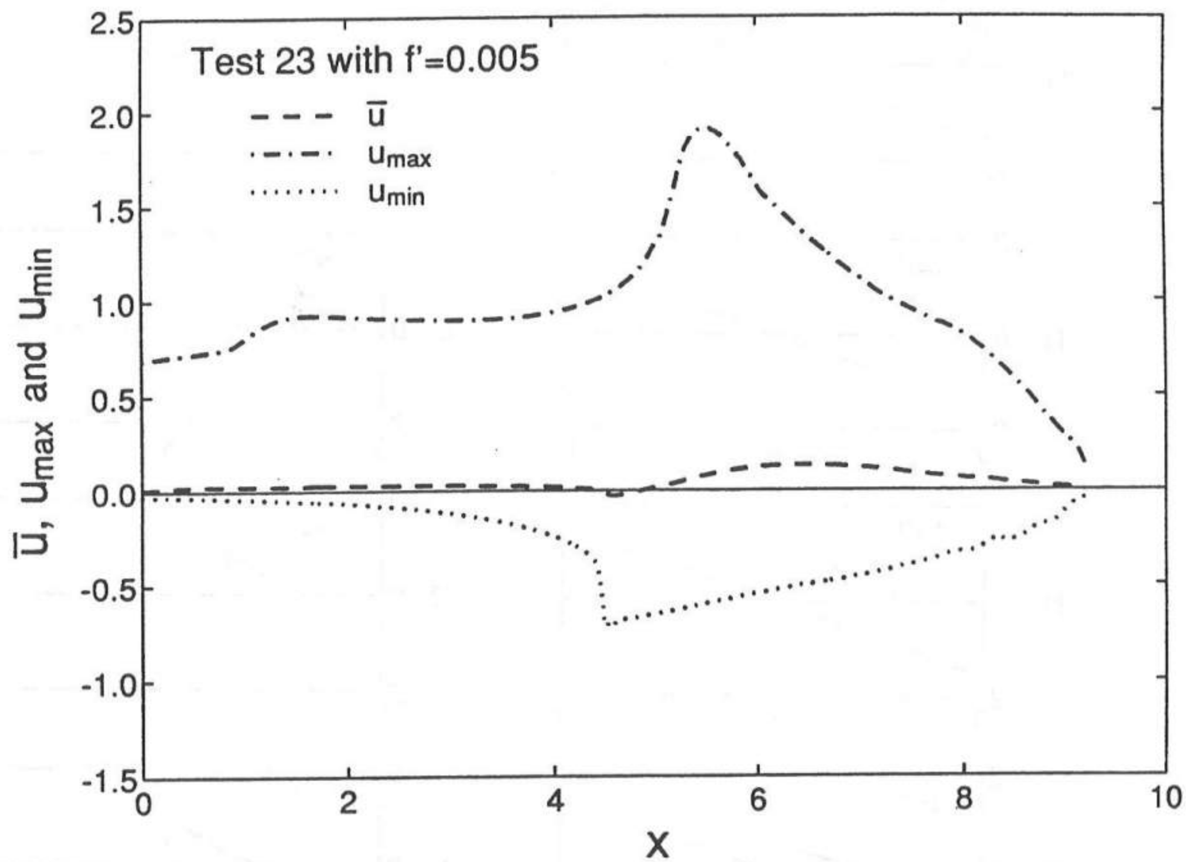

FIGURE 29. Spatial Variations of Maximum, Mean and Minimum Values of Normalized Depth-Averaged Velocity u for Test 23 with f' = 0.005.

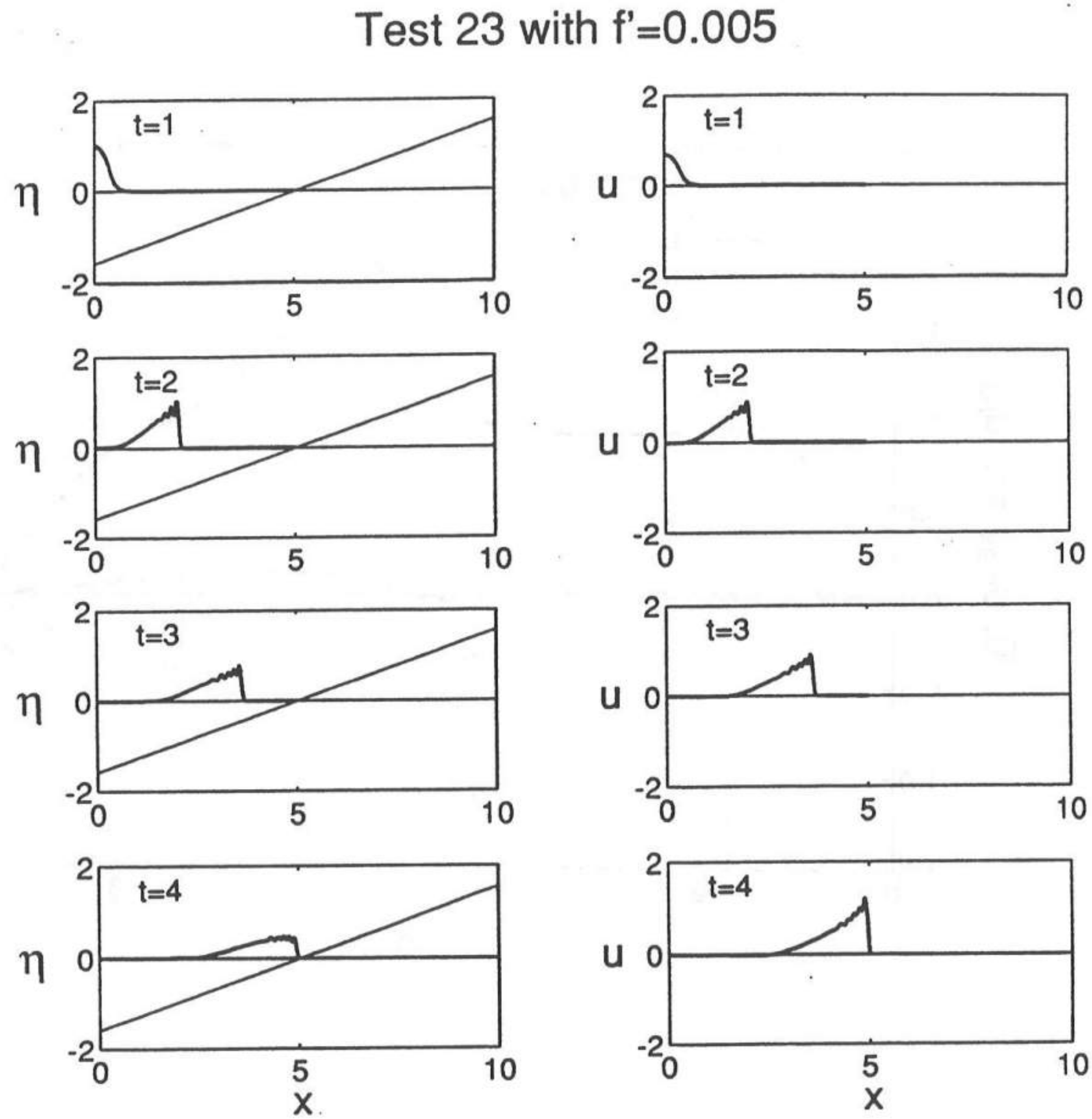

FIGURE 30. Spatial Variations of η and u at t = 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 for Test 23 with f' = 0.005.

