COMPUTER PROGRAM CSHORE FOR PREDICTING CROSS-SHORE TRANSFORMATION OF IRREGULAR BREAKING WAVES

By

Nobuhisa Kobayashi and Bradley D. Johnson

Sponsored by
U.S. Army Corps of Engineers
Waterways Experiment Station
Coastal and Hydraulics Laboratory

RESEARCH REPORT NO. CACR-98-04

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

ABSTRACT

The prediction and prevention of the deterioration of a rubble mound breakwater requires an accurate numerical model for predicting the incident wave conditions at the toe of the breakwater because the damage to the breakwater is very sensitive to the incident wave height. Such a model needs to account for irregular wave breaking because most breakwaters in the U.S. are exposed to depth-limited breaking waves during storms. The model must also be efficient computationally because the prediction of the cumulative damage due to a series of storms requires numerical simulation of a long duration. A numerical model satisfying these requirements has been developed in this project to predict the incident wave conditions required for the damage progression model developed by Melby and Kobayashi (1998).

The numerical model called CSHORE in this report is based on the time-averaged, cross-shore continuity, momentum and energy equations including nonlinear effects such as undertow, skewness and kurtosis. The hydrodynamic model is coupled with a nonlinear probabilistic model that describes the probability distribution of the free surface elevation. Empirical formulas are proposed for the skewness and kurtosis as well as the ratio of the root-mean-square wave height to the mean water depth which increases rapidly near the still waterline. The developed model is shown to be capable of predicting the measured cross-shore variations of the mean, standard deviation, skewness and kurtosis of the free surface elevation from outside the surf zone to the lower swash zone for three irregular wave tests on a 1:16 smooth impermeable slope and two tests of quasi-equilibrium terraced and barred beaches. The developed model will be expanded further to predict irregular wave runup and overtopping on undamaged and damaged rubble mound breakwaters.

ACKNOWLEDGMENT

This is the report resulting from research sponsored by the U.S. Army Corps of Engineers, Waterways Experiment Station, Coastal and Hydraulics Laboratory under contract No. DACW 39-98-K-0021. The authors would like to thank J. A. Melby for his enthusiastic support of our efforts to develop a relatively simple but versatile model for the design of coastal structures.

Contents

1 INTRODUCTION

1.1 Background

The maintenance and repair of coastal structures such as breakwaters and jetties is an important element in the nation's rehabilitation of deteriorating infrastructure. Most of these structures in the U.S. are constructed of locally available stone and exposed to depth-limited breaking waves during storms. The conventional rubble mound structure has been designed typically for no or limited damage to its armor layer during the peak of a design storm. However, a new design procedure for aging and deteriorating infrastructure is required to estimate the maintenance cost and repair frequency of the armor layer during its service life. Such a design procedure will be probabilistic to account for sequences of future storms and the variability of damage along the structure.

The stability of the armor layer of a rubble mound against wind-generated waves has been designed using hydraulic model tests and empirical formulas. Hudson (1959) conducted regular wave experiments and proposed an empirical formula which is now called the Hudson formula. To apply the Hudson formula to irregular waves, the Shore Protection Manual (1984) proposed the use of $H_{1/10}$ as a representative wave height of irregular waves, where $H_{1/10}$ is the average height of the highest 1/10 of waves. For practical applications, it is more straightforward to develop empirical formulas directly from irregular wave experiments (Van der Meer 1988). Most previous regular and irregular wave experiments were conducted on undamaged breakwaters in relatively deep water to reduce the number of parameters involved in the resulting empirical formulas. Furthermore, the existing formulas for breakwater armor stability are limited to constant incident wave conditions and water level.

As a first step to develop a probabilistic method for the maintenance and repair of rubble mound breakwaters, Melby and Kobayashi (1998) conducted one 28.5-hr test and two shorter tests, which were repeated twice, in a wave flume. Damage progression and variability were measured on a conventional rubble mound exposed to depth-limited breaking waves in sequences of storms with varying irregular wave conditions and water levels. Measurements were made of 16 or 32 damaged profiles every 30 min of irregular wave action. Each measured profile was characterized by the eroded area, depth and length of the armor layer and the remaining cover depth. The mean and standard deviation of these statistical variables changed with damage progression, whereas the probability distributions of the normalized variables were practically invariant. The mean and standard deviation of the damage variables were shown to be represented empirically by the mean eroded area alone. The damage variability along the structure was significant even in the flume experiment because of the irregularity of placed stone along the structure. This variability of damage along the structure was taken into account in estimating the occurrence of the localized failure of the armor layer. Finally, an empirical formula was developed to predict the mean eroded area in sequences of storms. This formula was shown to be in good agreement with the three tests where the incident waves measured in front of the structure were used as input to the formula. The incident waves were represented by the significant wave height H_s and the mean wave period T_m based on the zero-upcrossing method as well as by the spectral significant wave height H_{mo}

and the spectral peak period T_p.

To predict damage progression and variability on a conventional rubble mound in sequence of storms using the empirical formulas developed by Melby and Kobayashi (1998), it is required to specify as input the time series of the pair of H_s and T_m or the pair of H_{mo} and T_p in front of the structure. Such wave data do not exist for typical applications because available wave data are normally given in relatively deep water such as 20-m depth. It is hence necessary to compute irregular wave transformation from the location of offshore wave data to the nearshore site of the structure. Various numerical models are now available to compute the nearshore wave transformation as reviewed by Kobayashi (1998). Such a numerical model should also predict irregular wave runup and overtopping on the structure that determine the required height of the structure.

Time-dependent numerical models such as those based on the finite-amplitude shallow-water equations have been shown to be capable of predicting wave runup and overtopping on coastal structures (Kobayashi et al. 1987; Kobayashi 1995) as well as surf and swash dynamics on beaches (Kobayashi et al. 1989; Kobayashi and Wurjanto 1992; Raubenheimer et al. 1995; Raubenheimer and Guza 1996). However, time-dependent numerical models require considerable computation time to resolve the wave profiles varying in time and space. Moreover, a probabilistic model for the maintenance and repair of rubble mound breakwaters will require a simpler model for irregular wave transformation that can be coupled more easily with the empirical formulas for damage progression and variability proposed by Melby and Kobayashi (1998). The time-averaged models for random waves represented by the root-mean-square wave height (Battjes and Janssen 1978; Thornton and Guza 1983) or expressed as the superposition of regular waves [e.g., Dally (1992)] are mush more efficient computationally but may considerable underpredict the wave setup and root-mean-square wave height near the still waterline (Cox et al. 1994).

