NUMERICAL MODEL RBREAK2 FOR RANDOM WAVES ON IMPERMEABLE COASTAL STRUCTURES AND BEACHES

by

Nobuhisa Kobayashi

and

Michael T. Poff

Sponsored by
U.S. Army Coastal Engineering Research Center

RESEARCH REPORT NO. CACR-94-12

June 1994

CENTER FOR APPLIED COASTAL RESEARCH

OCEAN ENGINEERING LABORATORY
UNIVERSITY OF DELAWARE
NEWARK, DELAWARE 19716

Abstract

A computer program called RBREAK2 is presented in this report. RBREAK2 extends our previous computer program RBREAK by allowing the cross-shore variation of the bottom friction factor and the specification of the measured total free surface oscillation at the seaward boundary as input to RBREAK2. Most coastal structures in the U.S.A. are located in relatively shallow water and designed against depth-limited breaking irregular waves. Since no wave model is available to estimate the incident breaking waves at the toe of such a structure in the presence of appreciable reflected waves, the seaward boundary of RBREAK2 is normally taken to be immediately outside the surf zone. Linear and nonlinear wave propagation models outside the surf zone are available and may be used to predict the incident irregular wave train required as input to RBREAK2. The bottom of the computational domain in this typical application of RBREAK2 may vary from a gentle sandy beach to a steep quarry stone slope with a smooth or rough horizontal crest. The cross-shore variation of the bottom roughness and corresponding friction factor will need to be included in such a case. On the other hand, the specification of the measured total free surface oscillation as input is convenient when RBREAK2 is compared with laboratory and field data for which incident and reflected waves are unknown or can not be separated using linear wave theory.

RBREAK2 may be used to predict the interactions of impermeable coastal structures and beaches of arbitrary geometry and roughness with normally-incident waves. Any incident or total wave profile can be specified by a user at the seaward boundary of the computation domain. RBREAK2 computes the reflected wave train at the seaward boundary using the method of characteristics. At the landward boundary, wave runup, overtopping or transmission is computed depending on the crest elevation of a structure or dune (or sand bar). In addition to the one-dimensional, time-dependent equations of mass and momentum used to compute the water depth and depth-averaged horizontal velocity, an equation of energy is used to estimate the rate of energy dissipation due to wave breaking without modeling wave breaking processes explicitly. Furthermore, RBREAK2 computes the hydraulic stability or sliding motion of individual armor units under the action of the computed flow if requested by a user.

This report presents the governing equations, numerical procedures, computer programs, input and output involved in RBREAK2 in such a way that users will be able to apply RBREAK2 effectively and modify it if necessary. Our previous work based on RBREAK2 and related numerical models is summarized concisely so that users may be able to make the best use of our previous work. New experimental data on breaking irregular wave overtopping of steep rough and smooth slopes fronted by gentle smooth fixed and movable slopes is summarized for the use of other coastal engineers. Wave overtopping processes were not affected much by beach profile changes in the movable bed experiment. RBREAK2 is then compared with the tests for a 1:2

rough slope with a smooth horizontal crest fronted by a gentle smooth slope. RBREAK2 using empirical parameters calibrated previously is found to predict the highly unsteady overtopping flow characteristics within errors of about ±100%. The overtopping of breaking irregular waves turns out to be very sensitive to several parameters and hard to predict very accurately.

ACKNOWLEDGEMENT

This is the second report resulting from research sponsored by the Coastal Engineering Research Center, U. S. Army Engineer Waterways Experiment Station under contract No. DACW 39-93-C-0074. The authors would like to thank D. L. Ward, J. A. Melby, and D. D. Davidson for their enthusiastic support of our continued efforts to improve the capabilities of numerical models for the design of coastal structures. The authors would also like to thank A. W. Raichle, D. T. Cox and E. A. Karjadi for their contributions to the development of the computer program RBREAK2 and its users' manual.

CONTENTS

PART I
INTRODUCTION

• 1.1 •
BACKGROUND

In the planning and design of rubble mound breakwaters and riprap revetments against incident wind waves, it is generally required to predict irregular wave runup, overtopping, transmission, and reflection as well as the hydrodynamic stability and movement of armor units (*e.g.*, Shore Protection Manual 1984; Bruun 1985). On the other hand, the quantitative understanding of irregular wave transformation and runup on beaches is essential for assessing shoreline protection measures against flooding and beach erosion caused by storms (*e.g.*, Kobayashi 1988).

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

On the other hand, numerical models are generally less accurate and versatile than hydraulic model tests, especially for problems involving irregular breaking and broken waves. The advantages of numerical models are low cost, little start-up time, and high spatial and temporal resolutions for interpreting physical processes. During a preliminary design, numerical models may be used together with empirical formulas, if available, to reduce the number of feasible alternatives. During a detailed design, numerical models 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 numerical models.

Considering the need of numerical models for the problems which may not be modeled using potential flow theory, the authors and co-workers have developed, calibrated, and evaluated one-dimensional, time-dependent numerical models as summarized in the following for those who are not familiar with our previous publications.

• 1.2 •

Previous Work Based on IBREAK

Kobayashi and Wurjanto (1989c) presented a computer program called IBREAK. IBREAK may be used for the design of rough or smooth impermeable coastal structures of arbitrary geometry against normally-incident waves. Any incident wave train can be specified as input at the seaward boundary of the computation domain, although the computation was limited to monochromatic and transient waves. IBREAK computes the reflected wave train at the seaward boundary from which the reflection coefficient is predicted. For a subaerial structure, IBREAK computes wave runup on the seaward slope of the structure or wave overtopping over the crest of the structure if it is not high enough to prevent flow over its crest. For a submerged structure, IBREAK computes the transmitted wave train at the landward boundary of the computation domain from which the transmission coefficient is predicted. In addition to the equations of mass and momentum used to compute the flow field for any structure, an equation of energy is used in IBREAK to estimate the spatial variations of energy dissipation rates due to wave breaking and bottom friction. Furthermore, IBREAK computes the hydraulic stability and sliding motion of individual armor units under the action of the computed flow if the structure is protected with armor units. In addition to coastal structures, IBREAK can be used to predict the wave transformation in the surf zone and the resulting swash oscillation on a beach. In the following, the previous work based on IBREAK is summarized.

Kobayashi *et al.* (1986, 1987) developed a numerical flow model to predict the flow characteristics on a rough impermeable slope for specified normally-incident monochromatic waves. The numerical flow model was based on the finite-amplitude shallow-water equations including the effects of bottom friction (Madsen and White 1976) which were solved numerically in the time domain using an explicit dissipative Lax-Wendroff finite difference method (Richtmyer and Morton 1967; Hibberd and Peregrine 1979; Packwood 1980). A recent review of numerical methods developed for flows with shocks was given by Moretti (1987). The adopted numerical method is a shock-capturing method for which a separate treatment of a wave front (shock) is not required, although it can not describe the detailed behavior of breaking waves (*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 monochromatic wave runup, run-down, and reflection fairly well, although the friction factor needs to be calibrated.

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)

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 free surface and waterline oscillation well. The prediction of the dynamic pressure was reasonable. The spatial variations of the amount of the gravel movement was predicted only qualitatively.

Kobayashi and Watson (1987) applied the developed numerical flow model to predict monochromatic 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.

The numerical flow model developed for coastal structures was also modified slightly to predict the flow characteristics in the swash zone on a beach. The applications of the numerical flow model for predicting the monochromatic or transient wave transformation and swash oscillation on beaches were presented by Kobayashi *et al.* (1988, 1989). On the other hand, 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 landward and seaward boundary conditions. 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 fair agreement with the small-scale monochromatic wave 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 landward boundary condition. In addition to the equations of mass and momentum used to compute the flow field, an equation of energy was derived to estimate the rate of energy dissipation due to wave breaking. The computed reflection and transmission coefficients were shown to be in fair agreement with the small-scale monochromatic wave 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 which occurred in 1987. 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; Tørum 1992, Mase and Kobayashi 1993) applied the numerical model IBREAK or a similar numerical model to predict the monochromatic and transient wave motions on coastal structures and beaches. The accuracy of the numerical models was evaluated using their own laboratory measurements. As a whole, their results are consistent with our experiences with IBREAK. The numerical model IBREAK is fairly versatile and robust although it may not predict some quantities very accurately.

• 1.3 •
Previous Work Related to 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 an unnecessary increase in the computation time, Wurjanto and Kobayashi (1991) 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. The computer program RBREAK developed by Wurjanto and Kobayashi (1991) is 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 runup 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 in the 1:3 slope. These low-frequency components were secondary in comparison to the corresponding wind-wave components unlike swash oscillations on natural beaches (Guza and Thornton 1982).

Kobayashi and Wurjanto (1992a) 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 these one-dimensional equations that are used to predict normally-incident wave motions on coastal structures and beaches. 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 (Holman and Sallenger 1985). 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 (Elgar and Guza 1985a, 1985b). RBREAK was shown to predict the dominant low frequency components of the measured spectrum fairly well.

Cox *et al.* (1992) developed a frequency-domain model for elucidating the nonlinear transformation processes of the Fourier amplitudes and phases of normally incident random waves in surf and swash zones. The continuity and momentum equations used in RBREAK were rearranged to derive the equations expressing the cross-shore variations of the Fourier components of normally incident random waves. The derived equations were solved numerically using forcing terms computed by RBREAK. The frequency-domain model attempted to quantify the importance of the nonlinear forcing due to the cross-shore variations of instantaneous radiation stress and bottom shear stress as well as the seaward boundary conditions related to incoming low frequency waves for generating two-dimensional surf beat in the surf and swash zones. The computed low-frequency wave motions appeared to be standing waves but modified by the forcing terms, especially in the swash zone.

Raubenheimer *et al.* (1994) compared RBREAK to observations of waves in the surf and swash zones on a fine grained, gently sloping natural beach. Pressure fluctuations were measured at 6 seabed locations on a cross-shore transect in very shallow water. The incident wave trains were estimated using observations from the most offshore pressure sensor and a current meter collocated about 50m from the mean shoreline. Swash oscillations were measured with a vertical stack of runup wires positioned at 5, 10, 15, 20 and 25cm above the beach face. Runup measurements were sensitive to the wire elevation, owing to thin concave runup tongues. As the wire elevation increased, the measured mean runup location moved seawards, low frequency infragravity energy decreased and higher frequency sea-swell energy increased. These trends, and the variation of wave energy and wave shape (e.g., wave skewness) across the inner surf zone, were well predicted by RBREAK.

• 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. In addition to the continuity and momentum equations used to compute the flow field, an equation of energy was used to estimate the rate of energy dissipation due to wave breaking. 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 periods 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 inappropriate for additional 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 (1992, 1993) developed the one-dimensional, time-dependent numerical model PBREAK 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 (1992b) compared the computed results for the rough permeable and impermeable slopes to quantify the differences caused by the thick permeable underlayer.

Van Gent (1994) developed a numerical model similar to PBREAK developed by Wurjanto and Kobayashi (1992). The major difference between his model and PBREAK is the treatment of the waterline on the rough permeable slope. The free surface elevation outside and inside the rough permeable slope is assumed to be continuous at the waterline in PBREAK. Van Gent (1994) allows the discontinuity of the free surface at the waterline but an empirical procedure is required to describe the discontinuity. Lissev (1993) compared the computed results using these two computer programs and found noticeable differences. Detailed measurements in the vicinity of the waterline on the rough permeable slope will be required to determine how the waterline should be treated in numerical modeling.

• 1.5 •

PREVIOUS WORK RELATED TO SBREAK

Kobayashi and Karjadi (1993, 1994) developed a computer program called SBREAK by expanding IBREAK to predict the runup of solitary waves on beaches. SBREAK was calibrated and evaluated using the data of Synolakis (1987) and Hall and Watts (1953) 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 were identified using the normalized incident solitary wave profile and governing equations. The representative solitary wave period and associated surf similarity parameter were introduced so as to examine the similarity and difference between solitary and monochromatic waves on smooth uniform slopes using the surf similarity parameter proposed by Battjes (1974) for monochromatic waves. The breaking, runup and reflection of solitary and monochromatic waves are qualitatively similar in terms of the surf similarity parameter (Karjadi and Kobayashi 1994). 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. SBREAK was shown to be in good agreement with the data of Synolakis (1987) on breaking or broken solitary wave runup with a limited calibration of the bottom friction factor.

• 1.6 •

DEVELOPMENT OF RBREAK2 AND OUTLINE OF REPORT

The computer program RBREAK2 reported herein has been developed by expanding RBREAK developed by Wurjanto and Kobayashi (1991) in two steps. First, Kobayashi and Raichle (1993, 1994) allowed the spatial variation of the bottom friction factor to account for the cross-shore variation of bottom roughness that tends to occur for coastal structures exposed to depth-limited wave conditions in shallow water. RBREAK2 was compared with the experiment on breaking irregular wave overtopping over a quarry stone revetment fronted by a gentle smooth slope. A three wave gage method was used to separate the incident and reflected wave trains immediately outside the breaker zone on the gentle slope. The separated incident wave train was specified as input to RBREAK2. It is noted that no time-averaged model is presently available for predicting wave transformation in the surf zone in the presence of appreciable reflected waves. Limited calibrations were made of the bottom friction factors for the rough and smooth slopes and the fraction of entrained air in overtopping flow. The calibrated numerical model was capable of predicting the highly unsteady overtopping flow characteristics within errors of about ±100%. Second, Cox *et al.* (1994) allowed the specification of the measured total free surface oscillation as input at the seaward boundary when RBREAK2 was compared with the free surface measurements in the surf and swash zones on sand beaches from the SUPERTANK Laboratory Data Collection Project. The incident and reflected waves could not be separated from these specific measurements. RBREAK2 with the modified seaward boundary algorithm was capable of predicting the free surface oscillations in the surf and swash zones within errors of about ±20%.

This report is written in such a way that users should be able to run RBREAK2 without reading the report of RBREAK. Part II of this report concisely explains the equations and numerical procedures used in the computer program RBREAK2. The computer program RBREAK2 listed in Appendix B is explained in detail in Part III, although the listed program is written in self-explanatory manners. The additional explanations will facilitate usage of RBREAK2 and modifications of RBREAK2 by users if necessary. Since RBREAK2 includes a large number of parameters, variables, and subroutines in its present form, a separate computer program called BEFORR2 is developed to implement the automated adjustment procedure discussed in Section 1.3. The computer program BEFORR2 listed in Appendix C is explained in Part IV. BEFORR2 is run before RBREAK2 to find the time series of appropriate values of the time step size Δt and the small water depth used to define the location of the computational waterline or shoreline. The computed time series of these values are read as input to RBREAK2 to ensure a successful computation without any numerical problem. This arrangement requires users to run both BEFORR2 and RBREAK2. Our experiences indicate that running both programs should be much easier than running a large number of trial runs to complete the computation of a long duration.

Part V summarizes the irregular wave overtopping experiments presented in the thesis of Poff (1993) and compares RBREAK2 with two tests on irregular wave overtopping on a smooth crest of a 1:2 rough slope fronted by a gentle smooth slope. For each test, two computational runs are made by specifying the measured total free surface oscillation and the measured incident wave train as input to RBREAK2 at its seaward boundary which is located at the most seaward gage of three wave gages. Four subroutines used to separate the incident and reflected waves are explained in Part VI and listed in Appendix A. The agreement between the measured and computed overtopping flow characteristics for both runs is found to be similar to the agreement between RBREAK2 and the rough slope tests of Kobayashi and Raichle (1993, 1994). In addition, the method of characteristics used for predicting the reflected wave train at the seaward boundary in RBREAK2 is compared with the three gage method based on the computed free surface oscillations stored at the locations of the three wave gages. Both methods are found to predict practically the same reflected wave train, although it was hoped that the three wave gage method would improve the agreement between the measured and computed reflected waves.

Part II
Numerical Model

• 2.1 •
Governing Equations

The wave motion on a rough or smooth impermeable slope is computed for the normally incident waves in 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 of the computation domain
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; θ' may vary along the slope
η' = free surface elevation above SWL
h' = water depth above the impermeable slope
u' = depth-averaged horizontal velocity

For the flow over the impermeable slope, the vertically-integrated equations for mass and x'-momentum may be expressed as (Kobayashi, *et al.* 1987; Kobayashi and Wurjanto 1992a)

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

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

where

t' = time
g = gravitational acceleration
f' = bottom friction factor related to the shear stress acting on the slope

In this simplified analysis, the friction factor f' accounts for the spatial variation of the roughness characteristics of the impermeable slope surface. Moreover, 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 are introduced to normalize Eqs. 1 and 2:

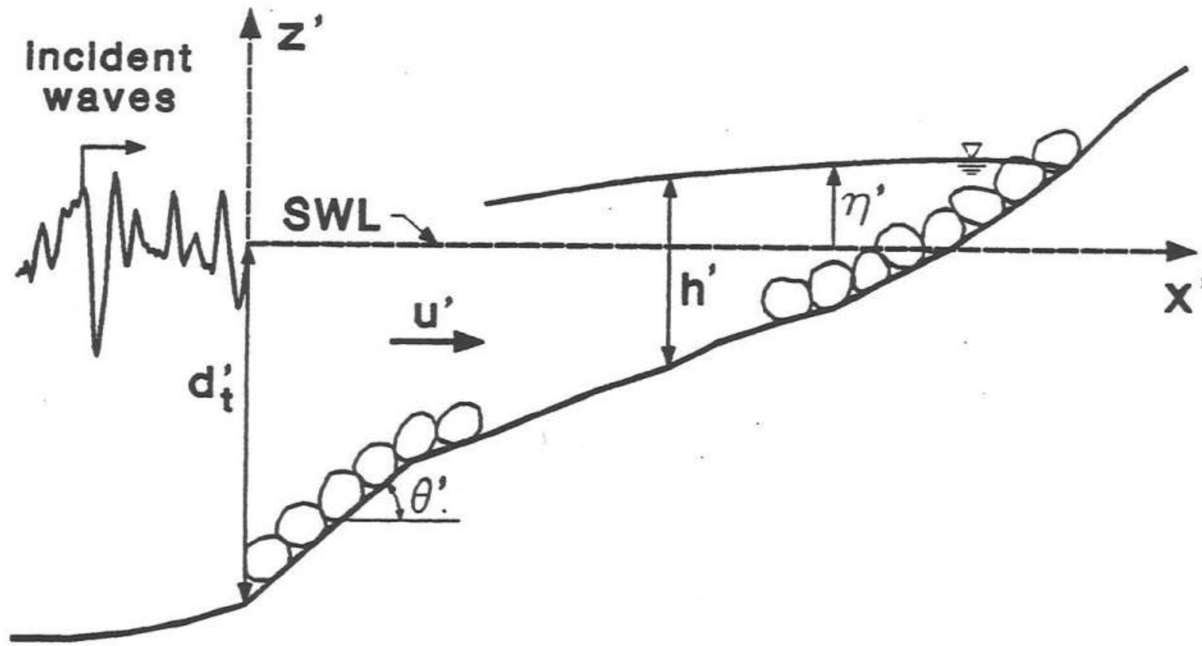

Figure 1: Wave runup on a rough slope for IJOB=1.

$$t = \frac{t'}{T'_r} \quad ; \quad x = \frac{x'}{T'_r\sqrt{gH'_r}} \quad ; \quad u = \frac{u'}{\sqrt{gH'_r}} \tag{3}$$

$$z = \frac{z'}{H'_r} \quad ; \quad h = \frac{h'}{H'_r} \quad ; \quad \eta = \frac{\eta'}{H'_r} \quad ; \quad d_t = \frac{d'_t}{H'_r} \tag{4}$$

$$\sigma = T'_r\sqrt{\frac{g}{H'_r}} \quad ; \quad \theta = \sigma \, tan\theta' \quad ; \quad f = \frac{1}{2}\sigma f' \tag{5}$$

where

T'_r = reference wave period
H'_r = reference wave height
σ = ratio between the horizontal and vertical length scales
θ = normalized gradient of local slope
f = normalized bottom friction factor

The numerical model RBREAK2 assumes that $\sigma^2 \gg 1$ and $(\cot\theta')^2 \gg 1$ (Kobayashi *et al.* 1987; Kobayashi and Wurjanto 1992a). The reference 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 are allowed to vary with respect to x and 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, $\xi = \theta/\sqrt{2\pi}$. 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 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:

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

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

with $$c = \sqrt{h} \quad ; \quad \alpha = u + 2c \quad ; \quad \beta = -u + 2c \tag{11}$$

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)

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

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

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 may be specified by prescribing the variation of η_i with respect to $t \geq 0$. Alternatively, the free surface elevation measured at $x = 0$ may be specified as input by prescribing the variation of η with respect to $t \geq 0$. The normalized reflected wave train η_r is approximately expressed in terms of the seaward advancing characteristic β at $x = 0$

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

where β is given by Eq. 10 and linear long wave theory is used to derive Eq. 13. The correction term C_t in Eq. 13 introduced by Kobayashi, DeSilva, and Watson (1989) to predict wave set-down and setup on a beach may be expressed as

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

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 $$\mathbf{U} = \begin{bmatrix} m \\ h \end{bmatrix} \quad ; \quad \mathbf{F} = \begin{bmatrix} mu + \frac{1}{2}h^2 \\ m \end{bmatrix} \quad ; \quad \mathbf{G} = \begin{bmatrix} \theta h + f|u|u \\ 0 \end{bmatrix} \tag{16}$$

where $m = uh$ is the normalized volume flux per unit width. The vectors $\mathbf{F}$ and $\mathbf{G}$ depend on the vector $\mathbf{U}$ for given spatial variations of θ and f.

Eq. 15 is discretized using a finite difference grid of space size Δx and time step Δt based on an explicit dissipative Lax-Wendroff method (*e.g.*, Richtmyer and Morton 1967). The space size Δx is constant during the computation duration, whereas the time step Δt may vary in the following fashion:

> The computation duration is divided into *computation units*. Each computation unit is one wave period long except the last unit, the length of which may be less than one wave period. The *wave period* used as reference is the reference wave period, T'_r. The time step Δt is held constant within a computation unit, but may vary from one computation unit to another.

In the following, the known quantities at the node located at $x = (j-1)\Delta x$, with $j = 1, 2, ..., s$, and at the present time t are indicated by subscript $_j$ without superscript. The integer s indicates the wet node next to the moving waterline at the time 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 next time level $t^* = (t + \Delta t)$ are denoted by subscript $_j$ <u>and</u> an asterisk. The asterisk indicates the quantities at the time t^*. The values of $\mathbf{U}_1^*$ and $\mathbf{U}_j^*$ for $j \geq (s-1)$ are computed using the seaward and landward boundary conditions, respectively. The values of $\mathbf{U}_j^*$ for $j = 2, 3, ..., (s-2)$ are computed using the known values of $\mathbf{U}_{j-1}$, $\mathbf{U}_j$, and $\mathbf{U}_{j+1}$ at the time t

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

where $\lambda = \Delta t / \Delta x$ and $\mathbf{G}_j$ is given in Eq. 16 with $f = f_j$ at the node j. The vector $\mathbf{g}_j$ in Eq. 17 is given by

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

with

$$\mathbf{A}_j = \begin{bmatrix} 2u_j & (h_j - u_j^2) \\ 1 & 0 \end{bmatrix} \tag{19}$$

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

$$\mathbf{S}_j = \begin{bmatrix} \Delta x \; e_j - \frac{1}{2}\theta_j (m_{j+1} - m_{j-1}) \\ 0 \end{bmatrix} \tag{20}$$

with

$$
\begin{aligned}
e_j &= 2 f_j \frac{|u_j|}{h_j} \left[\left(u_j^2 - h_j \right) \frac{h_{j+1} - h_{j-1}}{2 \Delta x} \right. \\
&\quad \left. - u_j \frac{m_{j+1} - m_{j-1}}{2 \Delta x} - \theta_j h_j - f_j |u_j| u_j \right]
\end{aligned}
\tag{21}
$$

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

$$
\begin{aligned}
p_j &= - \frac{1}{2 \left(c_{j+1} + c_j \right)} \left[\, \epsilon_1 |v_{j+1} - v_j| \left(w_{j+1} + w_j \right) \right. \\
&\quad \left. - \epsilon_2 |w_{j+1} - w_j| \left(v_{j+1} + v_j \right) \, \right]
\end{aligned}
\tag{24}
$$

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

with

$$
c = \sqrt{h} \quad ; \quad v = u + c \quad ; \quad w = u - c \tag{26}
$$

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. Use has been made of $\epsilon_1 = \epsilon_2 = 1$ or $\epsilon_1 = \epsilon_2 = 2$ where the increase of $\epsilon_1 = \epsilon_2$ tends to improve numerical stability with negligible effects on computed results (Kobayashi and Wurjanto 1992a).

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

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

which requires that $|u_j| < (\Delta x / \Delta t)$ at any time for any j, where

u_j = normalized horizontal fluid velocity at the node j
u_m = maximum value of u expected to be encountered in the flow field
c_m = maximum expected value of $\sqrt{h}$
ϵ = the greater of the positive coefficients ϵ_1 and ϵ_2

The values of Δx, ϵ_1, and ϵ_2 need to be specified by a user. The value of Δt will be assigned automatically by the computer program BEFORR2, which should be executed before the computer program RBREAK2. The assigned value of Δt is based on the above stability criterion, numerical problems encountered, and convenience for storing computed time series at a specified time interval.

To facilitate the use of Eq. 27, the numerical stability indicator Ω is introduced.

$$\Omega = \frac{\Delta x}{\Delta t} \frac{\left[\sqrt{\left(1+\frac{1}{4}\epsilon^2\right)} - \frac{1}{2}\epsilon\right]}{1+\sqrt{d_t}} \tag{28}$$

where the normalized depth at the seaward boundary, d_t, has been defined in Eq. 4. If the following approximations are made

$$u_m \simeq 1$$
$$c_m \simeq \sqrt{d_t}$$

the numerical stability criterion given by Eq. 27 can then be expressed as

$$\Omega > 1 \tag{29}$$

In the computer program RBREAK2, Eq. 29 is actually used in lieu of Eq. 27 to determine whether the numerical stability criterion is violated or not. Detailed explanations for the numerical stability will be presented in Part IV.

• 2.3 •
INCIDENT OR MEASURED WAVE PROFILE

One of the required input for the computer program RBREAK2 is the normalized incident or measured wave profile at the seaward boundary of the computation domain, that is, $\eta_i(t) = \eta_i'(t')/H_r'$ or $\eta(t) = \eta'(t')/H_r'$ at $x = 0$, with $t = t'/T_r'$ as defined in Eq. 3, where H_r' and T_r' are the reference wave height and period used for the normalization in Eqs. 3-5. The specification of $\eta_i(t)$ or $\eta(t)$ at $x = 0$ for $t \geq 0$ needs to satisfy the condition $\eta_i = 0$ or $\eta = 0$ at $x = 0$ at the initial time $t = 0$ to be consistent with the assumed initial conditions of no wave action in the region of $x \geq 0$ at $t = 0$.

The temporal variation of $\eta_i(t)$ can be

1. the incident of monochromatic wave profile computed (by the computer program RBREAK2) using an appropriate wave theory, or

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

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

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

2.3.1 Incident Regular Wave (IWAVE=1)

For the case of incident regular waves identified by the integer IWAVE=1, the periodic variation of $\eta_i(t)$ is computed by the computer program RBREAK2 using either cnoidal or Stokes second-order wave theory. The height and period of the incident regular waves at the seaward boundary located at $x = 0$ are denoted by H' and T'. The reference wave period T'_r is taken as $T'_r = T'$ for the incident regular waves. 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, Strzelecki, and Wurjanto 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 regular wave profile $\eta_i(t)$ at $x = 0$ is K_s and unity, respectively.

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

$$\eta_i(t) = K_s\left\{\frac{1}{2}cos\left[2\pi(t+t_0)\right] + a_2 cos\left[4\pi(t+t_0)\right]\right\} \quad \text{for } t \geq 0 \tag{30}$$

with

$$a_2 = \frac{2\pi}{L}cosh\left(\frac{2\pi}{L}\right)\left[2 + cosh\left(\frac{4\pi}{L}\right)\right]\left[16\frac{d_t}{K_s}sinh^3\left(\frac{2\pi}{L}\right)\right]^{-1} \tag{31}$$

$$L = L_0\, tanh\frac{2\pi}{L} \tag{32}$$

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
$L = L'/d'_t$
$L_0 = L'_0/d'_t$
L' = dimensional wavelength at $x = 0$
L'_0 = dimensional linear wavelength in deep water

The normalized wavelength L satisfying Eq. 32 for given L_0 is computed using a Newton-Raphson iteration method. Eq. 31 yields the value of a_2 for given $d_t = d'_t/H'_r$, K_s, and L. Since Eq. 30 satisfies $\eta_i(t+1) = \eta_i(t)$ and $\eta_i(-t-t_0) = \eta_i(t+t_0)$, it is sufficient to compute the profile $\eta_i(t)$ for $0 \leq (t+t_0) \leq 0.5$. Eq. 30 may be appropriate if the Ursell parameter $U_r < 26$ where $U_r = [H'(L')^2/(d'_t)^3] = (K_s L^2/d_t)$ at $x = 0$. It is noted that the value of U_r based on the normalized wavelength L computed from Eq. 32 is simply used to decide whether cnoidal or Stokes second-order wave theory is applied.

For the case of $U_r \geq 26$, cnoidal wave theory (*e.g.*, Svendsen and Brink-Kjær 1972) is used to compute the incident wave profile $\eta_i(t)$ at $x = 0$

$$\eta_i(t) = \eta_{min} + K_s\, cn^2\left[2K(t+t_0)\right] \quad \text{for } t \geq 0 \tag{33}$$

with
$$\eta_{min} = \frac{K_s}{m}\left(1 - \frac{E}{K}\right) - K_s \tag{34}$$

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
m = parameter determining the complete elliptic integrals $K(m)$ and $E(m)$

The parameter m is related to the Ursell parameter U_r

$$U_r = \frac{K_s L^2}{d_t} = \frac{16}{3} m K^2 \tag{35}$$

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

$$\frac{\sigma}{L\sqrt{d_t}} \sqrt{\left[1 + \frac{K_s}{m\, d_t}\left(-m + 2 - 3\frac{E}{K}\right)\right]} - 1 = 0 \tag{36}$$

where the normalized wavelength L is given by Eq. 35 as a function of m for given d_t and K_s. The left hand side of Eq. 36 is a reasonably simple function of m in the range $0.8 < m < 1$. As a result, Eq. 36 can be solved using an iteration method which successively narrows down the range of m bracketing the root of Eq. 36. 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. 35, while Eq. 34 yields the value of η_{min}. The incident wave profile $\eta_i(t)$ is computed using Eq. 33 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. 30. 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 Incident Irregular Wave (IWAVE=2)

The specification of incident irregular waves as input to RBREAK2 is identified by IWAVE=2 and is the same as in RBREAK. The computer program RBREAK (Wurjanto and Kobayashi 1991) was an improvement over its predecessor, IBREAK (Kobayashi and Wurjanto 1989c), in the sense that RBREAK has the capability of overcoming numerical difficulties for the cases where the incident wave train of a long duration at the seaward boundary is specified by a user. IBREAK was used only for the wave train of a relatively short duration at the seaward boundary specified by a user (*e.g.*, Kobayashi and Greenwald 1986, 1988). The computer program RBREAK, on the other hand, was developed for user-specified wave trains of *long* durations.

The generation of numerically generated wave trains was explained in the report by Cox, Kobayashi, and Wurjanto (1991). Various examples of user-specified random wave trains were given in Appendices D through G in the RBREAK report (Wurjanto and Kobayashi 1991). These examples are directly applicable to RBREAK2.

Caution should be used to specify the heights and periods of random waves. This numerical model distinguishes the reference wave (H_r', T_r') from the representative wave at the seaward boundary (H', T'). The reference wave (H'_r, T'_r) is used to normalize the governing equations (Eqs. 1 and 2), whereas (H', T') is a representative wave of random waves being investigated at the seaward boundary. The representative wave height H' can be, for example, the incident significant wave height (Kobayashi, Cox, and Wurjanto 1990) or the spectral estimate of the incident significant wave height (Kobayashi and Wurjanto 1992a). The representative wave period T' can be the mean wave period (Kobayashi, Cox, and Wurjanto 1990) or the spectral peak period (Kobayashi and Wurjanto 1992a). Since the relation between the two waves is not simple for the case of random waves, it is recommended that the representative wave (H', T') at the seaward boundary be taken as the reference wave (H'_r, T'_r), as is the case with the computed results presented in Part V. Thus, $H' = H'_r$, $T' = T'_r$, and $K_s = H'/H'_r = 1$. $K_s = 1$ implies that the normalized height of the representative wave at the seaward boundary is unity.

2.3.3 Measured Wave Profile (IWAVE=3)

If the free surface oscillation is measured at the seaward boundary of the computation, it is more direct and straightforward to specify the measured free surface oscillation as input to RBREAK2. This option identified by IWAVE=3 eliminates the uncertainty associated with the separation of incident and reflected waves using linear wave theory, which is required for the option of IWAVE=2. Packwood (1980) found that the specification of the measured free surface oscillation at the seaward boundary of his numerical model based on the continuity and momentum equations similar to Eqs. 1 and 2 somehow produced spurious long-period oscillations in the computation domain. The implicit algorithm for IWAVE=3 described in Section 2.5.2 has not produced spurious long-period oscillations in the computation domain.

For IWAVE=3, the representative wave (H', T') associated with the measured free surface oscillation, which includes both incident and reflected waves, may be difficult to specify in a meaningful manner. It is hence recommended to take $H' = H'_r(K_s = 1)$ and $T' = T'_r$. The reference wave (H'_r, T'_r) used for the normalization in Eqs. 3, 4 and 5 may be selected in such a manner that the computed normalized quantities may be interpreted easily.

• 2.4 •

Statistical Calculations

The *statistical calculations* in this report imply the calculations of the time-averaged (mean) value and the extreme (maximum and minimum) values of computed time-series. The statistical calculations are performed over the specified time interval described below.

1. For the case of *regular* waves, the statistical calculations are executed over the last wave period, assuming that the duration of the computation is long enough to reach the periodicity of time-varying quantities. This duration ranges from approximately five wave periods for coastal structures (Kobayashi and Wurjanto 1989c) to about 30 wave periods for beaches (Kobayashi, DeSilva, and Watson 1989).

2. For *irregular* wave computations, the statistical calculations are conducted over most or all of the computation duration. The initial transient waves may be excluded for the

case where the *random* sea state is modeled using a spectral approach (Kobayashi and Wurjanto 1992a). On the other hand, if measured time series are available, no exclusion may be necessary (Kobayashi, Cox, and Wurjanto 1990).

In the subsequent sections, the time-averaged (mean) value will be denoted by an overbar, whereas the maximum and minimum values are indicated by the subscripts $_{max}$ and $_{min}$. It should be kept in mind that these values are associated with the time interval described above.

• 2.5 •

WAVE REFLECTION

The normalized reflected wave train $\eta_r(t)$ at the seaward boundary is computed using Eq. 13. It is also required to find the unknown value of the vector $\mathbf{U}_1^*$ at $x = 0$ and the next time level $t^* = (t + \Delta t)$ which can not be computed using Eq. 17. In the following, the seaward boundary algorithm adopted in RBREAK2 is described first for the cases of IWAVE=1 and 2 where the incident wave profile $\eta_i(t)$ in Eq. 12b is specified as input and then for IWAVE=3 where the total free surface profile $\eta(t)$ in Eq. 12a is specified as input.

2.5.1 SEAWARD BOUNDARY ALGORITHM FOR IWAVE=1 AND 2

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^* for the cases of IWAVE=1 and 2.

$$\beta_1^* = \beta_1 - \frac{\Delta t}{\Delta x}(u_1 - c_1)(\beta_2 - \beta_1) + \Delta t\, \theta_1 \tag{37}$$

where $\beta_1 = (-u_1 + 2c_1)$ and $\beta_2 = (-u_2 + 2c_2)$. The right hand side of Eq. 37 can be computed for the known values of $\mathbf{U}_j$ with $j = 1$ and 2 at the present time t where the spatial nodes are located at $x = (j-1)\Delta x$. The value of η_r^* at the time t^* is calculated using Eq. 13. Eq. 12 with the incident wave profile $\eta_i(t)$ specified as input yields the value of h_1^*, while $u_1^* = [2\sqrt{(h_1^*)} - \beta_1^*]$ 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 the time t^* are obtained.

2.5.2 SEAWARD BOUNDARY ALGORITHM FOR IWAVE=3

For IWAVE=3 where $\eta(t)$ at $x = 0$ in Eq. 12a is specified as input, the values of η_1^* and h_1^* at node 1 and the time t^* are known. The value of u_1^* at node 1 and the time t^* is computed using Eq. 10 for the characteristic variable β advancing seaward from the computation domain.

A simple first-order finite difference approximation of Eq. 10 with $f = 0$ along the straight line, $dx/dt = (u_1^* - c_1^*) < 0$, originating from the point at node 1 and the time $t^* = (t + \Delta t)$ may be expressed as

$$\beta_1^* = \beta_{12} + \Delta t \theta_1 \tag{38}$$

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

$$\delta x = -(u_1^* - c_1^*)\,\Delta t > 0 \tag{39}$$

The numerical stability criterion given by Eq. 27 requires that $\delta x < \Delta x$. As a result, the location of $x = \delta x$ is located between nodes 1 and 2. The linear interpolation between the known values of β_1 and β_2 at the time t yields

$$\beta_{12} = \beta_1 + \frac{\delta x}{\Delta x}(\beta_2 - \beta_1) \tag{40}$$

Using Eqs. 39 and 40, Eq. 38 may be rewritten as

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

Eq. 41 corresponds to Eq. 37 except that $(u_1 - c_1)$ in Eq. 37 is replaced by $(u_1^* - c_1^*)$. Eq. 41 for $\beta_1^* = (2c_1^* - u_1^*)$ is an implicit scheme for u_1^* for the known value of $c_1^* = \sqrt{(h_1^*)}$. Solving Eq. 41 for u_1^* yields

$$u_1^* = \frac{2c_1^* - \beta_1 - \frac{\Delta t}{\Delta x}(\beta_2 - \beta_1)c_1^* - \Delta t \theta_1}{1 - \frac{\Delta t}{\Delta x}(\beta_2 - \beta_1)} \tag{42}$$

If the absolute value of the denominator on the right hand side of Eq. 42 becomes almost zero, this implicit algorithm may not be appropriate. The computation in BEFORR2 will stop if this absolute value becomes less than 0.0001 as explained in Section 4.5. This problem has never been encountered so far partly because the numerical stability criterion expressed as Eq. 29 generally requires a value of $\Delta t/\Delta x$ that is much less than unity.

After u_1^* is computed using Eq. 42, the value of β_1^* is obtained from $\beta_1^* = (2c_1^* - u_1^*)$. The value of η_r^* for the reflected wave profile is then calculated using Eq. 13 at the time t^*. The value of η_i^* for the incident wave profile is obtained from $\eta_i^* = (\eta_1^* - \eta_r^*)$ based on Eq. 12b.

2.5.3 Nonlinear Correction for Gentle Slopes

The nonlinear correction term C_t given by Eq. 14 needs to be estimated to compute $\eta_r(t)$ using Eq. 13. For incident regular waves on gentle slopes, C_t may be estimated by (Kobayashi, DeSilva, and Watson 1989)

$$C_t = \frac{K_s^2}{16 d_t} \quad \text{for gentle slopes} \tag{43}$$

where the assumptions of linear long wave and negligible wave reflection were made in Eq. 14 to derive Eq. 43. Wave reflection may not be negligible for coastal structures. 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. 43 as a rough guideline.

2.5.4 Wave Reflection Coefficients

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

$$r_1 = \frac{1}{K_s}[(\eta_r)_{max} - (\eta_r)_{min}] \tag{44}$$

$$r_2 = \sqrt{\frac{\overline{\eta_r^2}}{\overline{\eta_i^2}}} \quad (45)$$

$$r_3 = \sqrt{\frac{\overline{(\eta_r - \overline{\eta_r})^2}}{\overline{\eta_i^2}}} \quad (46)$$

where the overbar indicates time averaging and the subscripts $_{max}$ and $_{min}$ indicate the maximum and minimum values of $\eta_r(t)$ over the time interval of the statistical calculations (Section 2.4).

K_s represents the normalized height of the representative wave at the seaward boundary. The numerator $[(\eta_r)_{max} - (\eta_r)_{min}]$ in Eq. 44 is the normalized vertical distance between the highest and the lowest surface elevations associated with the reflected wave train. For regular waves, this is the height of the reflected waves. The reflection coefficient r_1 for the case of regular waves is the ratio of the height of the reflected wave to that of the incident wave, For random waves, however, this quantity indicates only the magnitude of the reflected waves and overestimates the degree of wave reflection.

Eqs. 45 and 46 are based on the time-averaged reflected wave energy as compared to the time-averaged incident wave energy, $\overline{\eta_i^2}$. For the case of regular waves, the energy $\overline{\eta_i^2}$ may be estimated using linear wave theory as $\overline{\eta_i^2} = K_s^2/8$. Eq. 46 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. 44 and 45 yield $r_1 = r_2$. Eqs. 45 and 46 are applicable to regular waves as well as irregular waves. The computer program RBREAK2 computes all the three reflection coefficients for the case of regular waves, but provides only r_1 for the case of irregular waves since the time series of the incident and reflected waves, from which r_2 and r_3 can be computed, are available as standard output of the program. Furthermore, these time series allow one to compute the reflection coefficient as a function of the frequency for random waves (*e.g.*, Kobayashi, Cox, and Wurjanto 1990).

• 2.6 •

WAVE RUNUP

For the case of no wave overtopping on a subaerial coastal structure as shown in Fig. 1, identified by the integer IJOB=1, the landward boundary of the numerical model is located at the moving waterline on the slope where the water depth is essentially zero. The kinematic boundary condition requires that the horizontal waterline velocity be the same as the horizontal fluid velocity. In reality, it is difficult to pinpoint the exact location of the moving waterline on the slope. For the computation, the waterline is defined as the location where the normalized instantaneous water depth equals a small value δ, such as $\delta = 10^{-3}$.

The following numerical procedure dealing with the moving waterline located at $h = \delta$ is used to compute the values of $\mathbf{U}_j^*$ at the next time level $t^* = (t + \Delta t)$ 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 present time t where the integer s indicates the wet node next to the moving waterline at the time t such that $h_{s+1} \le \delta < h_s$.

2. Compute h_j^* and m_j^* at the time t^* for the nodes $j = (s-1)$ and $j = s$, using Eq. 17 without the damping term $\mathbf{D}_j$ since the water depth h can be very small at these nodes.

3. Check whether $h_{s-1}^* \le \delta$, which may be encountered during a downrush. This is considered a computation failure since the waterline should not move more than Δx because of the numerical stability criterion of the adopted explicit method given by Eq. 27. Denoted by `IPROB=2`, this failure is one of three identified numerical difficulties, which will be elaborated more in Part IV.

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

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

5. If $h_s^* \le \delta$, set $s^* = (s-1)$ and go to Step 13. The integer s^* indicates the wet node next to the waterline at the next time level 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 go to Step 13.

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

$$m_{s+1}^* = m_{s-1}^* - \frac{\Delta x}{\Delta t}\left(h_s^{**} - h_s\right) \tag{47}$$

$$u_{s+1}^* = u_{s-1}^* - \frac{1}{u_s^*}\left[\frac{\Delta x}{\Delta t}\left(u_s^{**} - u_s\right) + h_{s+1}^* - h_{s-1}^* + 2\Delta x\,\theta_s\right] \tag{48}$$

   The upper limit of the absolute value of $(u_s^*)^{-1}$ in Eq. 48 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 go to Step 13.

10. If $h_{s+1}^* \le h_s^*$ and $h_{s+1}^* \le \delta$, set $s^* = s$ and go to Step 13.

11. If $h_{s+1}^* \le h_s^*$ and $h_{s+1}^* > \delta$, set $s^* = (s+1)$ and go to Step 13.

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 $h^*_{s+1} \leq h^*_s$ where h^*_{s+1} is the adopted value given by $h^*_{s+1} = (2h^*_s - h^*_{s-1})$.

13. After $s*$ is obtained, set $h^*_j = 0$, $u^*_j = 0$, and $m^*_j = 0$ for $j \geq (s^* + 1)$ 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. The normalized runup R, run-down R_d, and setup $\overline{Z_r}$ for given δ'_r are obtained from the computed oscillation of $Z_r(t)$. The computed results such as those presented by Kobayashi, DeSilva, and Watson (1989) indicate that wave runup is fairly 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.

• 2.7 •

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. The case of wave overtopping is identified by IJOB=2, for which the computation domain for the numerical model is limited to the region $0 \leq x \leq x_e$ where $x_e = x'_e/(T'_r\sqrt{gH'_r})$ 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 as the normalized depth 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 the present time 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 the time t, the values of $\mathbf{U}^*_j$ at the next time level $t^* = (t + \Delta t)$ for the nodes $j = (s-1)$ and s are computed as follows

1. Compute $\mathbf{U}^*_{s-1}$ using Eq. 17 with $j = (s-1)$ without the damping term $\mathbf{D}_{s-1}$ since the water depth h can be very small at this node.

2. If $u_{s-1} > c_{s-1}$, where $c_{s-1} = \sqrt{h_{s-1}}$, 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 $f = 0$

$$h_s^* = h_s - \frac{\Delta t}{\Delta x}(m_s - m_{s-1}) \tag{49}$$

$$m_s^* = m_s - \frac{\Delta t}{\Delta x}\left[\left(m_s u_s + \frac{1}{2}h_s^2\right) - \left(m_{s-1}u_{s-1} + \frac{1}{2}h_{s-1}^2\right)\right] - \Delta t\ \theta_s\ h_s \tag{50}$$

   Then, $u_s^* = m_s^*/h_s^*$, and $\mathbf{U}_s^*$ is obtained.

3. If $u_{s-1} \leq c_{s-1}$, 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 the time t^*. Use is made of the following finite difference equation derived from Eq. 9 with $f = 0$

$$\alpha_s^* = \alpha_s - \frac{\Delta t}{\Delta x}(u_s + c_s)(\alpha_s - \alpha_{s-1}) - \Delta t\ \theta_s \tag{51}$$

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

4. If $h_s^* \leq \delta$, wave overtopping is assumed to cease. The integer s^* is set to be equal to $(s-1)$ and the landward boundary computation switches to the runup mode discussed in Section 2.6. The integer s^* indicates the wet node next to the waterline at the time t^*.

5. If $h_s^* > \delta$, wave overtopping continues and $s^* = j_e$.

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

$$\bar{Q} = \frac{\overline{Q'}}{H_r'\sqrt{gH_r'}} = \overline{m} \quad \text{at } x = x_e \tag{52}$$

where the overbar indicates the time averaging of $m(t, x)$ at $x = x_e$. Eq. 52 does not include the volume flux during the interval when $h \leq \delta$ at $x = x_e$ since the values of m at the nodes landward of the computational waterline are set to be zero during the computation. Eq. 52 is applicable for both regular and irregular waves. The time interval over which the time averaging is performed has been described in Section 2.4.

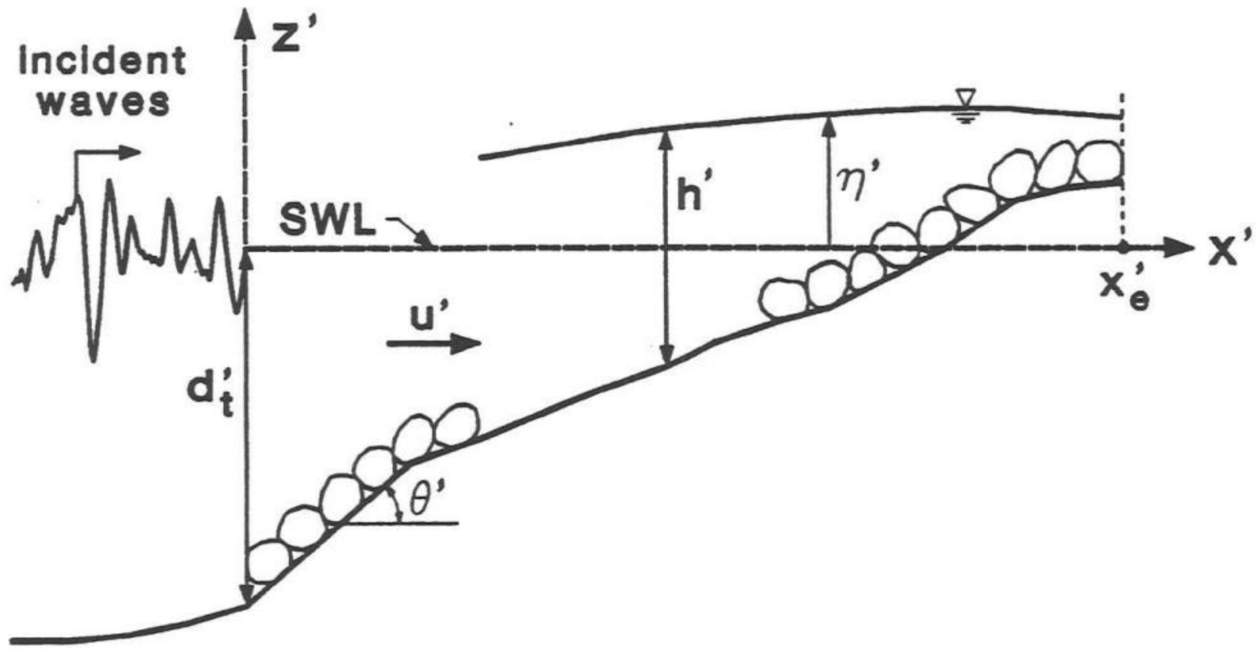

Figure 2: Wave overtopping over a subaerial structure for IJOB=2.

• 2.8 •

WAVE TRANSMISSION

For wave transmission over a submerged breakwater, the landward boundary is always located at $x' = x'_e$ as shown in Fig. 3 where x'_e = x'-coordinate of the landward boundary which can be taken to be any convenient location such as the landward toe of the submerged breakwater. The case of wave transmission is identified by IJOB=3, for which the computation domain for the numerical model is the fixed region $0 \le x \le x_e$ where $x_e = x'_e/(T'_r\sqrt{gH'_r})$ and the dimensionless variables defined in Eqs. 3-5 remain the same. To avoid the appearance of the waterline in the region $0 \le x \le x_e$, the normalized water depth h in the computation domain is taken as $h = \delta$ if the computed value of h becomes less than δ. It is assumed that the transmitted waves propagate landward without being reflected from the shoreline, and the transmitted water flows landward without a return current. If the effects of the shoreline and return current need to be included, it will be required to extend the computation domain to the shoreline in a manner similar to the computations made by Kobayashi, Strzelecki, and Wurjanto (1988) and Kobayashi, DeSilva, and Watson (1989) for the regular wave transformation over a shore-parallel bar and resulting swash oscillation on a beach, as well as Kobayashi and Wurjanto (1992a) for the irregular wave transformation over a barred beach.

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

$$h = d_e + \eta_t(t) \quad \text{at } x = x_e \tag{53}$$

$$\eta_t(t) = \frac{1}{2}\sqrt{d_e}\,\alpha(t) - d_e \quad \text{at } x = x_e \tag{54}$$

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. 54 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. 54.

The following numerical procedure is used to compute the values of $\mathbf{U}_j^*$ at the next time level 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 $\mathbf{U}_{s-1}^*$ using Eq. 17 with $j = (s-1)$ with the damping term $\mathbf{D}_{s-1}$ since the water depth h is large at this node.

2. Compute α_s^* at the time t^* using Eq. 51.

3. Compute η_t^* at the time t^* using Eq. 54 with $\alpha = \alpha_s^*$. Then, $h_s^* = (d_e + \eta_s^*)$ from Eq. 53, while $u_s^* = (\alpha_s^* - 2\sqrt{h_s^*})$. Thus, $m_s^* = h_s^* u_s^*$, and $\mathbf{U}_s^*$ is obtained.

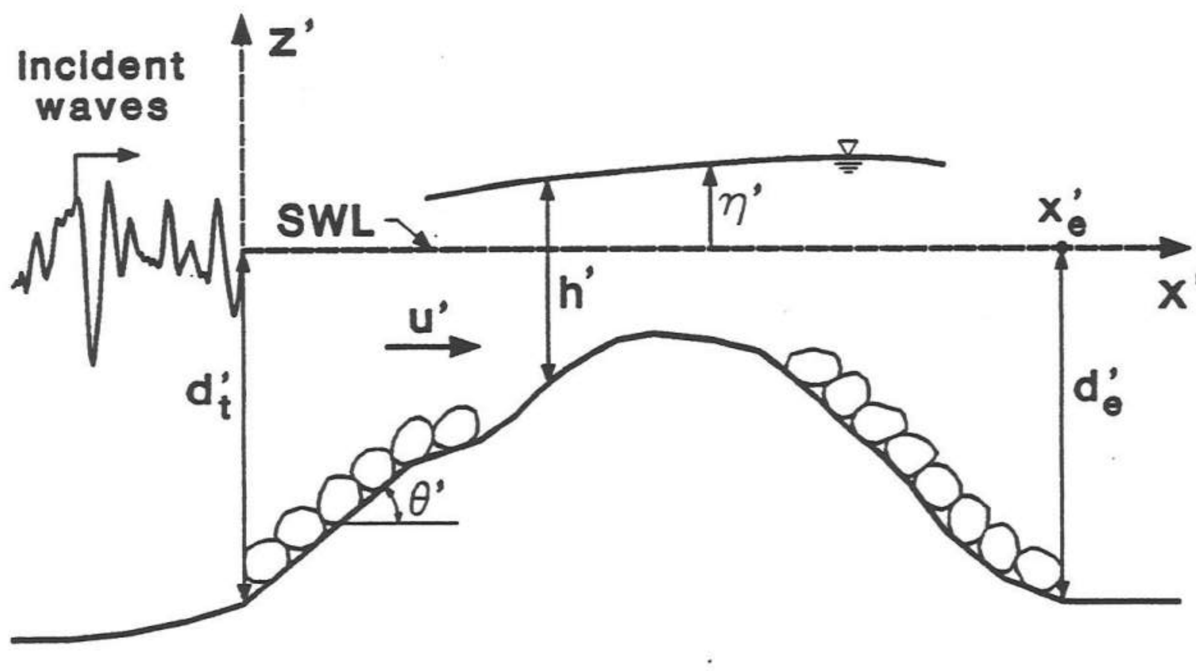

Figure 3: Wave transmission over a submerged structure for IJOB=3.

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

$$T_1 = \frac{1}{K_s}\left[(\eta_t)_{max} - (\eta_t)_{min}\right] \tag{55}$$

$$T_2 = \sqrt{\frac{\overline{\eta_t^2}}{\overline{\eta_i^2}}} \tag{56}$$

$$T_3 = \sqrt{\frac{\overline{(\eta_t - \overline{\eta_t})^2}}{\overline{\eta_i^2}}} \tag{57}$$

Eqs. 55, 56, and 57 correspond to Eqs. 44, 45, and 46, respectively for the reflection coefficients discussed in Section 2.5.4. For the case of regular waves, the transmission coefficient T_1 represents the ratio of the height of the transmitted wave to that of the incident wave. For random waves, T_1 indicates only the magnitude of the transmitted irregular wave. Eqs. 56 and 57, which are applicable to both regular and irregular waves, are based on the time-averaged transmitted wave energy as compared to the time-averaged incident wave energy, $\overline{\eta_i^2}$. The estimation $\overline{\eta_i^2} = K_s^2/8$ based on linear wave theory may be employed for the case of regular waves. Eq. 57 accounts for the difference $\overline{\eta_t}$ between the still water level and the mean water level at $x = x_e$. The computer program RBREAK2 computes T_1, T_2, and T_3 for the case of regular waves, and only T_1 for the case of irregular waves. For the latter, T_2 and T_3 can be computed from the irregular incident and transmitted wave trains stored during the computation.

• 2.9 •

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, 1992a)

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}(E_F) = -D_f - D_B \tag{58}$$

with

$$E = \begin{cases} \frac{1}{2}\left(hu^2 + \eta^2\right) & \text{for } h > \eta \\ \frac{1}{2}\left[hu^2 + \eta^2 - (h-\eta)^2\right] & \text{for } h < \eta \end{cases} \tag{59}$$

$$E_F = uh\left(\frac{1}{2}u^2 + \eta\right) \tag{60}$$

$$D_f = f|u|u^2 \tag{61}$$

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

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 the time $t = 0$ when the incident wave train arrives at $x = 0$ as shown in Figs. 1-3. The first and second parts of Eq. 59 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. 58

$$\overline{D_B} = -\frac{d}{dx}(\overline{E_F}) - \overline{D_f} - \frac{1}{t_{end} - t_{begin}}\left[E|_{t_{end}} - E|_{t_{begin}}\right] \tag{62}$$

where the overbar indicates time averaging over the time period spanning from the normalized time t_{begin} to the time t_{end} for the statistical calculations discussed in Section 2.4. For regular waves, $t_{end} = (t_{begin} + 1)$ and the last term of Eq. 62 is zero because of periodicity. For irregular waves, t_{begin} and t_{end} should be chosen such that the last term of Eq. 62 is negligible so as to obtain the time-averaged values for the stationary sea state.

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. 61 is positive. It should be noted that Eq. 62 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. 62 from the seaward boundary to the landward boundary yields the time-averaged energy equation for the region $0 \leq x \leq x_e$.

$$\overline{E_F}(x=0) - \overline{E_F}(x=x_e) = \int_0^{x_e}\left(\overline{D_f} + \overline{D_B}\right)dx + \int_0^{x_e}\frac{1}{t_{end} - t_{begin}}\left[E|_{t_{end}} - E|_{t_{begin}}\right]dx \tag{63}$$

where the first and second terms on the left hand side of Eq. 63 are the values of $\overline{E_F}$ at $x = 0$ and $x = x_e$, respectively. The last term in Eq. 63 is zero (because of periodicity) for the case of regular waves, and should be negligible for the case of irregular waves. Eq. 63 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 stationary sea state. For the case of wave runup on a slope, Eq. 63 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 $\overline{E_F}$ for regular waves at the seaward boundary where $\eta = (\eta_i + \eta_r)$ at $x = 0$ from Eq. 12b may approximately be given by (Kobayashi and Wurjanto 1989b, 1989c)

$$\overline{E} \simeq \overline{\eta_i^2} + \overline{(\eta_r - \overline{\eta_r})^2} \quad \text{at } x = 0 \tag{64}$$

$$\overline{E_F} \simeq \sqrt{d_t}\left[\overline{\eta_i^2} - \overline{(\eta_r - \overline{\eta_r})^2}\right] \quad \text{at } x = 0 \tag{65}$$

where $\sqrt{d_t}$ is the normalized group velocity at $x = 0$ based on linear long wave theory. The reflection coefficient r_3 given by Eq. 46 including the effect of $\overline{\eta_r}$ is based on Eqs. 64 and 65. The reflection coefficient r_2 given by Eq. 45 corresponds to Eqs. 64 and 65 with $\overline{\eta_r} = 0$.

For the case of regular 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 (Kobayashi and Wurjanto 1989b, 1989c).

$$\overline{E} \simeq \overline{(\eta_t - \overline{\eta_t})^2} \quad \text{at } x = x_e \tag{66}$$

$$\overline{E_F} \simeq \sqrt{d_e}\ \overline{(\eta_t - \overline{\eta_t})^2} \quad \text{at } x = x_e \tag{67}$$

where $\sqrt{d_e}$ is the normalized group velocity at $x = x_e$ based on linear long wave theory. The transmission coefficient T_3 given by Eq. 57 for the case of $d_e = d_t$ is based on Eqs. 66 and 67, whereas the transmission coefficient T_2 given by Eq. 56 does not include the wave setup $\overline{\eta_t}$ at $x = x_e$.

• 2.10 •
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

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -\frac{\partial h}{\partial x} - \theta - \frac{f|u|u}{h} \tag{68}$$

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

$$|N_s + E_1| + E_2 N_s \leq E_3 \tag{69}$$

which is applicable for the case of $u \neq 0$. The stability number N_s in Eq. 69 is defined as

$$N_s = \frac{H'_r}{[W'/(\rho s_g)]^{1/3}(s_g - 1)} \tag{70}$$

where

H'_r = reference wave height used for the normalization of the governing equations
s_g = specific gravity of the armor unit whose unit mass is given by ρs_g
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. 70 for given H'_r. E_1, E_2, and E_3 in Eq. 69 are defined as

$$E_1 = \frac{2C_3^{2/3}}{C_2 C_D |u| u}\left[\frac{C_M}{(s_g - 1)\sigma}\frac{du}{dt} - sin\theta'\right] \tag{71}$$

$$E_2 = \frac{C_L\, tan\phi}{C_D} \tag{72}$$

$$E_3 = \frac{2C_3^{2/3}}{C_2 C_D u^2} cos\theta'\, tan\phi \tag{73}$$

where

C_D = drag coefficient involved in the hydrodynamic drag force.
C_L = lift coefficient involved in the hydrodynamic lift force.
C_M = inertia coefficient involved in the hydrodynamic inertia force.
C_2 = area coefficient of the armor unit.
C_3 = volume coefficient of the armor unit.
ϕ = frictional angle of armor unit.

Eq. 69 can be solved in terms of N_s

$$N_s \leq N_R(t,x) = \begin{cases} (E_1 + E_3)/(E_2 - 1) & \text{if } E_1 < 0,\ E_2 > 1, \text{ and } E_3 < (-E_1 E_2) \\ (E_3 - E_1)/(E_2 + 1) & \text{otherwise} \end{cases} \tag{74}$$

where N_R = dimensionless function expressing the degree of the armor unit stability as a function of t and x. For the computation, Eq. 74 is used if $|u| \geq 10^{-3}$ and N_R is set to be $N_R = 1000$ if $|u| < 10^{-3}$.

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

$$\left|\frac{C_M}{(s_g - 1)\sigma}\frac{du}{dt} - sin\theta'\right| \leq cos\theta'\, tan\phi \tag{75}$$

The condition given by Eq. 75 is satisfied if the normalized fluid acceleration remain within the following lower and upper bounds

$$\sigma\, a_{min} \le \frac{du}{dt} \le \sigma\, a_{max} \tag{76}$$

$$a_{min} \ge -\frac{s_g - 1}{C_M}\,\frac{sin(\phi - \theta')}{cos\phi} \quad ; \quad a_{max} \le \frac{s_g - 1}{C_M}\,\frac{sin(\phi + \theta')}{cos\phi} \tag{77}$$

In terms of the dimensional variables, Eq. 76 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. 77 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 regular 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. It should be noted that the hydraulic stability of armor units is computed in the entire domain under the wave action even if only a portion of the domain is protected with armor units. This reduces the required input to RBREAK2. Furthermore, the computed armor stability in the entire domain may be useful in determining the necessary extent of armor protection.

• 2.11 •
MOVEMENT OF ARMOR UNITS

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

The normalized forces acting on an armor unit, which is assumed to be completely submerged, are separated into

$$F_D = \frac{C_2 C_D}{2C_3 d}\,\sigma\,|u - u_a|\,(u - u_a) \tag{78}$$

$$F_L = \frac{C_2 C_L}{2C_3 d}\,\sigma\,(u - u_a)^2 \tag{79}$$

$$F_I = C_M\,\frac{du}{dt} \tag{80}$$

$$W_c = \sigma\,(s_g - 1)\,cos\theta' \tag{81}$$

$$W_s = \sigma\,(s_g - 1)\,sin\theta' \tag{82}$$

with

$$d = \frac{d'}{H'_r} = \frac{1}{H'_r}\left(\frac{W'}{C_3 \rho s_g}\right)^{1/3} \quad ; \quad u_a = \frac{u'_a}{\sqrt{gH'_r}} \tag{83}$$

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 characteristic length of the armor unit
u_a = normalized velocity of the armor unit along the slope

The prime in Eq. 83 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. 78–82 are based on the normalization by $(gW'/\sigma s_g)$. It is noted that the condition given by Eq. 76 is *not* imposed on the value of du/dt in Eq. 80 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

$$|F_D + F_I - W_s| > F_R \tag{84}$$

with

$$F_R = (W_c - F_L)\, tan\phi \geq 0 \tag{85}$$

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. 84 and 85, F_D and F_L are given by Eqs. 78 and 79 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

$$(s_g + C_m)\frac{du_a}{dt} = F_D + F_I - W_c - J\, F_R \tag{86}$$

with

$$J = \frac{u_a}{|u_a|} \tag{87}$$

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

$$X_a(t) = \frac{X'_a}{d'} = \frac{\sigma}{d}\int_{t_0}^{t} u_a\, dt \tag{88a}$$

$$X_{aa}(t) = \frac{X'_a}{T'_r\sqrt{gH'_r}} = \int_{t_0}^{t} u_a\, dt \tag{88b}$$

where t_0 = normalized time when the armor unit starts moving. Eq. 88a is used to estimate the degree of the armor movement relative its characteristic length d', whereas Eq. 88b is used to find the x-coordinate of the moving unit since the values of u and du/dt in Eqs. 78-80 should be those at the instantaneous location of the unit.

For the computation of the movement of individual armor units, the grid points used for the computation of the flow fields are used to specify the locations of the units before the armor movement computation. The movement of the armor unit located at the node j starts when the condition given by Eq. 84 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. 86 is used to find the normalized velocity u_a of the identified unit whose instantaneous location is computed using Eq. 88b. The values of u and du/dt in Eqs. 78-80 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. 84 is not satisfied. The stopped unit resumes its movement when the condition given by Eq. 84 is satisfied. The temporal variation of X_a defined by Eq. 88a is also computed for the armor unit identified by its initial location on the slope. It should be noted that the movement of armor units is computed at all the nodes under the wave action even if only a portion of the computation is protected with armor units. The computed armor movement in the entire domain under the wave action may be useful in determining the necessary extent of armor protection and the appropriateness of the selected armor size. Alternatively, the armor movement analysis in this section could be modified to predict the movement of individual armor units whose initial locations are specified as input to RBREAK2. However, the specification of the armor unit locations would be difficult before the actual placement of armor units.

PART III
COMPUTER PROGRAM RBREAK2

• 3.1 •
INTRODUCTION

The computer program RBREAK2 attached in Appendix B consists of the main program, 49 subroutines, and one function. Full double precision mode is used throughout the program to gain maximum numerical accuracy. The program has been tested on IBM 3090 running under VM/XA and on Sun 4/280 operating under UNIX-based SunOS. Written in standard FORTRAN-77, RBREAK2 should run on other machines.

The term ***hydrodynamic computation*** in this report refers to the time-marching computation as primarily explained in Section 2.2 with the seaward boundary condition described in Sections 2.3 and 2.5 and the landward boundary conditions in Sections 2.6, 2.7, and 2.8. The hydrodynamic computation constitutes the heart of the computer program RBREAK2. Numerical problems, which determine the success or failure of an RBREAK2 computation, arise mostly from the hydrodynamic computation.

The computer programs RBREAK and RBREAK2 are extensions of its predecessor, IBREAK (Kobayashi and Wurjanto 1989d), which is based on essentially the same theory as described in Part II. The extension is necessary to overcome numerical difficulties which cause abortion of the hydrodynamic computation for irregular waves. These are the same numerical difficulties encountered in the hydrodynamic computation for regular or transient waves using IBREAK. In IBREAK, a single set of the computation parameters such as the time step size is used throughout the computation duration. If a failure occurs, another attempt is carried out using a different set of the computation parameters. This trial and error approach is tolerable for regular or transient waves since the computation duration is typically less than 10 wave periods and very short. This approach becomes very time-consuming for random waves since the required computation duration is long to simulate a representative random wave train which may consist of 200 individual waves.

Potential numerical problems in the computer program RBREAK2 have been identified and remedial measures have been devised. Numerical problems can be minimized by adjusting the computation parameters during the hydrodynamic computation. The computation duration is divided into *computation units*, and each unit is treated separately. The computation for each unit is considered successful when a set of the computation parameters with no numerical problem is found. A number of trial runs may be required before this set is found. Since RBREAK2 already contains many options and is too long to make major modifications, a separate program called BEFORR2 is developed to find the set of the computation parameters for the successful computation of each unit. This program is the forerunner of RBREAK2, that is, BEFORR2 should be executed before RBREAK2. The use of the computation parameters for each unit produced by BEFORR2 will result in the successful hydrodynamic computation of RBREAK2.

The computer program BEFORR2 will be discussed in Part IV. In the following sections, the inner working of the computer program RBREAK2 will be discussed with the assumption that the computation parameters for each computation unit have been supplied by the preceding BEFORR2 computation. If the preceding BEFORR2 computation is successful, the subsequent hydrodynamic computation by RBREAK2 should not fail. If a failure occurs, it must be either because input data files have been mixed up or because the non-hydrodynamic parts of RBREAK2 such as armor movement have encountered numerical difficulties.

The duo BEFORR2-RBREAK2 can do everything that IBREAK can do but does it more efficiently because of the self-adjusting capability for the successful numerical computation embedded in BEFORR2-RBREAK2. RBREAK2 is primarily intended for irregular wave computations since the self-adjusting capability is crucial for the computation of long duration. The subsequent sections are written with irregular waves in mind.

• 3.2 •

INPUT DATA FILES FOR RBREAK2

To execute the computer program RBREAK2 for the case of irregular waves, three data files are needed.

1. *Primary input data file* used in the preceeding BEFORR2 computation. This file contains all the variables and parameters needed to specify the case being investigated, except the input wave train, which is prescribed in the second input data file.

2. File containing the *input wave profile* used in the preceeding BEFORR2 computation as explained in Sections 2.3.2 and 2.3.3.

3. File `BINPUT` created by the preceding BEFORR2 computation for the successful hydrodynamic computation of each computation unit.

The first two input data files are prepared by a user, and should have been used in the preceding BEFORR2 computation as well. The third is produced by the preceding BEFORR2 computation. The user has the freedom for selecting the names of the first and second input data files which are to be read by RBREAK2 as the variables `FINP1` and `FINP2`, respectively. The only limitation imposed by RBREAK2 is that the name should consist of no more than ten characters. The operating system under which RBREAK2 is running may dictate a certain convention regarding file naming.

The user enters the name of the primary input data file interactively at the beginning of RBREAK2 computation. The name of the file containing the input wave train is specified in the primary input data file. The preparation of the primary input data file may be best explained by use of the example given in Section 5.3.2. In addition, the order of the contents of the primary input data file will be presented in Section 3.6. The generation of the file BINPUT will be elaborated in Part IV. In the following, the contents of the second input data file are explained.

The input wave train is read by Subr. 03 INPUT2, where the subroutines are explained in Section 3.4, as follows:

```
      READ (12,9000) NDATA
      READ (12,8000) (ETA(I),I=1,NDATA)
 8000 FORMAT (5D15.6)
 9000 FORMAT (I8)
```

where

NDATA = number of data points
ETA = array of size NDATA prescribing the normalized free surface elevation at the seaward boundary

It is required that $1 \leq$ NDATA $\leq$ N2, where the parameter N2 is to be explained in the COMMON /DIMENS/ in Section 3.5. The data points of ETA are equally spaced in time and NDATA is given by

$$\text{NDATA} = \frac{\text{TMAX}}{\delta t} + 1 \tag{89}$$

where

TMAX = normalized computation duration specified in the primary input data file by the user
δt = normalized time increment of the data points

It is noted that the values at the very beginning ($t = 0$) and at the very end ($t =$TMAX) of the input wave train need to be specified even if they are the same. In other words, The NDATA points cover the time span $0 \leq t \leq$ TMAX. This caution is applicable to the case where the input wave train is generated for given frequency spectrum using random phases. The wave train generated in this way is periodic over TMAX, in which case the first and NDATA-th data points are identical.

The time increment of the input wave train, δt, is normally much larger than the finite difference time step Δt for the stable numerical computation discussed in Part II. The time step Δt, which may vary from one computation unit to another, has been determined by the preceding BEFORR2 computation. The computer program RBREAK2 performs a simple linear interpolation of the input wave train to get the appropriate value at each time level during the time-marching computation. The interpolation is carried out in Subr. 19 SEABC.

• 3.3 •

MAIN PROGRAM RBREAK2

The main program lists all the important parameters and variables in the **COMMON** blocks. These parameters and variables are described in Section 3.5. The main program coordinates tasks which are actually executed by subroutines. The tasks can be categorized into five groups.

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

3.3.1 READING INPUT DATA

The first variable read by RBREAK2 is **MREP**, which is specified interactively by the user. **MREP** determines the interval between two consecutive progress messages RBREAK2 displays on a terminal screen. The progress message is displayed every **MREP** wave periods where the period herein is the reference wave period. For example, if **MREP**=4, the following message (without the horizontal lines) will appear on the screen when the computation has just finished 56 wave periods.

```
Finished            56 Wave Periods
```

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

The next input is the name of the primary input data file, which is also entered interactively. The three input data files explained in Section 3.2 are then read from the prepared files. Input and output files are opened by calling Subr. 01 **OPENIO**. The contents of the first and the second input data files are read by Subr. 02 **INPUT1** with **MODE**=1, where **MODE** is used to identify different parts of subroutines, and Subr. 03 **INPUT2**, respectively, at the beginning of the computation before the time-marching computation begins. A list of the **READ** statements corresponding to the primary input data file will be presented in Section 3.6. The computation parameters in the file **BINPUT** are read sequentially (unit by unit) by Subr. 02 **INPUT1** with **MODE**=9 during the time-marching computation.

3.3.2 CHECKING FORTRAN PARAMETERs IN THE SUBROUTINES

In the computer program RBREAK2, the dimension of an array is specified using an integer which is independently declared as a **PARAMETER** by each program unit where the term *program unit* is used herein to represent the main program, subroutines, and functions. The almost all of RBREAK2's variables, which are mostly arrays, are passed between program units using **COMMON** blocks. Only few variables are passed as arguments of subroutines and functions. This arrangement demands that the dimensions of arrays in the **COMMON** blocks throughout RBREAK2 remain the same. Subr. 48 **CKSUBR** detects possible mismatches in the array dimensions among the program units. Using the **PARAMETER**s specified in the main program as reference, Subr. 48 **CKSUBR** calls each applicable subroutine with **MODE**=0 or **ICALL**=0 to check if the required agreement is satisfied. In Subr. 10 **CPARAM**, **MODE**=0 is also used to ensure that RBREAK2 reads the correct file **BINPUT** produced by the preceding BEFORR2 computation. In this effort, four case-specific parameter values, called *case-signatures* herein, are passed from BEFORR2 to RBREAK2 through the file **BINPUT** to assure that these values are equal to their RBREAK2 counterparts. A mismatch means that a wrong file **BINPUT** is being used. The checking of the case-signatures is carried out in Subr. 02 **INPUT1** and 03 **INPUT2** in addition to Subr. 10 **CPARAM**.

3.3.3 GROUNDWORK

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

- Computation of the bottom geometry using Subr. 04 **BOTTOM**.
- Computation of the wave parameters using Subr. 05 **WPARAM**, and the armor stability parameters using Subr. 40 **SPARAM** if applicable.
- Assignment of the initial values using Subr. 26 **INIT**.
- For regular waves, compute the initial regular wave profile based on the minimum allowable value of **NONE** using Subr. 06 **REGWAV** where **NONE** is the number of time steps in one wave period.

3.3.4 TIME-MARCHING COMPUTATION

The time-marching computation is executed for **MWAVE** computation units in sequence. Each computation unit is one wave period long except the last unit, which may be less than one wave period long. One wave period in the time-marching computation is unity on the basis of the normalization by the reference wave period. The time step Δt is constant within each computation unit, but may vary from one computation unit to another. Δt is the inverse of **NONE**.

$$\Delta t = \frac{1}{\texttt{NONE}} \tag{90}$$

The value of **NONE** for each computation unit is supplied by the file **BINPUT**. RBREAK2 uses the integer **M** to indicate the computation unit and the integer **N** to denote the present time level within each computation unit. **M** is varied from 1 to **MWAVE**, whereas **N** is changed from 1 to **NEND**, where **NEND** is the final time level in each computation unit. **NEND** is equal to **NONE** for the computation units with **M = 1,2,...,(MWAVE-1)**, and may be less than **NONE** for the final computation unit.

The *present* time level t discussed in Part II should now be denoted as the time level $t = [(\texttt{M-1}) + (\texttt{N-1})\Delta t]$. Similarly, the *next* time level $t^* = (t + \Delta t)$ in Part II is now the time $t^* = [(\texttt{M-1}) + \texttt{N}\Delta t]$.

During the time-marching computation, the unknown quantities at the time $t^* = [(\mathtt{M-1}) + \mathtt{N}\Delta t]$ are computed from the known quantities at the time $t = [(\mathtt{M-1}) + (\mathtt{N-1})\Delta t]$.

At the beginning of each computation unit, the following is performed.

- Initialize the quantities used to calculate time-averaged values (Subr. 28 **INSTAT**)
- Read the computation parameters **NONE**, **NEND**, and **DELTA** (Subr. 02 **INPUT1** with **MODE**=9). **NONE** and **NEND** have been described above. **DELTA** is the normalized depth at the computational waterline.
- If the value of **NONE** of the present computation unit differs from that of the previous unit:
  - Calculate the computation parameters which are dependent on **NONE** (Subr. 10 **CPARAM**).
  - For the case of regular waves: Compute the regular wave profile (Subr. 06 **REGWAV**).

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

1. Estimate h_{s+1}, u_{s+1}, and m_{s+1} at the time $t = [(\mathtt{M-1}) + (\mathtt{N-1})\Delta t]$ by linear extrapolation for the case of wave runup where the integer s indicates the wet node next to the moving waterline at $t = [(\mathtt{M-1}) + (\mathtt{N-1})\Delta t]$ (Subr. 12 **EXTRAP**).
2. Retain the values of the quantities at the time $t = [(\mathtt{M-1}) + (\mathtt{N-1})\Delta t]$ which are required for the seaward and landward boundary computations (Subr. 13 **RETAIN**).
3. Compute $c_j = \sqrt{h_j}$ for $j = 1, 2, ..., s$ used in the characteristic equations given by Eqs. 9 and 10 with Eq. 11 (Subr. 14 **CRITV**).
4. Compute the unknown quantities at the time $t^* = [(\mathtt{M-1}) + \mathtt{N}\Delta t]$ (Subr. 15 **MARCH**, 16 **LANDBC**, and 19 **SEABC**).
5. Check the simplified condition of $|u| < (\Delta x / \Delta t)$ that should be satisfied if the numerical stability criterion given by Eq. 27 is satisfied (Subr. 11 **NUMSTA**).
6. Compute the quantities related to wave energy balance, if applicable (Subr. 30 **ENERGY**).
7. 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 (Subr. 41 **VECMAT** and 35 **STAT1**).
8. Compute the hydraulic stability or movement of armor units, if applicable (Subr. 31 **STABNO** or 32 **MOVE**).

3.3.5 FINISHING

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

- Completion of the statistical calculations (Subr. 37 **STATC**).
- Completion of the computation of the time-averaged energy quantities (Subr. 39 **ENERGC**).
- Writing the results (Subr. 45 **DOC3**). It should be noted that some documentation is written before the time-marching computation (Subr. 43 **DOC1**) and some results are written during the time-marching computation (Subr. 44 **DOC2**).

• 3.4 •

SUBROUTINES AND FUNCTION

The 49 subroutines and one function arranged in numerical order in the computer program RBREAK2 are listed in Table 1. The page numbers for the subroutines and function listed in Table 1 correspond to the page numbers in the RBREAK2 listing presented in Appendix B. Interdependence among the program units are mapped out in Table 2. Each of the subroutines and function are explained concisely in the following. Explanation is given in the format: Number - **NAME** - Description, where the Number refers to the numerical order in the computer program RBREAK2.

01 **OPENIO** opens the input and output files with the assistance from Subr. 50 **OPENF**.

02 **INPUT1** reads information from the primary input data file and the file **BINPUT**. It also performs checks on (1) parameters related to the dimensions of arrays and (2) two of the case-signatures, **MWAVE** and **FINP2**, as explained in Section 4.6. A rigorous scrutiny on the most of input data has been conducted in the preceding BEFORR2 computation.

03 **INPUT2** reads the input wave train, that is, the wave profile at the seaward boundary prescribed by a user as explained in Sections 2.3.2 and 2.3.3.

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

05 **WPARAM** calculates the dimensionless wave parameters used in the other subroutines.

06 **REGWAV** computes the incident wave profile at the seaward boundary using Eq. 30 or 33 for the case of regular waves.

07 **FINDM** computes the value of the parameter m which satisfies Eq. 36.

08 **CEL** computes the values of the complete elliptic integrals K and E used in Eqs. 33-36 for given m. **CEL** is the only function in the computer program RBREAK2.

09 **SNCNDN** computes the Jacobian elliptic function cn used in Eq. 33.

10 **CPARAM** computes the parameters which are dependent on **NONE** for each computation unit. This subroutine also checks one of the case-signatures, **NONM**, when **MODE**=0.

11 **NUMSTA** checks if the numerical stability criterion is violated (**IPROB**=3).

12 **EXTRAP** estimates the hydrodynamic quantities at the node immediately landward of the present waterline node by extrapolation.

13 **RETAIN** retains the quantities at the present time level used for the landward and seaward boundary computations.

14 **CRITV** calculates the critical velocities used in the characteristic variables and checks for negative water depth (**IPROB**=1).

Table 1: List of 49 subroutines and one function in computer program RBREAK2.

No.	Subroutine (S) or Function (F)		Page No. in RBREAK2	No.	Subroutine (S) or Function (F)		Page No. in RBREAK2
01	S	OPENIO	B-9 - B-12	26	S	INIT	B-51 - B-54
02	S	INPUT1	B-12 - B-20	27	S	INITH	B-54
03	S	INPUT2	B-20 - B-21	28	S	INSTAT	B-54 - B-55
04	S	BOTTOM	B-21 - B-24	29	S	SVSTAT	B-55 - B-57
05	S	WPARAM	B-24 - B-26	30	S	ENERGY	B-57 - B-58
06	S	REGWAV	B-26 - B-28	31	S	STABNO	B-58 - B-61
07	S	FINDM	B-28 - B-30	32	S	MOVE	B-61 - B-63
08	F	CEL	B-30 - B-31	33	S	FORCES	B-63 - B-64
09	S	SNCNDN	B-31 - B-32	34	S	ACCEL	B-64
10	S	CPARAM	B-32 - B-34	35	S	STAT1	B-64 - B-65
11	S	NUMSTA	B-34	36	S	STAT2	B-65
12	S	EXTRAP	B-34 - B-35	37	S	STATC	B-65 - B-67
13	S	RETAIN	B-35 - B-36	38	S	COEF	B-67 - B-68
14	S	CRITV	B-36 - B-37	39	S	ENERGC	B-68 - B-70
15	S	MARCH	B-37 - B-38	40	S	SPARAM	B-70 - B-71
16	S	LANDBC	B-38 - B-42	41	S	VECMAT	B-71
17	S	RUNUP	B-42 - B-44	42	S	DERIV	B-71 - B-72
18	S	OVERT	B-44	43	S	DOC1	B-72 - B-75
19	S	SEABC	B-44 - B-47	44	S	DOC2	B-75 - B-79
20	S	MATAFG	B-47	45	S	DOC3	B-79 - B-84
21	S	MATGJR	B-47 - B-48	46	S	CKFPAR	B-84
22	S	MATS	B-48 - B-49	47	S	CKVAL	B-84 - B-85
23	S	MATD	B-49 - B-50	48	S	CKSUBR	B-85 - B-87
24	S	MATU	B-50	49	S	STOPP	B-87
25	S	NSI	B-50 - B-51	50	S	OPENF	B-87

Table 2: Interdependence among the program units of computer program RBREAK2.