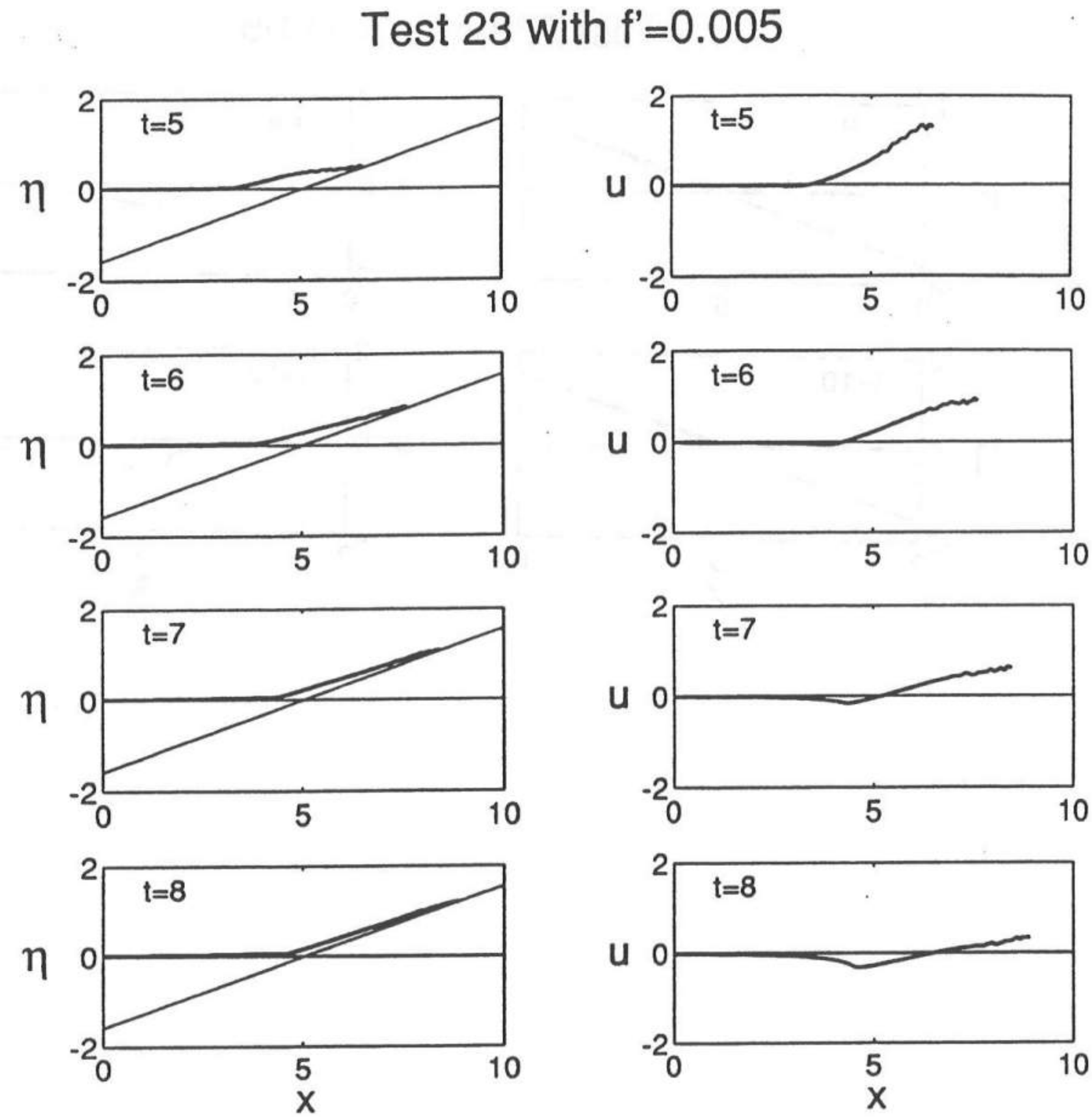

FIGURE 30. (Continued)

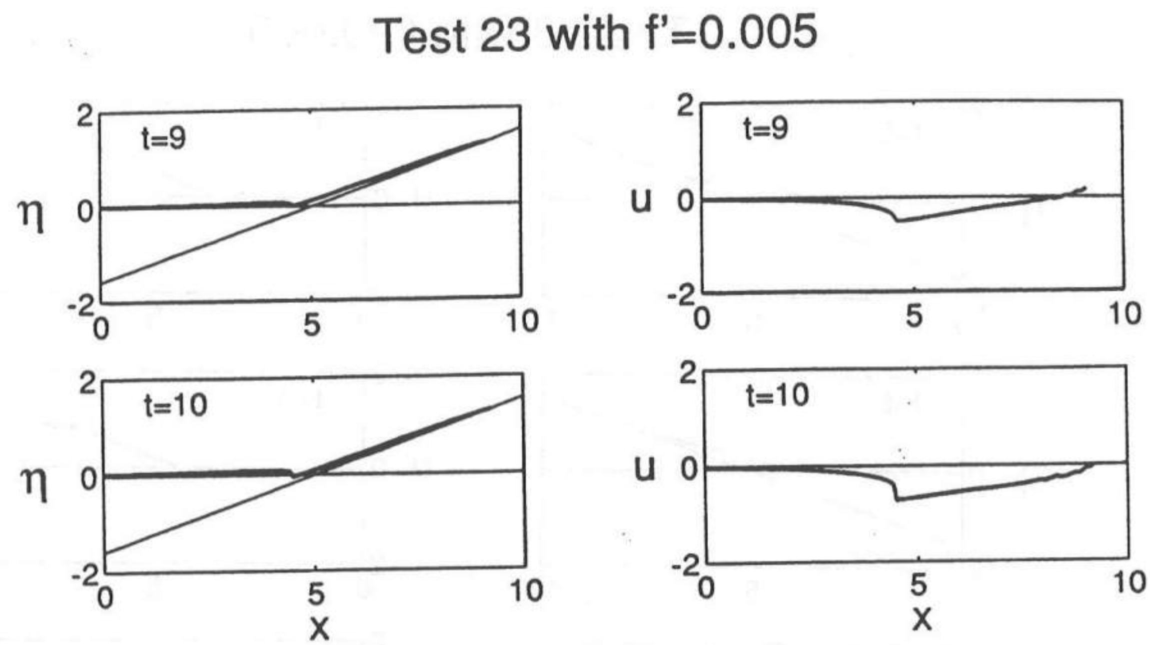

FIGURE 30. (Continued)

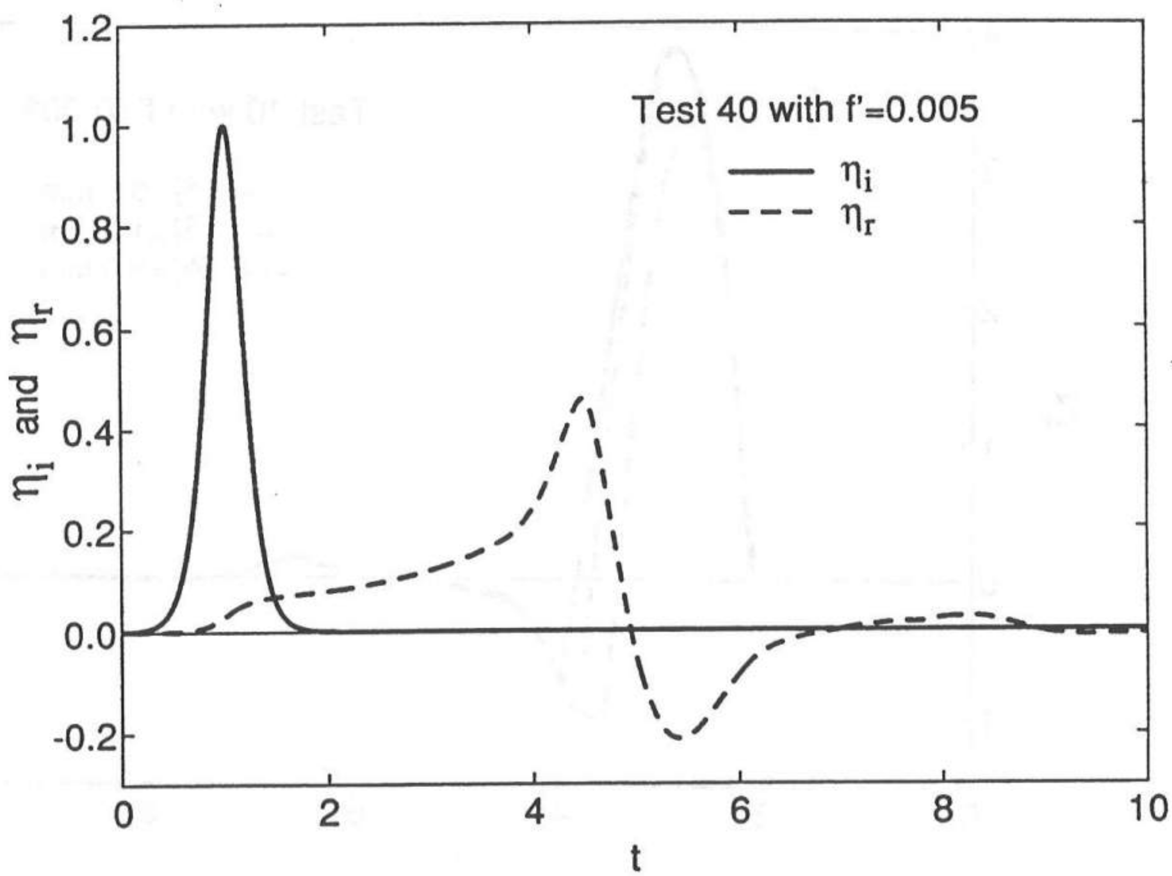

FIGURE 31. Specified Incident Solitary Wave Profile $\eta_i(t)$ and Computed Reflected Wave Profile $\eta_r(t)$ at x=0 for Test 40 with f' = 0.005.