A new time-averaged model called CSHORE is developed here to predict the cross-shore variations of the mean, standard deviation, skewness and kurtosis of the free surface elevation from outside the surf zone to the swash zone on beaches and coastal structures. The spectral significant wave height H_{mo} is defined here as $H_{mo} = 4\sigma$ with σ = standard deviation of the free surface elevation. This new model includes nonlinear correction terms in the cross-shore radiation stress and energy flux that become important in very shallow water. Empirical formulas are proposed for the skewness and kurtosis as well as the ratio of the root-mean-square wave height, $H_{rms} = \sqrt{8}\,\sigma$, to the mean water depth which increases rapidly near the still waterline. The developed model is shown to be in agreement with three irregular wave tests on a 1:16 smooth impermeable slope and two tests of quasi-equilibrium terraced and barred beaches. The new model can predict the observed large increase of wave setup above the still waterline. This time-averaged model may hence be applied to predict beach erosion and recovery near the shoreline [e.g., Kriebel (1990); Hedegaard et al. (1992); Nairn and Southgate (1993)]. The model will be compared with experiments in which measurements are being made of irregular breaking wave transformation and runup on a stone revetment fronted by a gentle slope.

1.2 Outline of Report

This report consists of two parts. Section 2 explains the numerical model CSHORE, new empirical formulas, comparisons with five tests, and conclusions without reference to the computer program. This part essentially corresponds to the paper written by Kobayashi and Johnson (1998). Section 3 presents the details of the computer program CSHORE listed in Appendix. The main program, subroutines, common statements, input and output are explained sufficiently to alow modifications of the computer program without difficulties.

All users of CSHORE should read Section 2 and understand the limitations and capabilities of CSHORE. To use CSHORE in its present form, it may be sufficient to read only the input and output in Section 3. On the other hand, CSHORE may be expanded for various applications in future by modifying the main program and adding subroutines. Section 3 provides detailed explanations for this purpose.

2 NUMERICAL MODEL CSHORE

A nonlinear time-averaged model is developed here to predict the cross-shore variations of the wave setup, $\overline{\eta}$, and the root-mean-square wave height, H_{rms}, from outside the surf zone to the swash zone where H_{rms} is defined as $H_{rms} = \sqrt{8}\,\sigma$ with σ = standard deviation of the free surface elevation. The spectral significant wave height H_{mo} is given by $H_{mo} = \sqrt{2}\,H_{rms}$. This model is based on the time-averaged continuity, momentum and energy equations derived by time-averaging the nonlinear equations used in the time-dependent model of Kobayashi and Wurjanto (1992). The time-averaged equations can be solved numerically with much less computation time but require empirical relationships to close the problem. The time-averaged rate of energy dissipation due to random wave breaking is estimated by modifying the empirical formula of Battjes and Stive (1985) to account for the landward increase of $H_{rms}/\overline{h}$ near the waterline where $\overline{h}$ = mean water depth. The skewness s and the kurtosis K of the free surface elevation included in the time-averaged momentum and energy equations are expressed empirically as a function of $H_{rms}/\overline{h}$.

The developed model is compared with three tests conducted on a 1:16 smooth impermeable slope and two tests on quasi-equilibrium terraced and barred beaches consisting of fine sand. This new time-averaged model is shown to be capable of predicting the cross-shore variations of $\overline{\eta}$, H_{rms}, s and K of the free surface elevation from outside the surf zone to the lower swash zone of frequent wave uprush and downrush. The model of Battjes and Stive (1985) considerably underpredicts $\overline{\eta}$ and H_{rms} near the still waterline. The new model will need to be verified using additional experiments because the empirical formulas adopted in the model are developed using the same five tests.

2.1 New Time-Averaged Model

The assumptions of alongshore uniformity and normally incident irregular waves are made in the following. To account for nonlinear effects in very shallow water, use is made of the time-averaged equations derived from the finite-amplitude shallow-water equations including bottom friction. Assuming that the bottom is impermeable, the time-averaged continuity equation with the overbar denoting time-averaging is expressed as

$$\overline{hU} = 0 \tag{1}$$

where h = instantaneous water depth; and U = instantaneous depth-averaged horizontal velocity. The time-averaged cross-shore momentum equation is written as (Kobayashi et al. 1989)

$$\frac{dS_{xx}}{dx} = -\rho g \overline{h} \ \frac{d\overline{\eta}}{dx} - \overline{\tau}_b \tag{2}$$

with

$$S_{xx} = \rho \left[\overline{hU^2} + \frac{1}{2} g \overline{(\eta - \overline{\eta})^2} \right] \tag{3}$$

$$\overline{\tau}_b = \frac{1}{2} \rho \, f_b \, \overline{|U|U} \tag{4}$$

in which x = cross-shore coordinate taken to be positive landward; S_{xx} = cross-shore radiation stress; ρ = fluid density; g = gravitational acceleration; η = instantaneous free surface elevation above the still water level (SWL); $\overline{\tau}_b$ = time-averaged bottom shear stress; f_b = bottom friction factor. The bottom elevation z_b given by $z_b = (\eta - h)$ is assumed to depend on x only. The time-averaged energy equation corresponding to (1) and (2) may be expressed as (Kobayashi and Wurjanto 1992)

$$\frac{d}{dx} \left(\overline{E_F} \right) = -\overline{D_f} - \overline{D_B} \tag{5}$$

with

$$\overline{E_F} = \frac{1}{2} \rho \, \overline{hU^3} + \rho g \overline{\eta h U} \tag{6}$$

$$\overline{D_f} = \frac{1}{2} \rho \, f_b \overline{|U|U^2} \tag{7}$$

in which $\overline{E_F}$ = energy flux per unit width; $\overline{D_f}$ = energy dissipation rate due to bottom friction; and $\overline{D_B}$ = energy dissipation rate due to wave breaking which needs to be estimated empirically in this time-averaged model.

To simplify (1), (2) and (5), the instantaneous free surface elevation η is expressed as

$$\eta = \overline{\eta} + \sigma \eta_* \tag{8}$$

where $\overline{\eta}$ and σ = mean and standard deviation of η; and η_* = normalized free surface elevation with $\overline{\eta_*} = 0$ and $\overline{\eta_*^2} = 1$. If wave reflection is negligible, linear long wave theory may be used locally to relate the oscillatory components $(\eta - \overline{\eta})$ and $(U - \overline{U})$ inside and outside the surf zone (Guza and Thornton 1980; Kobayashi et al. 1998). This relationship together with (8) yields

$$U = \overline{U} + \sqrt{\frac{g}{\overline{h}}}\,\sigma\eta_* \tag{9}$$

Eq. (9) is necessary to reduce the number of unknown variables in the time-averaged model although the local reflection coefficient may not be negligible. Substitution of (8) and (9) into (1) with $h = (\eta - z_b)$ and $\overline{h} = (\overline{\eta} - z_b)$ yields

$$\overline{U} = -\sigma_*^2\sqrt{g\overline{h}} \quad ; \quad \sigma_* = \frac{\sigma}{\overline{h}} \tag{10}$$

which indicates that $\overline{U}$ is negative and represents return current (Kobayashi et al. 1989). Although (10) does not account for the landward mass flux due to a surface roller (Svendsen 1984), it predicted the undertow measured at the mid-depth below SWL fairly accurately (Kobayashi et al. 1997, 1998).