No.	PROGRAM UNIT	CALLED FROM	MAKES CALL(S) TO
00	MAIN	-	01, 02, 03, 04, 05, 06, 10, 11, 12, 13, 14, 15, 16, 19, 26, 28, 29, 30, 31, 32, 35, 37, 39, 40, 41, 43, 44, 45, 48, 49
01	OPENIO	00	50
02	INPUT1	00, 48	46, 47, 49
03	INPUT2	00, 48	46, 47, 49
04	BOTTOM	00, 48	46, 47, 49
05	WPARAM	00, 48	46
06	REGWAV	00, 48	07, 08, 09, 46
07	FINDM	06	08, 46
08	CEL	06, 07	-
09	SNCNDN	06	49
10	CPARAM	00	46, 49
11	NUMSTA	00, 48	25, 46, 49
12	EXTRAP	00, 48	46
13	RETAIN	00, 48	46
14	CRITV	00, 48	25, 46, 49
15	MARCH	00, 48	20, 21, 22, 23, 24, 25, 46, 49
16	LANDBC	00, 48	17, 18, 35, 36, 46
17	RUNUP	16, 48	20, 21, 22, 46
18	OVERT	16, 48	46
19	SEABC	00, 48	36, 46
20	MATAFG	15, 17, 48	46
21	MATGJR	15, 17, 48	46
22	MATS	15, 17, 48	46
23	MATD	15, 48	46
24	MATU	15, 48	46
25	NSI	11, 14, 15	-
26	INIT	00, 48	27, 46
27	INITH	26	-
28	INSTAT	00, 48	46
29	SVSTAT	00, 48	46
30	ENERGY	00, 48	46

Continued on the next page.

Table 2 continued from the previous page.

No.	Program Unit	Called from	Makes Call(s) to
31	STABNO	00, 48	34, 46
32	MOVE	00, 48	33, 34, 46
33	FORCES	32, 48	46
34	ACCEL	31, 32, 48	41, 42, 46
35	STAT1	00, 16	-
36	STAT2	16, 19	-
37	STATC	00, 48	38, 46
38	COEF	37	-
39	ENERGC	00, 48	41, 42, 46
40	SPARAM	00, 48	46
41	VECMAT	00, 34, 39	-
42	DERIV	34, 39	-
43	DOC1	00, 48	46
44	DOC2	00, 48	46
45	DOC3	00, 48	46
46	CKFPAR	All units except for 00, 01, 08, 09, 25, 27, 35 36, 38, 41, 42, 46, 47, 48 49, 50	-
47	CKVAL	02, 03, 04	-
48	CKSUBR	00	All units except for 00, 01, 07, 08, 09, 10, 25, 27, 35, 36, 38, 41, 42, 46, 47, 48, 49, 50
49	STOPP	00, 02, 03, 04, 09, 10, 11, 14, 15	-
50	OPENF	01	-

15 `MARCH` performs the time-marching computation on the basis of Eq. 17.

16 `LANDBC` 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.

17 `RUNUP` computes the waterline movement on the slope of a subaerial structure on the basis of the procedure discussed in Section 2.6.

18 `OVERT` computes the overtopping flow at the landward edge of the crest of a subaerial structure on the basis of the procedure discussed in Section 2.7.

19 `SEABC` computes the flow at the seaward boundary using Eq. 37 or 42 and the reflected wave train $\eta_r(t)$ using Eq. 13.

20 `MATAFG` computes the values of the elements of the first row of the matrix **A** defined in Eq. 19, the elements of the vector **F** in Eq. 16, and the first element of the vector **G** in Eq. 16.

21 `MATGJR` computes the values of the elements of the vector **g** defined by Eq. 18.

22 `MATS` computes the value of the first element of the vector **S** defined by Eq. 20.

23 `MATD` computes the values of the elements of the vector **D** defined by Eq. 22.

24 `MATU` computes the values of the elements of the vector $\mathbf{U}^*$ using Eq. 17.

25 `NSI` computes the numerical stability indicator Ω defined by Eq. 28.

26 `INIT` 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.

27 `INITH` facilitates the assignment of the initial values in Subr. 26 `INIT`.

28 `INSTAT` initializes statistical quantities representing mean values at the beginning of a computation unit.

29 `SVSTAT` saves statistical quantities related to mean values at the end of a computation unit.

30 `ENERGY` computes the values of E, E_F, and D_f defined by Eqs. 59, 60, and 61, respectively, in relation to the normalized equation of wave energy. This subroutine also saves the values of E at the normalized time t_{begin} and t_{end}, which are introduced in Eq. 62.

31 `STABNO` computes the armor stability function $N_R(t,x)$ using Eqs. 71–74 and the local stability number $N_{sx}(x)$ defined as the minimum value of $N_R(t,x)$ at a given location.

32 `MOVE` computes the displacement of armor units using Eqs. 84–88.

33 `FORCES` computes the normalized forces given by Eqs. 78–82.

34 `ACCEL` computes the value of du/dt using Eq. 68.

35 `STAT1` is used to calculate the sum, maximum, and minimum values of quantities varying with time.

36 STAT2 is used to calculate the sum, maximum, and minimum values of the surface elevations at the seaward and landward boundaries.

37 STATC completes the statistical calculations after the time-marching computation.

38 COEF computes the wave reflection coefficients given by Eqs. 44–46 as well as the wave transmission coefficients given by Eqs. 55–57 for the case of a submerged structure.

39 ENERGC completes the computation of energy quantities after the time-marching computation.

40 SPARAM calculates the parameters needed for computation of armor stability or movement.

41 VECMAT creates a vector from a specified row of a given matrix

42 DERIV computes the first derivative of a function using a finite difference method.

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

44 DOC2 stores some of the computed results at designated time levels during the time-marching computation.

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

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

47 CKVAL checks whether some of the values selected by a user are within the ranges available in the present form of RBREAK2.

48 CKSUBR calls subroutines which use any of the PARAMETERs N1, N2, N3, N4, and N5 to check the values of the PARAMETERs being used. The check is actually performed by Subr. 46 CKFPAR. Calls made from Subr. 48 CKSUBR are identified with MODE=1 or ICALL=1.

49 STOPP executes a programmed stop if some of the input requirements for RBREAK2 are not satisfied.

50 OPENF opens specified files.

• 3.5 •

PARAMETERS AND VARIABLES IN COMMON BLOCKS

The parameters and variables included in the COMMON blocks in the main program RBREAK2 are explained in the following so that a user may be able to comprehend the computer program RBREAK2 and modify it if required.

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

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

N2R = N2 = maximum allowed value of (NONE+1) for a regular wave computation, or maximum number of data points allowed in the input wave train. NONE is described in COMMON /CPAR2/ below.

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

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

N5R = N5 = maximum number of spatial nodes where the time series of certain computed quantities are stored as well as maximum number of specified time levels at which special storing is requested.

The present setting for these integers is N1=800, N2=24000, N3=3, N4=40, N5=40. These integers can be changed as long as the changes are made throughout the program.

/CONSTA/ contains constants.

PI = π = 3.141592...

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

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

IJOB indicates the type of the landward boundary condition. IJOB=1 for wave runup on the seaward slope of a subaerial structure, IJOB=2 for wave overtopping over a subaerial structure, IJOB=3 for wave transmission over a submerged structure.

ISTAB indicates the type of the armor analysis. ISTAB=0 if no computation of armor stability or movement is desired, ISTAB=1 for the computation of armor stability, ISTAB=2 for the computation of armor movement.

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

IBOT indicates the type of input data for the structure geometry divided into linear segments of different slopes and roughness. IBOT=1 for the width and slope of linear segments, IBOT=2 for the locations of the seaward end points of linear segments. See Fig. 4 for clarification.

INONCT indicates whether the nonlinear correction term C_t is included or not in Eq. 13 in calculating the reflected wave profile $\eta_r(t)$. INONCT=0 for $C_t = 0$, INONCT=1 for C_t given by Eq. 43.

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

IWAVE indicates the type of the wave profile specified at the seaward boundary. IWAVE=1 for the incident wave profile $\eta_i(t)$ computed using Subr. 06 REGWAV, IWAVE=2 for the incident irregular wave profile $\eta_i(t)$ read by Subr. 03 INPUT2, IWAVE=3 for the measured wave profile $[\eta_i(t)+\eta_r(t)]$ read by Subr. 03 INPUT2. It is noted that IWAVE=3 corresponds to the free surface oscillation measured at the seaward boundary in the presence of a coastal structure or beach.

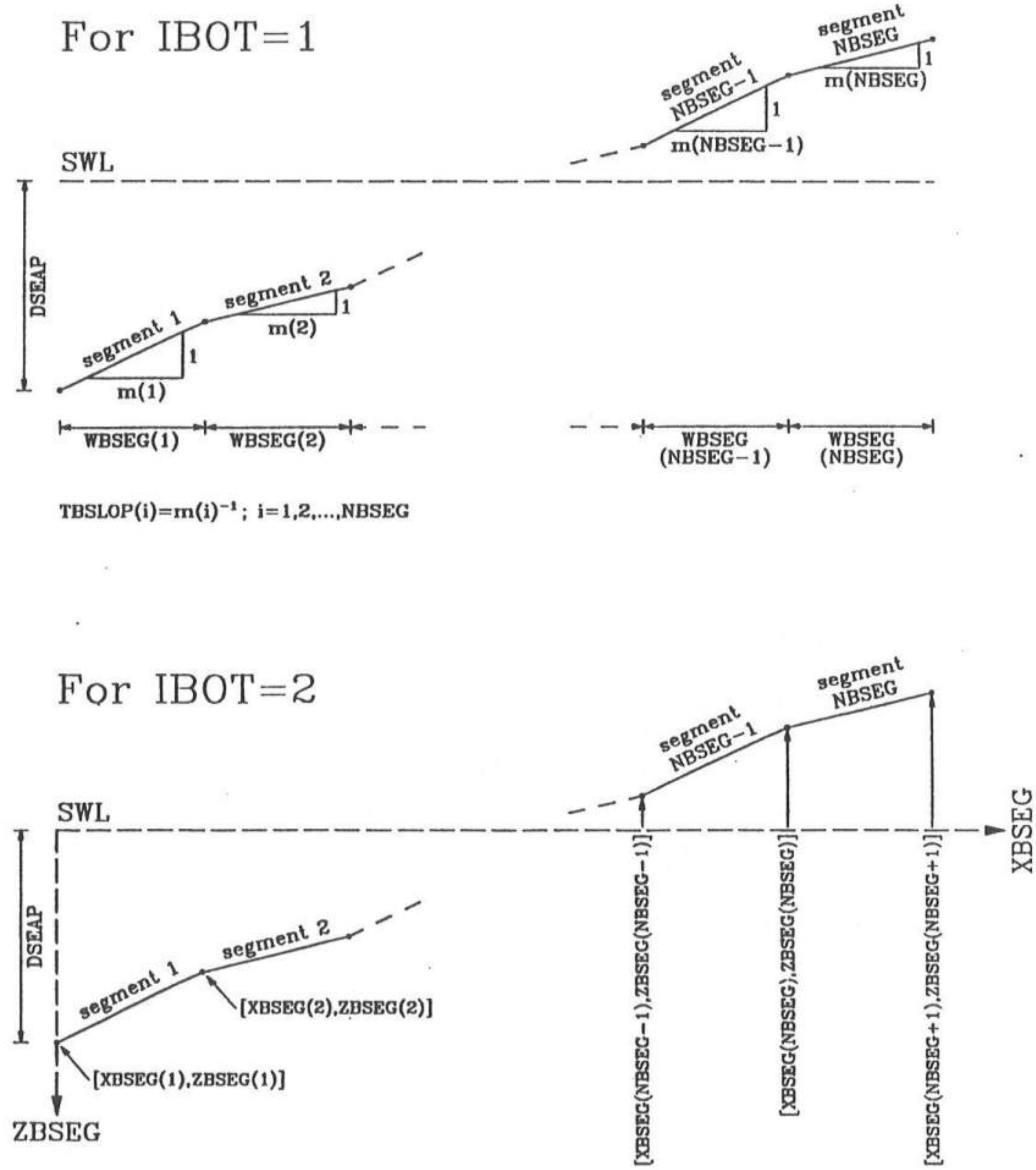

Figure 4: Input of dimensional structure geometry for IBOT=1 and IBOT=2.

ISAVA: ISAVA=1 indicates the storage of the spatial variations of η and u in the computation domain at specified moments. ISAVA=0 indicates no storage.

ISAVB: ISAVB=1 indicates the storage of the time series of m and h at specified nodes. ISAVB=0 indicates no storage.

ISAVC: ISAVC=1 indicates the storage of the time series of the armor stability function $N_R(t, x)$ given by Eq. 74 or the time series of the armor displacement X_a given by Eq. 88a from specified initial nodal locations. ISAVC=0 indicates no storage.

/CPAR1/ contains the computation parameters related to computation units.

MWAVE = number of computation units.

MSTAT: the first MSTAT computation units are excluded from the statistical calculations.

MSTAB: the first MSTAB computation units are excluded from the computation of armor stability or movement.

NTIMES, MSAVA1, MSAVA2: the spatial variations of η and u are stored NTIMES (>1) times at equal intervals per wave period from the computation unit MSAVA1 to MSAVA2, inclusive.

/CPAR2/ contains the computation parameters related to number of time steps.

MULTIF = multiplication factor associated with NONEM. See Eq. 91 in Section 4.2.1 with additional explanations.

NONEM = minimum allowable value of NONE.

NONM = case-signature read from the file BINPUT, which must be equal to NONEM.

NONE = number of time steps in one wave period.

NEND = number of time steps in one computation unit, which does not exceed NONE.

NRATE = rate (data points per wave period) of storing output time series.

NHOP = NONE/NRATE

/CPAR3/ contains the computation parameters related to spatial nodes.

INITS = number of the spatial nodes along the bottom below SWL (IJOB<3), or number of the spatial nodes between the seaward and landward boundaries, inclusive (IJOB=3). The spatial nodes are viewed to be located on the surface of the structure. INITS = 100–600 has been used, depending on the horizontal distance of the computation domain.
It is required that $1 \leq$ INITS $\leq$ (N1 $-$ 1).

JE = landward edge node (IJOB<3) or landward boundary node (IJOB=3).

JE1 = JE-1

/CPAR4/ contains the computation parameters related to the finite difference method used in RBREAK2.

X1 = damping coefficient ϵ_1 in Eqs. 24 and 25.

X2 = damping coefficient ϵ_2 in Eqs. 24 and 25 where $\epsilon_1 = \epsilon_2 = 1$ or $\epsilon_1 = \epsilon_2 = 2$ has been used and the increase of $\epsilon_1 = \epsilon_2$ tends to improve numerical stability with negligible effects on computed results (Kobayashi and Wurjanto 1992a).

DELTA = normalized water depth δ defining the computational waterline. The range of δ = 0.001–0.003 has been used and the increase of δ tends to improve numerical stability near the moving waterline. The primary input data file for RBREAK2 specifies an initial value of DELTA. This initial value should not exceed the value of DELTA for each computation unit determined by BEFORR2 starting from this initial value. This will be ensured if the value of DELTA in the promary input data file is the same for RBREAK2 amd BEFORR2.

TMAX = normalized duration of the computation.

TIME = normalized present time.

T = time step Δt used for the discretization of Eq. 15. T may vary from one computation unit to another. T=1/NONE as defined in Eq. 90.

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 = $2\Delta x$

/REQ1/ integers used for the special storage of the spatial variations of specified quantities. This special storage has been used in relation to the critical stability of armor units as described in Section 2.10 and the following integers can be specified to be zero if no special storage is required.

IREQ: IREQ=1 indicates the special storage, IREQ=0 indicates no special storage.

In the following, IELEV, IV, IDUDT, and ISNR are indicators for the storage of the spatial variation of their respective quantities at specified moments: **1** for storage, **0** for no storage.

IELEV: for the normalized free surface elevation η.

IV: for the normalized depth-averaged velocity u.

IDUDT: for the normalized horizontal fluid acceleration du/dt.

ISNR: for the dimensionless armor stability function N_R.

NREQ = number of the specified moments at which the spatial variations of the requested quantities are stored. It is required that $1 \leq$ NREQ $\leq$ N5 when IREQ=1.

/REQ2/ contains the specified moments for the special storage.

TREQ(i) = specified normalized time with i=1,2,...,NREQ at which the spatial variations of the requested quantities are stored.

/WAVE1/ contains physical wave parameters.

HREFP = physical reference wave height, H_r', specified in *meters* when SI is used or in *feet* when USCS is usea. H_r' is used to normalize the physical variables and parameters in Eqs. 3–5.

TP = physical reference wave period. T_r', in *seconds*. T_r' is used to normalize the physical variables and parameters in Eqs. 3–5.

WLOP = physical deep-water linear wavelength based on TP.

/WAVE2/ contains dimensionless wave parameters.

KS = KSSEA/KSREF. For regular wave, KS = normalized wave height at the seaward boundary. For irregular waves, it is recommended to take H_r' at the seaward boundary and specify KSREF = KSSEA = 1.0.

KSREF = shoaling coefficient at the location associated with H_r'.

KSSEA = shoaling coefficient at the seaward boundary where it is sufficient to specify KSSEA = KSREF = 1.0 unless H_r' is not at the seaward boundary.

WLO = normalized deep-water linear wavelength based on TP.

WL = normalized linear wavelength at the seaward boundary based on TP unless cnoidal wave theory is adopted for IWAVE=1.

UR = Ursell number at the seaward boundary which is equal to URPRE unless cnoidal wave theory is adopted for IWAVE = 1.

URPRE = Ursell number at the seaward boundary based on linear wavelength.

KSI = surf similarity parameter based on the slope TSLOPS, specified as input.

SIGMA = ratio of the horizontal and vertical length scales which must be sufficiently larger than unity to apply RBREAK2.

/WAVE3/ contains time series of normalized free surface elevations.

ETA = the incident regular wave train for IWAVE=1 computed by Subr. 06 REGWAV, the incident irregular wave train for IWAVE=2 read from the inpur file or the measured total free surface oscillation for IWAVE = 3 read from the input file.

The following variables are used only for regular wave computations. They are used to store their respective quantities during the last wave period.

ETAIS: for the incident wave at the seaward boundary, η_i.

ETARS: for the reflected wave at the seaward boundary, η_r.

ETATS: for the transmitted wave at the landward boundary, η_t.

/WAVE4/ contains the extreme values of ETA in /WAVE3/.

ETAMAX = maximum value of ETA in /WAVE3/.

ETAMIN = minimum value of ETA in /WAVE3/.

/WAVE5/ contains parameters related to cnoidal wave theory.

KCNO = complete elliptic integral of the first kind, $K(m)$, used in Eqs. 33-36.

ECNO = complete elliptic integral of the second kind, $E(m)$, used in Eqs. 34 and 36.

MCNO = parameter m computed from Eq. 36.

KC2 = value of $(1-m)$ used to compute the values of $K(m)$ and $E(m)$ using the Function 08 CEL.

/WAVE6/ contains integers related to ETA in /WAVE3/ for IWAVE>1 only.

NDATA = number of data points in the input wave train.

NDAT = case-signature read from the file BINPUT, which must be equal to NDATA.

/BOT1/ contains 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(i) = friction factor f' in Eq. 2 associated with linear segment i (see /BOT5/) of the structure geometry as illustrated in Fig. 4. Use has been made of $f' = 0.05$–0.3 for rough slopes and $f' = 0.01$–0.05 for smooth slopes. The computed results except for wave runup and overtopping have been found to be fairly insensitive to f'.

/BOT2/ contains 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 $\sqrt{d_t}$ 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 only.

DLAND2 = normalized group velocity $\sqrt{d_e}$ at $x = x_e$ based on linear long wave theory.

FW(j) = normalized friction factor f_j defined in Eq. 5 at the spatial node j (see /BOT3/).

TSLOPS = tangent of slope, $tan\theta'_\xi$ used to define the surf similarity parameter ξ.

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

/BOT3/ contains vectors related to the normalized structure geometry. The spatial nodes are viewed to be located on the surface of the structure. Index j refers to node number with j=1,2,...,JE.

U2INIT(j) = normalized water depth below SWL at the node j, 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 is the local angle of the slope at the node j. This quantity is required for the computation of armor stability and movement.

XB(j) = normalized x-coordinate of the node j

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

/BOT4/ contains the integer NBSEG described below.

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

/BOT5/ contains dimensional quantities associated with linear segments of the structure. All quantities listed below are segmental properties. Index i refers to segment number, with i=1,2,...,NBSEG for IBOT=1 (pairs of [WBSEG(i),TBSLOP(i)] specified), or i=1,2,...,(NBSEG+1) for IBOT=2 (pairs of [XBSEG(i),ZBSEG(i)] specified),

WBSEG(i) = horizontal width of the segment i

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/ contains the hydrodynamic quantities computed by the numerical model.

U(k,j) = 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) = normalized depth-averaged velocity $u_j = m_j/h_j$ at the node j.

ELEV(j) = normalized free surface elevation η_j above SWL at the node j.

C(j) = $c_j = \sqrt{h_j}$ defined in Eq. 11 at the node j.

DUDT(j) = normalized fluid acceleration du/dt given by Eq. 68 at the node j.

/MATRIX/ contains elements of matrices used in the numerical model. In the following, the index k=1,2.

A1(k,j) = 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) = elements of the vector $\mathbf{F}_j$ defined in Eq. 16 at the node j such that F(1,j)= $(m_j u_j + \frac{1}{2}h_j^2)$ and F(2,j)=m_j.

G1(j) = $[\theta_j h_j + f_j|u_j|u_j]$ = non-zero element of the vector $\mathbf{G}_j$ defined in Eq. 16 at the node j.

GJR(k,j) = elements of the vector $\mathbf{g}_j$ given by Eq. 18 at the node j.

S1(j) = $\left[\Delta x\ e_j - \frac{1}{2}\theta_j(m_{j+1} - m_{j-1})\right]$ = non-zero element of the vector $\mathbf{S}_j$ given by Eq. 20 at the node j where e_j is given by Eq. 21.

D(k,j) = elements of the vector $\mathbf{D}_j$ defined by Eq. 22 at the node j.

/RUNP1/ contains integers related to wave runup.

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 $1 \leq$ NDELR $\leq$ N3 if IJOB<3. For IJOB=3, RBREAK2 enforces NDELR=0.

S: For IJOB<3, S is the integer s indicating the wet node next to the moving waterline such that $h_{s+1} \leq \delta < h_s$:

- at the *present* time level $[t = (\text{M-1}) + (\text{N-1})\Delta t]$ *before* Subr. 16 LANDBC is called by the main program,
- at the *next* time level $[t = (\text{M-1}) + \text{N}\Delta t]$ *after* Subr. 16 LANDBC is called by the main program.

For IJOB=3, S=JE as discussed below.

SM1 = (s–1), used *before* Subr. 16 LANDBC is called by the the main program.

SP1 = (s+1), used *before* Subr. 16 LANDBC is called by the the main program.

JMAX = largest value of S during the time interval of the statistical calculations as explained in Section 2.4.
JMAX≤JE for IJOB=1,
JMAX=JE for IJOB=2, and
JMAX=JE=S for IJOB=3.

/RUNP2/ contains quantities related to wave runup. In the following, the index i=1,2,..., NDELR.

DELRP(i) = different values of δ'_r being specified in *millimeters* when SI is used and in *inches* when USCS is used. Each value of DELRP is independent of the others.

DELTAR(i) = normalized water depth $\delta_r = \delta'_r / H'_r$ corresponding to the different values of δ'_r. Derived from DELRP, each value of DELTAR is also independent of the others. If an armor analysis is requested (ISTAB>0), the first value, DELTAR(1), determines how far landward the armor analysis is performed. See the explanation for the related integer JSTAB in COMMON /ARMOR3/.

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

RSTAT(k,i) = mean, maximum, and minimum values of RUNUPS(i), indicated by k = 1, 2, and 3, respectively, during the time interval of the statistical calculations.

RMEAN(i) = variable used as temporary storage in the process of calculating RSTAT(1,i).

/OVER/ contains quantities related to wave overtopping during the time interval of the statistical calculations.

OV(1) = normalized average overtopping rate.

OV(2) = normalized maximum overtopping rate.

OV(3) = normalized overtopping duration.

OV(4) = normalized time when OV(2) occurs.

OVMEAN = variable used as temporary storage in the process of calculating OV(1).

/COEFS/ contains the reflection and transmission coefficients.

RCOEF(k) = wave reflection coefficients defined by Eqs. 44, 45, and 46 for k = 1, 2, and 3, respectively.

TCOEF(k) = wave transmission coefficients defined by Eqs. 55, 56, and 57 for k = 1, 2, and 3, respectively.

/STT1/ contains the statistics of the hydrodynamic quantities computed during the time interval of the statistical calculations as explained in Section 2.4. In the following, the index j refers to node number with j=1,2,...,JE. Other indices are explained following their respective variables.

ELSTAT(k,i) = mean (k=1), maximum (k=2), and minimum (k=3) values of the incident wave profile η_i (i=1), the reflected wave profile η_r (i=2), and the transmitted wave profile η_t (i=3).

U1STAT(j) = mean value of the volume flux per unit width at the node j, m_j.

ESTAT(k,j) = mean, maximum, and minimum values of the free surface elevation at the node j, η_j indicated by k = 1, 2, and 3, respectively.

VSTAT(k,j) = mean, maximum, and minimum values of the depth-averaged velocity at the node j, u_j indicated by k = 1, 2, and 3, respectively.

ELMEAN(i) = variable used as temporary storage in the process of calculating **ELSTAT(1,i)**, where the index i=1,2,3 indicates the quantities explained for **ELSTAT** above.

U1MEAN(j) = variable used as temporary storage in the process of calculating **U1STAT(j)**.

EMEAN(j) = variable used as temporary storage in the process of calculating **ESTAT(1,j)**.

VMEAN(j) = variable used as temporary storage in the process of calculating **VSTAT(1,j)**.

/STT2/ contains the parameters used to specify the time interval of the statistical calculations.

TSTAT1 = normalized time at the beginning of the time interval = t_{begin} introduced in Eq. 62.

TSTAT2 = normalized time at the end of the time interval = t_{end} introduced in Eq. 62.

/ENERG/ contains quantities related to wave energy which are computed excluding the first **MSTAT** computation units. In the following, the index j refers to node number with j=1,2,...,JE.

ENER(i,j) = quantities related to the time-averaged energy equation expressed by Eq. 62.
ENER(i,j) with i = 1, 2, 3, and 4 correspond to the values of $\overline{E}, \overline{E_F}, \overline{E_f}$, and $\overline{D_B}$, respectively. $\overline{E}, \overline{E_F}$, and $\overline{E_f}$ are the time-averaged values of E, E_F, and D_f given by Eqs. 59, 60, and 61, respectively, whereas $\overline{D_B}$ is computed using Eq. 62.
ENER(i,j) with i= 5 and 6 save the values of E at the time $t = t_{begin}$ and $t = t_{end}$, respectively, introduced in Eq. 62.
ENER(7,j) = value of the last term of Eq. 62.

ENERB(m) = quantities related to the time-averaged energy balance in the computation domain expressed by Eqs. 63–67. The index m=1,2,...15 corresponds to the following:
ENERB(1) = $\overline{E_F}(x = 0)$ in Eq. 63.
ENERB(2) = $\overline{E_F}(x = x_e)$ in Eq. 63.
ENERB(3) = $\int_0^{x_e} \overline{D_f} dx$ in Eq. 63.
ENERB(4) = $\int_0^{x_e} \overline{D_B} dx$ in Eq. 63.
ENERB(5) = last integral in Eq. 63.

ENERB(6) = left hand side of Eq. 63.
ENERB(7) = right hand side of Eq. 63.
ENERB(8) = difference between the right and left hand sides of Eq. 63.
ENERB(9) = percentage error defined as 100 x ENERB(8)/ENERB(6).
ENERB(10) = $\sqrt{d_t}\ \overline{\eta_i^2}$ in Eq. 65.
ENERB(11) = $\sqrt{d_t}\ \overline{(\eta_r - \overline{\eta_r})^2}$ in Eq. 65.
ENERB(12) = $\sqrt{d_e}\ \overline{(\eta_t - \overline{\eta_t})^2}$ in Eq. 67.
ENERB(13) = right hand side of Eq. 65.
ENERB(14) = percentage error in the approximate expression given by Eq. 65.
ENERB(15) = percentage error in the approximate expression given by Eq. 67.
The last two quantities may be used to estimate the uncertainties associated with the computed reflection and transmission coefficients.

ENTEMP(k,j) = variable used as temporary storage in the process of calculating ENER(k,j) at the node j with k = 1, 2, and 3.

/ARMOR1/ contains input parameters related to armor stability and movement in Section 2.10.

C2 = area coefficient C_2 of the armor unit.

C3 = volume coefficient C_3 of the armor unit.

CD = drag coefficient C_D involved in the hydrodynamic drag force..

CL = lift coefficient C_L involved in the hydrodynamic lift force.

CM = inertia coefficient C_M involved in the hydrodynamic inertia force.

SG = specific gravity s_g of the armor unit.

TANPHI = $\tan\phi$, with ϕ = frictional angle of the armor unit.

AMIN = parameter a_{min} specified as input. The condition for a_{min} given in Eq. 77 needs to be satisfied if ISTAB=1.

AMAX = parameter a_{max} specified as input. The condition for a_{max} given in Eq. 77 needs to be satisfied if ISTAB=1.

DAP = characteristic length d' of the armor unit given in Eq. 83 which needs to be specified as input if ISTAB=2 and the sliding motion of the armor unit is to be computed.

/ARMOR2/ contains computed parameters related to armor stability and movement.

SG1 = (s_g−1) used in Eqs. 81 and 82.

CTAN(j) = value of ($cos\theta_j'\ tan\phi$) in Eq. 73 at the node j, where θ_j' = local angle of the slope at the node j.

/ARMOR3/ contains integers related to armor stability and movement.

JSTAB = integer indicating the largest node number for which the computation of armor stability or movement is performed. For IJOB<3, the node number JSTAB corresponds to the normalized water depth DELTAR(1) which, in turn, corresponds to the *first* value of δ_r'. For IJOB=3, JSTAB=JE set in Subr. 26 INIT.

JSTABM = largest value of JSTAB during the time interval of the statistical calculations.

/ARMOR4/ contains integers related to armor stability (ISTAB=1).

JSNSC = node number where the minimum armor stability occurs.

JATMIN = value of JSTAB at the time of the minimum armor stability. JSTAB is described above.

/ARMOR5/ contains parameters and values related to armor stability.

CSTAB1 = $[2C_3^{2/3}/(C_2C_D)]$ used in Eq. 71.

CSTAB2 = $[C_M/[(s_g-1)\sigma]]$ used in Eqs. 71 and 75.

AMAXS = $(\sigma\ a_{max})$ used in Eq. 76.

AMINS = $(\sigma\ a_{min})$ used in Eq. 76.

E2 = parameter E_2 defined by Eq. 72.

E3PRE(j) = value of $[2C_3^{2/3}\ cos\theta'\ tan\phi/(C_2C_D)]$ at the node j used in Eq. 73.

SNSC = critical stability number N_{sc} for initiation of armor movement.

TSNSC = normalized time when the minimum armor stability occurs.

SNR(j) = armor stability function $N_R(t,x)$ at the node j and at the time $t^* =$ [(M-1) + (N)Δt].

SNSX(j) = local stability number $N_{sx}(x)$ at the node j defined as the minimum value of $N_R(t,x)$ at the node j.

TSNSX(j) = normalized time when the local stability number $N_{sx}(x)$ at the node j occurs.

ATMIN(i,j) saves the following quantities at node j at the time of the minimum armor stability:
i=1 for the normalized free surface elevation η
i=2 for the normalized depth-averaged velocity u
i=3 for the normalized horizontal fluid acceleration du/dt
i=4 for the dimensionless armor stability function N_R

/ARMOR6/ contains integers related to armor movement (ISTAB=2).

NMOVE = number of armor units moved from their initial locations where armor movement is computed excluding the first MSTAB computation units.

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.

NODFI(j) = node number closest to the armor unit initially located at the node j, at the end of each time step.

/ARMOR7/ contains parameters and values related to armor movement.

CSTAB3 = $[C_2C_D/(2C_3d)]$ used in Eq. 78.

CSTAB4 = $[C_2C_L/(2C_3d)]$ used in Eq. 79.

CM1 = (C_M-1) = added mass coefficient C_m used in Eq. 86.

DA = normalized characteristic length d of the armor unit defined in Eq. 83.

SIGDA = (σ/d) used in Eq. 88a.

WEIG = $[\sigma\,(s_g - 1)]$ used in Eqs. 81 and 82.

VA(j) = normalized velocity u_a of the armor unit located initially at the node j which is computed using Eq. 86.

XAA(j) = normalized displacement X_{aa} defined by Eq. 88b of the armor unit from its initial location at the node j.

XA(j) = normalized displacement X_a defined by Eq. 88a of the armor unit from its initial location at the node j.

TDIS(j) = normalized time when the armor unit initially located at the node j started moving.

/SAVBC/ contains integers associated with options ISAVB=1 and ISAVC=1.

NNODB = number of nodes for which the time series of m and h are stored if ISAVB=1. It is required that $1 \leq$ NNODB $\leq$ N5 when ISAVB=1.

NNODC = number of nodes for which the time series of either of the following two quantities are stored if ISAVC=1.
(1) For ISTAB=1: the stability function $N_R(t,x)$ given by Eq. 74.
(2) For ISTAB=2: the displacement X_a given by Eq. 88a
It is required that $1 \leq$ NNODC $\leq$ N5 when ISAVC=1.

NODB(i) = node number for the storage of the time series for ISAVB=1. The index i runs from 1 to NNODB. It is required that $1 \leq$ NODB(i) $\leq$ JE, where JE is described in COMMON /CPAR3/.

NODC(i) = node number for the storage of the time series for ISAVC=1. The index i runs from 1 to NNODC. It is required that $1 \leq$ NODC(i) $\leq$ JE, where JE is described in COMMON /CPAR3/.

/VALUEN/ retains some of the values at the *present* time level t = [(M-1) + (N-1)Δt].

VSN = u_s used for the landward boundary computation.

USN(1) = m_s used for the landward boundary computation.

USN(2) = h_s used for the landward boundary computation.

VMN = u_{s-1} used for the landward boundary computation.

UMN(1) = m_{s-1} used for the landward boundary computation.

V1N = u_1 used in Eq. 37 or 42.

V2N = u_2 used in Eq. 37 or 42.

• 3.6 •

READING THE PRIMARY INPUT DATA FILE

The contents of the primary input data file is read by Subr. 02 **INPUT1**. This subroutine provides clear explanations on the input parameters and variables it reads. It is recommended that a user follow the explanations in preparing the primary input data file. This section summarizes concisely the **READ** statements corresponding to the the primary input data file.

3.6.1 FORMATS

Input data in the primary input data file are read by Subr. 02 **INPUT1** using nine **FORMAT**s listed below.

```
1110 FORMAT (3I8)
1120 FORMAT (15A5)
1130 FORMAT (2I1)
1140 FORMAT (I1,I8,2X,A10)
1150 FORMAT (I1,3I8)
1160 FORMAT (5I1,I8)
1170 FORMAT (F15.6)
1180 FORMAT (3F13.6)
1190 FORMAT (5I6)
```

These formats are used in Sections 3.6.2 and 3.6.3.

3.6.2 HEADER

The primary input data file has a header in which a user may write pertinent information or comments. The header is read by Subr. 02 **INPUT1** in the following order.

```
   READ (11,1110) NLINES
   DO 110 L = 1,NLINES
      READ (11,1120) (COMMEN(I),I=1,15)
110 CONTINUE
```

where **NLINES** is the number of lines containing the user's comments. Thus, the first line in the primary input data file contains the integer to be read as **NLINES** by Subr. 02 **INPUT1**. The **NLINES** lines containing the user's comments follow immediately. These comments are then written as the header of the output files **ODOC** and **OMSG**.

3.6.3 INPUT DATA

Immediately following the header, input parameters and variables are read according to the formats listed in Section 3.6.1 in the order to be described below. In the following list: (1) **READ** statements are listed on the left, (2) Notes written on the right show where the parameter(s) or variable(s) read are explained, and (3) Sequential numerals in the middle are provided for convenience. An example of the input data is given in Section 5.3.2.

```
READ (11,1130) IJOB,ISTAB                        [1]     COMMON /ID/

READ (11,1130) ISYST                             [2]     COMMON /ID/

READ (11,1130) IBOT                              [3]     COMMON /ID/

READ (11,1130) INONCT                            [4]     COMMON /ID/

READ (11,1130) IENERG                            [5]     COMMON /ID/

READ (11,1140) IWAVE,MSTAT,FINP2                 [6]     COMMON /ID/
                                                         COMMON /CPAR1/
                                                         Section 3.2

READ (11,1150) ISAVA,MSAVA1,MSAVA2,NTIMES        [7]     COMMON /ID/
                                                         COMMON /CPAR1/

READ (11,1150) ISAVB,NNODB                       [8]     COMMON /ID/
                                                         COMMON /SAVBC/

READ (11,1150) ISAVC,NNODC                       [9]     COMMON /ID/
                                                         COMMON /SAVBC/

READ (11,1160) IREQ,IELEV,IV,IDUDT,ISNR,NREQ     [10]    COMMON /REQ1/

IF (IREQ.EQ.1) THEN                                      COMMON /REQ1/
   READ (11,1180) (TREQ(I),I=1,NREQ)             [11]    COMMON /REQ2/
ENDIF
```

```
      READ (11,1110) INITS                              [12]    COMMON /CPAR3/

      READ (11,1110) MULTIF                             [13]    COMMON /CPAR2/

      READ (11,1170) DELTA                              [14]    COMMON /CPAR4/

      READ (11,1170) TMAX                               [15]    COMMON /CPAR4/

      READ (11,1110) NRATE                              [16]    COMMON /CPAR2/

      READ (11,1110) NDELR                              [17]    COMMON /RUNP1/

      DO 130 L = 1,NDELR                                        COMMON /RUNP1/
         READ (11,1180) DELRP(L)                        [18]    COMMON /RUNP2/
  130 CONTINUE

      READ (11,1180) X1,X2                              [19]    COMMON /CPAR4/

      READ (11,1180) HREFP,TP                           [20]    COMMON /WAVE1/

      READ (11,1180) KSREF,KSSEA                        [21]    COMMON /WAVE2/

      READ (11,1180) DSEAP                              [22]    COMMON /BOT1/

      READ (11,1180) TSLOPS                             [23]    COMMON /BOT2/

      READ (11,1110) NBSEG                              [24]    COMMON /BOT4/

      IF (IBOT.EQ.1) THEN                                       COMMON /ID/
                                                                COMMON /BOT1/
         DO 140 K = 1,NBSEG                                     COMMON /BOT4/
            READ (11,1180) WBSEG(K),TBSLOP(K),FWP(K)    [25]    COMMON /BOT5/
  140    CONTINUE
      ELSE
                                                                COMMON /BOT1/
         DO 150 K = 1,NBSEG+1                                   COMMON /BOT4/
            READ (11,1180) XBSEG(K),ZBSEG(K),FWP(K)     [26]    COMMON /BOT5/
  150 CONTINUE
      ENDIF

      IF (ISAVB.EQ.1) READ (11,1190) (NODB(I),I=1,NNODB)
                                                        [27]    COMMON /ID/
                                                                COMMON /SAVBC/

      IF (ISTAB.GT.0) THEN                                      COMMON /ID/
         READ (11,1180) C2,C3,SG                        [28]    COMMON /ARMOR1/
```

```
   READ (11,1180) CD,CL,CM                           [29]     COMMON /ARMOR1/
   READ (11,1180) TANPHI                             [30]     COMMON /ARMOR1/
   READ (11,1180) AMAX,AMIN                          [31]     COMMON /ARMOR1/
   IF (ISAVC.EQ.1) READ (11,1190) (NODC(I),I=1,NNODC)
ENDIF                                                [32]     COMMON /ID/
                                                              COMMON /SAVBC/

IF (ISTAB.EQ.2) READ (11,1180) DAP                   [33]     COMMON /ID/
                                                              COMMON /ARMOR1/
```

It is noted that for IBOT=2, FWP(NBSEC+1) is not used in the computation and this value can be set to be zero.

• 3.7 •

WARNING AND ERROR MESSAGES

Warning and error messages are written in the file ODOC and displayed on screen. There is only one warning message that the computer program RBREAK may issue, that is,

```
From Subr.  07 FINDM:
Criterion for parameter m=MCNO not satisfied
```

The above warning is related to the iteration scheme to compute the parameter m using Eq. 36, and thus corresponds to the case of regular cnoidal waves only. This warning has never been experienced by the authors. RBREAK2 does not automatically cease computation when this warning is issued. The results obtained under this circumstance, however, may not be reliable.

Computation is terminated immediately following any error message. If the preceding BEFORR2 computation is successful, serious potential errors in an RBREAK2 execution are likely limited to those related to the dimensions of the vectors and matrices, *i.e.*, the PARAMETERs N1, N2, N3, N4, and N5 explained in COMMON /DIMENS/ in Section 3.5. Here is a list of error messages related to the dimensions. The error messages are self-explanatory.

1. If the value of any of the PARAMETERs N1, N2, N3, N4, and N5 in a subroutine that utilizes the PARAMETERs does not match with the corresponding value specified in the main program:

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

2. If the requirement $1 \leq$ INITS $\leq$ (N1 $-$ 1) is not satisfied:

```
Input Error:  INITS = ···
Specify INITS in the range of [1,···]
Change PARAMETER N1 if necessary
```

3. If JE > N1:

```
End Node = ···; N1 = ···
Slope/Structure is too long.
Cut it, or change PARAMETER N1.
```

4. If the requirement $1 \leq$ NDATA $\leq$ N2 when IWAVE>1 is not satisfied:

```
Input Error:  NDATA = ···
Specify NDATA in the range of [1,···]
Change PARAMETER N2 if necessary
```

5. If the requirement $1 \leq$ NDELR $\leq$ N3 when IJOB<3 is not satisfied:

```
Input Error:  NDELR = ···
Specify NDLER in the range of [1,···]
Change PARAMETER N3 if necessary
```

6. If the requirement $1 \leq$ NBSEG $\leq$ (N4 $-$ 1) is not satisfied:

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

7. If the requirement $1 \leq$ NREQ $\leq$ N5 when IREQ=1 is not satisfied:

```
Input Error:  NREQ = ···
Specify NREQ in the range of [1,···]
Change PARAMETER N5 if necessary
```

8. If the requirement $1 \leq$ NNODB $\leq$ N5 when ISAVB=1 is not satisfied:

```
Input Error:  NNODB = ···
Specify NNODB in the range of [1,···]
Change PARAMETER N5 if necessary
```

9. If the requirement $1 \leq$ NNODC $\leq$ N5 when ISAVC=1 is not satisfied:

```
Input Error:  NNODC = ···
Specify NNODC in the range of [1,···]
Change PARAMETER N5 if necessary
```

10. If the requirement $1 \leq$ NODB(i) $\leq$ JE when ISAVB=1 is not satisfied:

```
Input Error:  NODB = ···
Specify NODB in the range of [1,···]
```

11. If the requirement $1 \leq$ NODC(i) $\leq$ JE when ISAVC=1 is not satisfied:

```
Input Error:  NODC = ···
Specify NODC in the range of [1,···]
```

12. For the case of regular waves, the file BINPUT also contains NONEMA = the largest value of NONE used in the preceding BEFORR2 computation, to be compared with the value of PARAMETER N2 of the computer program RBREAK2. It is required that N2>NONEMA. The following message is issued if this condition is not met:

```
From Subr.  02 INPUT1:
Computation will need NONE as large as ···
Dimension N2 must be greater than NONE.
Currently, N2 = ···
Specify larger N2 as required.
Computation aborted.
```

The parameters and variables involved in the above error messages are explained in Section 3.5 as indicated below.

VARIABLE	DESCRIPTION IN SECTION 3.5
N1,N2,N3,N4,N5	COMMON /DIMENS/
NONE	COMMON /CPAR2/
INITS,JE	COMMON /CPAR3/
NREQ	COMMON /REQ1/
NDATA	COMMON /WAVE6/
NBSEG	COMMON /BOT4/
NDELR	COMMON /RUNP1/
NODB,NODC,NNODB,NNODC	COMMON /SAVBC/

Less serious errors and their causes are listed in the following. Error no. (1) is a simple error. Errors no. (2) to (5) indicate that the case-signatures from the file BINPUT do not match with the corresponding values in RBREAK2. Error no. (6) should never happen if the file BINPUT has not been altered.

1. If MREP<0:

```
MREP must be positive or zero.
Programmed Stop.
```

2. If the case-signature MWAV does not match with MWAVE:

```
Computation duration does not match.  Check BINPUT.
Programmed Stop.
```

3. If the case-signature FID2 does not match with FINP2:

```
Second input file name does not match.  Check BINPUT.
Programmed Stop.
```

4. If the case-signature NDAT does not match with NDATA:

```
Number of data points for input wave train
does not match.  Check BINPUT.
Programmed Stop.
```

5. If the case-signature NONM does not match with NONEM:

```
Minimum allowable NONE does not match.  Check BINPUT.
Programmed Stop.
```

6. If the present computation unit M does not match with the sequential number in the file BINPUT:

```
Computation unit not matched.  Check BINPUT.
Programmed Stop.
```

The variables involved in the above error messages are explained previously as indicated below.

VARIABLE	DESCRIPTION
MREP	Described in Section 3.3.1
MWAVE=MWAV	Described in COMMON /CPAR1/ in Section 3.5
FINP2=FID2	Described in Section 3.2
NDATA=NDAT	Described in COMMON /WAVE6/ in Section 3.5
NONEM=NONM	Described in COMMON /CPAR2/ in Section 3.5
M	Described in Section 3.3.4

The computer program RBREAK2 contains other error messages which are highly unlikely to occur if the duo BEFORR2-RBREAK2 are executed properly. These messages are put in RBREAK2 to provide programmed escape routes just in case that something unexpected occurs. These error messages are self-explanatory.

• 3.8 •
OUTPUT PRODUCED BY RBREAK2

Output from the computer program RBREAK2 is stored in files whose names start with the letter "O" for easy identification. The number of output files varies depending on the options selected by a user. Table 3 lists the names of all possible output files generated by RBREAK2. The first two files in Table 3, ODOC and OMSG, contain information which should be read and checked by the user. The rest contain the computed results that may need further processing to yield useful information for the user. The files OSTAT, OSPACE, OREQ, and OSTAB contain spatial variations. The files OSEAWAV, ORUNUP, OOVER, and OTRANS contain time series covering the time interval $0 < t \leq$ TMAX and so do the families of files OSAV1Bxx, OSAV2Bxx, and OSAVECxx.

Except for the contents of the files ODOC and OMSG, all other output values are written using only two kinds of FORMATs, that is, FORMAT 8000 for double precision values and FORMAT 9000 for integers.

```
8000 FORMAT (5D15.6)
9000 FORMAT (8I8)
```

FORMATs 8000 and 9000 appearing in the WRITE statements in the subsequent sections refer to the above specifications. In addition, the variables involved in the WRITE statements are tabulated with concise descriptions in the following.

3.8.1 GENERAL

The following output files are always generated by RBREAK2.

Table 3: List of output files generated by RBREAK2.

Output File Name	Condition for Creation	Discussed in Section
ODOC OMSG OSTAT OSPACE OSEAWAV	No Condition	3.8.1
ORUNUP	IJOB<3	3.8.2
OOVER	IJOB=2	3.8.2
OTRANS	IJOB=3	3.8.3
OREQ	IREQ=1	3.8.4
OSTAB	ISTAB>0	3.8.5
OSAV1B01 OSAV1B02 . . and so on	ISAVB=1	3.8.6
OSAV2B01 OSAV2B02 . . and so on	ISAVB=1	3.8.6
OSAVEC01 OSAVEC02 . . and so on	ISAVC=1 and ISTAB>0	3.8.6

File ODOC contains a summary of input data and computed results. The contents of this file can be seen in the example given in Section 5.3.2.

File OMSG contains warning and error messages explained in Section 3.7. There should be no error message generated by a successful RBREAK2 computation.

File OSTAT stores the statistics of hydrodynamic quantities. The contents of this file are written in Subr. 45 DOC3 in the following order:

```
      WRITE (51,9000) JMAX
      WRITE (51,8000) (U1STAT(J),J=1,JMAX)
      DO 120 I = 1,3
         WRITE (51,8000) (ESTAT(I,J),J=1,JMAX)
         WRITE (51,8000) (VSTAT(I,J),J=1,JMAX)
  120 CONTINUE
```

VARIABLE	DESCRIPTION IN SECTION 3.5
JMAX	COMMON /RUNP1/
U1STAT,ESTAT,VSTAT	COMMON /STT1/

File OSPACE stores the normalized structure geometry written in Subr. 43 DOC1 as follows:

```
      WRITE (52,9000) JE
      WRITE (52,8000) (XB(J),ZB(J),J=1,JE)
```

In addition, if ISAVA=1, Subr. 44 DOC2 writes the following output [NTIMESx(MSAVA2-MSAVA1+1)] times.

```
      WRITE (52,9000) S
      WRITE (52,8000) (ELEV(J),V(J),J=1,S),TIME
```

VARIABLE	DESCRIPTION IN SECTION 3.5
ISAVA	COMMON /ID/
NTIMES,MSAVA1,MSAVA2	COMMON /CPAR1/
JE	COMMON /CPAR3/
TIME	COMMON /CPAR4/
XB,ZB	COMMON /BOT3/
S	COMMON /RUNP1/
ELEV,V	COMMON /HYDRO/

File OSEAWAV stores the time series of hydrodynamic quantities at the seaward boundary. The following four quantities are stored at the rate of NRATE data points per wave period in Subr. 44 DOC2:

```
WRITE (61,8000) TIME,ETAI,ETAR,U(1,1)
```

VARIABLE	DESCRIPTION
NRATE	Described in COMMON /CPAR2/ in Section 3.5
TIME	Described in COMMON /CPAR4/ in Section 3.5
ETAI	= η_i at $x = 0$ at t=TIME
ETAR	= η_r at $x = 0$ at t=TIME
U(1,1)	= m at $x = 0$ at t=TIME

3.8.2 SUBAERIAL STRUCTURES (IJOB<3)

If IJOB<3, the file **ORUNUP** is produced. In addition, if IJOB=2, the file **OOVER** is also created.

File **ORUNUP** contains the time series of quantities related to runup. The time series are written in Subr. 44 DOC2 at the rate of **NRATE** data points per wave period.

```
WRITE (62,8000) DBLES,(RUNUPS(L),L=1,NDELR),TIME.
```

VARIABLE	DESCRIPTION
IJOB	Described in COMMON /ID/ in Section 3.5
NRATE	Described in COMMON /CPAR2/ in Section 3.5
TIME	Described in COMMON /CPAR4/ in Section 3.5
NDELR,S	Described in COMMON /RUNP1/ in Section 3.5
RUNUPS	Described in COMMON /RUNP2/ in Section 3.5
DBLES	= DBLE(S) at t=TIME

File **OOVER** contains the time series of hydrodynamic quantities at the landward-end node written in Subr. 44 DOC2 at the rate of **NRATE** data points per wave period.

```
WRITE (63,8000) U(1,JE),U(2,JE)
```

VARIABLE	DESCRIPTION
NRATE	Described in COMMON /CPAR2/ in Section 3.5
U(1,JE)	= m at $x = x_e$
U(2,JE)	= h at $x = x_e$

3.8.3 SUBMERGED STRUCTURES (IJOB=3)

File **OTRANS** is created if IJOB=3. It contains the time series of hydrodynamic quantities at the landward-end node written in Subr. 44 DOC2 at the rate of NRATE data points per wave period.

```
WRITE (64,8000) U(1,JE),U(2,JE),ETAT,TIME
```

VARIABLE	DESCRIPTION
IJOB	Described in COMMON /ID/ in Section 3.5
NRATE	Described in COMMON /CPAR2/ in Section 3.5
TIME	Described in COMMON /CPAR4/ in Section 3.5
U(1,JE)	= m at $x = x_e$ at t=TIME
U(2,JE)	= h at $x = x_e$ at t=TIME
ETAT	= η_t at $x = x_e$ at t=TIME

3.8.4 SPECIAL REQUEST (IREQ=1)

File **OREQ** is reserved for the special request in the computer program RBREAK2 to store the spatial variations of certain quantities at designated moments. The designated moments are the normalized time TREQ(i) with i=1,2,...NREQ. The special request is activated by specifying IREQ=1. Quantities available for the request are η, u, du/dt, and N_R, which are stored if IELEV=1, IV=1, IDUDT=1, and ISNR=1, respectively. Note that the spatial variation of N_R can be requested only if ISTAB=1. The requested spatial variations at the designated time are written in the file OREQ in Subr. 44 DOC2 as follows:

```
WRITE (55,9000) S
WRITE (55,8000) TIME
IF (IELEV.EQ.1) WRITE (55,8000) (ELEV(J),J=1,S)
IF (IV.EQ.1)    WRITE (55,8000) (V(J),J=1,S)
IF (IDUDT.EQ.1) WRITE (55,8000) (DUDT(J),J=1,S)
IF (ISNR.EQ.1)  WRITE (55,8000) (SNR(J),J=1,S)
```

VARIABLE	DESCRIPTION IN SECTION 3.5
ISTAB	COMMON /ID/
TIME	COMMON /CPAR4/
IREQ,NREQ,IELEV,IV,IDUDT,ISNR	COMMON /REQ1/
TREQ	COMMON /REQ2/
ELEV,V,DUDT	COMMON /HYDRO/
S	COMMON /RUNP1/
SNR	COMMON /ARMOR5/

3.8.5 Armor Analysis (ISTAB>0)

File OSTAB is created if the analysis of either armor stability (ISTAB=1) or armor movement (ISTAB=2) is performed.

For ISTAB=1, Subr. 45 DOC3 writes the following:

```
      WRITE (54,9000) JSTABM
      DO 140 J = 1,JSTABM
         WRITE (54,8000) XB(J),ZB(J),SNSX(J),TSNSX(J)
  140 CONTINUE
      WRITE (54,9000) JATMIN
      WRITE (54,8000) TSNSC
      DO 150 I = 1,4
         WRITE (54,8000) (ATMIN(I,J),J=1,JATMIN)
  150 CONTINUE
```

For ISTAB=2, Subr. 45 DOC3 writes the following:

```
      WRITE (54,9000) NMOVE
      WRITE (54,9000) NSTOP
      WRITE (54,9000) JSTABM
      DO 160 J = 1,JSTABM
         IF (ISTATE(J).GE.1) THEN
            WRITE (54,9000) J,NODFI(J),ISTATE(J)
            WRITE (54,8000) XB(J),ZB(J),XA(J),TDIS(J)
         ENDIF
  160 CONTINUE
```

Variable	Description in Section 3.5
ISTAB	COMMON /ID/
XB,ZB	COMMON /BOT3/
JSTABM	COMMON /ARMOR3/
JATMIN	COMMON /ARMOR4/
SNSX,TSNSX,TSNSC,ATMIN	COMMON /ARMOR5/
NMOVE,NSTOP,NODFI,ISTATE	COMMON /ARMOR6/
XA,TDIS	COMMON /ARMOR7/

3.8.6 Nodal Time Series (ISAVB=1 and ISAVC=1)

The user has the option to obtain the time series of m and h at specified nodes by specifying ISAVB=1. The time series are stored in groups of 100 waves to avoid creating too large an output file. The rate of storing is NRATE data points per wave period. The values at all specified nodes are stored simultaneously in one file at the storage time. The output files are named as follows:

OSAV1Bxx for m, and
OSAV2Bxx for h.

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

The nodes for the storage of the time series are specified by the node number NODB(i) with i=1,2,...,NNODB. Writing is performed in Subr. 44 DOC2 at each storage time as follows:

```
      DO 110 I = 1,NNODB
         J = NODB(I)
         VONE(I) = U(1,J)
         VTWO(I) = U(2,J)
  110 CONTINUE
      WRITE (71,8000) (VONE(I),I=1,NNODB),TIME
      WRITE (72,8000) (VTWO(I),I=1,NNODB)
```

In addition, if ISTAB>0, the time series of either N_R (for ISTAB=1) or X_a (for ISTAB=2) at specified nodes can be stored by specifying ISAVC=1. The method of storing these time series is similar to that of the option ISAVB=1. The specified nodes are at the node number NODC(i) with i=1,2,...,NNODC. The output file names are OSAVECxx.

The following is written in Subr. 44 DOC2 at each storage time.

```
      IF (ISTAB.EQ.1) THEN
         DO 120 I = 1,NNODC
            J = NODC(I)
            VONE(I) = SNR(J)
  120    CONTINUE
      ELSE
         DO 130 I = 1,NNODC
            J = NODC(I)
            IF (ISTATE(J).EQ.0) THEN
               VONE(I) = 0.D+00
            ELSE
               VONE(I) = XA(J)
            ENDIF
  130    CONTINUE
      ENDIF
      WRITE (73,8000) (VONE(I),I=1,NNODC),TIME
```

VARIABLE	DESCRIPTION IN SECTION 3.5
ISTAB,ISAVB,ISAVC	COMMON /ID/
NRATE	COMMON /CPAR2/
TIME	COMMON /CPAR4/
U	COMMON /HYDRO/
SNR	COMMON /ARMOR5/
XA	COMMON /ARMOR7/
NODB,NODC,NNODB,NNODC	COMMON /SAVBC/

The present setting of the computer program RBREAK2 enables the storage of the time series associated with ISAVB=1 and ISAVC=1 of up to 2000 waves. If a larger storage is needed, the dimension of the variables FSAV1B, FSAV2B, and FSAVEC in Subr. 01 OPENIO, which is presently set at 20, has to be increased. Consequently, the statements DATA FSAV1B, DATA FSAV2B, and DATA FSAVEC in Subr. 01 OPENIO need to be adjusted accordingly.

PART IV
COMPUTER PROGRAM BEFORR2

• 4.1 •
INTRODUCTION

Section 3.1 describes the reason behind the creation of the computer program BEFORR2, which is listed in Appendix C. To be compatible with RBREAK2, the computer program BEFORR2 is also written in standard FORTRAN-77 with full double precision mode.

BEFORR2 and RBREAK2 read the same primary input data file and, for the case of IWAVE=2 and 3, the same file containing the input wave profile. The computer program BEFORR2 reads the primary input data file in the same way as the computer program RBREAK2, that is, following the sequence and formats listed in Section 3.6.3. It is BEFORR2 that checks most of the contents of the primary input data file. RBREAK2 checks only the dimensions of the vectors and matrices. The dimensions of the vectors and matrices N1, N2, N3, N4, and N5 for BEFORR2 and RBREAK2 are independently specified in each program, and hence can be different. In the present setting, as listed in Appendices B and C, the values of N1, N2, N3, N4, and N5 for both programs are identical, that is, N1=800, N2=24000, N3=3, N4=40, N5=40. The other checks are not repeated in RBREAK2, assuming that the primary input data file is not altered after a successful BEFORR2 computation. A list of the requirements enforced by BEFORR2 is presented in Section 4.5 along with warning and error messages.

BEFORR2 produces two kinds of computation logs. The first log lists all the numerical problems chronologically as explained in Section 4.2.2. This log is written in the file BMSG as well as on screen. The second computation log is written in the file BDOC and includes the computation parameters and indicators for each successful computation unit. Some of these parameters and indicators are written on screen every MREP wave periods as explained in Section 3.3.1. Together, the two computation logs provide concise output on how the hydrodynamic computation has been executed by BEFORR2 for each case.

Part IV of this report is written in such a way that items discussed already in Part III are referred to the appropriate Sections of Part III, assuming the user is familiar with RBREAK2.

• 4.2 •
MAIN PROGRAM BEFORR2

The computer program BEFORR2 contains essentially the hydrodynamic part of the computer program RBREAK2. BEFORR2 examines the hydrodynamic computation and obtains the appropriate computation parameters for each computation unit to avoid numerical problems in the hydrodynamic computation for that unit if possible. The main program BEFORR2 includes a mechanism for adjusting the computation parameters for each computation unit automatically. However, the adjustment may not always be successful in the first attempt as explained in Section 4.2.1.

BEFORR2 and RBREAK2 are organized in a similar manner. The tasks performed by the main program of BEFORR2 can also be categorized into five groups: (1) Reading input data, (2) Checking FORTRAN PARAMETERs in the subroutines, (3) Groundwork, (4) Time-marching computation, and (5) Finishing. The tasks of the groups (1), (2), (3), and (5) for BEFORR2 are similar to those for RBREAK2 as explained in Section 3.3.

The self-adjusting mechanism of BEFORR2 to overcome numerical difficulties during the time-marching computation is explained in detail in the rest of this section. The numerical method for the hydrodynamic computation from one time level to next is the same as that for RBREAK2 explained in Section 3.3.4.

4.2.1 Three Crucial Computation Parameters

The numerical stability and accuracy of the computation are mainly determined by the following three parameters:

1. **INITS** = number of the spatial nodes below SWL. A user specifies the value of **INITS**. **INITS** should be so large that the spacing between two adjacent nodes is small enough to resolve the spatial variation of the wave motion. **INITS**=100 to 600 has been used depending on the horizontal distance of the computation domain. A larger value of **INITS** is desirable for the accuracy of computation, but leads to a longer computation time.

2. **NONE** = number of time steps in one wave period. **NONE** is the inverse of the time step Δt, as expressed in Eq. 90. The following three requirements are imposed on the selection of an appropriate value of **NONE**.

   (a) The numerical stability criterion expressed by Eq. 29 with Eq. 28 needs to be satisfied.

   (b) For storing the computed time series at the rate of **NRATE** data points per wave period, **NONE** should be divisible by **NRATE**.

   (c) For regular wave computations, **NONE** should be an even number for the symmetry about the crest of the assumed Stokes second-order wave or cnoidal wave profile.

   Denoting the smallest integer satisfying the above three requirements by $\mathcal{N}$ and introducing an integer multiplication factor **MULTIF**, the minimum allowable value **NONEM** of **NONE** is given by

   $$\text{NONEM} = \text{MULTIF} \times \mathcal{N} \tag{91}$$

   Starting from **NONEM**, the computer program BEFORR2 attempts to find the value of **NONE** for each computation unit which does not cause numerical problems. The integer **MULTIF** should be unity or greater and may be used to adjust the accuracy of the computation since the increase of **MULTIF** results in the increase of **NONE** and hence the decrease of the time step Δt. It is suggested to increase **MULTIF** from unity if the computed results show unrealistic spikes and kinks (Wurjanto and Kobayashi 1991).

3. DELTA = normalized water depth used to specify the computational waterline. An *initial* value of DELTA needs to be specified by a user. For a regular wave computation, the value of DELTA supplied by the user is not modified. For the case of IWAVE=2 and 3, BEFORR2 may reduce the initial value of DELTA for subsequent computation units if it is deemed necessary to overcome numerical problems. The values of DELTA ranging from 0.001 to 0.003 have been used. It is suggested that the user specify DELTA=0.001 first. If the resulting computation fails or experiences too many trial runs to overcome numerical difficulties, the value of DELTA should be increased to 0.002 or 0.003.

4.2.2 Potential Numerical Problems in Hydrodynamic Computation

There are three potential numerical problems that may be encountered in the hydrodynamic computation. The three problems are identified by IPROB=1, IPROB=2, and IPROB=3. If any of the three problems occurs in BEFORR2, pertinent log messages will be written both in the file BMSG and on terminal screen. If IPROB is not equal to zero, the computation for each computation unit is resumed from the beginning of the unit using a larger value of NONE or a smaller value of DELTA as explained in Section 4.2.3.

1. IPROB=1 identifies negative water depth which may occur at the computational waterline. Subr. 14 CRITV checks whether this problem occurs, and issues the following log message if it occurs.

```
      ICOUNT = 0
      DO 100 J = 1,S
         IF (U(2,J).LT.O.D+00) THEN
            ICOUNT = ICOUNT + 1
         ELSE
            C(J) = DSQRT(U(2,J))
         ENDIF
  100 CONTINUE
      IF (ICOUNT.GT.0) THEN
         IPROB=1
         CALL NSI (OMEGA,NONE)
         WRITE (*,9910) M,S,NONE,OMEGA,TIME,ICOUNT,ITRY,IREV
         WRITE (99,9910) M,S,NONE,OMEGA,TIME,ICOUNT,ITRY,IREV
      ENDIF
 9910 FORMAT (/' From Subr.  14 CRITV: IPROB=1'/
     + ' (Negative water depth)'/
     + ' Computation unit                M =',I8/
     + ' Waterline node                  S =',I8/
     + ' Computation parameters:      NONE =',I8/
     + '                             OMEGA =',F11.2/
     + ' Time of occurrence           TIME =',F18.9/
     + ' Number of occurrence       ICOUNT =',I8/
     + ' Indicators              ITRY,IREV =',I4,',',I3)
```

2. IPROB=2 identifies the seaward movement of the computational waterline in excess of one grid spacing over one time step, which violates the numerical stability criterion expressed by Eq. 27. Subr. 15 **MARCH** checks whether this problem occurs, and issues the following log message if it occurs.

```
      IF (U(2,SM1).LE.DELTA) THEN
         IPROB=2
         U2STOP=U(2,SM1)
         CALL NSI (OMEGA,NONE)
         WRITE (*,9910) M,S,NONE,OMEGA,DELTA,U2STOP,TIME,ITRY,IREV
         WRITE (99,9910) M,S,NONE,OMEGA,DELTA,U2STOP,TIME,ITRY,IREV
         RETURN
      ENDIF
 9910 FORMAT (/' From Subr.  15 MARCH: IPROB=2'/
     + ' (Water depth at (S-1) <or= DELTA)'/
     + ' Computation unit               M =',I8/
     + ' Waterline node                 S =',I8/
     + ' Computation parameters:     NONE =',I8/
     + '                            OMEGA =',F11.2/
     + '                            DELTA =',F18.9/
     + ' Water depth at (S-1)  U(2,S-1) =',F18.9/
     + ' Time of occurrence          TIME =',F18.9/
     + ' Indicators             ITRY,IREV =',I4,',',I3)
```

3. IPROB=3 identifies the occurrence of $|u_j| > (\Delta x/\Delta t)$ which implies the violation of the the numerical stability criterion given by Eq. 27. Subr. 11 **NUMSTA** checks whether this problem occurs, most likely at the computational waterline, and issues the following log message if it occurs.

```
      ICOUNT = 0
      DO 100 J = 1,S
         IF (DABS(V(J)).GT.XT) ICOUNT=ICOUNT+1
  100 CONTINUE
      IF (ICOUNT.GT.0) THEN
         IPROB=3
         CALL NSI (OMEGA,NONE)
         WRITE (*,9910) M,S,NONE,OMEGA,TIME,ICOUNT,ITRY,IREV
         WRITE (99,9910) M,S,NONE,OMEGA,TIME,ICOUNT,ITRY,IREV
      ENDIF
 9910 FORMAT (/' From Subr.  11 NUMSTA: IPROB=3'/
     + ' (Stability criterion violated)'/
     + ' Computation unit               M =',I8/
     + ' Waterline node                 S =',I8/
     + ' Computation parameters:     NONE =',I8/
     + '                            OMEGA =',F11.2/
     + ' Time of occurrence          TIME =',F18.9/
```

```
+ ' Number of occurrence       ICOUNT =',I8/
+ ' Indicators               ITRY,IREV =',I4,',',I3)
```

The variables involved in these messages are summarized below.

VARIABLE	DESCRIPTION
TIME,XT	Described in COMMON /CPAR4/ in Section 3.5
S	Described in COMMON /RUNP1/ in Section 3.5
U,V,C	Described in COMMON /HYDRO/ in Section 3.5
SM1	= (S-1)
U2STOP	= U(2,SM1)
M	= Computation unit
OMEGA	= Ω given by Eq. 28
NONE	Described in Section 4.2.1
DELTA	Described in Section 4.2.1
ITRY,IREV	Described in COMMON /DIAGID/ in Section 4.4
ICOUNT	Counter counting the number of occurence of numerical problems.

Subr. 25 NSI is called to calculate the numerical stability indicator OMEGA = Ω given by Eq. 28. The log messages, which are written both on terminal screen and in the file BMSG, can be used to assess the numerical difficulties experienced by the BEFORR2 computation.

4.2.3 ASSIGNMENT OF VALUES TO NONE AND DELTA

In a BEFORR2 computation, the value of the normalized nodal spacing Δx is determined at the beginning of the computation based on the value of INITS supplied by a user. The value of Δx is fixed during the entire computation duration. The value of NONE, which determines the size of the time step Δt, may vary from one computation unit to another. In the case of regular waves, the value of DELTA supplied by the user is used throughout the entire computation duration. In the case of IWAVE=2 and 3, on the other hand, the value of DELTA may be reduced in subsequent computation units as explained in the following.

It is the job of the computer program BEFORR2 to seek the values of NONE and DELTA that cause no numerical problem for each computation unit. The computation units are examined in sequence. In the first trial run for each computation unit, the values of the pair (NONE,DELTA) are assumed in a certain manner. If there is no numerical problem encountered, the assumed values are adopted. Otherwise, additional trial runs are made by increasing the value of NONE or reducing the value of DELTA incrementally until there is no numerical problem for the computation unit under examination. The number of trial runs for each computation unit is counted by the counter ITRY.

Subr. 10 CPARAM determines the values of NONE and DELTA in each trial run for each computation unit according to the following guidelines:

1. For the first trial run for the first computation unit, use NONE=NONEM and DELTA as specified

in the primary input data file.

2. For the second or subsequent trial run because of IPROB = 1, 2, or 3 in the previous trial run for any computation unit:

   (a) For regular wave computations, increase NONE by about 50%. DELTA is not modified.

   (b) For irregular wave computations with IWAVE=2 and 3:

      (b1) If IPROB=2 and $h_{s-1} > 0$: decrease DELTA by about 5%. NONE is not modified.

      (b2) Otherwise, increase NONE by about 50%. DELTA is not modified.

3. For the first trial run for the second and subsequent computation units:

   (a) For regular wave computations, use the value of NONE that has been assigned to the previous computation unit. DELTA is never modified.

   (b) For irregular wave computations: use the value of DELTA that has been assigned to the previous computation unit and choose NONE as follows:

$$\text{NONE}=(\text{IREV+1})\times\text{NONEM} \tag{92}$$

   where the reverse indicator IREV=0, unless IPROB=9 as explained in the following.

4. It can happen that NONE has been increased several times but numerical problems still occur. There is an upper limit for the value of NONE. The limit is set in terms of the numerical stability indicator Ω given by Eq. 28 which implies that Ω increases with NONE. Whenever $\Omega > 20$, IPROB=9 is set, and additional measures are taken as explained in Section 4.2.4.

5. The following requirements are always imposed:

   (a) NONE is divisible by NRATE.

   (b) DELTA is a multiple (10^{-6}).

   (c) Each trial run starts at the beginning of a computation unit using the stored hydrodynamic quantities at the end of the previous computation unit (LEVEL=1 as explained below).

It should be stated that the above guidelines are based on the experiences gained from various computations made using the computer programs BEFORR2-RBREAK2. The values of "50%" (increase of NONE), "5%" (decrease of DELTA), and "20" (the upper limit on Ω) could be changed according to the experiences of the user.

4.2.4 Computation Reversal and Storage Time Levels

IPROB=9 indicates that NONE has become so large that the corresponding value of the numerical stability indicator Ω is greater than 20 but numerical problems still exist. Further increases of the value of NONE will usually be ineffective. For *regular* wave computations, this is the end of a BEFORR2 computation. The following error message, written in Subr. 10 CPARAM, will appear on screen as well as in the file BMSG.

```
OMEGA is too large.  Computation aborted.
Suggested to input larger DELTA.
```

Another attempt may be carried out using a larger initial value of DELTA as explained in Section 4.2.1.

Experiences suggest that for the case of *irregular* waves, the problem IPROB=9 may still be overcome by reversing the computation to a certain time level, setting the counter IREV=IREV+1 and resume the computation. The increased value of IREV will result in the increased value of NONE for the first trial run for the subsequent computation units because of Eq. 92. This computation reversal requires the storage of the hydrodynamic quantities including DELTA at past time levels. To minimize memory requirements and simplify the computational procedures, hydrodynamic quantities are stored at the following three time levels.

- LEVEL=1: the end the *previous* computation unit corresponding to the unit (M-1) where the integer M indicates the present computation unit.
- LEVEL=2: the beginning of the *present* set of five computation units indicated by the integer MM where the computation units are grouped into sets of five units. The last set may contain less than five units. The total number of sets is denoted by MFIVE.
- LEVEL=3: the beginning of the *previous* set of five computation units corresponding to the set (MM-1).

The hydrodynamic quantities at LEVEL=1, LEVEL=2, and LEVEL=3 are stored in the following variables with the index k = 1, 2, and 3, respectively.

- SSAV(k) stores S.
- JMSAV(k) stores JMAX.
- DELSAV(k) stores DELTA.
- VSAV(k,j) stores V(j) at the node j.
- ESAV(k,j) stores ELEV(j) at the node j.
- USAV(2k-1,j) stores U(1,j) at the node j.
- USAV(2k,j) stores U(2,j) at the node j.

The stored variables are explained in the following sections:

VARIABLE	DESCRIPTION
DELTA	Described in COMMON /CPAR4/ in Section 3.5
U,V,ELEV	Described in COMMON /HYDRO/ in Section 3.5
S	Described in COMMON /RUNP1/ in Section 3.5
JMAX	Described in COMMON /RUNP1/ in Section 4.4

These values are stored using Subr. 29 **SAVEM** for LEVEL=1 and Subr. 30 **SAVEMM** for LEVEL=2 and LEVEL=3. On the other hand, the stored values are recalled by Subr. 27 **INITM** for LEVEL=1 and Subr. 28 **INITMM** for LEVEL=2 and LEVEL=3.

It should be noted that the storing and recalling procedures for LEVEL=1 are necessary and used for trial runs for each computation unit to overcome the numerical problems of IPROB = 1, 2, and 3 discussed in Section 4.2.3.

For the case of IPROB=9, the following measures based on LEVEL=2 and LEVEL=3 are taken:

1. Increase IREV by one. The counter IREV indicates the number of reversals made because of IPROB=9. IREV also increases the value of **NONE** given by Eq. 92 for the first trial run for the subsequent computation units.
2. If any of the following three conditions is met, reverse the computation to LEVEL=2.
   (a) **MM**=1, implying that the computation is in the *first* set of five computation units and there is no storage at LEVEL=3 yet.
   (b) ILEV3=1, which indicates that the reversal to LEVEL=3 has already been made and there is no storage at new LEVEL=3 yet.
   (c) The time interval between the end of the present computation unit and the time level of LEVEL=2 is greater than 2 wave periods.
3. Otherwise, reverse the computation to LEVEL=3, set ILEV3=1, and substract one from the index **MM**.
4. Set initial hydrodynamic quantities and **DELTA** using Subr. 28 **INITMM**, which differentiate LEVEL=2 and LEVEL=3.
5. Set IPROB=0 and march the computation forward from the designated level of LEVEL=2 or LEVEL=3.
6. If the computation reaches the end of the present set of five computation units without IPROB=9, set IREV=0 and ILEV3=0, which indicate that the problem of IPROB=9 is over and the computation proceeds from one computation unit to the next.
7. If the reversals are made five times (IREV=5), the computation is terminated with the following error message:

```
Computation aborted after IREV reached 5.
Suggested to input larger DELTA.
Programmed stop.
```

• 4.3 •

SUBROUTINES AND FUNCTION

The computer program BEFORR2 consists of 41 program units: the main program, 39 subroutines, and one function. Table 4 lists the subroutines and function arranged in numerical order in the computer program BEFORR2. The page numbers for the subroutines and function listed in Table 4 correspond to the page numbers in the BEFORR2 listing presented in Appendix C. Table 5 shows the interdependence among the 41 program units.

The computer programs BEFORR2 and RBREAK2 share many subroutine names. The subroutines with the same names do similar or identical tasks. The explanations given in Section 3.4 generally apply to the corresponding subroutines in BEFORR2. Explained in the following are only the subroutines that (1) do not exist in RBREAK2, or (2) are modified noticeably from the corresponding subroutines in RBREAK2. These subroutines are explained in the format: [Number] – [NAME] – [Description], where the [Number] refers to the numerical order in the computer program BEFORR2. The subroutines and function that are not listed below have already been explained in Section 3.4.

01 `OPENIO` opens input and output files.

02 `INPUT1` reads information from the primary input data file and performs checks on the majority of input data.

10 `CPARAM` calculates the minimum value of `NONE` *before* the time-marching computation. *During* the time-marching computation, this subroutine assigns appropriate values to the computation parameters `NONE` and `DELTA` for each computation unit following the procedures described in Section 4.2.3. `NONE` is the number of time steps in one wave period, whereas `DELTA` is the normalized water depth defining the computational waterline.

16 `LANDBC` manages the landward boundary conditions for wave runup, overtopping, or transmission.

26 `INIT` specifies the initial conditions given by $\eta = 0$ and $u = 0$ at $t = 0$ as well as the initial values for the variables saved at LEVEL=1 and LEVEL=2 as described in Section 4.2.4.

27 `INITM` sets the initial hydrodynamic quantities at the beginning of a computation unit using the saved values at LEVEL=1, that is, at the end of the preceding computation unit.

28 `INITMM` sets the initial hydrodynamic quantities at the beginning of one set of five computation units using the saved values at LEVEL=2 or LEVEL=3 as explained in Section 4.2.4.

29 `SAVEM` saves the hydrodynamic quantities at the end of a computation unit (LEVEL=1).

30 `SAVEMM` saves the hydrodynamic quantities at the end of one set of five computation units (LEVEL=2 and LEVEL=3).

31 `DOC1` documents the input data and dimensionless parameters before the time-marching computation in the file `BDOC`.

Table 4: List of 39 subroutines and one function in computer program BEFORR2.

No.	Subroutine (S) or Function (F)		Page No. in BEFORR2	No.	Subroutine (S) or Function (F)		Page No. in BEFORR2
01	S	OPENIO	C-8 - C-9	21	S	MATGJR	C-45
02	S	INPUT1	C-9 - C-17	22	S	MATS	C-45 - C-46
03	S	INPUT2	C-17 - C-18	23	S	MATD	C-46 - C-47
04	S	BOTTOM	C-18 - C-21	24	S	MATU	C-47 - C-48
05	S	WPARAM	C-21 - C-23	25	S	NSI	C-48
06	S	REGWAV	C-23 - C-25	26	S	INIT	C-48 - C-49
07	S	FINDM	C-25 - C-26	27	S	INITM	C-50
08	F	CEL	C-26 - C-28	28	S	INITMM	C-50 - C-51
09	S	SNCNDN	C-28 - C-29	29	S	SAVEM	C-51 - C-52
10	S	CPARAM	C-29 - C-33	30	S	SAVEMM	C-52 - C-53
11	S	NUMSTA	C-33 - C-34	31	S	DOC1	C-53 - C-57
12	S	EXTRAP	C-34	32	S	DOC2	C-57 - C-58
13	S	RETAIN	C-34 - C-35	33	S	DOC3	C-58 - C-59
14	S	CRITV	C-35 - C-36	34	S	CKFPAR	C-59 - C-60
15	S	MARCH	C-36 - C-38	35	S	CKOPTI	C-60
16	S	LANDBC	C-38 - C-39	36	S	CKVAL	C-60 - C-61
17	S	RUNUP	C-39 - C-41	37	S	CKVAL1	C-61
18	S	OVERT	C-41 - C-42	38	S	CKVAL2	C-61 - C-62
19	S	SEABC	C-42 - C-44	39	S	CKSUBR	C-62 - C-63
20	S	MATAFG	C-44 - C-45	40	S	STOPP	C-63 - C-64

Table 5: Interdependence among the program units of computer program BEFORR2.

No.	Program Unit	Called from	Makes Call(s) to
00	MAIN	-	01, 02, 03, 04, 05, 06, 10, 11, 12, 13, 14, 15, 16, 19, 26, 27, 28, 29, 30, 31, 32, 33, 39, 40
01	OPENIO	00	-
02	INPUT1	00, 39	34, 35, 36, 37, 38, 40
03	INPUT2	00, 39	34, 37
04	BOTTOM	00, 39	34, 37, 40
05	WPARAM	00, 39	34
06	REGWAV	00, 39	07, 08, 09, 34
07	FINDM	06	08, 34
08	CEL	06, 07	-
09	SNCNDN	06	40
10	CPARAM	00	25, 34
11	NUMSTA	00, 39	25, 34
12	EXTRAP	00, 39	34
13	RETAIN	00, 39	34
14	CRITV	00, 39	25, 34
15	MARCH	00, 39	20, 21, 22, 23, 24, 25, 34
16	LANDBC	00, 39	17, 18, 34
17	RUNUP	16, 39	20, 21, 22, 34
18	OVERT	16, 39	34
19	SEABC	00, 39	34
20	MATAFG	15, 17, 39	34
21	MATGJR	15, 17, 39	34
22	MATS	15, 17, 39	34
23	MATD	15, 39	34
24	MATU	15, 39	34
25	NSI	10, 11, 14, 15, 25	-
26	INIT	00, 39	34
27	INITM	00, 39	34
28	INITMM	00, 39	34
29	SAVEM	00, 39	34
30	SAVEMM	00, 39	34

Continued on the next page.

Table 5 continued from the previous page.

No.	PROGRAM UNIT	CALLED FROM	MAKES CALL(S) TO
31	DOC1	00, 39	34
32	DOC2	00, 39	25, 34, 40
33	DOC3	00, 39	34
34	CKFPAR	All units except for 00, 01, 08, 09, 25, 34, 35, 36, 37, 38, 39, 40	-
35	CKOPTI	02	-
36	CKVAL	02	38
37	CKVAL1	02, 03, 04	-
38	CKVAL2	02, 36	-
39	CKSUBR	00	All units except for 00, 01, 07, 08, 09, 10, 25, 34, 35, 36, 37, 38, 39, 40
40	STOPP	00, 02, 04, 09, 32	-

32 DOC2 writes the computation log during the time-marching computation in the file BDOC as well as on screen.

33 DOC3 writes in the file BINPUT and completes the documentation in the file BDOC.

34 CKFPAR checks whether the values of the integers N1R, N2R, N3R, N4R, N5R, and N6R used to specify the size of matrices and vectors in the main program are equal to the values of the corresponding integers N1, N2, N3, N4, N5, and N6 used in the subroutines.