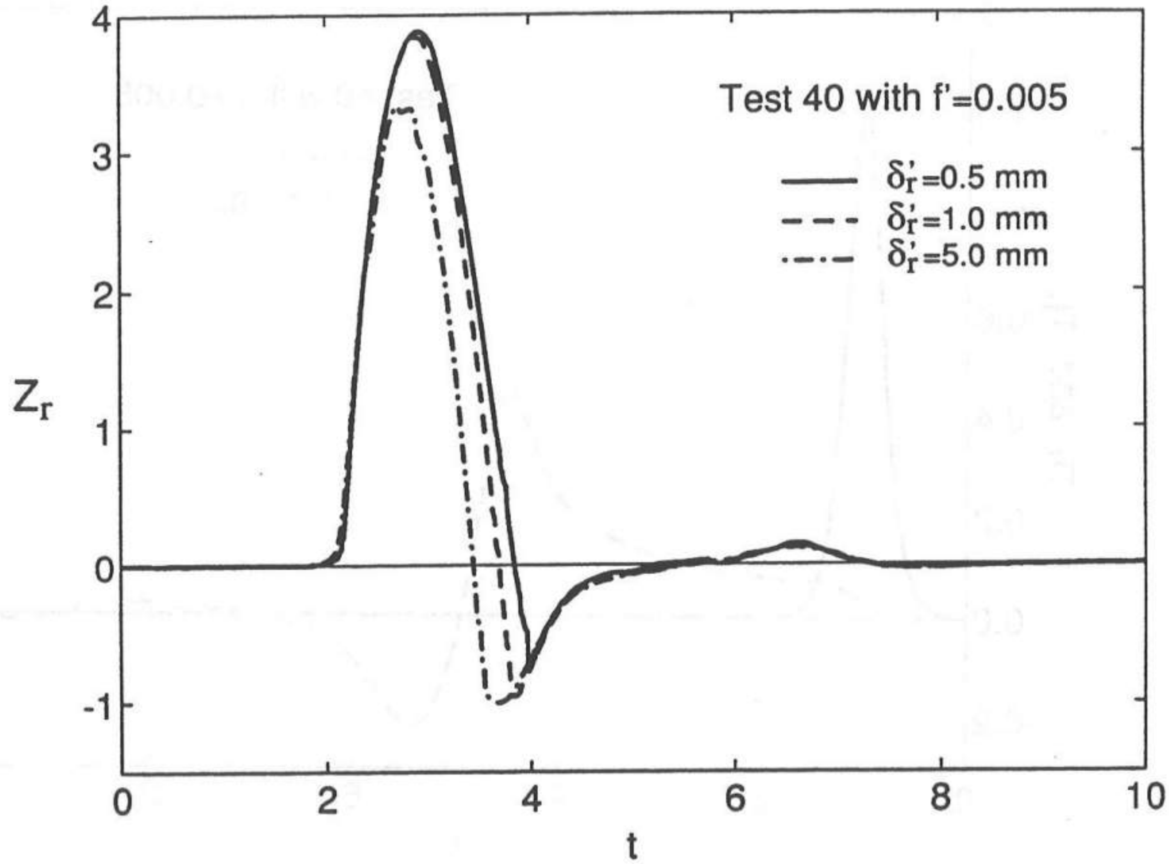

FIGURE 32. Temporal Variations of Normalized Shoreline Elevation Z_r above SWL for $\delta_r' = 0.5$, 1.0 and 5.0 mm for Test 40 with $f' = 0.005$.

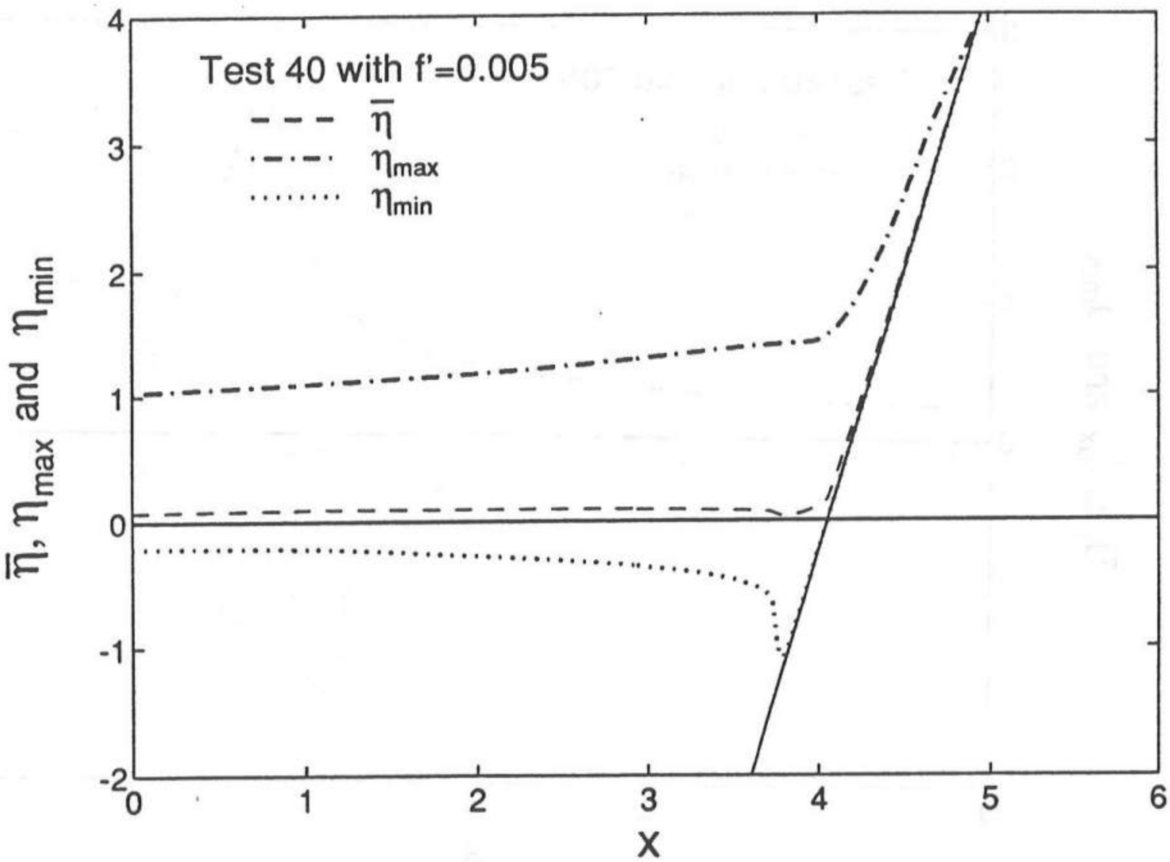

FIGURE 33. Spatial Variations of Maximum, Mean and Minimum Values of Normalized Free Surface Elevation η for Test 40 with f' = 0.005.