Substitution of (8) and (9) with (10) into (3) and (4) yields

$$S_{xx} = \frac{1}{8}\,\rho\,g\,H_{rms}^2\left[\left(2n - \frac{1}{2}\right) + C_s\right] \quad ; \quad H_{rms} = \sqrt{8}\,\sigma \tag{11}$$

$$\overline{\tau}_b = \frac{1}{2}\,\rho\,f_b\,\sigma_*^2\,g\overline{h}\,G_b \tag{12}$$

with

$$C_s = \sigma_* s - \sigma_*^2 \tag{13}$$

$$G_b = \int_{-\infty}^{\infty} |\eta_* - \sigma_*|(\eta_* - \sigma_*) f(\eta_*) d\eta_* \tag{14}$$

where s = skewness of η and η_* with $\overline{\eta_*^3} = s$; n = finite-depth adjustment parameter added herein with $n = 1$ in shallow water; C_s = nonlinear correction term for S_{xx}; and G_b = dimensionless parameter related to $\overline{\tau}_b$ with $f(\eta_*)$ = probability density function of η_*. The assumption of equivalency of the time and probabilistic averaging is made to obtain (14) as explained by Kobayashi et al. (1998). For linear progressive waves in finite depth, n is normally expressed as [e.g., Battjes and Stive (1985)]

$$n = \frac{1}{2}\left[1 + \frac{2k_p\overline{h}}{\sinh\left(2k_p\overline{h}\right)}\right] \tag{15}$$

where k_p = linear wave number corresponding to the spectral peak period T_p outside the surf zone. The cross-shore variation of T_p may be neglected in (15) because $n = 1$ in shallow water for any reasonable representative wave period used to calculate k_p. The cross-shore

radiation stress S_{xx} based on linear wave theory is given by (11) with $C_s = 0$ [e.g., Battjes and Stive (1985)]. It will be shown that C_s is on the order of unity near the still waterline and can not be neglected in the swash zone.

Substitution of (8) and (9) with (10) into (6) and (7) yields

$$\overline{E}_F = \frac{1}{8} \rho\, g\, H_{rms}^2\, n\, C_p (1 + C_F) \tag{16}$$

$$\overline{D}_f = \frac{1}{2} \rho\, f_b\, \sigma_*^3 \left(g\overline{h}\right)^{1.5} G_f \tag{17}$$

with

$$C_F = \frac{3}{2} s\, \sigma_* \left(1 - \sigma_*^2\right) + \frac{1}{2} \sigma_*^2 (K - 5) + \sigma_*^4 \tag{18}$$

$$G_f = \int_{-\infty}^{\infty} \mid \eta_* - \sigma_* \mid (\eta_* - \sigma_*)^2 f(\eta_*)\, d\eta_* \tag{19}$$

where C_p = phase velocity based on T_p with $C_p = \sqrt{g\overline{h}}$ in shallow water; C_F = nonlinear correction term for $\overline{E}_F$; K = kurtosis of η and η_* with $\overline{\eta_*^4} = K$; and G_f = dimensionless parameter related to $\overline{D}_f$. The finite-depth adjustment is included in (16) in the same way as (11) where $n\, C_p$ in (16) is the group velocity based on T_p. The cross-shore energy flux $\overline{E}_F$ based on linear wave theory is given by (16) with $C_F = 0$ [e.g., Battjes and Stive (1985)] where C_F will be shown later to be on the order of unity near the still waterline.

The momentum equation (2) with (11) and (12) and the energy equation (5) with (16) and (17) need to be solved numerically to predict the cross-shore variations of the wave setup $\overline{\eta} = (\overline{h} + z_b)$ and the root-mean-square wave height $H_{rms} = \sqrt{8}\, \sigma$. These equations reduce to those used in the existing time-averaged models [e.g., Battjes and Stive (1985)] if $C_s = 0$, $C_F = 0$, $\overline{\tau}_b = 0$ and $\overline{D}_f = 0$. To estimate the nonlinear correction terms C_s and C_F using (13) and (18) with $\sigma_* = \sigma/\overline{h}$, the skewness s and the kurtosis K are assumed to be expressed in the following empirical forms

$$s = f_s\left(H_{rms}/\overline{h}\right) \quad ; \quad K = f_K(s) \tag{20}$$

where f_s and f_K = empirical functions which will be obtained using the five tests discussed later.

To estimate the parameters G_b and G_f using (14) and (19), the probability density function $f(\eta_*)$ with $\overline{\eta}_* = 0$ and $\overline{\eta_*^2} = 1$ is assumed to be given by the exponential gamma distribution which was shown by Kobayashi et al. (1997, 1998) to be capable of describing the measured probability distributions of η_* from outside the surf zone to the lower swash zone. The exponential gamma distribution for η_* depends on the skewness s only and is limited to the range $0 \leq s \leq 2$. This distribution reduces to the normal distribution for $s = 0$ and the exponential distribution for $s = 2$. The analytical expressions of G_b and G_f can be derived from (14) and (19) for the normal and exponential distributions. For the

actual computation of G_b and G_f in Section 3.2.8, the normal distribution is assumed for $s \leq 0.15$, whereas the exponential distribution with $\sigma_* = 1$ is adopted for $s \geq 1.99$ where $\sigma_* = 1$ for the lower limit of η_* in the exponential distribution imposed by the bottom elevation as explained by Kobayashi et al. (1998). For $0.15 < s < 1.99$, (14) and (19) are integrated numerically to obtain G_b and G_f for given s and σ_*. $G_b = 0$ and $G_f = 1.60$ for $s = 0$ and $\sigma_* = 0$, whereas $G_b = -1.46$ and $G_f = 3.62$ for $s = 2$ and $\sigma_* = 1$.