35 CKOPTI checks whether the options selected by a user are within the ranges available in the present form of BEFORR2-RBREAK2.

36 CKVAL checks some of the user-specified values with the assistance of Subr. 38 CKVAL2. These checks could have been a part of Subr. 02 INPUT1 but are separated because of possible limitations on some FORTRAN compilers in handling a large number of IF statements.

37 CKVAL1 checks whether some of the values selected by a user are within the ranges available in the present form of BEFORR2.

38 CKVAL2 facilitates the error count and reporting for Subr. 02 INPUT1 and 36 CKVAL.

39 CKSUBR calls subroutines which use any of the PARAMETERs N1, N2, N3, N4, N5, and N6 to check the values of the PARAMETERs being used. The check is actually performed by Subr. 34 CKFPAR. Calls made from Subr. 39 CKSUBR are identified with MODE=1 or ICALL=1.

40 STOPP executes a programmed stop if some of the input requirements for BEFORR2 are not satisfied.

• 4.4 •

PARAMETERS AND VARIABLES IN COMMON BLOCKS

The computer program BEFORR2 and RBREAK2 share many names of **COMMON** blocks, parameters, and variables. The **COMMON** blocks, parameters, and variables with the same names have similar or identical definitions.

Table 6 lists the names of all **COMMON** blocks included in the computer program BEFORR2. Also provided in that table is a brief remark on the contents of each block as compared to those of the counterpart in RBREAK2. The *counterpart* herein means **COMMON** block with the same name from the *other* program. The only parameters and variables added or modified in the **COMMON** blocks marked with an asterisk in Table 6 are explained in the following since the others have been explained in Section 3.5.

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

> **N6R** = **N6** = maximum number of computation units allowed. Presently **N6**=2000.

/CPAR4/ contains the computation parameters related to the finite difference method used herein.

> **DELTA** = normalized water depth defining the computational waterline. An *initial* value of **DELTA** is specified by a user in the primary input data file. This initial value may be reduced during the BEFORR2 computation to overcome numerical difficulties.

/RUNP1/ contains parameters and quantities related to wave runup.

> **JMAX** = largest value of **S** during the time interval $\boxed{0 < t \leq \mathtt{TMAX}}$ as compared to **JMAX** in RBREAK2 corresponding to the time interval $\boxed{\mathtt{MSTAT} < t \leq \mathtt{TMAX}}$.

/DIAGID/ contains the identifiers used in the self-adjusting mechanism.

> **INONE**=1 indicates that the regular incident wave profile needs to be re-computed at the beginning of a computation unit because of the increased value of **NONE**. **INONE**=0 indicates otherwise.
>
> **ITRY** counts the number of trial runs carried out for a computation unit.
>
> **IREV** indicates the number of computational reversals made to overcome **IPROB**=9 as explained in Section 4.2.4.
>
> **LEVEL** specifies the past time levels to which the computation is reversed.

/DIAGV1/ contains the integers used as indicator or to save certain quantities at specified past time levels or computation units.

> **SSAV** stores the values of the waterline node **S** at the specified past time levels.
>
> **JMSAV** stores the values of **JMAX** at the specified past time levels.
>
> **NONESV(m)** = the value of **NONE** assigned to the computation unit **m**, where the index **m**=1,2,...,**MWAVE** and **MWAVE** is the number of computation units.

Table 6: List of COMMON blocks in computer program BEFORR2.

No.	Name of COMMON Block		Contents Compared to Those of Its RBREAK2 Counterpart
1	/DIMENS/	*	Same except for N6R added in BEFORR2
2	/CONSTA/		Same
3	/ID/		Same except for IENERG deleted in BEFORR2
4	/CPAR1/		Same
5	/CPAR2/		Same except for NONM, NHOP deleted in BEFORR2
6	/CPAR3/		Same
7	/CPAR4/	*	Same except for DELTA modified in BEFORR2
8	/REQ1/		Same
9	/REQ2/		Same
10	/WAVE1/		Same
11	/WAVE2/		Same
12	/WAVE3/		Same except for ETAIS,ETARS,ETATS deleted in BEFORR2
13	/WAVE4/		Same
14	/WAVE5/		Same
15	/WAVE6/		Same except for NDAT deleted in BEFORR2
16	/BOT1/		Same
17	/BOT2/		Same
18	/BOT3/		Same
19	/BOT4/		Same
20	/BOT5/		Same
21	/HYDRO/		Same
22	/MATRIX/		Same
23	/RUNP1/	*	Same except for JMAX modified in BEFORR2
34	/RUNP2/		Same except for RUNUPS,RSTAT,RMEAN deleted in BEFORR2
24	/ARMOR/		Same as /ARMOR1/ of RBREAK2
25	/SAVBC/		Same
26	/VALUEN/		Same
27	/DIAGID/	*	Added in BEFORR2
29	/DIAGV1/	*	Added in BEFORR2
30	/DIAGV2/	*	Added in BEFORR2

MFIVE = number of sets of five computation units.

/DIAGV2/ contains the double precision variables used to save certain quantities at specified past time levels or computation units.

USAV stores the spatial variations of m and h at the specified past time levels.

VSAV stores the spatial variations of $u = m/h$ at the specified past time levels.

ESAV stores the spatial variations of η at the specified past time levels.

DELSAV stores the values of DELTA at the specified past time levels.

DELTSV(m) = the value of DELTA assigned to the computation unit m, where the index m=1,2,...,MWAVE and MWAVE is the number of computation units.

• 4.5 •

Warning and Error Messages

The only one warning message that the computer program BEFORR2 may issue is the same as that of RBREAK2 described at the beginning of Section 3.7. The first eleven error messages related to the dimensions of the vectors and matrices in Section 3.7 are also included in BEFORR2.

The requirements of input data which are checked by the computer program BEFORR2 are grouped and presented in the following. Any error message will result in the termination of computation.

Group 1: Computation parameters MSTAT, MSAVA1, and MSAVA2 explained in COMMONs /CPAR1/ and /CPAR4/ in Section 3.5.

Requirement	Error Message Issued If Violated
0≤MSTAT≤(TMAX-1)	Input Error: MSTAT = ··· Specify MSTAT in the range of [0,···]
1≤MSAVA2≤TMAX	Input Error: MSAVA2 = ··· Specify MSAVA2 in the range of [1,···]
1≤MSAVA1≤MSAVA2	Input Error: MSAVA1 = ··· Specify MSAVA1 in the range of [···,···]

GROUP 2: Input data values that must be non-negative.

REQUIREMENT	ERROR MESSAGE ISSUED IF VIOLATED	REFER TO SECTION 3.5 EXCEPT FOR MREP
XBSEG(i)≥0	XBSEG must be non-negative.	COMMON /BOT5/
CL≥0	CL must be non-negative.	COMMON /ARMOR1/
CM≥0	CM must be non-negative.	COMMON /ARMOR1/
X1≥0	X1 must be non-negative.	COMMON /CPAR4/
X2≥0	X2 must be non-negative.	COMMON /CPAR4/
MREP≥0	MREP must be positive or zero.	Sections 4.1 & 3.3.1

GROUP 3: Input data values that must be positive.

REQUIREMENT	ERROR MESSAGE ISSUED IF VIOLATED	REFER TO SECTION 3.5 EXCEPT FOR DELTA
DELRP>0	DELRP must be positive.	COMMON /RUNP2/
TREQ>0	TREQ must be positive.	COMMON /REQ2/
WBSEG>0	WBSEG must be positive.	COMMON /BOT5/
C2>0	C2 must be positive.	COMMON /ARMOR1/
C3>0	C3 must be positive.	COMMON /ARMOR1/
SG>0	SG must be positive.	COMMON /ARMOR1/
CD>0	CD must be positive.	COMMON /ARMOR1/
TANPHI>0	TANPHI must be positive.	COMMON /ARMOR1/
DAP>0	DAP must be positive.	COMMON /ARMOR1/
DELTA>0	DELTA must be positive.	Section 4.2.1
TMAX>0	TMAX must be positive.	COMMON /CPAR4/
HREFP>0	HREFP must be positive.	COMMON /WAVE1/
TP>0	TP must be positive.	COMMON /WAVE1/
KSREF>0	KSREF must be positive.	COMMON /WAVE2/
KSSEA>0	KSSEA must be positive.	COMMON /WAVE2/
DSEAP>0	DSEAP must be positive.	COMMON /BOT1/
TSLOPS>0	TSLOPS must be positive.	COMMON /BOT2/

GROUP 4: Input data integers that must be at least unity where these parameters are explained in COMMONs /CPAR1/ and /CPAR2/ in Section 3.5.

REQUIREMENT	ERROR MESSAGE ISSUED IF VIOLATED
NTIMES≥1	NTIMES must be at least unity.
MULTIF≥1	MULTIF must be at least unity.
NRATE≥1	NRATE must be at least unity.

GROUP 5: Miscellaneous checks.

IF THIS HAPPENS	THIS ERROR MESSAGE WILL BE ISSUED
IWAVE=1 but TMAX not a whole number	TMAX must be a whole number for IWAVE=1. Programmed stop.
IREQ=1 but none of IELEV,IV,IDUDT,ISNR is unity	Special storing requested, but pertinent identifiers not specified correctly. Check identifiers IREQ,IELEV,IV,IDUDT,ISNR. Programmed stop.
TREQ(i)>TMAX	TREQ can not exceed TMAX. Programmed stop.
IJOB<3 but submerged structure specified	SWL is always above the structure. RUNUP/OVERTOPPING computation can not be performed. Programmed stop.
IJOB=3 but subaerial structure specified	Part of the structure is above SWL. TRANSMISSION computation can not be performed. Programmed stop.
Subr. SNCNDN fails	Failure in Subr. 09 SNCNDN. Programmed stop.
Function CEL fails	Failure in Function 08 CEL. Programmed stop.

A couple of error messages related to the self-adjusting mechanism have been discussed in Section 4.2.4.

It is possible that during the runup computation, the slope geometry specified as input may not be long enough to facilitate the movement of the computational waterline. When this happens, the following message is issued from Subr. 16 LANDBC:

```
From Subr. 16 LANDBC:
t = ···; S = ···; End Node = ···
Slope is not long enough to accomodate shoreline movement
Specify longer slope or choose overtopping computation
```

It is also possible that during the time-marching computation for regular waves, the dimension N2 may become too small to accomodate the computation of the regular wave profile. The following error message is issued from Subr. 06 REGWAV:

```
Specify larger dimension N2 to facilitate computation
of incident regular wave profile at seaward boundary.
N2 must be greater than NONE.
Currently, N2 = ···
Current try requires NONE = ···
Time of occurrence TIME = ···
Provide enough space for possible increase in NONE.
Computation aborted.
```

The measured free surface oscillation at $x = 0$ is specified as input for IWAVE=3. The seaward boundary algorithm for IWAVE=3 described in Section 2.5.2 uses Eq. 42 to compute u_1^* and may not be appropriate if the absolute value of the denominator on the right hand side of Eq. 42 becomes almost zero. The computation will stop if this absolute value becomes less than 0.0001 and the following error message is issued from Subr. 19 SEABC:

```
From Subr. 19 SEABC: IWAVE=3
(Denominator for V(1) is almost zero)
Computation unit M = ···
Time of occurrence TIME = ...
Value of denominator DENOM = ...
```

Furthermore, the seaward boundary algorithm is based on the assumption of subcritical flow at $x = 0$ for which $u_1^* < c_1^*$. If $u_1^* \geq c_1^*$, the following error message is written from Subr. 19 SEABC:

```
From Subr. 19 SEABC: Seaward Boundary
(Flow at x = 0 is not subcritical)
Computation unit M = ···
Time of occurrence TIME = ···
Water velocity at x = 0  V(1) = ···
Phase velocity at x = 0  c(1) = ···
```

These error messages have never been issued for the computations made so far using RBREAK2. It is noted that the assumption of subcritical flow at $x = 0$ is also employed for the seaward boundary algorithm for IWAVE=1 and 2 described in Section 2.5.1. The error message for IWAVE=1 and 2 has been deleted since the subcritical flow assumption has never been violated for the previous computations using IBREAK and RBREAK.

• 4.6 •

OUTPUT PRODUCED BY BEFORR2

The names of the four files containing the output from the computer program BEFORR2 start with the letter "B" for easy identification. The four output files are BDOC, BMSG, BSPACE, and BINPUT.

File BDOC contains (1) a summary of input data, written in the format similar to that of the file ODOC produced by the computer program RBREAK2 and (2) computation log written by Subr. 32 DOC2.

File BMSG contains the computation log described in Section 4.2.2 as well as the warning and error messages explained in Section 4.5.

File BSPACE stores the normalized structure geometry written in Subr. 31 DOC1 as follows:

```
     WRITE (52,9000) JE
     WRITE (52,8000) (XB(J),ZB(J),J=1,JE)
8000 FORMAT (5D15.6)
9000 FORMAT (I8)
```

The variables involved are explained in Section 3.5.

VARIABLE	DESCRIPTION IN SECTION 3.5
JE	COMMON /CPAR3/
XB,ZB	COMMON /BOT3/

File BINPUT contains the following:

1. Header copied from the primary input data file and written in Subr. 02 INPUT1 as follows:

```
     WRITE (51,1110) NLINES
     DO 110 L = 1,NLINES
        WRITE (51,1120) (COMMEN(I),I=1,15)
 110 CONTINUE
1110 FORMAT (3I8)
1120 FORMAT (15A5)
```

2. Case-signatures consisting of MWAVE, FINP2, NDATA, and NONEM, written in Subr. 31 DOC1 in this order:

```
      WRITE (51,9000) MWAVE
      IF (IWAVE.GT.1) THEN
         WRITE (51,5000) FINP2
         WRITE (51,9000) NDATA
      ENDIF
      WRITE (51,9000) NONEM
 5000 FORMAT (A10)
 9000 FORMAT (I8)
```

   These *case-signatures* are to be compared to their counterparts in the subsequent RBREAK2 computation. It is deemed necessary to do this comparison since any BEFORR2 computation produces the file BINPUT. The file BINPUT used in the subsequent RBREAK2 computation should be that produced by the preceding BEFORR2 computation.

3. Maximum value of NONE for regular waves, which is the value of NONE for the last computation unit, written by Subr. 33 DOC3 as follows:

```
      IF (IWAVE.EQ.1) WRITE (51,9000) NONESV(MWAVE)
 9000 FORMAT (I8)
```

   This value will be used to check the value of PARAMETER N2 used in the computer program RBREAK2 for the subsequent computation.

4. Computation parameters to be used in the subsequent RBREAK2 computation. These parameters are written by Subr. 33 DOC3 in the following order:

```
      DO 100 M = 1,MWAVE-1
         WRITE (51,5110) M,NONESV(M),NONESV(M),DELTSV(M)
  100 CONTINUE
      WRITE (51,5110) MWAVE,NONESV(MWAVE),NEND,DELTSV(MWAVE)
 5110 FORMAT (3I8,D18.9)
```

   The last MWAVE lines of the file BINPUT correspond to the MWAVE computation units executed by BEFORR2. Each of the MWAVE lines contains the following: (1) Sequential number M, (2) The value of NONE assigned to the computation unit M, (3) The number of time steps in the computation unit M, and (4) The value of DELTA assigned to the computation unit M. The M-th line of the MWAVE lines will be read at the beginning of the computation unit M in the subsequent RBREAK2 computation.

The variables used above have been explained before as shown below.

VARIABLE	DESCRIPTION
NLINES,COMMEN	Section 3.6.2
FINP2	Name of the file containing the input wave train
NDATA	COMMON /WAVE6/ in Section 3.5
MWAVE	COMMON /CPAR1/ in Section 3.5
NONEM	COMMON /CPAR2/ in Section 3.5
NONESV,DELTSV	COMMONs /DIAGV1/ and /DIAGV2/ in Section 4.4
M	= M-th computation unit

Part V
Irregular Wave Overtopping over Smooth and Rough Slopes in Surf Zones

• 5.1 •
Introduction

The computer program RBREAK2 has been developed by upgrading RBREAK incrementally out of necessity. Kobayashi and Raichle (1993, 1994) allowed the spatial variation of the bottom friction factor f' in Eq. 2 to compute irregular wave overtopping of a stone revetment (rough surface) in surf zones (smooth surface) for specified incident irregular wave trains immediately outside the breaker zone. The incident and reflected wave trains were separated using the procedure based on three wave gages and linear wave theory described by Kobayashi et al. (1990). The numerical model with the friction factors calibrated for the rough and smooth surfaces was shown to be fairly capable of predicting the average overtopping rate and the depth of overtopping flow on the crest of the revetment. The measured and predicted time series and spectra of the flow depth including entrained air showed the grouped nature of irregular wave overtopping events.

Cox et al. (1994) expanded RBREAK to allow the option of IWAVE=3 for specifying the measured total free surface elevation as input at the seaward boundary of the expanded numerical model. Computed free surface oscillations were compared with surf and swash measurements from the SUPERTANK Laboratory Data Collection Project where the free surface measurements appear to have been made without regard to the separation of incident and reflected waves. Comparisons of measured and computed results for regular and irregular waves showed that the time-dependent model could predict free surface oscillations including wave setup in the surf and swash zones, whereas conventional models based on the time-averaged momentum and wave energy equations could not predict swash dynamics.

The computer program RBREAK2 presented in this report allows the spatial variation of the bottom friction factor f' and the option of IWAVE=3 for specifying the measured total free surface elevation as input at its seaward boundary. In the following, RBREAK2 is compared with the irregular wave overtopping experiment conducted by Poff and Kobayashi (1993a). Kobayashi and Raichle (1993, 1994) employed the option of IWAVE=2 and specified the incident irregular wave trains as input to the computations made to compare RBREAK2 with their experiment. On the other hand, the experiment by Poff and Kobayashi (1993a) was performed such that the options of IWAVE=2 and 3 could be compared to assess whether one option is more accurate than the other. Furthermore, the separation of incident and reflected waves using Eqs. 12 and 13, which are based on the method of characteristics and linear long wave theory, is compared with that based on the three wave gage method along with linear wave theory in finite depth. These evaluations are necessary because the seaward boundary conditions for IWAVE=2 and 3 adopted in RBREAK2 are approximate with some uncertainties and the separation of incident and reflected waves using linear wave theory includes additional uncertainties.

• 5.2 •

EXPERIMENTS

Kobayashi and Raichle (1993, 1994) conducted 12 rough slope tests to investigate irregular wave overtopping over a revetment situated inside the surf zone because the effects of the mean sea level rise on such a structure may be significant. These tests have indicated that the probability and average rate of irregular wave overtopping are definitely affected by the incident wave spectral shape and wave grouping as well as by the crest height of the revetment above the still water level (SWL). Additional tests were conducted by Poff (1993) to investigate how the shallow water depth and beach profile change in front of 1:2 smooth and rough slopes affect irregular wave overtopping and reflection. These test results are discussed before the comparison of RBREAK2 with selected tests.

5.2.1 FIXED BEACH EXPERIMENT

The experiment by Poff (1993) was conducted in a wave tank which was 30 m long, 2.44 m wide, and 1.5 m high. Irregular wave trains were generated with a piston-type wave paddle. A gravel beach was located to absorb waves at the other end of the tank. A divider wall was constructed at a distance of 0.61 m from one of the side walls. A plywood beach, whose uniform slope was 1:23.5, was installed in front of a 1:2 impermeable smooth slope in the 0.61 m wide flume as shown in Fig. 5 for the 14 smooth slope tests performed by Poff (1993). A single layer of the quarry stone used by Kobayashi and Raichle (1993, 1994) was placed on the 1:2 smooth slope for the 10 rough slope tests. The toe of the 1:2 slope was 0.33 m above the horizontal bottom of the wave tank. The crest height of the 1:2 slope above its toe was 0.2 m. The width of the smooth crest was 0.13 m.

The symbols shown in Fig. 5, where the prime indicates the dimensional variables, are defined as follows: x' = horizontal coordinate taken to be positive landward with $x' = 0$ at the location of wave gage 1 in Fig. 5; z' = vertical coordinate taken to be positive upward with $z' = 0$ at SWL; d'_t = water depth below SWL at wave gage 1 which was 0.3 m in this experiment; d'_s = water depth below SWL at the toe of the 1:2 slope which was varied for the smooth and rough slope tests; and H'_c = crest height of the 1:2 slope above SWL given by $H'_c = (0.2 - d'_s)m$.

Fig. 5 also shows the locations of five capacitance wave gages used to measure the temporal variations of the free surface elevations at a sampling rate of 20 Hz. Wave gages 1, 2 and 3 at $x' = 0$, 0.11 and 0.44 m were used to separate the incident and reflected wave spectra and time series at $x' = 0$ as explained in Part VI. It is noted that wave gages 2 and 3 were placed landward of wave gage 1 for the purpose of comparison between this experiment and RBREAK2. Wave gage 4 measured the free surface elevation at the toe of the 1:2 slope. Wave gage 5 partially immersed in a gage well measured the depth of overtopping flow on the crest. The volume of overtopped water collected in a basin was measured. The average overtopping rate per unit length, $\overline{Q'}$, was obtained by dividing the measured volume by the crest length of 0.61 m and by the test duration that was 325 s for all the tests.

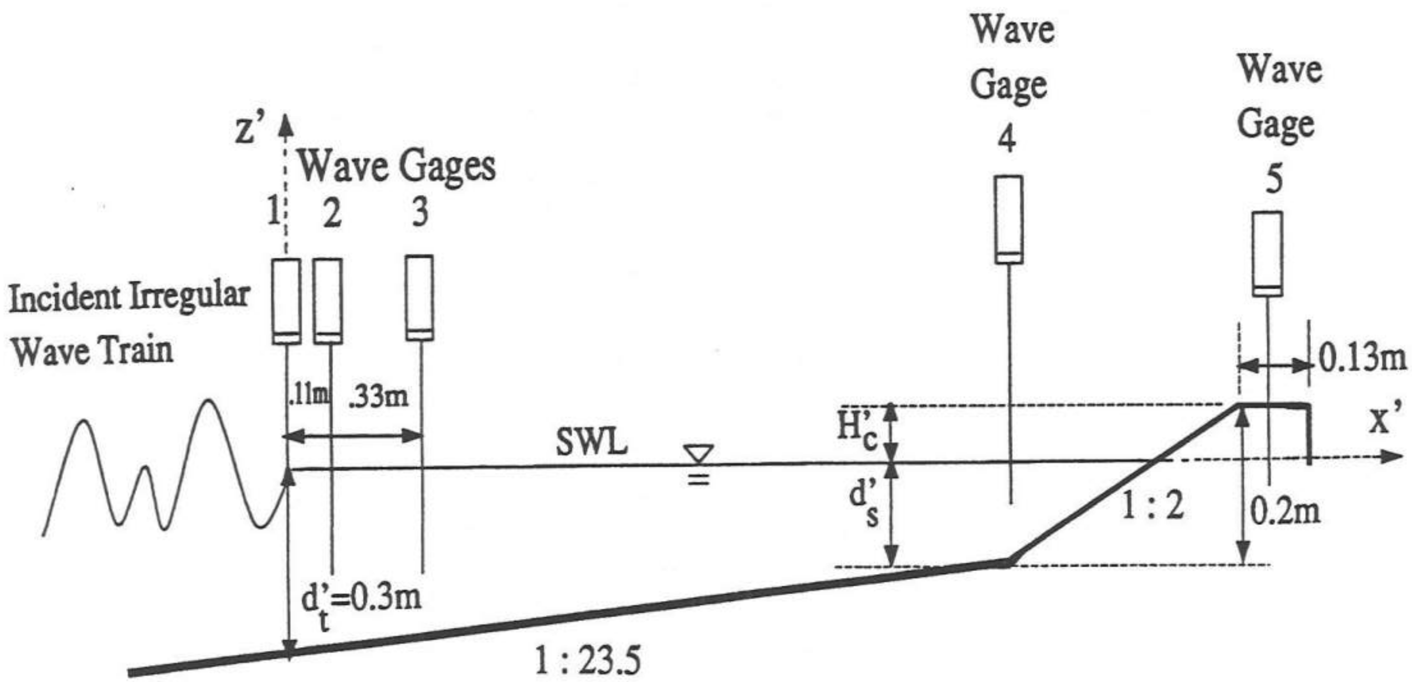

Figure 5: Experimental setup for smooth slope tests.

Irregular wave trains were generated in the deeper water depth $d'_p = (d'_s + 0.33)$m below SWL. An attempt was made to account for the effect of directional spreading on the incident irregular waves in the shallower water depth $d'_t = 0.3$ m immediately outside the breaker zone. The directional random wave spectrum in the deeper water depth d'_p was assumed to be given by the product of the TMA frequency spectrum (Bouws et al. 1985) and the Mitsuyasu-type directional spreading function (Goda 1985). The corresponding directional random wave spectrum in the shallower water depth d'_t was computed using linear finite-depth theory for a straight shoreline with parallel bottom contours (LeMéhauté and Wang 1982). An equivalent uni-directional frequency spectrum in the deeper water depth was then computed from the frequency spectrum in the shallower water depth using linear theory for shoaling of normally incident random waves as explained by Poff and Kobayashi (1993b). This equivalent uni-directional frequency spectrum was used for the irregular wave generation in this two-dimensional experiment to obtain the same frequency spectrum in the shallower water depth d'_t as that associated with the directional random waves. In this experiment, the difference between the deeper and shallower water depths was relatively small, and the effect of directional spreading on the incident irregular waves in the shallower depth turned out to be within the errors associated with the random wave generation based on linear wave theory.

The 24 smooth and rough slope tests were conducted by varying d'_s = 5–12 cm, H'_{mo} = 8.8–10.9 cm, Q_p = 2.6–6.5 and maintaining T'_p = 2 s where H'_{mo} = spectral estimate of the incident significant wave height, Q_p = spectral peakedness parameter (Goda 1985); and T'_p = spectral peak period. The 12 rough slope tests reported by Raichle and Kobayashi (1993) were also included in the data analysis of Poff and Kobayashi (1993a). The analyzed data has revealed the following:

- The average reflection coefficient $\bar{r}$ depends mostly on the slope roughness and the normalized toe depth $d_s = d'_s/H'_{mo}$ and is affected little by the normalized crest height $H_c = H'_c/H'_{mo}$ and the spectral shape represented by Q_p.
- The probability of wave overtopping, N_o/N_i, with N_o = number of overtopping events based on wave gage 5; and N_i = number of incident zero-upcrossing waves, is extremely sensitive (an order of magnitude) to both slope roughness and crest height, H'_c/d'_s, normalized by the toe depth and fairly sensitive (a factor of two) to the spectral shape.
- The average overtopping rate, $\overline{Q_s} = \overline{Q'}/(d'_s\sqrt{gd'_s})$, normalized by the toe depth d'_s for the depth-limited wave conditions is also extremely sensitive to the slope roughness and normalized crest height, H'_c/d'_s, and fairly sensitive to Q_p.
- Comparison between the time series and spectral characteristics measured by wave gages 4 and 5 shows that low frequency wave components are negligible seaward of the toe of the 1:2 slope but are appreciable on the crest due to the grouped nature of overtopping events.

5.2.2 Movable Beach Experiment

In order to examine the effects of beach profile changes on irregular wave overtopping for depth-limited wave conditions, an additional experiment was conducted by replacing the plywood beach in Fig. 5 with a beach consisting of well-sorted sand with a median diameter of 0.36 mm at an initial uniform 1:23.5 slope. Two cases corresponding to two of the 14 smooth slope test runs described in Section 5.2.1 were conducted on the 1:2 smooth slope fronted by the sand beach.

Table 7 summarizes the incident wave spectral and time series characteristics of the test runs with the 1:2 smooth slope fronted by the fixed and sand beaches. The first letters S and B associated with the run number in Table 7 indicate the fixed and movable beach test runs, respectively. The second letters N and W denote the narrow and wide spectra, respectively. The last number (1 to 7) for the fixed beach test runs is varied to indicate the change of the toe depth below SWL, d'_s, or the spectral estimate of the significant wave height, H'_{mo}, where the spectral peak period $T'_p = 2$ s for all the runs. The other symbols used in Table 7 are as follows: Q_p = spectral peakedness parameter (Goda 1985); κ = spectral correlated parameter (Battjes and Van Vledder 1984); N_i = number of incident zero-upcrossing waves; H'_s = significant wave height; T'_s = significant wave period; and $\bar{j}$ = mean run length of individual wave heights exceeding H'_s. The analysis of field data by Liu et al. (1993) indicated the increase of $\bar{j}$ with the increase of κ. The mean run length $\bar{j}$ in Table 7 tends to increase with the increase of Q_p and κ. For the narrow spectra, $5.4 \leq Q_p \leq 6.5$, $0.52 \leq \kappa \leq 0.61$ and $1.47 \leq \bar{j} \leq 2.29$. For the wide spectra, $2.7 \leq Q_p \leq 2.8$, $0.35 \leq \kappa \leq 0.40$, and $1.24 \leq \bar{j} \leq 1.58$. It should be stated that the values of κ tabulated by Poff and Kobayashi (1993a) are corrected in Table 7.

The specified wave paddle motions for the two cases BN and BW for the sand beach experiment in Table 7 corresponded to runs SN3 and SW3, respectively. Fifteen runs of the identical wave paddle motion for each of the two cases were conducted. The numeral next to BN and BW in Table 7 indicates the run number where the beach profile and wave gage measurements were made for runs 1, 3, 6, 10 and 15, whereas the average overtopping rate was measured for all the runs. The toe depth $d'_s = 7$ cm listed for these two cases is the toe depth on the initial 1:23.5 slope. The water depth d'_t at wave gage 1 fluctuated with beach profile changes slightly and the average value between the two sequential profile measurements was used to separate the incident and reflected waves. The measured wave conditions remained essentially the same during the beach profile evolution.

The average beach profile of the measured profiles along three lines using an ultrasonic profiler was presented by Poff (1993) for cases BN and BW with runs 1, 3, 6, 10 and 15. The upper panel of Fig. 6 shows the measured beach profile for run BW15 in comparison to the measured initial profile at the beginning of the 15 runs for case BW. Similar profile changes occurred for cases BN and BW including ripple formation, minor erosion seaward of the 1:2 smooth slope, and deposition of sand on the smooth slope. The measured profile changes were affected very little by the incident wave spectral shape. The sand deposition on the slope is consistent with the criterion of storm and normal beach profiles proposed by Dalrymple (1992). The well-sorted sand with $d_{50} = 0.36$ mm was too coarse to create a bar profile in this experiment. Consequently, an artificial bar was constructed in the wave tank and exposed to the irregular waves generated by the paddle motion corresponding to run SW3. This case indicated by the letters AB was run only once and the incident wave characteristics for run AB1 is listed at the bottom of Table 7. The beach profile after run AB1 together with the initial profile is shown in the lower panel of Fig. 6. It was hoped that the size of the bar was large enough to reduce irregular wave overtopping significantly.

Table 8 lists the following quantities for each run which are obtained in the same way as in Kobayashi and Raichle (1994): $\bar{r}$ = average reflection coefficient based on the zero moments of the reflected and incident wave spectra in water depth $d_t' = 0.3$ m; (N_o/N_i) = probability of wave overtopping estimated as the ratio between the number N_o of overtopping events recorded by wave gage 5 and the number N_i of incident zero-upcrossing waves; $\bar{Q}$ = averaged overtopping rate normalized as $\bar{Q} = \overline{Q'}/(H_{mo}'\sqrt{gH_{mo}'})$ with g = gravitational acceleration; and $\overline{Q_s} = \overline{Q'}/(d_s'\sqrt{gd_s'})$ = average overtopping rate normalized by the toe depth d_s'. It should be stated that the values of N_o/N_i tabulated by Poff and Kobayashi (1993a) are corrected in Table 8. The other dimensionless parameters listed in Table 8 are as follows: $H_c = (H_c'/H_{mo}')$ = normalized crest height above SWL; $d_s = (d_s'/H_{mo}')$ = normalized toe depth; $(L_p/d_s) = (L_p'/d_s')$ with $L_p = L_p'/H_{mo}'$ and L_p' = linear wavelength in the water depth d_s' based on the spectral peak period T_p'; and (H_c/d_s) = crest height normalized by the toe depth. For the test runs listed in Table 8, $0.56 \leq d_s \leq 1.03$ and the incident wave heights in front of the 1:2 slope were mostly depth-limited. For the depth-limited wave conditions, the plots of $\bar{r}$, (N_o/N_i) and $\bar{Q}$ against H_c do not show clear trends (Poff and Kobayashi 1993a). Furthermore, the waves in front of the 1:2 slope were essentially in shallow water because $20.6 \leq (L_p/d_s) \leq 27.8$. For the depth-limited waves in shallow water, the average overtopping rate $\overline{Q_s}$ normalized by the toe depth is more plausible than $\bar{Q}$ normalized by the incident wave height H_{mo}'. In the following, $\bar{r}$, (N_o/N_i) and $\overline{Q_s}$ are plotted in the same way as in Poff and Kobayashi (1993a) for the smooth and rough slope tests with the fixed beach.

Fig. 7 shows the approximately linear increase of $\bar{r}$ with the increase of d_s where the letters `SN`, `SW`, `BN`, `BW` and `AB1` correspond to those used in the run numbers in Tables 7 and 8. The increase of $\bar{r}$ with the increase of $d_s = d_s'/H_{mo}'$ is related to the wave energy dissipation seaward of the 1:2 slope that decreases as the toe depth d_s' increases relative to the incident wave height H_{mo}'. The normalized crest height H_c affects $\bar{r}$ little probably because the incident waves were reflected mostly from the 1:2 slope in the vicinity of SWL. The difference between the narrow and wide spectra is negligible possibly because the spectral shape and wave grouping may have been destroyed by wave breaking. The values of $\bar{r}$ for cases `BN` and `BW` as listed in Table 8 decreased only slightly as the beach profile evolved. Fig. 7 shows that the effect of the beach profile evolution is secondary in comparison to the normalized toe depth d_s.

Figs. 8 and 9 show the decrease of the wave overtopping probability, N_o/N_i and the normalized average overtopping rate, $\overline{Q_s}$, with the increase of the normalized crest height, H_c/d_s, respectively. The values of N_o/N_i and $\overline{Q_s}$ are larger (a factor of two or less) for the narrow spectra with grouped waves than for the wide spectra. The data points with $(H_c/d_s) \geq 2.33$ in Figs. 8 and 9 correspond to $d_s \leq 0.67$ in Table 8. For these data points, the differences between the narrow and wide spectra appear to be small probably because extensive wave breaking on the beach may have destroyed incident wave groups before they reached the 1:2 slope. The values of N_o/N_i and $\overline{Q_s}$ for cases `BN` and `BW` listed in Table 8 increased by a factor of about two as the beach profile evolved. Figs. 8 and 9 show that the effect of the beach profile evolution on wave overtopping is secondary in comparison to the crest height normalized by the toe depth. It is noted that the normalized results presented in Figs. 8 and 9 do not involve the incident wave height and period and are limited to the depth-limited wave conditions in shallow water.

The limited movable beach experimental results presented herein suggest that fixed beach experiments will yield essential information required for the design of revetments against irregular wave overtopping even for depth-limited wave conditions. Movable beach tests for the corresponding 1:2 rough slope were not conducted since the effects of beach profile changes

Table 7: Incident wave spectral and time series characteristics of test runs with 1:2 smooth slope fronted by fixed and sand beaches.

		Spectrum			Time Series			
RUN NO.	d'_s (cm)	H'_{mo} (cm)	Q_p	κ	N_i	H'_s (cm)	T'_s (sec)	$\bar{j}$
SN1	5	8.85	6.4	0.58	213	8.61	1.84	1.68
SW1	5	8.88	2.8	0.40	247	8.58	1.64	1.37
SN2	6	8.99	6.5	0.59	204	9.11	1.91	1.94
SW2	6	9.16	2.8	0.40	249	8.85	1.62	1.40
SN3	7	8.91	6.3	0.61	202	8.99	1.89	1.75
SW3	7	8.95	2.7	0.39	247	8.67	1.66	1.42
SN4	7	10.40	6.2	0.57	209	10.32	1.90	1.57
SW4	7	10.36	2.8	0.38	237	10.01	1.76	1.56
SN5	8	9.17	6.0	0.59	204	9.08	1.89	2.29
SW5	8	9.23	2.7	0.38	243	8.91	1.67	1.37
SN6	9	8.70	5.7	0.57	208	8.43	1.89	1.88
SW6	9	8.82	2.7	0.39	254	8.24	1.61	1.35
SN7	9	10.84	5.4	0.52	217	10.13	1.87	1.47
SW7	9	10.97	2.7	0.35	241	10.39	1.68	1.24
BN1	7	9.60	6.2	0.58	198	9.77	1.92	1.53
BN3	7	9.41	6.2	0.58	202	9.48	1.92	1.75
BN6	7	9.31	6.4	0.58	198	9.34	1.91	1.59
BN10	7	9.31	6.3	0.58	202	9.34	1.89	1.58
BN15	7	9.33	6.2	0.58	201	9.48	1.91	1.61
BW1	7	9.22	2.7	0.37	244	9.25	1.65	1.40
BW3	7	9.25	2.8	0.37	246	9.12	1.68	1.52
BW6	7	9.06	2.8	0.37	244	9.01	1.68	1.58
BW10	7	9.09	2.8	0.37	246	9.06	1.70	1.52
BW15	7	9.16	2.8	0.37	256	9.03	1.66	1.46
AB1	7	9.20	2.7	0.36	254	9.07	1.66	1.48

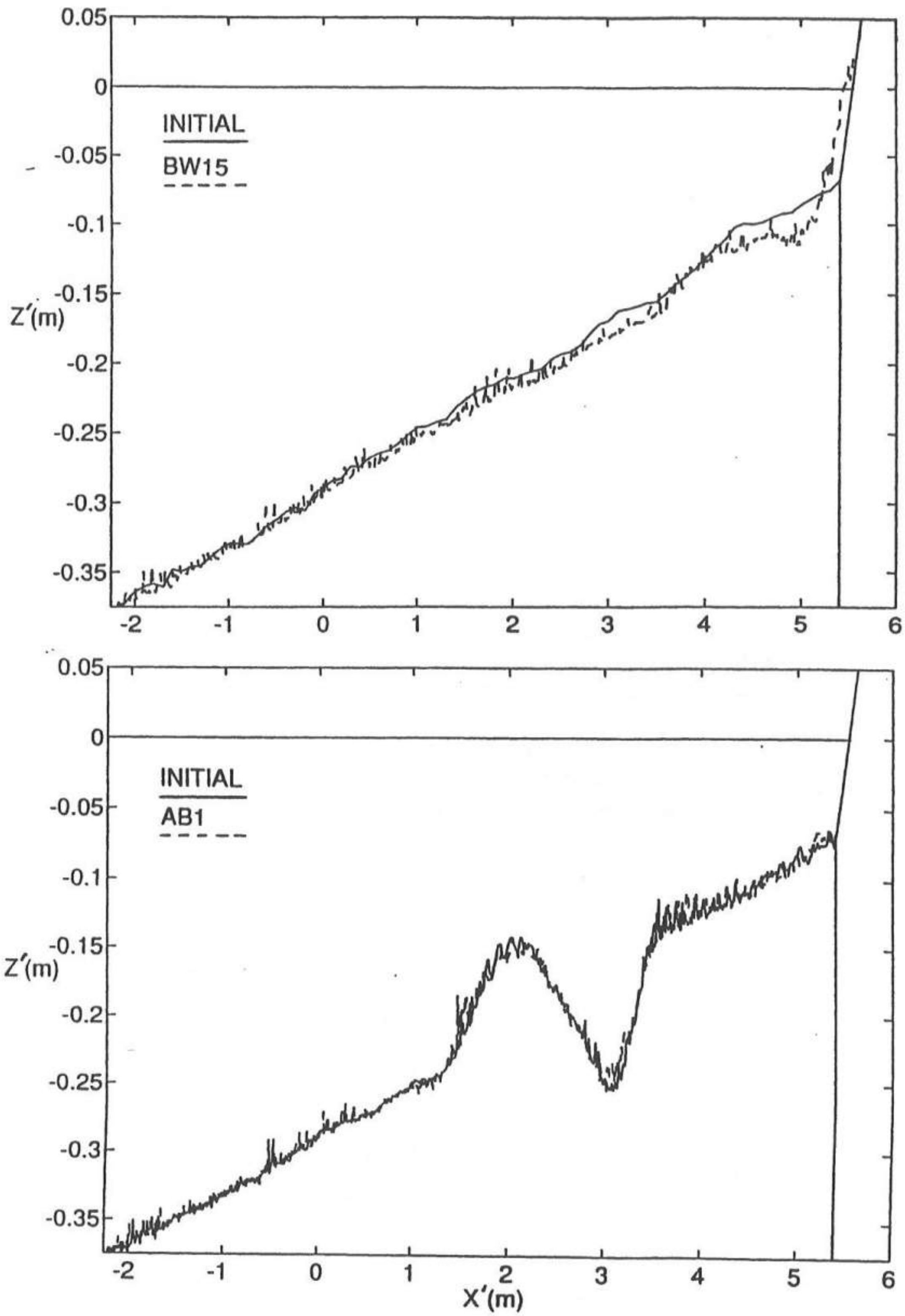

Figure 6: Measured beach profiles in front of 1:2 smooth slope for runs BW15 and AB1.

Table 8: Summary of analyzed data for test runs with 1:2 smooth slope fronted by fixed and sand beaches.

RUN NO.	H_c	d_s	$\frac{L_p}{d_s}$	$\frac{H_c}{d_s}$	$\overline{r}$	$\frac{N_o}{N_i}$	$\overline{Q}$ x 10^3	$\overline{Q_s}$ x 10^3
SN1	1.69	0.56	27.8	3.00	0.325	0.005	0.11	0.25
SW1	1.69	0.56	27.8	3.00	0.323	0.008	0.17	0.40
SN2	1.56	0.67	25.3	2.33	0.370	0.064	0.36	0.66
SW2	1.53	0.66	25.3	2.33	0.356	0.032	0.41	0.77
SN3	1.46	0.79	23.4	1.86	0.408	0.114	1.55	2.22
SW3	1.45	0.78	23.4	1.86	0.407	0.065	1.34	1.94
SN4	1.25	0.67	23.4	1.86	0.371	0.167	1.27	2.31
SW4	1.25	0.68	23.4	1.86	0.366	0.114	0.99	1.78
SN5	1.31	0.87	21.8	1.50	0.429	0.304	2.51	3.08
SW5	1.30	0.87	21.8	1.50	0.416	0.206	1.75	2.17
SN6	1.26	1.03	20.6	1.22	0.481	0.447	6.69	6.36
SW6	1.25	1.02	20.6	1.22	0.466	0.283	5.05	4.89
SN7	1.01	0.83	20.6	1.22	0.443	0.438	5.46	7.22
SW7	1.00	0.82	20.6	1.22	0.424	0.328	4.48	6.03
BN1	1.35	0.73	23.4	1.86	0.413	0.061	1.19	1.91
BN3	1.38	0.74	23.4	1.86	0.381	0.094	1.39	2.17
BN6	1.40	0.75	23.4	1.86	0.379	0.111	1.67	2.57
BN10	1.40	0.75	23.4	1.86	0.379	0.193	1.90	2.91
BN15	1.39	0.75	23.4	1.86	0.382	0.199	2.12	3.26
BW1	1.41	0.76	23.4	1.86	0.419	0.082	1.15	1.74
BW3	1.41	0.76	23.4	1.86	0.389	0.093	1.06	1.60
BW6	1.43	0.77	23.4	1.86	0.384	0.111	1.30	1.91
BW10	1.43	0.77	23.4	1.86	0.376	0.179	1.32	1.95
BW15	1.42	0.76	23.4	1.86	0.366	0.164	1.37	2.05
AB1	1.41	0.76	23.4	1.86	0.407	0.087	1.03	1.55

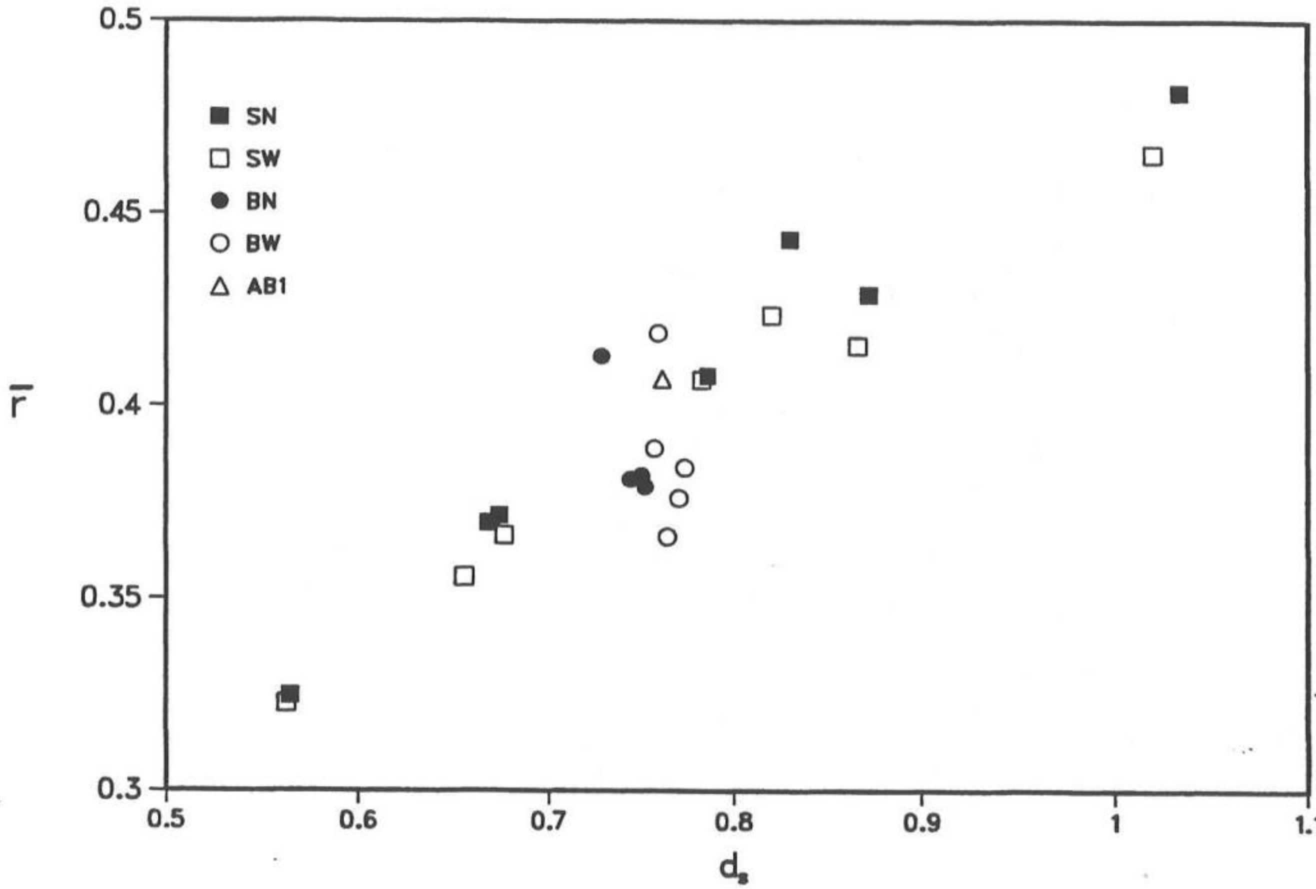

Figure 7: Average reflection coefficient $\bar{r}$ as a function of normalized toe depth $d_s = d'_s/H'_{mo}$ for 1:2 smooth slope fronted by fixed and sand beaches.

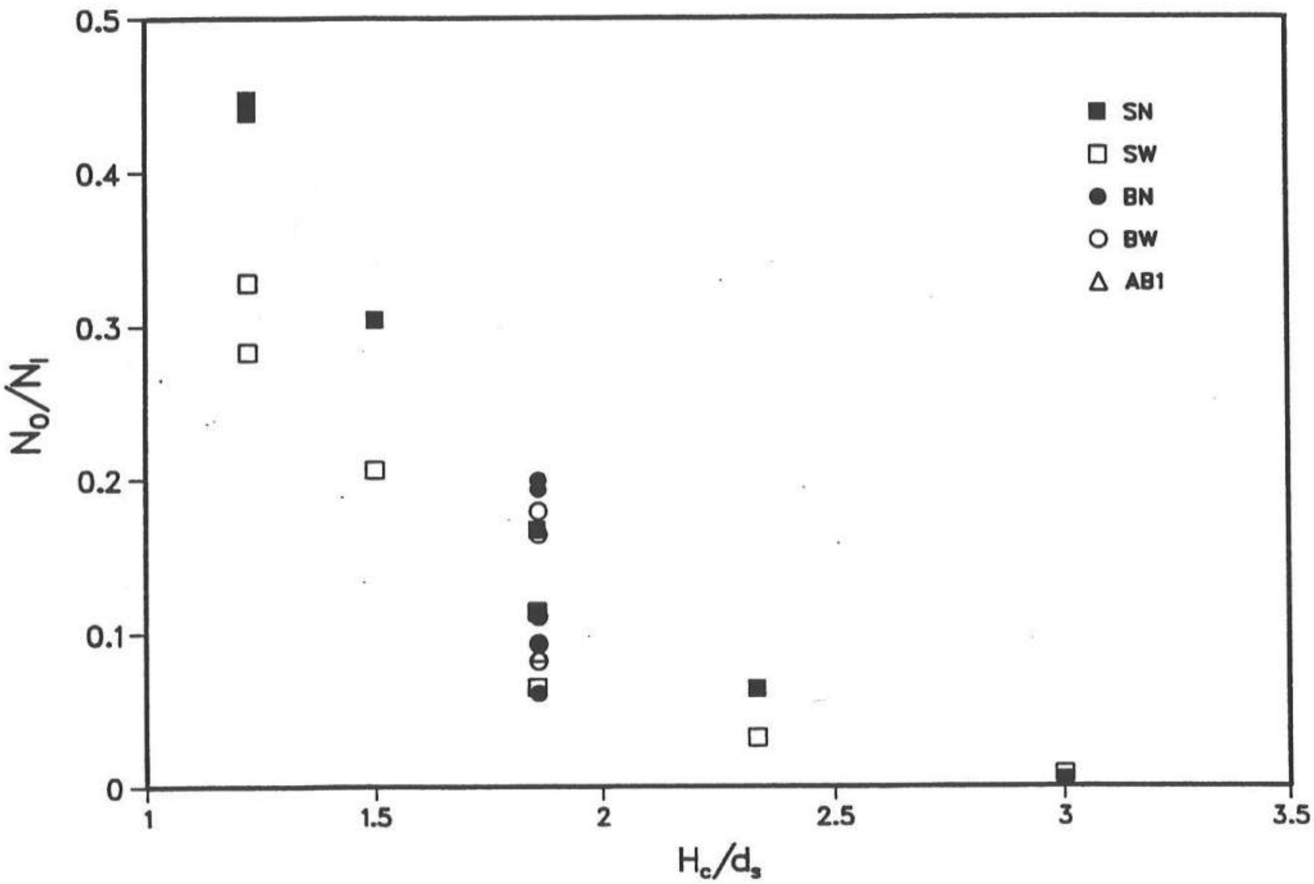

Figure 8: Wave overtopping probability, N_o/N_i, as a function of normalized crest height H_c/d_s for 1:2 smooth slope fronted by fixed and sand beaches.

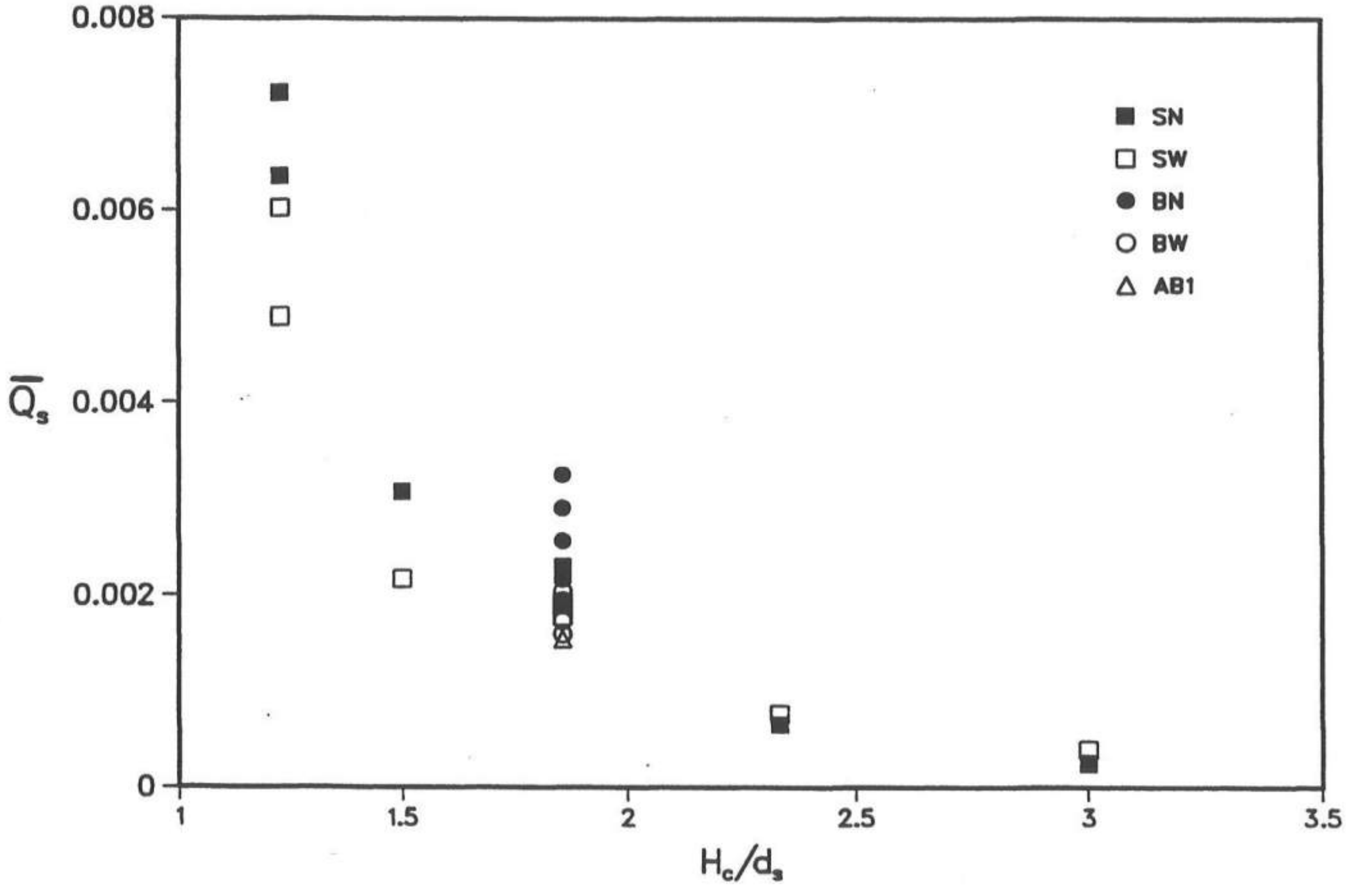

Figure 9: Average overtopping rate $\overline{Q_s}$ normalized by toe depth as a function of normalized crest height H_c/d_s for 1:2 smooth slope fronted by fixed and sand beaches.

on irregular wave overtopping over the 1:2 smooth slope have turned out to be smaller than expected.

• 5.3 •

COMPARISON OF RBREAK2 WITH ROUGH SLOPE TESTS

RBREAK2 is compared with the rough slope tests discussed in Section 5.2.1. For these tests, the 1:2 smooth slope shown in Fig. 5 was covered with a single layer of the quarry stone used by Kobayashi and Raichle (1993, 1994). As a result, the slope and roughness in the computation domain varied from the 1:23.5 smooth beach to the 1:2 rough slope with the smooth horizontal crest. Kobayashi and Raichle (1993, 1994) compared an earlier version of RBREAK2 with their rough slope tests where only the option of IWAVE=2 was considered and the measured incident wave train was based on the three wave gage method explained by Kobayashi et al. (1990). The following comparison also includes the option of IWAVE=3 for specifying the total free surface elevation measured by wave gage 1 in Fig. 5 as input to RBREAK2.

5.3.1 SUMMARY OF ROUGH SLOPE DATA

The 10 rough slope tests with the fixed smooth beach have not been included in Tables 7 and 8 for the 1:2 smooth slope test runs with the fixed and sand beaches. Table 9 lists the incident wave spectral and time series characteristics for the rough slope tests in the same manner as in Table 7. The four rows below each of runs RN3 and RW3 tabulates the corresponding computed results as will be explained in Section 5.3.3. The first letter R denotes the rough slope, whereas the second letters N and W indicate the narrow and wide spectra. In Table 9, d_s' = toe depth; H_{mo}' = spectral estimate of the significant wave height; Q_p = spectral peakedness parameter; κ = spectral correlated parameter; N_i = number of incident zero-upcrossing waves; H_s' = significant wave height; T_s' = significant wave period; and $\bar{j}$ = mean run length of individual wave heights exceeding H_s'. The spectral peak period $T_p' = 2$ s for all the runs. The values of κ listed by Poff and Kobayashi (1993a) are corrected in Table 9.

Table 10 summarizes the analyzed data for the 10 rough slope test runs in the same way as in Table 8. The computed results listed for runs RN3 and RW3 will be explained in Sections 5.3.4 and 5.3.6. In Table 10, $H_c = H_c'/H_{mo}'$ with H_c' = crest height of the 1:2 rough slope above SWL; $d_s = d_s'/H_{mo}'$; $L_p = L_p'/H_{mo}'$ with L_p' = linear wavelength in the water depth d_s' based on T_p'; $\bar{r}$ = average reflection coefficient; N_o/N_i = probability of wave overtopping with N_o = number of overtopping events recorded by wave gage 5; $\overline{Q_s} = \overline{Q'}/(d_s'\sqrt{gd_s'})$ normalized by d_s' for the depth-limited waves in shallow water. The values of N_o/N_i listed by Poff and Kobayashi (1993a) are corrected in Table 10.

5.3.2 COMPUTATIONS FOR RUNS RN3 AND RW3

Runs RN3 and RW3 are selected for evaluating the capabilities and limitations of RBREAK2 with the options of IWAVE=2 and IWAVE=3. For IWAVE=2, the incident wave train at the seaward boundary located at $x = 0$ is estimated using linear wave theory from the measured free surface oscillations at wave gages 1, 2 and 3 shown in Fig. 5 as will be explained in Section 6.1. The estimated incident wave train is specified as input to RBREAK2. The computed results for this option are indicated by the letter I placed immediately after the run number RN3 and RW3 in Tables 9 and 10. For IWAVE=3, the total free surface oscillation at $x = 0$ measured by wave gage 1 is specified directly as input to RBREAK2. The letter T placed after the run number RN3 and RW3 is used to indicate the computed results for this option.

Table 11 shows the primary input data file, FINP1, for run RN3-T. The input parameters and variables listed in Table 11 are explained in sequence where Section 3.6 has shown the input required for all the options included in RBREAK2. The number of comment lines proceeding the input data is 3 in Table 11.

- IJOB=2 for wave overtopping computation.
- ISTAB=0 for no computation of armor stability or movement where the quarry stone place on the 1:2 slope did not move in the rough slope experiment.
- ISYST=1 for the metric system used in the following.
- IBOT=1 for specifying the width and slope of each of the linear segments of the bottom geometry shown in Fig. 5.
- INONCT=1 to include the nonlinear correction term C_t in Eq. 13 where C_t is estimated by Eq. 43 for the seaward boundary located on the 1:23.5 fixed beach.
- IENERG=1 for computing the quantities involved in wave energy balance in Section 2.9.
- IWAVE=3 for specifying the measured total free surface oscillation at the seaward boundary as input to RBREAK2.
- MSTAT=0 to perform the statistical calculations from the first computation unit where the duration of transition from no wave action in the region $x > 0$ was only a small fraction of the test duration of 325 s.
- FINP2=rn3wg1 which is the file name containing the total free surface oscillation measured by wave gage 1 for run RN3 and normalized using the reference wave height and period as explained in Section 3.2.
- ISAVA=0 for no storage of the spatial variation of the normalized free surface elevation η and the normalized depth-averaged velocity u at specified time levels.
- MSAVA1=0 for ISAVA=0 where MSAVA1 specifies the first computation unit for storing the spatial variations of η and u if ISAVA=1.
- MSAVA2=0 for ISAVA=0 where MSAVA2 specifies the last computation unit for storing the spatial variations of η and u if ISAVA=1.
- NTIMES=0 for ISAVA=0 where NTIMES specifies the number of times at equal intervals per reference wave period for storing the spatial variations of η and u if ISAVA=1.
- ISAVB=1 for storing the temporal variations of the normalized volume flux m per unit width and the normalized water depth h at specified nodes.
- NNODB=5 where the temporal variations of m and h are stored at five nodes corresponding to the locations of wave gages 1–5 in Fig. 5.

Table 9: Incident wave spectral and time series characteristics of 10 test runs with 1:2 rough slope fronted by fixed smooth beach together with computed incident wave characteristics for runs RN3 and RW3.

RUN NO.	d'_s (cm)	Spectrum H'_{mo} (cm)	Q_p	κ	Time Series N_i	H'_s (cm)	T'_s (sec)	$\bar{j}$
RN1	9	9.33	5.9	0.56	207	9.02	1.90	1.88
RW1	9	9.59	2.6	0.37	252	9.10	1.59	1.41
RN2	9	10.84	5.6	0.52	211	10.26	1.89	1.43
RW2	9	10.93	2.7	0.33	241	10.39	1.68	1.26
RN3	10	9.03	5.7	0.56	218	8.69	1.87	1.67
RN3-T-MC	10	11.79	8.2	0.56	218	11.28	1.91	1.95
RN3-T-3G	10	11.59	8.6	0.58	213	11.25	1.91	1.94
RN3-I-MC	10	9.03	5.7	0.56	218	8.69	1.87	1.67
RN3-I-3G	10	9.09	5.3	0.54	236	8.81	1.87	1.70
RW3	10	9.24	2.7	0.36	256	8.73	1.61	1.48
RW3-T-MC	10	10.99	3.1	0.36	245	10.11	1.74	1.60
RW3-T-3G	10	10.80	3.2	0.34	245	9.98	1.67	1.59
RW3-I-MC	10	9.24	2.7	0.36	256	8.73	1.61	1.48
RW3-I-3G	10	9.34	2.6	0.40	284	8.76	1.56	1.46
RN4	11	9.21	5.8	0.56	219	9.06	1.87	2.12
RW4	11	9.28	2.7	0.38	246	8.93	1.70	1.54
RN5	12	9.20	6.1	0.57	214	9.17	1.83	2.31
RW5	12	9.19	2.8	0.38	252	8.77	1.65	1.41

Table 10: Summary of analyzed data for 10 test runs with 1:2 rough slope fronted by fixed smooth beach together with computed results for runs RN3 and RW3.

RUN NO.	H_c	d_s	$\frac{L_p}{d_s}$	$\frac{H_c}{d_s}$	$\overline{r}$	$\frac{N_o}{N_i}$	$\overline{Q} \times 10^3$	$\overline{Q_s} \times 10^3$
RN1	1.18	0.96	20.6	1.22	0.367	0.063	0.58	0.61
RW1	1.15	0.94	20.6	1.22	0.344	0.052	0.49	0.54
RN2	1.01	0.83	20.6	1.22	0.350	0.109	0.68	0.89
RW2	1.01	0.82	20.6	1.22	0.329	0.062	0.57	0.76
RN3	1.11	1.11	19.5	1.00	0.375	0.261	1.90	1.63
RN3-T-MC	0.85	0.85	19.5	1.00	0.347	0.216	2.09	2.68
RN3-T-3G	0.86	0.86	19.5	1.00	0.349	0.221	2.15	2.68
RN3-I-MC	1.11	1.11	19.5	1.00	0.382	0.092	0.92	0.79
RN3-I-3G	1.10	1.10	19.5	1.00	0.386	0.085	0.91	0.79
RW3	1.08	1.08	19.5	1.00	0.357	0.160	1.22	1.08
RW3-T-MC	0.91	0.91	19.5	1.00	0.379	0.171	1.96	2.26
RW3-T-3G	0.93	0.93	19.5	1.00	0.380	0.171	2.01	2.26
RW3-I-MC	1.08	1.08	19.5	1.00	0.377	0.055	0.82	0.73
RW3-I-3G	1.07	1.07	19.5	1.00	0.391	0.049	0.81	0.73
RN4	0.98	1.19	18.5	0.82	0.375	0.384	3.98	3.05
RW4	0.97	1.19	18.5	0.82	0.368	0.256	2.52	1.95
RN5	0.87	1.30	17.7	0.67	0.393	0.505	6.96	4.67
RW5	0.87	1.31	17.7	0.67	0.387	0.369	4.69	3.14

Table 11: Primary input data file, FINP1, for run RN3-T.

```
       3                           --> NLINES
-------------------------------------------------------------------------
RUN RN3: SPECIFY TOTAL WAVE
-------------------------------------------------------------------------
20                                 --> IJOB,ISTAB
1                                  --> ISYST
1                                  --> IBOT
1                                  --> INONCT
1                                  --> IENERG
3       0   rn3wg1                 --> IWAVE,MSTAT,FINP2
0       0        0        0        --> ISAVA,MSAVA1,MSAVA2,NTIMES
1       5                          --> ISAVB,NNODB
0       0                          --> ISAVC,NNODC
00000        0                     --> IREQ,IELEV,IV,IDUDT,ISNR,NNREQ
     490                           --> INITS
       1                           --> MULTIF
        .003000                    --> DELTA (NORMALIZED)
     162.500000                    --> TMAX  (NORMALIZED)
      40                           --> NRATE
       1                           --> NDELR
     1.000000                      --> DELRP(1)(MILLIMETERS)
     2.000000       2.000000       --> X1,X2
     0.090300       2.000000       --> HREFP(METERS),TP(SECONDS)
     1.000000       1.000000       --> KSREF,KSSEA
     0.300000                      --> DSEAP(METERS)
     0.500000                      --> TSLOPS
       3                           --> NBSEG
     4.700000      0.042500      0.005000   -->WBSEG(1),TBSLOP(1),FWP(1)
     0.400000      0.500000      0.010000   -->WBSEG(2),TBSLOP(2),FWP(2)
     0.130000      0.000000      0.005000   -->WBSEG(3),TBSLOP(3),FWP(3)
     1     12     45    471    517          -->NODB(1)-(2)-(3)-(4)-(5)
```

- ISAVC=0 for no storage of the temporal variation of the armor stability function or armor displacement.
- NNODC=0 for ISAVC=0 where NNODC specifies the number of nodes for storing the temporal variation of the armor stability function or armor displacement.
- IREQ=0 for no special storage.
- IELEV=0 for no special storage of the normalized free surface elevation η.
- IV=0 for no special storage of the normalized depth-averaged velocity u.
- IDUDT=0 for no special storage of the normalized horizontal fluid acceleration du/dt.
- ISNR=0 for no special storage of the armor stability function.
- NNREQ=0 for IREQ=0 where NNREQ specifies the number of the specified moments for the special storage if IREQ=1.
- INITS=490 where INITS specifies the number of the spatial nodes along the bottom below SWL and hence the dimensional nodal spacing $\Delta x'$. For the bottom geometry shown in Fig. 5 with $d_s' = 0.10$ m for run RN3 as listed in Table 9, the horizontal distance between the seaward boundary at wave gage 1 and the still waterline on the 1:2 slope is 4.90 m. The number of nodes below SWL is INITS=490 excluding the node on the 1:2 slope at SWL. Consequently, $\Delta x' = 0.01$ m which is sufficiently small relative to the reference wave height $H_r' = H_{mo}' = 0.0903$ m for run RN3.
- MULTIF=1 as recommended in Section 4.2.1 unless the computed results show unrealistic spikes and kinks.
- DELTA=0.003 which is the normalized water depth δ defining the computational waterline. The value of $\delta = 0.003$ is the upper limit of the range $\delta = 0.001$–0.003 suggested on the basis of our previous computations. The increase of δ tends to improve numerical stability near the moving waterline.
- TMAX=162.5 which is the normalized duration of the computation obtained by dividing the measurement duration 325 s by the reference wave period $T_r' = T_p' = 2$ s.
- NRATE=40 for storing output time series at the rate of 40 points per reference wave period, corresponding to the sampling rate of 20 Hz used in the experiment.
- NDELR=1 where NDELR specifies the number of different values of the physical water depth δ_r' used to compute wave runup as explained in Section 2.6.
- DELRP(1)=1 mm for ISYST=1 corresponding to $\delta_r' = 1$ mm for NDELR=1 where the value of δ_r' should correspond to the elevation of a runup meter but wave runup was not measured in the experiment.
- X1=2.0 for the numerical damping coefficient $\epsilon_1 = 2.0$ introduced in Eqs. 24 and 25.
- X2=2.0 for the numerical damping coefficient $\epsilon_2 = 2.0$ introduced in Eqs. 24 and 25.

- HREFP=0.0903 m where the reference wave height H'_r used for the normalization in Eqs. 3–5 is taken as the measured spectral estimate of the incident significant wave height, H'_{mo} = 0.0903 m, for run RN3 as listed in Table 9. It is noted that the computed incident wave is unknown before the computation for IWAVE=3.
- TP=2 s where the reference wave period T'_r used for the normalization in Eqs. 3–5 is taken as the spectral peak period of the incident wave, $T'_p = 2$ s, for run RN3.
- KSREF=1.0 for random waves as recommended in Sections 2.3.3 and 3.5.
- KSSEA=1.0 for random waves as recommended in Sections 2.3.3 and 3.5.
- DSEAP=0.3 m where the water depth below SWL at the seaward boundary located at $x = 0$ in Fig. 5 was $d'_t = 0.30$ m in the experiment.
- TSLOPS=0.5 to calculate the value of the surf similarity parameter ξ corresponding to the 1:2 ($\tan\theta' = 0.5$) slope where ξ is proportional to the normalized slope θ defined in Eq. 5 as explained in relation to Eq. 8.
- NBSEG=3 where the bottom geometry shown in Fig. 5 consists of 3 linear segments.
- WBSEG(1) = 4.70 m which is the horizontal length of the 1:23.5 beach for run RN3 with $d'_s = 0.10$ m.
- TBSLOP(1) = 0.0425 which is the value of $\tan\theta'$ for the 1:23.5 slope.
- FWP(1) = 0.005 which is the value of the bottom friction factor f' for the smooth slope calibrated by Kobayashi and Raichle (1994) for their rough slope tests.
- WBSEG(2)=0.40 m which is the horizontal distance of the 1:2 slope shown in Fig. 5.
- TBSLOP(2)=0.5 corresponding to the 1:2 slope.
- FWP(2)=0.01 which is the value of f' for the rough slope calibrated by Kobayashi and Raichle (1994) for the overtopping flow with considerable entrained air in their small-scale experiment.
- WBSEG(3)=0.13 m which is the width of the horizontal crest shown in Fig. 5.
- TBSLOP(3)=0.0 for the horizontal crest.
- FWP(3)=0.005 for the smooth crest calibrated by Kobayashi and Raichle (1994) for the overtopping flow with considerable entrained air in their small-scale experiment.
- NODB(1)=1 corresponding to the node number at the location of wave gage 1 located at $x' = 0$.
- NODB(2)=12 corresponding to the node number at the location of wave gage 2 located at $x' = 0.11$ m in Fig. 5 where the nodal spacing $\Delta x' = 0.01$ m for INITS=490.
- NODB(3)=45 corresponding to the node number at the location of wave gage 3 located at $x' = 0.44$ m.

- NODB(4)=471 corresponding to the node number at the location of wave gage 4 located at $x' = 4.70$ m from run RN3 with $d'_s = 0.10$ m.

- NODB(5)=517 corresponding to the adjacent node number at the location of wave gage 5 situated at $x' = 5.165$ m in the middle of the horizontal crest in Fig. 5.

It is noted that the required input could be reduced by providing standard input values in the program, which could be modified by users if desired. Hopefully, the preparation of the input data file FINP1 will become less cumbersome after users become familiar with the input parameters and variables.

The CPU time for both BEFORR2 and RBREAK2 computations for each run was about two hours using an IBM 3090- 300E computer or a Silicon Graphics workstation. This CPU time is typical of irregular wave computations of sufficient durations using RBREAK2.

All the output files produced by RBREAK2 have been explained in Section 3.8. The output files used in the subsequent comparisons between the measured and computed results are explained in the following. Table 12 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 those associated with wave overtopping and time-averaged wave energy balance presented in Section 5.3.6.

The normalized incident and reflected wave trains at $x = 0$ denoted by η_i and η_r, respectively, are obtained from the file OSEWAV where η_i and η_r are normalized by the reference wave height $H'_r = 9.03$ cm based on the measured value of H'_{mo} and stored at the rate of NRATE=40 points per reference wave period $T'_r = 2$ s in Table 12. These wave trains are computed using the method of characteristics as explained in Section 2.5 and indicated by the letters MC in the following. For example, η_i and η_r for run RN3-T-MC imply the computed incident and reflected wave trains based on the method of characteristics for run RN3 with IWAVE=3 (denoted by T). The time series parameters associated with η_i and η_r are computed using the subroutine TIMPAR, whereas the spectra and spectral parameters corresponding to η_i and η_r are computed using the subroutines SPCTRA and SPCPAR. These subroutines were presented by Cox et al. (1991) and modified slightly by Poff and Kobayashi (1993b). Some of the spectral and time series parameters for the computed incident wave train η_i based on the method of characteristics are tabulated in Table 9. It is noted that the dimensional value of H'_{mo} for runs with the letter I-MC is the same as the measured value of H'_{mo} because the measured incident wave train specified as input for IWAVE=2 (denoted by I) is also the output incident wave train for the method of characteristics (denoted by MC) adopted in RBREAK2. The average reflection coefficient $\bar{r}$ based on the zero moments of the reflected and incident wave spectra is listed in Table 10. It should be stated that the value of H'_{mo} used for the normalization in Table 10 is the measured or computed value of H'_{mo} listed in Table 9. For example, the average overtopping rate normalized by $H'_r = 9.03$ cm in Table 12 is 3.12×10^{-3}, while the average overtopping rate $\bar{Q}$ normalized by $H'_{mo} = 11.79$ cm for run RN3-T-MC is 2.09×10^{-3} in Table 10.

The time series of the computed normalized water depth h stored at the locations of wave gages 1–5 in Fig. 5 are obtained from the files OSAV2B01 and OSAV2B02 as explained in Section 3.8.6. The computed value of h at wave gage 5 multiplied by $1/(1-a)$ with $a = 0.8$ is compared with the measured normalized depth h_o of overtopping flow on the crest of the revetment in Section 5.3.6. The parameter a is the fraction of entrained air in volume in the measured depth

Table 12: Concise output from file ODOC for run RN3-T.

```
-----------------------------------
RUN RN3: SPECIFY TOTAL WAVE
-----------------------------------

WAVE CONDITION

Total Wave at Seaward Boundary Given as Input
Number of Data Points      NDATA =    6501
Norm. Maximum Surface Elev.      =        0.674396
Norm. Minimum Surface Elev.      =       -0.780853

Reference Wave Period            =        2.000000 sec.
Reference Wave Height            =        0.090300 meters
Depth at Seaward Boundary        =        0.300000 meters
Shoal. Coef. at Reference  Ks1 =        1.000
             at Seaw. Bdr. Ks2 =        1.000
             Ks = Ks2/Ks1       =        1.000
Norm. Depth at Seaw. Boundary    =        3.322
Normalized Wave Length           =       10.859
"Sigma"                          =       20.846
Ursell Number                    =       35.495
Surf Similarity Parameter        =        4.158

SLOPE PROPERTIES

Norm. Horiz. Length of
      Computation Domain    =         2.773361
Number of Segments          =         3
---------------------------------
 SEGMENT    WBSEG(I)   TBSLOP(I)    FWP(I)
     I       meters
---------------------------------
      1    4.700000    0.042500    0.005000
      2    0.400000    0.500000    0.010000
      3    0.130000    0.000000    0.005000
---------------------------------

COMPUTATION PARAMETERS

Total Number of Spatial Nodes        JE =        523
Number of Nodes Along Bottom Below SWL
                                  INITS =        490
Normalized Delta x           =        0.531295D-02
Damping Coefficients     x1 =             2.000
                         x2 =             2.000
Computation Duration   TMAX =        162.500000 Wave Periods
Minimum Allowable NONE            NONEM =       1320
Time Series Stored at Rate NRATE
     per Wave Period with         NRATE =         40
Statistical Calculations Are Performed Excluding
     the First MSTAT Computation Units
     with                         MSTAT =          0
```

Continued on next page.

Table 12. Continued.

```
REFLECTION COEFFICIENT

r1 =      0.683

RUNUP, RUNDOWN, SETUP

Largest Node Number Reached by Computational Waterline
                                          JMAX =      523
---------------------------------------------------------
      I   DELTAR(I)    RUNUP(I)   RUNDOWN(I)   SETUP(I)
            [mm]          R          Rd          Zr
---------------------------------------------------------
      1    1.000       1.237      -0.783       0.294
---------------------------------------------------------

OVERTOPPING

Norm. Average Overtopping Rate           =    0.312302D-02
Norm. Average Flow at Seaward Boundary =    0.412542D-02
Norm. Maximum Overtopping Rate           =    0.132118D+00
      Occuring at Normalized Time    t =         116.068939
Norm. Overtopping Duration               =         144.647084
Notes: Norm. duration of overtopping computation =         162.500000
       Norm. duration of RBREAK2 computation      =         162.500000

WAVE SET-DOWN OR SETUP

Average value of ETAI =     0.013416
                 ETAR =    -0.005232

TIME-AVERAGED ENERGY BALANCE

Normalized Energy Flux:
. at Seaw. Boundary   A =  0.168066D+00
. at Landw. Boundary  B =  0.433227D-02
Normalized Rate of Energy Dissipation
in the Computation Domain, Due to:
. Bottom Friction      C =  0.342324D-02
. Wave Breaking        D =  0.157162D+00
Additional Term        E =  0.314851D-02
Calculation 1:
              F = A-B =  0.163734D+00
            G = C+D+E =  0.163734D+00
  Must H=0, but H = G-F =  0.111022D-15
  % error  100G/E =          0.00
```

h_o and $a = 0.8$ has been applied by Kobayashi and Raichle (1994) for their overtopping flow experiment in which the toe depth was in the range $d'_s = 14$–17 cm as compared to $d'_s = 9$–12 cm in Table 9. The time series of the normalized free surface elevation η at the locations of wave gages 1–4 are calculated using $\eta = (h + z_b)$ where z_b is the normalized bottom elevation relative to SWL ($z_b < 0$ below SWL) and stored in the file OSPACE. The computed and measured time series of the normalized free surface elevation η_s at wave gage 4 located at the toe of the 1:2 rough slope are compared in Section 5.3.5.

The computed free surface oscillations at wave gages 1–3 are used to estimate the incident and reflected wave trains at wave gage 1 in the same way as the separation of incident and reflected waves using the measured free surface oscillations at wave gages 1–3. the computed incident and reflected wave trains based on the three gage method are indicated by the symbol 3G in the following. For example, η_i and η_r for run RN3-T-3G imply the computed incident and reflected wave trains based on the three gage method for run RN3 with IWAVE=3. The spectral and time series parameters associated with η_i and η_r based on the three gage method (denoted by 3G) are computed in the same manner as those based on the method of characteristics and listed in Tables 9 and 10.

In summary, runs RN3 and RW3 have been selected for the comparisons with RBREAK2. To compare the options of IWAVE=2 and 3 denoted by I and T, respectively, four computations using BEFORR2 and RBREAK2 have been made for runs RN3-I, RN3-T, RW3- I and RW3-T using the measured incident (I) and total (T) waves at $x = 0$ as input to RBREAK2. The incident and reflected wave trains for each of the four runs have been computed using the method of characteristics and the three gage method which are indicated by MC and 3G, respectively.

5.3.3 Incident Waves at Seaward Boundary

The measured and computed incident wave trains η_i at the seaward boundary based on the three gage method are compared in Fig. 10 for runs RN3-T and RN3-I and in Fig. 11 for runs RW3-T and RW3-I. The temporal variations of η_i normalized by the measured value of H'_{mo} for each run are computed for the normalized duration of $0 \leq t \leq 162.5$ where t is the time normalized by $T'_p = 2$ s. The measured and computed incident wave spectra S_i corresponding to η_i for $0 \leq t \leq 162.5$ are shown in Fig. 12 for runs RN3-T and RN3-I and in Fig. 13 for runs RW3-T and RW3-I. The dimensional frequency f'_* is normalized as $f_* = f'_* \, T'_p$ where f' is the bottom friction factor in this report. The normalized spectral peaks in Figs. 12 and 13 are located at $f_* = 1$. The spectra shown in this report are smoothed spectra with 16 degrees of freedom. The measured spectra for runs RN3 and RW3 correspond to the narrow (N) and wide (W) spectra. The effective frequency range of resolution for the three gage spacings shown in Fig. 5 in $0.39 \leq f_* \leq 5.05$. Figs. 10–13 indicate that the specification of the measured total (T) free surface elevation as input with IWAVE=3 results in a slight overprediction of the incident wave oscillations especially in the vicinity of $f_* = 1$, whereas the specification of the measured incident (I) wave train as input with IWAVE=2 results in good agreement. This is mainly because the computed incident wave train η_i when $\eta = (\eta_i + \eta_r)$ at $x = 0$ is specified as input is affected by the errors involved in the computed reflected wave train η_r. Correspondingly, the computed values of H'_{mo}, H'_s, Q_p and $\bar{j}$ for the option IWAVE=3 listed in Table 9 are larger than the measured values.

The computed incident wave trains η_i based on the three gage method and the method of characteristics are compared in Fig. 14 for runs RN3-T and RN3-I and in Fig. 15 for runs

RW3-T and RW3-I. The corresponding comparisons of the computed incident wave spectra S_i are shown in Figs. 16 and 17. These figures show that the two methods yield approximately the same incident waves. It is noted that small low-frequency wave components are present for runs RN3-T-MC and RW3-T-MC which do not employ the three gage method with the effective frequency range of resolution in specifying the measured total free surface elevation at $x = 0$ as input and estimating the computed incident wave at $x = 0$. Figs. 14–17 indicate that the method of characteristics adopted in RBREAK2 is as accurate as the three gage method for which additional computational efforts are required. Consequently, little additional information will be gained by storing and analyzing the computed free surface oscillations at three gage locations in the vicinity of the seaward boundary.