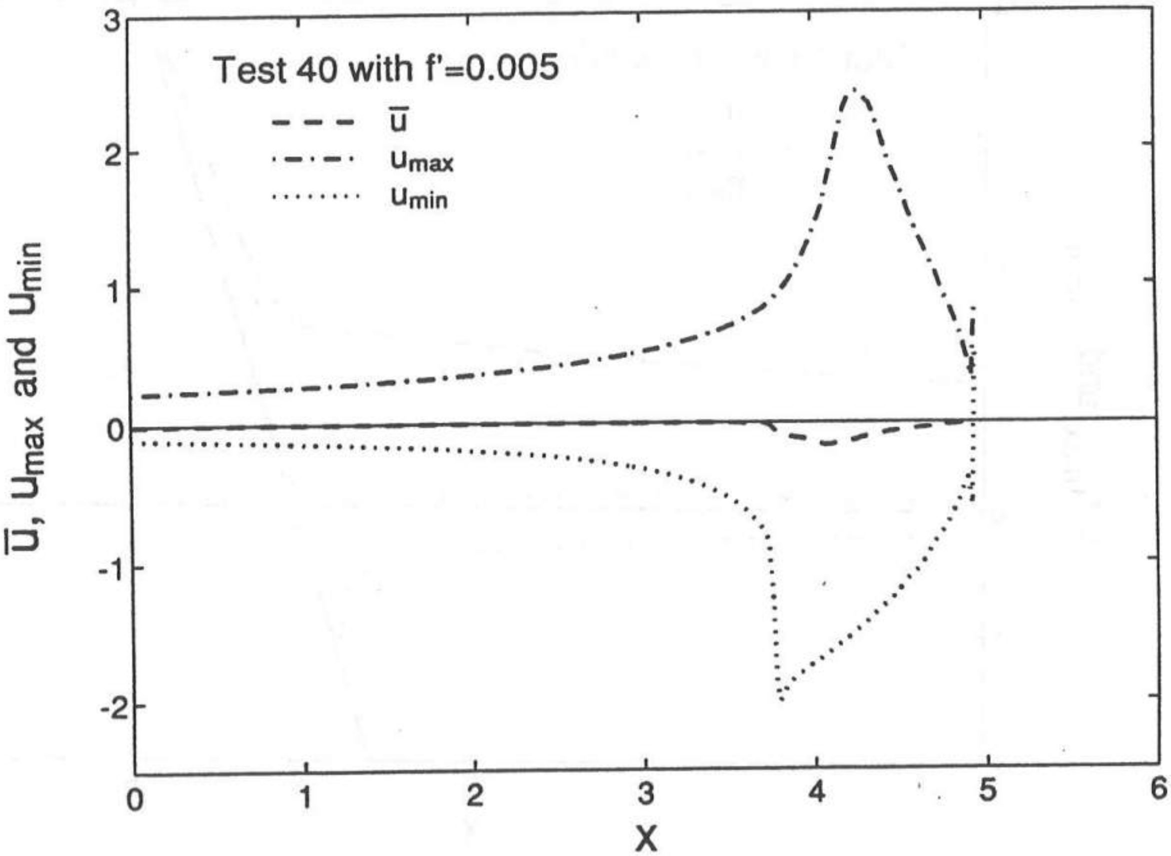

FIGURE 34. Spatial Variations of Maximum, Mean and Minimum Values of Normalized Depth-Averaged Velocity u for Test 40 with f' = 0.005.

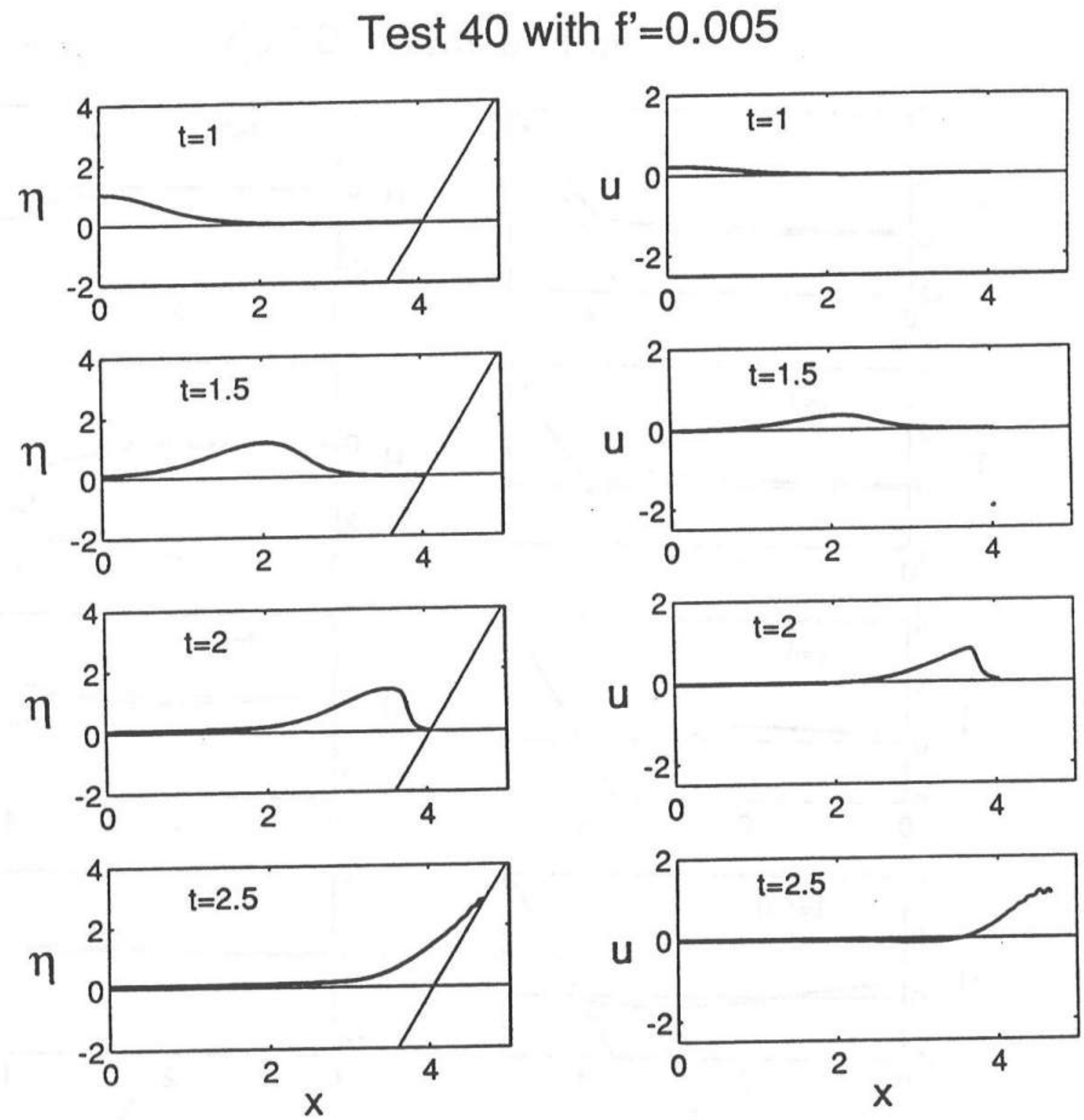

FIGURE 35. Spatial Variations of η and u at t = 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5 and 6.0 for Test 40 with f' = 0.005.

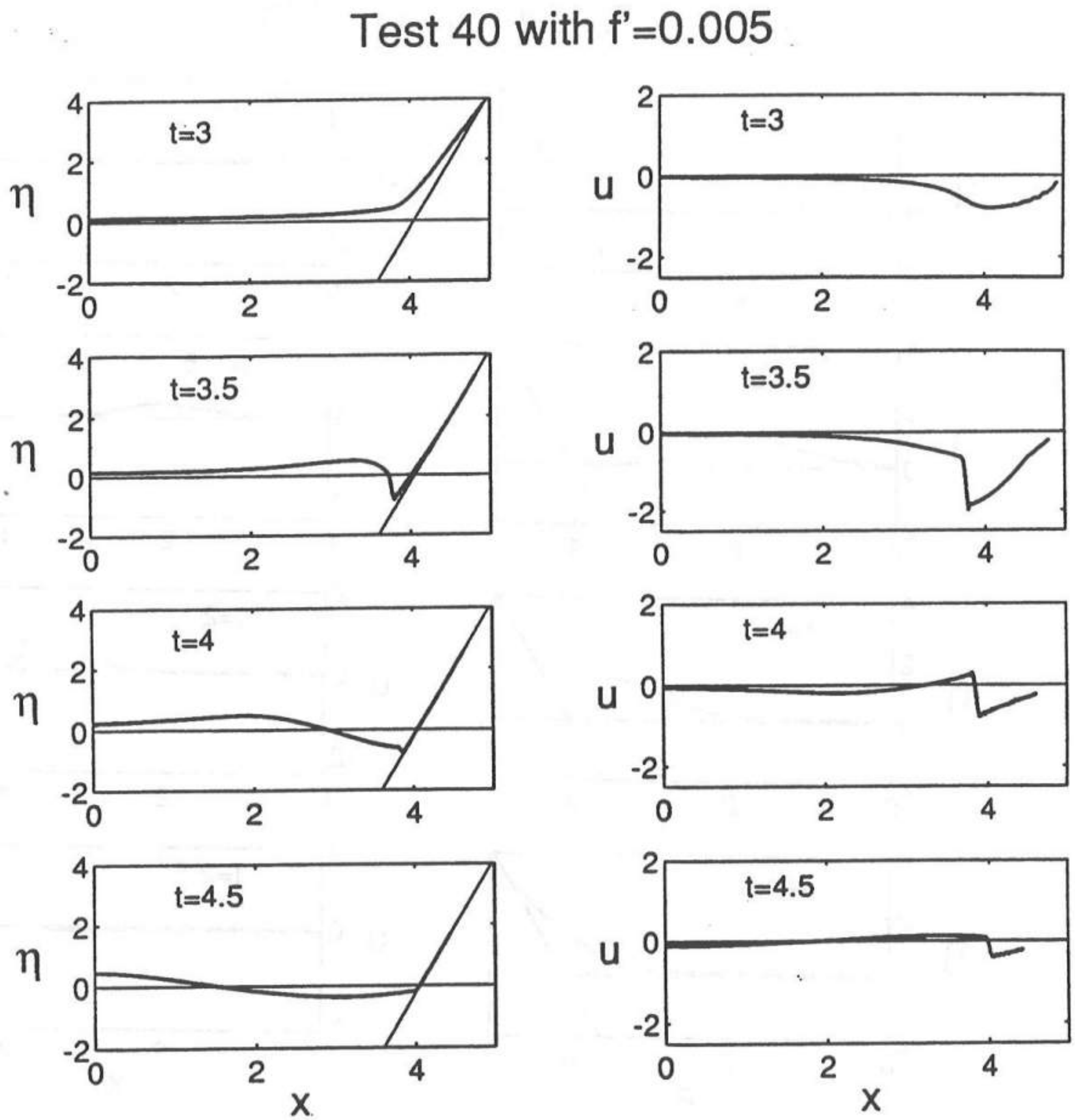

FIGURE 35. (Continued)