Finally, the energy dissipation rate $\overline{D}_B$ due to wave breaking in the energy equation (5) needs to be estimated. The empirical formula proposed by Battjes and Janssen (1978) and calibrated by Battjes and Stive (1985) is adopted here for its simplicity. The formula proposed by Thornton and Guza (1983) and improved by Lippmann et al. (1996) including the effect of a surface roller may predict the distributions of breaking and nonbreaking wave heights more accurately but requires additional empirical parameters. In the present formulation, the exponential gamma function may be used to describe the probability density function of η instead of wave heights after the cross-shore variations of $\overline{\eta}$, σ and s are predicted (Kobayashi et al. 1997, 1998).

The calibrated formula by Battjes and Stive (1985) is given by

$$\overline{D_B} = \frac{\alpha}{4} \rho\, g\, f_p\, Q\, H_m^2 \tag{21}$$

with

$$\frac{Q-1}{\ln Q} = \left(\frac{H_{rms}}{H_m}\right)^2 \tag{22}$$

$$H_m = \frac{0.88}{k_p} \tanh\left(\frac{\gamma\, k_p\, \overline{h}}{0.88}\right) \tag{23}$$

$$\gamma = 0.5 + 0.4 \tanh\left(33\, \frac{H_{rmso}}{L_o}\right) \quad ; \quad L_o = \frac{g\, T_p^2}{2\pi} \tag{24}$$

where α = empirical coefficient recommended as $\alpha = 1$; f_p = spectral peak frequency given by $f_p = T_p^{-1}$; Q = local fraction of breaking waves in the range $0 \leq Q \leq 1$; H_m = local depth-limited wave height; k_p = linear wave number calculated using f_p and $\overline{h}$; γ = empirical parameter determining $H_m = \gamma \overline{h}$ in shallow water; L_o = deep-water wavelength based on T_p; and H_{rmso} = deep-water value of H_{rms} calculated using linear wave shoaling theory with T_p, $\overline{h}$ and H_{rms} specified at the seaward boundary of the numerical model.

The empirical parameter γ is uncertain in light of the field data by Raubenheimer et al. (1996) but is estimated using (24) without any additional calibration. Relatedly, Battjes and Janssen (1978) indicated that $\overline{D}_B$ given by (21) would underestimate the actual energy dissipation rate and produce $H_{rms} > H_m$ in very shallow water, although (22) with $Q \leq 1$ requires $H_{rms} \leq H_m$. They recommended use of $H_{rms} = H_m$ when $H_{rms} > H_m$. This adjustment leads to $H_{rms} = \gamma \overline{h}$ in very shallow water. However, $H_{rms}/\overline{h}$ is not constant and increases landward where $H_{rms}/\overline{h} \simeq 2$ at the still waterline for the SUPERTANK data of Kriebel (1994). This landward increase of $H_{rms}/\overline{h}$ may be related to the landward increase

of the local reflection coefficient (Baquerizo et al. 1997) but wave reflection is not accounted for explicitly in this time-averaged model. As a result, (21) with (22)–(24) is assumed to be valid only in the outer zone $x < x_i$ with x_i = cross-shore location where Q computed by (22) becomes unity and the still water depth decreases landward in the region $x > x_i$. The latter condition is required for a barred beach to allow $Q < 1$ landward of the bar crest where $Q = 1$ may occur. For the inner zone $x > x_i$, the ratio $H_* = H_{rms}/\overline{h}$ is assumed to be expressed as

$$H_* = \gamma + (\gamma_s - \gamma)\, x_*^\beta \quad ; \quad x_* = \frac{x - x_i}{x_s - x_i} > 0 \tag{25}$$

where γ_s = value of H_* on the order of two at the still waterline located at $x = x_s$; and β = empirical parameter. The values of γ_s and β will be calibrated using the five tests discussed later. Eq. (25) describes the landward increase of H_* from $H_* = \gamma$ at $x = x_i$ to $H_* = \gamma_s$ at $x = x_s > x_i$. For the inner zone $x > x_i$, the momentum equation (2) and (25) are used to predict the cross-shore variations of $\overline{h}$ and H_{rms}, whereas the energy equation (5) is used to estimate $\overline{D_B}$ which must be positive or zero.

The numerical model called CSHORE is developed to solve (2) and (5) with (11)–(25). The seaward boundary of CSHORE is located at $x = 0$ where the values of T_p, H_{rms} and $\overline{\eta}$ at $x = 0$ are specified as input. The bottom elevation $z_b(x)$ in the region $x \geq 0$ is also specified as input and the location x_s of the still water shoreline is found using $z_b(x = x_s) = 0$. First-order finite-difference approximations of (2) and (5) are expressed as

$$\overline{\eta}_{j+1} = \overline{\eta}_j - \left[\rho\, g \left(\overline{h}_{j+1} + \overline{h}_j\right)\right]^{-1} \left\{2\left[(S_{xx})_{j+1} - (S_{xx})_j\right] + \Delta x \left[(\overline{\tau}_b)_{j+1} + (\overline{\tau}_b)_j\right]\right\} \tag{26}$$

$$\left(\overline{E}_F\right)_{j+1} = \left(\overline{E}_F\right)_j - \frac{\Delta x}{2}\left[\left(\overline{D}_f\right)_{j+1} + \left(\overline{D}_f\right)_j + \left(\overline{D}_B\right)_{j+1} + \left(\overline{D}_B\right)_j\right] \tag{27}$$

where the subscripts $(j+1)$ and j indicate the quantities at nodes located at x_{j+1} and x_j, respectively, with $\Delta x = (x_{j+1} - x_j)$ being the nodal spacing. In the subsequent computations for the laboratory data, use is made of $\Delta x \simeq 10$ cm. For the known quantities at node j, the unknown quantities at node $(j+1)$ are computed by solving (26) and (27) using an iteration method starting from σ_{j+1}^2 computed using (27) with $(\overline{D}_f)_{j+1} = (\overline{D}_f)_j$ and $(\overline{D}_B)_{j+1} = (\overline{D}_B)_j$. The adopted iteration method is found to converge within several iterations. The convergency is based on the differences between the iterated values of σ_{j+1} and $\overline{h}_{j+1}$ being less than the specified small value ϵ, where $\epsilon = 0.01$ mm is used in the subsequent computations. If $Q_{j+1} = 1$ and $(dz_b/dx) > 0$ for $x \geq x_{j+1}$, the inner zone is reached and $x_i = x_{j+1}$ is set.