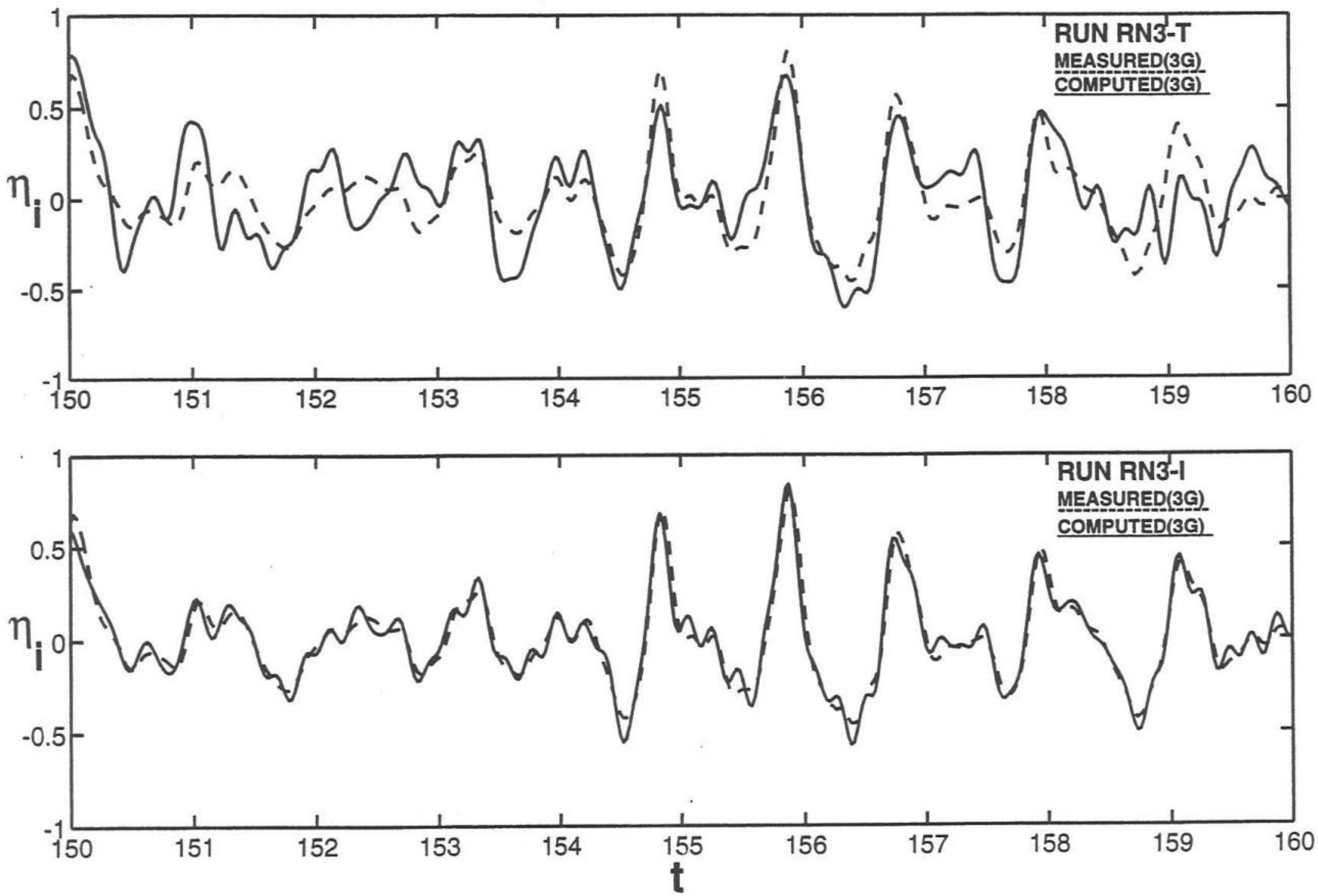

Figure 10: Measured and computed incident wave trains η_i for $150 \leq t \leq 160$ based on three gage method (3G) for runs RN3-T and RN3-I.

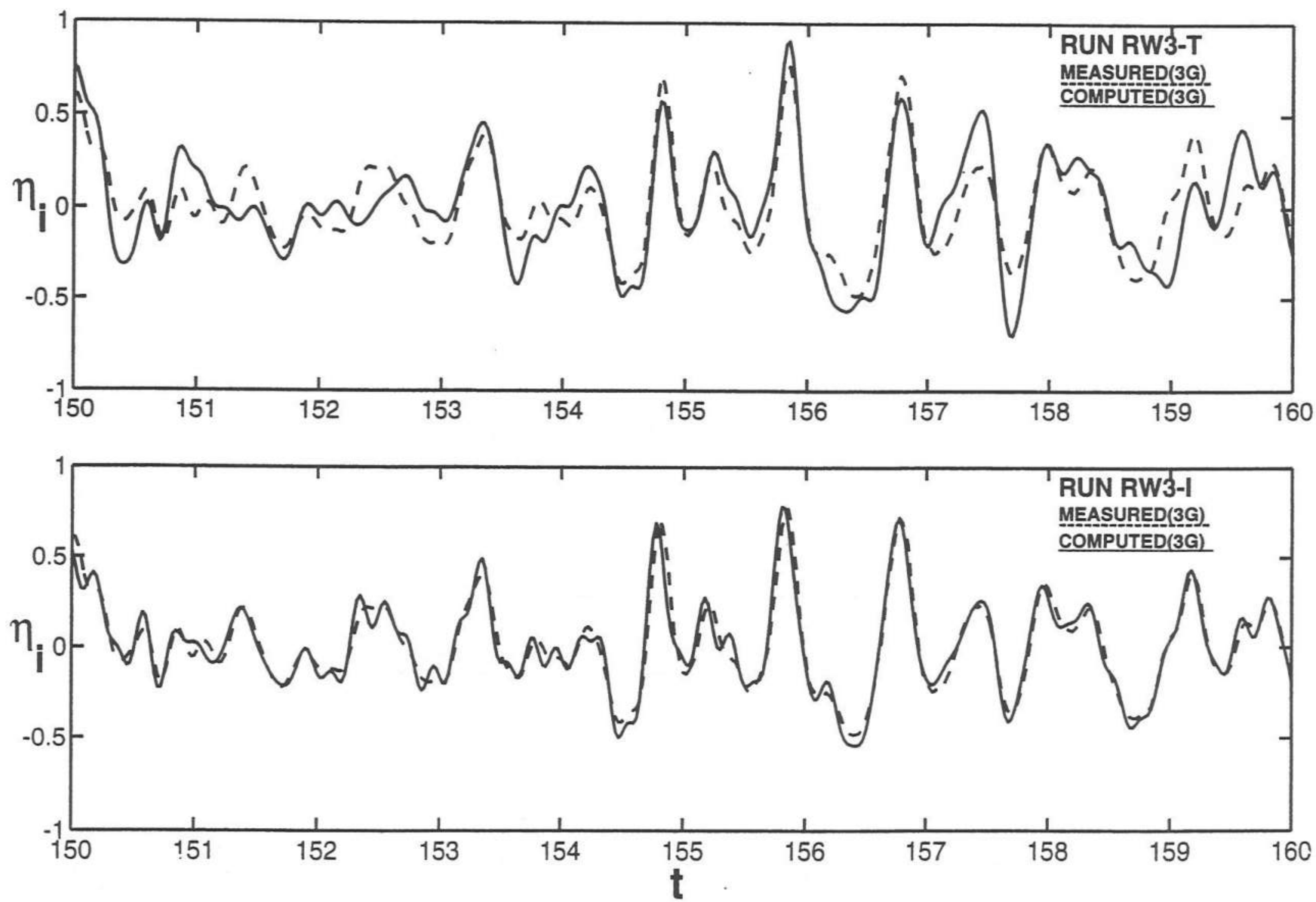

Figure 11: Measured and computed incident wave trains η_i for $150 \leq t \leq 160$ based on three gage method (3G) for runs RW3-T and RW3-I.

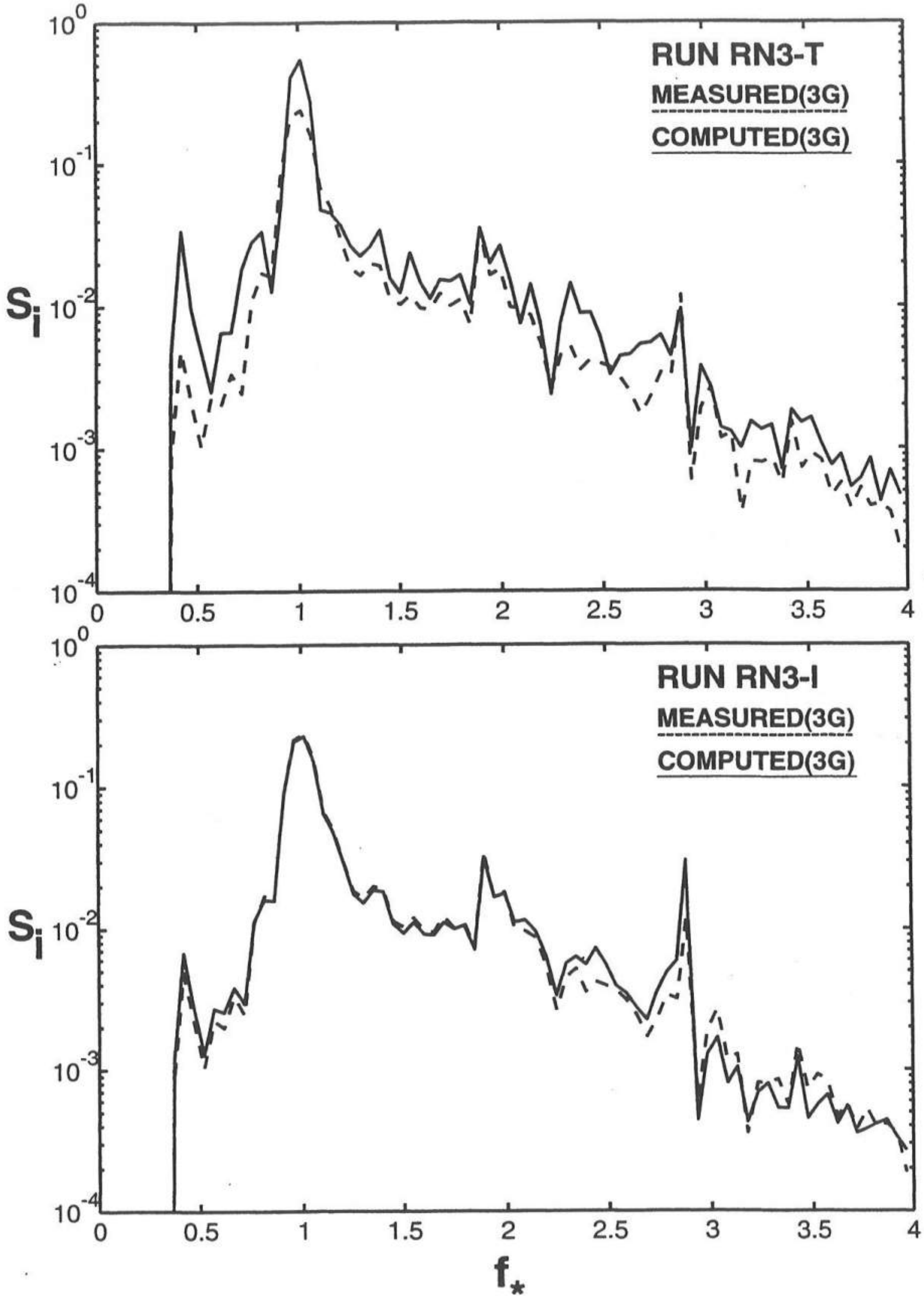

Figure 12: Measured and computed incident wave spectra S_i based on three gage method (3G) for runs RN3-T and RN3-I.

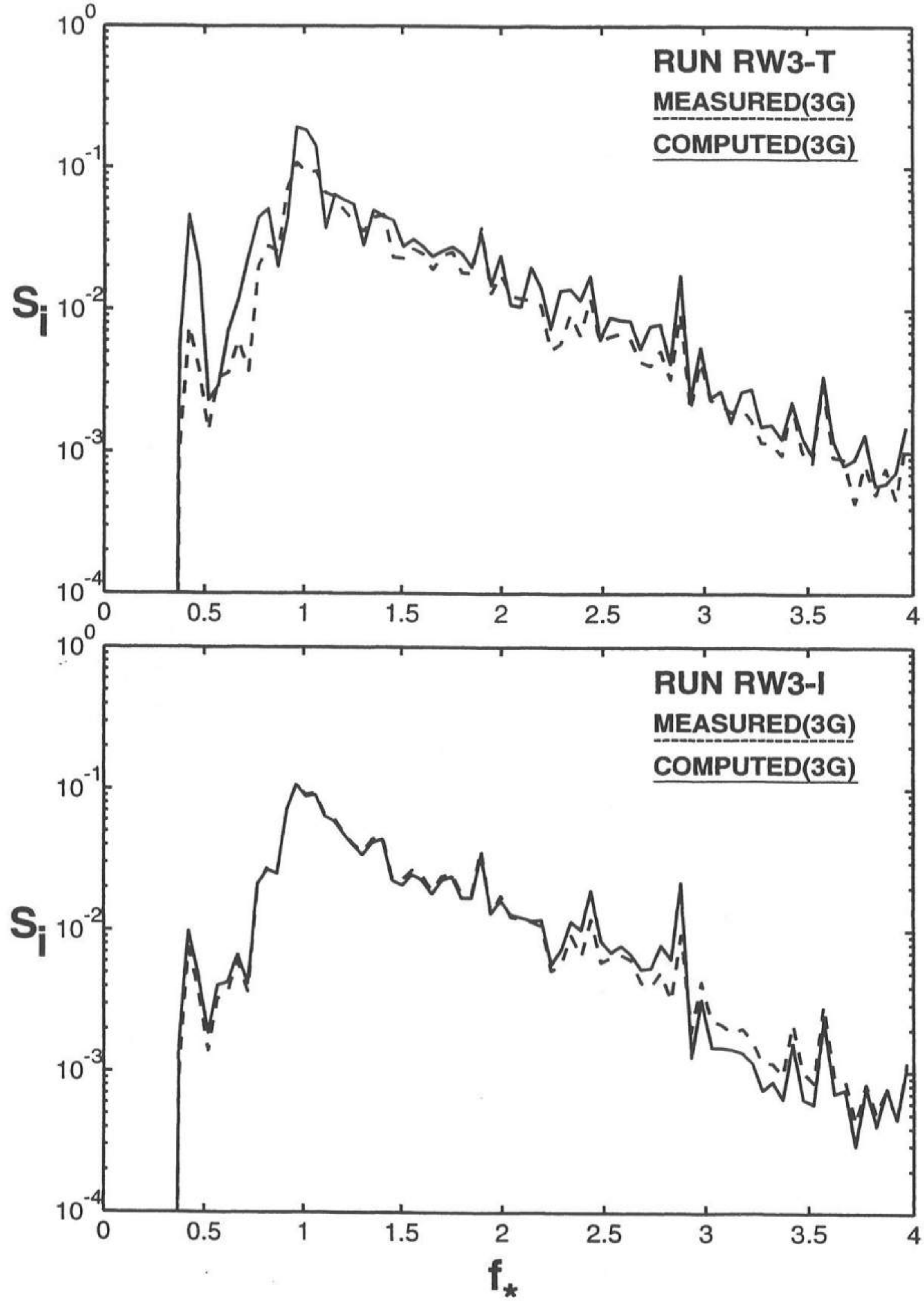

Figure 13: Measured and computed incident wave spectra S_i based on three gage method (3G) for runs RW3-T and RW3-I.

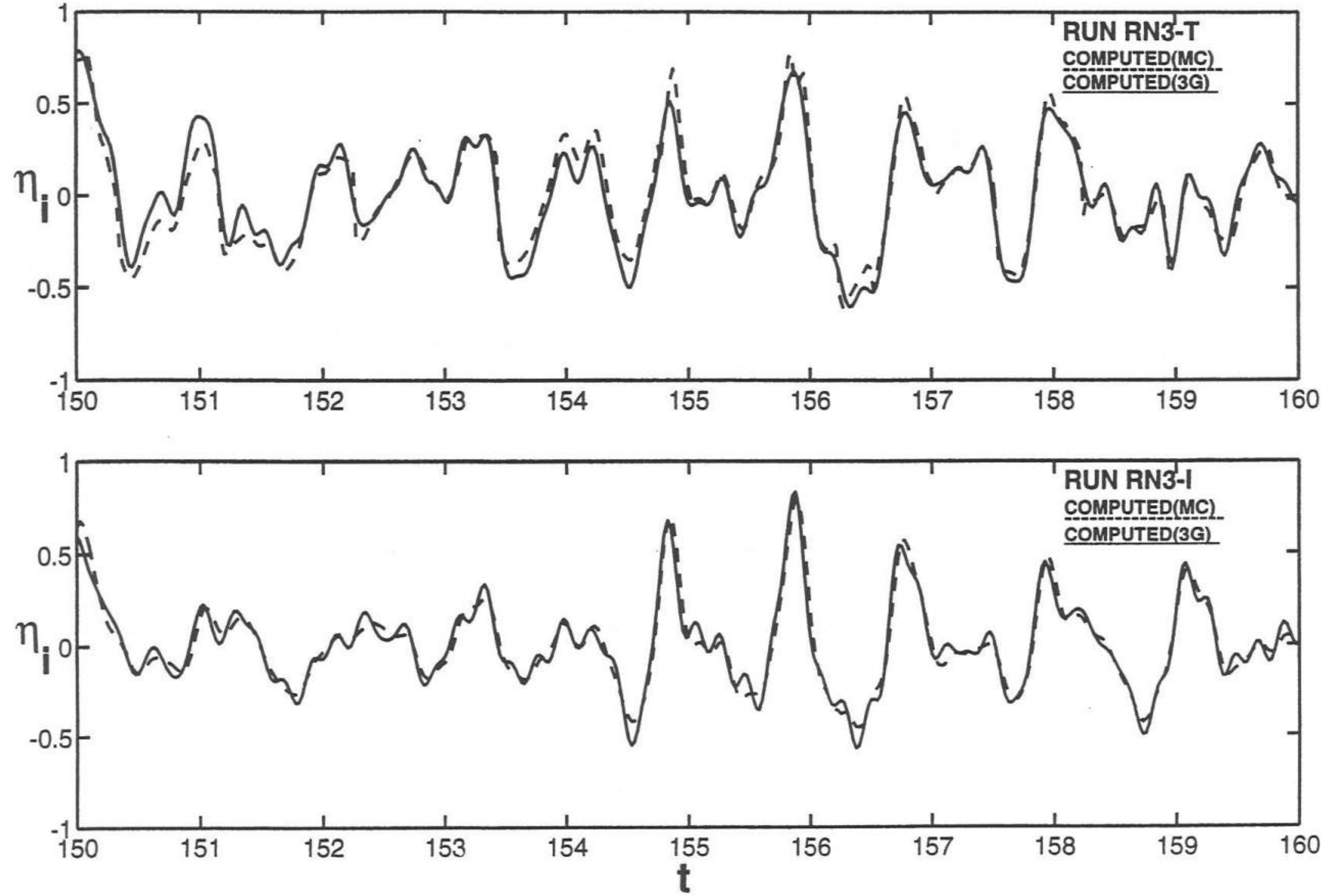

Figure 14: Computed incident wave trains η_i for $150 \leq t \leq 160$ based on three gage method (3G) and method of characteristics (MC) for runs RN3-T and RN3-I.

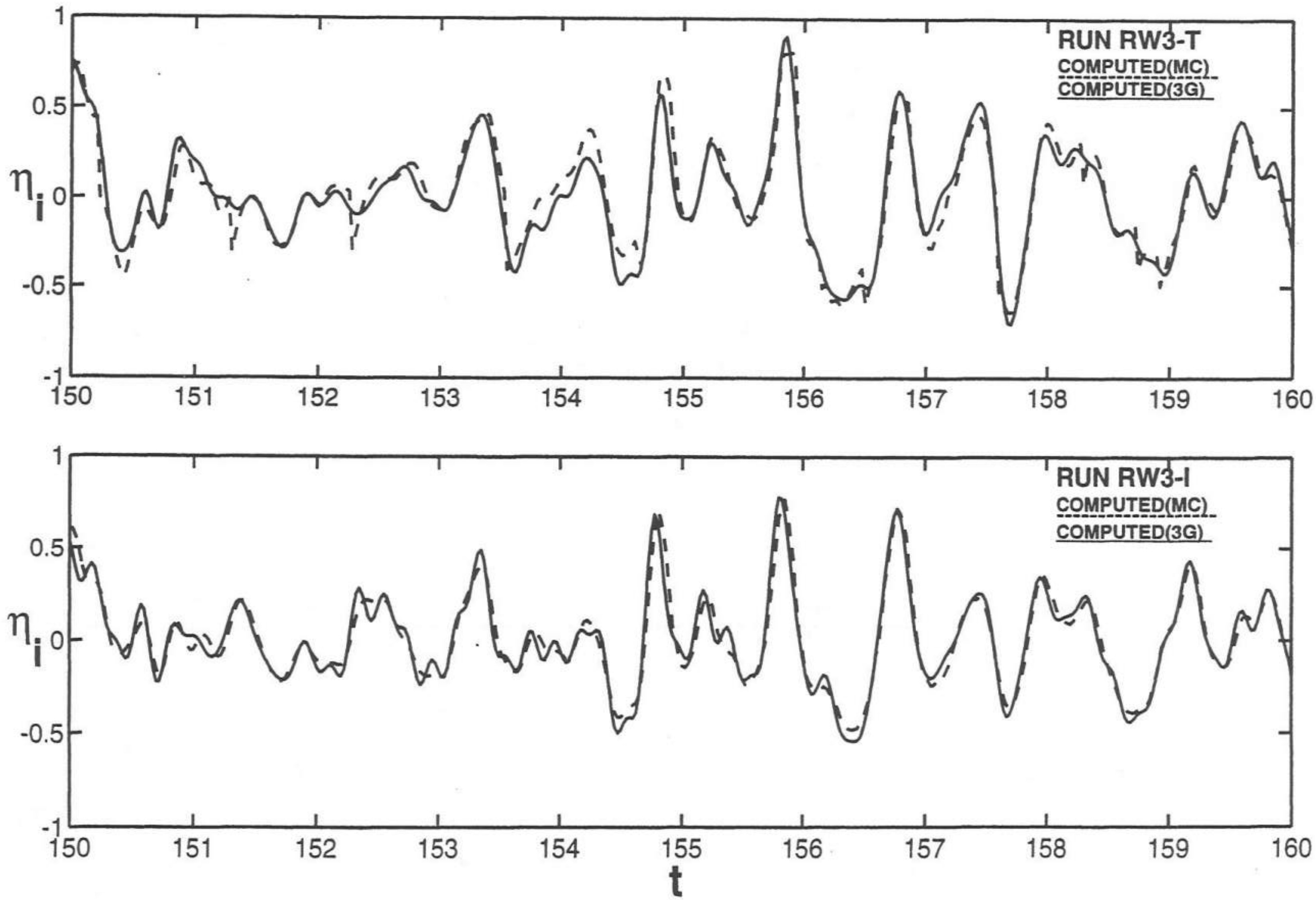

Figure 15: Computed incident wave trains η_i for $150 \leq t \leq 160$ based on three gage method (3G) and method of characteristics (MC) for runs RW3-T and RW3-I.

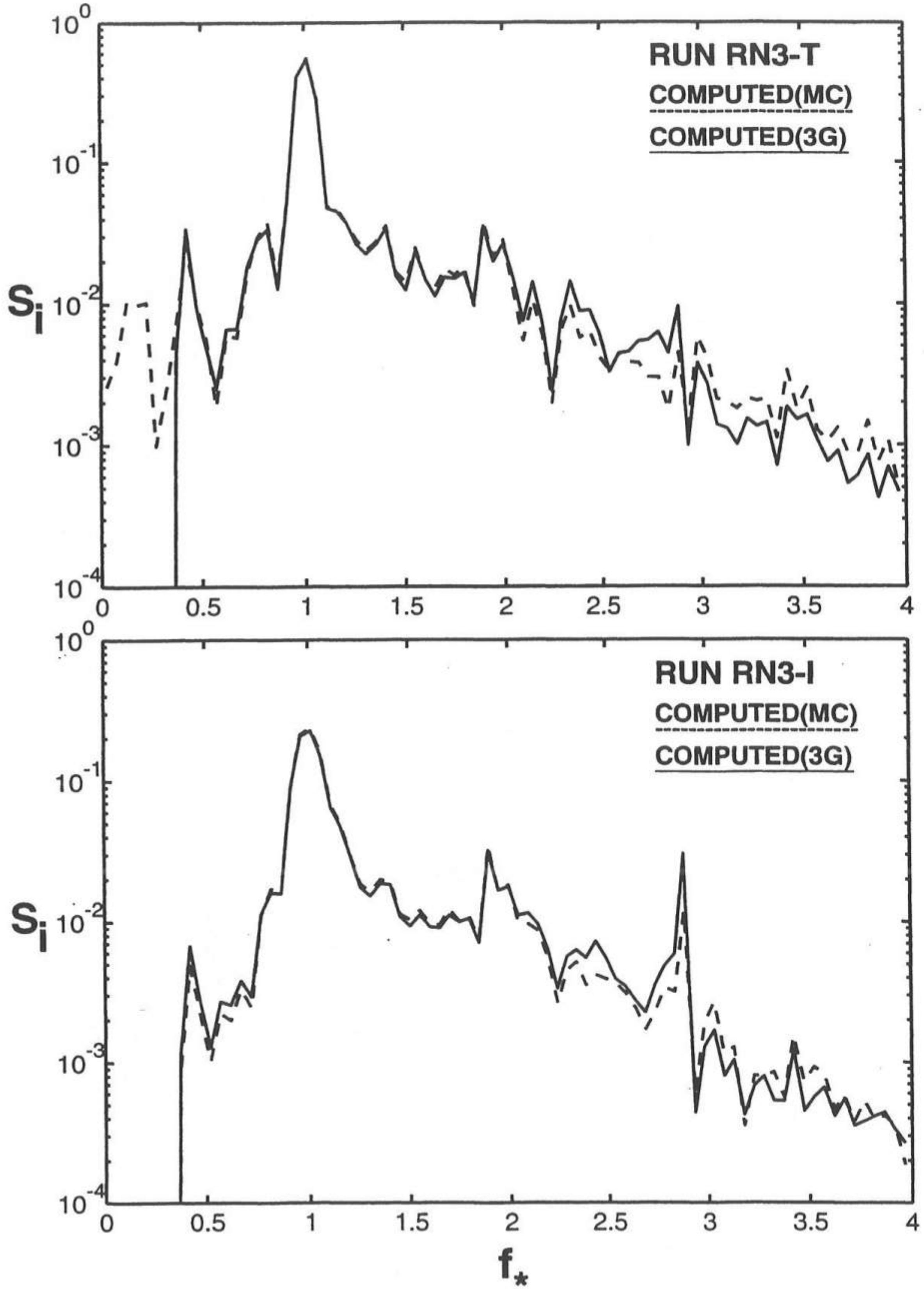

Figure 16: Computed incident wave spectra S_i based on three gage method (3G) and method of characteristics (MC) for runs RN3-T and RN3-I.

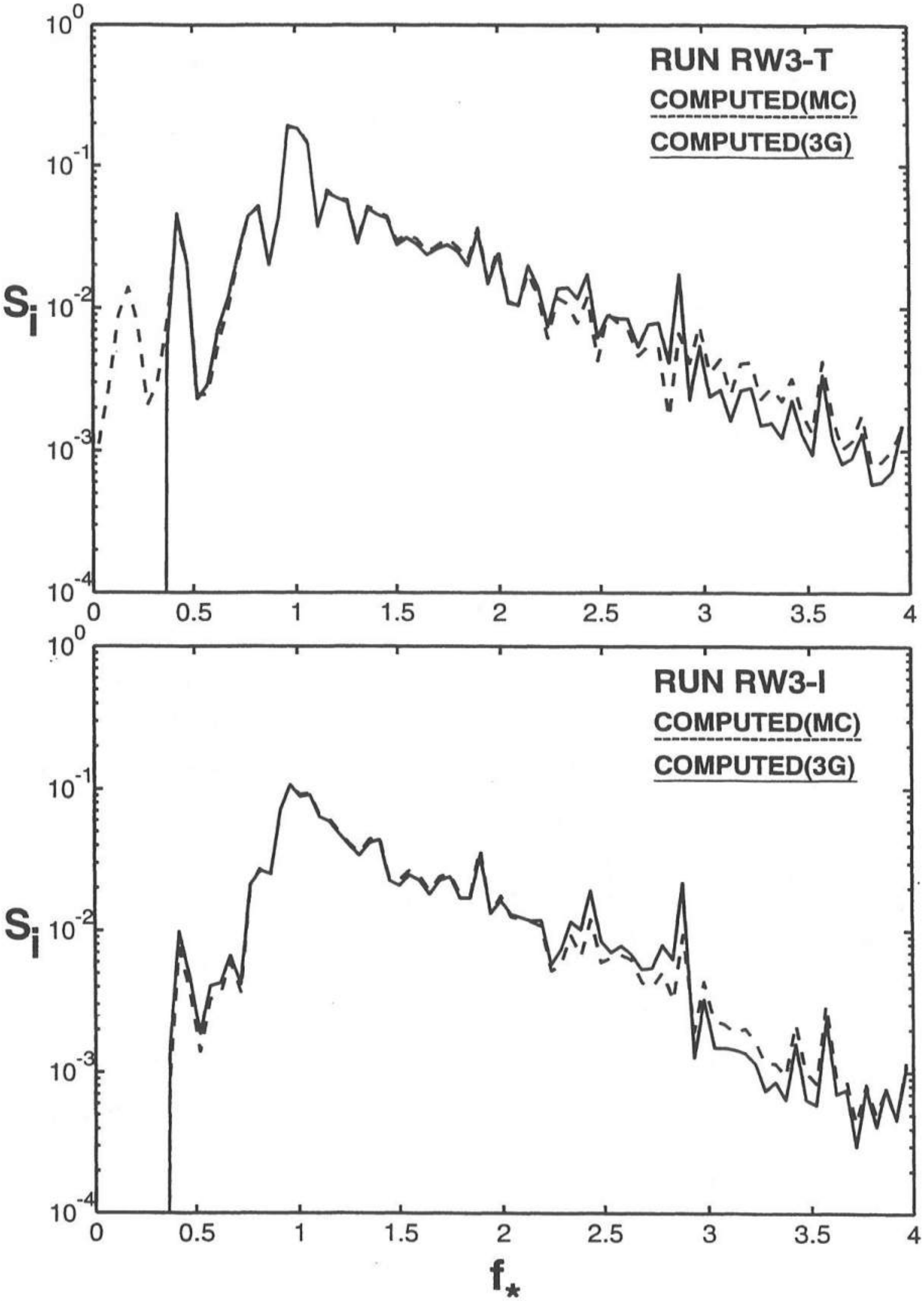

Figure 17: Computed incident wave spectra S_i based on three gage method (3G) and method of characteristics (MC) for runs RW3-T and RW3-I.

5.3.4 REFLECTED WAVES AT SEAWARD BOUNDARY

The measured and computed reflected waves at the seaward boundary are evaluated in the same manner as the incident waves in Section 5.3.3. The measured and computed reflected wave trains η_r based on the three gage method are compared in Fig. 18 for runs RN3-T and RN3-I and in Fig. 19 for runs RW3-T and RW3-I. The measured and computed reflected wave spectra S_r based on the three gage method are compared in Figs. 20 and 21. These figures indicate that the reflected wave spectra are predicted reasonably but the reflected wave trains including phase relationships are predicted poorly. The agreement of the average reflection coefficient $\bar{r}$ listed in Table 10 is good partly because $\bar{r}$ is based on the zero moment of S_r. Unfortunately, the use of the three gage method for both measured and computed reflected waves does not improve the agreement.

The computed reflected wave trains η_r based on the three gage method and the method of characteristics are compared in Fig. 22 for runs RN3-T and RN3-I and in Fig. 23 for runs RW3-T and RW3-I. The corresponding comparisons of the computed reflected wave spectra S_r are shown in Figs. 24 and 25. These figures indicate that these two methods predict similar reflected waves except that the method of characteristics without any frequency resolution restriction can predict low-frequency reflected wave components. The average reflection coefficients $\bar{r}$ listed in Table 10 are practically the same for both methods. Consequently, the three gage method with additional computational efforts is not warranted. The method of characteristics will need to be improved somehow to yield better agreement between the measured and computed reflected wave trains and spectra.

Fig. 26 shows the average reflection coefficient $\bar{r}$ as a function of the normalized toe depth $d_s = d'_s/H'_{mo}$ for the runs listed in Table 10. The computed values of $\bar{r}$ indicated by RN3-T/I and RW3-T/I in Fig. 26 are those based on the three gage method. The measured values of $\bar{r}$ indicated by RN and RW are those for the present rough slope test runs with the narrow and wide spectra of incident waves. The data points indicated by RKN and RKW are from the rough slope tests of Raichle and Kobayashi (1993) for the narrow and wide spectra. The average reflection coefficients $\bar{r}$ of these rough slope tests depend mostly on the normalized toe depth d_s and are affected little by the normalized crest height $H_c = H'_c/H'_{mo}$ and the spectral shape represented by Q_p listed in Table 10. The computed values of $\bar{r}$ tend to be slightly larger than the measured values following an approximately straight line.

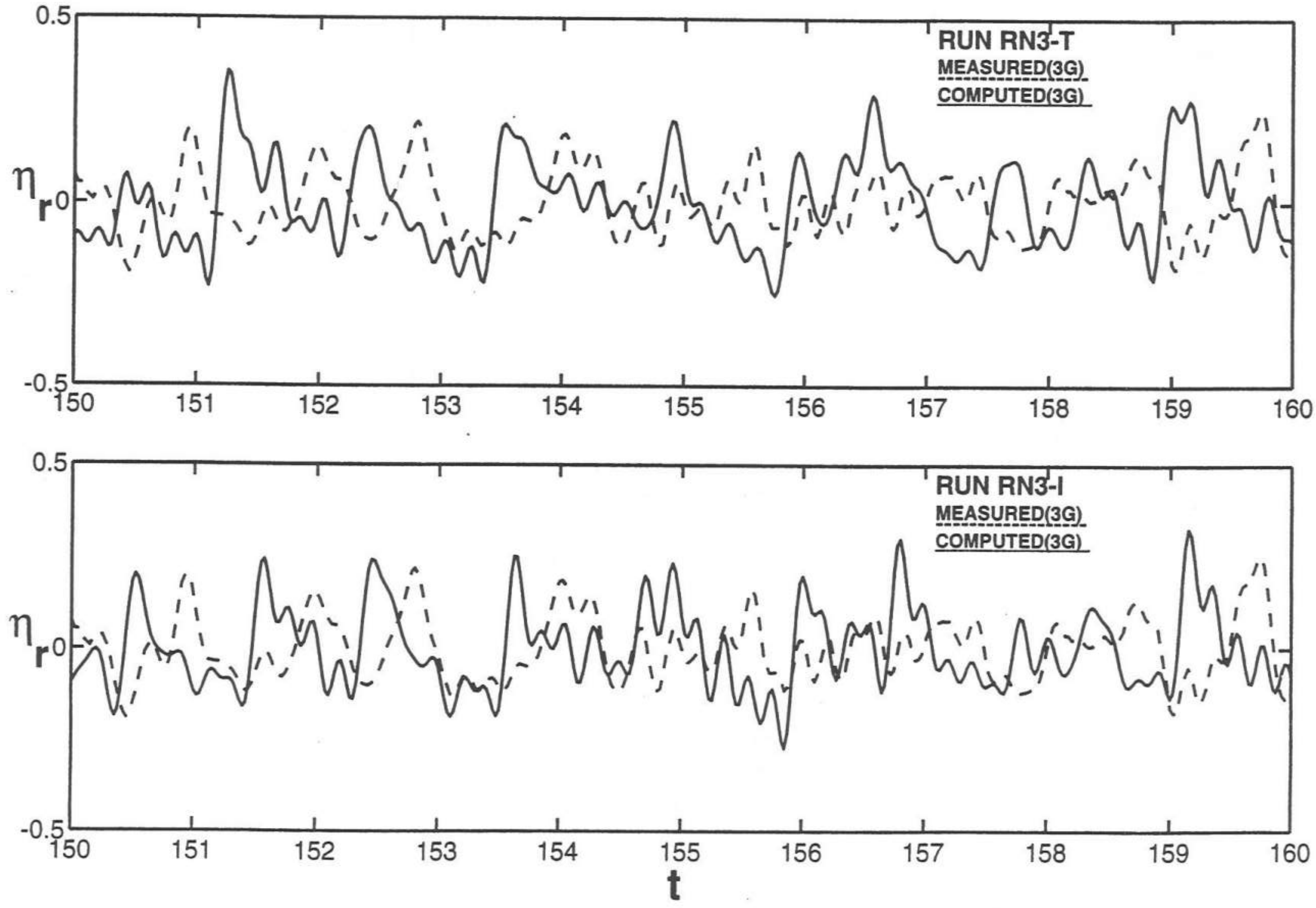

Figure 18: Measured and computed reflected wave trains η_i based on three gage method (3G) for runs RN3-T and RN3-I.

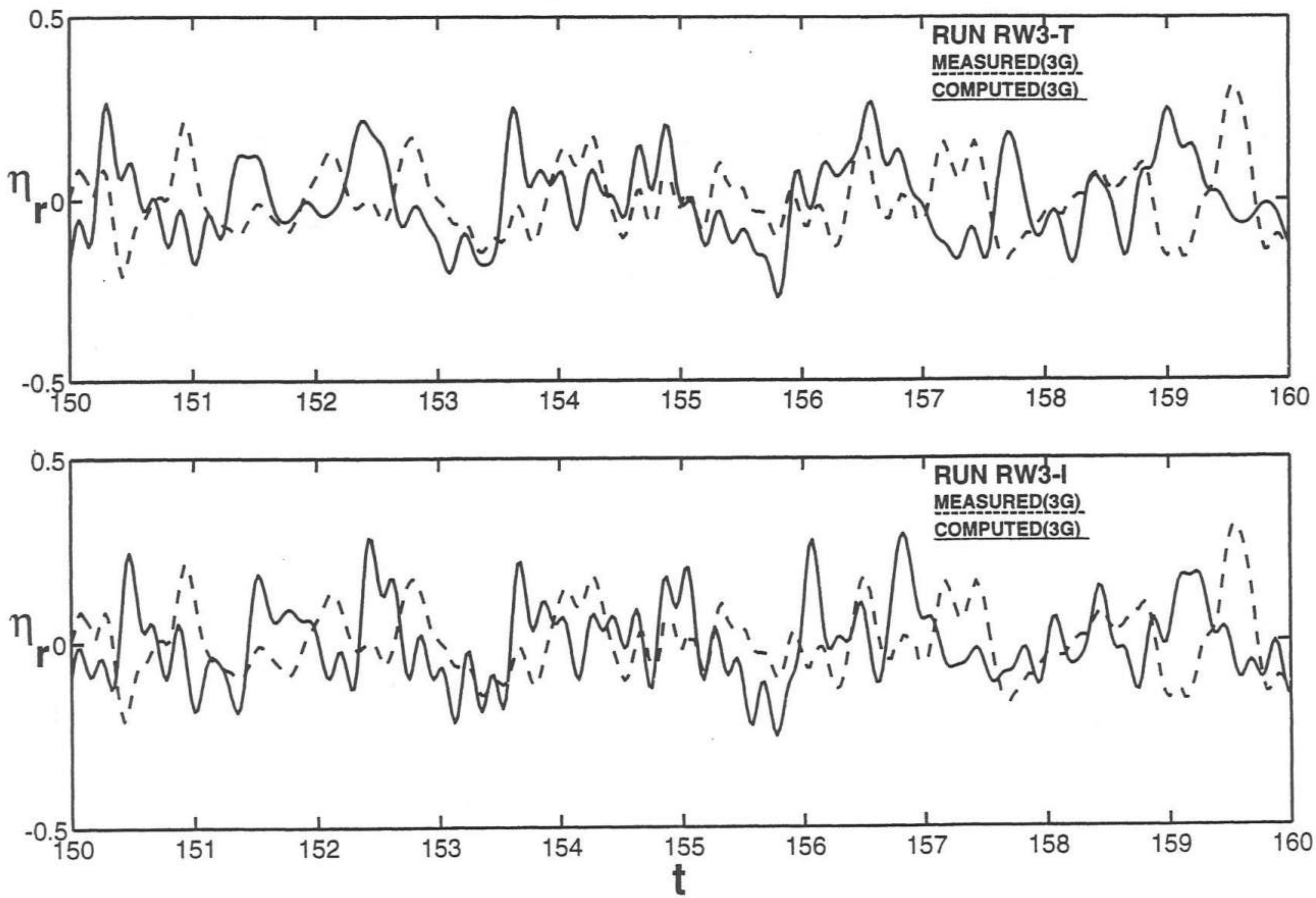

Figure 19: Measured and computed reflected wave trains η_r based on three gage method (3G) for runs RW3-T and RW3-I.

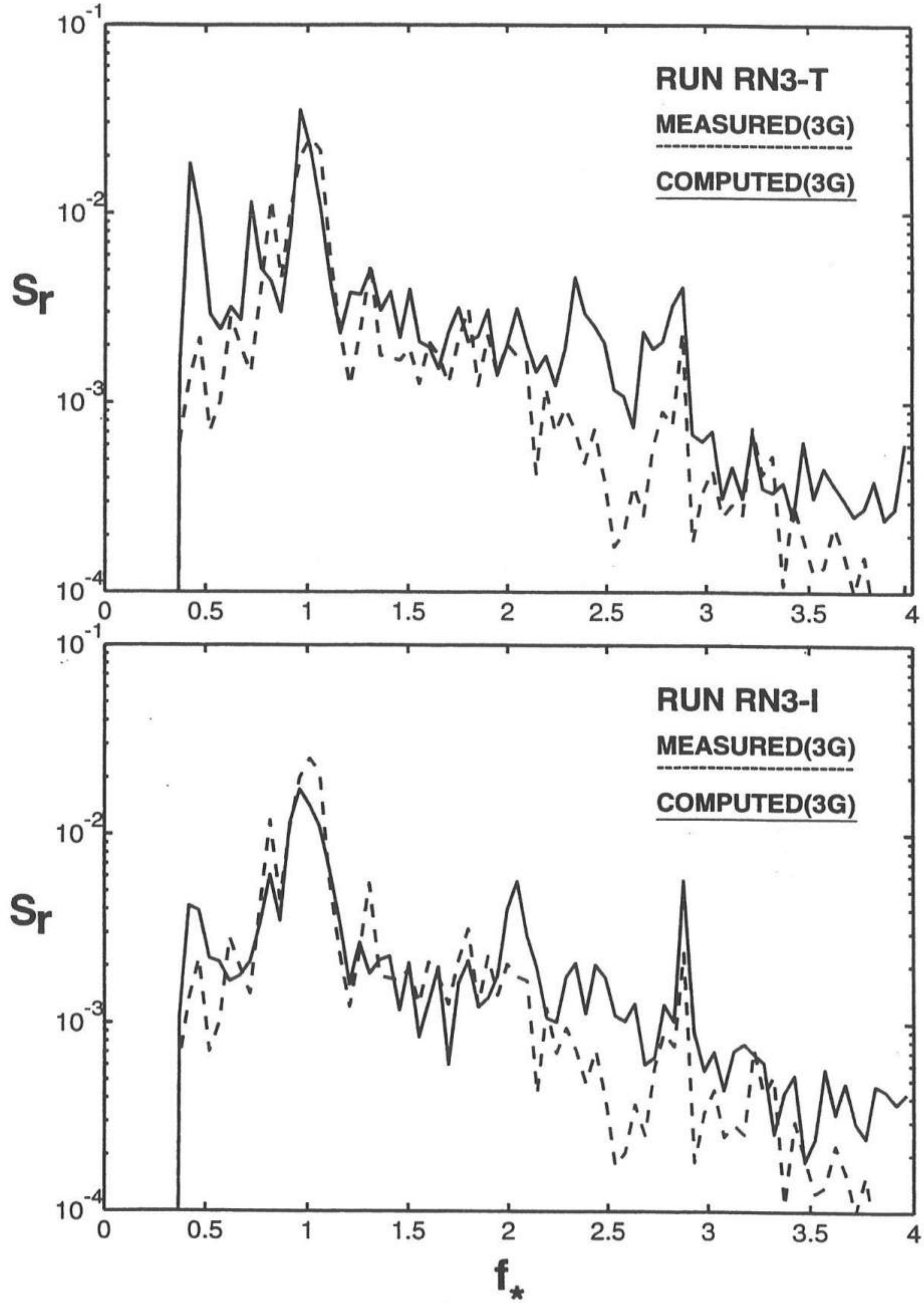

Figure 20: Measured and computed reflected wave spectra S_r based on three gage method (3G) for runs RN3-T and RN3-I.

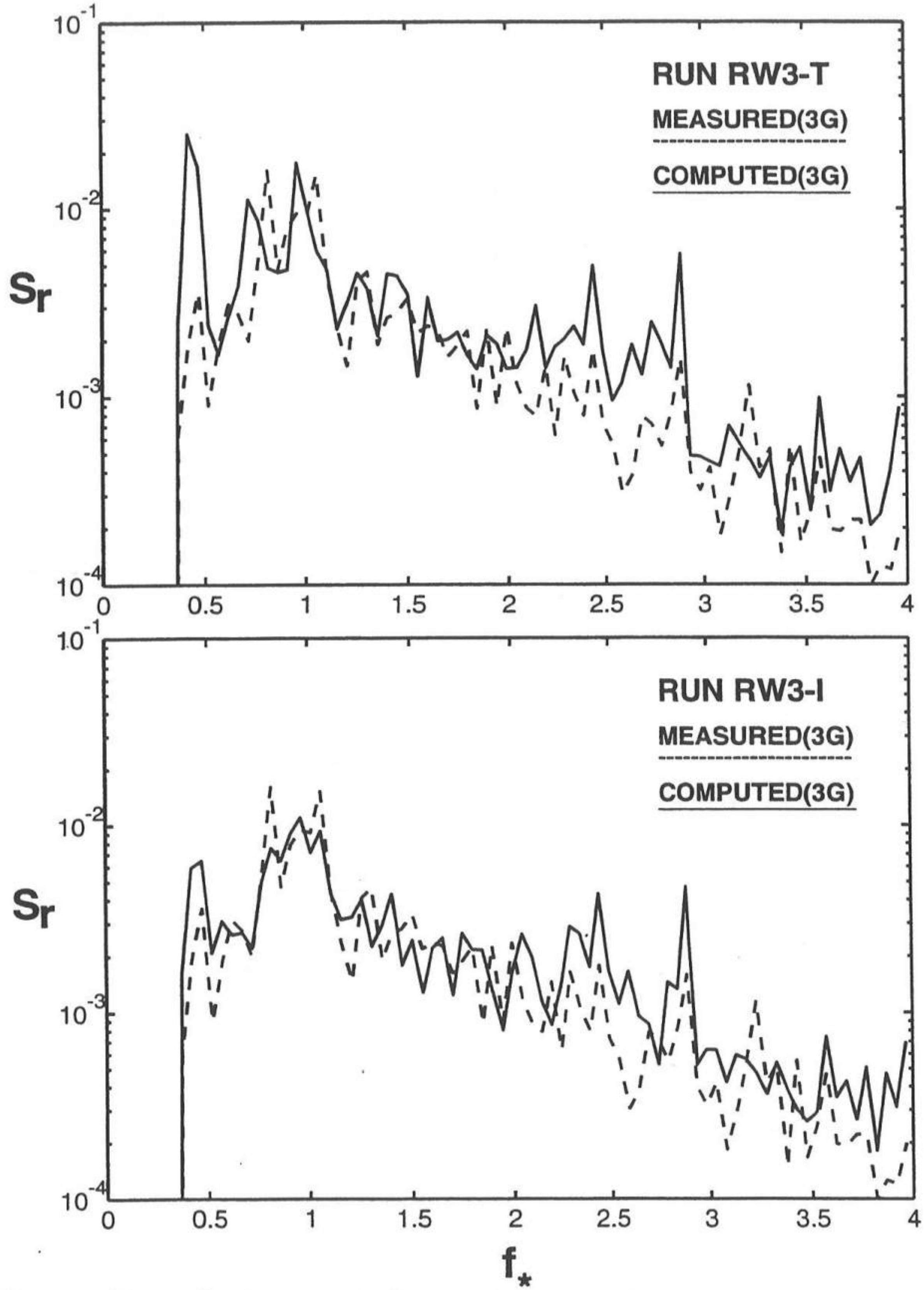

Figure 21: Measured and computed reflected wave spectra S_r based on three gage method (3G) for runs RW3-T and RW3-I.

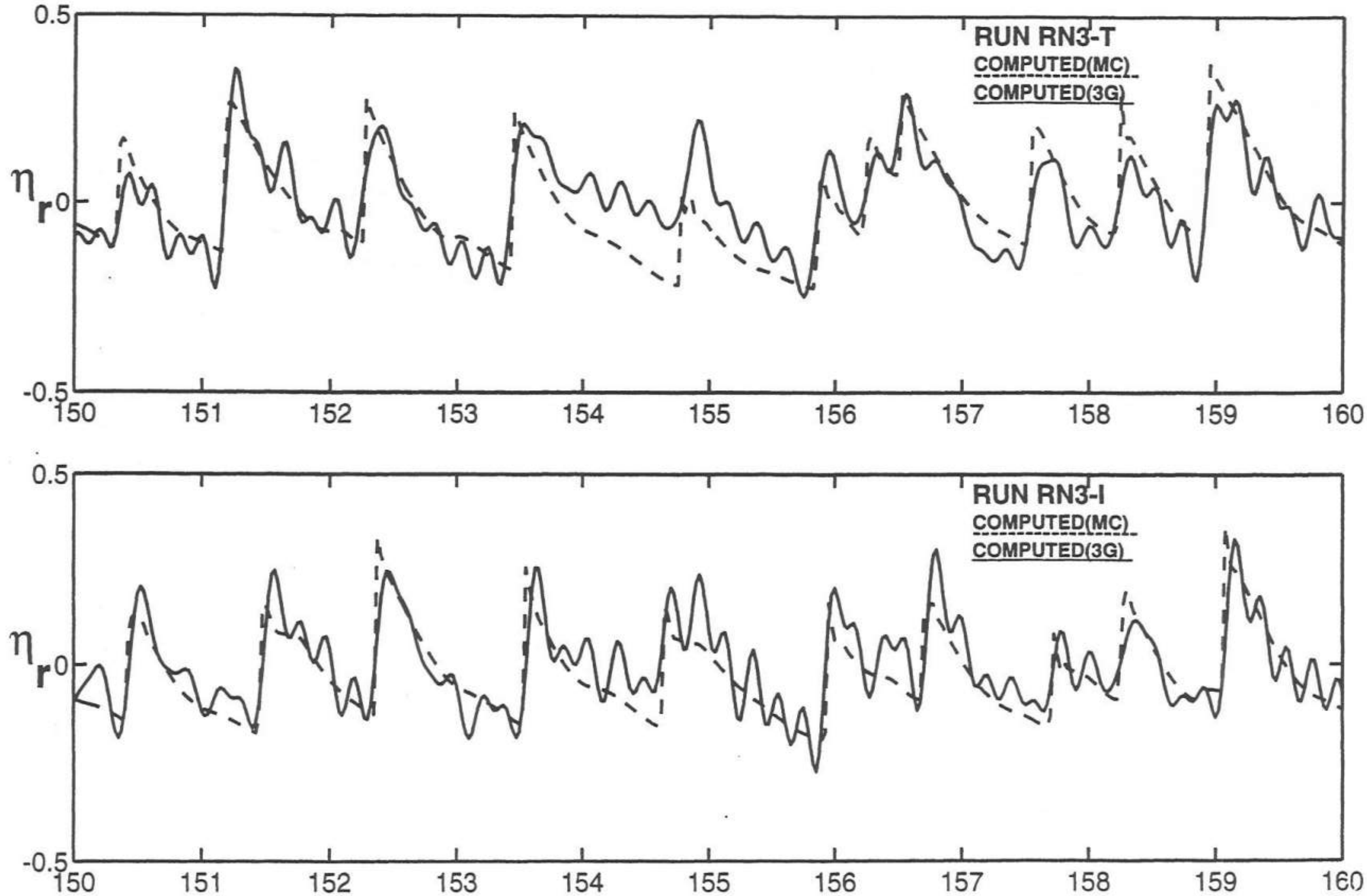

Figure 22: Computed reflected wave trains η_r based on three gage method (3G) and method of characteristics (MC) for runs RN3-T and RN3-I.

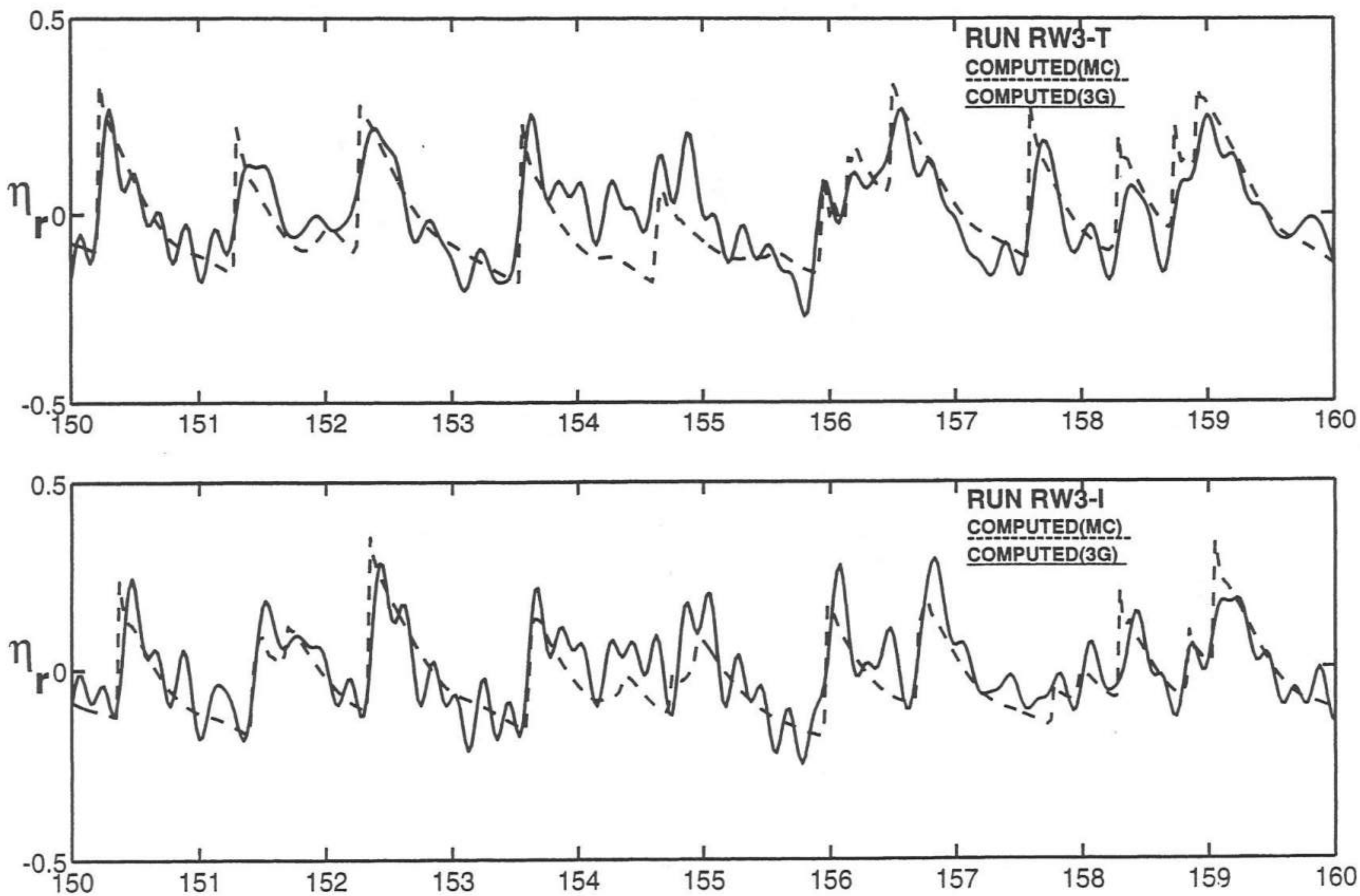

Figure 23: Computed reflected wave trains η_r based on three gage method (3G) and method of characteristics (MC) for runs RW3-T and RW3-I.

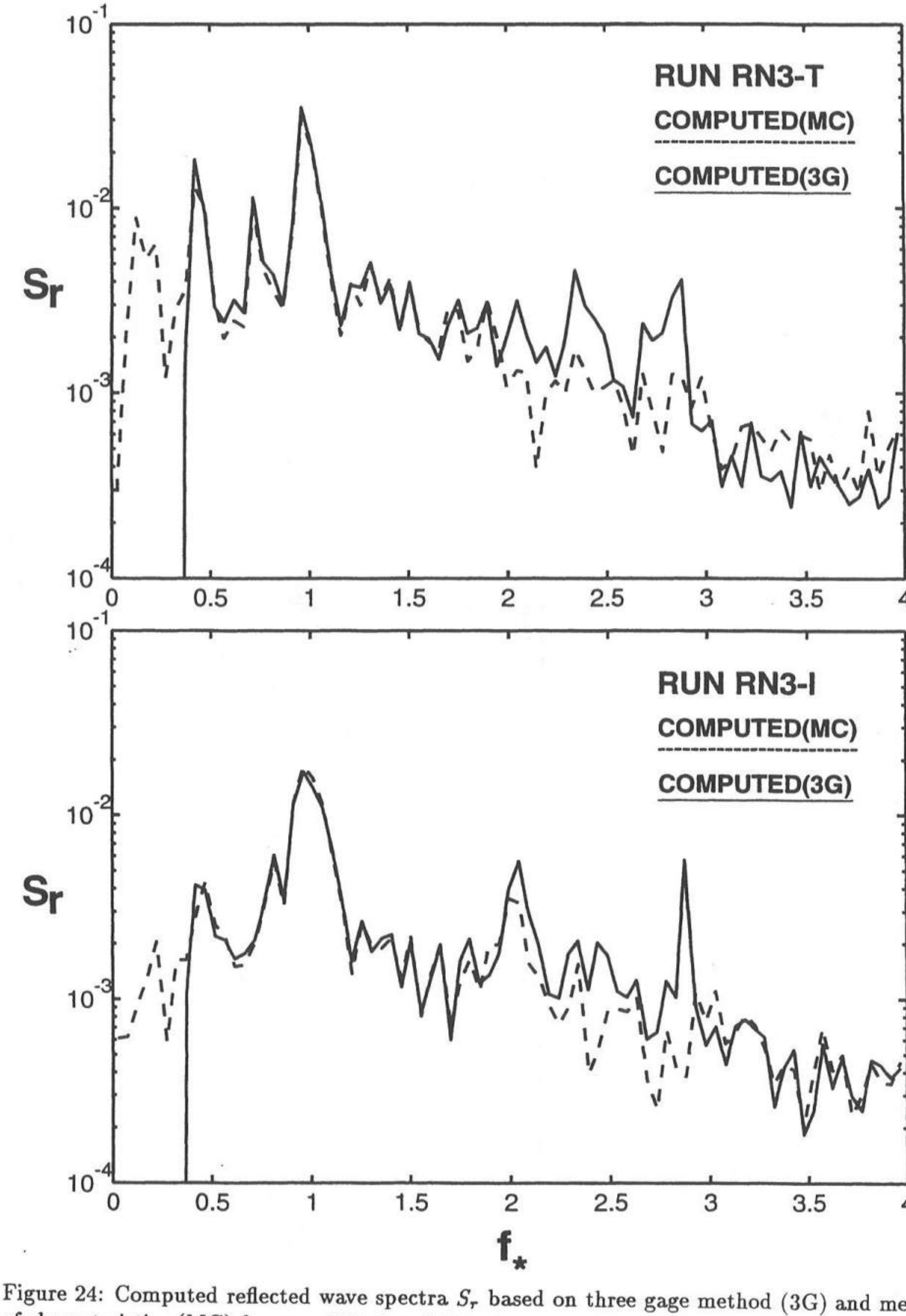

Figure 24: Computed reflected wave spectra S_r based on three gage method (3G) and method of characteristics (MC) for runs RN3-T and RN3-I.

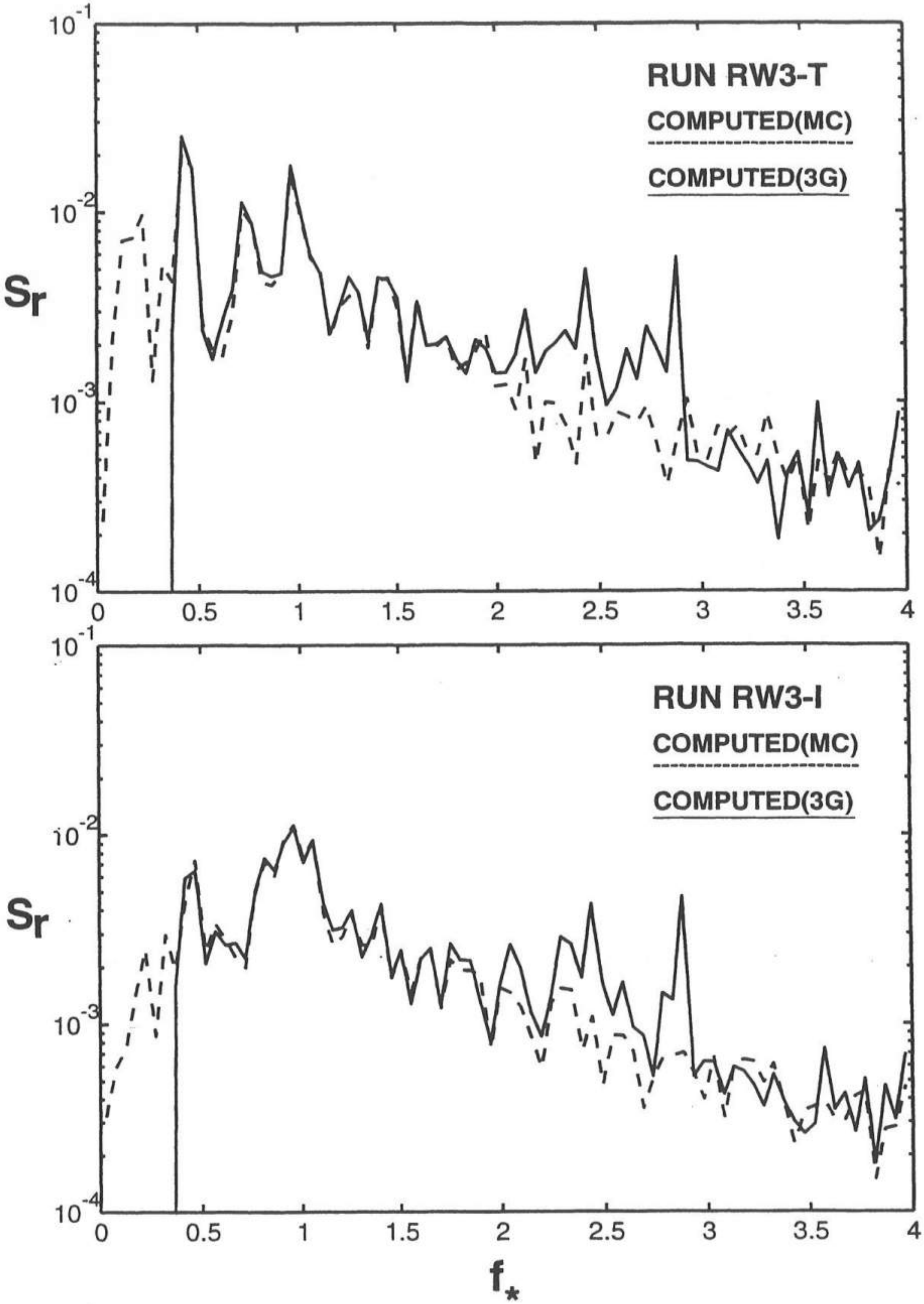

Figure 25: Computed reflected wave spectra S_r based on three gage method (3G) and method of characteristics (MC) for runs RW3-T and RW3-I.

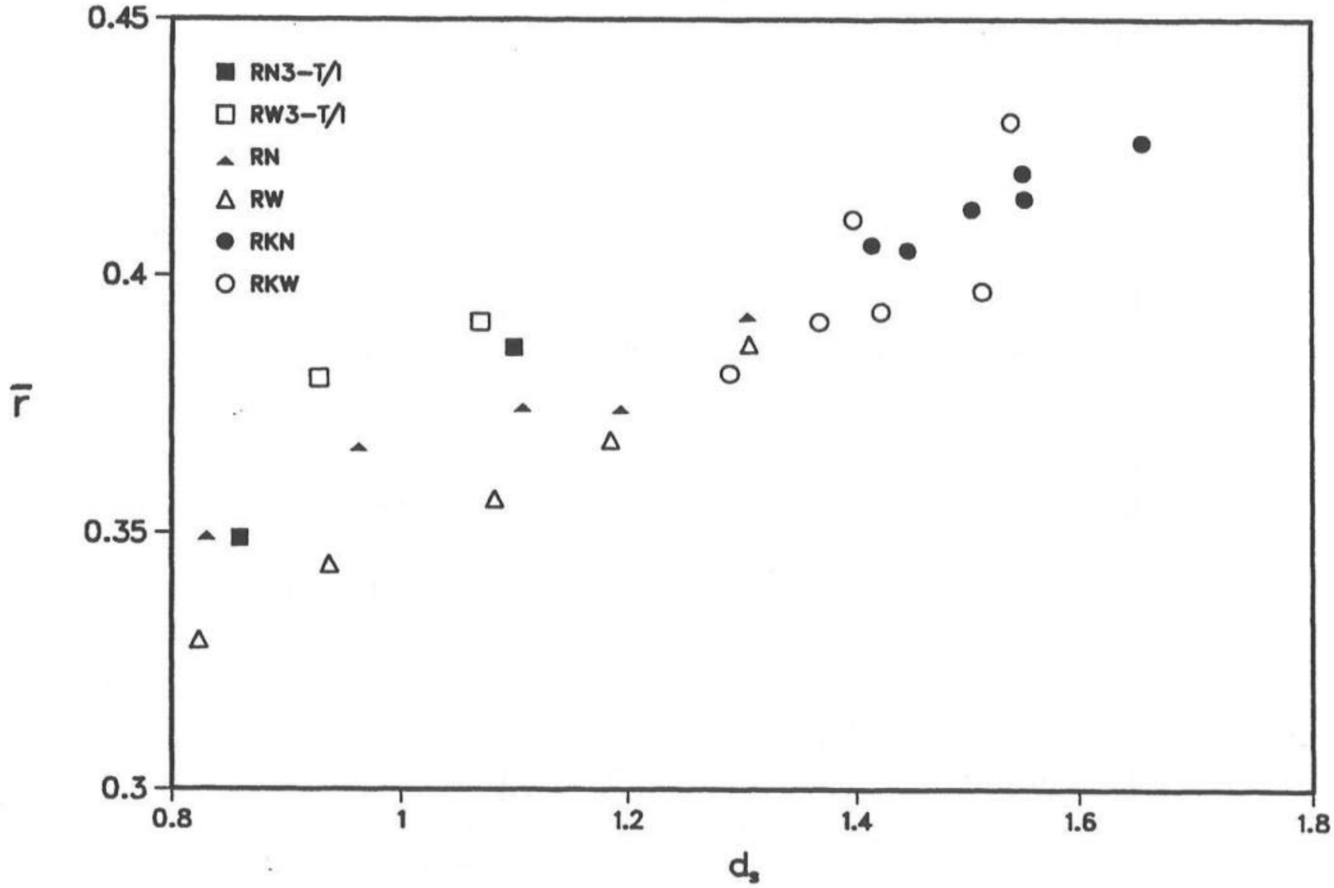

Figure 26: Measured and computed average reflection coefficient $\bar{r}$ as function of normalized toe depth $d_s = d'_s/H'_{mo}$.

5.3.5 Free Surface Oscillations at Toe of Revetment

The measured and computed free surface oscillations η_s at the toe of the 1:2 rough slope are compared in Fig. 27 for runs RN3-T and RN3-I and in Fig. 28 for runs RW3-T and RW3-I. The agreement in Figs. 27 and 28 is only qualitative for both options of IWAVE=2 and 3. The sharp peaks associated with the incident wave crests are underpredicted, while the secondary peaks associated with the reflected wave crests tend to be overpredicted.

The measured and computed spectra S_s of the free surface oscillations at the toe of the 1:2 rough slope are shown in Fig. 29 for runs RN3-T and RN3-I and in Fig. 30 for runs RW3-T and RW3-I. For IWAVE=3 where the measured total (T) wave profile at $x = 0$ is specified as input, the spectral peak and low-frequency components are overpredicted slightly, while the high-frequency components are underpredicted. For IWAVE=2 where the measured incident (I) wave train is specified as input, the spectral peak is predicted well and the low and high frequency components are underpredicted. The underprediction of the low-frequency components is caused partly by the specified incident wave trains which do not include the incident low-frequency waves as shown in Fig. 12 and 13.

5.3.6 Wave Overtopping on Crest of Revetment

The numerical model RBREAK2 is based on the assumption of constant water density. The entrained air in overtopping flow on the crest of the revetment might be assumed to increase the depth of overtopping flow without modifying the water velocity for the bottom friction factors calibrated using the measured average overtopping rates. The limited calibrations by Kobayashi and Raichle (1994) using their rough slope test data have indicated that for the overtopping flow with considerable entrained air in their small-scale experiment, the bottom friction factor $f' \simeq$ 0.01 and 0.005 for the rough and smooth slopes, respectively, and the fraction a of entrained air in volume is calibrated to be $a \simeq 0.8$. These empirical values have been adopted in the present computations for runs RN3 and RW3.

The measured normalized water depth h_o of the overtopping flow at wave gage 5 is compared with the computed water depth multiplied by $1/(1-a)$ with $a = 0.8$, which is simply called the computed water depth h_o. The measured and computed temporal variations of the water depth h_o on the crest of the revetment are shown in Fig. 31 for runs RN3-T and RN3-I and in Fig. 32 for runs RW3-T and RW3-I. It is noted that the measured depth was affected by noises. The measured and computed spectra S_o of the water depth h_o are shown in Fig. 33 for runs RN3-T and RN3-I and in Fig. 34 for runs RW3-T and RW3-I. The water depth h_o and the corresponding spectrum S_o tend to be overpredicted for runs RN3-T and RW3-T with IWAVE=3 and underpredicted for runs RN3-I and RW3-I with IWAVE=2. The appreciable low-frequency components in Figs. 33 and 34 are related to the grouped nature of overtopping events which may be identified by sharp crests in Figs. 31 and 32.

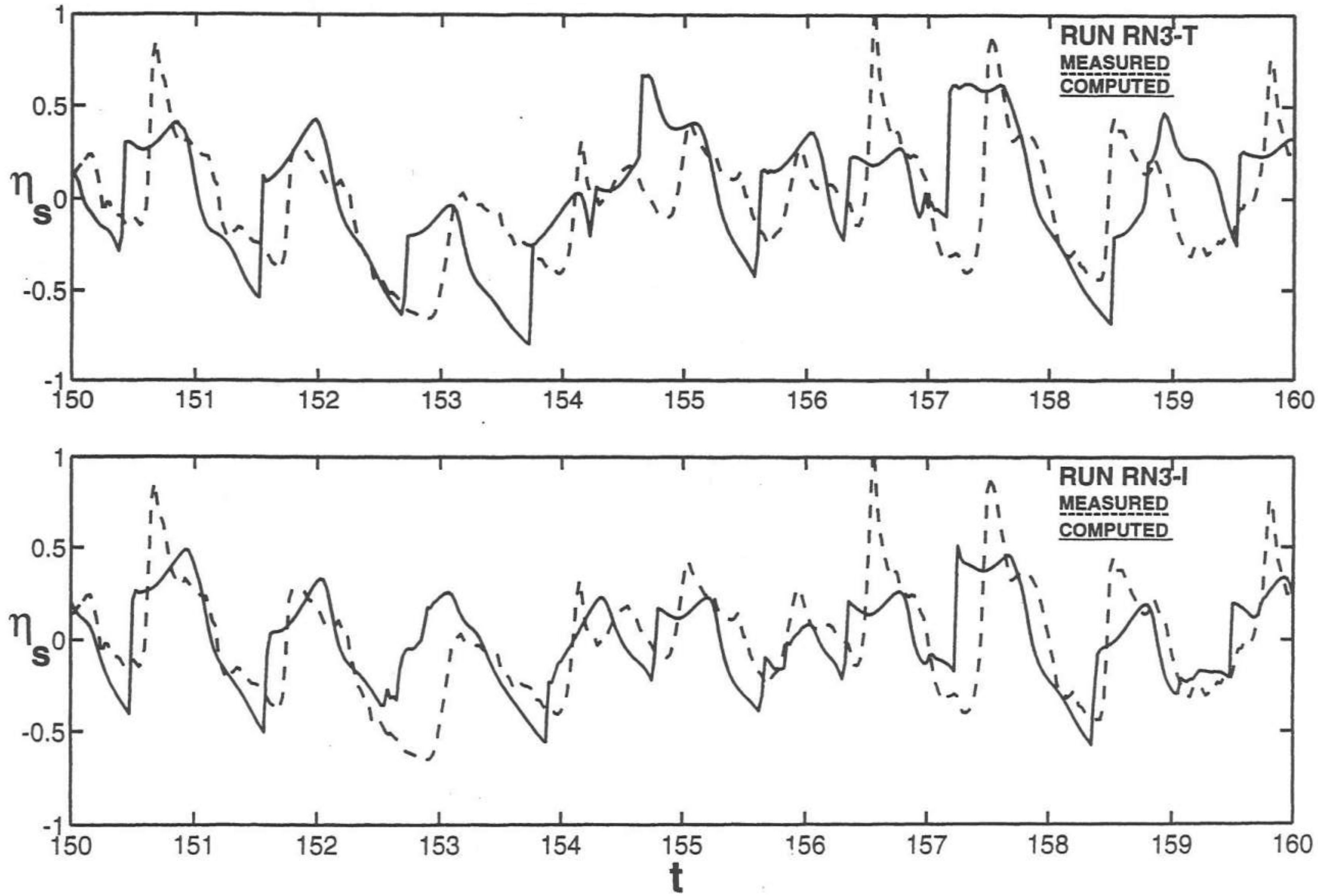

Figure 27: Measured and computed free surface oscillations η_s at toe of 1:2 rough slope for runs RN3-T and RN3-I.

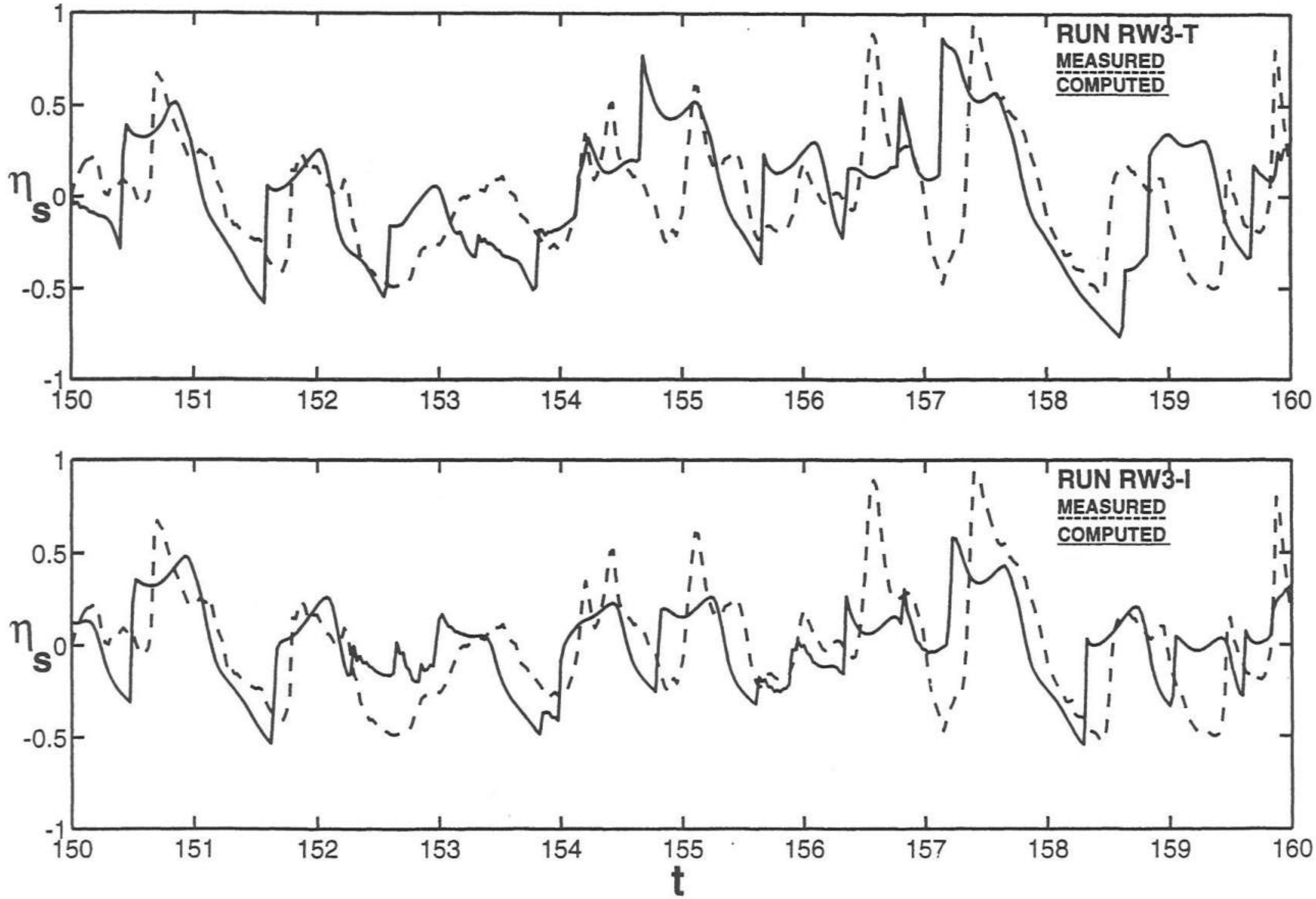

Figure 28: Measured and computed free surface oscillations η_s at toe of 1:2 rough slope for runs RW3-T and RW3-I.

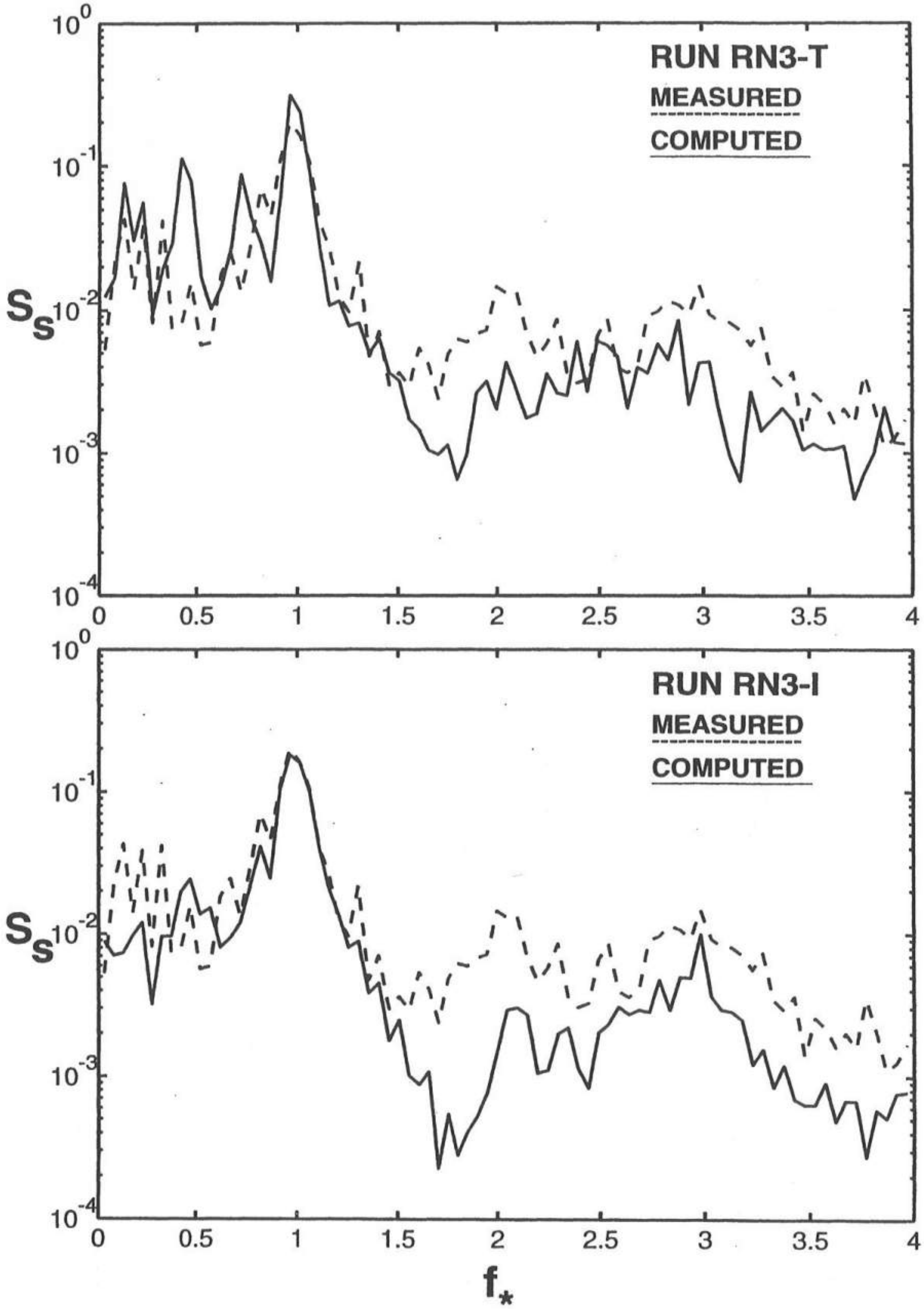

Figure 29: Measured and computed spectra S_s of free surface oscillations at toe of 1:2 rough slope for runs RN3-T and RN3-I.