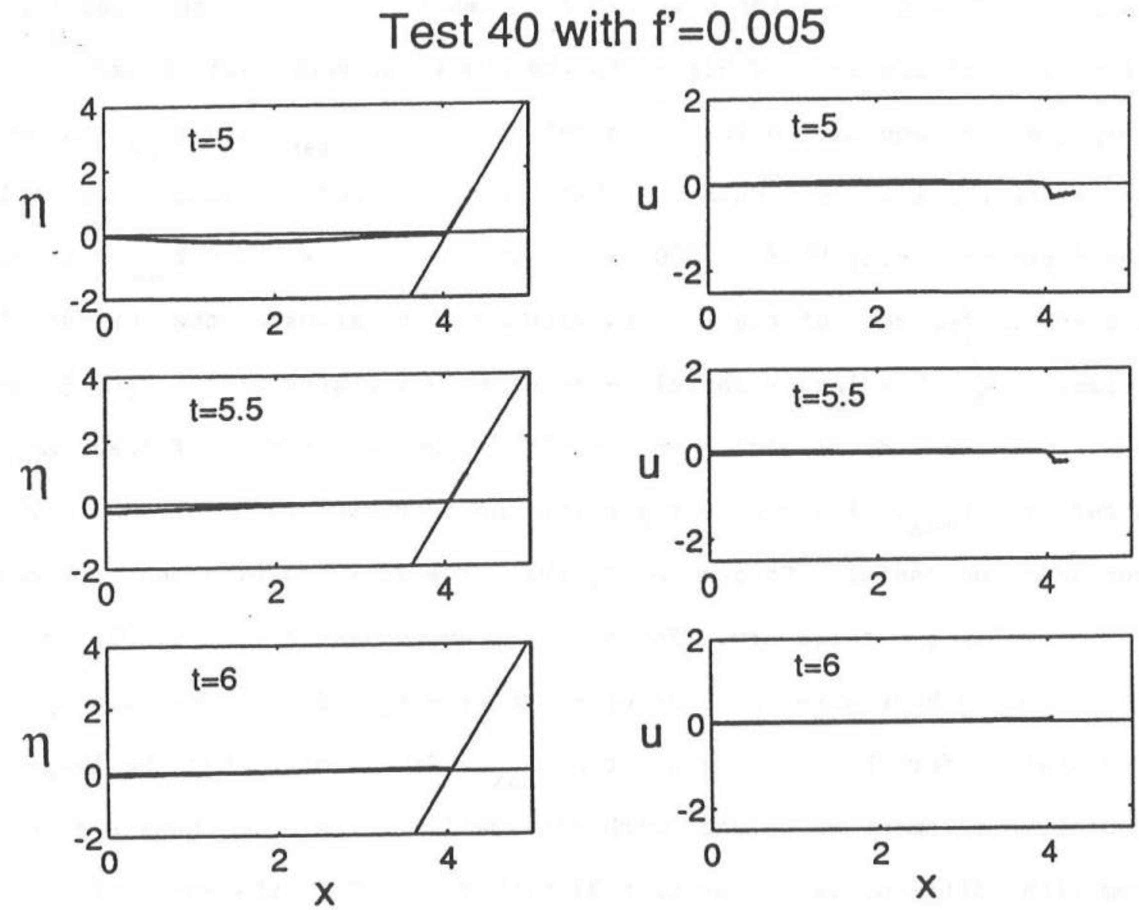

FIGURE 35. (Continued)

4.4 Comparison between Measured and Computed Runup

The measured and computed values of the normalized runup R for the selected tests are listed in Table 11. The bottom friction factor f' is taken as f' = 0.01 and 0.005 for each of the 9 tests. The try number for each test with given f' listed in Table 11 is the number of computations made to complete the computation for the duration $0 \leq t \leq t_{max}$ with t_{max} = NTOP/NONE where the primary input data file has been explained in relation to Table 8. The first try using NONE = 1000, $\delta = 0.001$, $\epsilon_1 = \epsilon_2 = 1$ and $t_{max} = 10$ has been successful for most of the 9 tests with smaller values of the surf similarity parameter ξ. The larger shoreline movement associated with larger ξ tends to cause numerical difficulties and result in the termination of the computation before $t = t_{max}$. The second try using the increased value of NONE = 2000 has not been successful. Consequently, the value of δ has also been increased to $\delta = 0.002$ in the third try. For tests 37 and 39 with f' = 0.005, the values of ϵ_1 and ϵ_2 have also been increased to $\epsilon_1 = \epsilon_2 = 2$ in order to complete the computation for the duration $0 \leq t \leq t_{max}$. It is noted that the larger shoreline movement associated with the smaller value of f' causes more numerical difficulties. For test 37 with f' = 0.005, the value of t_{max} has been reduced to $t_{max} = 5$ since the computed shoreline has reached the maximum elevation before t = 5. Table 11 lists the value of the numerical stability indicator ALPHAS given by Eq. 88 for each test with given f'. For the successful computations, ALPHAS = 2.6 - 6.9, which may be used as a guideline for the selection of a reasonable value of NONE.

The measured and computed values for the normalized wave runup R with $\delta_r' = 1$ mm are plotted in Fig. 36 for the 9 tests with f' = 0.01 and 0.005. The numerical model with f' = 0.01 underpredicts R slightly, whereas the use of f' = 0.005 results in slight overprediction of R. The computed values of R with

$f' = 0.01$ and 0.005 for the 9 tests are also plotted in Fig. 37 together with the measured values of R for the 43 tests. The empirical formulas given by Eqs. 92 and 94 for breaking monochromatic and solitary waves, respectively, are also shown in Fig. 37. The computed values of R using $f' = 0.005$ and 0.01 tend to give the upper and lower bounds of the 43 data points, respectively.

The computed values of R with $f' = 0.01$ and 0.005 are interpolated or extrapolated linearly to estimate the value of f' corresponding to the measured value of R for each of the 9 tests. The fitted values of f' for each test is listed in Table 11 and plotted as a function of ξ in Fig. 38. A linear regression analysis yields the following relationship

$$f' = 0.0059 + 0.0006\ \xi \tag{96}$$

which indicates that f' is almost independent of ξ. As a result, the constant value of $f' = 0.006$ or 0.007 is expected to yield very good agreement between the measured and computed runup for the 43 tests. In conclusion, the present numerical model yields fairly good agreement with the data on breaking solitary wave runup and the agreement could be improved by calibrating the friction factor f' more meticulously.

TABLE 11. Comparison between Measured and Computed Runup for 9 Tests with f' = 0.005 and 0.01.

Test No	ξ	Try No		*ALPHAS*		Computed R		Measured R	Fitted f'
		$f' = 0.01$	$f' = 0.005$	$f' = 0.01$	$f' = 0.005$	$f' = 0.01$	$f' = 0.005$		
7	0.181	1	1	6.0	6.0	1.28	1.48	1.43	0.0064
15	1.076	3	3	5.9	5.9	2.86	3.21	3.00	0.0079
18	0.298	1	1	4.9	4.9	1.62	1.87	1.85	0.0053
21	0.328	1	1	4.7	4.7	1.70	1.95	1.78	0.0082
23	0.125	1	1	6.9	6.9	1.12	1.29	1.33	0.0039
32	0.591	1	1	3.7	3.7	2.22	2.55	2.42	0.0070
37	1.300	3	5	5.4	3.6	3.13	3.46	3.44	0.0053
39	1.507	1	4	2.6	3.4	3.30	3.63	3.51	0.0068
40	1.757	3	3	4.8	4.8	3.47	3.84	3.70	0.0069

Try No	NONE	δ	$\varepsilon_1 = \varepsilon_2$	NTOP/ NONE
1	1000	0.001	1	10
2	2000	0.001	1	10
3	2000	0.002	1	10
4	2000	0.002	2	10
5	2000	0.002	2	5