For the nodes located in the inner zone $x > x_i$, (25) is used to obtain $H_* = H_{rms}/\overline{h}$ and $\sigma_* = H_*/\sqrt{8}$. Since the mean water depth $\overline{h}$ can become very small in the inner zone, (2) with (11) and (12) is rewritten as

$$(2P+1)\frac{d\overline{h}}{dx} = -\overline{h}\,\frac{dP}{dx} - \frac{dz_b}{dx} - \frac{1}{2} f_b G_b \sigma_*^2 \quad \text{for} \quad x > x_i \tag{28}$$

with

$$P = \sigma_*^2\left[\left(2n - \frac{1}{2}\right) + \sigma_* s - \sigma_*^2\right] \tag{29}$$

A first-order finite difference approximation of (28) between nodes j and $(j+1)$ yields

$$\overline{h}_{j+1} = (3P_{j+1}+P_j+2)^{-1}\left\{(P_{j+1}+3P_j+2)\,\overline{h}_j - 2\left[(z_b)_{j+1}-(z_b)_j\right]\right.$$
$$\left.-\frac{\Delta x}{2}\left[\left(f_b\,G_b\,\sigma_*^2\right)_{j+1}+\left(f_b\,G_b\,\sigma_*^2\right)_j\right]\right\} \tag{30}$$

Eq. (30) is solved using an iteration method starting from the value of n_{j+1} involved in P_{j+1} calculated using $\overline{h}_j$ where $(\sigma_*)_{j+1}$, s_{j+1} and $(G_b)_{j+1}$ are known using (25), (20) and (14), respectively. Since n given by (15) is essentially unity in shallow water, this interaction method converges rapidly. After $\overline{h}_{j+1}$ is computed, the energy equation (5) is used to obtain $(\overline{D}_B)_{j+1}$. The computation is marched landward until $\overline{h}_{j+1} < \epsilon$.

2.2 Experiments and Empirical Formulas

Two different experiments were conducted in a wave tank that was 30 m long, 2.4 m wide, and 1.5 m high. These experiments were explained in detail by Kobayashi et al. (1997, 1998). Irregular waves based on the TMA spectrum (Bouws et al. 1985) were generated with a piston-type wave paddle. A rock beach of a 1:8 slope was located at the other end of the wave tank to absorb waves. A divider was constructed along the center line in the tank to conduct five tests in the 1.2-m-wide flume.

Three tests were conducted with a plywood beach of a 1:16 slope. The water depth in the tank was 76.2 cm. For each test, 17 runs were performed to measure free surface elevations using eight capacitance wave gages. Wave gages partially immersed in gage wells were used for the free surface measurements near the still waterline. On the other hand, two tests were conducted with a fine sand beach whose initial slope was 1:12. The sand was well-sorted and its median diameter was 0.18 mm. These two tests with specified random waves were conducted after the sand beach was exposed to the specified wave action for several days and became quasi-equilibrium with the bottom elevation changes less than about 1 cm/hr. For each of the two tests, 21 runs were performed to measure free surface elevations using ten wave gages. Wave gages near the still waterline were partially buried in the sand. The duration of each run in these five tests was 400 s and the initial transient duration of 75 s was removed. The sampling rate was 20 Hz.

Table 1 lists the wave conditions at the seaward boundary located at $x = 0$ for each of the five tests where d = still water depth; $\overline{\eta}$ = wave setup or set-down; T_p = spectral peak period; and H_{rms} = root-mean-square wave height defined as $H_{rms} = \sqrt{8}\,\sigma$ with σ = standard deviation of the measured free surface oscillation. Tests 1, 2 and 3 are the 1:16 slope tests described by Kobayashi et al. (1998), whereas tests 4 and 5 correspond to the sand beach tests explained by Kobayashi et al. (1997). The wave setup or set-down is very small at $x = 0$ outside the surf zone. The measured wave conditions at $x = 0$ include the slight effects of reflected waves. The incident and reflected waves at $x = 0$ were estimated using a

Table 1: Wave Conditions at Seaward Boundary and Breaker Parameter γ for Five Tests

Test	d (cm)	$\overline{\eta}$ (cm)	T_p (s)	H_{rms} (cm)	H_{inc} (cm)	R	γ	x_i (m)	x_s (m)
1	75.0	0.03	1.5	12.4	12.2	0.14	0.84	11.1	12.0
2	75.0	-0.32	2.8	16.9	15.8	0.15	0.67	9.0	12.0
3	76.2	-0.24	4.7	18.4	18.4	0.17	0.56	8.3	13.0
4	60.0	-0.15	1.6	12.8	12.9	0.19	0.83	13.3	13.8
5	60.0	-0.12	2.8	14.6	14.3	0.25	0.65	12.4	13.7

three-gage method by Kobayashi et al. (1997, 1998). Table 1 lists the estimated values of the spectral root-mean-square wave height, $H_{inc} = \sqrt{8m_{oi}}$, with m_{oi} = zero-moment of the incident wave spectrum at $x = 0$, and the average reflection coefficient, $R = \sqrt{m_{or}/m_{oi}}$, with m_{or} = zero-moment of the reflected wave spectrum at $x = 0$. The difference between H_{rms} and H_{inc} is negligible except for test 2 with $(H_{rms} - H_{inc})/H_{rms} = 0.065$. The reflection coefficient was in the narrow range $0.14 \leq R \leq 0.25$ and slightly larger for tests 4 and 5 with the foreshore slope of about 1:5 at the still waterline.

The measured values of $\overline{\eta}$, T_p and H_{rms} at $x = 0$ listed in Table 1 are specified as input to CSHORE. The measured bottom elevation $z_b(x)$ in the region $x \geq 0$ is also specified as input where Table 1 lists the cross-shore location x_s of the still waterline for each test. The bottom profile $z_b(x)$ will be presented in conjunction with the measured and predicted cross-shore variations of $\overline{\eta}$ and H_{rms}. The breaker parameter γ calculated using (24) and the cross-shore location x_i at the landward limit of the outer zone computed by CSHORE are listed in Table 1. These computed values of x_i are for the bottom friction factor $f_b = 0$ in (12) and (17) which is the value of f_b used in the subsequent comparisons of CSHORE with the five tests as explained later.