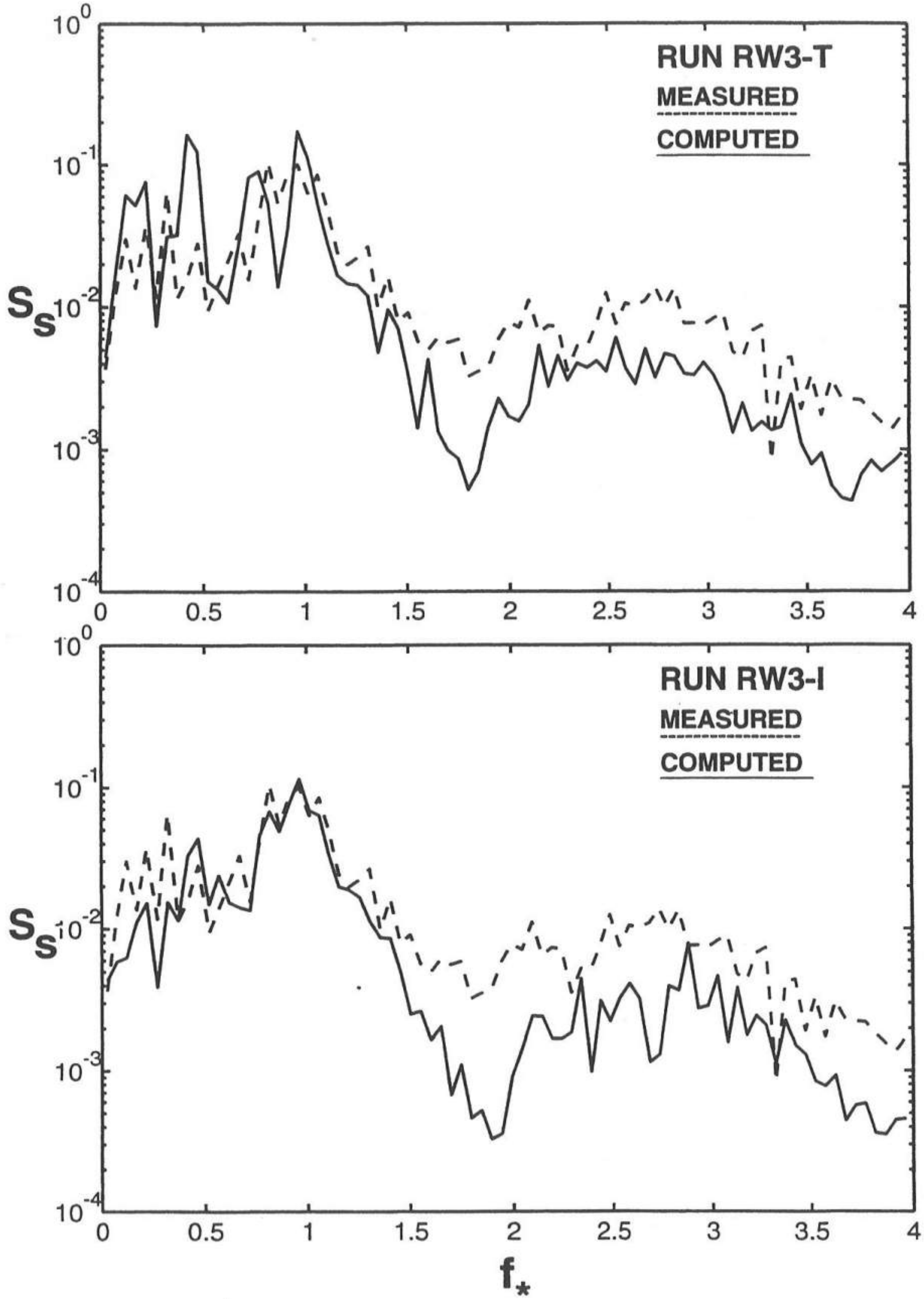

Figure 30: Measured and computed spectra S_s of free surface oscillations at toe of 1:2 rough slope for runs RW3-T and RW3-I.

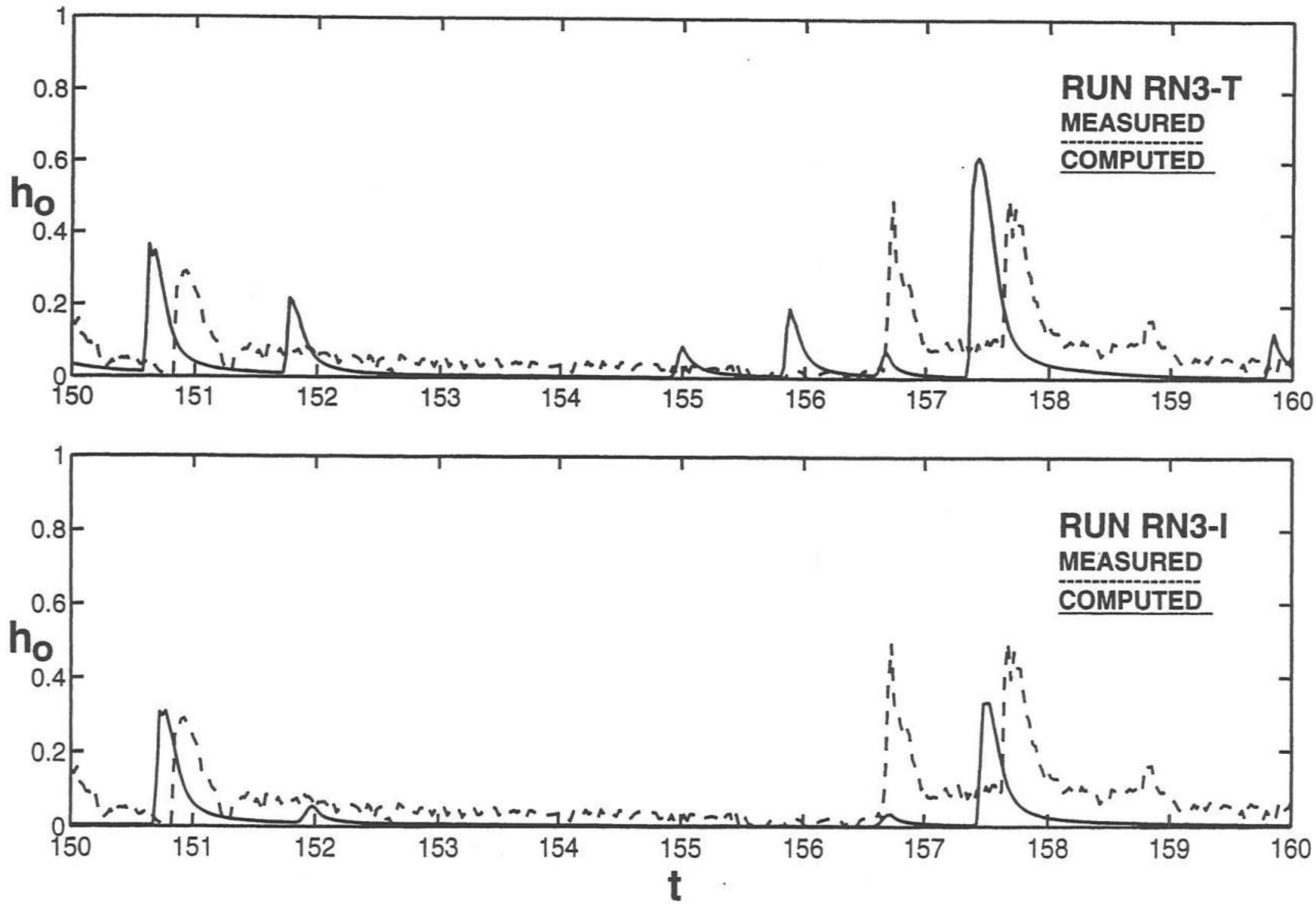

Figure 31: Measured and computed water depth h_o on crest of revetment for runs RN3-T and RN3-I.

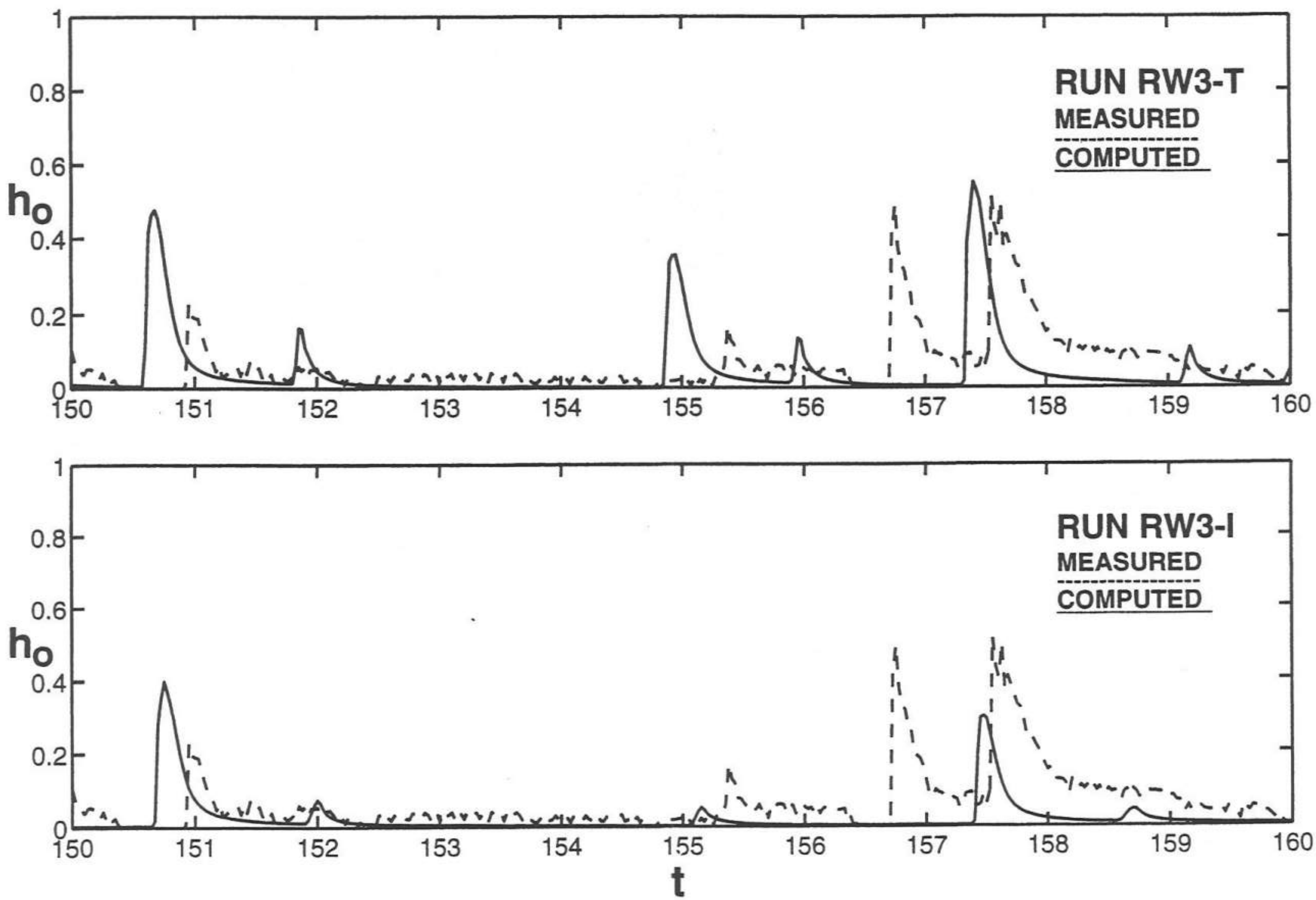

Figure 32: Measured and computed water depth h_o on crest of revetment for runs RW3-T and RW3-I.

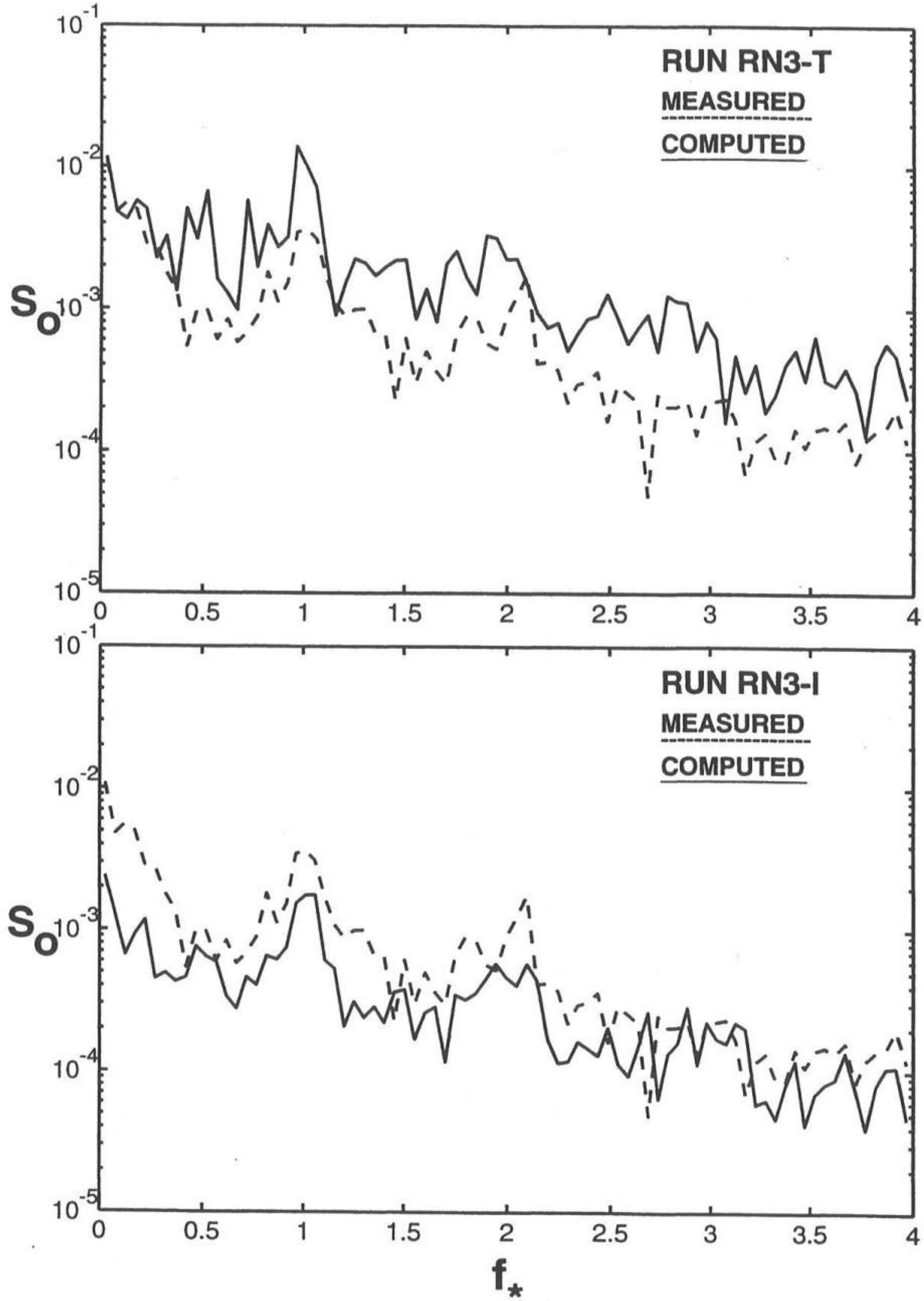

Figure 33: Measured and computed spectra S_o of overtopping flow depth h_o on crest of revetment for runs RN3-T and RN3-I.

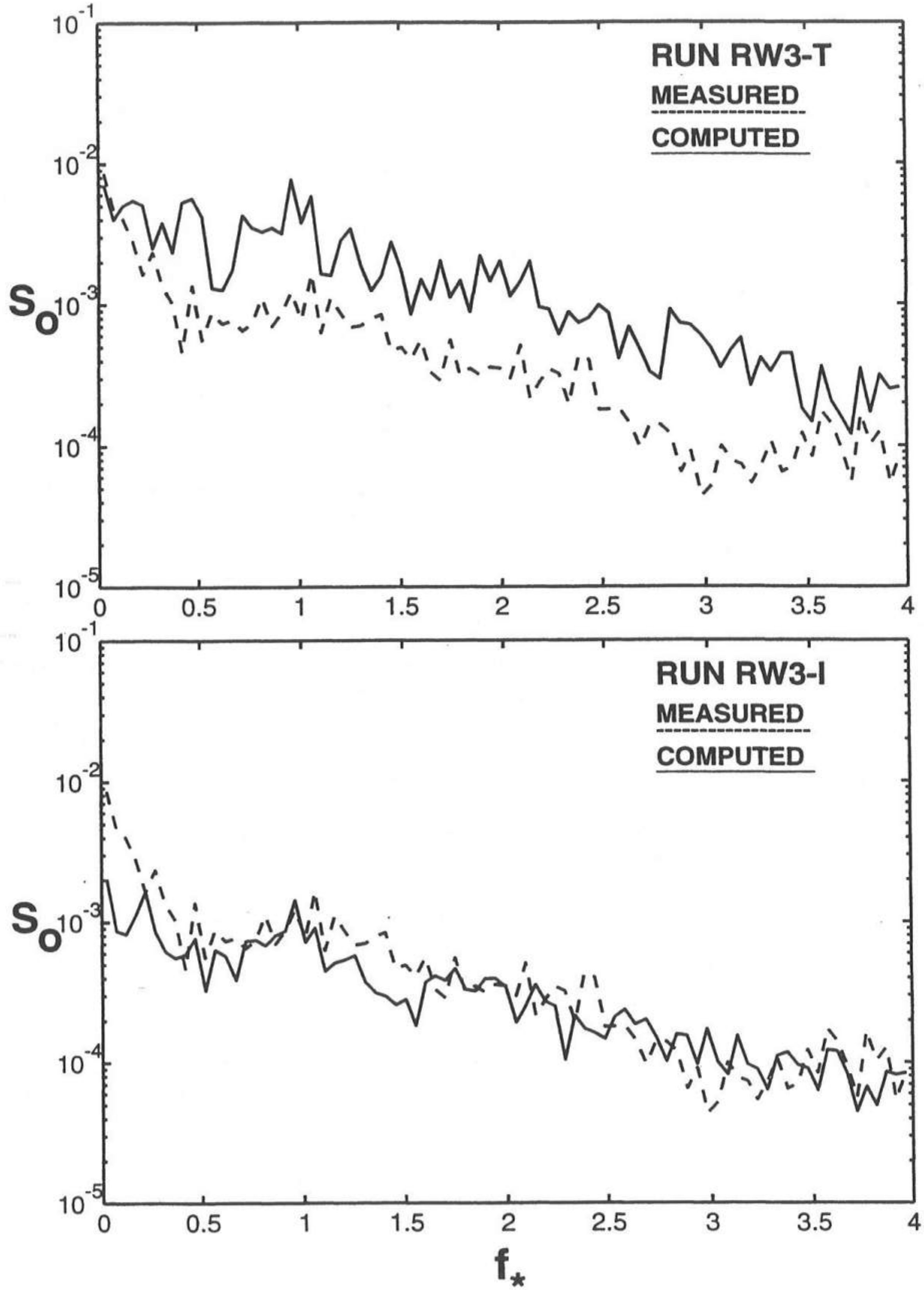

Figure 34: Measured and computed spectra S_o of overtopping flow depth h_o on crest of revetment for runs RW3-T and RW3-I.

The number of overtopping events, N_o for each run has been estimated as the number of sharp crests in the time series of the dimensional depth h'_o exceeding the minimum depth of 2 cm to reduce the effects of noises (Kobayashi and Raichle 1994). The probability of overtopping for incident N_i zero-upcrossing waves is estimated as the ratio of N_o to N_i where incident individual waves appeared to reach the revetment without being overtaken by other waves. Fig. 35 shows the probability of wave overtopping, N_o/N_i, as a function of the normalized crest height, $H_c/d_s = H'_c/d'_s$, in the same manner as in Poff and Kobayashi (1993a). The computed values of N_o/N_i indicated by RN3-T/I and RW3-T/I in Fig. 35 are those based on the three gage method (3G) in Table 10 where the values of N_i listed in Table 9 are slightly different for the three gage method and the method of characteristics (MC).

The measured values of N_o/N_i indicated by RN and RW are those for the present rough slope test runs with the narrow and wide spectra of incident waves. The data points indicated by RKN and RKW are from the rough slope tests of Raichle and Kobayashi (1993) for the narrow and wide spectra. The probability of wave overtopping for depth-limited wave conditions in shallow water decreases with the increase of the crest height of the revetment normalized by the toe depth. The probability of wave overtopping is larger (a factor of two or less) for the narrow spectra than for the wide spectra for given H_c/d_s. Table 10 indicates that the probability of wave overtopping is predicted well for runs RN3-T and RW3-T with IWAVE=3 and underpredicted for runs RN3-I and RW3-I with IWAVE=2.

The computed average overtopping rate normalized by the reference wave height H'_r as defined by Eq. 52 is obtained from the file ODOC for each run as listed in Table 12 for run RN3-T. The normalized average overtopping rates, $\bar{Q} = \overline{Q'}/\left(H'_{mo}\sqrt{gH'_{mo}}\right)$ and $\overline{Q_s} = \overline{Q'}/\left(d'_s\sqrt{gd'_s}\right)$, normalized by the spectral estimate of the incident significant wave height, H'_{mo}, and the toe depth, d'_s, respectively, are tabulated in Table 10 where the values of H'_{mo} and d'_s for each run are listed in Table 9. The computed values of $\bar{Q}$ and $\overline{Q_s}$ are overpredicted for runs RN3-T and RW3-T with IWAVE=3 and underpredicted for runs RN3-I and RW3-I with IWAVE=2. Fig. 36 shows the measured and computed average overtopping rates $\overline{Q_s}$ as a function of the normalized crest height, $H_c/d_s = H'_c/d'_s$, in the same manner as in Poff and Kobayashi (1993a). The symbols used in Fig. 36 are the same as those used in Fig. 35. For the depth-limited wave conditions in shallow water, $\overline{Q_s}$ decreases with the increase of H_c/d_s and is larger for the narrow spectra than for the wide spectra. However, $\overline{Q_s}$ is also affected by other parameters since the data points of Raichle and Kobayashi (1993) lie below the data points of the present rough slope tests.

The flow overtopping on the crest of the revetment is highly unsteady as shown in Figs. 31 and 32. The file ODOC listed in Table 12 for runs RN3-T includes the computed maximum overtopping rate q_{max} as well as the average overtopping rate $\bar{Q}$ where q_{max} is normalized in the same way as $\bar{Q}$ in Eq. 52. The value of $q_{max}/\bar{Q}$ listed in Table 13 for runs RN3-T, RN3-I, RW3-T and RW3-I is in the range $q_{max}/\bar{Q} = 42.3$–99.7. This implies that the average overtopping rate will not be sufficient to assess the severity of damage. RBREAK2 includes the option to compute armor stability and movement on overtopped revetments but has never been compared with such data.

Kobayashi and Raichle (1994) examined the time-averaged normalized wave energy balance per unit width in the computation domain $0 \le x \le x_e$ on the basis of Eq. 63 which was simply rewritten as

Table 13: Computed normalized quantities associated with wave overtopping and time-averaged wave energy balance from file ODOC where $q_{\max}$ = maximum overtopping rate; $\bar{Q}$ = average overtopping rate; F_i = net energy influx at seaward boundary; ΔE = change of specific energy in computation domain from the start to the end of the computation; L_f = rate of energy dissipation due to bottom friction in computation domain; L_B = rate of energy dissipation due to wave breaking in computation domain; F_o = energy outflux at landward edge of revetment, and H_c = normalized crest height above SWL.

RUN NO.	$\frac{q_{max}}{\overline{Q}}$	$\frac{\Delta E}{F_i}$	$\frac{L_f}{F_i}$	$\frac{L_B}{F_i}$	$\frac{F_o}{F_i}$	$\frac{\overline{Q}H_c}{F_o}$
RN3-T	42.3	0.019	0.020	0.935	0.026	0.80
RN3-I	79.5	0.018	0.021	0.949	0.012	0.88
RW3-T	55.4	0.016	0.021	0.937	0.026	0.79
RW3-I	99.7	0.012	0.020	0.957	0.011	0.86

$$F_i = \Delta E + L_f + L_B + F_o \tag{93}$$

with

$$F_o = \overline{m(0.5u^2 + \eta)} \quad \text{at } x = x_e \tag{94}$$

Comparing Eqs. 63 and 94, $F_i = \overline{E_F}(x = 0)$, $F_o = \overline{E_F}(x = x_e)$, $L_f = \int_0^{x_e} \overline{D_f} dx$, $L_B = \int_0^{x_e} \overline{D_B} dx$ and ΔE = last term on the right hand side of Eq. 63 where t_{end} = TMAX = 162.5 and $t_{\text{begin}} = 0$ because MSTAT =0 specified as input as described in Section 5.3.2. The net energy influx F_i at the seaward boundary includes the effect of the reflected waves. The specific energy change ΔE in the computation domain equals the specific energy in the computation domain at the end of the computation divided by the normalized computation duration of 162.5 because the specific energy is zero at the beginning of the computation due to the specified initial conditions of no wave motion. The energy dissipation rate L_f in the computation domain is computed using Eq. 61 for the computed depth-averaged velocity u. The energy dissipation rate L_B in the computation domain is calculated from $\overline{D_B}$ computed using Eq. 62 and is essentially the same as L_B calculated from Eq. 93 directly. The energy outflux F_o at the landward edge of the revetment computed using Eq. 94 may be approximated as $F_o > \bar{m}H_c = \bar{Q}H_c$ where the average overtopping rate $\bar{Q} = \bar{m}$ at $x = x_e$ and the normalized energy per unit water weight expressed by $(0.5\,u^2 + \eta)$ at $x = x_e$ in Eq. 94 must be larger than the normalized crest height H_c above SWL. The values of F_i, F_o, L_f, L_B, ΔE and $\bar{Q}$ can be found in the file ODOC, whereas

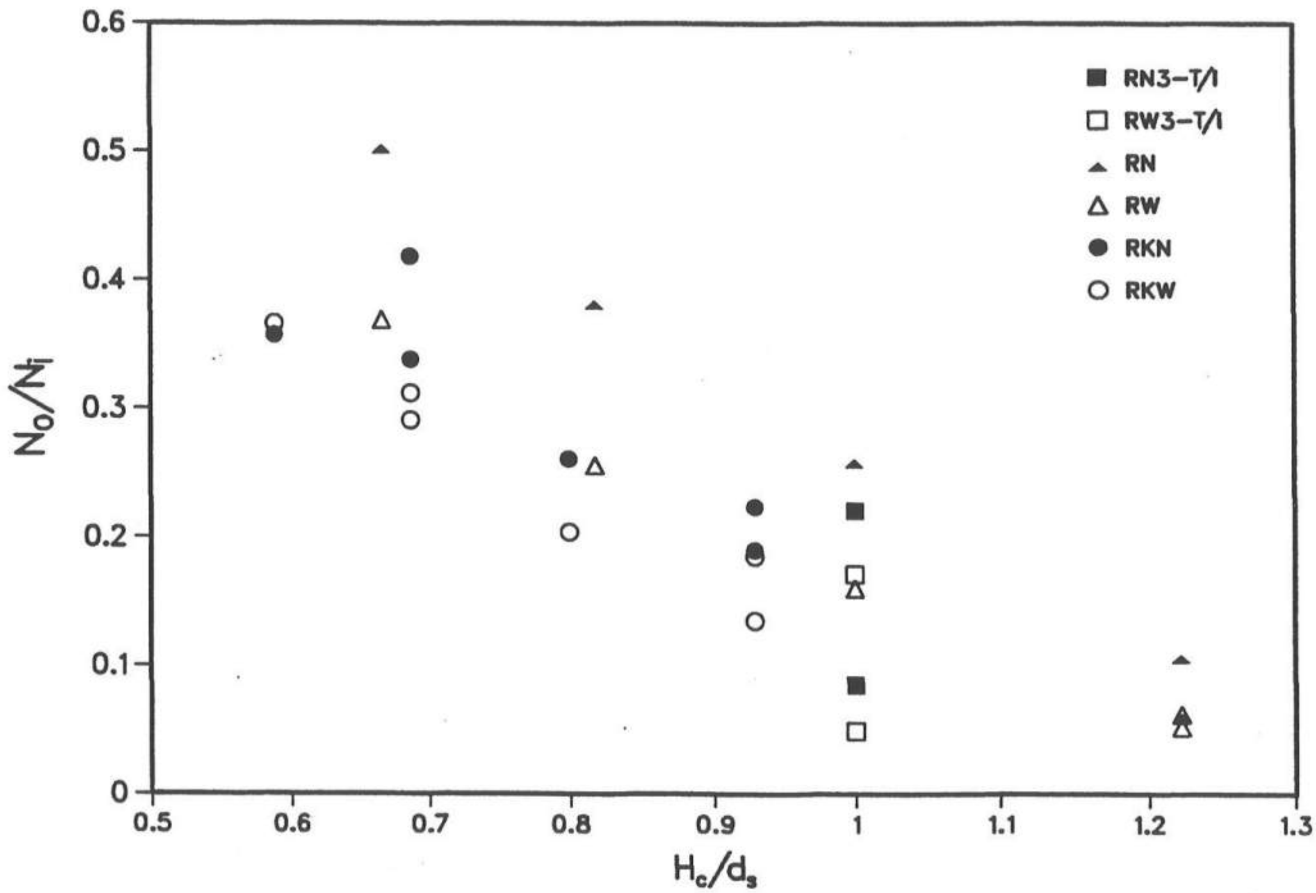

Figure 35: Probability of wave overtopping, N_o/N_i, as function of normalized crest height, $H_c/d_s = H'_c/d'_s$.

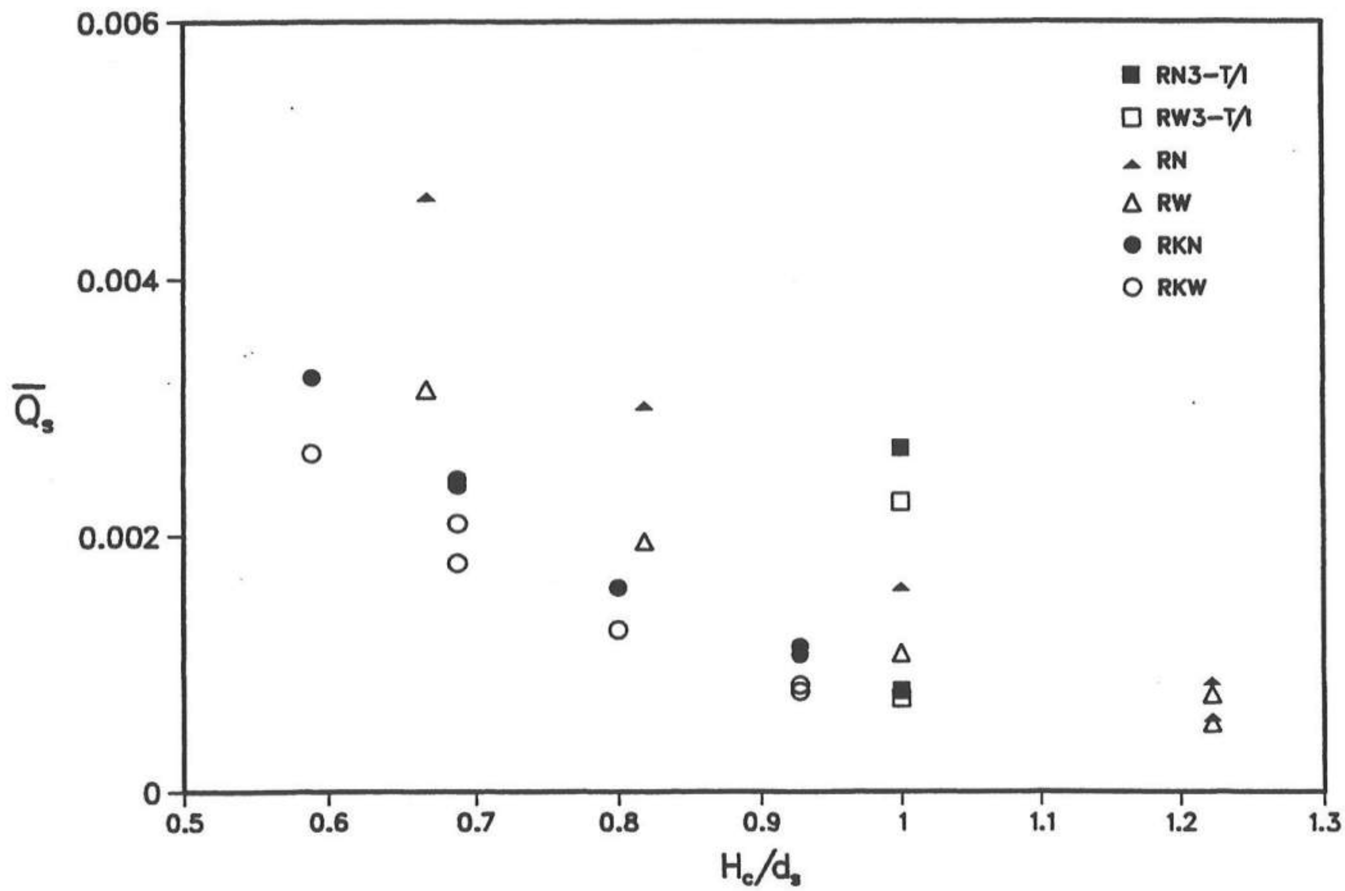

Figure 36: Normalized average overtopping rate $\overline{Q_s}$ as function of normalized crest height $H_c/d_s = H_c'/d_s'$.

$H_c = H'_c/H'_r$ with $H'_c = (20cm - d'_s)$ in this experiment as shown in Fig. 5 and H'_r = reference wave height listed in the file ODOC.

For runs RN3-T, RN3-I, RW3-T and RW3-I listed in Table 13, $\Delta E/F_i = 0.012$–0.019, $L_f/F_i = 0.020$–0.021, $L_B/F_i = 0.935$–0.957 and $F_o/F_i = 0.011$–0.026. For these runs, the net energy influx at the seaward boundary including the reflected wave energy is dissipated due to wave breaking in the computation domain. Furthermore, $\bar{Q}H_c/F_o = 0.79$–0.88 for these runs. The normalized energy per unit water weight at the landward edge of the revetment is hence dominated by the normalized potential energy η instead of the normalized kinetic energy $0.5u^2$, and the normalized free surface elevation η may be approximated by the lower bound H_c when H_c is sufficiently large. The simple relationship of $F_o \simeq \bar{Q}H_c$ indicates a linear relationship between F_o and $\bar{Q}$ as described by Kobayashi and Raichle (1994) for their computed results.

PART VI
SEPARATION OF INCIDENT AND REFLECTED WAVES

The separation of incident and reflected waves at the seaward boundary using three wave gages as shown in Fig. 5 has been performed to obtain the incident and reflected waves from the measured free surface oscillations at the three gages as well as to predict the incident and reflected waves from the computed free sruface oscillations at the three gages, which has been called the three gage method (3G) in Sections 5.3.3 and 5.3.4. The method based on two or three wave gages is widely used as summarized by Goda (1985). Alternatively, colocated pressure and current meters may be used to separate incident and reflected waves (e.g., Guza et al. 1984).

The three gage method used by Kobayashi et al. (1990) and documented by Cox et al. (1991) separated the incident and reflected wave trains at the most landward gage of the three wave gages located above the horizontal bottom. This three gage arrangement is suited for separating the incident and reflected waves at the toe of the steep slope in relatively deep water. Kobayashi and Raichle (1994) applied this method for the case where three wave gages were placed outside the surf zone above a gentle slope in front of a revetment exposed to breaking irregular waves in shallow water. In this three gage arrangement, two gages seaward of the most landward gage were situated outside the computation domain unlike the three gage arrangement shown in Fig. 5 where the incident and reflected wave trains have been obtained at the most seaward wave gage 1 located at $x' = 0$. The required modifications of the subroutines documeted by Cox et al. (1991) for the three gage arrangement shown in Fig. 5 are minor but presented in the following. It is noted that the prime used to indicate the dimensional variables is omitted in Sections 6.1 and 6.2 for simplicity and that the four subroutines explained in Section 6.2 are written using the metric system with the gravitational acceleration $g = 9.81 m/s^2$.

• 6.1 •
LINEAR THEORY

The mean water level is first removed from the measured or computed time series of duration $t_{\max}$ and the free surface displacement, η^i, is given by

$$\eta^i(t) = \eta^i(t)' - \overline{\eta^i(t)'} \text{ for } 0 \le t \le t_{\max} \tag{95}$$

where i is the gage number, $\eta^i(t)'$ is the time series before removal, and $\overline{\eta^i(t)'}$ is the mean water level. Each time series consists of N data points with N being an even number sampled at a constant rate Δt where $t_{\max} = N\Delta t$. Assuming an approximately horizontal seabed in the vicinity of the three gages located in the region $x \ge 0$ where the horizontal coordinate x is taken to be positive landward, the incident and reflected wave time series are assumed to be expressed as

$$\eta_i(x,t) = \sum_{n=1}^{N/2} \left[(a_i)_n \cos(k_n x - \omega_n t) + (b_i)_n \sin(k_n x - \omega_n t)\right] \quad \text{for } x \geq 0 \tag{96}$$

and

$$\eta_r(x,t) = \sum_{n=1}^{N/2} \left[(a_r)_n \cos(k_n x + \omega_n t) + (b_r)_n \sin(k_n x + \omega_n t)\right] \quad \text{for } x \geq 0 \tag{97}$$

in which $(a_i)_n$, $(b_i)_n$, $(a_r)_n$, and $(b_r)_n$ with $n = 1, 2, \ldots, N/2$ are the unknown coefficients for the wave trains with N data points, where $\eta_i(x,t)$ and $\eta_r(x,t)$ are the incident and reflected wave trains, respectively. The wave number k_n and the angular frequency ω_n are related by the linear dispersion relationship, and the frequency $f_n = \omega_n/(2\pi)$ is discretized as $f_n = n\Delta f$ with $\Delta f = 1/t_{\max}$ being the frequency resolution. The total free surface variation is given by

$$\eta(x,t) = \eta_i(x,t) + \eta_r(x,t) \quad \text{for } x \geq 0 \tag{98}$$

which can be written in expanded form as

$$\begin{aligned} \eta(x,t) \quad = \quad & \sum_{n=1}^{N/2} \{[(a_i)_n + (a_r)_n] \cos(k_n x) + [(b_i)_n + (b_r)_n] \sin(k_n x)\} \cos \omega_n t \\ & + \{[(b_r)_n - (b_i)_n] \cos(k_n x) + [(a_i)_n - (a_r)_n] \sin(k_n x)\} \sin \omega_n t \end{aligned} \tag{99}$$

On the other hand, the free surface oscillations are known at each gage location $x = x_i$, where i is the gage number, and is expressed as

$$\eta(x_i, t) = \sum_{n=1}^{N/2} \left[a_n^i \cos(\omega_n t) + b_n^i \sin(\omega_n t)\right] \quad \text{for } 0 \leq t \leq t_{\max} \tag{100}$$

where a_n^i and b_n^i are the Fourier coefficients computed for the know time series. Comparing Eqs. 99 and 100, the following equations must be satisfied:

$$a_n^i = [(a_i)_n + (a_r)_n] \cos k_n x_i + [(b_i)_n + (b_r)_n] \sin k_n x_i \tag{101}$$

and

$$b_n^i = [(b_r)_n - (b_i)_n] \cos k_n x_i + [(a_i)_n - (a_r)_n] \sin k_n x_i \tag{102}$$

where $n = 1, 2, \ldots, N/2$ indicates each harmonic and i indicates the location of the first gage of two gages. The second gage, j, with $x_j > x_i$ and $j > i$ located landward of the first gage yields the following equations:

$$a_n^j = [(a_i)_n + (a_r)_n] \cos k_n x_j + [(b_i)_n + (b_r)_n] \sin k_n x_j \tag{103}$$

and

$$b_n^j = [(b_r)_n - (b_i)_n] \cos k_n x_j + [(a_i)_n - (a_r)_n] \sin k_n x_j \tag{104}$$

Using these four equations, the unkonwn coefficients, $(a_i)_n$, $(b_i)_n$, $(a_r)_n$, and $(b_r)_n$ are solved in terms of the know Fourier coefficients, a_n^i, b_n^i, a_n^j and b_n^j and the gage positions, x_i and x_j.

The unknown Fourier coefficients in Eqs. 96 and 97 are thus computed from

$$(a_i)_n = \frac{-1}{2\sin k_n(x_j - x_i)}\left[-a_n^i \sin k_n x_j + a_n^j \sin k_n x_i + b_n^i \cos k_n x_j - b_n^j \cos k_n x_i\right] \quad (105)$$

$$(b_i)_n = \frac{-1}{2\sin k_n(x_j - x_i)}\left[+a_n^i \cos k_n x_j - a_n^j \cos k_n x_i + b_n^i \sin k_n x_j - b_n^j \sin k_n x_i\right] \quad (106)$$

$$(a_r)_n = \frac{-1}{2\sin k_n(x_j - x_i)}\left[-a_n^i \sin k_n x_j + a_n^j \sin k_n x_i - b_n^i \cos k_n x_j + b_n^j \cos k_n x_i\right] \quad (107)$$

$$(b_r)_n = \frac{-1}{2\sin k_n(x_j - x_i)}\left[+a_n^i \cos k_n x_j - a_n^j \cos k_n x_i - b_n^i \sin k_n x_j + b_n^j \sin k_n x_i\right] \quad (108)$$

where k_n is the wave number calculated at each frequency using the linear dispersion relation

$$(2\pi f_n)^2 = gk_n \tanh k_n d_t \quad ; \quad f_n = n\Delta f = n/t_{\max} \quad (109)$$

where d_t is the water depth below SWL at wave gage 1 located at $x_1 = 0$. An inverse transform of the Fourier coefficients gives the incident wave time series, $\eta_i(x_1, t)$, and the reflected wave time series, $\eta_r(x_1, t)$, at $x_1 = 0$, the position of the most seaward gage.

A limitation of this method is the singularity of $1/\sin k_n(x_j - x_i)$. Goda (1985) recommended that the effective frequency range of resolution should be limited to

$$\frac{\pi}{10} \le k_n(x_j - x_i) \le \frac{9\pi}{10} \quad (110)$$

For an array of three gages, there are three gage pairs and, therefore, three estimates. These estimates are averaged in the case that two or three estimates are within the cutoff criteria. The Fourier coefficients given by Eqs. 105–108 are set zero outside the range given by Eq. 110. With proper choice of gage spacing, a wide frequency band can be resolved; however, the minimum resolvable frequency $f_{\min}$ is limited by the largest gage spacing. Fourier components below this frequency limit are not resolvable. Additionally, in the higher frequencies, all three estimtaes are outside the maximum resolvable frequency $f_{\max}$. Appropriate gage spacing for given water depth and spectral peak frequency should be chosen such that most of the energy is contained in the effectgive frequency range of resolution. The minimum resolvable frequency, $f_{\min}$, is given by the linear dispersion relation

$$(2\pi f_{\min})^2 = gk_{\min} \tanh(k_{\min} d_t) \quad (111)$$

in which $k_{\min}$ is the minimum wave number given by

$$k_{\min} = \frac{\pi}{10\Delta x_{\max}} \quad (112)$$

where $\Delta x_{\max}$ is the maximum gage spacing. Similarly, the maximum resolvable frequency, $f_{\max}$, is limited by the minimum gage spacing $\Delta x_{\min}$, and is found by

$$(2\pi f_{\max})^2 = gk_{\max} \tanh(k_{\max} d_t) \quad (113)$$

with

$$k_{\max} = \frac{9\pi}{10\Delta x_{\min}} \quad (114)$$

For given d_t, $f_{\min}$ and $f_{\max}$ are hence determined by $\Delta x_{\max}$ and $\Delta x_{\min}$, respectively.

• 6.2 •

FOUR SUBROUTINES

The four subroutines used to separate the incident and reflected waves at the seaward boundary located at $x = 0$ in Fig. 5 are explained concisely. These subroutines are modified versions of the subroutines documented by Cox et al. (1991).

6.2.1 SUBROUTINE SEPIRS

This subroutine, written to separate the incident and reflected waves using the linear theory explained in Section 6.1, is called by

```
CALL SEPIRS (TS, NW, NP, DT, XG, DH, FMN, FMX, TI, TR)
```

where the arguments are defined as

- IN:
  - TS(NP,NW) = free surface oscillations at NW wave gages, $\eta^i(t)(m)$, in Eq. 95
  - NW = number of wave gages (NW = 2 or 3)
  - NP = even number of data points in the time series, N
  - DT = time step or sampling interval of the time series, $\Delta t(s)$
  - XG(NW) = location of each gage with the x-axis positive shoreward and gage number increasing shoreward, $x_i(m)$
  - DH = water depth below SWL at wave gage 1 located at $x_1 = 0$, $d_t(m)$
- OUT:
  - FMN = minimum resolvable frequency based on largest gage spacing, $f_{\min}(s^{-1})$
  - FMX = maximum resolvable frequency based on smallest gage spacing, $f_{\max}(s^{-1})$
  - TI(NP) = incident wave train at $x_1 = 0$, $\eta_i(t)(m)$
  - TR(NP) = reflected wave train at $x_1 = 0$, $\eta_r(t)(m)$
- EXTERNAL ROUTINES:
  - WAVNUM to return the wave number based on the lienar disersion relation for given frequency and water depth as explained in Section 6.2.2
  - TIMEDC to return the time series for known Fourier coefficients as explained in Section 6.2.3
  - FFTIMSL to return complex Fourier coefficients for the known time series at each gage location as explained in Section 6.2.4

The subroutine SEPIRS is based on the metric system because of the use of the gravitational acceleration $g = 9.81m/s^2$.

6.2.2 SUBROUTINE WAVNUM

For given frequency f and water depth d_t, the wave number k is determined from the lienar dispersion relation

$$(2\pi f)^2 = gk\tanh(kd_t) \tag{115}$$

In order to remove the inflection point, Eq. 115 is rewritten as (Goda 1985)

$$(kd_t) - \frac{d_t(2\pi f)^2}{g}\coth(kd_t) = 0 \tag{116}$$

which is solved using Newton's method for given f, d_t and $g = 9.81m^2/s$.

The subroutine **WAVNUM** is called by

```
CALL WAVNUM (FQ, H, WN)
```

where the arguments are defined as

- **IN**:
  - **FQ** = frequency, $f(s^{-1})$
  - **H** = water depth, $d_t(m)$
- **OUT**:
  - **WN** = wave number, $k(m^{-1})$
- **EXTERNAL ROUTINES**:
  - none

6.2.3 SUBROUTINE TIMEDC

The time series $\eta(t)$ representing $\eta_i(x = 0, t)$ and $\eta_r(x = 0, t)$ given by Eqs. 96 and 97, respectively, is expressed as

$$\eta(t) = \sum_{n=1}^{N/2} [a_n \cos(2\pi f_n t) + b_n \sin(2\pi f_n t)] \quad 0 \le t < t_{\max} \tag{117}$$

where the Fourier coefficients a_n and b_n can be found for $(a_i)_n$, $(b_i)_n$, $(a_r)_n$ and $(b_r)_n$ computed using Eqs. 105–108.

An inverse **FFT** routine is called in the subroutine **FFTIMSL** to obtai $\eta(t)$ for the known Fourier coefficients. The relations between the complex Fourier coefficients, c_n, used in the subroutine **FFTIMSL** and the real Fourier coefficients a_n and b_n used in Eq. 117 are given by

$$c_n = 0 \qquad \text{for } n = 0 \tag{118a}$$

$$c_n = \frac{a_n - ib_n}{2} \qquad \text{for } n = 1, 2, \ldots, N/2 - 1 \tag{118b}$$

$$c_n = a_n \qquad \text{for } n = N/2 \tag{118c}$$

$$c_n = \frac{a_{N-n} + ib_{N-n}}{2} \qquad \text{for } n = N/2 + 1, \ldots, N - 1 \tag{118d}$$

where $i^2 = -1$ and the integer $\mathtt{I} = (n+1), \mathtt{CN}(I) = c_n$, $\mathtt{A}(I) = a_n$ and $\mathtt{B}(I) = b_n$ is used in the subroutine TIMEDC listed in Appendix A.

The subroutine TIMEDC, which is essentially an inverse Fourier transform of known Fourier coefficients to return the time series, is called by

```
CALL TIMEDC (A, B, NP, TS)
```

where the arguments as defined as

- IN:
  - A(NP/2+1) = real Fourier coefficients, $a_n(m)$
  - B(NP/2+1) = real Fourier coefficients, $b_n(m)$
  - NP = even number of data ponts in the time series, N
- OUT:
  - TS(NP) = time series, $\eta(t)(m)$
- EXTERMAL ROUTINES:
  - FFTIMSL to return the time series for the known complex Fourier coefficients

6.2.4 SUBROUTINE FFTIMSL

The subroutine FFT2D in the IMSL library, which is widely available in USA and computationally efficient, is used to compute the complex Fourier coefficients, c_n, given by Eq. 118 where the real Fourier coefficients a_n and b_n for the time series $\eta(t)$ with zero mean are used in Eq. 117. The discrete time series η_j is expressed as

$$\eta_j = \eta(t_j) \quad \text{for} \;\; 1, 2, \ldots, N \tag{119}$$

with

$$t_j = (j-1)\Delta t \tag{120}$$

where Δt is the sampling interval and N is the even number of data points. Using this definition, Eqs. 117 and 118 can be shown to yield

$$N c_n = \sum_{j=1}^{N} \eta_j \exp\left[-\frac{2\pi i(j-1)(n-1)}{N}\right] \quad \text{for } n = 1, 2, \ldots, N \tag{121}$$

where $i^2 = -1$. Eq. 121 is in the form which allows the direct use of the subroutine FFT2D to compute c_n with $n = 1, 2, \ldots, N$. It is noted that c_n in Eq. 118 with $n = 0, 1, \ldots, N-1$ corresponds to c_{n-1} obtained from Eq. 121.

The subroutine FFT2B is the IMSL subroutine to compute the inverse Fourier transform of given coefficients c_n to find η_j with $j = 1, 2, \ldots, N$. To apply the subroutine FFT2B, Eqs. 117 and 118 are rewritten as

$$\eta_j = \sum_{n=1}^{N} c_n \exp\left[\frac{2\pi i(j-1)(n-1)}{N}\right] \quad \text{for } j = 1, 2, \ldots, N \tag{122}$$

where η_j is real and c_n in this equation corresponds to c_n in Eq. 121.

The FFTIMSL subroutine is called by

```
CALL FFTIMSL (TS, CN, NP, IO)
```

where the arguments are defined as

- IN/OUT:
  - TS(NP) = time series, $\eta(t)(m)$
  - CN(NP) = complex Fourier coefficients, $c_n(m)$
- IN:
  - NP = even number of data ponts in the time series, N
  - IO = option for an FFT(IO=+1) to return complex coefficients of known time series or for an inverse FFT (IO= -1) to return time series for known complex coefficients
- EXTERNAL ROUTINES:
  - FFT2D, an IMSL routine for FFT
  - FFT2B, an IMSL routine for inverse FFT

PART VII
SUMMARY AND CONCLUSIONS

The computer program RBREAK2 assisted by the computer program BEFORR2 simulates the interaction of normally incident irregular waves with an impermeable coastal structure fronted by a beach in the manner similar to hydraulic model tests in a wave flume. The options added to RBREAK2 include the spatial variation of the bottom friction factor to account for the cross-shore variation of bottom roughness as well as the specification of the measured total free surface oscillations as input to RBREAK2 at its seaward boundary.

The spatial variation of the bottom friction factor was added to RBREAK2 when Kobayashi and Raichle (1993, 1994) compared RBREAK2 with the experiment on breaking irregular wave overtopping over a quarry stone revetment fronted by a gentle smooth slope. A three gage method was used in this experiment to separate the incident and reflected waves immediately outside the breaker zone on the gentle slope. The separated incident wave train for each test was specified as input to RBREAK2 as has been the case with the previous comparisons with the numerical models IBREAK, RBREAK and PBREAK. The comparison of RBREAK2 with the wave overtopping experiment indicated that RBREAK2 with limited calibrations of the bottom friction factors for the rough and smooth slopes and the fraction of entrained air in overtopping flow was capable of predicating the highly unsteady overtopping flow characteristics on the crest of the revetment within errors of about ±100%. The wave overtopping rates have turned out to be sensitive to the empirical bottom friction factors and will be very difficult to predict accurately.

On the other hand, the specification of the measured total free surface oscillations as input at the seaward boundary of RBREAK2 was added to RBREAK2 when Cox *et al.* (1994) compared RBREAK2 with the measured free surface oscillations in the surf and swash zones on sand beaches from the SUPERTANK Laboratory Data Collection Project. This option arose out of necessity because the incident and reflected waves could not be separated from the available data. The comparisons of the measured and computed results for regular and irregular waves showed that the time-dependent numerical model RBREAK2 could predict the free surface oscillations in the surf and swash zones within errors of about ±20% in the same accuracy as the previous comparisons based on the incident wave trains specified as input. A conventional model based on the time-averaged momentum and energy equations for predicting the cross-shore variation of the wave setup and root-mean-square wave height was shown to be incapable of predicting swash dynamics.

Additional irregular wave overtopping experiments conducted by Poff (1993) have been reported herein since these data sets may be of use to other coastal engineers. The tests with a 1:2 smooth slope fronted by fixed and movable beaches have been compared to examine the effects of beach profile changes on irregular wave overtopping for depth-limited wave conditions in shallow water. The measured beach profile changes were affected very little by the incident wave spectral shape. The measured wave conditions remained essentially the same during the beach profile evolution. The probability and average rate of irregular wave overtopping varied by less than a factor of two as the beach profile changed. These experimental results confirm the use of fixed bed experiments as a first approximation even for depth-limited wave conditions in shallow water.

In order to further evaluate the accuracy of the options added to RBREAK2, comparisons have been made with the wave overtopping tests with a 1:2 rough slope fronted by a gentle smooth slope conducted by Poff (1993). The bottom friction factors and the fraction of entrained air in the overtopping flow calibrated by Kobayashi and Raichle (1993, 1994) have been adopted without additional calibrations where the toe depth in their tests was deeper than the tests of Poff (1993). The option of the measured total free surface oscillations as input at the seaward boundary of RBREAK2 has been compared with the previous option of the measured and separated incident wave train as input. The agreement between the measured and computed overtopping flow characteristics for both options is similar to the agreement between RBREAK2 and the tests of Kobayashi and Raichle (1993, 1994). Consequently, the specification of the measured total free surface oscillation as input is a useful option when the incident wave train at the seaward boundary can not be estimated in experiments. However, the incident wave train computed in this option is affected by errors in the computed reflected wave train. Since the accurate specification of the incident wave time series and spectrum is crucial for the design of coastal structures and the interpretations of measured and computed results, the incident waves should be measured in experiments.

RBREAK2 computes the reflected wave train at its seaward boundary using the method of characteristics combined with linear long wave theory (Kobayashi *et al.* 1989). A modified three gage method has been evaluated in the hope of improving the accuracy of the computed reflected wave train. In this method, the incident and reflected wave trains are separated at the most seaward gage and the other two gages are situated in the computation domain. The computed free surface oscillations stored at the locations of the three gages are used to compute the incident and reflected waves in the same manner as the measured incident and reflected waves obtained from the measured free surface oscillations at the three gages. The computed incident and reflected waves based on the method of characteristics and the modified three gage method have turned out to be practically the same for the four runs compared in this report. These comparisons may put some confidence in the method of characteristics employed in RBREAK2 but the task of improving the accuracy of the computed reflected waves remains to be achieved. The additional computational effort involved in the modified three gage method did not produce any improvement.

References

Ahrens, J.P., 1975. Large wave tank tests of riprap stability. *Technical Memo No. 51*, U.S. Army Coastal Engineering Research Center, Ft. Belvoir, Va.

Ahrens, J.P., and McCartney, B.L., 1975. Wave period effect on the stability of riprap. *Proc. Civil Engineering in the Oceans / III*, ASCE, 1019–1034.

Ahrens, J.P., and Martin, F.T., 1985. Wave runup formulas for smooth slopes. *J. Waterway, Port, Coastal, and Ocean Engineering*, 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 Coastal Engineering Conference*, ASCE, 281–291.

Battjes, J.A., 1974. Surf similarity. *Proc. 14th Coastal Engineering Conference*, ASCE, 466–480.

Battjes, J.A., and Van Vledder, G. Ph., 1984. Verification of Kimura's theory for wave group statistics. *Proc. 19th Coastal Engineering Conference*, ASCE, 642–648.

Bouws, E., Günther, H., Rosenthal, W., and Vincent, C.L., 1985. Similarity of the wind wave spectrum in finite depth water. 1. Spectral form. *J. Geophysical Research*, **90** (C1), 975–986.

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

Cox, D.T., Kobayashi, N., and Wurjanto, A., 1991. Computer programs for spectral and time series analyses for random waves. *Research Report No. CACR-91-06*, Center for Applied Coastal Research, Univ. of Delaware, Newark, Delaware.

Cox, D.T., Kobayashi, N., and Wurjanto, A., 1992. Irregular wave transformation processes in surf and swash zones. *Proc. 23rd Coastal Engineering Conference*, ASCE, 156–169.

Cox, D.T., Kobayashi, N. and Kriebel, D.L., 1994. Numerical model verification using SUPERTANK data in surf and swash zones. *Proc. International Conference on the Role of Large Scale Experiments in Coastal Research*, in press.

Dalrymple, R.A., 1992. Prediction of storm/normal beach profiles. *J. Waterway, Port, Coastal and Ocean Engineering*, ASCE, **118(2)**, 193–200.

Elgar, S., and Guza, R.T., 1985a. Shoaling gravity waves: comparisons between field observations, linear theory, and a nonlinear model. *J. Fluid Mechanics*, **158**, 47–70.

Elgar, S., and Guza, R.T., 1985b. Observation of bispectra of shoaling surface gravity waves. *J. Fluid Mechanics*, **161**, 425–448.

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

Guza, R.T., and Thornton, E.B., 1982. Swash oscillations on a natural beach. *J. Geophysical Research*, **87** (C1), 483–491.

Guza, R.T., Thornton, E.B., and Holman, R.A., 1984. Swash on steep and shallow beaches. *Proc. 19th Coastal Engineering Conference*, ASCE, 708–723.

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 Mechanics*, **95** (2), 323–345.

Holman, R.A., and Sallenger, A.H., Jr., 1985. Setup and swash on a natural beach. *J. Geophysical Research*, **90** (C1), 945–953.

Jonsson, I.G., 1980. A new approach to oscillatory rough turbulent boundary layers. *Ocean Engineering*, **7** (1), 109–152.

Karjadi, E.A., and Kobayashi, N., 1994. Numerical modelling of solitary wave breaking, runup and reflection. *Proc. International Symposium on Waves–Physical and Numerical Modelling*, in press.

Kobayashi, N., and Jacobs, B.K., 1985. Stability of armor units on composite slope. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **111** (5), 880–894.

Kobayashi, N., Roy, I., and Otta, A.K., 1986. Numerical simulation of wave run-up and armor stability. Paper 5088, *18th Offshore Technology Conference*, 51–56.

Kobayashi, N., and Greenwald, J.H., 1986. Prediction of wave run-up and riprap stability. *Proc. 20th Coastal Engineering Conference*, 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. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **113** (2), 171–186.

Kobayashi, N., Otta, A.K., and Roy, I., 1987. Wave reflection and run-up on rough slopes. *J. Waterway, Port, Coastal, and Ocean Engineering*, 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., 1988. Review of wave transformation and cross-shore sediment transport processes in surf zones. *J. Coastal Research*, **4** (3), 435–445.

Kobayashi, N., and Greenwald, J.H., 1988. Waterline oscillation and riprap movement. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **114** (3), 281–296.

Kobayashi, N., Strzelecki, M.S., and Wurjanto, A., 1988. Swash oscillation and resulting sediment movement. *Proc. 21st Coastal Engineering Conference*, ASCE, 1167–1181.

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

Kobayashi, N., and Wurjanto, A., 1989a. Wave overtopping on coastal structures. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **115** (2), 235–251.

Kobayashi, N., and Wurjanto, A., 1989b. Wave transmission over submerged breakwaters. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **115** (5), 662–680.

Kobayashi, N., and Wurjanto, A., 1989c. Numerical model for design of impermeable coastal structures. *Research Report No. CE-89-75*, Center for Applied Coastal Research, Univ. of Delaware, Newark, Delaware.

Kobayashi, N., and Wurjanto, A., 1989d. Armor stability on rough permeable slopes of marine structures. *Proc. 23rd Congress of International Association for Hydraulic Research*, 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. Coastal Research*, Special Issue No. **7** on Rational Design of Mound Structures, 149–166.

Kobayashi, N., Wurjanto, A., and Cox, D.T., 1990a. Irregular waves on rough permeable slopes. *J. Coastal Research*, Special Issue No. **7** on Rational Design of Mound Structures, 167–184.

Kobayashi, N., Wurjanto, A., and Cox, D.T., 1990b. Rock slopes under irregular wave attack. *Proc. 22nd Coastal Engineering Conference*, ASCE, 1306–1319.

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

Kobayashi, N., Cox, D.T., and Wurjanto, A., 1991. Permeability effects on irregular wave run-up and reflection. *J. Coastal Research*, **7** (1), 127–136.

Kobayashi, N., and Wurjanto, A., 1992a. Irregular wave setup and run-up on beaches. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **118(4)**, 368–386.

Kobayashi, N., and Wurjanto, A., 1992b. Irregular wave interaction with permeable slopes. *Proc. 23rd Coastal Engineering Conference*, ASCE, 1299–1312.

Kobayashi, N., and Karjadi, E.A., 1993. Documentation of computer program for predicting long wave run-up. *Research Report No. CACR-93-03*, Center for Applied Coastal Research, Univ. of Delaware, Newark, Delaware.

Kobayashi, N., and Raichle, A.W., 1993. Overtopping of irregular wave groups on coastal structures. *Proc. 25th Congress of International Association of Hydraulic Research*, Tokyo, Japan, **C**, 174–181.

Kobayashi, N., and Karjadi, E.A., 1994. Surf similarity parameter for breaking solitary wave runup. *J. Waterway, Port, Coastal and Ocean Engineering*, ASCE, **120(6)**, in press.

Kobayashi, N., and Raichle, A.W., 1994. Irregular wave overtopping of revetments in surf zones. *J. Waterway, Port, Coastal and Ocean Engineering*, ASCE, **120(1)**, 56–73.

LeMéhauté, B., and Wang, J.D., 1982. Wave spectrum changes on sloped beach. *J. Waterway, Port, Coastal and Ocean Engineering*, ASCE, **108(1)**, 33–47.

Lissev, N., 1993. Influence of the core configuration on the stability of berm breakwaters. *Rept. No. R-6-93*, Dept. of Structural Engineering, Norwegian Institute of Technology, Trondheim, Norway.

Liu, Z., Elgar, S., and Guza, R.T., 1993. Group of ocean waves: Linear theory, approximations to linear theory, and observations. *J. Waterway, Port, Coastal and Ocean Engineering*, ASCE, **119(2)**, 144–159.

Losada, M.A., Kobayashi, N., and Martin, F.L., 1992. Armor stability on submerged breakwaters. *J. Waterway, Port, Coastal and Ocean Engineering*, ASCE, **118(2)**, 207–212.

Madsen, O.S., and White, S.M., 1976. Energy dissipation on a rough slope. *J. Waterways, Harbors and Coastal Engineering Division*, ASCE, **102** (WW1), 31–48.

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

Mase, H., and Kobayashi, N., 1993. Low-frequency swash oscillation. *J. Japan Society of Civil Engineers*, **461(II-22)**, 49–57.

Moretti, G., 1987. Computation of flows with shocks. *Ann. Rev. Fluid Mechanics*, **19**, 313–337.

Packwood, A.R., 1980. *Surf and run-up on beaches.* Ph.D. Thesis. School of Mathematics, Univ. of Bristol, United Kingdom.

Peregrine, D.H., 1983. Breaking waves on beaches. *Ann. Rev. Fluid Mechanics*, **15**, 149–178.

Poff, M.T., 1993. Experiments on irregular wave overtopping over smooth and rough slopes in surf zones. Thesis for Master of Civil Engrg., Univ. of Delaware, Newark, Delaware.

Poff, M.T., and Kobayashi, N., 1993a. Effects of toe depth and slope roughness on wave overtopping. *Proc. Hilton Head International Coastal Symposium*, **1**, 115–120.

Poff, M.T., and Kobayashi, N., 1993b. Computer program for refraction of directional random waves. *Research Report No. CACR-93-04*, Center for Applied Coastal Research, University of Delaware, Newark, Delaware.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., 1986. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, Cambridge, United Kingdom.

Raichle, A.W., and Kobayashi, N., 1993. Irregular wave overtopping over coastal structures. *Proc. Hilton Head International Coastal Symposium*, **1**, 121–126.

Raubenheimer, B., Guza, R.T., Elgar, S., and Kobayashi, N., 1994. Swash on a gently sloping beach. *J. Geophysical Research*, submitted.

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

Saville, T., Jr., 1955. Laboratory data on wave runup and overtopping on shore structures. *Technical 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. *Technical Report 80-1*, U.S. Army Coastal Engineering Research Center, 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 Engineer 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-Kjær, O., 1972. Shoaling of cnoidal waves. *Proc. 13th Coastal Engineering Conference*, ASCE, 365–383.

Synolakis, C.E., 1987. The runup of solitary waves. *J. Fluid Mech*, **185**, 523–545.

Thompson, A.C., 1988. Numerical model of breakwater wave flows. *Proc. 21st Coastal Engineering Conference*, ASCE, 2013–2027.

Tørum, A., 1992. Wave induced water particle velocities and forces on an armor unit on a berm breakwater. *Report for MAST-G6 Project 3B*, SINTEF Norwegian Hydrotechnical Lab., Trondheim, Norway.

Van der Meer, J.A., 1987. Stability of breakwater armor layers—Design formulae. *Coastal Engineering*, **11**, 219–239.

Van der Meer, J.W., 1988. Deterministic and probabilistic design of breakwater armor layers. *J. Waterway, Port, Coastal, and Ocean Engineering*, 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 Coastal Engineering Conference*, ASCE, 191–204.

Van Gent, M.R.A., 1994. The modelling of wave action on and in coastal structures. *Coastal Engineering*, **22**, 311–339.

Wurjanto, A., and Kobayashi, N., 1991. Numerical model for random waves on impermeable coastal structures and beaches. *Research Report No. CACR-91-05*, Center for Applied Coastal Research, University of Delaware, Newark, Delaware.

Wurjanto, A., and Kobayashi, N., 1992. Numerical model for random waves on permeable coastal structures. *Research Report No. CACR-92-02*, Center for Applied Coastal Research, University of Delaware, Newark, Delaware.

Wurjanto, A., and Kobayashi, N., 1993. Irregular wave reflection and runup on permeable slopes. *J. Waterway, Port, Coastal and Ocean Engineering*, ASCE, **119(5)**, 537–557.

Appendix A

Listing of Four Subroutines Used to Separate Incident and Reflected Waves

```
C======================================================================C
C      SEPIRS
C
C      SEPARATES INCIDENT AND REFLECTED WAVE TRAINS ON GENTLE SLOPE
C      USING THREE WAVE GAGES.  THIS SUBROUTINE IS A MODIFIED VERSION OF
C      THE SUBROUTINE IRSORT IN COX ET AL. (1991) FOR THE THREE WAVE
C      GAGE ARRANGEMENT USED BY POFF AND KOBAYASHI (1993).
C
C         IN:
C      TS(NPTS,NW)..FREE SURFACE OSCILLATIONS (AT NW GAGES) (M)
C      NW...........WIDTH OF TIME SERIES ARRAY (EQUAL TO NO. OF GAGES)
C      NP...........EVEN NUMBER OF DATA POINTS IN TIME SERIES
C      DT...........TIME STEP (SAMPLING INTERVAL) (S)
C      XG(NW).......LOCATION OF EACH GAGE WITH X-AXIS POSITIVE SHOREWARD
C                   AND GAGE NUMBER INCREASING SHOREWARD (M)
C      DH...........WATER DEPTH AT GAGE 1 LOCATED AT X=0 (M)
C
C         OUT:
C      FMN..........MINUMUM RESOLVABLE FREQUENCY (HZ) BASED ON LARGEST
C                   GAGE SPACING
C      FMX..........MAXIMUM RESOLVABLE FREQUENCY (HZ) BASED ON SMALLEST
C                   GAGE SPACING
C      TI(NP).......INCIDENT TIME SERIES AT X=0 (M)
C      TR(NP).......REFLECTED TIME SERIES AT X=0 (M)
C
C      EXTERNAL ROUTINES: FFTIMSL, WAVNUM, TIMEDC
C
C======================================================================C
      SUBROUTINE SEPIRS (TS,NW,NP,DT,XG,DH,FMN,FMX,TI,TR)
C
      PARAMETER (NPTS=7000)
      PARAMETER (NDS=16000)
      PARAMETER (NWS=3)
      DIMENSION TS(NPTS,NW), XG(NW), TI(NPTS),TR(NPTS)
      DIMENSION AI(NDS/2+1),AR(NDS/2+1),BI(NDS/2+1),BR(NDS/2+1)
      DIMENSION A(NDS/2+1,NWS),B(NDS/2+1,NWS), XI(NWS,NWS), XJ(NWS,NWS)
      COMPLEX   CN(NDS)

      TM=NP*DT
      DF=1./TM
      NH=NP/2+1
      GRAV=9.81
      PI=4.*ATAN(1.)
      CMIN=0.1*PI
      CMAX=0.9*PI
C                               SET THE CORRECT GAGE LOCATION SO THAT
```

```
C XI(I,J) IS THE LOCATION OF GAGE I SEAWARD OF GAGE J AND XJ(I,J) IS
C THE LOCATION OF GAGE J LANDWARD OF GAGE I AND XJ(I,J)-XI(I,J) > 0
      DO 10 I=1,NW
         DO 5 J=1,NW
              XI(I,J) = 0.0
              XJ(I,J) = 0.0
5        CONTINUE
10    CONTINUE
      XI(1,2) = XG(1)
      XJ(1,2) = XG(2)
      IF(NW.GT.2)THEN
      XI(1,3) = XG(1)
      XJ(1,3) = XG(3)
      XI(2,3) = XG(2)
      XJ(2,3) = XG(3)
      ENDIF
C                                  GET FOURIER COEFFICIENTS
      DO 550 I = 1, NW
         CALL FFTIMSL(TS(1,I),CN,NP,+1)
         A(1,I) = 0.0
         B(1,I) = 0.0
         DO 525 K = 2,NH-1
              A(K,I) = 2.*REAL(CN(K))
              B(K,I) = -2.*AIMAG(CN(K))
525      CONTINUE
         A(NH,I) = REAL(CN(NH))
         B(NH,I) = 0.0
550   CONTINUE
C                                  LOOP TO SORT INC. AND REFL. A'S,B'S
      DO 910 L = 2, NH
         FQ = (L-1) * DF
         CALL WAVNUM(FQ,DH,WVNM)
         KOUNT = 0
         AI(L) = 0.0
         AR(L) = 0.0
         BI(L) = 0.0
         BR(L) = 0.0
         DO 880 I = 1, NW
              DO 870 J = 1, NW
                   IF (I .LT. J) THEN
C                                  CRITERION FOR .1PI < KX < .9PI
                        ARGIS = WVNM * (XJ(I,J) - XI(I,J))
                        IF (ARGIS.GE.CMIN.AND.ARGIS.LE.CMAX) THEN
                             KOUNT = KOUNT + 1
                             SI = SIN(WVNM * XI(I,J))
                             SJ = SIN(WVNM * XJ(I,J))
                             CI = COS(WVNM * XI(I,J))
```

```
                              CJ = COS(WVNM * XJ(I,J))
                              D1 = -0.5 / SIN(ARGIS)
                              D2 = A(L,I) * SJ
                              D3 = A(L,J) * SI
                              D4 = A(L,I) * CJ
                              D5 = A(L,J) * CI
                              D6 = B(L,I) * SJ
                              D7 = B(L,J) * SI
                              D8 = B(L,I) * CJ
                              D9 = B(L,J) * CI
                              AI(L) = AI(L) + D1*(-D2+D3+D8-D9)
                              BI(L) = BI(L) + D1*(+D4-D5+D6-D7)
                              AR(L) = AR(L) + D1*(-D2+D3-D8+D9)
                              BR(L) = BR(L) + D1*(+D4-D5-D6+D7)
                         ENDIF
                    ENDIF
870            CONTINUE
880       CONTINUE
          IF (KOUNT .NE. 0) THEN
               AI(L) =        AI(L) / FLOAT(KOUNT)
               BI(L) = -1. * BI(L) / FLOAT(KOUNT)
               AR(L) =        AR(L) / FLOAT(KOUNT)
               BR(L) =        BR(L) / FLOAT(KOUNT)
          ENDIF
910   CONTINUE
C                                  SET ZERO-TH HARMONIC TO ZERO
      AI(1) = 0.0
      BI(1) = 0.0
      AR(1) = 0.0
      BR(1) = 0.0
C                                  RESOLVABLE FREQUENCY RANGE
C                                  FIND MAX AND MIN GAGE PAIR
      XMIN=XG(2)-XG(1)
      XMAX=XMIN
      IF(NW.GT.2)THEN
         XMAX=XG(3)-XG(1)
         IF(XG(3)-XG(2) .LT. XMIN) XMIN=XG(3)-XG(2)
      ENDIF
C                                  FMN IS FROM LARGEST GAGE PAIR
      WNMIN = CMIN/XMAX
      FMN = SQRT(WNMIN*GRAV*TANH(WNMIN*DH)) / (2.*PI)
C                                  FMX IS FROM SMALLEST GAGE PAIR
      WNMAX = CMAX/XMIN
      FMX = SQRT(WNMAX*GRAV*TANH(WNMAX*DH)) / (2.*PI)
C                                  TO GET TIME SERIES AT X=0.0
      CALL TIMEDC(AI,BI,NP,TI)
      CALL TIMEDC(AR,BR,NP,TR)
```

```
C
      RETURN
      END
C======================================================================C

C======================================================================C
C     WAVNUM
C
C     COMPUTES WAVE NUMBER, K=2PI/L, FOLLOWING LINEAR WAVE THEORY
C     THIS SUBROUTINE IS AN IMPROVED VERSION OF THE SUBROUTINE WAVNUM
C     IN COX ET AL. (1991).  NOTE THAT DIMENSION IS 1/METER
C
C        IN:
C     FQ...........FREQUENCY (HZ)
C     H............WATER DAPTH (M)
C
C        OUT:
C     WN...........WAVE NUMBER (1/M)
C======================================================================C
      SUBROUTINE WAVNUM (FQ ,H, WN)
C
        PARAMETER(G=9.81)
        PI=4.*ATAN(1.)
        SIG=2.*PI*FQ
        R=SIG**2*H/G
        IF(R.GE.10.)THEN
         WN=R/H
         GO TO 200
        ENDIF
        IF(R.LE.1.E-8)THEN
         WN=SQRT(R)/H
         GO TO 200
        ENDIF
        IF(R.GE.1.)THEN
         X=R
        ELSE
         X=SQRT(R)
        ENDIF
 300    COTH=1./TANH(X)
        XP=X-((X-R*COTH)/(1.+R*(COTH**2-1.)))
        D=ABS(1.-XP/X)
        X=XP
        IF(D.GT.1.E-6) GO TO 300
        WN=X/H
 200   CONTINUE

       RETURN
```

```
      END
C=====================================================================C

C=====================================================================C
C     TIMEDC
C
C     DETERMINISTIC COEFFICIENT SCHEME TO COMPUTE TIME SERIES FOR
C     GIVEN FOURIER COEFFICIENTS FROM COX ET AL. (1991).
C
C        IN:
C     A(NP/2+1)....FOURIER COEFFICIENTS FOR COSINE (M)
C     B(NP/2+1)....FOURIER COEFFICIENTS FOR SINE (M)
C     NP...........EVEN NUMBER OF DATA POINTS IN TIME SERIES
C
C        OUT:
C     TS(NP).......TIME SERIES SOLUTION (M)
C
C        EXTERNAL ROUTINE:
C     FFTIMSL......INVERSE FOURIER TRANSFORM (IMSL) FOR IO= -1
C=====================================================================C
      SUBROUTINE TIMEDC (A, B, NP, TS)
C
      PARAMETER (NPTS=7000)
      PARAMETER (NDS=16000)
      REAL      TS(NPTS), A(NDS/2+1), B(NDS/2+1)
      COMPLEX   CN(NDS)
C
      NH = NP/2+1
C                                 AVERAGE VALUE SHOULD BE ZERO
      CN(1) = CMPLX(0.0, 0.0)
C                                 FILL FIRST PART OF COMPLEX ARRAY
      DO 500 I = 2, NH-1
         CN(I) = 0.5 * CMPLX(A(I),-B(I))
500   CONTINUE
C                                 AT NYQUIST FREQUENCY, B(NYQ) IS ZERO
      CN(NH) = CMPLX(A(NH), 0.0)
C                                 FILL SECOND PART OF COMPLEX ARRAY
      DO 600 I = NH+1, NP
         CN(I) = 0.5 * CMPLX(A(NP-I+2),B(NP-I+2))
600   CONTINUE
C                                 INVERSE TRANSFORM TO RETURN TIME SERIE
      CALL FFTIMSL(TS,CN,NP,-1)
C
      RETURN
      END
C=====================================================================C
```

```
C=====================================================================C
C     FFTIMSL
C
C     FAST FOURIER TRANSFOMR USING IMSL ROUTINES FROM COX ET AL. (1991)
C
C        IN/OUT:
C     TS(NP).......TIME SERIES (M)
C     CN(NP).......COMPLEX FOURIER COEFFICIENTS (M)
C
C        IN:
C     NP...........EVEN NUMBER OF DATA POINTS
C     IO...........+1 THEN FOURIER TRANS OF TS AND CN RETURNED
C                  -1 THEN INVERSE TRANS OF CN AND TS RETURNED
C
C        EXTERNAL ROUTINES:
C     FFT2D........FAST FOURIER TRANSFORM (IMSL)
C     FFT2B........INVERSE FAST FOURIER TRANSFORM (IMSL)
C=====================================================================C
      SUBROUTINE FFTIMSL (TS, CN, NP, IO)
C
      PARAMETER (NPTS=7000)
      PARAMETER (NDS=16000)
      REAL       TS(NPTS)
      COMPLEX    CN(NDS)
      COMPLEX    CTS(NDS,1), COEF(NDS,1), AFFT(NDS,1)
C                                 IF IO IS +1 THEN FFT OF TIME SERIES
      IF (IO .EQ. 1) THEN
C                                 CHANGE TO 2-D ARRAY FOR IMSL
      DO 100 I = 1,NP
         CTS(I,1) = CMPLX(TS(I), 0.0)
100   CONTINUE
      NRA = NP
      NCA = 1
      LDA = NDS
      LDCOEF = NDS
      CALL FFT2D (NRA, NCA, CTS, LDA, COEF, LDCOEF)
C
      DO 200 I = 1,NP
         CN(I) = 1.0/FLOAT(NP) * COEF(I,1)
200   CONTINUE
C                                 IF IO IS -1 THEN INVERSE FFT OF CN'S
      ELSEIF (IO .EQ. -1) THEN
C                                 CHANGE TO 2-D ARRAY FOR IMSL
      DO 300 I = 1,NP
         COEF(I,1) = CN(I)
300   CONTINUE
      NRCOEF = NP
```

```
      NCCOEF = 1
      LDCOEF = NDS
      LDA = NDS
      CALL FFT2B(NRCOEF, NCCOEF, COEF, LDCOEF, AFFT, LDA)
C                                TAKE REAL PART FOR TIME SERIES
      DO 400 I = 1, NP
         TS(I) = REAL(AFFT(I,1))
400   CONTINUE
      ENDIF
C
      RETURN
      END
C======================================================================C
```