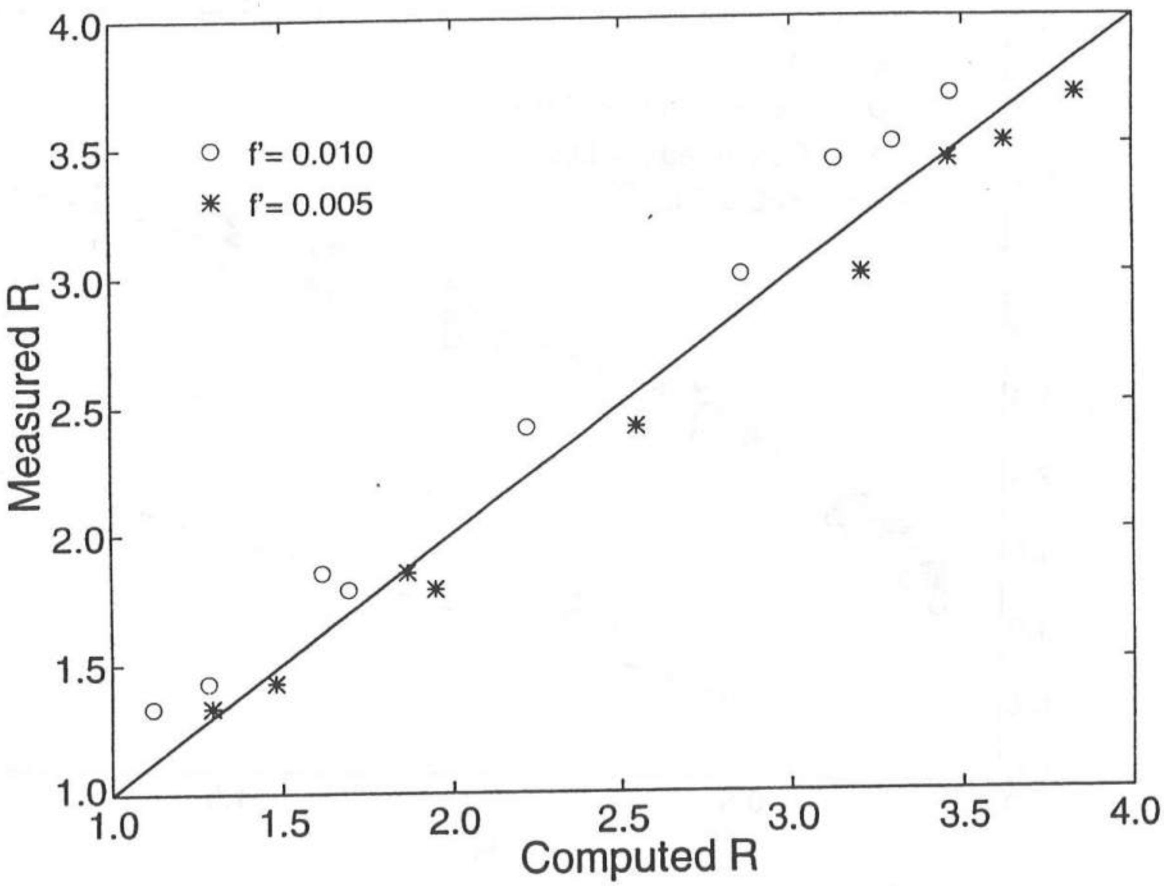

FIGURE 36. Measured and Computed Values of Normalized Runup R for 9 Tests with f' = 0.005 and 0.01.

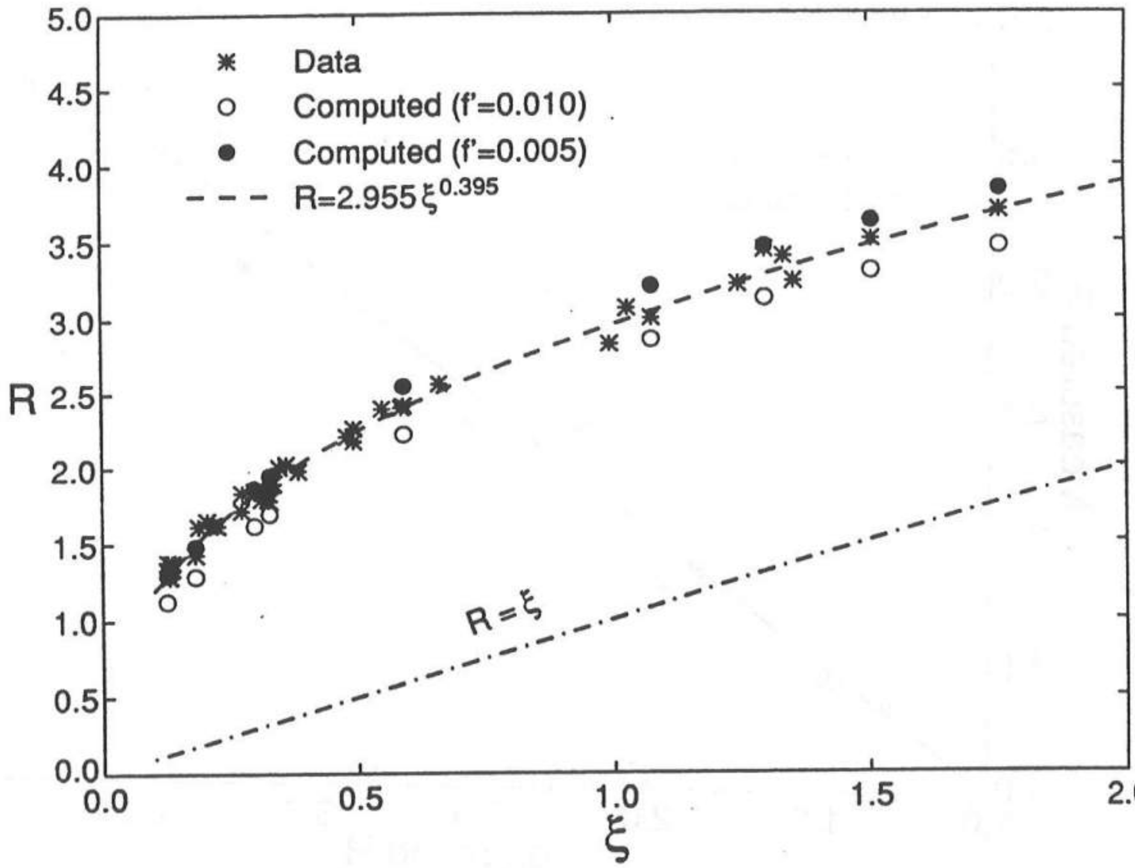

FIGURE 37. Measured and Computed Values of R as a Function of Surf Similarity Parameter ξ.

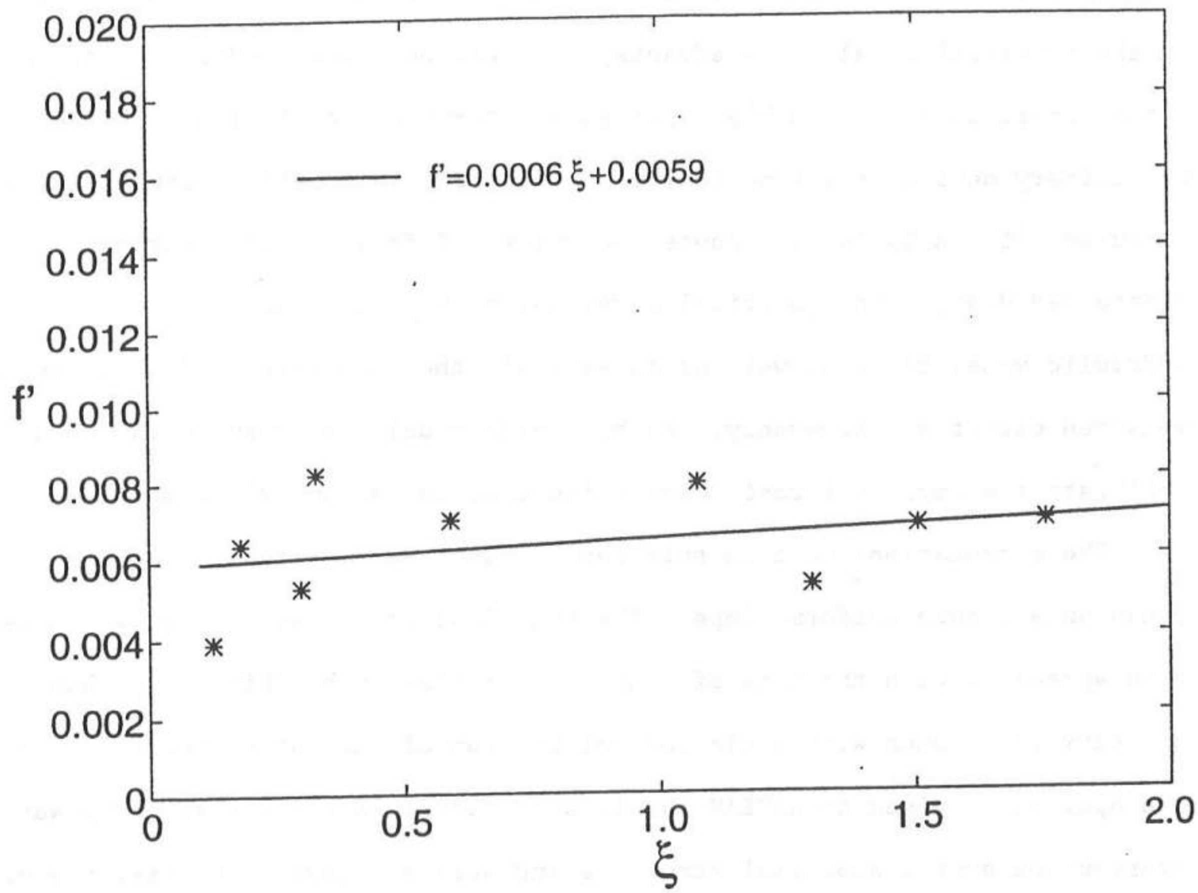

FIGURE 38. Fitted Values of Bottom Friction Factor f' as a Function of ξ.