The measured values of $H_* = H_{rms}/\overline{h}$ in the inner zone $x > x_i$ are used to calibrate the new empirical parameters γ_s and β in (25) for the five tests. Fig. 1 shows the measured values of $(H_* - \gamma)/(\gamma_s - \gamma)$ with $\gamma_s = 2$ as a function of $x_* = (x - x_i)/(x_s - x_i)$ where the values of γ, x_i and x_s for each test are listed in Table 1. The trend of the scattered data points for the five tests may be represented by (25) with $\gamma_s = 2$ and $\beta = 2.2$. Fig. 1 shows that H_* increases gradually from $H_* = \gamma$ at $x_* = 0$ and more rapidly above the still waterline located at $x_* = 1$. It is noted that the large scatter in the region $x_* > 1$ is caused partly by the scatter of data points obtained in repeated runs due to the difficulty in measuring $\overline{h}$ and

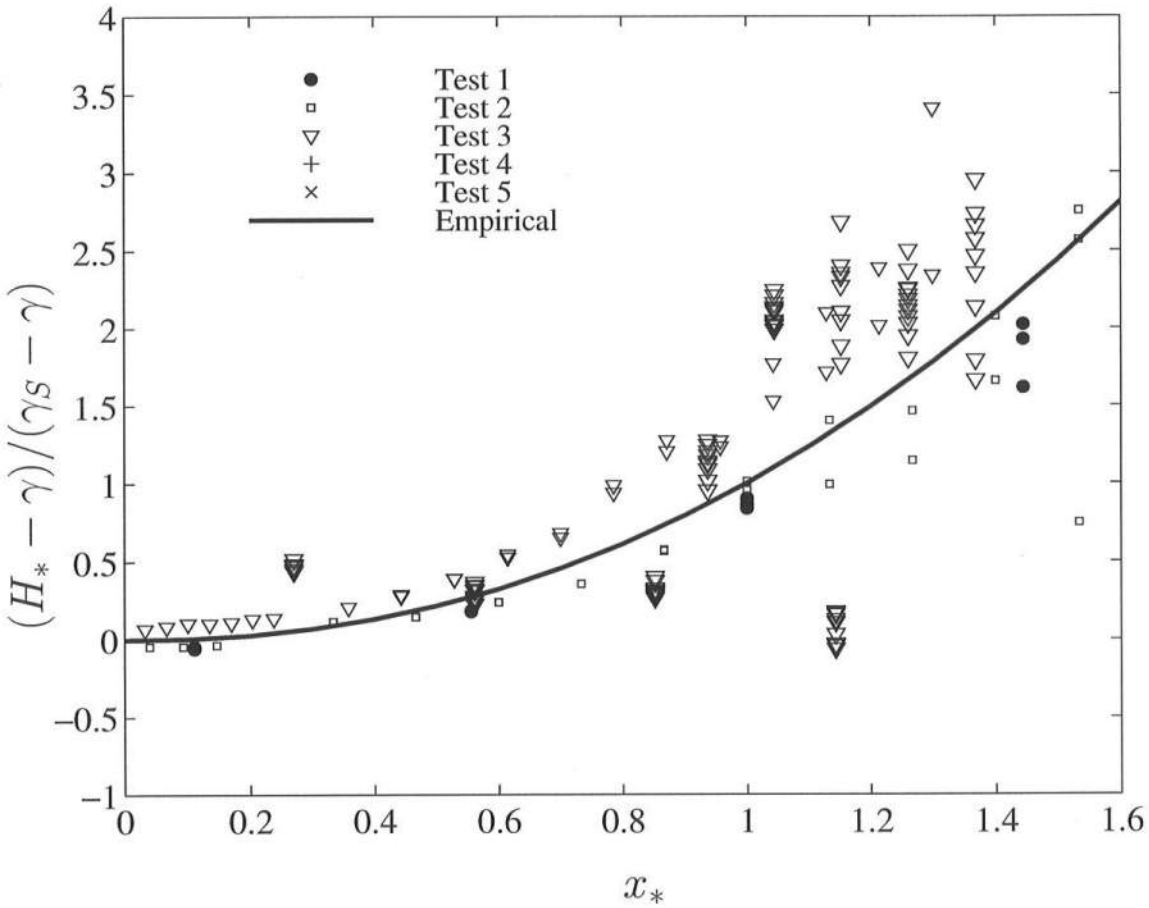

Figure 1: Empirical Formula for $H_* = H_{rms}/\bar{h}$ in Inner Zone $x > x_i$.

H_{rms} accurately in the swash zone.

The measured values of H_*, s and K in the entire region $x \geq 0$ for the five tests are analyzed to obtain the empirical relationships expressed by (20). Fig. 2 shows the skewness s as a function of $H_* = H_{rms}/\overline{h}$. The trend of the scattered data points in Fig. 2 are simply represented by three straight lines

$$s = 2H_* \qquad \text{for} \quad 0.1 < H_* \leq 0.5 \tag{31a}$$

$$s = 1.5 - H_* \qquad \text{for} \quad 0.5 \leq H_* \leq 1.0 \tag{31b}$$

$$s = 0.7H_* - 0.2 \qquad \text{for} \quad 1.0 \leq H_* \lesssim 5 \tag{31c}$$

The skewness s increases initially with the increase of H_* due to wave shoaling but decreases after wave breaking. Both s and H_* increase rapidly near and above the still waterline. On the other hand, Fig. 3 shows the relationship between the kurtosis K and the skewness s which may be expressed as

$$K = 3 + s^{2.2} \qquad \text{for} \quad 0.2 < s \lesssim 3 \tag{32}$$

The empirical relationship between K and s proposed by Ochi and Wang (1984) yields similar agreement as shown in Fig. 3. However, their expression is more complicated and (32) is adopted here for its simplicity.

2.3 Comparisons with Five Tests

The numerical model CSHORE is compared with the five tests listed in Table 1 and used to develop the empirical formulas (31) and (32) as well as (25) with $\gamma_s = 2$ and $\beta = 2.2$. The additional input parameter required for CSHORE is the bottom friction factor f_b used in (12) and (17). The bottom friction for the time-dependent model corresponding to CSHORE is normally important for stone revetments and breakwaters (Kobayashi et al. 1987) but only in the swash zone on beaches (Kobayashi and Wurjanto 1992). Raubenheimer et al. (1995) and Raubenheimer and Guza (1996) used $f_b = 0.015$ to obtain good agreement between the measured and predicted wave runup on natural beaches. They also showed that the computed runup was not sensitive to f_b in the range of $f_b = 0.01$–0.02.