Appendix B

Listing of Computer Program RBREAK2

```
C=====================================================================C
C
C     #######  #######  #######  ########  ######  ##    ##
C     ##    ## ##    ## ##    ## ##       ##    ## ##   ##
C     ##    ## ##    ## ##    ## ##       ##    ## ##  ##
C     #######  #######  #######  #######  ######## ## ##
C     ##  ##   ##    ## ##  ##   ##       ##    ## ######
C     ##   ##  ##    ## ##   ##  ##       ##    ## ###  ##
C     ##    ## #######  ##    ## ######## ##    ## ##    ##
C
C
C                       #####
C                     #########
C                   #####    #####
C                  ####     #####
C                          #####
C                        #####
C                      #####
C                    #####
C                  #################
C                  #################
C
C
C
C                      Numerical Simulation of
C        Random Waves on Impermeable Breakwaters and Beaches;
C        Computer Program RBREAK has been extended as follows:
C
C       1. Variation of bottom friction factor f' for each linear
C          segment of bottom profile
C       2. Specification of measured free surface oscillation at
C          seaward boundary (x=0) for IWAVE=3
C
C      Written by N. Kobayashi, A. Raichle, D. Cox, and M. Poff
C                Center for Applied Coastal Research
C                  Department of Civil Engineering
C           University of Delaware, Newark, Delaware 19716
C                         September  1993
C
C ############################## NOTES ##############################
C
C     To minimize numerical difficulties, especially at computational
C     waterline, RBREAK2 should be preceded by BEFORR2, which determines
C     two crucial computation parameters for each computation unit:
C         1. NONE  = number of time steps in one wave period
C         2. DELTA = normalized water depth at computational waterline
C     Values of NONE and DELTA obtained by BEFORR2 are stored in file
C         'BINPUT' and are to be read by RBREAK2.
C     About "computation unit":
C       . RBREAK2 and BEFORR2 break computation duration into computa-
C         tion units. Each computation unit is one wave period long
C         except for the last unit, the length of which may be less
```

```
C        than one wave period.
C      . The "wave period" is the reference period used for the
C        normalization of the governing equations.
C     RBREAK and BEFORR share many subroutine and COMMON block names,
C        but they are independent programs.
C
C ###################### MAIN PROGRAM RBREAK2 ######################
C
C     Main program performs time-marching computation using
C     subroutines
C
      PROGRAM RBREAK2
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI,KCNO,MCNO,KC2
      CHARACTER*10 FINP1,FINP2
      INTEGER S,SM1,SP1
      DIMENSION VDUM(N1)
C
C     COMMONs
C     ---------------------------------------------------------------
C      Name     Contents
C     ---------------------------------------------------------------
C     /DIMENS/  The values of FORTRAN PARAMETERs specified in the main
C               program
C     /CONSTA/  Basic constants
C     /ID/      Identifiers specifying user's options
C     /CPAR1/ to /CPAR4/ Computation parameters
C     /REQ1/ and /REQ2/ Identifiers and values associated with special
C               storing (see Subr. 02 INPUT1 for more explanation)
C     /WAVE1/   Physical wave parameters
C     /WAVE2/   Dimensionless wave parameters
C     /WAVE3/   Norm. free surface elevations as a function of time
C     /WAVE4/   Maximum and minimum of ETA in /WAVE3/
C     /WAVE5/   Cnoidal wave parameters (K, E, m and 1-m)
C     /WAVE6/   Number of data points specified in input file FINP2
C     /BOT1/    Physical quantities related to structure
C     /BOT2/    Normalized quantities related to structure
C     /BOT3/    Normalized structure geometry
C     /BOT4/ and /BOT5/ Physical 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     /STT1/ and /STT2/ Statistical quantities
C     /ENERG/   Quantities related to wave energy
C     /ARMOR1/  Parameters, read as input, for computation of armor
C               stability or movement
C     /ARMOR2/ and /ARMOR3/ Computed parameters for computation of
C               armor stability or movement
```

```
C     /ARMOR4/ and /ARMOR5/ Parameters and quantities associated with
C              computation of armor stability
C     /ARMOR6/ and /ARMOR7/ Parameters and quantities associated with
C              computation of armor movement
C     /SAVBC/  Node numbers related to options ISAVB=1 and ISAVC=1
C     /FILBC/  File names related to options ISAVB=1 and ISAVC=1
C     /VALUEN/ Values at present time level stored during computation
C     ------------------------------------------------------------------
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  MULTIF,NONEM,NONM,NONE,NEND,NRATE,NHOP
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /REQ1/   IREQ,IELEV,IV,IDUDT,ISNR,NREQ
      COMMON /REQ2/   TREQ(N5)
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,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/  NDATA,NDAT
      COMMON /BOT1/   DSEAP,DLANDP,FWP(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3),
     +                RMEAN(N3)
      COMMON /OVER/   OV(4),OVMEAN
      COMMON /COEFS/  RCOEF(3),TCOEF(3)
      COMMON /STT1/   ELSTAT(3,3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1),
     +                ELMEAN(3),U1MEAN(N1),EMEAN(N1),VMEAN(N1)
      COMMON /STT2/   TSTAT1,TSTAT2
      COMMON /ENERG/  ENER(7,N1),ENERB(15),ENTEMP(3,N1)
      COMMON /ARMOR1/ C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /ARMOR2/ SG1,CTAN(N1)
      COMMON /ARMOR3/ JSTAB,JSTABM
      COMMON /ARMOR4/ JSNSC,JATMIN
      COMMON /ARMOR5/ CSTAB1,CSTAB2,AMAXS,AMINS,E2,E3PRE(N1),SNR(N1),
     +                SNSX(N1),TSNSX(N1),ATMIN(4,N1),SNSC,TSNSC
      COMMON /ARMOR6/ NMOVE,NSTOP,ISTATE(N1),NODFI(N1)
      COMMON /ARMOR7/ CSTAB3,CSTAB4,CM1,DA,SIGDA,WEIG,
     +                VA(N1),XAA(N1),XA(N1),TDIS(N1)
      COMMON /SAVBC/  NNODB,NNODC,NODB(N5),NODC(N5)
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
```

```
C
C          READ INPUT
C          ----------------------------------------------------------
C          Subr. 01 OPENIO opens input and output files
C          Subr. 02 INPUT1 reads most of input data
C          Subr. 03 INPUT2 reads wave profile at seaward boundary
C          Subr. 49 STOPP  stops execution of the computation
           ----------------------------------------------------------
      WRITE (*,*) 'RBREAK2 reports progress on screen every MREP waves'
      WRITE (*,*) 'Enter MREP (0 if you do not want the report)'
      READ  (*,*)  MREP
      WRITE (*,*) 'Name of primary input-data-file?'
      READ  (*,5000) FINP1
C
      CALL OPENIO (1,0,FINP1,FINP2)
      CALL INPUT1 (1,0,FINP2)
      CALL OPENIO (2,0,FINP1,FINP2)
      IF (IWAVE.GT.1) CALL INPUT2 (1)
      IF (MREP.LT.0)  CALL STOPP (1,1)
 5000 FORMAT (A10)
C
C           CHECK FORTRAN PARAMETERS
C           ---------------------------------------------------------
C           Each subroutine having FORTRAN "PARAMETER" statement is
C           called by Subroutine 48 CKSUBR to check its "PARAMETER"
C           values, which must be the same as their counterparts in
C           the main program. These checking calls to subroutines
C           are indicated by MODE=0, as opposed to "working" calls
C           indicated by MODE=1 as used above in Call INPUT1 (1,...)
C           and Call INPUT2 (1).
C           ---------------------------------------------------------
      CALL CKSUBR (N1,N2,N3,N4,N5)
      MODE=1
C
C           GROUNDWORK
C           ---------------------------------------------------------
C         . Subr. 04 BOTTOM computes normalized structure geometry
C         . Subr. 05 WPARAM calculates wave parameters
C         . This call to Subr. 10 CPARAM checks case-signature NONM
C         . Subr. 40 SPARAM calculates parameters needed for compu-
C           tation of armor stability or movement
C         . Subr. 26 INIT specifies initial conditions
C         . This call to Subr. 06 REGWAV is for initial incident
C           wave profile and documentation purpose in Subr. 43 DOC1
C           where NONE=NONEM is used
C         . Subr. 43 DOC1 writes essential information before
C           time-marching computation
C           ---------------------------------------------------------
      CALL BOTTOM (MODE)
      CALL WPARAM (MODE)
      CALL CPARAM (0)
```

```
      IF (ISTAB.GT.0) CALL SPARAM (MODE)
      CALL INIT (MODE)
      IF (IWAVE.EQ.1) CALL REGWAV (MODE)
      CALL DOC1 (MODE)
      IF (IJOB.EQ.3) SM1=INITS-1
C
C             TIME MARCHING COMPUTATION
C             --------------------------------------------------------
C
      DO 550 M = 1,MWAVE
C
C       ------: 550 Begins :------------------------------------------
C             : There are MWAVE computation units to execute
C
C       Subr. 28 INSTAT initializes quantities used in Subr. 29 SVSTAT
C       Subr. 02 INPUT1 reads computation parameters from file 'BINPUT'
C       Subr. 06 REGWAV computes incident regular wave profile
C       Subr. 10 CPARAM calculates computation parameters which are
C                    dependent on NONE
C
        IF (M.GT.MSTAT) CALL INSTAT (MODE)
        NONEOL=NONE
        CALL INPUT1 (9,M,FINP2)
        IF (NONE.NE.NONEOL) THEN
          CALL CPARAM (MODE)
          IF (IWAVE.EQ.1) CALL REGWAV (MODE)
        ENDIF
        IF (ISAVB.EQ.1.OR.ISAVC.EQ.1) CALL OPENIO (3,M,FINP1,FINP2)
C
        DO 540 N = 1,NEND
C
C         ----: 540 Begins :------------------------------------------
C             : There are NEND time levels to execute
C
C       . TIME = normalized time
C       . Subr. 12 EXTRAP estimates hydrodynamic quantities at the
C         node immediately landward of the present waterline node
C       . Subr. 13 RETAIN retains some values at present time level
C         at landward and seaward boundaries
C       . Subr. 14 CRITV calculates critical velocities used in
C         characteristic variables and checks for negative water
C         depth
C       . Subr. 15 MARCH performs time-marching from present time
C         level (N-1) to next (N) excluding landward and seaward
C         boundaries, which are handled by Subr. 16 LANDBC and
C         19 SEABC, respectively
C       . Subr. 11 NUMSTA checks if numerical stability criterion
C         is violated
C
          TIME = DBLE(M-1) + DBLE(N)/DBLE(NONE)
          IF (IJOB.LT.3) CALL EXTRAP (MODE)
          CALL RETAIN (MODE)
```

```
      CALL CRITV  (MODE,M)
      CALL MARCH  (MODE,M)
      CALL LANDBC (MODE,M,N,ETAT)
      CALL SEABC  (MODE,M,N,ETAI,ETAR)
      CALL NUMSTA (MODE,M)
C
C     ----: Computation of Energy Quantities :--------------------
C           : Performed by Subr. 30 ENERGY excluding the first MSTAT
C             computation units
C
      IF (IENERG.EQ.1) THEN
        IF (IWAVE.GT.1.AND.N.EQ.NEND) THEN
          IF (M.EQ.MSTAT.OR.M.EQ.MWAVE) CALL ENERGY (9,M)
        ENDIF
        IF (M.GT.MSTAT) CALL ENERGY (1,M)
      ENDIF
C
C     ----: Statistical Calculations :----------------------------
C           : Statistical calculations are performed excluding the
C             first MSTAT computation units
C
C    . Subr. 41 VECMAT creates a vector from a row of a matrix
C    . Subr. 35 STAT1  finds mean, maximum, and minimum
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 the period when statistical calculations
C         are performed
C         (For IJOB=3, JMAX=INITS)
C
      IF (M.GT.MSTAT) THEN
        CALL VECMAT (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     ----: Armor Stability or Movement :-------------------------
C           : Computation is performed excluding the first MSTAB
C             computation units
C
C   :: ISTAB=1: INITIATION OF MOVEMENT OF ARMOR UNITS
C      . SNR at every node is computed by Subr. 31 STABNO
C      . SNR(j) = stability number against rolling/sliding at
C        node j
C   :: ISTAB=2: SLIDING MOTION OF ARMOR UNITS
```

```
C          . Individual armor units are tracked by Subr. 32 MOVE
C          . NMOVE = number of units dislodged from their initial
C            locations
C          . NSTOP = number of units stopped after moving
C          . XAA(j),XA(j) = displacement of moving or stopped armor
C            unit number j from its initial location, normalized by
C            TP*sqrt(GRAV*HREFP) and DAP, respectively
C
         IF (ISTAB.GT.0.AND.M.GT.MSTAB) THEN
           IF (JSTAB.GT.JSTABM) JSTABM=JSTAB
           IF (ISTAB.EQ.1) THEN
             CALL STABNO (MODE)
           ELSE
             CALL MOVE (MODE)
           ENDIF
         ENDIF
C
C        ----: Documentation :---------------------------------------
C
C      . Subr. 44 DOC2 documents computed results at designated
C        moments
C      . Calling DOC2(1,...) is for storing "A", i.e., spatial
C        variations of hydrodynamic quantities as specified in Subr.
C        02 INPUT1
C      . Calling DOC2(2,...) is for storing time series at specified
C        nodes at rate NRATE per wave period where NHOP=NONE/NRATE
C        in Subr. 10 CPARAM
C      . Calling DOC2(3,...) is for storing spatial variations at
C        specified normalized time t=TREQ(i) with i=1,2,...,NREQ
C
         IF (M.GE.MSAVA1.AND.M.LE.MSAVA2) THEN
           DO 110 K = 1,NTIMES
             NSAVA = K*NONE/NTIMES
             IF (N.EQ.NSAVA) CALL DOC2 (1,DUM,DUM,DUM)
  110      CONTINUE
         ENDIF
         IDUM = MOD(N,NHOP)
         IF (IDUM.EQ.0) CALL DOC2 (2,ETAI,ETAR,ETAT)
         IF (IREQ.EQ.1) THEN
           DO 120 I = 1,NREQ
             ITREQ = INT(TREQ(I))
             RES   = DMOD(TREQ(I),1.D+00)
             IF (DABS(RES).LT.1.D-08) THEN
               MRQ = ITREQ
               NRQ = NONE
             ELSE
               MRQ = ITREQ+1
               NRQ = RES*NONE
             ENDIF
             IF (M.EQ.MRQ.AND.N.EQ.NRQ) CALL DOC2 (3,DUM,DUM,DUM)
  120      CONTINUE
         ENDIF
```

```
C
C          ----: 540 Ends :-------------------------------------------
C
  540    CONTINUE
C
C        Subr. 29 SVSTAT saves quantities related to time-averaged
C        values
C
         IF (M.GT.MSTAT) CALL SVSTAT (MODE)
         IF (MREP.GT.0) THEN
           IDUM = MOD(M,MREP)
           IF (IDUM.EQ.0) THEN
             IF (M.EQ.1) THEN
               WRITE (*,*) ' Finished ',M,' Wave Period'
             ELSE
               WRITE (*,*) ' Finished ',M,' Wave Periods'
             ENDIF
           ENDIF
         ENDIF
C
C        ------: 550 Ends :-------------------------------------------
C
  550 CONTINUE
C
C              FINISHING
C              -------------------------------------------------------
C              Subr. 37 STATC  completes statistical calculations
C              Subr. 39 ENERGC completes computation of energy qtties.
C              Subr. 45 DOC3   writes results
C              -------------------------------------------------------
      CALL STATC (MODE)
      IF (IENERG.EQ.1) CALL ENERGC (MODE)
      CALL DOC3 (MODE)
C
      STOP
      END
C
C -------------------- END OF MAIN PROGRAM RBREAK2 -----------------
C #01####################### SUBROUTINE OPENIO ###############RBREAK#
C
C     This subroutine opens input and output files
C
      SUBROUTINE OPENIO (ICALL,M,FINP1,FINP2)
C
      COMMON /ID/      IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                 ISAVA,ISAVB,ISAVC
      COMMON /REQ1/    IREQ,IELEV,IV,IDUDT,ISNR,NREQ
      CHARACTER*10 FINP1,FINP2,FSAV1B(20),FSAV2B(20),FSAVEC(20)
      DATA FSAV1B /
     1 'OSAV1B01  ','OSAV1B02  ','OSAV1B03  ','OSAV1B04  ',
     2 'OSAV1B05  ','OSAV1B06  ','OSAV1B07  ','OSAV1B08  ',
     3 'OSAV1B09  ','OSAV1B10  ','OSAV1B11  ','OSAV1B12  ',
```

```
     4 'OSAV1B13  ','OSAV1B14  ','OSAV1B15  ','OSAV1B16  ',
     5 'OSAV1B17  ','OSAV1B18  ','OSAV1B19  ','OSAV1B20  '/
      DATA FSAV2B /
     1 'OSAV2B01  ','OSAV2B02  ','OSAV2B03  ','OSAV2B04  ',
     2 'OSAV2B05  ','OSAV2B06  ','OSAV2B07  ','OSAV2B08  ',
     3 'OSAV2B09  ','OSAV2B10  ','OSAV2B11  ','OSAV2B12  ',
     4 'OSAV2B13  ','OSAV2B14  ','OSAV2B15  ','OSAV2B16  ',
     5 'OSAV2B17  ','OSAV2B18  ','OSAV2B19  ','OSAV2B20  '/
      DATA FSAVEC /
     1 'OSAVEC01  ','OSAVEC02  ','OSAVEC03  ','OSAVEC04  ',
     2 'OSAVEC05  ','OSAVEC06  ','OSAVEC07  ','OSAVEC08  ',
     3 'OSAVEC09  ','OSAVEC10  ','OSAVEC11  ','OSAVEC12  ',
     4 'OSAVEC13  ','OSAVEC14  ','OSAVEC15  ','OSAVEC16  ',
     5 'OSAVEC17  ','OSAVEC18  ','OSAVEC19  ','OSAVEC20  '/
C .
C  DESCRIPTION OF INPUT AND OUTPUT FILES
C  ---------------------------------------------------------------------
C  (Category) Unit (Filename) Purpose
C  ---------------------------------------------------------------------
C  (Input)   11 (FINP1)     contains primary input data
C  (Input)   12 (FINP2)     contains time series of normalized free
C                           surface elevation at seaward boundary if
C                           IWAVE>1
C  (Input)   14 ('BINPUT')  contains computation parameters assigned
C                           by BEFORR2 for RBREAK2 computation
C                           --> m,NONE(m),NEND(m),DELTA(m)
C                               m = 1,2,...,MWAVE = computation unit
C  (Output) 51 ('OSTAT')    stores statistics of hydrodyn. quantities
C                           --> JMAX
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  (Output) 52 ('OSPACE')   stores structure geometry
C                           --> JE,(XB(j),ZB(j),j=1,JE)
C                           and "A" (see Subr. 02 INPUT1 for descrip-
C                           tion of "A")
C  (Output) 53 ('OENERG')   stores energy quantities if IENERG=1
C                           --> JMAX,(ENER(i,j),j=1,JMAX)
C                               i=1,2,...,7
C  (Output) 54 ('OSTAB')    stores quantities associated with armor
C                           stability or movement
C  (Output) 55 ('OREQ')     stores specially-requested quantities
C  (Output) 61 ('OSEAWAV') stores time series of hydrodyn. quantities
C                           at seaward boundary
C  (Output) 62 ('ORUNUP')   stores time series of runup (IJOB<3)
C  (Output) 63 ('OOVER')    stores time series of hydrodyn. quantities
C                           at landward boundary (IJOB=2)
C  (Output) 64 ('OTRANS')   stores time series of hydrodyn. quantities
C                           at landward boundary (IJOB=3)
C  (Output) 71 (FSAV1B)     stores time series of volume flux at
C                           specified nodes (one of "B")
```

```
C  (Output) 72 (FSAV2B)     stores time series of water depth at
C                            specified nodes (the other "B")
C  (Output) 73 (FSAVEC)     stores time series of "C", i.e., time
C                            series of either of the following:
C                            . for ISTAB=1, stability number SNR at
C                              specified nodes
C                            . for ISTAB=2, displacement of armor units
C                              from specified initial nodal locations
C  (Output) 98 ('ODOC')     stores essential output for concise
C                            documentation
C  (Output) 99 ('OMSG')     stores messages written under special
C                            circumstances during computation
C  -------------------------------------------------------------------
      IF (ICALL.EQ.1) THEN
        CALL OPENF (11,FINP1,'OLD')
        CALL OPENF (14,'BINPUT    ','OLD')
        CALL OPENF (51,'OSTAT     ','NEW')
        CALL OPENF (52,'OSPACE    ','NEW')
        CALL OPENF (61,'OSEAWAV   ','NEW')
        CALL OPENF (98,'ODOC      ','NEW')
        CALL OPENF (99,'OMSG      ','NEW')
      ELSEIF (ICALL.EQ.2) THEN
        IF (IJOB.LT.3) THEN
          CALL OPENF (62,'ORUNUP    ','NEW')
          IF (IJOB.EQ.2) CALL OPENF (63,'OOVER     ','NEW')
        ELSE
          CALL OPENF (64,'OTRANS    ','NEW')
        ENDIF
        IF (IWAVE.GT.1)  CALL OPENF (12,FINP2,'OLD')
        IF (IENERG.EQ.1) CALL OPENF (53,'OENERG    ','NEW')
        IF (ISTAB.GT.0)  CALL OPENF (54,'OSTAB     ','NEW')
        IF (IREQ.EQ.1)   CALL OPENF (55,'OREQ      ','NEW')
      ELSE
C       --------------------------------------------------------------
C       Storing "B" and "C" is storing a number of time series, each
C         being associated with a specified node, simultaneously. For
C         computation with long duration, a single output file may be
C         too large to store. Therefore, time series are stored in
C         groups of 100 wave periods, i.e., 100 wave periods to an
C         output file.
C       "B" time series of the first kind (volume flux) are stored in
C         files 'OSAV1B01' (the first 100 waves), 'OSAV1B02' (the
C         second 100 waves), and so on (under variable FSAV1B and
C         unit number 71).
C       "B" time series of the second kind (water depth) are stored in
C         files 'OSAV2B01', 'OSAV2B02', and so on (under variable
C         FSAV2B and unit number 72).
C       "C" time series are stored in files 'OSAVEC01', 'OSAVEC02',
C         and so on (under variable FSAVEC and unit number 73).
C       In the following, the opening and closing of applicable output
C         files are performed every 100 wave periods.
C       --------------------------------------------------------------
```

```
        IDUM = MOD(M,100)
        IF (IDUM.EQ.1) THEN
          MPACK = M/100 + 1
          IF (ISAVB.EQ.1) THEN
C             ::     For storing "B"
            IF (M.GT.1) THEN
              CLOSE (71)
              CLOSE (72)
            ENDIF
            CALL OPENF (71,FSAV1B(MPACK),'NEW')
            CALL OPENF (72,FSAV2B(MPACK),'NEW')
          ENDIF
          IF (ISAVC.EQ.1) THEN
C             ::     For storing "C"
            IF (M.GT.1) CLOSE (73)
            CALL OPENF (73,FSAVEC(MPACK),'NEW')
          ENDIF
        ENDIF
      ENDIF
C
      RETURN
      END
C
C -01------------------- END OF SUBROUTINE OPENIO ------------RBREAK-
C #02####################### SUBROUTINE INPUT1  ###############RBREAK#
C
C     This subroutine reads most of input data
C
C     Notes:
C    . User should run RBREAK2 after a successful run of BEFORR2 using
C     exactly the same input data set.
C    . User-chosen options and most of user-specified values are not
C     checked in RBREAK2, assuming that they are unchanged from the
C     corresponding BEFORR2 computation. Only values related to arrays'
C     dimensions are checked again since RBREAK2 and BEFORR2 may have
C     different dimension arrangement.
C
      SUBROUTINE INPUT1 (MODE,M,FINP2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      CHARACTER*10 FINP2,FID2
      CHARACTER*5 COMMEN(15)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  MULTIF,NONEM,NONM,NONE,NEND,NRATE,NHOP
      COMMON /CPAR3/  INITS,JE,JE1
```

```
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /REQ1/   IREQ,IELEV,IV,IDUDT,ISNR,NREQ
      COMMON /REQ2/   TREQ(N5)
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /WAVE6/  NDATA,NDAT
      COMMON /BOT1/   DSEAP,DLANDP,FWP(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      COMMON /BOT4/   NBSEG
      COMMON /BOT5/   WBSEG(N4),TBSLOP(N4),XBSEG(N4),ZBSEG(N4)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3),
     +                RMEAN(N3)
      COMMON /ARMOR1/ C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /SAVBC/  NNODB,NNODC,NODB(N5),NODC(N5)
      IF (MODE.EQ.0) THEN
C       ::   MODE=0 is used in Subr. 48 CKSUBR
C       ::   Although not used in any of the above COMMON blocks,
C            N1 and N2 need to be checked in relation with
C            Call CKVAL (5,...) and checking NONEMA in this subroutine
        CALL CKFPAR (2,1,N1,N1R)
        CALL CKFPAR (2,2,N2,N2R)
        CALL CKFPAR (2,3,N3,N3R)
        CALL CKFPAR (2,4,N4,N4R)
        CALL CKFPAR (2,5,N5,N5R)
        RETURN
      ELSEIF (MODE.EQ.9) THEN
C       ::   MODE=9 is called from the main program
C       ::   RBREAK2 reads NONE and DELTA from file 'BINPUT'
        READ (14,1410) MI,NONE,NEND,DELTA
        IF (M.NE.MI) CALL STOPP (2,2)
        RETURN
      ENDIF
C
C           COMMENT LINES
C           -------------------------------------------------------
C           NLINES = number of comment lines preceding primary
C                    input data
C           Comment lines of primary input file: read and then written
C           as the heading of output files 'ODOC' and 'OMSG'
C           -------------------------------------------------------
      READ  (11,1110) NLINES
      DO 110 L = 1,NLINES
        READ  (11,1120) (COMMEN(I),I=1,15)
        WRITE (98,1120) (COMMEN(I),I=1,15)
        WRITE (99,1120) (COMMEN(I),I=1,15)
  110 CONTINUE
C
C           OPTIONS
C           -------------------------------------------------------
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            ISYST=1: International System of Units (SI) is used
C                 =2: US Customary System of Units (USCS) is used
C            IBOT=1: "Type 1" bottom data (width-slope)
C                =2: "Type 2" bottom data (coordinates)
C            INONCT=0: No correction term in computing ETAR
C                  =1: Correction term for ETAR recommended for
C                      beaches
C            IENERG=0: Energy quantities NOT computed
C                  =1: Energy quantities computed
C            IWAVE=1: Incident waves at seaward boundary computed
C                 =2: Incident waves at seaward boundary given as input
C                 =3: Total waves at seaward boundary given as input
C            If IWAVE>1 --> Must specify MSTAT and FINP2
C              . FINP2 = name of input data file containing time series
C                of normalized free surface elevation at seaw. boundary
C              . Statistical calculations are performed excluding the
C                first MSTAT computation units
C              . A "computation unit" is one wave period long except
C                for the last unit, the length of which may be less
C                than one wave period
C              . The "wave period" is the reference period used for the
C                normalization of the governing equations
C            "A" = Spatial variations of hydrodynamic quantities at
C                  specified moments
C            "B" = Time series of volume flux and water depth at
C                  specified nodes
C            "C" = Time series of either of the following:
C                . For ISTAB=1: stability number SNR at specified nodes
C                . For ISTAB=2: displacement of armor units from
C                               specified initial nodal locations
C            "B" and "C" are stored at rate NRATE data points per wave
C                  period
C            ISAVA,ISAVB,ISAVC are identifiers associated with saving
C                  "A","B","C", respectively (1=save; 0=no)
C            MSAVA1, MSAVA2, AND NTIMES:
C            If ISAVA=1, "A" is saved NTIMES (>1) times at equal
C                          intervals per wave period from computation
C                          unit MSAVA1 to MSAVA2, inclusive
C            If ISAVB=1 --> Must specify NNODB
C            If ISAVC=1 --> Must specify NNODC
C            NNODB,NNODC = number of nodes for which "B","C" is to be
C                          saved
C            IREQ=0: No special storing
C                =1: Special storing requested
C            Special storing = storing spatial variations of requested
C                quantities at normalized time TREQ(i)
C                with i=1,2,...,NREQ
```

```
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          Notes: . DUDT can be requested only if ISTAB>0
C                 . SNR  can be requested only if ISTAB=1
C          -----------------------------------------------------------
      READ  (11,1130) IJOB,ISTAB
      READ  (11,1130) ISYST
      READ  (11,1130) IBOT
      READ  (11,1130) INONCT
      READ  (11,1130) IENERG
      READ  (11,1140) IWAVE,MSTAT,FINP2
      READ  (11,1150) ISAVA,MSAVA1,MSAVA2,NTIMES
      READ  (11,1150) ISAVB,NNODB
      READ  (11,1150) ISAVC,NNODC
      READ  (11,1160) IREQ,IELEV,IV,IDUDT,ISNR,NREQ
      IF (IREQ.EQ.1) THEN
        READ (11,1180) (TREQ(I),I=1,NREQ)
        CALL CKVAL (1,NREQ,N5)
      ENDIF
      IF (ISAVB.EQ.1) CALL CKVAL (2,NNODB,N5)
      IF (ISAVC.EQ.1) CALL CKVAL (3,NNODC,N5)
C
C          COMPUTATION PARAMETERS
C          -----------------------------------------------------------
C          INITS  = . number of spatial nodes along the bottom
C                     below SWL (IJOB<3)
C                   . number of nodes between seaward and
C                     landward boundaries, inclusive (IJOB=3)
C          MULTIF = multiplication factor associated with NONEM
C                   . NONEM = minimum allowable value of NONE
C                   . NONE  = number of time steps in one wave period
C                   . See Subr. 10 CPARAM for more explanation
C          DELTA as input is an initial value for normalized water
C                depth defining computational waterline
C          TMAX  = normalized duration of computation
C          NRATE = rate (data points per per wave period) of storing
C                  output time series
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)
C          X1,X2 = damping coefficients
C          MWAVE = number of computation units
C          MSTAT: Statistical calculations are performed excluding
C                 the first MSTAT computation units
C          MSTAB: Computation of armor stability or movement is per-
```

```
C                 formed excluding the first MSTAB computation units
C           ----------------------------------------------------------
      READ  (11,1110) INITS
      READ  (11,1110) MULTIF
      READ  (11,1170) DELTA
      READ  (11,1170) TMAX
      READ  (11,1110) NRATE
      READ  (11,1110) NDELR
      CALL CKVAL (5,INITS,N1-1)
      IF (IJOB.LT.3) THEN
        CALL CKVAL (4,NDELR,N3)
      ELSE
        NDELR = 0
      ENDIF
      DO 130 L = 1,NDELR
        READ (11,1180) DELRP(L)
  130 CONTINUE
      READ (11,1180) X1,X2
      ITMAX = INT(TMAX)
      RES   = DMOD(TMAX,1.D+00)
      IF (DABS(RES).LT.1.D-08) THEN
        MWAVE = ITMAX
      ELSE
        MWAVE = ITMAX+1
      ENDIF
      IF (IWAVE.EQ.1) THEN
        MSTAT = MWAVE-1
      ENDIF
      IF (ISTAB.EQ.1) THEN
        MSTAB = MSTAT
      ELSEIF (ISTAB.EQ.2) THEN
        MSTAB = 0
      ELSE
        MSTAB = MWAVE+1
        ISAVC = 0
      ENDIF
      IF (ISAVA.EQ.0) MSAVA1=MWAVE+1
C
C           CHECK CASE-SIGNATURES
C           to make sure that RBREAK2 reads the correct file 'BINPUT'
C           ----------------------------------------------------------
C           File 'BINPUT' was created by BEFORR2. It contains, among
C           other things, four pieces of information which are case-
C           specific, and referred to as "case-signatures".
C           On the left are signatures read from file 'BINPUT', on the
C           right their RBREAK2 counterparts:
C           . MWAV - MWAVE
C           . FID2 - FINP2
C           . NDAT - NDATA
C           . NONM - NONEM
C           Left must match with right. Check is conducted in:
C           . this subroutine (the first two)
```

```
C           . Subr. 03 INPUT2 (the third)
C           . Subr. 10 CPARAM (the last)
C           ----------------------------------------------------------
      READ  (14,1110) NLINES
      DO 135 L = 1,NLINES
        READ (14,1120) (COMMEN(I),I=1,15)
  135 CONTINUE
      READ  (14,1110) MWAV
      IF (MWAV.NE.MWAVE) CALL STOPP (3,3)
      IF (IWAVE.GT.1) THEN
        READ (14,5000) FID2
        IF (FID2.NE.FINP2) CALL STOPP (4,4)
        READ (14,1110) NDAT
      ENDIF
.     READ  (14,1110) NONM
C
C           CHECK PARAMETER N2
C           only for regular wave computations
C           ----------------------------------------------------------
C           Read NONEMA from file 'BINPUT'
C           NONEMA = largest NONE to be encountered in computation
C                    as determined by BEFORR2
C           NONE   = number of time steps in one wave period
C           PARAMETER N2 provides dimension for variable ETA, which
C           represents wave profile. For regular waves, wave profile
C           is of size (NONE+1). Thus, N2 must be at least (NONEMA+1)
C           ----------------------------------------------------------
      IF (IWAVE.EQ.1) THEN
        READ (14,1110) NONEMA
        IF (NONEMA.GT.(N2-1)) THEN
          WRITE  (*,9910) NONEMA,N2
          WRITE (99,9910) NONEMA,N2
          STOP
        ENDIF
      ENDIF
C
C           CONSTANTS
C           ----------------------------------------------------------
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)
C           ----------------------------------------------------------
      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           WAVE PROPERTIES AND FRICTION FACTOR
C           ----------------------------------------------------------
```

```
C           HREFP = physical wave height at "reference" location
C                   . in meters if ISYST=1 (SI)
C                   . in feet   if ISYST=2 (USCS)
C           TP    = physical reference wave period, in seconds
C           HREFP and TP are used to normalize the governing
C                   equations
C           KSREF = shoaling coefficient at "reference" location
C           KSSEA = shoaling coefficient at seaward boundary
C           SIGMA = ratio between horizontal and vertical length scales
C           ----------------------------------------------------------
      READ  (11,1180) HREFP,TP
      READ  (11,1180) KSREF,KSSEA
      SIGMA = TP*DSQRT(GRAV/HREFP)
C
C           STRUCTURE GEOMETRY
C           ----------------------------------------------------------
C           The structure geometry is divided into segments of
C           DIFFERENT INCLINATION OR ROUGHNESS
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             FWP(I)    = BOTTOM FRICTION FACTOR
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)
C           NOTE : FOR IBOT=2, SPECIFY FWP(NBSEG+1)=0.0, WHICH IS
C                  NOT USED FOR THE FOLLOWING COMPUTATION
C           ----------------------------------------------------------
      READ  (11,1180) DSEAP
      READ  (11,1180) TSLOPS
      READ  (11,1110) NBSEG
      CALL CKVAL (6,NBSEG,N4-1)
      IF (IBOT.EQ.1) THEN
        DO 140 K = 1,NBSEG
          READ (11,1180) WBSEG(K),TBSLOP(K),FWP(K)
  140   CONTINUE
      ELSE
        DO 150 K = 1,NBSEG+1
          READ (11,1180) XBSEG(K),ZBSEG(K),FWP(K)
  150   CONTINUE
      ENDIF
C
C           DATA RELATED TO SAVING "B", i.e., time series of volume
C           flux and total water depth at specified nodes
```

```
C          --------------------------------------------------------
C          NNODB   = no. of nodes for which "B" is to be saved
C          NODB(I) = I-th node number for which "B" is to be saved
C          --------------------------------------------------------
      IF (ISAVB.EQ.1) READ (11,1190) (NODB(I),I=1,NNODB)
C
C          DATA RELATED TO ARMOR STABILITY AND MOVEMENT
C          --------------------------------------------------------
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          NNODC   = no. of nodes for which "C" is to be saved
C          NODC(I) = I-th node number for which "C" is to be saved
C          "C" = time series of the following
C               . For ISTAB=1: stability number SNR at specified nodes
C               . For ISTAB=2: displacement of armor units from
C                              specified initial nodal locations
C          --------------------------------------------------------
C     ::   To compute SNR = stability number against rolling/sliding:
      IF (ISTAB.GT.0) THEN
        READ (11,1180) C2,C3,SG
        READ (11,1180) CD,CL,CM
        READ (11,1180) TANPHI
        READ (11,1180) AMAX,AMIN
        IF (ISAVC.EQ.1) READ (11,1190) (NODC(I),I=1,NNODC)
      ENDIF
C     ::   To compute movement of armor units, additional input is
C          required:
      IF (ISTAB.EQ.2) READ (11,1180) DAP
C
C            FORMATS
C          --------------------------------------------------------
 1110 FORMAT (3I8)
 1120 FORMAT (15A5)
 1130 FORMAT (2I1)
 1140 FORMAT (I1,I8,2X,A10)
 1150 FORMAT (I1,3I8)
 1160 FORMAT (5I1,I8)
 1170 FORMAT (F15.6)
 1180 FORMAT (3F13.6)
 1190 FORMAT (5I6)
 1410 FORMAT (3I8,D18.9)
```

```
 5000 FORMAT (A10)
 9910 FORMAT (/' From Subr. 02 INPUT1:'/
     +         ' Computation will need NONE as large as',I8,'.'/
     +         ' Dimension N2 must be greater than NONE.'/
     +         ' Currently, N2 =',I8,'.'/
     +         ' Specify larger N2 as required.'/
     +         ' Computation aborted.')
C
      RETURN
      END
C
C -02------------------- END OF SUBROUTINE INPUT1 ------------RBREAK-
C #03####################### SUBROUTINE INPUT2 ###############RBREAK#
C
C     This subroutine reads wave profile at seaward boundary
C
      SUBROUTINE INPUT2 (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /WAVE3/  ETA(N2),ETAIS(N2),ETARS(N2),ETATS(N2)
      COMMON /WAVE4/  ETAMAX,ETAMIN
      COMMON /WAVE6/  NDATA,NDAT
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (3,2,N2,N2R)
        RETURN
      ENDIF
C
C             READ DATA POINTS
C             -------------------------------------------------------
      READ (12,9000) NDATA
      CALL CKVAL (7,NDATA,N2)
      READ (12,8000) (ETA(I),I=1,NDATA)
 8000 FORMAT (5D15.6)
 9000 FORMAT (I8)
C
C             CHECK CASE-SIGNATURE NDAT
C             -------------------------------------------------------
      IF (NDAT.NE.NDATA) CALL STOPP (12,13)
C
C             FIND EXTREME VALUES
C             -------------------------------------------------------
C             ETA = given time series of free surface profile at
C                   seaward boundary
C             ETAMAX,ETAMIN = maximum,minimum of ETA
C             -------------------------------------------------------
      ETAMAX = -1.D+03
      ETAMIN =  1.D+03
      DO 100 I = 1,NDATA
        IF (ETA(I).GT.ETAMAX) ETAMAX=ETA(I)
        IF (ETA(I).LT.ETAMIN) ETAMIN=ETA(I)
```

```
  100 CONTINUE
C
      RETURN
      END
C
C -03------------------- END OF SUBROUTINE INPUT2 ------------RBREAK-
C #04###################### SUBROUTINE BOTTOM ###############RBREAK#
C
C     This subroutine calculates normalized structure geometry and
C     delta x between two adjacent nodes
C
      SUBROUTINE BOTTOM (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      DIMENSION TSLOPE(N1),FWNODE(N1)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /BOT1/   DSEAP,DLANDP,FWP(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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 /ARMOR1/ C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /ARMOR2/ SG1,CTAN(N1)
      COMMON /SAVBC/  NNODB,NNODC,NODB(N5),NODC(N5)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (4,1,N1,N1R)
        CALL CKFPAR (4,4,N4,N4R)
        CALL CKFPAR (4,5,N5,N5R)
        RETURN
      ENDIF
C
C           COMPLETE SEGMENT DATA NOT SPECIFIED AS INPUT
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           The structure geometry is divided into segments of
```

```
C           DIFFERENT INCLINATION OR ROUGHNESS
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           . FWP(I)    = BOTTOM FRICTION FACTOR
C           . XBSEG(i)  = physical horizontal distance from seaward
C                         boundary to the segment's seaward-end
C           . ZBSEG(i)  = physical water depth below SWL
C                         (+ below SWL) at the segment's seaward-end
C           BSWL,DSEAP,WBSEG,XBSEG,ZBSEG are
C           . in meters if ISYST=1 (SI)
C           . in feet   if ISYST=2 (USCS)
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           INITS = . number of spatial nodes along the bottom below
C                     SWL (IJOB<3)
C                   . number of nodes between seaward and landward
C                     boundaries, inclusive (IJOB=3)
C           -------------------------------------------------------
      IF (IJOB.LT.3) THEN
        K = 0
  910   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 910
        BSWL = XBSEG(K+1) + ZBSEG(K+1)/TBSLOP(K)
        X    = BSWL/DBLE(INITS)
      ELSE
        BSWL = XBSEG(NBSEG+1)
        X    = BSWL/DBLE(INITS-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 (physical,
C                       later normalized under the same name)
C           TSLOPE(j) = tangent of local slope at node j
C           FWNODE(J) = BOTTOM FRICTION FACTOR AT NODE J
C           -------------------------------------------------------
      IF (IJOB.LT.3) THEN
        DUM = XBSEG(NBSEG+1)/X
        JE  = INT(DUM)+1
      ELSE
        JE = INITS
      ENDIF
      IF (JE.GT.N1) THEN
        WRITE  (*,9910) JE,N1
        WRITE (99,9910) JE,N1
        STOP
      ELSE
        JE1 = JE-1
      ENDIF
C     ::  Some checks associated with storing time series
      IF (ISAVB.EQ.1) THEN
        DO 140 I = 1,NNODB
          CALL CKVAL (8,NODB(I),JE)
 140    CONTINUE
      ENDIF
      IF (ISAVC.EQ.1) THEN
        DO 150 I = 1,NNODC
          CALL CKVAL (9,NODC(I),JE)
 150    CONTINUE
      ENDIF
 9910 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 160 J = 1,JE
        DIST = DIST + X
        IF (DIST.GT.XCUM.AND.K.LT.NBSEG) THEN
 920      CONTINUE
            K    = K+1
            XCUM = XBSEG(K+1)
          IF (DIST.GT.XCUM.AND.K.LT.NBSEG) GOTO 920
        ENDIF
        U2INIT(J) = ZBSEG(K) - (DIST-XBSEG(K))*TBSLOP(K)
```

```
        TSLOPE(J) = TBSLOP(K)
        FWNODE(J)=FWP(K)
  160 CONTINUE
C
C           NORMALIZATION
C           -----------------------------------------------------------
C           WTOT = normalized width of computation domain
C           At node j:
C           . U2INIT(j) = normalized water depth below SWL
C                         (+ below SWL)
C           . THETA(j)  = normalized tangent of local slope
C           . FW(J)     = NORMALIZED BOTTOM FRICTION FACTOR
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 170 J = 1,JE
        U2INIT(J) = U2INIT(J)/HREFP
        THETA(J)  = TSLOPE(J)*SIGMA
        DIST      = DIST + X
        XB(J)     = DIST
        ZB(J)     = -U2INIT(J)
        FW(J)     = .5D+00*SIGMA*FWNODE(J)
  170 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 180 J = 1,JE
          ANGLE     = DATAN(TSLOPE(J))
          CSLOPE    = DCOS(ANGLE)
          SSLOPE(J) = DSIN(ANGLE)
          CTAN(J)   = CSLOPE*TANPHI
  180   CONTINUE
      ENDIF
C
      RETURN
      END
C
C -04------------------ END OF SUBROUTINE BOTTOM ------------RBREAK-
C #05###################### SUBROUTINE WPARAM ##############RBREAK#
C
C     This subroutine calculates wave parameters
C
```

```
      SUBROUTINE WPARAM (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV
      COMMON /ID/      IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                 ISAVA,ISAVB,ISAVC
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /BOT1/   DSEAP,DLANDP,FWP(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3),
     +                RMEAN(N3)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (5,1,N1,N1R)
        CALL CKFPAR (5,3,N3,N3R)
        RETURN
      ENDIF
C
C           PARAMETERS RELATED TO WAVE AND SLOPE CHARACTERISTICS
C           ------------------------------------------------------------
C           KSI    = surf similarity parameter
C           WLOP   = physical deep-water wavelength
C           DSEAP  = phys. water depth below SWL at seaward boundary
C           DLANDP = phys. water depth below SWL at landward boundary
C                    (only for IJOB=3)
C           DELRP  = water depths associated with visual or measured
C                    waterline
C           WLO,DSEA,DLAND,DELTAR ARE THE NORMALIZED COUNTERPARTS
C           OF WLOP,DSEAP,DLANDP,DELRP, RESPECTIVELY
C           ------------------------------------------------------------
      WLOP   = GRAV*(TP*TP)/(2.D+00*PI)
      WLO    = WLOP/DSEAP
      KS     = KSSEA/KSREF
      KSI    = SIGMA*TSLOPS/DSQRT(2.D+00*PI)
      DSEA   = DSEAP/HREFP
      DSEAKS = DSEA/KS
      DSEA2  = DSQRT(DSEA)
      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(INITS)
        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
C               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
      RETURN
      END
C
C -05----------------- END OF SUBROUTINE WPARAM ------------RBREAK-
C #06##################### SUBROUTINE REGWAV ###############RBREAK#
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 REGWAV (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DOUBLE PRECISION KCNO,MCNO,KC2,KC,KS,KSREF,KSSEA,KSI
      DIMENSION ETAU(N2)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CONSTA/ PI,GRAV
      COMMON /CPAR2/  MULTIF,NONEM,NONM,NONE,NEND,NRATE,NHOP
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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/  KCNO,ECNO,MCNO,KC2
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      IF (MODE.EQ.0) THEN
```

```
        CALL CKFPAR (6,1,N1,N1R)
        CALL CKFPAR (6,2,N2,N2R)
        RETURN
      ENDIF
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           NO = approx. 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.D+00) THEN
C           ::     Stokes II Wave Profile
        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
        NO   = 1
        DO 110 N = 1,NHALF1
          TIMEN = DBLE(N-1)*T
          ETAU(N) = .5D+00*DCOS(TWOPI*TIMEN)+AMP2*DCOS(FOURPI*TIMEN)
          ETAU(N) = KS*ETAU(N)
          IF (N.GT.1) THEN
            IF(ETAU(N).LE.0.D+00.AND.ETAU(N-1).GT.0.D+00)NO=N
          ENDIF
  110   CONTINUE
        ETAMIN = ETAU(NHALF1)
        ETAMAX = ETAU(1)
      ELSE
C           ::     Cnoidal Wave Profile
C                  FINDM is to find the parameter M of the Jacobian
C                  elliptic func. See Func. 08 CEL and Subr. 09 SNCNDN
        CALL FINDM (MCNO)
        KC2    = 1.D+00-MCNO
        KC     = DSQRT(KC2)
        KCNO   = CEL(KC,1.D+00,1.D+00,1.D+00)
        ECNO   = CEL(KC,1.D+00,1.D+00,KC2)
        UR     = 16.D+00*MCNO*KCNO*KCNO/3.D+00
        WL     = DSQRT(UR*DSEAKS)
        ETAMIN = (1.D+00-ECNO/KCNO)/MCNO - 1.D+00
```

```
        ETAMIN = KS*ETAMIN
        ETAMAX = ETAMIN + KS
        NO     = 1
        DO 120 N = 1,NHALF1
          TIMEN = DBLE(N-1)*T
          TETA = 2.D+00*KCNO*TIMEN
          CALL SNCNDN (TETA,KC2,SNU,CNU,DNU)
          ETAU(N) = ETAMIN + KS*CNU*CNU
          IF (N.GT.1) THEN
            IF (ETAU(N).LE.0.D+00.AND.ETAU(N-1).GT.0.D+00)NO=N
          ENDIF
  120   CONTINUE
        ETAU(NHALF1) = ETAMIN
      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 so that elevation=0 at time=0 and
C           decreases initially with time
C           ----------------------------------------------------------
C           ETAU = unadjusted surface elevation
C           ETA  = adjusted surface elevation
C           ----------------------------------------------------------
      NMARK = NONE-NO+2
      DO 140 N = 1,NONE1
        IF (N.LE.NMARK) THEN
          ETA(N) = ETAU(N+NO-1)
        ELSE
          ETA(N) = ETAU(N-NMARK+1)
        ENDIF
  140 CONTINUE
C
      RETURN
      END
C
C -06------------------ END OF SUBROUTINE REGWAV -----------RBREAK-
C #07###################### SUBROUTINE FINDM ################RBREAK#
C
C     This subroutine computes the parameter m (MLIL<m<MBIG) of the
C     Jacobian elliptic functions (m=MCNO in this program)
C
      SUBROUTINE FINDM (MCNO)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(N1=800,N2=24000,N3=3,N4=40,N5=40)
      DOUBLE PRECISION KCNO,MCNO,KC2,KC,MSAV,MLIL,MBIG
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
```

```
      COMMON /CONSTA/ PI,GRAV
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      CALL CKFPAR(7,1,N1,N1R)
      SMALL = 1.D-07
      MLIL  = .8D+00
      INDI  = 0
      I     = 0
      SIGDT = DSQRT(2.D+00*PI*WL0)
      MBIG  = 1.00D+00 - 1.00D-15
      MCNO  =  .95D+00
  900 CONTINUE
        I    = I+1
        MSAV = MCNO
        KC2  = 1.D+00-MCNO
        KC   = DSQRT(KC2)
        KCNO = CEL(KC,1.D+00,1.D+00,1.D+00)
        ECNO = CEL(KC,1.D+00,1.D+00,KC2)
        UR   = 16.D+00*MCNO*KCNO*KCNO/3.D+00
        WL   = DSQRT(UR*DSEAKS)
        FCNO = 1.D+00 + (-MCNO+2.D+00-3.D+00*ECNO/KCNO)/(MCNO*DSEAKS)
        FCNO = SIGDT*DSQRT(FCNO)/WL - 1.D+00
        IF (FCNO.LT.0.D+00) THEN
          MBIG = MCNO
        ELSEIF (FCNO.GT.0.D+00) THEN
          MLIL = MCNO
        ELSE
          RETURN
        ENDIF
        MCNO = (MLIL+MBIG)/2.D+00
        DIF  = DABS(MSAV-MCNO)
        IF (DIF.LT.SMALL) RETURN
        IF (INDI.EQ.0) THEN
          IF (I.EQ.50) THEN
            SMALL = 1.D-13
            INDI  = 1
          ELSE
            IF (MCNO.GT..9999D+00) THEN
              SMALL = 1.D-13
              INDI  = 1
            ENDIF
          ENDIF
        ENDIF
      IF (I.LT.100) GOTO 900
      WRITE  (*,9910)
      WRITE (99,9910)
 9910 FORMAT (/' From Subr. 07 FINDM:'/
     +          ' Criterion for parameter m=MCNO not satisfied')
C
      RETURN
      END
```

```
C
C -07------------------- END OF SUBROUTINE FINDM ------------RBREAK-
C #08################# DOUBLE PRECISION FUNCTION CEL #########RBREAK#
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 (99,*) '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 -08------------ END OF DOUBLE PRECISION FUNCTION CEL ------RBREAK-
C #09###################### SUBROUTINE SNCNDN ###############RBREAK#
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
      D   = 0.D+00
      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
          IF (ABS(D).LT.1.D-10) CALL STOPP (9,9)
          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 -09------------------ END OF SUBROUTINE SNCNDN ------------RBREAK-
C #10###################### SUBROUTINE CPARAM ###############RBREAK#
C
C     This subroutine
C     . checks case-signature NONM at the beginning of RBREAK2
C       computation
C     . calculates computation parameters which are dependent
C       on NONE for each computation unit
C
      SUBROUTINE CPARAM (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(N1=800,N2=24000,N3=3,N4=40,N5=40)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR2/  MULTIF,NONEM,NONM,NONE,NEND,NRATE,NHOP
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
C
```

```
C          CHECK THE CASE-SIGNATURE NONM
C          ---------------------------------------------------------
C        . NONEM = minimum allowable value of NONE
C        . NONE  = number of time steps in one wave period
C        . NONE must:
C          + satisfy a numerical stability criterion
C          + be divisible by NRATE for storing purposes
C          + be an even number, for regular waves (IWAVE=1) only
C        . Below, NDUM3 is the smallest value which satisfies the
C          above criteria. NONEM, the minimum allowable value of NONE,
C          is defined as NONEM=MULTIF*NDUM3, where MULTIF is an
C          integer, greater than or equal to unity, specified by user.
C          ---------------------------------------------------------
      IF (MODE.EQ.0) THEN
        CALL CKFPAR(10,1,N1,N1R)
        EPSI  = DMAX1(X1,X2)
        DUM1  = 1.D+00 + EPSI*EPSI/4.D+00
        DUM2  = DSQRT(DUM1) - EPSI/2.D+00
        DUM3  = (1.D+00+DSQRT(DSEA))/(DUM2*X)
        NDUM1 = INT(DUM3) + 1
        NDUM2 = NDUM1/NRATE + 1
        NDUM3 = NDUM2*NRATE
        IF (IWAVE.EQ.1) THEN
          IDUM = MOD(NDUM3,2)
          IF (IDUM.NE.0) NDUM3=2*NDUM3
        ENDIF
C       ::  Minimum allowable value of NONE is NONEM
        NONEM = MULTIF*NDUM3
        IF (NONM.NE.NONEM) CALL STOPP (14,14)
C       :: Initial value NONE=NONEM serves a practical purpose
        NONE = NONEM
      ENDIF
C
C          CALCULATE COMPUTATION PARAMETERS
C          which are dependent on NONE
C          ---------------------------------------------------------
C          T = constant time step within a computation unit
C          X = constant grid spacing between two adjacent nodes
C          NRATE = rate (data points per per wave period) of storing
C                  output time series
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
      NHOP = NONE/NRATE
C
      RETURN
      END
C
```

```
C -10------------------ END OF SUBROUTINE CPARAM ------------RBREAK-
C #11###################### SUBROUTINE NUMSTA ###############RBREAK#
C
C     This subroutine checks if numerical stability criterion is
C     violated (IPROB=3)
C
      SUBROUTINE NUMSTA (MODE,M)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CPAR2/  MULTIF,NONEM,NONM,NONE,NEND,NRATE,NHOP
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (11,1,N1,N1R)
        RETURN
      ENDIF
C           ---------------------------------------------------------
C           T = time step; X = spatial grid size; XT = X/T
C           ---------------------------------------------------------
      ICOUNT = 0
      DO 100 J = 1,S
        IF (DABS(V(J)).GT.XT) ICOUNT=ICOUNT+1
  100 CONTINUE
      IF (ICOUNT.GT.0) THEN
        IPROB=3
        CALL NSI (OMEGA,NONE)
        WRITE  (*,9910) M,S,NONE,OMEGA,TIME,ICOUNT
        WRITE (99,9910) M,S,NONE,OMEGA,TIME,ICOUNT
        CALL STOPP (10,11)
      ENDIF
 9910 FORMAT (/' From Subr. 11 NUMSTA: IPROB=3'/
     +          ' (Stability criterion violated)'/ .
     +          ' Computation unit             M =',I8/
     +          ' Waterline node               S =',I8/
     +          ' Computation parameters:   NONE =',I8/
     +          '                          OMEGA =',F11.2/
     +          ' Time of occurrence        TIME =',F18.9/
     +          ' Number of occurrence    ICOUNT =',I8)
C
      RETURN
      END
C
C -11------------------ END OF SUBROUTINE NUMSTA ------------RBREAK-
C #12###################### SUBROUTINE EXTRAP ###############RBREAK#
C
C     This subroutine estimates U(2,SP1) and V(SP1) with SP1=(S+1)
C     by extrapolation
C
```

```
              t landward node at present time level (N-1)
C      S = mos1owing values at node j are known at time level (N-1)
C      The fol) = volume flux
C      . U(1,j) = total water depth
C      . U(2.i  = depth-averaged velocity
       V(SP1)
C             INE EXTRAP (MODE)
       SUBROUT
C             T DOUBLE PRECISION (A-H,O-Z)
       IMPLICIER (N1=800,N2=24000,N3=3,N4=40,N5=40)
       PARAMET S,SM1,SP1
       INTEGER/DIMENS/ N1R,N2R,N3R,N4R,N5R
       COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
       COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
       COMMON E.EQ.0) THEN
       IF (MODCKFPAR (12,1,N1,N1R)
         CALL N
         RETUR
       ENDIF  -1
       SM1 = S+1
       SP1 = S  = 2.D+00*V(S) - V(SM1)
       U(2,SP1) = 2.D+00*U(2,S) - U(2,SM1)
       U(1,SP1) = U(2,SP1)*V(SP1)
       IF (U(2,SP1).GT.0.D+00) THEN
         C(SP1) = DSQRT(U(2,SP1))
       ELSE
         C(SP1) = 0.D+00
       ENDIF
C
       RETURN
       END
C
C -12------------------ END OF SUBROUTINE EXTRAP  ------------RBREAK-
C #13######################  SUBROUTINE RETAIN  ###############RBREAK#
C
C      This subroutine retains some values at present time level at
C      landward and seaward boundaries
C
       SUBROUTINE RETAIN (MODE)
C
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       INTEGER S,SM1,SP1
       PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
       COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
       COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
       COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
       COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
       IF (MODE.EQ.0) THEN
         CALL CKFPAR (13,1,N1,N1R)
         RETURN
       ENDIF
       VSN    = V(S)
```

```
      USN(1) = U(1,S)
      USN(2) = U(2,S)
      VMN    = V(SM1)
      UMN(1) = U(1,SM1)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      V2N    = V(2)
C
      RETURN
      END
C
C -13------------------ END OF SUBROUTINE RETAIN ------------RBREAK-
C #14####################### SUBROUTINE CRITV ################RBREAK#
C
C     This subroutine calculates critical velocities used in
C     characteristic variables and checks for negative water depth
C     (IPROB=1)
C
      SUBROUTINE CRITV (MODE,M)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CPAR2/  MULTIF,NONEM,NONM,NONE,NEND,NRATE,NHOP
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (14,1,N1,N1R)
        RETURN
      ENDIF
      ICOUNT = 0
      DO 100 J = 1,S
        IF (U(2,J).LT.0.D+00) THEN
          ICOUNT = ICOUNT + 1
        ELSE
          C(J) = DSQRT(U(2,J))
        ENDIF
  100 CONTINUE
      IF (ICOUNT.GT.0) THEN
        IPROB=1
        CALL NSI (OMEGA,NONE)
        WRITE  (*,9910) M,S,NONE,OMEGA,TIME,ICOUNT
        WRITE (99,9910) M,S,NONE,OMEGA,TIME,ICOUNT
        CALL STOPP (10,11)
      ENDIF
 9910 FORMAT (/' From Subr. 14 CRITV: IPROB=1'/
     +         ' (Negative water depth)'/
     +         ' Computation unit              M =',I8/
     +         ' Waterline node                S =',I8/
     +         ' Computation parameters:    NONE =',I8/
     +         '                           OMEGA =',F11.2/
```

```
     +          ' Time of occurrence        TIME =',F18.9/
     +          ' Number of occurrence    ICOUNT =',I8)
C
      RETURN
     +                ISAVA,ISAVB,ISAVC
C
C -14------------------ END OF SUBROUTINE CRITV ------------RBREAK-
C #15###################### SUBROUTINE MARCH ################RBREAK#
C
C     This subroutine marches the computation from present time level
C     (N-1) to next time level N excluding seaward and landward
C     boundaries, which are treated separately
C
      SUBROUTINE MARCH (MODE,M)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
      COMMON /CPAR2/  MULTIF,NONEM,NONM,NONE,NEND,NRATE,NHOP
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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 /RUNP1/  NDELR,S,SM1,SP1,JMAX
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (15,1,N1,N1R)
        RETURN
      ENDIF
C
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 numer. damping term
C          JE  = landward edge node
C          JE1 = JE-1
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. 20 MATAFG computes non-constant elements of Matrices
```

```
C           A and G, and the elements of Matrix F
C         . Subr. 22 MATS computes the first element of Matrix S
C         . Subr. 21 MATGJR, 23 MATD, and 24 MATU compute the elements
C           of Matrices g, D, and U, respectively
        WRITE (99,9910) M,S,NONE,OMEGA,DELTA,U(2,SM1),TIME
C           the results obtained from the other four subroutines
C           -----------------------------------------------------------
      CALL MATAFG (MODE,1,JLAX1)
      CALL MATGJR (MODE,1,JLAX)
      CALL MATS   (MODE,2,JLAX)
      CALL MATD   (MODE,JDAM,JLAX)
      CALL MATU   (MODE,2,JLAX)
C
C           EMERGENCY STOP
C           IPROB=2: water depth at (S-1) <or= DELTA
C           -----------------------------------------------------------
      IF (U(2,SM1).LE.DELTA) THEN
        IPROB=2
        CALL NSI (OMEGA,NONE)
        WRITE  (*,9910) M,S,NONE,OMEGA,DELTA,U(2,SM1),TIME
        CALL STOPP (10,11)
      ENDIF
 9910 FORMAT (/' From Subr. 15 MARCH: IPROB=2'/
     +         ' (Water depth at (S-1) <or= DELTA)'/
     +         ' Computation unit             M =',I8/
     +         ' Waterline node               S =',I8/
     +         ' Computation parameters:   NONE =',I8/
     +         '                          OMEGA =',F11.2/
     +         '                          DELTA =',F18.9/
     +         ' Water depth at (S-1)  U(2,S-1) =',F18.9/
     +         ' Time of occurrence        TIME =',F18.9)
C
C           COMPLETE THE COMPUTATION OF HYDRODYNAMIC QUANTITIES
C           -----------------------------------------------------------
C           Water depth h is taken to be not less than DELTA for
C           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 -15------------------- END OF SUBROUTINE MARCH  ------------RBREAK-
C #16##################### SUBROUTINE LANDBC  ##############RBREAK#
C
C     This subroutine manages the computation for
C     landward boundary conditions
```

```
C
      SUBROUTINE LANDBC (MODE,M,N,ETAT)
C
      IMPLICIT DOUBLE PRECISION (A-H.O-Z)
      COMMON /STT1/   ELSTAT(3,3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1),
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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(N1),
     +                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,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3),
     +                RMEAN(N3)
     +                ELMEAN(3),U1MEAN(N1),EMEAN(N1),VMEAN(N1)
      COMMON /OVER/   OV(4),OVMEAN
      COMMON /ARMOR3/ JSTAB,JSTABM
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (16,1,N1,N1R)
        CALL CKFPAR (16,2,N2,N2R)
        CALL CKFPAR (16,3,N3,N3R)
        RETURN
      ENDIF
C
C           MANAGE LANDWARD B.C.
C           -------------------------------------------------------
C         . Subr. 17 RUNUP computes shoreline movement if
C           computational waterline is on the slope (IJOB<3)
C         . Subr. 18 OVERT computes hydrodynamic quantities at
C           landward edge node if overtopping occurs (IJOB=2)
C         . ALPHA = landward-advancing characteristics
C         . ETAT  = surface elevation associated with transmitted wave
C           at landward boundary (IJOB=3)
C         . DLAND = norm. water depth below SWL at landward boundary
C           (IJOB=3)
C         . S used in Subr. 17 RUNUP is of present time level
C           -------------------------------------------------------
      IF (IJOB.EQ.1) THEN
        CALL RUNUP (MODE)
        IF (S.GT.JE1) THEN
          WRITE  (*,9910) TIME,S,JE
          WRITE (99,9910) TIME,S,JE
          STOP
        ENDIF
```

```
      ELSEIF (IJOB.EQ.2) THEN
        IF (S.LT.JE) THEN
          CALL RUNUP (MODE)
        ELSE
C           CONDITIONS LANDWARD OF NEW WATERLINE NODE S
        ENDIF
      ELSE
        DUM     = TX*(VSN+C(S))*(VSN-VMN+2.D+00*(C(S)-C(SM1)))
        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
 9910 FORMAT (/' From Subr. 16 LANDBC:'/
     +    ' t =',F18.6,'; S =',I8,'; End Node =',I8/
     +    ' Slope is not long enough to accomodate shoreline movement'/
     +    ' Specify longer slope or choose overtopping computation')
C
C           ---------------------------------------------------------
      IF (IJOB.LT.3) THEN
        DO 100 L = S+1,JE
          U(1,L)  = 0.D+00
          U(2,L)  = 0.D+00
          V(L)    = 0.D+00
          ELEV(L) = ZB(L)
  100   CONTINUE
      ENDIF
C
C           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 200 performed
C           If IJOB=3, NDELR=0, DO 200 void
C           ---------------------------------------------------------
C         . MSTAB: Computation of armor stability or movement is
C           performed excluding the first MSTAB computation units
C         . JSTAB = the largest node number based on DELTAR(1) for
C           armor stability or movement.
C           Notes: . If the computed armor stability or movement at
C                    the shoreline is not realistic, it is recommended
C                    to increase the value of DELTAR(1)
C                  . For IJOB=3, JSTAB=JE, set in Subr. 26 INIT
C         . DELTAR = water depth associated with visual or measured
C           waterline
C         . RUNUPS = free surface elevation where the water depth
C           equals DELTAR
C         . NDELR = number of DELTARs
C           ---------------------------------------------------------
      DO 200 L = 1,NDELR
        IF (IJOB.EQ.2.AND.S.EQ.JE.AND.U(2,S).GE.DELTAR(L))THEN
```

```
          IF (L.EQ.1.AND.M.GT.MSTAB) JSTAB=S
          RUNUPS(L) = ZB(S) + U(2,S)
        ELSE
          INDIC = 0
        ENDIF
 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.M.GT.MSTAB) 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
 200 CONTINUE
C
C           STATISTICAL CALCULATIONS
C           Performed excluding the first MSTAT computation units
C           ---------------------------------------------------------
C         . IJOB<3:
C           RSTAT(1,L),RSTAT(2,L),RSTAT(3,L) = mean,maximum,minimum of
C           RUNUPS(L)
C         . IJOB=2:
C           OV(1) = normalized average overtopping rate
C           OV(2) = normalized maximum overtopping rate
C           OV(3) = normalized overtopping duration
C           OV(4) = normalized time when OV(2) occurs
C         . IJOB=3:
C           For regular wave, during the last computation unit ETAT
C           is saved in time-array ETATS (ETAT = surface elevation
C           associated with transmitted wave at landward boundary)
C           ---------------------------------------------------------
      IF (M.GT.MSTAT) 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)+T
            IF (U(1,JE).GT.OV(2)) THEN
              OV(4) = TIME
              OV(2) = U(1,JE)
            ENDIF
          ENDIF
        ELSE
          IF (IWAVE.EQ.1) THEN
```

```
          ETATS(N) = ETAT
        ELSE
          CALL STAT2 (ETAT,ELSTAT(1,3),ELSTAT(2,3),ELSTAT(3,3))
        ENDIF
      DIMENSION USN2(2),US1N1(2)
      ENDIF
C
      RETURN
      END
C
C -16------------------ END OF SUBROUTINE LANDBC ------------RBREAK-
C #17###################### SUBROUTINE RUNUP ################RBREAK#
C
C     This subroutine computes waterline movement for IJOB=1 and if
C     no overtopping occurs for IJOB=2
C
      SUBROUTINE RUNUP (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      INTEGER S,SM1,SP1,SNEW
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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 /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (17,1,N1,N1R)
        RETURN
      ENDIF
C
C             ADJUST VALUES AT S IF U(2,S)>U(2,S-1)
C             ---------------------------------------------------
      IF (U(2,S).GE.U(2,SM1)) THEN
        V(S)    = 2.D+00*V(SM1) - V(S-2)
        U(2,S)  = 2.D+00*U(2,SM1) - U(2,S-2)
        IF (ABS(V(S)).GT.ABS(V(SM1))) V(S)=.9*V(SM1)
        IF (U(2,S).LT.0.D+00) U(2,S)=.5*U(2,SM1)
        IF (U(2,S).GT.U(2,SM1)) U(2,S)=.9*U(2,SM1)
        U(1,S)  = V(S)*U(2,S)
        ELEV(S) = U(2,S) - U2INIT(S)
      ENDIF
C
C             DETERMINE THE NEXT WATERLINE NODE
C             ---------------------------------------------------
      IF (U(2,S).LE.DELTA) THEN
        SNEW = SM1
      ELSE
        V(SP1)    = 2.D+00*V(S) - V(SM1)
```

```
        U(2,SP1) = 2.D+00*U(2,S) - U(2,SM1)
        U(1,SP1) = V(SP1)*U(2,SP1)
        IF (U(2,SP1).LE.DELTA) THEN
          SNEW = S
C                 at time level N, i=1,2
C         ---------------------------------------------------------
C         ::      USN2(i),VSN2 = U(i,S) and V(S), respectively,
C                 at time level (N+1), i=1,2
          CALL MATAFG (MODE,SM1,SP1)