5. CONCLUSIONS

The computer program SBREAK presented herein simulates the interaction of normally incident waves with a rough or smooth impermeable coastal structure or beach 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.

The computations made in this report have been limited to solitary wave runup on a smooth uniform slope. The numerical model has been shown to be in good agreement with the data of Synolakis (1987a) on breaking or broken solitary wave runup with a limited calibration of the bottom friction factor. The options provided in SBREAK should allow users to compute solitary wave overtopping over a subaerial structure and solitary wave transmission over a submerged structure. Furthermore, SBREAK may also be applied to compute the hydraulic stability and sliding motion of individual armor units under the action of solitary waves if the structure is protected with armor units. It is recommended to calibrate and verify SBREAK if it is to be applied to the other problems associated with solitary waves. Reference may be made to the previous work for monochromatic and random waves described in Section 1. The comparisons between solitary and monochromatic wave runup on smooth uniform slopes have been discussed in Section 4 by introducing the representative wave period and the surf similarity parameter for solitary waves. The breaking, runup and reflection of solitary and monochromatic waves on smooth uniform slopes are qualitatively similar in terms of the surf similarity parameter. For given surf similarity parameter, breaking solitary wave runup is definitely larger than breaking monochromatic wave runup affected by the interaction between wave uprush and downrush on the slope.

The present numerical model is probably the simplest one-dimensional, time-dependent model for breaking or broken waves on slopes. Various numerical models are being developed by other researchers (e.g., Zelt, 1991; Liu and Cho, 1993). It is desirable to compare the capabilities and limitations of available numerical models for different problems. These comparisons will provide a guideline for selecting the most appropriate model for a specific problem. Since such comparisons have not been made yet, users of SBREAK will need to judge whether SBREAK is appropriate for specific applications.

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. 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.

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

Cox, D.T., Kobayashi, N. and Wurjanto, A. (1991). "Computer programs for spectral and time series analyses for random waves." Res. Rept. No. CACR-91-06, Ctr. for Applied Coast. Res., Univ. of Delaware, Newark, Del.

Dean, R.G. and Dalrymple, R.A. (1984). Water Wave Mechanics for Engineers and Scientists, Prentice-Hall, Englewood Cliffs, N.J.

Hall, J.V. and Watts, J.W. (1953). "Laboratory investigation of the vertical rise of solitary waves on impermeable slopes." Tech. Memo. No. 33, Beach Erosion Board, U.S. Army Corps of Engineers.

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

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., 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). "Wave transmission over submerged breakwaters." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 115(5), 662-680.

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

Kobayashi, N. and Wurjanto, A. (1989d). "Armor stability on rough permeable slopes of marine structures." Proc. 23rd Congress of Intl. Assoc. Hydraulic Res., Ottawa, Canada, C, 407-414.

Kobayashi, N. and Wurjanto, A. (1989e). "Numerical prediction of hydrodynamic forces and sliding motion of dolos units." Proc. Stresses in Concrete Armor Units, ASCE, 355-378.

Kobayashi, N. and Wurjanto, A. (1990). "Numerical model for waves on rough permeable slopes." J. Coast. Res., SI(7), 149-166.

Kobayashi, N., Wurjanto, A. and Cox, D.T. (1990a). "Irregular waves on rough permeable slopes." J. Coast. Res., SI(7), 167-184.

Kobayashi, N., Wurjanto, A. and Cox, D.T. (1990b). "Rock slopes under irregular wave attack." Proc. 22nd Coast. Engrg. Conf., ASCE, 1306-1319.

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

Kobayashi, N., Cox, D.T. and Wurjanto, A. (1991). "Permeability effects on irregular wave run-up and reflection." J. Coast. Res., 7(1), 127-136.

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

Kobayashi, N. and Wurjanto, A. (1993). "Irregular wave interaction with permeable slopes." Proc. 23rd Coast. Engrg. Conf., ASCE, 1299-1312.

Liu, P. L.-F. and Cho, Y.-S. (1993). "Run-up of long waves." J. Wtrway. Port Coast. and Oc. Engrg., ASCE (submitted).

Losada, M.A., Kobayashi, N. and Martin, F.L. (1992). "Armor stability on submerged breakwaters." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 118(2), 207-212.

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.

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.

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.

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

Synolakis, C.E. (1987a). "The runup of solitary waves." J. Fluid Mech., 185, 523-545.

Synolakis, C.E. (1987b). "The runup and reflection of solitary waves." Proc. Coast. Hydrodynamics, ASCE, 533-547.

Synolakis, C.E. and Skjelbreia, J.E. (1993). "The evolution of the maximum amplitude of solitary waves on plane beaches." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 119(3), 323-342.

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.

Van der Meer, J.W. and Breteler, M.K. (1990). "Measurement and computation of wave induced velocities on a smooth slope." Proc. 22nd Coast. Engrg. Conf., ASCE, 191-204.

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

Wurjanto, A. and Kobayashi, N. (1992). "Numerical model for random waves on permeable coastal structures." Res. Rept. No. CACR-92-02, Ctr. for Applied Coast. Res., Univ. of Delaware, Newark, Del.

Wurjanto, A. and Kobayashi, N. (1993). "Irregular wave reflection and run-up on permeable slopes." J. Wtrway. Port Coast. and Oc. Engrg., ASCE, 119(5) (in press).

Zelt, J.A. (1991). "The run-up of nonbreaking and breaking solitary waves." Coastal Engineering, 15, 205-246.

APPENDIX: LISTING OF SBREAK

The computer program SBREAK is listed in the following. The listed computer program together with the tabulated input and output for the example presented in Section 4 is available on a diskette. The computer program SBREAK has been run using a SUN SPARC-IPC machine. Some modifications may be required if other computers are used. The CPU time for each of the 9 comoputed tests with f′ = 0.01 and 0.005 listed in Table 11 is listed in Table 12 and about 10 min.