The assessment of the bottom friction effect in CSHORE is not straightforward because the empirical formula (25) for $H_* = H_{rms}/\overline{h}$ used in the inner zone includes the bottom friction effect implicitly. The energy equation (5) used to estimate $\overline{D}_B$ in the inner zone includes this implicit effect in addition to the explicit energy dissipation rate $\overline{D}_f$ due to bottom friction. As a result, the wave setup $\overline{\eta}$ computed using the momentum equation (2) is used to assess the sensitivity to f_b. Fig. 4 shows the computed cross-shore variations of $\overline{\eta}$ for $f_b = 0$, 0.01 and 0.02 for test 3 with the largest wave setup above the still waterline among the five tests listed in Table 1. The straight line in Fig. 4 corresponds to the 1:16 slope. The data was limited to the region of the mean water depth $\overline{h} \gtrsim 0.4$ cm (Kobayashi et al. 1998). Wave setup becomes tangential to the slope as explained by Bowen et al. (1968) for regular waves. The computed wave setup increases slightly with the increase of f_b. This

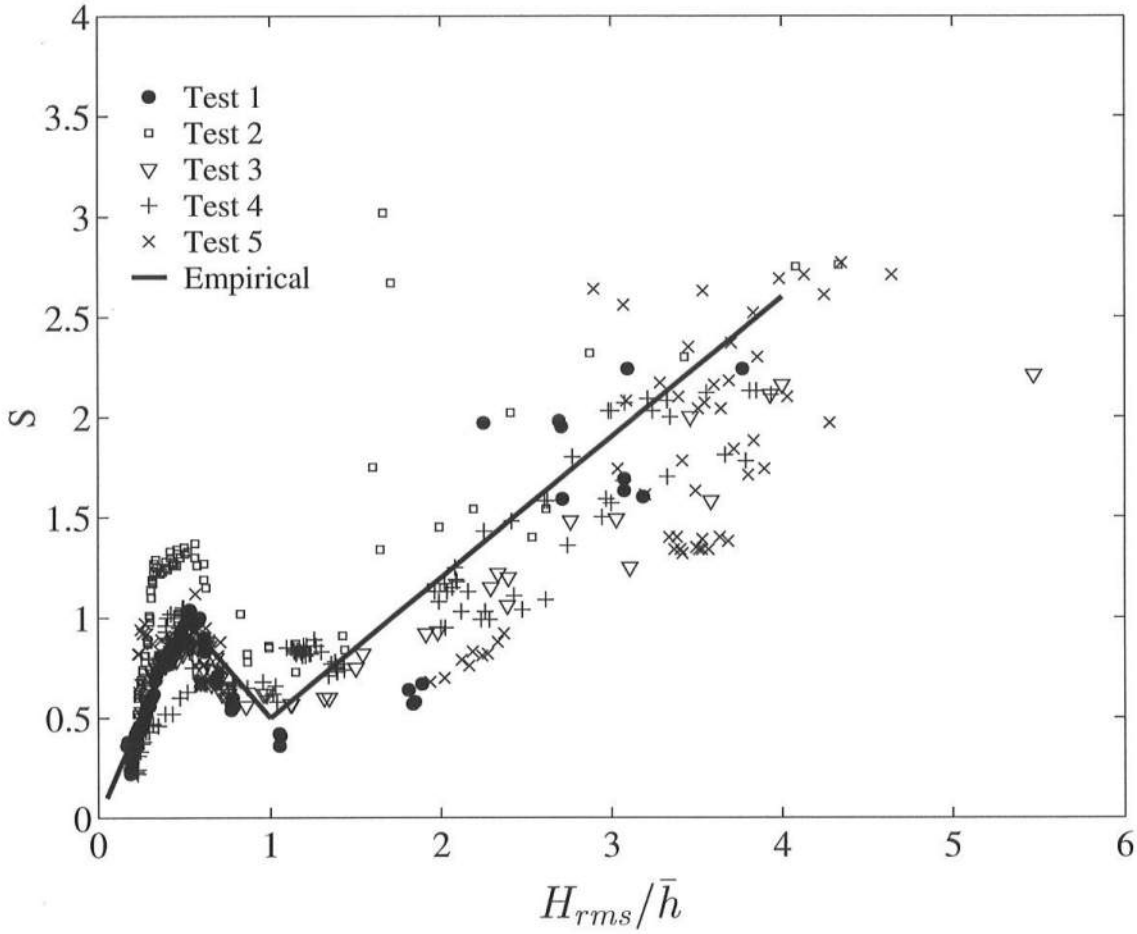

Figure 2: Empirical Formula for Skewness s as a Function of H_*.

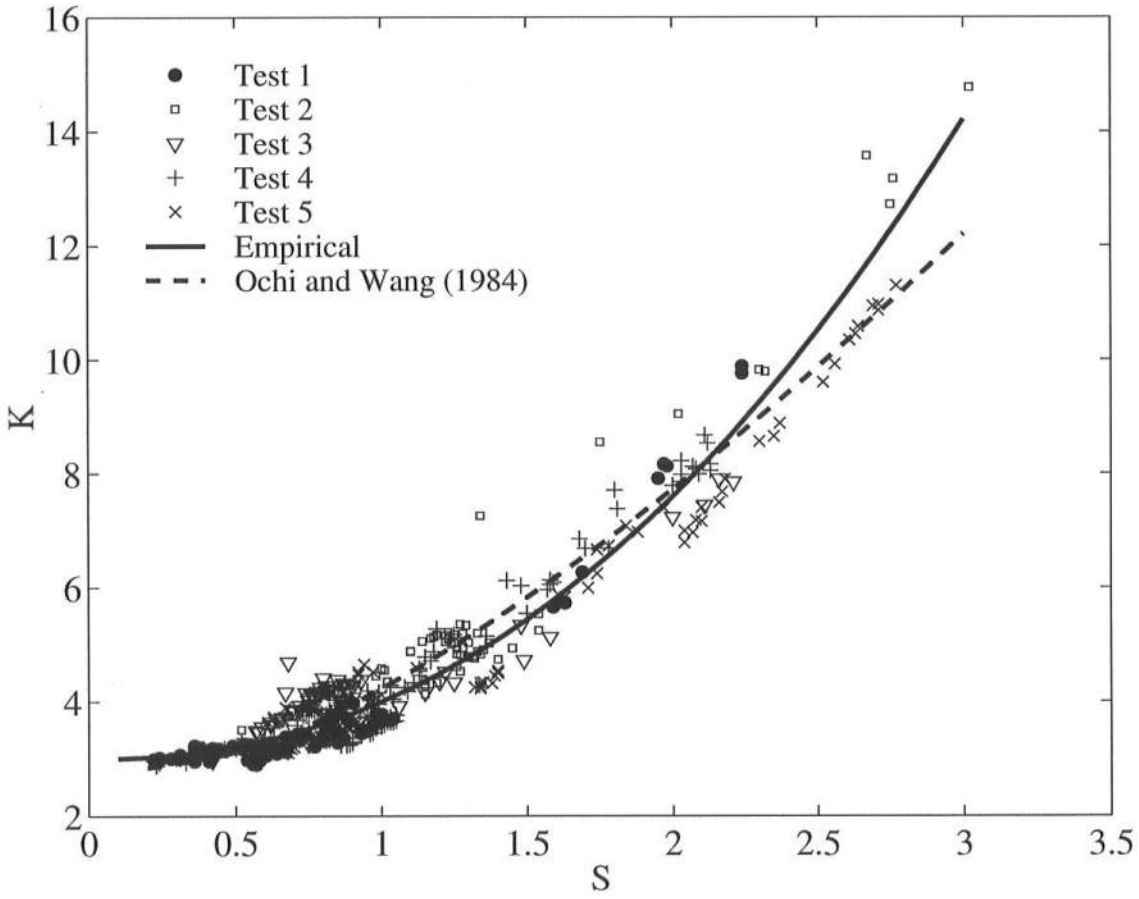

Figure 3: Empirical Formula for Kurtosis K as a Function of Skewness s.