          CALL MATGJR (MODE,SM1,S)
          CALL MATS   (MODE,S,S)
          DUM1 = TX*((F(1,SP1)-F(1,SM1))/2.D+00+X*G1(S))
          DUM2 = TTXX*(GJR(1,S)-GJR(1,SM1))
          DUM3 = TX*(F(2,SP1)-F(2,SM1))
          DUM4 = TTXX*(GJR(2,S)-GJR(2,SM1))
          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,
          VS = V(S)
          IF (DABS(VS).LT.DELTA) VS=DSIGN(DELTA,VS)
          DUM1  = (XT*(VSN2-VSN)+U(2,SP1)-U(2,SM1)+TWOX*THETA(S))/VS
          VS1N1 = V(SM1)-DUM1
          US1N1(1) = U(1,SM1) - 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   = SP1
                U(2,SP1) = US1N1(2)
                U(1,SP1) = US1N1(1)
                V(SP1)   = VS1N1
              ENDIF
            ELSE
              IF (U(2,SP1).LE.U(2,S)) THEN
                SNEW = SP1
              ELSE
                SNEW = S
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
      IF (SNEW.EQ.SP1) ELEV(SP1)=U(2,SP1)-U2INIT(SP1)
      S = SNEW
```

```
C
C     S at next time level has been found
C
      RETURN
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
C
C -17------------------- END OF SUBROUTINE RUNUP ------------RBREAK-
C #18###################### SUBROUTINE OVERT ###############RBREAK#
C
C     This subroutine computes quantities at landward-end node for
C     IJOB=2 if overtopping occurs, that is, S=JE and SM1=(S-1)=JE1
C
      SUBROUTINE OVERT (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /MATRIX/ A1(2,N1),F(2,N1),G1(N1),GJR(2,N1),S1(N1),D(2,N1)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (18,1,N1,N1R)
        RETURN
      ENDIF
      IF (VMN.GT.C(SM1)) THEN
        U(1,S) = USN(1) - TX*(F(1,S)-F(1,SM1)) - 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(SM1)
        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 = SM1
      ELSE
        ELEV(S) = U(2,S) - U2INIT(S)
      ENDIF
C
      RETURN
      END
C
C -18------------------- END OF SUBROUTINE OVERT ------------RBREAK-
C #19###################### SUBROUTINE SEABC ###############RBREAK#
C
C     This subroutine treats seaward boundary conditions at node j=1
```

```
C
      SUBROUTINE SEABC (MODE,M,N,ETAI,ETAR)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /WAVE3/  ETA(N2),ETAIS(N2),ETARS(N2),ETATS(N2)
      COMMON /WAVE6/  NDATA,NDAT
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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 /STT1/   ELSTAT(3,3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1),
     +                ELMEAN(3),U1MEAN(N1),EMEAN(N1),VMEAN(N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (19,1,N1,N1R)
        CALL CKFPAR (19,2,N2,N2R)
        RETURN
      ENDIF
C
C           ESTIMATE ETAR
C           ----------------------------------------------------------
C         . BETA = seaward-advancing characteristics
C         . ETAR = surface elevation associated with reflected wave at
C           seaward boundary
C         . A correction term included in ETAR if INONCT=1 to improve
C           prediction of wave set-down and setup on beach
C           ----------------------------------------------------------
      VC1 = -V1N+2.D+00*C(1)
      VC2 = -V2N+2.D+00*C(2)
      IF (IWAVE.LT.3) THEN
        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)
      ENDIF
C
C           VALUES AT NODE ONE
C           ----------------------------------------------------------
      IF (IWAVE.EQ.1) THEN
C       :: Regular wave.
C          Surface elevation ETA has been computed by Subr. 06 REGWAV.
        ETAI    = ETA(N+1)
        U(2,1)  = DSEA+ETAR+ETAI
        ELEV(1) = ETAI+ETAR
      ELSE
```

```
C        :: Incident or total wave train is specified by user.
C           The specified time series ETA has NDATA points from the
C           normalized time t=0 to t=TMAX.
C           Interpolate the specified time series to obtain ETAI
C           at t=TIME.
          V(1)   = 2.D+00*DSQRT(U(2,1))-BETA
          DJJ  = DBLE(NDATA-1)*TIME/TMAX
          JJ   = INT(DJJ)
          ETA1 = ETA(JJ+1)
          ETA2 = ETA(JJ+2)
          DEL  = DJJ - DBLE(JJ)
          ETAI = ETA1 + DEL*(ETA2-ETA1)
          IF (IWAVE.EQ.2) THEN
            U(2,1)  = DSEA+ETAI+ETAR
            ELEV(1) = ETAI+ETAR
          ELSE
            U(2,1)  = DSEA+ETAI
            ELEV(1) = ETAI
          ENDIF
        ENDIF
        IF (IWAVE.LT.3) THEN
          U(1,1) = U(2,1)*V(1)
        ENDIF
C
        IF (IWAVE.EQ.3) THEN
          C1 = DSQRT(U(2,1))
          DENOM = 1.D+00-TX*(VC2-VC1)
          V(1) = (2.D+00*C1-VC1 -TX*(VC2-VC1)*C1 -T*THETA(1))/DENOM
          U(1,1) = U(2,1)*V(1)
          BETA = 2.D+00*C1-V(1)
          ETAR = BETA*DSEA2/2.D+00 - DSEA
          IF (INONCT.EQ.1) ETAR=ETAR-KS*KS/(16.D+00*DSEA)
        ENDIF
C
C            STATISTICAL CALCULATIONS
C            Performed excluding the first MSTAT computation units
C            ----------------------------------------------------------
C          . ETAI = surface elevation associated with incident wave
C            (IWAVE=1 OR 2) or total wave (IWAVE=3) at seaward boundary
C          . ETAR = surface elevation associated with reflected wave at
C            seaward boundary
C          . For regular wave (IWAVE=1), during the last computation
C            unit: ETAI,ETAR are saved in time-arrays ETAIS,ETARS
C            ----------------------------------------------------------
        IF (M.GT.MSTAT) THEN
          IF (IWAVE.EQ.1) THEN
            ETAIS(N) = ETAI
            ETARS(N) = ETAR
          ELSE
            IF (IWAVE.EQ.3) THEN
              CALL STAT2 (ETAI-ETAR,ELSTAT(1,1),ELSTAT(2,1),ELSTAT(3,1))
            ELSE
```

```
            CALL STAT2 (ETAI,ELSTAT(1,1),ELSTAT(2,1),ELSTAT(3,1))
          ENDIF
          CALL STAT2 (ETAR,ELSTAT(1,2),ELSTAT(2,2),ELSTAT(3,2))
        ENDIF
C
C
      RETURN
      END
C
C -19------------------- END OF SUBROUTINE SEABC ------------RBREAK-
C #20###################### SUBROUTINE MATAFG ###############RBREAK#
C
C     This subroutine computes, for each node,
C     . the elements of the the first row of Matrix A (2x2)
C                                         --> A1(1,j) and A1(2,j)
C     . the elements of Matrix F (2x1)      --> F(1,j) and F(2,j)
C     . the first element of Matrix G (2x1) --> G1(j)
C     j=node number
C
      SUBROUTINE MATAFG (MODE,JBEGIN,JEND)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +.               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 (MODE.EQ.0) THEN
        CALL CKFPAR (20,1,N1,N1R)
        RETURN
      ENDIF
      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(J)*DABS(V(J))*V(J)
  100 CONTINUE
C
      RETURN
      END
C
C -20------------------ END OF SUBROUTINE MATAFG ------------RBREAK-
C #21###################### SUBROUTINE MATGJR ###############RBREAK#
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 (MODE,JBEGIN,JEND)
C
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
C     COMMON /MATRIX/ A1(2,N1),F(2,N1),G1(N1),GJR(2,N1),S1(N1),D(2,N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (21,1,N1,N1R)
        RETURN
      ENDIF
C
      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 -21------------------ END OF SUBROUTINE MATGJR ------------RBREAK-
C #22####################### SUBROUTINE MATS  ################RBREAK#
C
C     This subroutine computes, for each node, the first element of
C     Matrix S (2x1) --> S1(j), j=node number
C
      SUBROUTINE MATS (MODE,JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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 (MODE.EQ.0) THEN
        CALL CKFPAR (22,1,N1,N1R)
        RETURN
      ENDIF
C
      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(J)*DABS(V(J))*V(J)
        DUM5 = 2.D+00*FW(J)*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
        CALL CKFPAR (23.1.N1.N1R)
C #23######################## SUBROUTINE MATD  ##################RBREAK#
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 (MODE,JDAM,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DIMENSION Q(2,2,N1),UU(2,N1)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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 (MODE.EQ.0) THEN
        RETURN
      ENDIF
C
      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
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
C
      RETURN
      END
C
C -23------------------- END OF SUBROUTINE MATD -------------RBREAK-
C #24####################### SUBROUTINE MATU ################RBREAK#
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 (MODE,JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      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 (MODE.EQ.0) THEN
        CALL CKFPAR (24,1,N1,N1R)
        RETURN
      ENDIF
C
      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 -24------------------- END OF SUBROUTINE MATU -------------RBREAK-
C #25####################### SUBROUTINE NSI #################RBREAK#
C
C     This subroutine calculates numerical stability indicator OMEGA
C     CHECKING N1=N1R IS OMITTED IN THIS SUBROUTINE
C
      SUBROUTINE NSI (OMEGA,NONE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(N1=800,N2=24000,N3=3,N4=40,N5=40)
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
```

```
     +                TSLOPS,WTOT
      EPSI  = DMAX1(X1,X2)
      DUM1  = 1.D+00 + EPSI*EPSI/4.D+00
      DUM2  = DSQRT(DUM1) - EPSI/2.D+00
      COMMON /ID/      IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
C
      RETURN
      END
C
C -25-------------------- END OF SUBROUTINE NSI  -------------RBREAK-
C #26####################### SUBROUTINE INIT  ################RBREAK#
C
C     This subroutine assigns initial values
C
      SUBROUTINE INIT (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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 /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3),
     +                RMEAN(N3)
      COMMON /OVER/   OV(4),OVMEAN
      COMMON /STT1/   ELSTAT(3,3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1),
     +                ELMEAN(3),U1MEAN(N1),EMEAN(N1),VMEAN(N1)
      COMMON /ENERG/  ENER(7,N1),ENERB(15),ENTEMP(3,N1)
      COMMON /ARMOR3/ JSTAB,JSTABM
      COMMON /ARMOR4/ JSNSC,JATMIN
      COMMON /ARMOR5/ CSTAB1,CSTAB2,AMAXS,AMINS,E2,E3PRE(N1),SNR(N1),
     +                SNSX(N1),TSNSX(N1),ATMIN(4,N1),SNSC,TSNSC
      COMMON /ARMOR6/ NMOVE,NSTOP,ISTATE(N1),NODFI(N1)
      COMMON /ARMOR7/ CSTAB3,CSTAB4,CM1,DA,SIGDA,WEIG,
     +                VA(N1),XAA(N1),XA(N1),TDIS(N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (26,1,N1,N1R)
        CALL CKFPAR (26,3,N3,N3R)
        RETURN
      ENDIF
C
C           HYDRODYNAMIC VARIABLES
C           At node j:
C           ----------------------------------------------------------
C           U(1,j)  = volume flux
C           U(2,j)  = total water depth
C                     (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
C         . U1STAT(j) = mean volume flux at node j
        IF (J.LE.INITS) 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           STATISTICAL QUANTITIES
C           ---------------------------------------------------------
C         . Subr. 27 INITH is used to specify initial values for
C           statistical quantities
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         . Mean, maximum, and minimum at seaward boundary:
C               ELSTAT(1,1),ELSTAT(2,1),ELSTAT(3,1):
C                                              due to incident wave
C               ELSTAT(1,2),ELSTAT(2,2),ELSTAT(3,2):
C                                              due to reflected wave
C         . Mean, maximum, and minimum at landward boundary:
C               ELSTAT(1,3),ELSTAT(2,3),ELSTAT(3,3):
C                                              due to transmitted wave
C         . The variable on the left is used as temporary storage in
C           the process of calculating the variable on the right
C           (index j indicates node number):
C               U1MEAN(j) <-- U1STAT(j)
C               EMEAN(j)  <-- ESTAT(1,j)
C               VMEAN(j)  <-- VSTAT(1,j)
C               ELMEAN(k) <-- ELSTAT(1,k) with k=1,2,3
C           ---------------------------------------------------------
      CALL INITH (1,U1STAT,1,JE)
      CALL INITH (2,ESTAT, 3,JE)
      CALL INITH (2,VSTAT, 3,JE)
      CALL INITH (2,ELSTAT,3,3)
      CALL INITH (1,U1MEAN,1,JE)
      CALL INITH (1,EMEAN, 1,JE)
      CALL INITH (1,VMEAN, 1,JE)
      CALL INITH (1,ELMEAN,1,3)
C
C           RUNUP
C           ---------------------------------------------------------
C         . S = node number closest to the instantaneous computational
```

```
C            waterline
C          . JMAX = the largest node number reached by computational
C            waterline
C          . RSTAT(1,k),RSTAT(2,k),RSTAT(3,k) = mean, maximum, and
C            OV(1) = normalized average overtopping rate
C          . RMEAN(k) is used as temporary storage in the process of
C            calculating RSTAT(1,k)
C          . k=1,2,...,NDELR
C            ---------------------------------------------------------
      S    = INITS
      JMAX = INITS
      IF (IJOB.LT.3) THEN
        CALL INITH (2,RSTAT,3,NDELR)
        CALL INITH (1,RMEAN,1,NDELR)
      ELSE
        IF (ISTAB.GT.1) JSTAB=JE
      ENDIF
C
C            OVERTOPPING
C            ---------------------------------------------------------
C            OV(2) = normalized maximum overtopping rate
C            OV(3) = normalized overtopping duration
C            OV(4) = normalized time when OV(2) occurs
C            OVMEAN is used as temporary storage in the process of
C                   calculating OV(1)
C            ---------------------------------------------------------
      IF (IJOB.EQ.2) THEN
        CALL INITH (1,OV,1,4)
        OVMEAN = 0.D+00
      ENDIF
C
C            WAVE ENERGY
C            ---------------------------------------------------------
C          . See Subr. 30 ENERGY for description of ENER(i,j) with
C            i=1,2,...,7
C          . ENTEMP(k,j) is used as temporary storage in the process of
C            calculating ENER(k,j) at node j with k=1,2,3
C            ---------------------------------------------------------
      IF (IENERG.EQ.1) THEN
        CALL INITH (1,ENER  ,7,JE)
        CALL INITH (1,ENTEMP,3,JE)
      ENDIF
C
C            ARMOR STABILITY OR MOVEMENT
C            ---------------------------------------------------------
C          . See Subr. 31 STABNO for description of variables
C            associated with ISTAB=1 (computation of armor stability)
C          . See Subr. 32 MOVE for description of variables
C            associated with ISTAB=2 (computation of armor movement)
C            ---------------------------------------------------------
      IF (ISTAB.GT.0) JSTABM=0
```

```
      IF (ISTAB.EQ.1) THEN
        SNSC = 1.D+03
        DO 130 J = 1,JE
          SNSX(J) = 1.D+03
C #27######################  SUBROUTINE INITH  ################RBREAK#
      ELSEIF (ISTAB.EQ.2) THEN
        NMOVE = 0
        NSTOP = 0
        DO 140 J = 1,JE
          ISTATE(J) = 0
          NODFI(J)  = J
          VA(J)     = 0.D+00
          XAA(J)    = 0.D+00
  140   CONTINUE
      ENDIF
C
      RETURN
      END
C
C -26------------------- END OF SUBROUTINE INIT  -------------RBREAK-
C
C     This subroutine facilitates assignment of initial values in
C     Subr. 26 INIT
C
      SUBROUTINE INITH (IMODE,VAL,ND1,ND2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VAL(ND1,ND2)
      IF (IMODE.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 -27------------------- END OF SUBROUTINE INITH  ------------RBREAK-
C #28######################  SUBROUTINE INSTAT  ###############RBREAK#
C
C     This subroutine initializes statistical quantities representing
C     mean values at the beginning of a computation unit, since
C     delta t = (1/NONE) may vary from one unit to another
```

```
C
      SUBROUTINE INSTAT (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        RETURN
      [illegible]
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3),
     +                RMEAN(N3)
      COMMON /OVER/   OV(4),OVMEAN
      COMMON /STT1/   ELSTAT(3,3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1),
     +                ELMEAN(3),U1MEAN(N1),EMEAN(N1),VMEAN(N1)
      COMMON /ENERG/  ENER(7,N1),ENERB(15),ENTEMP(3,N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (28,1,N1,N1R)
        CALL CKFPAR (28,3,N3,N3R)
      ENDIF
C
      DO 110 J = 1,2
        ELSTAT(1,J) = 0.D+00
  110 CONTINUE
      DO 120 J = 1,JE
        U1STAT(J)  = 0.D+00
        ESTAT(1,J) = 0.D+00
        VSTAT(1,J) = 0.D+00
  120 CONTINUE
      DO 130 J = 1,NDELR
        RSTAT(1,J) = 0.D+00
  130 CONTINUE
      IF (IJOB.EQ.2) THEN
        OV(1) = 0.D+00
      ELSEIF (IJOB.EQ.3) THEN
        ELSTAT(1,3) = 0.D+00
      ENDIF
      IF (IENERG.EQ.1) THEN
        DO 150 L = 1,3
          DO 140 J = 1,JE
            ENER(L,J) = 0.D+00
  140     CONTINUE
  150   CONTINUE
      ENDIF
C
      RETURN
      END
C
C -28------------------ END OF SUBROUTINE INSTAT ------------RBREAK-
C #29###################### SUBROUTINE SVSTAT ###############RBREAK#
```

```
C
C     This subroutine saves statistical quantities related to
C     mean values at the end of a computation unit, since
C     delta t = (1/NONE) may vary from one unit to another
      COMMON /OVER/   OV(4),OVMEAN
C
      SUBROUTINE SVSTAT (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  MULTIF,NONEM,NONM,NONE,NEND,NRATE,NHOP
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3),
     +                RMEAN(N3)
      COMMON /STT1/   ELSTAT(3,3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1),
     +                ELMEAN(3),U1MEAN(N1),EMEAN(N1),VMEAN(N1)
      COMMON /ENERG/  ENER(7,N1),ENERB(15),ENTEMP(3,N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (29,1,N1,N1R)
        CALL CKFPAR (29,1,N3,N3R)
        RETURN
      ENDIF
C
      CONFAC = DBLE(NONE)*(TMAX-DBLE(MSTAT))
      DO 110 J = 1,2
        ELMEAN(J) = ELMEAN(J) + ELSTAT(1,J)/CONFAC
  110 CONTINUE
      DO 120 J = 1,JE
        U1MEAN(J) = U1MEAN(J) + U1STAT(J)/CONFAC
        EMEAN(J)  = EMEAN(J)  + ESTAT(1,J)/CONFAC
        VMEAN(J)  = VMEAN(J)  + VSTAT(1,J)/CONFAC
  120 CONTINUE
      DO 130 J = 1,NDELR
        RMEAN(J) = RMEAN(J) + RSTAT(1,J)/CONFAC
  130 CONTINUE
      IF (IJOB.EQ.2) THEN
        OVMEAN = OVMEAN + OV(1)/CONFAC
      ELSEIF (IJOB.EQ.3) THEN
        ELMEAN(3) = ELMEAN(3) + ELSTAT(1,3)/CONFAC
      ENDIF
      IF (IENERG.EQ.1) THEN
        DO 150 L = 1,3
          DO 140 J = 1,JE
            ENTEMP(L,J) = ENTEMP(L,J) + ENER(L,J)/CONFAC
  140     CONTINUE
```

```
  150   CONTINUE
      ENDIF
C
      RETURN
C       ENER(4,j): due to wave breaking, per unit surface area
C
C -29------------------ END OF SUBROUTINE SVSTAT  ------------RBREAK-
C #30####################### SUBROUTINE ENERGY  ###############RBREAK#
C
C     This subroutine computes quantities related to wave energy
C
C     At node j:
C       E(i,j) with i=1,2,3 is instantaneous quantity
C       ENER(i,j) with i=1,2,3,4 is time-averaged quantity
C       E(1,j),ENER(1,j) = norm. energy per unit surface area
C       E(2,j),ENER(2,j) = norm. energy flux per unit width
C       ENER(5,j) saves the value of E(1,j) at t=TSTAT1
C       ENER(6,j) saves the value of E(1,j) at t=TSTAT2
C     Normalized rate of energy dissipation at node j:
C       E(3,j),ENER(3,j): due to bottom friction, per unit bottom area
C     TSTAT1,TSTAT2 = normalized time at the beginning and at the end,
C       respectively, of statistical calculations. Special treatment
C       at t=TSTAT1 and t=TSTAT2 is provided only for random waves
C       (IWAVE>1) and is handled by ICALL=9.
C     At node j:
C       ENER(7,j) = (ENER(6,j)-ENER(5,j))/(TSTAT2-TSTAT1) for random
C       waves in Subr. 39 ENERGC
C
      SUBROUTINE ENERGY (ICALL,M)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DIMENSION E(3,N1)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /STT2/   TSTAT1,TSTAT2
      COMMON /ENERG/  ENER(7,N1),ENERB(15),ENTEMP(3,N1)
      IF (ICALL.EQ.0) THEN
C                ::  Checking mode  ::
        CALL CKFPAR (30,1,N1,N1R)
        RETURN
      ELSEIF (ICALL.EQ.9) THEN
C                ::  Special treatment  ::
        DO 110 J = 1,S
          E(1,J) = (U(1,J)*V(J) + ELEV(J)*ELEV(J))/2.D+00
```

```
          IF (U(2,J).LT.ELEV(J)) THEN
            E(1,J) = E(1,J) - (U(2,J)-ELEV(J))**2/2.D+00
          ENDIF
  110   CONTINUE
        IF (M.EQ.MSTAT) THEN
          TSTAT1 = TIME
          DO 120 J = 1,S
            ENER(5,J) = E(1,J)
  120     CONTINUE
        ELSEIF (M.EQ.MWAVE) THEN
          TSTAT2 = TIME
          IF (MSTAT.EQ.0) TSTAT1 = 0.D+00
          DO 130 J = 1,S
            ENER(6,J) = E(1,J)
  130     CONTINUE
        ENDIF
        RETURN
      ENDIF
C                   ::  Normal mode  ::
      DO 220 J = 1,S
        E(1,J) = (U(1,J)*V(J) + ELEV(J)*ELEV(J))/2.D+00
        IF (U(2,J).LT.ELEV(J)) THEN
          E(1,J) = E(1,J) - (U(2,J)-ELEV(J))**2/2.D+00
        ENDIF
        E(2,J) = U(1,J)*(V(J)*V(J)/2.D+00+ELEV(J))
        E(3,J) = FW(J)*DABS(V(J))*V(J)*V(J)
        DO 210 L = 1,3
          ENER(L,J) = ENER(L,J) + E(L,J)
  210   CONTINUE
  220 CONTINUE
C
      RETURN
      END
C
C -30------------------ END OF SUBROUTINE ENERGY ------------RBREAK-
C #31###################### SUBROUTINE STABNO ###############RBREAK#
C
C     This subroutine computes stability number against
C     rolling/sliding, SNR
C
      SUBROUTINE STABNO (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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,S,SM1,SP1,JMAX
```

```
      COMMON /ARMOR2/ SG1,CTAN(N1)
      COMMON /ARMOR3/ JSTAB,JSTABM
      COMMON /ARMOR4/ JSNSC,JATMIN
      COMMON /ARMOR5/ CSTAB1,CSTAB2,AMAXS,AMINS,E2,E3PRE(N1),SNR(N1),
     +                SNSX(N1),TSNSX(N1),ATMIN(4,N1),SNSC,TSNSC
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (31,1,N1,N1R)
        RETURN
      ENDIF
C
C           FLUID ACCELERATION
C           Computed in Subr. 34 ACCEL
C           ----------------------------------------------------------
      CALL ACCEL (MODE)
C
C           STABILITY NUMBER SNR
C           ----------------------------------------------------------
C         . SNR(j) = stability number against rolling/sliding at
C           node j
C         . SNR is computed for j=1,2,...,JSTAB where JSTAB is defined
C           in Subr. 16 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. 02 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) TIME,J
            WRITE (29,2910) TIME,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. 31 STABNO'/' Armor stability impossible'/
    +         ' Time =',F18.6,'; J =',I8)
C
  110 CONTINUE
C
C           FIND SNSX
C           ---------------------------------------------------------
C           SNSX(j)  = local stability number
C                    = minimum of SNR at node j
C           TSNSX(j) = normalized time when SNSX(j) occurs
C           ---------------------------------------------------------
      DO 120 J = 1,JSTAB
        IF (SNR(J).LT.SNSX(J)) THEN
          SNSX(J)  = SNR(J)
          TSNSX(J) = TIME
        ENDIF
  120 CONTINUE
C
C           CRITICAL STABILITY NUMBER, SNSC
C           ---------------------------------------------------------
C           SNSC  = critical stability number
C                 = minimum of SNSX along the slope
C           TSNSC = normalized time when SNSC occurs
C           JSNSC = node number where SNSC occurs
C           At the time of minimum stability, i.e., t=TSNSC:
C           . ATMIN(i,j) saves spatial variations, i=1,2,3,4
C           . JATMIN = the value of JSTAB
C           ---------------------------------------------------------
      INDI = 0
      DO 130 J = 1,JSTAB
        IF (SNSX(J).LT.SNSC) THEN
          SNSC  = SNSX(J)
          JSNSC = J
          TSNSC = TSNSX(J)
          INDI  = 1
        ENDIF
  130 CONTINUE
      IF (INDI.EQ.1) THEN
        JATMIN = JSTAB
        DO 140 J = 1,JATMIN
          ATMIN(1,J) = ELEV(J)
          ATMIN(2,J) = V(J)
          ATMIN(3,J) = DUDT(J)
          ATMIN(4,J) = SNR(J)
  140   CONTINUE
      ENDIF
C
      RETURN
      END
C
```

```
C -31----------------- END OF SUBROUTINE STABNO ------------RBREAK-
C #32####################### SUBROUTINE MOVE ################RBREAK#
C
C     This subroutine computes movement of armor units
C
      SUBROUTINE MOVE (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /ARMOR1/ C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /ARMOR3/ JSTAB,JSTABM
      COMMON /ARMOR6/ NMOVE,NSTOP,ISTATE(N1),NODFI(N1)
      COMMON /ARMOR7/ CSTAB3,CSTAB4,CM1,DA,SIGDA,WEIG,
     +                VA(N1),XAA(N1),XA(N1),TDIS(N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (32,1,N1,N1R)
        RETURN
      ENDIF
C
C           FLUID ACCELERATION
C           Computed in Subr. 34 ACCEL
C           ----------------------------------------------------------
      CALL ACCEL (MODE)
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           NODFI(j) = node number closest to the armor unit at the
C                      end of each time step
C         . Dynamics:
C           FDES = normalized destabilizing force
C           FR   = normalized resistance force
C           FDES and FR are computed in Subr. 33 FORCES
C           ISTATE(j) indicates the state of armor unit initially
C             located 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           + TDIS(j) = norm. time when it starts moving the first
C             time
C         . It is assumed that once an armor unit is dislodged from a
```

```
C           node, 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 (MODE,V(J),J,FDES,FR)
          IF (DABS(FDES).GT.FR) THEN
            IF (FDES.LT.0.D+00) FR=-FR
            NMOVE      = NMOVE+1
            ISTATE(J) = 1
            TDIS(J)    = TIME
            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 (MODE,VREL,NOD,FDES,FR)
          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) = 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 (MODE,V(NOD),NOD,FDES,FR)
          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)   = 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 -32------------------- END OF SUBROUTINE MOVE  -------------RBREAK-
C #33####################### SUBROUTINE FORCES  ###############RBREAK#
C
C     This subroutine computes destabilizing force FDES and resistance
C     force FR used in Subr. 32 MOVE
C
      SUBROUTINE FORCES (MODE,VELO,NODE,FDES,FR)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      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 /ARMOR1/ C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /ARMOR7/ CSTAB3,CSTAB4,CM1,DA,SIGDA,WEIG,
     +                VA(N1),XAA(N1),XA(N1),TDIS(N1)
      COMMON /ARMOR2/ SG1,CTAN(N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (33,1,N1,N1R)
        RETURN
      ENDIF
C
C     WEIG = normalized submerged weight of armor unit defined in
C            Subr. 40 SPARAM
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. 32 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 -33------------------ END OF SUBROUTINE FORCES ------------RBREAK-
C #34###################### SUBROUTINE ACCEL ################RBREAK#
C
C     This subroutine computes total fluid acceleration using
C     Subr. 41 VECMAT and Subr. 42 DERIV
C
      SUBROUTINE ACCEL (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DIMENSION VDUM1(N1),VDUM2(N1)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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,S,SM1,SP1,JMAX
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (34,1,N1,N1R)
        RETURN
      ENDIF
      CALL VECMAT (VDUM1,U,2,S,2)
      CALL DERIV  (VDUM1,VDUM2,X,S)
      DO 100 J = 1,S
        DUDT(J) = -VDUM2(J)-THETA(J)-FW(J)*V(J)*DABS(V(J))/U(2,J)
  100 CONTINUE
C
      RETURN
      END
C
C -34------------------- END OF SUBROUTINE ACCEL ------------RBREAK-
C #35###################### SUBROUTINE STAT1 ################RBREAK#
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 (IMODE,VAL1,VAL2,ND1,ND2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VAL1(ND1,ND2),VAL2(ND2)
      IF (IMODE.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
      RETURN
      END
C
C -35------------------- END OF SUBROUTINE STAT1  ------------RBREAK-
C #36###################### SUBROUTINE STAT2  ################RBREAK#
C
C     This subroutine computes sum, maximum value, and minimum value
C     of time series VAL
C
      SUBROUTINE STAT2 (VAL,VALAVE,VALMAX,VALMIN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      VALAVE=VALAVE+VAL
      IF (VAL.GT.VALMAX) VALMAX=VAL
      IF (VAL.LT.VALMIN) VALMIN=VAL
      RETURN
      END
C
C -36------------------- END OF SUBROUTINE STAT2  ------------RBREAK-
C #37###################### SUBROUTINE STATC  ################RBREAK#
C
C     This subroutine completes statistical calculations
C
      SUBROUTINE STATC (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR2/  MULTIF,NONEM,NONM,NONE,NEND,NRATE,NHOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /WAVE3/  ETA(N2),ETAIS(N2),ETARS(N2),ETATS(N2)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3),
     +                RMEAN(N3)
      COMMON /OVER/   OV(4),OVMEAN
      COMMON /COEFS/  RCOEF(3),TCOEF(3)
      COMMON /STT1/   ELSTAT(3,3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1),
     +                ELMEAN(3),U1MEAN(N1),EMEAN(N1),VMEAN(N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (37,1,N1,N1R)
```

```
          CALL CKFPAR (37,2,N2,N2R)
          CALL CKFPAR (37,3,N3,N3R)
          RETURN
        ENDIF
C
C             REFLECTION AND TRANSMISSION COEFFICIENTS AND MEAN SURFACE
C             ELEVATIONS AT BOUNDARIES
C             --------------------------------------------------------
C           . RCOEF(i),TCOEF(i) = reflection,transmission coefficient of
C             the i-th kind with i=1,2,3
C           . For regular waves, all the three kinds of coefficients are
C             calculated using Subr. 38 COEF. For random waves, only the
C             first kind is calculated.
C           . Mean surface elevations:
C               ELSTAT(1,1): due to incident wave
C               ELSTAT(1,2): due to reflected wave
C               ELSTAT(1,3): due to transmitted wave
C             --------------------------------------------------------
      IF (IWAVE.EQ.1) THEN
        CALL COEF (1,DUM  ,DUM1,ETAIS,NONE)
        CALL COEF (2,RCOEF,DUM2,ETARS,NONE)
        ELSTAT(1,1) = DUM1
        ELSTAT(1,2) = DUM2
        IF (IJOB.EQ.3) THEN
          CALL COEF (2,TCOEF,DUM3,ETATS,NONE)
          ELSTAT(1,3) = DUM3
        ENDIF
      ELSE
        ELSTAT(1,1) = ELMEAN(1)
        ELSTAT(1,2) = ELMEAN(2)
        RCOEF(1) = (ELSTAT(2,2)-ELSTAT(3,2))/KS
        IF (IJOB.EQ.3) THEN
          ELSTAT(1,3) = ELMEAN(3)
          TCOEF(1) = (ELSTAT(2,3)-ELSTAT(3,3))/KS
        ENDIF
      ENDIF
C
C             WAVE SETUP ON SLOPE COMPUTED FROM WATERLINE MOTION
C             --------------------------------------------------------
      DO 110 J = 1,NDELR
        RSTAT(1,J) = RMEAN(J)
  110 CONTINUE
C
C             MEAN HYDRODYNAMIC QUANTITIES
C             --------------------------------------------------------
C             Mean value at node j:
C             . U1STAT(j):  volume flux
C             . ESTAT(1,j): surface elevation above SWL
C             . VSTAT(1,j): depth-averaged velocity
C             --------------------------------------------------------
      DO 120 J = 1,JMAX
        U1STAT(J)  = U1MEAN(J)
```

```
        ESTAT(1,J) = EMEAN(J)
        VSTAT(1,J) = VMEAN(J)
  120 CONTINUE
C
C           AVERAGE OVERTOPPING RATE
C           ----------------------------------------------------------
      OV(1) = OVMEAN
C
      RETURN
      END
C
C -37------------------- END OF SUBROUTINE STATC ------------RBREAK-
C #38####################### SUBROUTINE COEF ################RBREAK#
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 (IMODE,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 (IMODE.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           ----------------------------------------------------------
C           Monochromatic incident wave profile is assumed to be given
C           by ETAI=(KS/2)COS[2*PI*(t+t0)] since linear wave theory is
C           normally used to estimate reflection and transmission
C           coefficients
C           ----------------------------------------------------------
        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 -38------------------ END OF SUBROUTINE COEF  -------------RBREAK-
C #39###################### SUBROUTINE ENERGC  ###############RBREAK#
C
C     This subroutine completes computation of energy quantities
C
      SUBROUTINE ENERGC (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      DIMENSION VDUM1(N1),VDUM2(N1)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /COEFS/  RCOEF(3),TCOEF(3)
      COMMON /STT2/   TSTAT1,TSTAT2
      COMMON /ENERG/  ENER(7,N1),ENERB(15),ENTEMP(3,N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (39,1,N1,N1R)
        RETURN
      ENDIF
C
C          -------------------------------------------------------
C         . TSTAT1,TSTAT2 = normalized time at the beginning and at
C           the end, respectively, of statistical calculations
C         . Time-averaged quantities at node j:
C             ENER(1,j) = norm. energy per unit surface area
C             ENER(2,j) = norm. energy flux per unit width
C             ENER(3,j) = norm. rate of energy dissipation due to
C                         bottom friction, per unit bottom area
C             ENER(4,j) = norm. rate of energy dissipation due to
C                         wave breaking, per unit surface area
C         . Norm. energy per unit surface area at node j:
C             ENER(5,j): at t=TSTAT1
C             ENER(6,j): at t=TSTAT2
C         . At node j:
C             ENER(7,j) = change of norm. energy per unit surface area
C                         from TSTAT1 to TSTAT2
C          -------------------------------------------------------
      DO 120 L = 1,3
        DO 110 J = 1,JMAX
```

```
          ENER(L,J) = ENTEMP(L,J)
  110   CONTINUE
  120 CONTINUE
C           :: Use Subr. 41 VECMAT and Subr. 42 DERIV
      CALL VECMAT (VDUM1,ENER,7,JMAX,2)
      CALL DERIV  (VDUM1,VDUM2,X,JMAX)
      DO 140 J = 1,JMAX
        ENER(4,J) = -VDUM2(J)-ENER(3,J)
  140 CONTINUE
      IF (IWAVE.GT.1) THEN
        TSPAN = TSTAT2-TSTAT1
        DO 150 J = 1,JMAX
          ENER(7,J) = (ENER(6,J)-ENER(5,J))/TSPAN
          ENER(4,J) = ENER(4,J)-ENER(7,J)
  150   CONTINUE
      ENDIF
C
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
C           domain:
C             ENERB(3): due to bottom friction
C             ENERB(4): due to wave breaking
C         . ENERB(5) = integral of ENER(7,j) in the computation domain
C           -------------------------------------------------------------
      ENERB(1) = ENER(2,1)
      IF (IJOB.EQ.1) THEN
        ENERB(2) = 0.D+00
      ELSE
        ENERB(2) = ENER(2,JMAX)
      ENDIF
      DO 170 I = 3,5
        IF (I.LT.5) THEN
          IE=I
        ELSE
          IE=7
        ENDIF
        ENERB(I) = (ENER(IE,1)+ENER(IE,JMAX))/2.D+00
        DO 160 J = 2,JMAX-1
          ENERB(I) = ENERB(I) + ENER(IE,J)
  160   CONTINUE
        ENERB(I) = ENERB(I)*X
  170 CONTINUE
C
      ENERB(6) = ENERB(1) - ENERB(2)
      ENERB(7) = ENERB(3) + ENERB(4) + ENERB(5)
      ENERB(8) = ENERB(7) - ENERB(6)
      ENERB(9) = 100.D+00*ENERB(8)/ENERB(6)
C
C           -------------------------------------------------------------
```

```
C          Approximate energy flux based on regular linear long wave:
C          ENERB(10): due to incident wave at seaward boundary
C          ENERB(11): due to reflected wave at seaward boundary
C          ENERB(12): due to transmitted wave at landward boundary
C          -------------------------------------------------------
      IF (IWAVE.EQ.1) THEN
        ENERB(10) = KS*KS*DSEA2/8.D+00
        ENERB(11) = DSEA2*(KS*RCOEF(3))**2/8.D+00
        ENERB(13) = ENERB(10)-ENERB(11)
        ENERB(14) = 100.D+00*(ENERB(13)-ENERB(1))/ENERB(1)
        IF (IJOB.EQ.3) THEN
          ENERB(12) = DLAND2*(KS*TCOEF(3))**2/8.D+00
          ENERB(15) = 100.D+00*(ENERB(12)-ENERB(2))/ENERB(2)
        ENDIF
      ENDIF
C
      RETURN
      END
C
C -39------------------ END OF SUBROUTINE ENERGC ------------RBREAK-
C #40####################### SUBROUTINE SPARAM ###############RBREAK#
C
C     This subroutine calculates parameters needed for computation
C     of armor stability or movement
C
      SUBROUTINE SPARAM (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /ARMOR1/ C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /ARMOR2/ SG1,CTAN(N1)
      COMMON /ARMOR3/ JSTAB,JSTABM
      COMMON /ARMOR5/ CSTAB1,CSTAB2,AMAXS,AMINS,E2,E3PRE(N1),SNR(N1),
     +                SNSX(N1),TSNSX(N1),ATMIN(4,N1),SNSC,TSNSC
      COMMON /ARMOR7/ CSTAB3,CSTAB4,CM1,DA,SIGDA,WEIG,
     +                VA(N1),XAA(N1),XA(N1),TDIS(N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (40,1,N1,N1R)
        RETURN
      ENDIF
C
C             -------------------------------------------------------
C           . SG = specific gravity
C           . Coefficients:
C             C2: area, C3: volume, CD: drag, CL: lift, CM: inertia
```

```
C           . TANPHI = armor friction factor
C           . AMAX,AMIN = upper,lower bounds of fluid acceleration,
C             normalized by gravitational acceleration
C           . E3PRE is prepared for computing E3 in Subr. 31 STABNO
C           . DAP,DA = physical,normalized armor diameters
C           . WEIG = normalized submerged weight of armor unit
C           -----------------------------------------------------
      SG1 = SG-1.D+00
      IF (IJOB.EQ.3) JSTAB=JE
      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 100 J = 1,JE
          E3PRE(J) = CSTAB1*CTAN(J)
  100   CONTINUE
      ELSE
        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 -40------------------ END OF SUBROUTINE SPARAM ------------RBREAK-
C #41##################### SUBROUTINE VECMAT ################RBREAK#
C
C     This subroutine creates vector VAL1 from row NROW of matrix VAL2
C
      SUBROUTINE VECMAT (VAL1,VAL2,ND1,ND2,NROW)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VAL1(ND2),VAL2(ND1,ND2)
      DO 100 J = 1,ND2
        VAL1(J) = VAL2(NROW,J)
  100 CONTINUE
      RETURN
      END
C
C -41------------------ END OF SUBROUTINE VECMAT ------------RBREAK-
C #42##################### SUBROUTINE DERIV ################RBREAK#
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
      RETURN
      END
C
C -42------------------- END OF SUBROUTINE DERIV ------------RBREAK-
C #43####################### SUBROUTINE DOC1  ################RBREAK#
C
C     This subroutine documents essential information before
C     time-marching computation
C
      SUBROUTINE DOC1 (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DOUBLE PRECISION KCNO,MCNO,KC2,KS,KSREF,KSSEA,KSI
      CHARACTER*7 UL
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  MULTIF,NONEM,NONM,NONE,NEND,NRATE,NHOP
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /WAVE4/  ETAMAX,ETAMIN
      COMMON /WAVE5/  KCNO,ECNO,MCNO,KC2
      COMMON /WAVE6/  NDATA,NDAT
      COMMON /BOT1/   DSEAP,DLANDP,FWP(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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 /ARMOR1/ C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (43,1,N1,N1R)
        CALL CKFPAR (43,4,N4,N4R)
        RETURN
      ENDIF
C
C           SYSTEM OF UNITS
C           ------------------------------------------------------------
```

```
      IF (ISYST.EQ.1) THEN
        UL = ' meters'
      ELSE
        UL = ' feet   '
      ENDIF
C
C           WAVE CONDITION
C           ---------------------------------------------------------
      WRITE (98,9811)
      IF (IWAVE.EQ.1) THEN
        IF (URPRE.LT.26.D+00) THEN
          WRITE (98,9812)
        ELSE
          WRITE (98,9813) KC2,ECNO,KCNO
        ENDIF
      ELSEIF (IWAVE.EQ.2) THEN
        WRITE (98,9814) NDATA
      ELSE
        WRITE (98,9815) NDATA
      ENDIF
      WRITE (98,9816) ETAMAX,ETAMIN
      WRITE (98,9817) TP,HREFP,UL,DSEAP,UL,KSREF,KSSEA,KS
      WRITE (98,9818) DSEA,WL,SIGMA,UR,KSI
      IF (IJOB.EQ.3)  WRITE (98,9819) DLANDP,UL
 9811 FORMAT (/'WAVE CONDITION')
 9812 FORMAT (/'Stokes II Incident Wave at Seaward Boundary')
 9813 FORMAT (/'Cnoidal Incident Wave at Seaward Boundary'/
     +          '1-m = ',D20.9/
     +          'E   = ',D20.9/
     +          'K   = ',D20.9)
 9814 FORMAT (/'Incident Wave at Seaward Boundary Given as Input'/
     +          'Number of Data Points     NDATA =',I8)
 9815 FORMAT (/'Total Wave at Seaward Boundary Given as Input'/
     +          'Number of Data Points     NDATA =',I8)
 9816 FORMAT ('Norm. Maximum Surface Elev.     = ',F14.6/
     +         'Norm. Minimum Surface Elev.     = ',F14.6/)
 9817 FORMAT ('Reference Wave Period           = ',F14.6,' sec.'/
     +          'Reference Wave Height           = ',F14.6,A7/
     +          'Depth at Seaward Boundary       = ',F14.6,A7/
     +          'Shoal. Coef. at Reference  Ks1 = ',F11.3/
     +          '              at Seaw. Bdr. Ks2 = ',F11.3/
     +          '              Ks = Ks2/Ks1      = ',F11.3)
 9818 FORMAT ('Norm. Depth at Seaw. Boundary  = ',F11.3/
     +         'Normalized Wave Length          = ',F11.3/
     +         '"Sigma"                         = ',F11.3/
     +         'Ursell Number                   = ',F11.3/
     +         'Surf Similarity Parameter       = ',F11.3)
 9819 FORMAT ('Depth at Landward Boundary      = ',F14.6,A7)
C
C           STRUCTURE PROPERTIES
C           ---------------------------------------------------------
      WRITE (98,9821) WTOT,NBSEG
```

```
      IF (IBOT.EQ.1) THEN
        WRITE (98,9822) UL
        WRITE (98,9824) (K,WBSEG(K),TBSLOP(K),FWP(K),K=1,NBSEG)
      ELSE
        WRITE (98,9823) UL,UL
        WRITE (98,9824) (K,XBSEG(K),ZBSEG(K),FWP(K),K=1,NBSEG+1)
      ENDIF
      WRITE (98,9825)
 9821 FORMAT (/'SLOPE PROPERTIES'//
     +        'Norm. Horiz. Length of'/
     +        '       Computation Domain    = ',F15.6/
     +        'Number of Segments           = ',I8)
 9822 FORMAT (32(1H-)/' SEGMENT     WBSEG(I)   TBSLOP(I)   FWP(I)'/
     +        '       I      ',A7/32(1H-))
 9823 FORMAT (32(1H-)/' SEGMENT     XBSEG(I)     ZBSEG(I)   FWP(I)'/
     +        '       I      ',A7,'      ',A7/32(1H-))
 9824 FORMAT (I8,3F12.6)
 9825 FORMAT (32(1H-))
C
C          PARAMETERS FOR ARMOR STABILITY AND MOVEMENT
C          ------------------------------------------------------------
      IF (ISTAB.GT.0) WRITE (98,9831) TANPHI,SG,C2,C3,CD,CL,CM
      IF (ISTAB.EQ.1) WRITE (98,9832) AMAX,AMIN
      IF (ISTAB.EQ.2) WRITE (98,9833) DAP,UL
 9831 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)
 9832 FORMAT ('Norm. Upper and Lower Bounds of du/dt'/
     +        '                      AMAX  = ',F9.3/
     +        '                      AMIN  = ',F9.3)
 9833 FORMAT ('Armor Diameter              = ',F12.6,A7)
C
C          COMPUTATION PARAMETERS
C          ------------------------------------------------------------
      WRITE (98,9841) JE
      IF (IJOB.LT.3)  WRITE (98,9842) INITS
      WRITE (98,9843) X,X1,X2,TMAX,NONEM,NRATE,MSTAT
      IF (ISTAB.GT.0) WRITE (98,9844) MSTAB
      IF (ISAVA.EQ.1) WRITE (98,9845) NTIMES,MSAVA1,MSAVA2
 9841 FORMAT (/'COMPUTATION PARAMETERS'//
     +  'Total Number of Spatial Nodes       JE = ',I8)
 9842 FORMAT ('Number of Nodes Along Bottom Below SWL'/
     +  '                                 INITS = ',I8)
 9843 FORMAT (
     +  'Normalized Delta x           = ',D19.6/
     +  'Damping Coefficients    x1 = ',F15.3/
     +  '                        x2 = ',F15.3/
```

```
     + 'Computation Duration  TMAX = ',F18.6,' Wave Periods'/
     + 'Minimum Allowable NONE          NONEM = ',I8/
     + 'Time Series Stored at Rate NRATE'/
     + '      per Wave Period with      NRATE = ',I8/
     + 'Statistical Calculations Are Performed Excluding'/
     + '      the First MSTAT Computation Units'/
     + '      with                      MSTAT = ',I8)
 9844 FORMAT (
     + 'Computation of Armor Stability or Movement is Performed'/
     + '      Excluding the First MSTAB Computation Units'/
     + '      with                      MSTAB = ',I8)
 9845 FORMAT (
     + 'Spatial Variations are Stored NTIMES at Equal'/
     + '      Intervals per Wave Period from'/
     + '      Computation Unit MSAVA1 to MSAVA2,'/
     + '      Inclusive, with          NTIMES = ',I8/
     + '                               MSAVA1 = ',I8/
     + '                               MSAVA2 = ',I8)
C
C          NORMALIZED STRUCTURE GEOMETRY
C          -----------------------------------------------------------
C          File 52 = 'OSPACE'
C          (XB(j),ZB(j)) = normalized coordinates of the structure
C                          at node j
C          ZB negative below SWL
C          -----------------------------------------------------------
      WRITE (52,9000) JE
      WRITE (52,8001) (XB(J),ZB(J),J=1,JE)
 9000 FORMAT (I8)
 8000 FORMAT (5D15.6)
 8001 FORMAT (5E15.6)
C
      WRITE (*,*) 'Time-marching computation begins'
C
      RETURN
      END
C
C -43------------------- END OF SUBROUTINE DOC1  -------------RBREAK-
C #44######################### SUBROUTINE DOC2  ################RBREAK#
C
C     This subroutine stores computed results at designated time
C     during time-marching computation
C
      SUBROUTINE DOC2 (ICALL,ETAI,ETAR,ETAT)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      DIMENSION VONE(N5),VTWO(N5)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/       IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                  ISAVA,ISAVB,ISAVC
```

```
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /REQ1/   IREQ,IELEV,IV,IDUDT,ISNR,NREQ
      COMMON /REQ2/   TREQ(N5)
      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,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3),
     +                RMEAN(N3)
      COMMON /ARMOR5/ CSTAB1,CSTAB2,AMAXS,AMINS,E2,E3PRE(N1),SNR(N1),
     +                SNSX(N1),TSNSX(N1),ATMIN(4,N1),SNSC,TSNSC
      COMMON /ARMOR6/ NMOVE,NSTOP,ISTATE(N1),NODFI(N1)
      COMMON /ARMOR7/ CSTAB3,CSTAB4,CM1,DA,SIGDA,WEIG,
     +                VA(N1),XAA(N1),XA(N1),TDIS(N1)
      COMMON /SAVBC/  NNODB,NNODC,NODB(N5),NODC(N5)
C
      IF (ICALL.EQ.0) THEN
C
C           CHECKING PARAMETERs
C           ------------------------------------------------------------
        CALL CKFPAR (44,1,N1,N1R)
        CALL CKFPAR (44,2,N2,N2R)
        CALL CKFPAR (44,3,N3,N3R)
        CALL CKFPAR (44,5,N5,N5R)
C
      ELSEIF (ICALL.EQ.1) THEN
C
C           STORING "A"
C           "A" = spatial variations of hydrodynamic quantities
C           ------------------------------------------------------------
C           File 52 = 'OSPACE'
C           S = waterline node (IJOB<3) or landward-end node (IJOB=3)
C           TIME = current normalized time
C           At node j:
C              ELEV(j) = surface elevation above SWL
C              V(j)    = depth-averaged velocity
C           ------------------------------------------------------------
        WRITE (52,9000) S
        WRITE (52,8001) (ELEV(J),V(J),J=1,S),TIME
C
      ELSEIF (ICALL.EQ.2) THEN
C
C           STORING "B"
C           "B" = time series of volume flux and total water depth at
C                 specified nodes
C           ------------------------------------------------------------
C           File 71 ('OSAV1Bxx') stores volume flux
C           File 72 ('OSAV2Bxx') stores total water depth
C           where xx=01 contains the first 100 waves
C                 xx=02 contains the second 100 waves
C                 and so on
C           ------------------------------------------------------------
```

```
      IF (ISAVB.EQ.1) THEN
        DO 110 I = 1,NNODB
          J = NODB(I)
          VONE(I) = U(1,J)
          VTWO(I) = U(2,J)
 110    CONTINUE
        WRITE (71,8001) (VONE(I),I=1,NNODB),TIME
        WRITE (72,8001) (VTWO(I),I=1,NNODB)
      ENDIF
C
C         STORING "C"
C         "C" = time series of X, where X is
C             . for ISTAB=1: stability number SNR at specified nodes
C             . for ISTAB=2: displacement of armor units from
C                            specified initial nodal locations
C         ---------------------------------------------------------
C         File 73 = 'OSAVECxx'
C         where xx=01 contains the first 100 waves
C               xx=02 contains the second 100 waves
C         ---------------------------------------------------------
      IF (ISAVC.EQ.1) THEN
        IF (ISTAB.EQ.1) THEN
          DO 120 I = 1,NNODC
            J = NODC(I)
            VONE(I) = SNR(J)
 120      CONTINUE
        ELSE
          DO 130 I = 1,NNODC
            J = NODC(I)
            IF (ISTATE(J).EQ.0) THEN
              VONE(I) = 0.D+00
            ELSE
              VONE(I) = XA(J)
            ENDIF
 130      CONTINUE
        ENDIF
        WRITE (73,8001) (VONE(I),I=1,NNODC),TIME
      ENDIF
C
C         STORING VALUES AT LANDWARD-END NODE
C         ---------------------------------------------------------
C         File 62 = 'ORUNUP'
C         File 63 = 'OOVER'
C         File 64 = 'OTRANS'
C         JE = landward-end node
C         S  = waterline node
C         RUNUPS = free surface elevation where the water depth
C                  equals DELTAR
C         DELTAR = water depth associated with visual or measured
C                  waterline
C         NDELR  = number of DELTARs
C         ETAT   = surface elevation associated with transmitted
```

```
C                    wave at landward boundary
C              At node j:
C                 U(1,j) = volume flux
C                 U(2,j) = total water depth
C              -----------------------------------------------------------
          IF (IJOB.LT.3) THEN
            DBLES=DBLE(S)
            WRITE (62,8001) DBLES,(RUNUPS(L),L=1,NDELR),TIME
            IF (IJOB.EQ.2) WRITE (63,8001) U(1,JE),U(2,JE)
          ELSE
            WRITE (64,8001) U(1,JE),U(2,JE),ETAT,TIME
          ENDIF
C
C              STORING VALUES AT SEAWARD BOUNDARY
C              -----------------------------------------------------------
C              File 61 = 'OSEAWAV'
C              TIME = current normalized time
C              Surface elevations at seaward boundary:
C              . ETAI --> due to incident wave
C              . ETAR --> due to reflected wave
C              U(1,1) = volume flux at node 1
C              -----------------------------------------------------------
          IF (IWAVE.EQ.3) ETAI=ETAI-ETAR
          WRITE (61,8001) TIME,ETAI,ETAR,U(1,1)
C
        ELSE
C
C              SPECIAL STORING IF ICALL=3
C              -----------------------------------------------------------
C              File 55 = 'OREQ'
C              S = waterline node (IJOB<3) or landward-end node (IJOB=3)
C              TIME = current normalized time
C              At node j for specified time:
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 (55,9000) S
          WRITE (55,8001) TIME
          IF (IELEV.EQ.1) WRITE (55,8001) (ELEV(J),J=1,S)
          IF (IV.EQ.1)    WRITE (55,8001) (V(J),J=1,S)
          IF (IDUDT.EQ.1) WRITE (55,8001) (DUDT(J),J=1,S)
          IF (ISNR.EQ.1)  WRITE (55,8001) (SNR(J),J=1,S)
C
        ENDIF
 9000 FORMAT (2I8)
 8000 FORMAT (5D15.6)
 8001 FORMAT (5E15.6)
C
        RETURN
        END
```

```
C
C -44------------------ END OF SUBROUTINE DOC2  -------------RBREAK-
C #45####################### SUBROUTINE DOC3  ################RBREAK#
C
C     This subroutine documents essential results after time-marching
C     computation
C
      SUBROUTINE DOC3 (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40)
      CHARACTER*7 UL
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /COEFS/  RCOEF(3),TCOEF(3)
      COMMON /STT1/   ELSTAT(3,3),U1STAT(N1),ESTAT(3,N1),VSTAT(3,N1),
     +                ELMEAN(3),U1MEAN(N1),EMEAN(N1),VMEAN(N1)
      COMMON /STT2/   TSTAT1,TSTAT2
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3),RUNUPS(N3),RSTAT(3,N3),
     +                RMEAN(N3)
      COMMON /OVER/   OV(4),OVMEAN
      COMMON /ENERG/  ENER(7,N1),ENERB(15),ENTEMP(3,N1)
      COMMON /ARMOR3/ JSTAB,JSTABM
      COMMON /ARMOR4/ JSNSC,JATMIN
      COMMON /ARMOR5/ CSTAB1,CSTAB2,AMAXS,AMINS,E2,E3PRE(N1),SNR(N1),
     +                SNSX(N1),TSNSX(N1),ATMIN(4,N1),SNSC,TSNSC
      COMMON /ARMOR6/ NMOVE,NSTOP,ISTATE(N1),NODFI(N1)
      COMMON /ARMOR7/ CSTAB3,CSTAB4,CM1,DA,SIGDA,WEIG,
     +                VA(N1),XAA(N1),XA(N1),TDIS(N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (45,1,N1,N1R)
        CALL CKFPAR (45,3,N3,N3R)
        RETURN
      ENDIF
C
C           SYSTEM OF UNITS
C           -------------------------------------------------------
      IF (ISYST.EQ.1) THEN
        UL = '   [mm]'
      ELSE
        UL = ' [inch]'
      ENDIF
C
C           REFLECTION COEFFICIENTS
C           -------------------------------------------------------
```

```
      IF (IWAVE.EQ.1) THEN
        WRITE (98,9811) (RCOEF(I),I=1,3)
      ELSE
        WRITE (98,9812) RCOEF(1)
      ENDIF
 9811 FORMAT (/'REFLECTION COEFFICIENTS'//
     +         'r1 = ',F9.3/
     +         'r2 = ',F9.3/
     +         'r3 = ',F9.3)
 9812 FORMAT (/'REFLECTION COEFFICIENT'//
     +         'r1 = ',F9.3)
C
C          RUNUP, RUNDOWN, SETUP
C          -------------------------------------------------------
      IF (IJOB.LT.3) THEN
        WRITE (98,9821) JMAX
        WRITE (98,9822) UL
        DO 110 L = 1,NDELR
          WRITE (98,9823) L,DELRP(L),RSTAT(2,L),RSTAT(3,L),RSTAT(1,L)
  110   CONTINUE
        WRITE (98,9824)
      ENDIF
 9821 FORMAT (/'RUNUP, RUNDOWN, SETUP'//
     +'Largest Node Number Reached by Computational Waterline'/
     +'                                          JMAX = ',I8)
 9822 FORMAT (56(1H-)/
     +'       I  DELTAR(I)   RUNUP(I)  RUNDOWN(I)   SETUP(I)'/
     +'           ',A7,'        R          Rd          Zr'/56(1H-))
 9823 FORMAT (I8,1X,F9.3,3(3X,F9.3))
 9824 FORMAT (56(1H-))
C
C          OVERTOPPING
C          -------------------------------------------------------
      TSPAN = TMAX-DBLE(MSTAT)
      IF (IJOB.EQ.2) WRITE (98,9831)
     +             OV(1),U1STAT(1),OV(2),OV(4),OV(3),TSPAN,TMAX
 9831 FORMAT (/'OVERTOPPING'//
     +'Norm. Average Overtopping Rate           = ',D15.6/
     +'Norm. Average Flow at Seaward Boundary = ',D15.6/
     +'Norm. Maximum Overtopping Rate           = ',D15.6/
     +'      Occuring at Normalized Time      t = ',F18.6/
     +'Norm. Overtopping Duration               = ',F18.6/
     +'Notes: Norm. duration of overtopping computation =',F18.6/
     +'       Norm. duration of RBREAK2 computation     =',F18.6)
C
C          WAVE SET-DOWN OR SETUP
C          -------------------------------------------------------
      IF (IJOB.EQ.3) THEN
        DELMWL = ELSTAT(1,3) - ELSTAT(1,2)
        WRITE (98,9841) (ELSTAT(1,I),I=1,3),DELMWL
      ELSE
        WRITE (98,9842) (ELSTAT(1,I),I=1,2)
```

```
      ENDIF
 9841 FORMAT (/'WAVE SET-DOWN OR SETUP'//
     +         'Average value of ETAI = ',F12.6/
     +         '                 ETAR = ',F12.6/
     +         '                 ETAT = ',F12.6/
     +         'MWL Difference        = ',F12.6)
 9842 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))
        IF (IWAVE.EQ.1) THEN
          WRITE (98,9851) (TCOEF(I),I=1,3)
        ELSE
          WRITE (98,9852) TCOEF(1)
        ENDIF
        WRITE (98,9853) U1STAT(1),U1STAT(JE),QAVER
      ENDIF
 9851 FORMAT (/'TRANSMISSION'//
     +        'Transmission Coefficient T1 = ',F9.3/
     +        '                         T2 = ',F9.3/
     +        '                         T3 = ',F9.3)
 9852 FORMAT (/'TRANSMISSION'//
     +        'Transmission Coefficient T1 = ',F9.3)
 9853 FORMAT ('Norm. Average Flow at Seaward Boundary  = ',F18.9/
     +        'Norm. Average Flow at Landward Boundary = ',F18.9/
     +        'Average of the Above Two                = ',F18.9)
C
C          STATISTICS OF HYDRODYNAMIC QUANTITIES
C          ---------------------------------------------------------
C          File 51 = '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 (51,9000) JMAX
      WRITE (51,8001) (U1STAT(J),J=1,JMAX)
      DO 120 I = 1,3
        WRITE (51,8001) (ESTAT(I,J),J=1,JMAX)
        WRITE (51,8001) (VSTAT(I,J),J=1,JMAX)
  120 CONTINUE
C
C          ENERGY QUANTITIES
```

```
C          -------------------------------------------------------
C          File 53 = '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 (98,9861)
        WRITE (98,9862) (ENERB(I),I=1,9)
        IF (IWAVE.EQ.1) THEN
          WRITE (98,9863) (ENERB(I),I=10,11)
          IF (IJOB.EQ.3)  WRITE (98,9864) ENERB(12)
          WRITE (98,9865) (ENERB(I),I=13,14)
          IF (IJOB.EQ.3)  WRITE (98,9866) ENERB(15)
        ENDIF
        WRITE (53,9000) JMAX
        DO 130 I = 1,7
          WRITE (53,8001) (ENER(I,J),J=1,JMAX)
 130    CONTINUE
      ENDIF
 9861 FORMAT (/'TIME-AVERAGED ENERGY BALANCE'//
     +        'Normalized Energy Flux:')
 9862 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/
     +        'Additional Term        E =',D14.6/
     +        'Calculation 1:'/
     +        '                  F = A-B =',D14.6/
     +        '                G = C+D+E =',D14.6/
     +        ' Must H=0, but H = G-F =',D14.6/
     +        ' % error  100G/E =',F14.2)
 9863 FORMAT ('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)
 9864 FORMAT ('. Transmitted Wave at Landw. Bndry.   R =',D14.6)
 9865 FORMAT ('Calculation 2:'/
     +        'Net Energy Flux at Seaw. Bndry. S = P-Q =',D14.6/
     +        '% Error at Seaward Boundary  100(S-A)/A =',F14.2)
 9866 FORMAT ('% Error at Landward Boundary 100(R-B)/B =',F14.2)
C
C          ARMOR STABILITY OR MOVEMENT
C          -------------------------------------------------------
C          File 54 = 'OSTAB'
C          (XB(j),ZB(j)) = normalized coordinates of the structure at
C                          node j
```

```
C           ISTAB=1:
C             SNSX(j) = local stability number
C                     = minimum of SNR at a node j
C             SNR(j)  = stability number against rolling/sliding at
C                       node j
C           ISTAB=2:
C             ISTATE(j) indicates the state of armor unit initially
C               located at node j: 0=stationary, 1=moving, 2=stopped
C             For moving/stopped armor unit number j:
C               NODFI(j) = node closest to its final location
C               TDIS(j)  = time when it started moving
C               XA(j)    = displacement from its initial location,
C                          normalized by DAP
C           -------------------------------------------------------
        IF (ISTAB.EQ.1) THEN
          WRITE (98,9871) JSTABM
          WRITE (98,9872) SNSC,JSNSC,TSNSC
          WRITE (54,9000) JSTABM
          DO 140 J = 1,JSTABM
            WRITE (54,8001) XB(J),ZB(J),SNSX(J),TSNSX(J)
  140     CONTINUE
          WRITE (54,9000) JATMIN
          WRITE (54,8001) TSNSC
          DO 150 I = 1,4
            WRITE (54,8001) (ATMIN(I,J),J=1,JATMIN)
  150     CONTINUE
        ELSEIF (ISTAB.EQ.2) THEN
          WRITE (98,9873) JSTABM,NMOVE,NSTOP
          WRITE (54,9000) NMOVE
          WRITE (54,9000) NSTOP
          WRITE (54,9000) JSTABM
          DO 160 J = 1,JSTABM
            IF (ISTATE(J).GE.1) THEN
              WRITE (54,9000) J,NODFI(J),ISTATE(J)
              WRITE (54,8001) XB(J),ZB(J),XA(J),TDIS(J)
            ENDIF
  160     CONTINUE
        ENDIF
 9871 FORMAT (/'STABILITY NUMBER'//
     +         'Largest Node Number for Which Armor Stability'/
     +         '  Computation Performed  JSTABM = ',I11)
 9872 FORMAT ( 'Critical Stability Number   Nsc = ',F11.3/
     +         'At Node Number                  = ',I11/
     +         'At Time                         = ',F18.6)
 9873 FORMAT (/'ARMOR UNITS MOVEMENT'//
     +         'Largest Node Number for Which Armor Movement'/
     +         '  Computation Performed  JSTABM = ',I8/
     +         'Number of Units Moved     NMOVE = ',I8/
     +         'Number of Units Stopped   NSTOP = ',I8)
C
 9000 FORMAT (8I8)
 8000 FORMAT (5D15.6)
```

```
 8001 FORMAT (5E15.6)
C
      RETURN
      END
C
C -45------------------- END OF SUBROUTINE DOC3  -------------RBREAK-
C #46#######################  SUBROUTINE CKFPAR  ###############RBREAK#
C
C     This subroutine checks if FORTRAN PARAMETER NCHEK=N1,N2,N3,N4,N5
C     specified in given Subroutine (ICALL) matches its counterpart
C     NREF=N1R,N2R,N3R,N4R,N5R from the main program
C
      SUBROUTINE CKFPAR (ICALL,NW,NCHEK,NREF)
C
      CHARACTER*2 WHICH(5)
      CHARACTER*6 SUBR(50)
      DATA WHICH /'N1','N2','N3','N4','N5'/
      DATA SUBR  /'OPENIO','INPUT1','INPUT2','BOTTOM','WPARAM',
     2            'REGWAV','FINDM ','CEL   ','SNCNDN','CPARAM',
     3            'NUMSTA','EXTRAP','RETAIN','CRITV ','MARCH ',
     4            'LANDBC','RUNUP ','OVERT ','SEABC ','MATAFG',
     5            'MATGJR','MATS  ','MATD  ','MATU  ','NSI   ',
     6            'INIT  ','INITH ','INSTAT','SVSTAT','ENERGY',
     7            'STABNO','MOVE  ','FORCES','ACCEL ','STAT1 ',
     8            'STAT2 ','STATC ','COEF  ','ENERGC','SPARAM',
     9            'VECMAT','DERIV ','DOC1  ','DOC2  ','DOC3  ',
     +            'CKFPAR','CKVAL ','CKSUBR','STOPP ','OPENF '/
      IF (NCHEK.NE.NREF) THEN
        WRITE  (*,9910)
     +         WHICH(NW),NCHEK,ICALL,SUBR(ICALL),WHICH(NW),NREF
        WRITE (99,9910)
     +         WHICH(NW),NCHEK,ICALL,SUBR(ICALL),WHICH(NW),NREF
        STOP
      ENDIF
 9910 FORMAT (/
     +' PARAMETER Error: ',A2,' =',I8,' in Subroutine',I3,' ',A6/
     +' Correct Value:   ',A2,' =',I8)
C
      RETURN
      END
C
C -46------------------- END OF SUBROUTINE CKFPAR  ------------RBREAK-
C #47#######################  SUBROUTINE CKVAL  ################RBREAK#
C
C     This subroutine checks user-specified values
C
      SUBROUTINE CKVAL (IC,ITEM,IUP)
C
      CHARACTER*2 WHICH(7)
      CHARACTER*6 WHAT(9)
      DATA WHICH /'N5','N5','N5','N3','N1','N4','N2'/
      DATA WHAT  /'NREQ  ','NNODB ','NNODC ','NDELR ','INITS ',
```

```
     +            'NBSEG ','NDATA ','NODB  ','NODC  '/
      IF (IC.LE.7) THEN
        IF (ITEM.LT.1.OR.ITEM.GT.IUP) THEN
          WRITE  (*,9910) WHAT(IC),ITEM,WHAT(IC),IUP,WHICH(IC)
          WRITE (99,9910) WHAT(IC),ITEM,WHAT(IC),IUP,WHICH(IC)
          STOP
        ENDIF
      ELSE
        IF (ITEM.LT.1.OR.ITEM.GT.IUP) THEN
          WRITE  (*,9920) WHAT(IC),ITEM,WHAT(IC),IUP
          WRITE (99,9920) WHAT(IC),ITEM,WHAT(IC),IUP
          STOP
        ENDIF
      ENDIF
 9910 FORMAT (/' Input Error: ',A6,'=',I8/
     +        ' Specify ',A6,' in the range of [1,',I8,']'/
     +        ' Change PARAMETER ',A2,' if necessary')
 9920 FORMAT (/' Input Error: ',A6,'=',I8/
     +        ' Specify ',A6,' in the range of [1,',I8,']')
C
      RETURN
      END
C
C -47------------------ END OF SUBROUTINE CKVAL  ------------RBREAK-
C #48####################### SUBROUTINE CKSUBR  ###############RBREAK#
C
C     This subroutine checks the values of the FORTRAN PARAMETERs
C     specified in subroutines after reading input data file
C
      SUBROUTINE CKSUBR (N1,N2,N3,N4,N5)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*10 CDUM
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R
      COMMON /ID/      IJOB,ISTAB,ISYST,IBOT,INONCT,IENERG,IWAVE,
     +                 ISAVA,ISAVB,ISAVC
      DATA IDUM /0/
      DATA DDUM /0.D+00/
      DATA CDUM /'Tcharacter'/
C
C           DUMMIES FOR FORTRAN PARAMETERS USED IN THIS PROGRAM
C           ----------------------------------------------------------
C           Variables specified in PARAMETER statement cannot be
C           passed through COMMON statement. The following dummy
C           integers are used in COMMON /DIMENS/.
C           ----------------------------------------------------------
      N1R = N1
      N2R = N2
      N3R = N3
      N4R = N4
      N5R = N5
C
```

```
C           CHECK ALL SUBROUTINES POSSESSING FORTRAN "PARAMETER"
C           STATEMENT
C           Checking is indicated by MODE=0
C           ---------------------------------------------------------
      MODE=0
      CALL INPUT1 (MODE,IDUM,CDUM)
      CALL BOTTOM (MODE)
      CALL WPARAM (MODE)
      CALL NUMSTA (MODE,IDUM)
      CALL RETAIN (MODE)
      CALL CRITV  (MODE,IDUM)
      CALL MARCH  (MODE,IDUM)
      CALL LANDBC (MODE,IDUM,IDUM,DDUM)
      CALL SEABC  (MODE,IDUM,IDUM,DDUM,DDUM)
      CALL MATAFG (MODE,IDUM,IDUM)
      CALL MATGJR (MODE,IDUM,IDUM)
      CALL MATS   (MODE,IDUM,IDUM)
      CALL MATD   (MODE,IDUM,IDUM)
      CALL MATU   (MODE,IDUM,IDUM)
      CALL INIT   (MODE)
      CALL INSTAT (MODE)
      CALL SVSTAT (MODE)
      CALL STATC  (MODE)
      CALL DOC1   (MODE)
      CALL DOC2   (MODE,DDUM,DDUM,DDUM)
      CALL DOC3   (MODE)
      IF (IJOB.LT.3) THEN
        CALL EXTRAP (MODE)
        CALL RUNUP  (MODE)
        IF (IJOB.EQ.2) CALL OVERT (MODE)
      ENDIF
      IF (IWAVE.EQ.1) THEN
        CALL REGWAV (MODE)
      ELSE
        CALL INPUT2 (MODE)
      ENDIF
      IF (IENERG.EQ.1) THEN
        CALL ENERGY (MODE,IDUM)
        CALL ENERGC (MODE)
      ENDIF
      IF (ISTAB.GT.0) THEN
        CALL SPARAM (MODE)
        CALL ACCEL  (MODE)
        IF (ISTAB.EQ.1) THEN
          CALL STABNO (MODE)
        ELSE
          CALL MOVE   (MODE)
          CALL FORCES (MODE,DDUM,IDUM,DDUM,DDUM)
        ENDIF
      ENDIF
C
      RETURN
```

```
      END
C
C -48------------------ END OF SUBROUTINE CKSUBR ------------RBREAK-
C #49###################### SUBROUTINE STOPP ################RBREAK#
C
C     This subroutine executes a programmed stop
C
      SUBROUTINE STOPP (IBEGIN,IEND)
C
      CHARACTER*55 MSG(14)
      DATA MSG /
     1 ' MREP must be positive or zero.',
     2 ' Computation unit not matched. Check BINPUT.',
     3 ' Computation duration does not match. Check BINPUT.',
     4 ' Second input file name does not match. Check BINPUT.',
     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.',
     9 ' Failure in Subr. 09 SNCNDN.',
     + ' IPROB nonzero.',
     1 ' Was corresponding BEFORR computation okay?',
     2 ' Number of data points for input time series',
     3 ' does not match. Check BINPUT.',
     4 ' Minimum allowable NONE does not match. Check BINPUT.'/
      DO 100 I = IBEGIN,IEND
        WRITE  (*,9910) MSG(I)
        WRITE (99,9910) MSG(I)
  100 CONTINUE
      WRITE  (*,9920)
      WRITE (99,9920)
 9910 FORMAT (/A55)
 9920 FORMAT (' Programmed Stop.')
C
      STOP
      END
C
C -49------------------- END OF SUBROUTINE STOPP ------------RBREAK-
C #50###################### SUBROUTINE OPENF ################RBREAK#
C
C     This subroutine opens a specified file
C
      SUBROUTINE OPENF (NUNIT,FNAME,FSTAT)
      CHARACTER FNAME*10,FSTAT*3
      OPEN (UNIT=NUNIT,FILE=FNAME,STATUS=FSTAT,ACCESS='SEQUENTIAL')
      RETURN
      END
C
C -50------------------ END OF SUBROUTINE OPENF -------------RBREAK-
```