TABLE 12. CPU Time Using SUN SPARC-IPC Machine

Test No.	CPU Time in Minutes	
	f′ = 0.01	f′ = 0.005
7	9.1	9.4
15	14.9	14.9
18	8.8	9.1
21	8.7	8.9
23	9.2	9.5
32	8.2	8.3
37	14.6	7.5
39	7.5	14.4
40	14.2	14.3

```
C
C    #######   #######   #######   ########   ######   ##   ##
C   ##    ##   ##   ##   ##   ##   ##         ##   ##   ##  ##
C   ##         ##   ##   ##   ##   ##         ##   ##   ## ##
C    #######   #######   #######   #######    ########   ####
C          ##   ##   ##   ##  ##    ##         ##   ##   ######
C   ##    ##   ##   ##   ##   ##   ##         ##   ##   ###  ##
C    #######   #######   ##    ##   ########   ##   ##   ##   ##
C
C                       Numerical Simulation of
C        Solitary Waves on Impermeable Beaches and Breakwaters;
C                Extension of Computer Program IBREAK
C
C       Nobuhisa Kobayashi, Entin A. Karjadi and Andojo Wurjanto
C              Center for Applied Coastal Research
C          University of Delaware, Newark, Delaware 19716
C                          March, 1993
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,KTWO
      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
C     /IDREQ/   Integers for special storing for armor stability
```

```
C              (see Subroutine 2 INPUT1 for special storing)
C     /TLEVEL/   Integers related to time levels
C     /NODES/    Integers for spatial nodes
C     /GRID/     Time step, grid size, and related quantities
C     /WAVE1/    Dimensional wave data
C     /WAVE2/    Dimensionless wave parameters
C     /WAVE3/    Normalized surface elevations
C     /WAVE4/    Max. and min. of normalized incident wave profile
C     /WAVE5/    Cnoidal wave parameters (K, E, m and 1-m)
C     /WAVE6/    Solitary wave parameters (Tc and K2)
C     /BOT1/     Dimensional parameters related to structure
C     /BOT2/     Normalized parameters related to structure
C     /BOT3/     Normalized structure geometry
C     /BOT4/ and /BOT5/ Dimensional structure geometry
C     /HYDRO/    Hydrodynamic quantities computed
C     /MATRIX/   Elements of matrices used in numerical method
C     /RUNP1/ and /RUNP2/ Quantities related to wave runup
C     /OVER/     Quantities related to wave overtopping
C     /COEFS/    Reflection and transmission coefficients
C     /STAT/     Mean, max. and min. of hydrodynamic quantities
C     /ENERG/    Quantities related to wave energy
C     /STAB1/    Armor stability parameters read as input
C     /STAB2/    Computed armor stability parameters
C     /STAB3/    Armor stability parameters used in Subr. 19 STABNO
C     /STAB4/    Armor movement parameters used in Subr. 20 MOVE
C     /STAB5/ and /STAB6/ Stability numbers and associated quantities
C                in Subr. 19 STABNO
C     /STAB7/ and /STAB8/ Quantities associated with armor movement
C                in Subr. 20 MOVE
C     /FILES/    File names and associated node numbers related to
C                options ISAVB=1 and ISAVC=1
C     /VALUEN/   Values at time level (N-1) stored during computation
      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 /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 /WAVE3/  ETA(N2),ETAIS(N2),ETARS(N2),ETATS(N2)
      COMMON /WAVE4/  ETAMAX,ETAMIN
      COMMON /WAVE5/  KCNO,ECNO,MCNO,KC2
      COMMON /WAVE6/  TCSOL,KTWO
      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 /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 /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 /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=2
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.EQ.2) 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 periodic 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=2
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 periodic wave profile at seaward bounda
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')
      ELSEIF (IWAVE.EQ.2) THEN
        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,KTWO
      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 /WAVE4/  ETAMAX,ETAMIN
      COMMON /WAVE6/  TCSOL,KTWO
      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                 regular and irregular waves on beaches
C          IENERG=0: Energy quantities NOT computed
C               =1: Energy quantities computed
C          IWAVE=1: Incident periodic waves at seaward boundary computed
C               =2: Incident waves at seaward boundary given as input
C               =3: Incident solitary waves at seaward boundary computed
C          If IWAVE=2 --> 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=2 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 = numerical 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                   TP of solitary wave is computed later if IWAVE=3
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
      KS = KSSEA/KSREF
      IF (IWAVE.LT.3) THEN
      SIGMA = TP*DSQRT(GRAV/HREFP)
      ENDIF
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     DSEA = normalized water depth below SWL at seaward boundary
      DSEA = DSEAP/HREFP
C
C     For IWAVE=3, the following wave parameters associated with
C         the incident solitary wave need to be computed.
C         The normalized crest arrival time tc and the value of
C         Delta-i associated with the reference wave period TP are
C         taken as
      IF (IWAVE.EQ.3) THEN
C ...   The maximum and minimum incident wave elevations are KS
C       and zero, respectively
        ETAMAX = KS
        ETAMIN = 0.D+00
C
        TCSOL  = 1.D+00
        DELTAI = 5.D-02
C     where these values can be modified by changing the above values only.
C     Then, the parameters K2 and SIGMA are given by
         DUM    = KS/DELTAI
         KTWO   = 2.D+00*DLOG(DSQRT(DUM)+DSQRT(DUM-1.D+00))
         SIGMA  = 2.D+00*DSEA*KTWO/DSQRT(3.D+00*(KS+KS*KS/DSEA))
C     The reference wave period TP is thus found
         TP     = SIGMA*DSQRT(HREFP/GRAV)
      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 incident wave profile data at
C     seaward boundary if IWAVE=2
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=2)
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'/
     +' Incident 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     Note : KS and DSEA have been computed in Subr.02 INPUT1
C
      WL0P   = GRAV*(TP*TP)/(2.D+00*PI)
      WL0    = WL0P/DSEAP
C     KS     = KSSEA/KSREF
      KSI    = SIGMA*TSLOPS/DSQRT(2.D+00*PI)
C     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,KTWO
      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 /WAVE4/  ETAMAX,ETAMIN
      COMMON /WAVE6/  TCSOL,KTWO
      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 regular or irregular wave set-down
C            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
C       For IWAVE=3, the normalized solitary wave profile is given by
        DUM      = KTWO*(N*T-TCSOL)
        ETA(N)   = KS/(DCOSH(DUM)**2)
        U(2,1)   = DSEA+ETA(N)+ETAR
        ELEV(1)  = ETA(N)+ETAR
C ... The maximum and minimum value of ETA should be equal to KS and zero,
C       respectively
        IF(ETA(N).GT.ETAMAX) THEN
          WRITE  (*,2910)
          WRITE (29,2910)
          STOP
        ELSEIF(ETA(N).LT.ETAMIN) THEN
          WRITE  (*,2920)
          WRITE (29,2920)
          STOP
        ENDIF
      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)
        ELSE
          ETAI  = ETA(N)
        ENDIF
        ETAIS(N-NSTAT+1) = ETAI
        ETARS(N-NSTAT+1) = ETAR
      ENDIF
C
 2910 FORMAT (/' From Subr. 16 SEABC:'/
     +          ' ETAMAX exceeds KS')
 2920 FORMAT (/' From Subr. 16 SEABC:'/
     +          ' ETAMIN is less than zero')
      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 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,KTWO
      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 /WAVE6/  TCSOL,KTWO
      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)TCSOL,KTWO
      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 Read as Input')
 2815 FORMAT (/'Solitary Incident Wave at Seaward Boundary'/
     +        'Tc        = ',D20.9/
     +        'K2        = ',D20.9)
 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 (2I8)
```

```
 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
C
        IF (IJOB.GT.1) C(JE)=DSQRT(U(2,JE))
        IF (IJOB.LT.3) THEN
          WRITE (31,3110) N,S
          WRITE (31,3120) (RUNUPS(L),L=1,NDELR)
```

```
          IF (IJOB.EQ.2) WRITE (32,3210) N,U(1,JE),U(2,JE),V(JE),C(JE)
        ELSE
          WRITE (33,3310) N,U(1,JE),V(JE),C(JE),ETAT
        ENDIF
C
C ..... STORING VALUES AT SEAWARD BOUNDARY
C
C       File 21 = 'OSEAWAV'
C       N = current time level
C       Surface elevations at seaward boundary:
C       . ETAI --> due to incident wave
C       . ETAR --> due to reflected wave
C       . ETATOT = ETAI+ETAR
C       At node j:
C         V(j)   = depth-averaged velocity
C         U(1,j) = volume flux
C
        IF (IWAVE.EQ.1) THEN
          NWAVE  = MOD(N,NONE) + 1
          ETAI   = ETA(NWAVE)
          ETATOT = ETAI+ETAR
        ELSE
          ETAI   = ETA(N)
          ETATOT = ETAI+ETAR
        ENDIF
        WRITE (21,2110) N,ETAI,ETAR,ETATOT,V(1),U(1,1)
C
      ELSE
C
C ..... SPECIAL STORING IF ICALL=3
C
C       File 40 = 'OREQ'
C       S = waterline node (IJOB<3) or landward-end node (IJOB=3)
C       At node j for specified time level N:
C         ELEV(j) = surface elevation above SWL
C         V(j)    = depth-averaged velocity
C         DUDT(j) = total fluid acceleration
C         SNR(j)  = stability number against rolling/sliding
C
        WRITE (40,4010) N,S
        IF (IELEV.EQ.1) WRITE (40,4020) (ELEV(J),J=1,S)
        IF (IV.EQ.1)    WRITE (40,4020) (V(J),J=1,S)
        IF (IDUDT.EQ.1) WRITE (40,4020) (DUDT(J),J=1,S)
        IF (ISNR.EQ.1)  WRITE (40,4020) (SNR(J),J=1,S)
C
      ENDIF
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.',
     7 ' Part of the structure is above SWL.',
     8 ' TRANSMISSION computation can not be performed.'/
      DO 100 I = IBEGIN,IEND
        WRITE  (*,2910) MSG(I)
        WRITE (29,2910) MSG(I)
  100 CONTINUE
      WRITE  (*,2920)
      WRITE (29,2920)
 2910 FORMAT (A55)
 2920 FORMAT (' Programmed Stop.')
C
      STOP
      END
C
C -38----------------- END OF SUBROUTINE STOPP ---------------------
```