Appendix C

Listing of

Computer Program BEFORR2

```
C=====================================================================C
C
C     #######  ########  ########  #######   #######  #######
C     ##   ##  ##        ##        ##     ## ##    ## ##    ##
C     ##   ##  ##        ##        ##     ## ##    ## ##    ##
C     #######  #######   #######   ##     ## #######  #######
C     ##   ##  ##        ##        ##     ## ## ##    ## ##
C     ##   ##  ##        ##        ##     ## ##  ##   ##  ##
C     #######  ########  ##         #######  ##    ## ##    ##
C
C
C                           ####
C                        #########
C                      #####  #####
C                            #####
C                          #####
C                        #####
C                      #####
C                     #############
C                    ##############
C
C
C
C
C
C          BEFORe Rbreak2 for numerically stable computation
C
C       1. Variation of bottom friction factor f' for each linear
C          segment of bottom profile
C       2. Specification of measured free surface oscillation at
C          seaward boundary (x=0) for IWAVE=3
C
C      Written by N. Kobayashi, A. Raichle, D. Cox, and M. Poff
C              Center for Applied Coastal Research
C                Department of Civil Engineering
C          University of Delaware, Newark, Delaware 19716
C                         September 1993
C
C ############################ PURPOSE ##############################
C
C     BEFORR2 should be executed before computer program RBREAK2 for
C         numerical simulation of random waves on impermeable
C         breakwaters and beaches.
C     There are three crucial computation parameters that need to be
C         specified in RBREAK2 computation:
C             1. INITS = number of computational nodes below SWL
C             2. NONE  = number of time steps in one wave period
C             3. DELTA = normalized water depth at computational
C                        waterline
C     The value of INITS and an initial value of DELTA are supplied
C         by users.
C     BEFORR2 and RBREAK2 break computation duration into computation
```

```
C          units. Each computation unit is one wave period long except
C          for the last unit, the length of which may be less than one
C          wave period.
C      The main purpose of BEFORR2 is to assign the values of NONE and
C          DELTA to each computation unit of RBREAK2 so as to eliminate
C          numerical difficulties, especially at the computational
C          waterline.
C      The "wave period" is the reference period used for the
C          normalization of the governing equations.
C      BEFORR2 reads RBREAK2's input data, checks some of them, and
C          writes the computation parameters it assigns in the output
C          file 'BINPUT', which will be the second (if IWAVE=1) or
C          third (if IWAVE>1) input data file for RBREAK2 computation.
C      Notes:
C          BEFORR2 and RBREAK2 share many subroutine and COMMON block
C          names, but they are independent programs.
C
C ######################## MAIN PROGRAM BEFORR2 #####################
C
C      Main program performs time-marching computation using
C      subroutines
C
      PROGRAM BEFORR2
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI,KCNO,MCNO,KC2
      CHARACTER*10 FINP1,FINP2
      INTEGER S,SM1,SP1,SSAV
C
C      COMMONs
C
C      -----------------------------------------------------------------
C       Name      Contents
C      -----------------------------------------------------------------
C      /DIMENS/  The values of FORTRAN PARAMETERs specified in the main
C                program
C      /CONSTA/  Basic constants
C      /ID/      Identifiers specifying user's options
C      /CPAR1/ to /CPAR4/ Computation parameters
C      /REQ1/ and /REQ2/ Identifiers and values associated with special
C                storing (see Subr. 02 INPUT1 for more explanation)
C      /WAVE1/   Physical wave parameters
C      /WAVE2/   Dimensionless wave parameters
C      /WAVE3/   Time series of normalized surface elevation at
C                seaward boundary
C      /WAVE4/   Maximum and minimum of time series in /WAVE3/
C      /WAVE5/   Cnoidal wave parameters (K, E, m and 1-m)
C      /WAVE6/   Number of data points specified in input file FINP2
C      /BOT1/    Physical quantities related to structure
C      /BOT2/    Normalized quantities related to structure
C      /BOT3/    Normalized structure geometry
C      /BOT4/ and /BOT5/ Physical 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     /ARMOR/   Armor stability parameters read as input
C     /SAVBC/   Node numbers related to options ISAVB=1 and ISAVC=1
C     /VALUEN/  Values at present time level stored during computation
C     /DIAGID/, /DIAGV1/, and /DIAGV2/ Indicators and saving variables
C               for diagnostic purposes
C     ---------------------------------------------------------------
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CONSTA/ PI,GRAV
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  MULTIF,NONEM,NONE,NEND,NRATE
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /REQ1/   IREQ,IELEV,IV,IDUDT,ISNR,NREQ
      COMMON /REQ2/   TREQ(N5)
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /WAVE3/  ETA(N2)
      COMMON /WAVE4/  ETAMAX,ETAMIN
      COMMON /WAVE5/  KCNO,ECNO,MCNO,KC2
      COMMON /WAVE6/  NDATA
      COMMON /BOT1/   DSEAP,DLANDP,FWP(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3)
      COMMON /ARMOR/  C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      COMMON /SAVBC/  NNODB,NNODC,NODB(N5),NODC(N5)
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      COMMON /DIAGID/ INONE,ITRY,IREV,LEVEL
      COMMON /DIAGV1/ SSAV(3),JMSAV(3),NONESV(N6),MFIVE
      COMMON /DIAGV2/ USAV(6,N1),VSAV(3,N1),ESAV(3,N1),DELSAV(3),
     +                DELTSV(N6)
C
C           READ INPUT
C           -----------------------------------------------------
C           Subr. 01 OPENIO opens input and output files
C           Subr. 02 INPUT1 reads primary input data
C           Subr. 03 INPUT2 reads wave profile at seaward boundary
C           -----------------------------------------------------
      WRITE (*,*) 'BEFORR reports progress on screen every MREP waves'
      WRITE (*,*) 'Enter MREP (0 if you do not want the report)'
```

```
      READ  (*,*)  MREP
      WRITE (*,*) 'Name of primary input-data-file?'
      READ  (*,5000) FINP1
C
      CALL OPENIO (1,FINP1,FINP2)
      CALL INPUT1 (1,FINP2)
      CALL OPENIO (2,FINP1,FINP2)
      IF (IWAVE.GT.1) CALL INPUT2 (1)
 5000 FORMAT (A10)
C
C             CHECK FORTRAN PARAMETERS
C             -------------------------------------------------------
C             Each subroutine having FORTRAN "PARAMETER" statement is
C             called by Subroutine 39 CKSUBR to check its "PARAMETER"
C.            values, which must be the same as their counterparts in
C             the main program. These checking calls to subroutines
C             are indicated by MODE=0, as opposed to "working" calls
C             indicated by MODE=1 as used above in Call INPUT1 (1,...)
C             and Call INPUT2 (1).
C             -------------------------------------------------------
      CALL CKSUBR (N1,N2,N3,N4,N5,N6)
      MODE=1
C
C             GROUNDWORK
C             -------------------------------------------------------
C           . Subr. 04 BOTTOM computes normalized structure geometry
C           . Subr. 05 WPARAM calculates wave parameters
C           . This call to Subr. 10 CPARAM calculates NONEM = mininum
C             NONE
C           . Subr. 26 INIT specifies initial conditions
C           . This call to Subr. 06 REGWAV is for initial incident
C             wave profile and documentation purpose in Subr. 31 DOC1
C             where NONE=NONEM is used
C           . Subr. 31 DOC1 writes essential information before
C             time-marching computation
C             -------------------------------------------------------
      CALL BOTTOM (MODE)
      CALL WPARAM (MODE)
      CALL CPARAM (0,0,0,0.D+00)
      TIME = 0.D+00
      CALL INIT   (MODE)
      IF (IWAVE.EQ.1) CALL REGWAV (MODE)
      CALL DOC1   (MODE,FINP2)
      IF (IJOB.EQ.3) SM1=INITS-1
C
C             TIME MARCHING COMPUTATION
C             -------------------------------------------------------
C             Loop with bigger number nests to one with smaller number
C             . 510 contains one set of five computation units indi-
C                   cated by MM (MFIVE is defined in Subr. 02 INPUT1)
C             . 520 represents one computation unit indicated by M
C             . 530 constitutes one try for a computation unit,
```

```
C                 indicated by ITRY
C               . 540 performs time-marching from one time level to the
C                 next, indicated by N
C              ---------------------------------------------------------
      M     = 0
      MM    = 0
      ILEV3 = 0
      IPROB = 0
  510 IF (MM.LT.MFIVE) THEN
C       ========: 510 Begins :========================================
C       IREV = reverse indicator only for random waves.
C          IPROB=9 dictates that computation be reversed to either
C          LEVEL=2 or LEVEL=3 (reversal to LEVEL=3 can be made only
C          once when ILEV3=0). IREV indicates how many times reversal
C          has been made.
C       LEVEL=2: beginning of the present set of five comp. units
C       LEVEL=3: beginning of the previous set of five comp. units
C       Subr. 28 INITMM sets initial hydrodynamic quantities at the
C          beginning of one set of five computation units using the
C          saved values at LEVEL=2 or LEVEL=3
C       Subr. 30 SAVEMM saves hydrodynamic quantities at the end of
C          one set of five computation units (LEVEL=2 and LEVEL=3)
C       --------:
        IF (IPROB.EQ.9) THEN
          IF (IREV.EQ.5) CALL STOPP (1,2)
          IREV = IREV+1
          MFAR = M-5*(MM-1)
          IF (MFAR.GE.2.OR.ILEV3.EQ.1.OR.MM.EQ.1) THEN
            LEVEL = 2
          ELSE
            LEVEL = 3
            ILEV3 = 1
            MM = MM-1
          ENDIF
          CALL INITMM (MODE)
          IPROB = 0
        ELSE
          CALL SAVEMM (MODE)
          IREV  = 0
          ILEV3 = 0
          MM = MM+1
        ENDIF
        M = (MM-1)*5
        IF (MM.LT.MFIVE) THEN
          MEND = MM*5
        ELSE
          MEND = MWAVE
        ENDIF
  520   IF (M.LT.MEND.AND.IPROB.NE.9) THEN
C         ------: 520 Begins :----------------------------------------
          M = M+1
          ITRY = 0
```

```
  530     CONTINUE
C           ----: 530 begins :-----------------------------------------
C         . Subr. 27 INITM sets initial hydrodynamic quantities at the
C           beginning of a computation unit using the saved values at
C           LEVEL=1
C         . Subr. 10 CPARAM assigns computation parameters NONE and
C           DELTA. When NONE is considered too big, IPROB=9 is set,
C           meaning that computation must be reversed to LEVEL=2 or
C           LEVEL=3. Otherwise, IPROB=0 at the beginning of each trial.
C         . Subr. 06 REGWAV computes incident regular wave profile
C           ----:
          ITRY = ITRY+1
          IF (IPROB.NE.0) CALL INITM (MODE)
          CALL CPARAM (MODE,M,IPROB,U2STOP)
          IF (INONE.EQ.1) CALL REGWAV (MODE)
          N = 0
  540     IF (IPROB.EQ.0.AND.N.LT.NEND) THEN
C             --: 540 Begins -----------------------------------------
C           . TIME = normalized time
C           . Subr. 12 EXTRAP estimates hydrodynamic quantities at the
C             node immediately landward of the present waterline node
C           . Subr. 13 RETAIN retains some values at present time
C             level at landward and seaward boundaries
C           . Subr. 14 CRITV calculates critical velocities used in
C             characteristic variables and checks for negative water
C             depth (IPROB=1)
C           . Subr. 15 MARCH performs time-marching from present time
C             level (N-1) to next (N) excluding landward and seaward
C             boundaries. If the waterline moves seaward more than one
C             grid spacing, IPROB=2 is set.
C           . Landward and seaward boundary conditions are handled by
C             Subr. 16 LANDBC and 19 SEABC, respectively
C           . Subr. 11 NUMSTA checks if numerical stability criterion
C             is violated (IPROB=3)
C             --:
            N = N+1
            TIME = DBLE(M-1) + DBLE(N)/DBLE(NONE)
            IF (IJOB.LT.3) CALL EXTRAP (MODE)
            CALL RETAIN (MODE)
            CALL CRITV  (MODE,M,IPROB)
            IF (IPROB.EQ.0) THEN
              CALL MARCH (MODE,M,IPROB,U2STOP)
              IF (IPROB.EQ.0) THEN
                CALL LANDBC (MODE,M)
                CALL SEABC  (MODE,M,N)
                CALL NUMSTA (MODE,M,IPROB)
                IF (IJOB.LT.3.AND.S.GT.JMAX) JMAX=S
              ENDIF
            ENDIF
C             --: 540 Ends :-----------------------------------------
          GOTO 540
          ENDIF
```

```
C          ----: 530 Ends :----------------------------------------
         IF (IPROB.EQ.0) THEN
C          ::     Subr. 32 DOC2 writes computation log during
C                 time-marching computation
C          ::     Subr. 29 SAVEM saves hydrodynamic quantities at the
C                 end of a computation unit (LEVEL=1)
           CALL DOC2  (MODE,M,MREP)
           CALL SAVEM (MODE)
         ELSE
           IF (IPROB.NE.9) GOTO 530
         ENDIF
C          ------: 520 Ends :----------------------------------------
       GOTO 520
       ENDIF
C .      ========: 510 Ends :========================================
     GOTO 510
     ENDIF
C
C            FINISHING
C            --------------------------------------------------
C            Subr. 33 DOC3 writes in file 'BINPUT' and completes
C            documentation in file 'BDOC'
C            --------------------------------------------------
     CALL DOC3 (MODE)
C
     STOP
     END
C
C -------------------- END OF MAIN PROGRAM BEFORR -----------------
C #01####################### SUBROUTINE OPENIO  ###############BEFORR#
C
C    This subroutine opens input and output files
C
     SUBROUTINE OPENIO (ICALL,FINP1,FINP2)
C
     COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
    +                ISAVA,ISAVB,ISAVC
     CHARACTER OL*3,NE*3,SE*10,FINP1*10,FINP2*10
     DATA OL,NE,SE /'OLD','NEW','SEQUENTIAL'/
C
C  DESCRIPTION OF INPUT AND OUTPUT FILES
C  ------------------------------------------------------------------
C  (Category) Unit (Filename) Purpose
C  ------------------------------------------------------------------
C  (Input)  11 (FINP1)      contains primary input data
C  (Input)  12 (FINP2)      contains time series of normalized  surface
C                           elevation at seaward boundary
C  (Output) 51 ('BINPUT')  stores computation parameters to be used
C                           as input to RBREAK2 computation
C                           --> m,NONE(m),NEND(m),DELTA(m)
C                               m = 1,2,...,MWAVE = computation unit
C  (Output) 52 ('BSPACE')  stores structure geometry
```

```
C                            --> JE,(XB(j),ZB(j),j=1,JE)
C  (Output) 98 ('BDOC')      stores essential output for concise
C                            documentation
C  (Output) 99 ('BMSG')      stores messages written under special
C                            circumstances during computation
C  ---------------------------------------------------------------------
      IF (ICALL.EQ.1) THEN
        OPEN (UNIT=11,FILE=FINP1   ,ACCESS=SE,STATUS=OL)
        OPEN (UNIT=51,FILE='BINPUT',ACCESS=SE,STATUS=NE)
        OPEN (UNIT=52,FILE='BSPACE',ACCESS=SE,STATUS=NE)
        OPEN (UNIT=98,FILE='BDOC'  ,ACCESS=SE,STATUS=NE)
        OPEN (UNIT=99,FILE='BMSG'  ,ACCESS=SE,STATUS=NE)
      ELSE
        IF (IWAVE.GT.1) OPEN (UNIT=12,FILE=FINP2,ACCESS=SE,STATUS=OL)
      ENDIF
      RETURN
      END
C
C -01------------------- END OF SUBROUTINE OPENIO ------------BEFORR-
C #02####################### SUBROUTINE INPUT1  ###############BEFORR#
C
C     This subroutine reads data from primary input data file and
C     checks some of them
C
      SUBROUTINE INPUT1 (MODE,FINP2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      CHARACTER*10 FINP2
      CHARACTER*5 COMMEN(15)
      INTEGER S,SM1,SP1,SSAV
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CONSTA/ PI,GRAV
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  MULTIF,NONEM,NONE,NEND,NRATE
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /REQ1/   IREQ,IELEV,IV,IDUDT,ISNR,NREQ
      COMMON /REQ2/   TREQ(N5)
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /BOT1/   DSEAP,DLANDP,FWP(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      COMMON /BOT4/   NBSEG
      COMMON /BOT5/   WBSEG(N4),TBSLOP(N4),XBSEG(N4),ZBSEG(N4)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3)
      COMMON /ARMOR/  C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
```

```
      COMMON /SAVBC/  NNODB,NNODC,NODB(N5),NODC(N5)
      COMMON /DIAGV1/ SSAV(3),JMSAV(3),NONESV(N6),MFIVE
      IF (MODE.EQ.0) THEN
C       ::   MODE=0 is used in Subr. 39 CKSUBR
C       ::   Although not used in any of the above COMMON blocks, N1
C            needs to be checked in relation with Call CKVAL1 (5,...)
        CALL CKFPAR (2,1,N1,N1R)
        CALL CKFPAR (2,3,N3,N3R)
        CALL CKFPAR (2,4,N4,N4R)
        CALL CKFPAR (2,5,N5,N5R)
        CALL CKFPAR (2,6,N6,N6R)
        RETURN
      ENDIF
C
C           COMMENT LINES
C           -------------------------------------------------------------
C           NLINES = number of comment lines preceding primary
C                    input data
C           Comment lines are read from primary input file and then
C           written as the heading of output files 'BINPUT', 'BDOC',
C           and 'BMSG'
C           -------------------------------------------------------------
      READ  (11,1110) NLINES
      WRITE (51,1110) NLINES
      DO 110 L = 1,NLINES
        READ  (11,1120) (COMMEN(I),I=1,15)
        WRITE (51,1120) (COMMEN(I),I=1,15)
        WRITE (98,1120) (COMMEN(I),I=1,15)
        WRITE (99,1120) (COMMEN(I),I=1,15)
  110 CONTINUE
C
C           OPTIONS
C           -------------------------------------------------------------
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           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 beaches
C           IENERG=0: Energy quantities NOT computed
C                 =1: Energy quantities computed
C           IWAVE=1: Incident waves at seaward boundary computed
C                =2: Incident waves at seaward boundary given as input
C                =3: Total waves at seaward boundary given as input
C           If IWAVE>1 --> Must specify MSTAT and FINP2
C             . FINP2 = name of input data file containing time series
```

```
C               of normalized free surface elevation at seaw. boundary
C             . Statistical calculations are performed excluding the
C               first MSTAT computation units
C             . A "computation unit" is one wave period long except
C               for the last unit, the length of which may be less
C               than one wave period
C             . The "wave period" is the reference period used for the
C               normalization of the governing equations
C           "A" = Spatial variations of hydrodynamic quantities at
C                 specified moments
C           "B" = Time series of volume flux and water depth at
C                 specified nodes
C           "C" = Time series of the following:
C                 . For ISTAB=1: stability number SNR at specified nodes
C                 . For ISTAB=2: displacement of armor units from
C                                specified initial nodal locations
C           "B" and "C" are stored at rate NRATE data points per wave
C                 period
C           ISAVA,ISAVB,ISAVC are identifiers associated with saving
C                 "A","B","C", respectively (1=save; 0=no)
C           MSAVA1, MSAVA2, AND NTIMES:
C           If ISAVA=1, "A" is saved NTIMES (>1) times at equal
C                              intervals per wave period from computation
C                              unit MSAVA1 to MSAVA2, inclusive
C           If ISAVB=1 --> Must specify NNODB
C           If ISAVC=1 --> Must specify NNODC
C           NNODB,NNODC = number of nodes for which "B","C" is to be
C                         saved
C           IREQ=0: No special storing
C               =1: Special storing requested
C           Special storing = storing spatial variations of requested
C               quantities at normalized time TREQ(i)
C               with i=1,2,...,NREQ
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           Notes: . DUDT can be requested only if ISTAB>0
C                  . SNR  can be requested only if ISTAB=1
C           -----------------------------------------------------------
      READ  (11,1130) IJOB,ISTAB
      READ  (11,1130) ISYST
      READ  (11,1130) IBOT
      READ  (11,1130) INONCT
      READ  (11,1130) IENERG
      READ  (11,1140) IWAVE,MSTAT,FINP2
      READ  (11,1150) ISAVA,MSAVA1,MSAVA2,NTIMES
      READ  (11,1150) ISAVB,NNODB
      READ  (11,1150) ISAVC,NNODC
```

```
      READ  (11,1160) IREQ,IELEV,IV,IDUDT,ISNR,NREQ
C     ::    Check user-chosen options using Subr. 35 CKOPTI
      INDIC=0
      CALL CKOPTI ( 1,INDIC,IJOB  ,1,3)
      CALL CKOPTI ( 2,INDIC,ISTAB ,0,2)
      CALL CKOPTI ( 3,INDIC,ISYST ,1,2)
      CALL CKOPTI ( 4,INDIC,IBOT  ,1,2)
      CALL CKOPTI ( 5,INDIC,INONCT,0,1)
      CALL CKOPTI ( 6,INDIC,IENERG,0,1)
      CALL CKOPTI ( 7,INDIC,IWAVE ,1,3)
      CALL CKOPTI ( 8,INDIC,ISAVA ,0,1)
      CALL CKOPTI ( 9,INDIC,ISAVB ,0,1)
      CALL CKOPTI (11,INDIC,IREQ  ,0,1)
      IF (ISTAB.GT.0) CALL CKOPTI (10,INDIC,ISAVC,0,1)
      IF (IREQ.EQ.1) THEN
        CALL CKOPTI (12,INDIC,IELEV,0,1)
        CALL CKOPTI (13,INDIC,IV   ,0,1)
        CALL CKOPTI (14,INDIC,IDUDT,0,1)
        CALL CKOPTI (15,INDIC,ISNR ,0,1)
      ENDIF
      IF (INDIC.GT.0) CALL STOPP (3,4)
C     ::    Subr. 37,38 CKVAL1,CKVAL2 check user-specified values
C           Subr. 40 STOPP stops execution of the computation
      IF (IREQ.EQ.1) THEN
        CALL CKVAL1 (1,NREQ,1,N5)
        IF (ISTAB.EQ.0) IDUDT=0
        IF (ISTAB.NE.1) ISNR=0
        NOREQ = IELEV+IV+IDUDT+ISNR
        IF (NOREQ.EQ.0) THEN
          CALL STOPP (5,7)
        ELSE
          READ (11,1180) (TREQ(I),I=1,NREQ)
        ENDIF
      ELSE
        NREQ = 0
      ENDIF
      IF (ISAVB.EQ.1) CALL CKVAL1 (2,NNODB,1,N5)
      IF (ISAVC.EQ.1) CALL CKVAL1 (3,NNODC,1,N5)
C
C           COMPUTATION PARAMETERS
C           --------------------------------------------------------
C           INITS = . number of spatial nodes along the bottom
C                     below SWL (IJOB<3)
C                   . number of nodes between seaward and
C                     landward boundaries, inclusive (IJOB=3)
C           Notes:  . INITS should be so large that delta x between
C                     two adjacent nodes is sufficiently small
C                   . INITS=100 to 600 has been used
C           MULTIF = multiplication factor associated with NONEM
C           Notes:  . NONEM = minimum allowable value of NONE
C                   . NONE  = number of time steps in one wave period
C                   . See Subr. 10 CPARAM for more explanation about
```

```
C                      MULTIF and NONEM
C           DELTA as input is an initial value for normalized water
C                 depth defining computational waterline. A range of
C                 values of DELTA from 0.001 to 0.003 has been used.
C           TMAX  = normalized duration of computation
C           NRATE = rate (data points per per wave period) of storing
C                   output time series
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)
C           X1,X2 = damping coefficients
C           ---------------------------------------------------------
      READ  (11,1110) INITS
      READ  (11,1110) MULTIF
      READ  (11,1170) DELTA
      READ  (11,1170) TMAX
      READ  (11,1110) NRATE
      READ  (11,1110) NDELR
      CALL CKVAL1 (5,INITS,1,N1-1)
      IF (IJOB.LT.3) THEN
        CALL CKVAL1 (4,NDELR,1,N3)
      ELSE
        NDELR = 0
      ENDIF
      INDIC=0
      DO 120 L = 1,NDELR
        READ (11,1180) DELRP(L)
        IF (DELRP(L).LE.0.D+00) CALL CKVAL2 (2,INDIC,'DELRP ')
  120 CONTINUE
      READ (11,1180) X1,X2
C
C           SOME MORE CHECKS
C           ---------------------------------------------------------
C           MWAVE = number of computation units
C           MFIVE = number of sets of five computation units
C           MSTAT: Statistical calculations are performed excluding
C                  the first MSTAT computation units
C           MSTAB: Computation of armor stability or movement is
C                  performed excluding the first MSTAB computation
C                  units
C           ---------------------------------------------------------
      ITMAX = INT(TMAX)
      RES   = DMOD(TMAX,1.D+00)
      IF (DABS(RES).LT.1.D-08) THEN
        MWAVE = ITMAX
      ELSE
        MWAVE = ITMAX+1
        IF (IWAVE.EQ.1) CALL STOPP (8,8)
      ENDIF
      IRES = MOD(MWAVE,5)
```

```
      IF (IRES.EQ.0) THEN
        MFIVE = MWAVE/5
      ELSE
        MFIVE = MWAVE/5+1
      ENDIF
      DO 130 I = 1,NREQ
        IF (TREQ(I).GT.TMAX)   CALL STOPP  (9,9)
        IF (TREQ(I).LE.0.D+00) CALL CKVAL2 (2,INDIC,'TREQ  ')
  130 CONTINUE
      CALL CKVAL1 (8,MSTAT,0,ITMAX-1)
      IF (IWAVE.EQ.1) THEN
        MSTAT = MWAVE-1
      ENDIF
      IF (ISTAB.EQ.1) THEN
        MSTAB = MSTAT
      ELSEIF (ISTAB.EQ.2) THEN
        MSTAB = 0
      ELSE
        MSTAB = MWAVE+1
        ISAVC = 0
      ENDIF
      IF (ISAVA.EQ.1) THEN
        IF (NTIMES.LT.1) CALL CKVAL2 (1,INDIC,'NTIMES')
        CALL CKVAL1  (9,MSAVA2,1,ITMAX)
        CALL CKVAL1 (10,MSAVA1,1,MSAVA2)
      ELSE
        MSAVA1=MWAVE+1
      ENDIF
C
C           CONSTANTS
C           ----------------------------------------------------------
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)
C           ----------------------------------------------------------
      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           WAVE PROPERTIES AND FRICTION FACTOR
C           ----------------------------------------------------------
C           HREFP = physical wave height at "reference" location
C                   . in meters if ISYST=1 (SI)
C                   . in feet   if ISYST=2 (USCS)
C           TP    = physical reference wave period, in seconds
C           HREFP and TP are used to normalize the governing
C                   equations
C           KSREF = shoaling coefficient at "reference" location
```

```
C           KSSEA = shoaling coefficient at seaward boundary
C           SIGMA = ratio between horizontal and vertical length scales
C           ----------------------------------------------------------
      READ  (11,1180) HREFP,TP
      READ  (11,1180) KSREF,KSSEA
      IF (HREFP.LE.0.D+00) CALL STOPP (10,10)
      SIGMA = TP*DSQRT(GRAV/HREFP)
C
C           STRUCTURE GEOMETRY
C           ----------------------------------------------------------
C           The structure geometry is divided into segments of
C           DIFFERENT INCLINATION OR ROUGHNESS
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             FWP(I)    = BOTTOM FRICTION FACTOR
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)
C           NOTE: FOR IBOT=2, SPECIFY FWP(NBSEG+1)=0.0, WHICH IS
C                 NOT USED FOR THE FOLLOWING COMPUTATION.
C           ----------------------------------------------------------
      READ  (11,1180) DSEAP
      READ  (11,1180) TSLOPS
      READ  (11,1110) NBSEG
      CALL CKVAL1 (6,NBSEG,1,N4-1)
      IF (IBOT.EQ.1) THEN
        DO 140 K = 1,NBSEG
          READ (11,1180) WBSEG(K),TBSLOP(K),FWP(K)
          IF (WBSEG(K).LE.0.D+00) CALL CKVAL2 (2,INDIC,'WBSEG ')
  140   CONTINUE
      ELSE
        DO 150 K = 1,NBSEG+1
          READ (11,1180) XBSEG(K),ZBSEG(K),FWP(K)
C         IF (XBSEG(K).LT.0.D+00) CALL CKVAL2 (3,INDIC,'XBSEG ')
  150   CONTINUE
      ENDIF
C
C           DATA RELATED TO SAVING "B", i.e., time series of volume
C           flux and total water depth at specified nodes
C           ----------------------------------------------------------
C           NNODB   = no. of nodes for which "B" is to be saved
C           NODB(I) = I-th node number for which "B" is to be saved
C           ----------------------------------------------------------
```

```
      IF (ISAVB.EQ.1) READ (11,1190) (NODB(I),I=1,NNODB)
C
C           DATA RELATED TO ARMOR STABILITY AND MOVEMENT
C           ----------------------------------------------------------
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           NNODC   = no. of nodes for which "C" is to be saved
C           NODC(I) = I-th node number for which "C" is to be saved
C           "C" = time series of the following
C               . For ISTAB=1: stability number SNR at specified nodes
C               . For ISTAB=2: displacement of armor units from
C                              specified initial nodal locations
C           ----------------------------------------------------------
C     ::    To compute SNR = stability number against rolling/sliding:
      IF (ISTAB.GT.0) THEN
        READ (11,1180) C2,C3,SG
        READ (11,1180) CD,CL,CM
        READ (11,1180) TANPHI
        READ (11,1180) AMAX,AMIN
        IF (ISAVC.EQ.1) READ (11,1190) (NODC(I),I=1,NNODC)
        IF (C2     .LE.0.D+00) CALL CKVAL2 (2,INDIC,'C2    ')
        IF (C3     .LE.0.D+00) CALL CKVAL2 (2,INDIC,'C3    ')
        IF (SG     .LE.0.D+00) CALL CKVAL2 (2,INDIC,'SG    ')
        IF (CD     .LE.0.D+00) CALL CKVAL2 (2,INDIC,'CD    ')
        IF (CL     .LT.0.D+00) CALL CKVAL2 (3,INDIC,'CL    ')
        IF (CM     .LT.0.D+00) CALL CKVAL2 (3,INDIC,'CM    ')
        IF (TANPHI.LE.0.D+00) CALL CKVAL2 (2,INDIC,'TANPHI')
      ENDIF
C     ::    To compute movement of armor units, additional input is
C           required:
      IF (ISTAB.EQ.2) THEN
        READ (11,1180) DAP
        IF (DAP.LE.0.D+00) CALL CKVAL2 (2,INDIC,'DAP   ')
      ENDIF
C     ::    Subroutine 36 CKVAL performs check on a number of user-
C           specified values. The check in that subroutine involves
C           IF statements, and has to be separated from the present
C           subroutine because some FORTRAN compilers may have
C           problems with subroutines having too many IF statements.
      CALL CKVAL (INDIC)
      IF (INDIC.GT.0) CALL STOPP (3,4)
```

```
C
C          FORMATS
C          ----------------------------------------------------------
 1110 FORMAT (3I8)
 1120 FORMAT (15A5)
 1130 FORMAT (2I1)
 1140 FORMAT (I1,I8,2X,A10)
 1150 FORMAT (I1,3I8)
 1160 FORMAT (5I1,I8)
 1170 FORMAT (F15.6)
 1180 FORMAT (3F13.6)
 1190 FORMAT (5I6)
C
      RETURN
      END
C
C -02------------------ END OF SUBROUTINE INPUT1 ------------BEFORR-
C #03####################### SUBROUTINE INPUT2 ###############BEFORR#
C
C     This subroutine reads wave profile at seaward boundary
C
      SUBROUTINE INPUT2 (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /WAVE3/  ETA(N2)
      COMMON /WAVE4/  ETAMAX,ETAMIN
      COMMON /WAVE6/  NDATA
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (3,2,N2,N2R)
        RETURN
      ENDIF
C
C             READ DATA POINTS
C             ------------------------------------------------------
      READ (12,9000) NDATA
      CALL CKVAL1 (7,NDATA,1,N2)
      READ (12,8000) (ETA(I),I=1,NDATA)
 8000 FORMAT (5D15.6)
 9000 FORMAT (I8)
C
C             FIND EXTREME VALUES
C             ------------------------------------------------------
C             ETA = given time series of free surface profile at
C                   seaward boundary
C             ETAMAX,ETAMIN = maximum,minimum of ETA
C             ------------------------------------------------------
      ETAMAX = -1.D+03
      ETAMIN =  1.D+03
      DO 100 I = 1,NDATA
        IF (ETA(I).GT.ETAMAX) ETAMAX=ETA(I)
```

```
        IF (ETA(I).LT.ETAMIN) ETAMIN=ETA(I)
  100 CONTINUE
C
      RETURN
      END
C
C -03------------------- END OF SUBROUTINE INPUT2 ------------BEFORR-
C #04###################### SUBROUTINE BOTTOM  ###############BEFORR#
C
C     This subroutine calculates normalized structure geometry and
C     delta x between two adjacent nodes
C
      SUBROUTINE BOTTOM (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      DIMENSION TSLOPE(N1),FWNODE(N1)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CONSTA/ PI,GRAV
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WL0,WL,UR,URPRE,KSI,SIGMA
      COMMON /BOT1/   DSEAP,DLANDP,FWP(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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 /SAVBC/  NNODB,NNODC,NODB(N5),NODC(N5)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (4,1,N1,N1R)
        CALL CKFPAR (4,4,N4,N4R)
        CALL CKFPAR (4,5,N5,N5R)
        RETURN
      ENDIF
C
C           COMPLETE SEGMENT DATA NOT SPECIFIED AS INPUT
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           The structure geometry is divided into segments of
C           DIFFERENT INCLINATION OR ROUGHNESS
```

```
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           . FWP(I)    = BOTTOM FRICTION FACTOR
C           . XBSEG(i)  = physical horizontal distance from seaward
C                         boundary to the segment's seaward-end
C           . ZBSEG(i)  = physical water depth below SWL
C                         (+ below SWL) at the segment's seaward-end
C           BSWL,DSEAP,WBSEG,XBSEG,ZBSEG are
C           . in meters if ISYST=1 (SI)
C           . in feet   if ISYST=2 (USCS)
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           INITS = . number of spatial nodes along the bottom below
C                     SWL (IJOB<3)
C                   . number of nodes between seaward and landward
C                     boundaries, inclusive (IJOB=3)
C           ---------------------------------------------------------
      IF (IJOB.LT.3) THEN
        K = 0
  910   CONTINUE
          IF (K.EQ.NBSEG) CALL STOPP (11,12)
          K = K+1
          CROSS = ZBSEG(K)*ZBSEG(K+1)
        IF (CROSS.GT.0.D+00) GOTO 910
        BSWL = XBSEG(K+1) + ZBSEG(K+1)/TBSLOP(K)
        X    = BSWL/DBLE(INITS)
      ELSE
        BSWL = XBSEG(NBSEG+1)
        X    = BSWL/DBLE(INITS-1)
        DO 130 K = 1,NBSEG+1
          IF (ZBSEG(K).LT.0.D+00) CALL STOPP (13,14)
  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 (physical,
C                      later normalized under the same name)
C          TSLOPE(j) = tangent of local slope at node j
C          FWNODE(J) = BOTTOM FRICTION FACTOR AT NODE J
C          ----------------------------------------------------------
      IF (IJOB.LT.3) THEN
        DUM = XBSEG(NBSEG+1)/X
        JE  = INT(DUM)+1
      ELSE
        JE = INITS
      ENDIF
      IF (JE.GT.N1) THEN
        WRITE  (*,9910) JE,N1
        WRITE (99,9910) JE,N1
        STOP
      ELSE
        JE1 = JE-1
      ENDIF
C     ::  Some checks associated with storing time series
      IF (ISAVB.EQ.1) THEN
        DO 140 I = 1,NNODB
          CALL CKVAL1 (11,NODB(I),1,JE)
  140   CONTINUE
      ENDIF
      IF (ISAVC.EQ.1) THEN
        DO 150 I = 1,NNODC
          CALL CKVAL1 (12,NODC(I),1,JE)
  150   CONTINUE
      ENDIF
 9910 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 160 J = 1,JE
        DIST = DIST + X
        IF (DIST.GT.XCUM.AND.K.LT.NBSEG) THEN
  920     CONTINUE
            K    = K+1
            XCUM = XBSEG(K+1)
          IF (DIST.GT.XCUM.AND.K.LT.NBSEG) GOTO 920
        ENDIF
        U2INIT(J) = ZBSEG(K) - (DIST-XBSEG(K))*TBSLOP(K)
        TSLOPE(J) = TBSLOP(K)
```

```
         FWNODE(J) = FWP (K)
  160 CONTINUE
C
C           NORMALIZATION
C           -------------------------------------------------------
C           WTOT = normalized width of computation domain
C           At node j:
C           . U2INIT(j) = normalized water depth below SWL
C                         (+ below SWL)
C           . THETA(j)  = normalized tangent of local slope
C           . FW(J)     = NORMALIZED BOTTOM FRICTION FACTOR
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 170 J = 1,JE
        U2INIT(J) = U2INIT(J)/HREFP
        THETA(J)  = TSLOPE(J)*SIGMA
        DIST      = DIST + X
        XB(J)     = DIST
        ZB(J)     = -U2INIT(J)
        FW(J)     = .5D+00*SIGMA*FWNODE(J)
  170 CONTINUE
C
      RETURN
      END
C
C -04------------------ END OF SUBROUTINE BOTTOM ------------BEFORR-
C #05###################### SUBROUTINE WPARAM ###############BEFORR#
C
C     This subroutine calculates wave parameters
C
      SUBROUTINE WPARAM (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CONSTA/ PI,GRAV
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /BOT1/   DSEAP,DLANDP,FWP(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      COMMON /BOT3/   U2INIT(N1),THETA(N1),SSLOPE(N1),XB(N1),ZB(N1)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
```

```
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (5,1,N1,N1R)
        CALL CKFPAR (5,3,N3,N3R)
        RETURN
      ENDIF
C
C           PARAMETERS RELATED TO WAVE AND SLOPE CHARACTERISTICS
C           -----------------------------------------------------------
C           KSI    = surf similarity parameter
C           WLOP   = physical deep-water wavelength
C           DSEAP  = phys. water depth below SWL at seaward boundary
C           DLANDP = phys. water depth below SWL at landward boundary
C                    (only for IJOB=3)
C           DELRP  = water depths associated with visual or measured
C                    waterline
C           WLO,DSEA,DLAND,DELTAR ARE THE NORMALIZED COUNTERPARTS
C           OF WLOP,DSEAP,DLANDP,DELRP, RESPECTIVELY
C           -----------------------------------------------------------
      WLOP   = GRAV*(TP*TP)/(2.D+00*PI)
      WLO    = WLOP/DSEAP
      KS     = KSSEA/KSREF
      KSI    = SIGMA*TSLOPS/DSQRT(2.D+00*PI)
      DSEA   = DSEAP/HREFP
      DSEAKS = DSEA/KS
      DSEA2  = DSQRT(DSEA)
      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(INITS)
        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
C                wavelength
C           -----------------------------------------------------------
      TWOPI = 2.D+00*PI
      WL    = WLO
      FUN1  = WL - WLO*DTANH(TWOPI/WL)
  900 IF (DABS(FUN1).GT.1.D-04) THEN
        FUN2 = 1.D+00 + WLO*TWOPI/(WL*DCOSH(TWOPI/WL))**2
        WL   = WL - FUN1/FUN2
        FUN1 = WL - WLO*DTANH(TWOPI/WL)
```

```
      GOTO 900
      ENDIF
      UR    = WL*WL/DSEAKS
      URPRE = UR
C
      RETURN
      END
C
C -05------------------ END OF SUBROUTINE WPARAM ------------BEFORR-
C #06###################### SUBROUTINE REGWAV ###############BEFORR#
C
C     This subroutine computes incident regular wave profile at
C     seaward boundary
C     Wave profile: Stokes II if UR<26
C                   Cnoidal otherwise
C
      SUBROUTINE REGWAV (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      DOUBLE PRECISION KCNO,MCNO,KC2,KC,KS,KSREF,KSSEA,KSI
      DIMENSION ETAU(N2)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CONSTA/ PI,GRAV
      COMMON /CPAR2/  MULTIF,NONEM,NONE,NEND,NRATE
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /WAVE3/  ETA(N2)
      COMMON /WAVE4/  ETAMAX,ETAMIN
      COMMON /WAVE5/  KCNO,ECNO,MCNO,KC2
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      IF (MODE.EQ.0) THEN
        CALL CKFPAR(6,1,N1,N1R)
        CALL CKFPAR (6,2,N2,N2R)
        RETURN
      ENDIF
C
C           CHECK DIMENSION N2
C           ---------------------------------------------------------
      IF (NONE.GT.(N2-1)) THEN
        WRITE  (*,9910) N2,NONE,TIME
        WRITE (99,9910) N2,NONE,TIME
        STOP
      ENDIF
 9910 FORMAT (/' From Subr. 06 REGWAV:'/
     +       ' Specify larger dimension N2 to facilitate computation'/
     +       ' of incident regular wave profile at seaward boundary.'/
     +       ' N2 must be greater than NONE.'/
     +       ' Currently,              N2 =,I8'/
     +       ' Current try requires NONE =,I8'/
     +       ' Time of occurrence   TIME =,F15.6'/
```

```
     +       ' Provide enough space for possible increase in NONE.'/
     +       ' Computation aborted.')
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          NO = approx. 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.D+00) THEN
C          ::    Stokes II Wave Profile
        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
        NO   = 1
        DO 110 N = 1,NHALF1
          TIMEN = DBLE(N-1)*T
          ETAU(N) = .5D+00*DCOS(TWOPI*TIMEN)+AMP2*DCOS(FOURPI*TIMEN)
          ETAU(N) = KS*ETAU(N)
          IF (N.GT.1) THEN
            IF(ETAU(N).LE.0.D+00.AND.ETAU(N-1).GT.0.D+00)NO=N
          ENDIF
  110   CONTINUE
        ETAMIN = ETAU(NHALF1)
        ETAMAX = ETAU(1)
      ELSE
C          ::    Cnoidal Wave Profile
C                FINDM is to find the parameter M of the Jacobian
C                elliptic func. See Func. 08 CEL and Subr. 09 SNCNDN
        CALL FINDM (MCNO)
        KC2    = 1.D+00-MCNO
        KC     = DSQRT(KC2)
        KCNO   = CEL(KC,1.D+00,1.D+00,1.D+00)
        ECNO   = CEL(KC,1.D+00,1.D+00,KC2)
        UR     = 16.D+00*MCNO*KCNO*KCNO/3.D+00
        WL     = DSQRT(UR*DSEAKS)
        ETAMIN = (1.D+00-ECNO/KCNO)/MCNO - 1.D+00
        ETAMIN = KS*ETAMIN
        ETAMAX = ETAMIN + KS
```

```
        NO      = 1
        DO 120 N = 1,NHALF1
          TIMEN = DBLE(N-1)*T
          TETA  = 2.D+00*KCNO*TIMEN
          CALL SNCNDN (TETA,KC2,SNU,CNU,DNU)
          ETAU(N) = ETAMIN + KS*CNU*CNU
          IF (N.GT.1) THEN
            IF (ETAU(N).LE.0.D+00.AND.ETAU(N-1).GT.0.D+00)NO=N
          ENDIF
  120   CONTINUE
        ETAU(NHALF1) = ETAMIN
C
      ENDIF
C           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 so that elevation=0 at time=0 and
C           decreases initially with time
C           ------------------------------------------------------------
C           ETAU = unadjusted surface elevation
C           ETA  = adjusted surface elevation
C           ------------------------------------------------------------
      NMARK = NONE-NO+2
      DO 140 N = 1,NONE1
        IF (N.LE.NMARK) THEN
          ETA(N) = ETAU(N+NO-1)
        ELSE
          ETA(N) = ETAU(N-NMARK+1)
        ENDIF
  140 CONTINUE
C
      RETURN
      END
C
C -06------------------- END OF SUBROUTINE REGWAV ------------BEFORR-
C #07###################### SUBROUTINE FINDM ################BEFORR#
C
C     This subroutine computes the parameter m (MLIL<m<MBIG) of the
C     Jacobian elliptic functions (m=MCNO in this program)
C
      SUBROUTINE FINDM (MCNO)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      DOUBLE PRECISION KCNO,MCNO,KC2,KC,MSAV,MLIL,MBIG
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CONSTA/ PI,GRAV
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
```

```
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),TSLOPS,WTOT
      CALL CKFPAR(7,1,N1,N1R)
      SMALL = 1.D-07
      MLIL  = .8D+00
      INDI  = 0
      I     = 0
      SIGDT = DSQRT(2.D+00*PI*WL0)
      MBIG  = 1.00D+00 - 1.00D-15
      MCNO  =  .95D+00
  900 CONTINUE
        I    = I+1
        MSAV = MCNO
        KC2  = 1.D+00-MCNO
        KC   = DSQRT(KC2)
        KCNO = CEL(KC,1.D+00,1.D+00,1.D+00)
        ECNO = CEL(KC,1.D+00,1.D+00,KC2)
        UR   = 16.D+00*MCNO*KCNO*KCNO/3.D+00
        WL   = DSQRT(UR*DSEAKS)
        FCNO = 1.D+00 + (-MCNO+2.D+00-3.D+00*ECNO/KCNO)/(MCNO*DSEAKS)
        FCNO = SIGDT*DSQRT(FCNO)/WL - 1.D+00
        IF (FCNO.LT.0.D+00) THEN
          MBIG = MCNO
        ELSEIF (FCNO.GT.0.D+00) THEN
          MLIL = MCNO
        ELSE
          RETURN
        ENDIF
        MCNO = (MLIL+MBIG)/2.D+00
        DIF  = DABS(MSAV-MCNO)
        IF (DIF.LT.SMALL) RETURN
        IF (INDI.EQ.0) THEN
          IF (I.EQ.50) THEN
            SMALL = 1.D-13
            INDI  = 1
          ELSE
            IF (MCNO.GT..9999D+00) THEN
              SMALL = 1.D-13
              INDI  = 1
            ENDIF
          ENDIF
        ENDIF
      IF (I.LT.100) GOTO 900
      WRITE  (*,9910)
      WRITE (99,9910)
 9910 FORMAT (/' From Subr. 07 FINDM:'/
     +         ' Criterion for parameter m=MCNO not satisfied')
C
      RETURN
      END
C
C -07------------------- END OF SUBROUTINE FINDM  ------------BEFORR-
C #08################ DOUBLE PRECISION FUNCTION CEL  #########BEFORR#
```

```
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 (99,*) '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 -08------------  END OF DOUBLE PRECISION FUNCTION CEL  ------BEFORR-
C #09######################  SUBROUTINE SNCNDN  ###############BEFORR#
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
      D   = 0.D+00
      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
          IF (ABS(D).LT.1.D-10) CALL STOPP (15,15)
          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 -09------------------ END OF SUBROUTINE SNCNDN ------------BEFORR-
C #10##################### SUBROUTINE CPARAM ##############BEFORR#
C
C     This subroutine
C     . calculates NONEM at the beginning of BEFORR2 computation
C     . assigns values to NONE and DELTA for each computation unit
C
      SUBROUTINE CPARAM (MODE,M,IPROB,U2STOP)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  MULTIF,NONEM,NONE,NEND,NRATE
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      COMMON /DIAGID/ INONE,ITRY,IREV,LEVEL
C
C           CALCULATE NONEM
C           ----------------------------------------------------------
C         . NONEM = minimum allowable value of NONE
```

```
C           . NONE  = number of time steps in one wave period
C           . NONE must:
C             + satisfy a numerical stability criterion
C             + be divisible by NRATE for storing purposes
C             + be an even number, for regular waves (IWAVE=1) only
C           . Below, NDUM3 is the smallest value which satisfies the
C             above criteria. BEFORR2 could start examining a computation
C             unit with NONE=NDUM3, but this manner is not always
C             efficient. BEFORR2 will instead begin its diagnosis with
C                        NONE = MULTIF*NDUM3 = NONEM,
C             where MULTIF is an integer, greater than or equal to unity,
C             specified by user. MULTIF may be increased from unity to
C             improve the temporal resolution of computed variations
C           . See description of INONE farther below
C             ---------------------------------------------------------
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (10,1,N1,N1R)
        EPSI  = DMAX1(X1,X2)
        DUM1  = 1.D+00 + EPSI*EPSI/4.D+00
        DUM2  = DSQRT(DUM1) - EPSI/2.D+00
        DUM3  = (1.D+00+DSQRT(DSEA))/(DUM2*X)
        NDUM1 = INT(DUM3) + 1
        NDUM2 = NDUM1/NRATE + 1
        NDUM3 = NDUM2*NRATE
        IF (IWAVE.EQ.1) THEN
          IDUM = MOD(NDUM3,2)
          IF (IDUM.NE.0) NDUM3=2*NDUM3
        ENDIF
C       :: Minimum allowable value of NONE is NONEM
        NONEM = MULTIF*NDUM3
C       :: Initial value NONE=NONEM serves a practical purpose
        NONE = NONEM
        T    = 1.D+00/DBLE(NONE)
        IF (IWAVE.GT.1) INONE=0
        RETURN
      ENDIF
C
C             ASSIGN VALUES TO NONE AND DELTA
C             NONE  = number of time steps in one wave period
C             DELTA = normalized water depth at computational waterline
C             ---------------------------------------------------------
C             Guidelines
C             ---------------------------------------------------------
C           . General:
C             1. At the very beginning:
C                . NONE=NONEM (see above)
C                . An initial value of DELTA is supplied by user
C             2. Increase NONE by about 50% whenever IPROB>0
C           . Regular waves (IWAVE=1):
C             1. The value of DELTA supplied by user is not modified
C             2. At the beginning of a computation unit, use the
C                previous value of NONE
```

```
C          . Random waves (IWAVE>1):
C            1. The value of DELTA may vary during computation
C            2. At the beginning of a computation unit use NONE=
C               (IREV+1)*NONEM and DELTA from previous computation unit
C            3. IPROB=2 may be fixed by decreasing DELTA, to a value
C               approximately 95% of the water depth at node (S-1),
C               without increasing NONE
C          . IREV = reverse indicator for random waves
C               (see main program)
C            -------------------------------------------------------
      IF (ITRY.EQ.1) THEN
        IF (M.EQ.1.OR.IWAVE.GT.1) NONE=(IREV+1)*NONEM
      ELSE
        CALL NSI (OMEGA,NONE)
        IF (OMEGA.GT.20) THEN
          WRITE  (*,9910) M,OMEGA,ITRY,IREV,NONE,DELTA
          WRITE (99,9910) M,OMEGA,ITRY,IREV,NONE,DELTA
          IPROB = 9
          INONE = 0
          IF (IWAVE.EQ.1) THEN
            WRITE  (*,9920)
            WRITE (99,9920)
            STOP
          ELSE
            WRITE  (*,9930)
            WRITE (99,9930)
            RETURN
          ENDIF
        ENDIF
        IF (IWAVE.GT.1.AND.IPROB.EQ.2.AND.U2STOP.GT.0.D+00)THEN
          DUM1  = .95*U2STOP
          DUM2  = DUM1/.000001D+00
          IDUM  = INT(DUM2)+1
          DELTA = DBLE(IDUM)*.000001D+00
        ELSE
          NDUM1 = NONE/(2*NRATE)
          NDUM2 = NDUM1*NRATE
          NONE  = NONE + NDUM2
        ENDIF
      ENDIF
 9910 FORMAT (/' From Subr. 10 CPARAM:'/
     +        ' Computation unit                   M =',I8/
     +        ' Numerical stability indicator > 20:'/
     +        '                                OMEGA =',F11.2/
     +        ' Number of failed trials         ITRY =',I8/
     +        ' Reverse indicator               IREV =',I8/
     +        ' Last used:                      NONE =',I8/
     +        '                                DELTA =',F18.9)
 9920 FORMAT (' OMEGA is too large. Computation aborted.'/
     +        ' Suggested to input larger DELTA.')
 9930 FORMAT (' Computation reversed to LEVEL=2 or LEVEL=3.')
C
```

```
C          INDICATOR FOR REGULAR WAVE COMPUTATION
C          --------------------------------------------------------
C        . INONE indicates whether or not regular incident wave
C          profile needs to be re-computed after this routine returns
C          to the main program because of the increase of NONE
C        . INONE=1: regular wave profile re-computed
C               =0: not re-computed
C        . For IWAVE>1, INONE=0 is set above when MODE=0
C        . For IWAVE=1, INONE=1 if NONE has been increased by this
C          routine; otherwise INONE=0
C          --------------------------------------------------------
      IF (IWAVE.EQ.1) THEN
        IF (ITRY.GT.1) THEN
          INONE = 1
        ELSE
          INONE = 0
        ENDIF
      ENDIF
C
C          DETERMINE NEND
C          NEND = number of time steps for this computation unit
C          NEND = NONE, except for the last unit MWAVE that may be
C                 less than one wave period
C          --------------------------------------------------------
      IF (M.LT.MWAVE) THEN
        NEND = NONE
      ELSE
        RES = DMOD(TMAX,1.D+00)
        IF (DABS(RES).LT.1.D-08) THEN
          NEND = NONE
        ELSE
          DEND = RES*DBLE(NONE)
          NEND = INT(DEND)
        ENDIF
      ENDIF
C
C          CALCULATE COMPUTATION PARAMETERS
C          which are dependent on NONE
C          --------------------------------------------------------
C          T = constant time step within a computation unit
C          X = constant grid spacing between two adjacent nodes
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          INITIALIZE IPROB
C          for each trial
C          --------------------------------------------------------
```

```
      IPROB = 0
C
      RETURN
      END
C
C -10------------------ END OF SUBROUTINE CPARAM ------------BEFORR-
C #11####################### SUBROUTINE NUMSTA ###############BEFORR#
C
C     This subroutine checks if numerical stability criterion is
C     violated (IPROB=3)
C
      SUBROUTINE NUMSTA (MODE,M,IPROB)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR2/  MULTIF,NONEM,NONE,NEND,NRATE
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /DIAGID/ INONE,ITRY,IREV,LEVEL
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (11,1,N1,N1R)
        RETURN
      ENDIF
C           ------------------------------------------------------------
C           T = time step; X = spatial grid size; XT = X/T
C           ------------------------------------------------------------
      ICOUNT = 0
      DO 100 J = 1,S
        IF (DABS(V(J)).GT.XT) ICOUNT=ICOUNT+1
  100 CONTINUE
C
      IF (ICOUNT.GT.0) THEN
        IPROB=3
        CALL NSI (OMEGA,NONE)
        WRITE  (*,9910) M,S,NONE,OMEGA,TIME,ICOUNT,ITRY,IREV
        WRITE (99,9910) M,S,NONE,OMEGA,TIME,ICOUNT,ITRY,IREV
      ENDIF
 9910 FORMAT (/' From Subr. 11 NUMSTA: IPROB=3'/
     +        ' (Stability criterion violated)'/
     +        ' Computation unit              M =',I8/
     +        ' Waterline node                S =',I8/
     +        ' Computation parameters:    NONE =',I8/
     +        '                           OMEGA =',F11.2/
     +        ' Time of occurrence         TIME =',F18.9/
     +        ' Number of occurrence     ICOUNT =',I8/
     +        ' Indicators            ITRY,IREV =',I4,',',I3)
C
      RETURN
      END
```

```
C
C -11------------------ END OF SUBROUTINE NUMSTA ------------BEFORR-
C #12###################### SUBROUTINE EXTRAP ###############BEFORR#
C
C     This subroutine estimates U(2,SP1) and V(SP1) with SP1=(S+1)
C     by extrapolation
C
C     S = most landward node at present 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
      SUBROUTINE EXTRAP (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (12,1,N1,N1R)
        RETURN
      ENDIF
      SM1 = S-1
      SP1 = S+1
      V(SP1)   = 2.D+00*V(S) - V(SM1)
      U(2,SP1) = 2.D+00*U(2,S) - U(2,SM1)
      U(1,SP1) = U(2,SP1)*V(SP1)
      IF (U(2,SP1).GT.0.D+00) THEN
        C(SP1) = DSQRT(U(2,SP1))
      ELSE
        C(SP1) = 0.D+00
      ENDIF
C
      RETURN
      END
C
C -12------------------ END OF SUBROUTINE EXTRAP ------------BEFORR-
C #13###################### SUBROUTINE RETAIN ###############BEFORR#
C
C     This subroutine retains some values at present time level at
C     landward and seaward boundaries
C
      SUBROUTINE RETAIN (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER S,SM1,SP1
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
```

```
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (13,1,N1,N1R)
        RETURN
      ENDIF
      VSN    = V(S)
      USN(1) = U(1,S)
      USN(2) = U(2,S)
      VMN    = V(SM1)
      UMN(1) = U(1,SM1)
      V1N    = V(1)
      V2N    = V(2)
C
      RETURN
      END
C
C -13------------------ END OF SUBROUTINE RETAIN  ------------BEFORR-
C #14######################## SUBROUTINE CRITV  ################BEFORR#
C
C     This subroutine calculates critical velocities used in
C     characteristic variables and checks for negative water depth
C     (IPROB=1)
C
      SUBROUTINE CRITV (MODE,M,IPROB)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR2/  MULTIF,NONEM,NONE,NEND,NRATE
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /DIAGID/ INONE,ITRY,IREV,LEVEL
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (14,1,N1,N1R)
        RETURN
      ENDIF
      ICOUNT = 0
      DO 100 J = 1,S
        IF (U(2,J).LT.0.D+00) THEN
          ICOUNT = ICOUNT + 1
        ELSE
          C(J) = DSQRT(U(2,J))
        ENDIF
  100 CONTINUE
      IF (ICOUNT.GT.0) THEN
        IPROB=1
        CALL NSI (OMEGA,NONE)
        WRITE  (*,9910) M,S,NONE,OMEGA,TIME,ICOUNT,ITRY,IREV
        WRITE (99,9910) M,S,NONE,OMEGA,TIME,ICOUNT,ITRY,IREV
```

```
      ENDIF
 9910 FORMAT (/' From Subr. 14 CRITV: IPROB=1'/
     +         ' (Negative water depth)'/
     +         ' Computation unit                   M =',I8/
     +         ' Waterline node                     S =',I8/
     +         ' Computation parameters:         NONE =',I8/
     +         '                                OMEGA =',F11.2/
     +         ' Time of occurrence              TIME =',F18.9/
     +         ' Number of occurrence          ICOUNT =',I8/
     +         ' Indicators                 ITRY,IREV =',I4,',',I3)
C
      RETURN
      END
C
C -14------------------- END OF SUBROUTINE CRITV ------------BEFORR-
C #15###################### SUBROUTINE MARCH ################BEFORR#
C
C     This subroutine marches the computation from present time level
C     (N-1) to next time level N excluding seaward and landward
C     boundaries, which are treated separately
C
      SUBROUTINE MARCH (MODE,M,IPROB,U2STOP)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR2/  MULTIF,NONEM,NONE,NEND,NRATE
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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 /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /DIAGID/ INONE,ITRY,IREV,LEVEL
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (15,1,N1,N1R)
        RETURN
      ENDIF
C
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 numer. damping term
C           JE  = landward edge node
C           JE1 = JE-1
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. 20 MATAFG computes non-constant elements of Matrices
C           A and G, and the elements of Matrix F
C         . Subr. 22 MATS computes the first element of Matrix S
C         . Subr. 21 MATGJR, 23 MATD, and 24 MATU compute the elements
C           of Matrices g, D, and U, respectively
C         . Subr. 24 MATU computes values of U at time level N using
C           the results obtained from the other four subroutines
C           ----------------------------------------------------------
      CALL MATAFG (MODE,1,JLAX1)
      CALL MATGJR (MODE,1,JLAX)
      CALL MATS   (MODE,2,JLAX)
      CALL MATD   (MODE,JDAM,JLAX)
      CALL MATU   (MODE,2,JLAX)
C
C           IPROB=2: WATER DEPTH AT (S-1) <or= DELTA
C           ----------------------------------------------------------
      IF (U(2,SM1).LE.DELTA) THEN
        IPROB=2
        U2STOP=U(2,SM1)
        CALL NSI (OMEGA,NONE)
        WRITE  (*,9910) M,S,NONE,OMEGA,DELTA,U2STOP,TIME,ITRY,IREV
        WRITE (99,9910) M,S,NONE,OMEGA,DELTA,U2STOP,TIME,ITRY,IREV
        RETURN
      ENDIF
 9910 FORMAT (/' From Subr. 15 MARCH: IPROB=2'/
     +        ' (Water depth at (S-1) <or= DELTA)'/
     +        ' Computation unit               M =',I8/
     +        ' Waterline node                 S =',I8/
     +        ' Computation parameters:     NONE =',I8/
     +        '                            OMEGA =',F11.2/
     +        '                            DELTA =',F18.9/
     +        ' Water depth at (S-1)  U(2,S-1) =',F18.9/
     +        ' Time of occurrence          TIME =',F18.9/
     +        ' Indicators             ITRY,IREV =',I4,',',I3)
C
C           COMPLETE THE COMPUTATION OF HYDRODYNAMIC QUANTITIES
C           ----------------------------------------------------------
C           Water depth h is taken to be not less than DELTA for
C           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 -15------------------- END OF SUBROUTINE MARCH ------------BEFORR-
C #16###################### SUBROUTINE LANDBC ###############BEFORR#
C
C     This subroutine manages the computation for
C     landward boundary conditions
C
      SUBROUTINE LANDBC (MODE,M)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER S,SP1,SM1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE3/  ETA(N2)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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,S,SM1,SP1,JMAX
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (16,1,N1,N1R)
        CALL CKFPAR (16,2,N2,N2R)
        RETURN
      ENDIF
C
C           MANAGE LANDWARD B.C.
C           ---------------------------------------------------------
C         . Subr. 17 RUNUP computes shoreline movement if
C           computational waterline is on the slope (IJOB<3)
C         . Subr. 18 OVERT computes hydrodynamic quantities at
C           landward edge node if overtopping occurs (IJOB=2)
C         . ALPHA = landward-advancing characteristics
C         . ETAT  = surface elevation associated with transmitted wave
C           at landward boundary (IJOB=3)
C         . DLAND = norm. water depth below SWL at landward boundary
C           (IJOB=3)
C         . S used in Subr. 17 RUNUP is of present time level
C           ---------------------------------------------------------
      IF (IJOB.EQ.1) THEN
        CALL RUNUP (MODE)
        IF (S.GT.JE1) THEN
          WRITE  (*,9910) TIME,S,JE
          WRITE (99,9910) TIME,S,JE
```

```
          STOP
        ENDIF
      ELSEIF (IJOB.EQ.2) THEN
        IF (S.LT.JE) THEN
          CALL RUNUP (MODE)
        ELSE
          CALL OVERT (MODE)
        ENDIF
      ELSE
        DUM     = TX*(VSN+C(S))*(VSN-VMN+2.D+00*(C(S)-C(SM1)))
        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
 9910 FORMAT (/' From Subr. 16 LANDBC:'/
     +   ' t =',F18.6,'; 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
C           ----------------------------------------------------------
      IF (IJOB.LT.3) THEN
        DO 100 L = S+1,JE
          U(1,L)  = 0.D+00
          U(2,L)  = 0.D+00
          V(L)    = 0.D+00
          ELEV(L) = ZB(L)
  100   CONTINUE
      ENDIF
C
      RETURN
      END
C
C -16------------------ END OF SUBROUTINE LANDBC ------------BEFORR-
C #17###################### SUBROUTINE RUNUP  ################BEFORR#
C
C     This subroutine computes waterline movement for IJOB=1 and if
C     no overtopping occurs for IJOB=2
C
      SUBROUTINE RUNUP (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      DIMENSION USN2(2),US1N1(2)
      INTEGER S,SM1,SP1,SNEW
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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 /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (17,1,N1,N1R)
        RETURN
      ENDIF
C
C             ADJUST VALUES AT S IF U(2,S)>U(2,S-1)
C             --------------------------------------------------
      IF (U(2,S).GE.U(2,SM1)) THEN
        V(S)     = 2.D+00*V(SM1) - V(S-2)
        U(2,S)   = 2.D+00*U(2,SM1) - U(2,S-2)
        IF (ABS(V(S)).GT.ABS(V(SM1))) V(S)=.9*V(SM1)
        IF (U(2,S).LT.0.D+00) U(2,S)=.5*U(2,SM1)
        IF (U(2,S).GT.U(2,SM1)) U(2,S)=.9*U(2,SM1)
        U(1,S)   = V(S)*U(2,S)
        ELEV(S) = U(2,S) - U2INIT(S)
      ENDIF
C
C             DETERMINE THE NEXT WATERLINE NODE
C             --------------------------------------------------
      IF (U(2,S).LE.DELTA) THEN
        SNEW = SM1
      ELSE
        V(SP1)    = 2.D+00*V(S) - V(SM1)
        U(2,SP1) = 2.D+00*U(2,S) - U(2,SM1)
        U(1,SP1) = V(SP1)*U(2,SP1)
        IF (U(2,SP1).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 (MODE,SM1,SP1)
          CALL MATGJR (MODE,SM1,S)
          CALL MATS    (MODE,S,S)
          DUM1 = TX*((F(1,SP1)-F(1,SM1))/2.D+00+X*G1(S))
          DUM2 = TTXX*(GJR(1,S)-GJR(1,SM1))
          DUM3 = TX*(F(2,SP1)-F(2,SM1))
          DUM4 = TTXX*(GJR(2,S)-GJR(2,SM1))
          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)
          DUM1   = (XT*(VSN2-VSN)+U(2,SP1)-U(2,SM1)+TWOX*THETA(S))/VS
          VS1N1 = V(SM1)-DUM1
          US1N1(1) = U(1,SM1) - 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   = SP1
                U(2,SP1) = US1N1(2)
                U(1,SP1) = US1N1(1)
                V(SP1)   = VS1N1
              ENDIF
            ELSE
              IF (U(2,SP1).LE.U(2,S)) THEN
                SNEW = SP1
              ELSE
                SNEW = S
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
      IF (SNEW.EQ.SP1) ELEV(SP1)=U(2,SP1)-U2INIT(SP1)
      S = SNEW
C
C     S at next time level has been found
C
      RETURN
      END
C
C -17------------------- END OF SUBROUTINE RUNUP ------------BEFORR-
C #18###################### SUBROUTINE OVERT ################BEFORR#
C
C     This subroutine computes quantities at landward-end node for
C     IJOB=2 if overtopping occurs, that is, S=JE and SM1=(S-1)=JE1
C
      SUBROUTINE OVERT (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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 /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /VALUEN/ VSN,USN(2),VMN,UMN(1),V1N,V2N
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (18,1,N1,N1R)
```

```
          RETURN
        ENDIF
        IF (VMN.GT.C(SM1)) THEN
          U(1,S) = USN(1) - TX*(F(1,S)-F(1,SM1)) - 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(SM1)
          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 = SM1
        ELSE
          ELEV(S) = U(2,S) - U2INIT(S)
        ENDIF
C
        RETURN
        END
C
C -18------------------- END OF SUBROUTINE OVERT  ------------BEFORR-
C #19####################### SUBROUTINE SEABC  ################BEFORR#
C
C       This subroutine treats seaward boundary conditions at node j=1
C
        SUBROUTINE SEABC (MODE,M,N)
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
        DOUBLE PRECISION KS,KSREF,KSSEA,KSI
        COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
        COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                  ISAVA,ISAVB,ISAVC
        COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
        COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
        COMMON /WAVE3/  ETA(N2)
        COMMON /WAVE6/  NDATA
        COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                  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 (MODE.EQ.0) THEN
          CALL CKFPAR (19,1,N1,N1R)
          CALL CKFPAR (19,2,N2,N2R)
          RETURN
        ENDIF
C
C             ESTIMATE ETAR
C             ----------------------------------------------------------
```

```
C         . BETA = seaward-advancing characteristics
C         . ETAR = surface elevation due to reflected wave at
C           seaward boundary
C         . A correction term included in ETAR if INONCT=1 to improve
C           prediction of wave set-down and setup on beach
C           ----------------------------------------------------------
      VC1 = -V1N+2.D+00*C(1)
      VC2 = -V2N+2.D+00*C(2)
      IF (IWAVE.LT.3) THEN
        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)
      ENDIF
C
C           VALUES AT NODE ONE
C           ----------------------------------------------------------
      IF (IWAVE.EQ.1) THEN
C       :: Regular wave.
C          Surface elevation ETA has been computed by Subr. 06 REGWAV.
        ETAI    = ETA(N+1)
        U(2,1)  = DSEA+ETAR+ETAI
        ELEV(1) = ETAI+ETAR
      ELSE
C       :: Incident or total wave train is specified by user.
C          The specified time series ETA has NDATA points from the
C          normalized time t=0 to t=TMAX.
C          Interpolate the specified time series to obtain ETAI
C          at t=TIME.
        DJJ  = DBLE(NDATA-1)*TIME/TMAX
        JJ   = INT(DJJ)
        ETA1 = ETA(JJ+1)
        ETA2 = ETA(JJ+2)
        DEL  = DJJ - DBLE(JJ)
        ETAI = ETA1 + DEL*(ETA2-ETA1)
        IF (IWAVE.EQ.2) THEN
          U(2,1)  = DSEA+ETAI+ETAR
          ELEV(1) = ETAI+ETAR
        ELSE
          U(2,1)  = DSEA+ETAI
          ELEV(1) = ETAI
        ENDIF
      ENDIF
      IF (IWAVE.LT.3) THEN
        V(1)   = 2.D+00*DSQRT(U(2,1))-BETA
        U(1,1) = U(2,1)*V(1)
      ENDIF
C
      IF (IWAVE.EQ.3) THEN
        C1 = DSQRT(U(2,1))
        DENOM = 1.D+00-TX*(VC2-VC1)
        IF (DABS(DENOM).LT.1.D-04) THEN
          WRITE(*,9910) M,TIME,DENOM
```

```
          WRITE(99,9910) M,TIME,DENOM
          STOP
        ENDIF
        V(1) = (2.D+00*C1-VC1 -TX*(VC2-VC1)*C1 -T*THETA(1))/DENOM
        IF (V(1).GE.C1) THEN
          WRITE(*,9920) M,TIME,V(1),C1
          WRITE(99,9920) M,TIME,V(1),C1
          STOP
        ENDIF
        U(1,1) = U(2,1)*V(1)
        BETA = 2.D+00*C1-V(1)
        ETAR = BETA*DSEA2/2.D+00 - DSEA
        IF (INONCT.EQ.1) ETAR=ETAR-KS*KS/(16.D+00*DSEA)
      ENDIF
 9910 FORMAT ('From Subr. 19 SEABC: IWAVE=3'/
     +        '(Denominator for V(1) is almost zero)'/
     +        'Computation unit          M = ',I8/
     +        'Time of occurrence    TIME = ',F18.9/
     +        'Value of denominator DENOM = ',F18.9)
C
 9920 FORMAT ('From Subr. 19 SEABC: Seaward Boundary'/
     +        '(Flow at x=0 is not subcritical)'/
     +        'Computation unit          M = ',I8/
     +        'Time of occurrence    TIME = ',F18.9/
     +        'Water velocity at x=0 V(1) = ',F18.9/
     +        'Phase velocity at x=0 C(1) = ',F18.9)
C
      RETURN
      END
C
C -19------------------- END OF SUBROUTINE SEABC  ------------BEFORR-
C #20###################### SUBROUTINE MATAFG  ###############BEFORR#
C
C     This subroutine computes, for each node,
C     . the elements of the the first row of Matrix A (2x2)
C                                        --> A1(1,j) and A1(2,j)
C     . the elements of Matrix F (2x1)      --> F(1,j) and F(2,j)
C     . the first element of Matrix G (2x1) --> G1(j)
C     j=node number
C
      SUBROUTINE MATAFG (MODE,JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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 (MODE.EQ.0) THEN
        CALL CKFPAR (20,1,N1,N1R)
```

```
        RETURN
      ENDIF
      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(J)*DABS(V(J))*V(J)
  100 CONTINUE
C
      RETURN
      END
C
C -20------------------ END OF SUBROUTINE MATAFG ------------BEFORR-
C #21####################### SUBROUTINE MATGJR ##############BEFORR#
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 (MODE,JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /MATRIX/ A1(2,N1),F(2,N1),G1(N1),GJR(2,N1),S1(N1),D(2,N1)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (21,1,N1,N1R)
        RETURN
      ENDIF
C
      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 -21------------------ END OF SUBROUTINE MATGJR ------------BEFORR-
C #22######################## SUBROUTINE MATS ################BEFORR#
C
C     This subroutine computes, for each node, the first element of
C     Matrix S (2x1) --> S1(j), j=node number
C
      SUBROUTINE MATS (MODE,JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
```

```
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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 (MODE.EQ.0) THEN
        CALL CKFPAR (22,1,N1,N1R)
        RETURN
      ENDIF
C
      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(J)*DABS(V(J))*V(J)
        DUM5 = 2.D+00*FW(J)*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 -22------------------- END OF SUBROUTINE MATS  -------------BEFORR-
C #23######################## SUBROUTINE MATD  ################BEFORR#
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 (MODE,JDAM,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      DIMENSION Q(2,2,N1),UU(2,N1)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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 (MODE.EQ.0) THEN
        CALL CKFPAR (23,1,N1,N1R)
        RETURN
      ENDIF
C
      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 -23------------------- END OF SUBROUTINE MATD  -------------BEFORR-
C #24####################### SUBROUTINE MATU  ################BEFORR#
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 (MODE,JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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 (MODE.EQ.0) THEN
        CALL CKFPAR (24,1,N1,N1R)
        RETURN
      ENDIF
C
      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 -24------------------- END OF SUBROUTINE MATU -------------BEFORR-
C #25####################### SUBROUTINE NSI #################BEFORR#
C
C     This subroutine calculates numerical stability indicator OMEGA
C     CHECKING N1=N1R IS OMITTED IN THIS SUBROUTINE
C
      SUBROUTINE NSI (OMEGA,NONE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      EPSI  = DMAX1(X1,X2)
      DUM1  = 1.D+00 + EPSI*EPSI/4.D+00
      DUM2  = DSQRT(DUM1) - EPSI/2.D+00
      OMEGA = DUM2*X*DBLE(NONE)/(1.D+00+DSQRT(DSEA))
      RETURN
      END
C
C -25-------------------- END OF SUBROUTINE NSI ------------BEFORR-
C #26####################### SUBROUTINE INIT ################BEFORR#
C
C     This subroutine assigns initial values
C
      SUBROUTINE INIT (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER S,SM1,SP1,SSAV
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,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 /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /DIAGV1/ SSAV(3),JMSAV(3),NONESV(N6),MFIVE
      COMMON /DIAGV2/ USAV(6,N1),VSAV(3,N1),ESAV(3,N1),DELSAV(3),
     +                DELTSV(N6)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (26,1,N1,N1R)
```

```
        CALL CKFPAR (26,6,N6,N6R)
        RETURN
      ENDIF
C
C           HYDRODYNAMIC VARIABLES
C           ----------------------------------------------------------
C         . S = node number closest to the instantaneous computational
C           waterline
C         . JMAX = the largest node number reached by computational
C           waterline
C         . Hydrodynamic quantities at node j:
C               U(1,j)  = volume flux
C               U(2,j)  = total water depth
C                          (not less than DELTA for IJOB=3)
C               V(j)    = depth-averaged velocity
C               ELEV(j) = surface elevation above SWL
C           ----------------------------------------------------------
      S    = INITS
      JMAX = INITS
      DO 110 J = 1,JE
        U(1,J) = 0.D+00
        IF (J.LE.INITS) 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           SAVING VARIABLES AT LEVEL=1 AND LEVEL=2
C           ----------------------------------------------------------
      DO 130 L = 1,2
        SSAV(L)   = S
        JMSAV(L)  = JMAX
        DELSAV(L) = DELTA
        LU1 = 2*L-1
        LU2 = 2*L
        DO 120 J = 1,JE
          USAV(LU1,J) = U(1,J)
          USAV(LU2,J) = U(2,J)
          VSAV(L,J)   = V(J)
          ESAV(L,J)   = ELEV(J)
  120   CONTINUE
  130 CONTINUE
C
      RETURN
      END
C
C -26------------------- END OF SUBROUTINE INIT  -------------BEFORR-
```

```
C #27######################## SUBROUTINE INITM ################BEFORR#
C
C     This subroutine sets initial hydrodynamic quantities at the be-
C     ginning of a computation unit using the saved values at LEVEL=1
C
      SUBROUTINE INITM (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER S,SM1,SP1,SSAV
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /DIAGV1/ SSAV(3),JMSAV(3),NONESV(N6),MFIVE
      COMMON /DIAGV2/ USAV(6,N1),VSAV(3,N1),ESAV(3,N1),DELSAV(3),
     +                DELTSV(N6)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (27,1,N1,N1R)
        CALL CKFPAR (27,6,N6,N6R)
        RETURN
      ENDIF
      S     = SSAV(1)
      JMAX  = JMSAV(1)
      DELTA = DELSAV(1)
      DO 100 J = 1,JE
        U(1,J)  = USAV(1,J)
        U(2,J)  = USAV(2,J)
        V(J)    = VSAV(1,J)
        ELEV(J) = ESAV(1,J)
  100 CONTINUE
      RETURN
      END
C
C -27------------------- END OF SUBROUTINE INITM ------------BEFORR-
C #28######################## SUBROUTINE INITMM ###############BEFORR#
C
C     This subroutine sets initial hydrodynamic quantities at the
C     beginning of one set of five computation units using the saved
C     values at LEVEL=2 or LEVEL=3
C
      SUBROUTINE INITMM (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER S,SM1,SP1,SSAV
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /DIAGID/ INONE,ITRY,IREV,LEVEL
```

```
      COMMON /DIAGV1/ SSAV(3),JMSAV(3),NONESV(N6),MFIVE
      COMMON /DIAGV2/ USAV(6,N1),VSAV(3,N1),ESAV(3,N1),DELSAV(3),
     +                DELTSV(N6)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (28,1,N1,N1R)
        CALL CKFPAR (28,6,N6,N6R)
        RETURN
      ENDIF
      S     = SSAV(LEVEL)
      JMAX  = JMSAV(LEVEL)
      DELTA = DELSAV(LEVEL)
      LU1 = 2*LEVEL-1
      LU2 = 2*LEVEL
      DO 110 J = 1,JE
        U(1,J)  = USAV(LU1,J)
        U(2,J)  = USAV(LU2,J)
        V(J)    = VSAV(LEVEL,J)
        ELEV(J) = ESAV(LEVEL,J)
  110 CONTINUE
      IF (LEVEL.EQ.3) THEN
C       ::          Rearrange saving variables
        DO 120 J = 1,JE
          USAV(3,J) = USAV(5,J)
          USAV(4,J) = USAV(6,J)
          VSAV(2,J) = VSAV(3,J)
          ESAV(2,J) = ESAV(3,J)
  120   CONTINUE
      ENDIF
      RETURN
      END
C
C -28------------------- END OF SUBROUTINE INITMM ------------BEFORR-
C #29###################### SUBROUTINE SAVEM ################BEFORR#
C
C     This subroutine saves hydrodynamic quantities at the end of
C     a computation unit (LEVEL=1)
C
      SUBROUTINE SAVEM (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER S,SM1,SP1,SSAV
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDRO/  U(2,N1),V(N1),ELEV(N1),C(N1),DUDT(N1)
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /DIAGV1/ SSAV(3),JMSAV(3),NONESV(N6),MFIVE
      COMMON /DIAGV2/ USAV(6,N1),VSAV(3,N1),ESAV(3,N1),DELSAV(3),
     +                DELTSV(N6)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (29,1,N1,N1R)
```

```
        CALL CKFPAR (29,6,N6,N6R)
        RETURN
      ENDIF
      SSAV(1)   = S
      JMSAV(1)  = JMAX
      DELSAV(1) = DELTA
      DO 100 J = 1,JE
        USAV(1,J) = U(1,J)
        USAV(2,J) = U(2,J)
        VSAV(1,J) = V(J)
        ESAV(1,J) = ELEV(J)
  100 CONTINUE
      RETURN
      END
C
C -29------------------- END OF SUBROUTINE SAVEM  ------------BEFORR-
C #30###################### SUBROUTINE SAVEMM  ###############BEFORR#
C
C     This subroutine saves hydrodynamic quantities at the end of one
C     set of five computation units (LEVEL=2 and LEVEL=3)
C
      SUBROUTINE SAVEMM (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER SSAV
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /DIAGV1/ SSAV(3),JMSAV(3),NONESV(N6),MFIVE
      COMMON /DIAGV2/ USAV(6,N1),VSAV(3,N1),ESAV(3,N1),DELSAV(3),
     +                DELTSV(N6)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (30,1,N1,N1R)
        CALL CKFPAR (30,6,N6,N6R)
        RETURN
      ENDIF
C     ::            Saving variables at LEVEL=3: beginning of the
C                   previous set of five computation units
      SSAV(3)   = SSAV(2)
      JMSAV(3)  = JMSAV(2)
      DELSAV(3) = DELSAV(2)
      DO 110 J = 1,JE
        USAV(5,J) = USAV(3,J)
        USAV(6,J) = USAV(4,J)
        VSAV(3,J) = VSAV(2,J)
        ESAV(3,J) = ESAV(2,J)
  110   CONTINUE
C     ::            Saving variables at LEVEL=2: beginning of the
C                   present set of five computation units
      SSAV(2)   = SSAV(1)
      JMSAV(2)  = JMSAV(1)
      DELSAV(2) = DELSAV(1)
```

```
      DO 120 J = 1,JE
        USAV(3,J) = USAV(1,J)
        USAV(4,J) = USAV(2,J)
        VSAV(2,J) = VSAV(1,J)
        ESAV(2,J) = ESAV(1,J)
  120 CONTINUE
      RETURN
      END
C
C -30------------------ END OF SUBROUTINE SAVEMM ------------BEFORR-
C #31####################### SUBROUTINE DOC1  ################BEFORR#
C
C     This subroutine documents essential information before
C     time-marching computation
C
      SUBROUTINE DOC1 (MODE,FINP2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      DOUBLE PRECISION KCNO,MCNO,KC2,KS,KSREF,KSSEA,KSI
      CHARACTER UL1*7,UL2*7,FINP2*10
      INTEGER S,SM1,SP1
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  MULTIF,NONEM,NONE,NEND,NRATE
      COMMON /CPAR3/  INITS,JE,JE1
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /WAVE4/  ETAMAX,ETAMIN
      COMMON /WAVE5/  KCNO,ECNO,MCNO,KC2
      COMMON /WAVE6/  NDATA
      COMMON /BOT1/   DSEAP,DLANDP,FWP(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                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 /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /RUNP2/  DELRP(N3),DELTAR(N3)
      COMMON /ARMOR/  C2,C3,CD,CL,CM,SG,TANPHI,AMIN,AMAX,DAP
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (31,1,N1,N1R)
        CALL CKFPAR (31,3,N3,N3R)
        CALL CKFPAR (31,4,N4,N4R)
        RETURN
      ENDIF
C
C           SYSTEM OF UNITS
C           -------------------------------------------------------
```

```
      IF (ISYST.EQ.1) THEN
        UL1 = ' meters'
        UL2 = '    [mm]'
      ELSE
        UL1 = ' feet   '
        UL2 = ' [inch]'
      ENDIF
C
C            WAVE CONDITION
C            ------------------------------------------------------------
      WRITE (98,9811)
      IF (IWAVE.EQ.1) THEN
        IF (URPRE.LT.26.D+00) THEN
          WRITE (98,9812)
        ELSE
          WRITE (98,9813) KC2,ECNO,KCNO
        ENDIF
      ELSEIF (IWAVE.EQ.2) THEN
        WRITE (98,9814) NDATA
      ELSE
        WRITE (98,9815) NDATA
      ENDIF
      WRITE (98,9816) ETAMAX,ETAMIN
      WRITE (98,9817) TP,HREFP,UL1,DSEAP,UL1,KSREF,KSSEA,KS
      WRITE (98,9818) DSEA,WL,SIGMA,UR,KSI
      IF (IJOB.EQ.3)  WRITE (98,9819) DLANDP,UL1
 9811 FORMAT (/'WAVE CONDITION')
 9812 FORMAT (/'Stokes II Incident Wave at Seaward Boundary')
 9813 FORMAT (/'Cnoidal Incident Wave at Seaward Boundary'/
     +          '1-m = ',D20.9/
     +          'E   = ',D20.9/
     +          'K   = ',D20.9)
 9814 FORMAT (/'Incident Wave at Seaward Boundary Given as Input'/
     +          'Number of Data Points     NDATA =',I8)
 9815 FORMAT (/'Total Wave at Seaward Boundary Given as Input'/
     +          'Number of Data Points     NDATA =',I8)
 9816 FORMAT ('Norm. Maximum Surface Elev.     = ',F14.6/
     +         'Norm. Minimum Surface Elev.     = ',F14.6/)
 9817 FORMAT ('Reference Wave Period           = ',F14.6,' sec.'/
     +         'Reference Wave Height           = ',F14.6,A7/
     +         'Depth at Seaward Boundary       = ',F14.6,A7/
     +         'Shoal. Coef. at Reference  Ks1 = ',F11.3/
     +         '             at Seaw. Bdr. Ks2 = ',F11.3/
     +         '             Ks = Ks2/Ks1      = ',F11.3)
 9818 FORMAT ('Norm. Depth at Seaw. Bdr.       = ',F11.3/
     +         'Normalized Wave Length          = ',F11.3/
     +         '"Sigma"                         = ',F11.3/
     +         'Ursell Number                   = ',F11.3/
     +         'Surf Similarity Parameter       = ',F11.3)
 9819 FORMAT ('Depth at Landward Boundary      = ',F14.6,A7)
C
C            STRUCTURE PROPERTIES
```

```
C          --------------------------------------------------------
      WRITE (98,9821) WTOT,NBSEG
      IF (IBOT.EQ.1) THEN
        WRITE (98,9822) UL1
        WRITE (98,9824) (K,WBSEG(K),TBSLOP(K),FWP(K),K=1,NBSEG)
      ELSE
        WRITE (98,9823) UL1,UL1
        WRITE (98,9824) (K,XBSEG(K),ZBSEG(K),FWP(K),K=1,NBSEG+1)
      ENDIF
      WRITE (98,9825)
 9821 FORMAT (/'SLOPE PROPERTIES'//
     +        'Norm. Horiz. Length of'/
     +        '       Computation Domain    = ',F15.6/
     +        'Number of Segments           = ',I8)
 9822 FORMAT (32(1H-)/' SEGMENT    WBSEG(I)   TBSLOP(I)    FWP(I)'/
     +               '        I        ',A7/32(1H-))
 9823 FORMAT (32(1H-)/' SEGMENT    XBSEG(I)    ZBSEG(I)    FWP(I)'/
     +               '        I        ',A7,'      ',A7/32(1H-))
 9824 FORMAT (I8,3F12.6)
 9825 FORMAT (32(1H-))
C
C           PARAMETERS FOR ARMOR STABILITY AND MOVEMENT
C          --------------------------------------------------------
      IF (ISTAB.GT.0) WRITE (98,9831) TANPHI,SG,C2,C3,CD,CL,CM
      IF (ISTAB.EQ.1) WRITE (98,9832) AMAX,AMIN
      IF (ISTAB.EQ.2) WRITE (98,9833) DAP,UL1
 9831 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)
 9832 FORMAT ('Norm. Upper and Lower Bounds of du/dt'/
     +        '                      AMAX  = ',F9.3/
     +        '                      AMIN  = ',F9.3)
 9833 FORMAT ('Armor Diameter              = ',F12.6,A7)
C
C           COMPUTATION PARAMETERS
C          --------------------------------------------------------
      WRITE (98,9841) JE
      IF (IJOB.LT.3)  WRITE (98,9842) INITS
      WRITE (98,9843) X,X1,X2,TMAX,NONEM,NRATE,MSTAT
      IF (ISTAB.GT.0) WRITE (98,9844) MSTAB
      IF (ISAVA.EQ.1) WRITE (98,9845) NTIMES,MSAVA1,MSAVA2
 9841 FORMAT (/'COMPUTATION PARAMETERS'//
     + 'Total Number of Spatial Nodes       JE = ',I8)
 9842 FORMAT ('Number of Nodes Along Bottom Below SWL'/
     + '                                 INITS = ',I8)
 9843 FORMAT (
     + 'Normalized Delta x         = ',D19.6/
```

```
     + 'Damping Coefficients     x1 = ',F15.3/
     + '                         x2 = ',F15.3/
     + 'Computation Duration  TMAX = ',F18.6,' Wave Periods'/
     + 'Minimum Allowable NONE            NONEM = ',I8/
     + '(NONE = number of time steps in one wave period)'/
     + 'Time Series Stored at Rate NRATE'/
     + '     per Wave Period with         NRATE = ',I8/
     + 'Statistical Calculations Are Performed Excluding'/
     + '     the First MSTAT Computation Units'/
     + '     with                         MSTAT = ',I8)
 9844 FORMAT (
     + 'Armor Stability Computations Are Performed Excluding'/
     + '     the First MSTAB Computation Units'/
     + '     with                         MSTAB = ',I8)
 9845 FORMAT (
     + 'Spatial Variations are Stored NTIMES at Equal'/
     + '     Intervals per Wave Period from'/
     + '     Computation Unit MSAVA1 to MSAVA2,'/
     + '     Inclusive, with             NTIMES = ',I8/
     + '                                 MSAVA1 = ',I8/
     + '                                 MSAVA2 = ',I8)
C
C          RUNUP METER
C          ----------------------------------------------------------
      IF (IJOB.LT.3) THEN
       WRITE (98,9851) UL2
       DO 110 L = 1,NDELR
         WRITE (98,9852) L,DELRP(L)
  110  CONTINUE
       WRITE (98,9853)
      ENDIF
 9851 FORMAT (/'RUNUP METER'//5x,17(1H-)/
     +         '        I   DELTAR(I)'/'           ',A7/5x,17(1H-))
 9852 FORMAT (I8,1X,F9.3)
 9853 FORMAT (5x,17(1H-))
C
C          COMPUTATION LOG
C          ----------------------------------------------------------
      WRITE (98,9861)
 9861 FORMAT (/'COMPUTATION LOG'//68(1H-)/
     + 1x,'Comp.   NONE    NEND    NEND/       DELTA',
     +    '    Num.      No.     Re-'/
     + 1x,'Unit',22x,'NONE',15x,'Stab.     of    verse'/
     + 45x,'Indic.  Trials  Indic.'/68(1H-))
C
C          NORMALIZED STRUCTURE GEOMETRY
C          ----------------------------------------------------------
C          File 52 = 'BSPACE'
C          (XB(j),ZB(j)) = normalized coordinates of the structure
C                          at node j
C          ZB negative below SWL
C          ----------------------------------------------------------
```

```
      WRITE (52,9000) JE
      WRITE (52,8000) (XB(J),ZB(J),J=1,JE)
 9000 FORMAT (I8)
 8000 FORMAT (5D15.6)
C
C           WRITE "CASE-SIGNATURES" IN FILE 'BINPUT'
C           ------------------------------------------------------
      WRITE (51,9000) MWAVE
      IF (IWAVE.GT.1) THEN
        WRITE (51,5000) FINP2
        WRITE (51,9000) NDATA
      ENDIF
      WRITE (51,9000) NONEM
 5000 FORMAT (A10)
C
C           CONDITIONAL STOP BEFORE TIME-MARCHING COMPUTATION
C           ------------------------------------------------------
c      WRITE (*,6010)
c      READ  (*,*) ISTOP
c      IF (ISTOP.EQ.1) STOP
c 6010 FORMAT (' Time-marching computation is about to begin'/
c     +         ' 1 = stop here, else = proceed')
C
      RETURN
      END
C
C -31------------------- END OF SUBROUTINE DOC1  -------------BEFORR-
C #32######################## SUBROUTINE DOC2  ################BEFORR#
C
C     This subroutine writes computation log during time-marching
C     computation
C
      SUBROUTINE DOC2 (MODE,M,MREP)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER SSAV
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /CPAR2/  MULTIF,NONEM,NONE,NEND,NRATE
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /DIAGID/ INONE,ITRY,IREV,LEVEL
      COMMON /DIAGV1/ SSAV(3),JMSAV(3),NONESV(N6),MFIVE
      COMMON /DIAGV2/ USAV(6,N1),VSAV(3,N1),ESAV(3,N1),DELSAV(3),
     +                DELTSV(N6)
      IF (MODE.EQ.0) THEN
        IF (MREP.LT.0) CALL STOPP (16,16)
        CALL CKFPAR (32,1,N1,N1R)
        CALL CKFPAR (32,6,N6,N6R)
        RETURN
      ENDIF
C
C           WRITE IN FILE 'BDOC'
```

```
C          ---------------------------------------------------------
      CALL NSI (OMEGA,NONE)
      IF (NONE.EQ.NEND) THEN
        DUR = 1.D+00
        WRITE (98,9810) M,NONE,NEND,DUR,DELTA,OMEGA,ITRY,IREV
      ELSE
        DUR = DBLE(NEND)/DBLE(NONE)
        WRITE (98,9820) M,NONE,NEND,DUR,DELTA,OMEGA,ITRY,IREV
      ENDIF
      NONESV(M) = NONE
      DELTSV(M) = DELTA
 9810 FORMAT (I5,2I8,F10.1,F12.8,F8.2,2I8)
 9820 FORMAT (I5,2I8,F10.6,F12.8,F8.2,2I8)
C
C          WRITE ON SCREEN every MREP waves
C          ---------------------------------------------------------
      IF (MREP.GT.0) THEN
        IDUM = MOD(M,MREP)
        IF (IDUM.EQ.0) WRITE (*,6010) M,NONE,DELTA,ITRY,IREV
      ENDIF
 6010 FORMAT (' M,NONE,DELTA,ITRY,IREV',I5,I8,F12.8,2I4)
C
      RETURN
      END
C
C -32------------------- END OF SUBROUTINE DOC2 -------------BEFORR-
C #33####################### SUBROUTINE DOC3 ################BEFORR#
C
C     This subroutine writes in file 'BINPUT' and completes
C     documentation in file 'BDOC'
C
      SUBROUTINE DOC3 (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      INTEGER S,SM1,SP1,SSAV
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      COMMON /CPAR1/  MWAVE,MSTAT,MSTAB,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  MULTIF,NONEM,NONE,NEND,NRATE
      COMMON /RUNP1/  NDELR,S,SM1,SP1,JMAX
      COMMON /DIAGV1/ SSAV(3),JMSAV(3),NONESV(N6),MFIVE
      COMMON /DIAGV2/ USAV(6,N1),VSAV(3,N1),ESAV(3,N1),DELSAV(3),
     +                DELTSV(N6)
      IF (MODE.EQ.0) THEN
        CALL CKFPAR (33,1,N1,N1R)
        CALL CKFPAR (33,6,N6,N6R)
        RETURN
      ENDIF
C
      IF (IWAVE.EQ.1) WRITE (51,9000) NONESV(MWAVE)
```

```
      DO 100 M = 1,MWAVE-1
        WRITE (51,5110) M,NONESV(M),NONESV(M),DELTSV(M)
  100 CONTINUE
      WRITE (51,5110) MWAVE,NONESV(MWAVE),NEND,DELTSV(MWAVE)
 5110 FORMAT (3I8,D18.9)
 9000 FORMAT (I8)
C
      IF (IJOB.LT.3) WRITE (98,9811) JMAX
 9811 FORMAT (68(1H-)//'RUNUP'//
     + 'Over the time span 0 < t <or= TMAX,'/
     + '   the largest node number reached by'/
     + '   the computational waterline is    =',I8/
     + 'Notes:'/
     + '1. This information may be useful to figure out the minimum'/
     + '   value of the dimension N1 that should be provided for'/
     + '   the subsequent RBREAK2 computation.'/
     + '2. The above number may not be identical to JMAX to be'/
     + '   computed by RBREAK2. JMAX is the largest node number'/
     + '   reached by the computational waterline over the'/
     + '   time span MSTAT < t <or= TMAX.')
C
      RETURN
      END
C
C -33-------------------- END OF SUBROUTINE DOC3 -------------BEFORR-
C #34######################## SUBROUTINE CKFPAR ###############BEFORR#
C
C     This subroutine checks if FORTRAN PARAMETER NCHEK=N1,N2,N3,N4,
C     N5,N6 specified in given Subroutine (ICALL) matches its
C     counterpart NREF=N1R,N2R,N3R,N4R,N5R,N6R from the main program
C
      SUBROUTINE CKFPAR (ICALL,NW,NCHEK,NREF)
C
      CHARACTER*2 WHICH(6)
      CHARACTER*6 SUBR(40)
      DATA WHICH /'N1','N2','N3','N4','N5','N6'/
      DATA SUBR  /'OPENIO','INPUT1','INPUT2','BOTTOM','WPARAM',
     2            'REGWAV','FINDM ','CEL   ','SNCNDN','CPARAM',
     3            'NUMSTA','EXTRAP','RETAIN','CRITV ','MARCH ',
     4            'LANDBC','RUNUP ','OVERT ','SEABC ','MATAFG',
     5            'MATGJR','MATS  ','MATD  ','MATU  ','NSI   ',
     6            'INIT  ','INITM ','INITMM','SAVEM ','SAVEMM',
     7            'DOC1  ','DOC2  ','DOC3  ','CKFPAR','CKOPTI',
     8            'CKVAL ','CKVAL1','CKVAL2','CKSUBR','STOPP '/
      IF (NCHEK.NE.NREF) THEN
        WRITE  (*,9910)
     +          WHICH(NW),NCHEK,ICALL,SUBR(ICALL),WHICH(NW),NREF
        WRITE (99,9910)
     +          WHICH(NW),NCHEK,ICALL,SUBR(ICALL),WHICH(NW),NREF
        STOP
      ENDIF
 9910 FORMAT (/
```

```
     +' PARAMETER Error: ',A2,' =',I8,' in Subroutine',I3,' ',A6/
     +' Correct Value:   ',A2,' =',I8)
      RETURN
      END
C
C -34------------------ END OF SUBROUTINE CKFPAR ------------BEFORR-
C #35###################### SUBROUTINE CKOPTI ###############BEFORR#
C
C     This subroutine checks user-chosen options
C
      SUBROUTINE CKOPTI (IC,INDIC,ITEM,ILOW,IUP)
C
      CHARACTER*6 OPTI(15)
      DATA OPTI /'IJOB  ','ISTAB ','ISYST ','IBOT  ','INONCT',
     1           'IENERG','IWAVE ','ISAVA ','ISAVB ','ISAVC ',
     2           'IREQ  ','IELEV ','IV    ','IDUDT ','ISNR  '/
      IF (ITEM.LT.ILOW.OR.ITEM.GT.IUP) THEN
        WRITE  (*,9910) OPTI(IC),ITEM,OPTI(IC),ILOW,IUP
        WRITE (99,9910) OPTI(IC),ITEM,OPTI(IC),ILOW,IUP
        INDIC = INDIC+1
      ENDIF
 9910 FORMAT (/' Input Error: ',A6,'=',I1/
     +         ' Specify ',A6,' in the range of [',I1,',',I1,']')
      RETURN
      END
C
C -35------------------ END OF SUBROUTINE CKOPTI ------------BEFORR-
C #36###################### SUBROUTINE CKVAL ################BEFORR#
C
C     This subroutine checks a number of user-specified values
C     Note: Checkings performed in this subroutine are separated from
C           Subr. 02 INPUT1 because some FORTRAN compilers may have
C           problems with subroutines having too many IF statements
C     CHECKING N1=N1R IS OMITTED IN THIS SUBROUTINE
C
      SUBROUTINE CKVAL (INDIC)
      PARAMETER (N1=800,N2=24000,N3=3,N4=40,N5=40,N6=2000)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION KS,KSREF,KSSEA,KSI
      COMMON /CPAR2/  MULTIF,NONEM,NONE,NEND,NRATE
      COMMON /CPAR4/  X1,X2,DELTA,TMAX,TIME,T,X,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  KS,KSREF,KSSEA,WLO,WL,UR,URPRE,KSI,SIGMA
      COMMON /BOT1/   DSEAP,DLANDP,FWP(N1)
      COMMON /BOT2/   DSEA,DSEAKS,DSEA2,DLAND,DLAND2,FW(N1),
     +                TSLOPS,WTOT
      IF (MULTIF.LT.1)       CALL CKVAL2 (1,INDIC,'MULTIF')
      IF (NRATE .LT.1)       CALL CKVAL2 (1,INDIC,'NRATE ')
      IF (DELTA .LE.0.D+00) CALL CKVAL2 (2,INDIC,'DELTA ')
      IF (TMAX  .LE.0.D+00) CALL CKVAL2 (2,INDIC,'TMAX  ')
      IF (X1    .LT.0.D+00) CALL CKVAL2 (3,INDIC,'X1    ')
      IF (X2    .LT.0.D+00) CALL CKVAL2 (3,INDIC,'X2    ')
```

```
      IF (TP     .LE.0.D+00) CALL CKVAL2 (2,INDIC,'TP    ')
      IF (KSREF .LE.0.D+00) CALL CKVAL2 (2,INDIC,'KSREF ')
      IF (KSSEA .LE.0.D+00) CALL CKVAL2 (2,INDIC,'KSSEA ')
      IF (DSEAP .LE.0.D+00) CALL CKVAL2 (2,INDIC,'DSEAP ')
      PRINT *,TSLOPS
      IF (TSLOPS.LE.0.D+00) CALL CKVAL2 (2,INDIC,'TSLOPS')
      RETURN
      END
C
C -36------------------ END OF SUBROUTINE CKVAL  ------------BEFORR-
C #37####################### SUBROUTINE CKVAL1  ###############BEFORR#
C
C     This subroutine checks user-specified values
C
      SUBROUTINE CKVAL1 (IC,ITEM,ILOW,IUP)
C
      CHARACTER*2 WHICH(7)
      CHARACTER*6 WHAT(12)
      DATA WHICH /'N5','N5','N5','N3','N1','N4','N2'/
      DATA WHAT  /'NREQ  ','NNODB ','NNODC ','NDELR ','INITS ',
     +            'NBSEG ','NDATA ','MSTAT ','MSAVA1','MSAVA2',
     +            'NODB  ','NODC  '/
      IF (IC.LE.7) THEN
        IF (ITEM.LT.ILOW.OR.ITEM.GT.IUP) THEN
          WRITE  (*,9910) WHAT(IC),ITEM,WHAT(IC),IUP,WHICH(IC)
          WRITE (99,9910) WHAT(IC),ITEM,WHAT(IC),IUP,WHICH(IC)
          STOP
        ENDIF
      ELSE
        IF (ITEM.LT.ILOW.OR.ITEM.GT.IUP) THEN
          WRITE  (*,9920) WHAT(IC),ITEM,WHAT(IC),ILOW,IUP
          WRITE (99,9920) WHAT(IC),ITEM,WHAT(IC),ILOW,IUP
          STOP
        ENDIF
      ENDIF
 9910 FORMAT (/' Input Error: ',A6,'=',I8/
     +        ' Specify ',A6,' in the range of [1,',I8,']'/
     +        ' Change PARAMETER ',A2,' if necessary')
 9920 FORMAT (/' Input Error: ',A6,'=',I8/
     +        ' Specify ',A6,' in the range of [',I8,',',I8,']')
      RETURN
      END
C
C -37------------------ END OF SUBROUTINE CKVAL1 ------------BEFORR-
C #38####################### SUBROUTINE CKVAL2  ###############BEFORR#
C
C     This subroutine checks user-specified values
C
      SUBROUTINE CKVAL2 (IMSG,INDIC,WHAT)
C
      CHARACTER WHAT*6,MSG(3)*24
      DATA MSG /' must be at least unity.',
```

```
     +          ' must be positive.      ',
     +          ' must be non-negative.  '/
      INDIC = INDIC+1
      WRITE  (*,9910) WHAT,MSG(IMSG)
      WRITE (99,9910) WHAT,MSG(IMSG)
 9910 FORMAT (/' Input Error: ',A6,A24)
      RETURN
      END
C
C -38------------------ END OF SUBROUTINE CKVAL2 ------------BEFORR-
C #39###################### SUBROUTINE CKSUBR ###############BEFORR#
C
C     This subroutine checks the values of the FORTRAN PARAMETERs
C     specified in subroutines
C
      SUBROUTINE CKSUBR (N1,N2,N3,N4,N5,N6)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*10 CDUM
      COMMON /DIMENS/ N1R,N2R,N3R,N4R,N5R,N6R
      COMMON /ID/     IJOB,ISTAB,ISYST,IBOT,INONCT,IWAVE,
     +                ISAVA,ISAVB,ISAVC
      DATA IDUM /0/
      DATA DDUM /0.D+00/
      DATA CDUM /'Tcharacter'/
C
C           DUMMIES FOR FORTRAN PARAMETERS USED IN THIS PROGRAM
C           ---------------------------------------------------------
C           Variables specified in PARAMETER statement cannot be
C           passed through COMMON statement. The following dummy
C           integers are used in COMMON /DIMENS/.
C           ---------------------------------------------------------
      N1R = N1
      N2R = N2
      N3R = N3
      N4R = N4
      N5R = N5
      N6R = N6
C
C           CHECK ALL SUBROUTINES POSSESING FORTRAN "PARAMETER" STATEMEN
C           Checking is indicated by MODE=0
C           ---------------------------------------------------------
      MODE=0
      CALL INPUT1 (MODE,CDUM)
      CALL BOTTOM (MODE)
      CALL WPARAM (MODE)
      CALL NUMSTA (MODE,IDUM,IDUM)
      CALL RETAIN (MODE)
      CALL CRITV  (MODE,IDUM,IDUM)
      CALL MARCH  (MODE,IDUM,IDUM,DDUM)
      CALL LANDBC (MODE,IDUM)
      CALL SEABC  (MODE,IDUM,IDUM)
```

```
      CALL MATAFG (MODE,IDUM,IDUM)
      CALL MATGJR (MODE,IDUM,IDUM)
      CALL MATS   (MODE,IDUM,IDUM)
      CALL MATD   (MODE,IDUM,IDUM)
      CALL MATU   (MODE,IDUM,IDUM)
      CALL INIT   (MODE)
      CALL INITM  (MODE)
      CALL INITMM (MODE)
      CALL SAVEM  (MODE)
      CALL SAVEMM (MODE)
      CALL DOC1   (MODE,CDUM)
      CALL DOC2   (MODE,IDUM,IDUM)
      CALL DOC3   (MODE)
      IF (IJOB.LT.3) THEN
        CALL EXTRAP (MODE)
        CALL RUNUP  (MODE)
        IF (IJOB.EQ.2) CALL OVERT (MODE)
      ENDIF
      IF (IWAVE.EQ.1) THEN
        CALL REGWAV (MODE)
      ELSE
        CALL INPUT2 (MODE)
      ENDIF
C
      RETURN
      END
C
C -39------------------ END OF SUBROUTINE CKSUBR -----------BEFORR-
C #40####################### SUBROUTINE STOPP ################BEFORR#
C
C     This subroutine executes a programmed stop
C
      SUBROUTINE STOPP (IBEGIN,IEND)
C
      CHARACTER*55 MSG(16)
      DATA MSG /
     1 ' Computation aborted after IREV reached 5.',
     2 ' Suggested to input larger DELTA.',
     3 ' There is at least one error in primary input data.',
     4 ' See file BMSG to find out the error(s).',
     5 ' Special storing requested,',
     6 ' but pertinent identifiers not specified correctly.',
     7 ' Check identifiers IREQ,IELEV,IV,IDUDT,ISNR.',
     8 ' TMAX must be a whole number for IWAVE=1.',
     9 ' TREQ can not exceed TMAX.',
     + ' HREFP must be positive.',
     1 ' SWL is always above the structure.',
     2 ' RUNUP/OVERTOPPING computation can not be performed.',
     3 ' Part of the structure is above SWL.',
     4 ' TRANSMISSION computation can not be performed.',
     5 ' Failure in Subr. 09 SNCNDN.',
     6 ' MREP must be positive or zero.'/
```

```
      DO 100 I = IBEGIN,IEND
        WRITE  (*,9910) MSG(I)
        WRITE (99,9910) MSG(I)
  100 CONTINUE
      WRITE  (*,9920)
      WRITE (99,9920)
 9910 FORMAT (/A55)
 9920 FORMAT (' Programmed Stop.')
C
      STOP
      END
C
C -40------------------- END OF SUBROUTINE STOPP ------------BEFORR-
```

Appendix D

Contents of the Accompanying Disk

The 3.5 inch floppy disk accompanying this report contains the following computer programs.

- Four subroutines explained in Part VI and listed in Appendix A.
- Computer program RBREAK2 explained in Part III and listed in Appendix B.
- Computer program BEFORR2 explained in Part IV and listed in Appendix C.
- File `prn3t` corresponding to the primary input data file for run `RN3-T` listed in Table 11.
- File `rn3wg1` containing the measured and normalized total wave time series at wave gage 1 for run `RN3-T` where `FINP2 = rn3wg1` in Table 11.