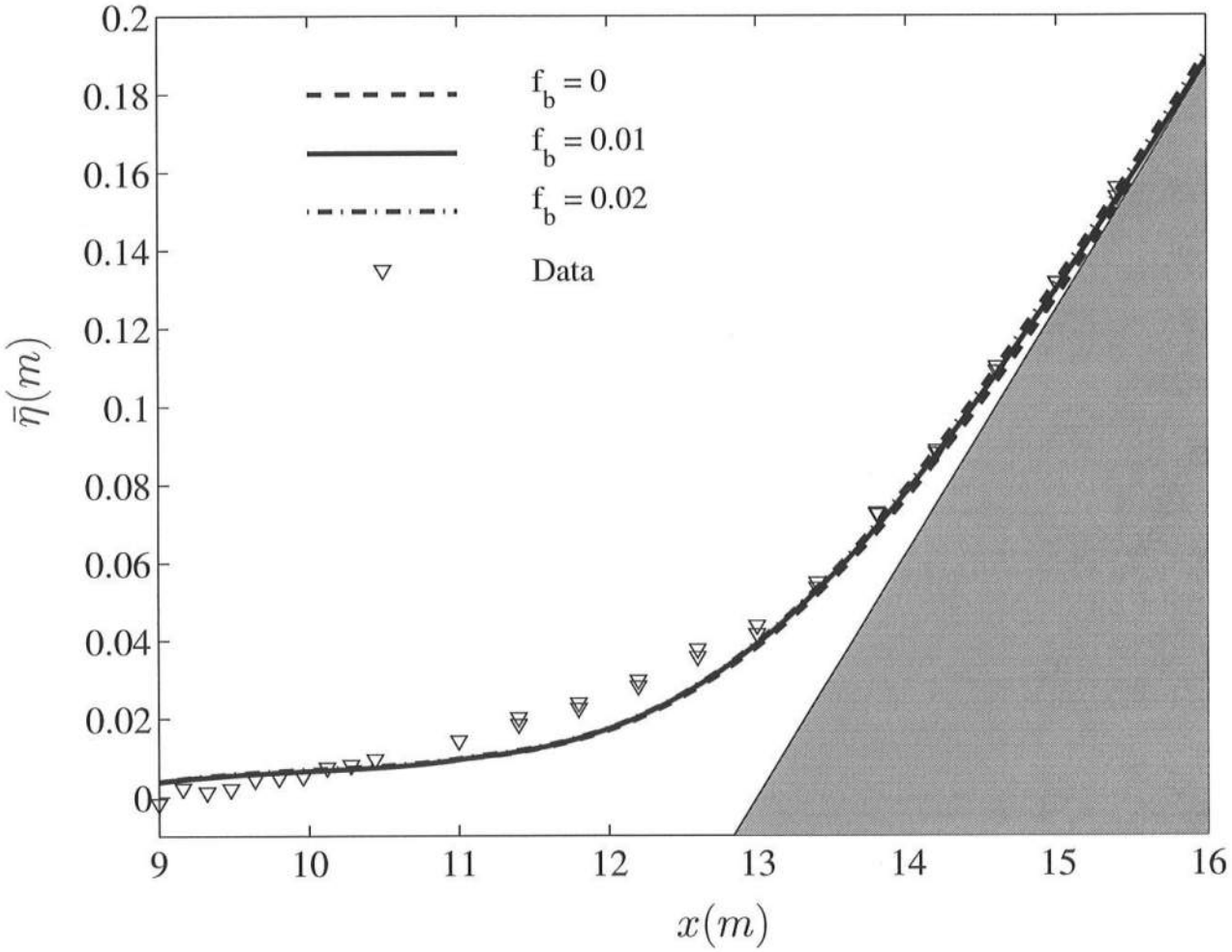

Figure 4: Sensitivity of Computed Wave Setup $\bar{\eta}$ to Bottom Friction Factor f_b = 0, 0.01 and 0.02 for Test 3.

trend is expected from (28) which indicates the increase of $d\overline{h}/dx$ with the increase of f_b where G_b is on the order of -1 in the swash zone. Since the computed setup is not very sensitive to f_b in the range $f_b = 0$–0.02, the subsequent comparisons are made using $f_b = 0$. The use of $f_b = 0$ also facilitates the comparison of CSHORE with the model developed by Battjes and Janssen (1978) and calibrated by Battjes and Stive (1985), which is referred to as BJS model hereafter. The present model with $f_b = 0$ reduces to BJS model if $C_s = 0$ in (11), $C_F = 0$ in (16), and $\gamma_s = \gamma$ in (25).

The assumption of $\gamma_s = \gamma$ in (25) cannot account for the landward increase of $H_* = H_{rms}/\overline{h}$ near the still waterline as shown in Fig. 1. To examine the assumptions of $C_s = 0$ and $C_F = 0$, Fig. 5 shows the computed cross-shore variations of n, Q, C_s and C_F for test 3. The finite-depth adjustment parameter n included in (11) and (16) approaches $n = 1$ in shallow water. The fraction Q of breaking waves estimated by (22) is used in (21) for $\overline{D}_B$ and to find the landward limit x_i of the outer zone where $Q = 1$ is plotted in the inner zone in Fig. 5. The nonlinear corrections C_s and C_F included in (11) and (16) can be neglected if $C_s \ll (2n - 0.5)$ and $C_F \ll 1$. Fig. 5 and the computed results for the other tests indicate that C_s and C_F are on the order of unity in the inner zone. Consequently, the cross-shore radiation stress S_{xx} given by (11) and the energy flux $\overline{E}_F$ expressed by (16) are increased by the nonlinear corrections C_s and C_F in the inner zone.

Fig. 6 shows the computed cross-shore variations of $S^*_{xx} = S_{xx}/\rho g$, $E^*_F = \overline{E}_F/\rho g$ and $D^*_B = \overline{D}_B/\rho g$ for test 3. The inclusion of C_s and C_F and the use of $\gamma_s = 2 > \gamma$ in CSHORE allow the extension of BJS model into the swash zone where the still waterline is located at $x = 13$ m in Fig. 6. The energy dissipation rate $\overline{D}_B$ in the outer zone $x < x_i$ estimated by (21) is positive or zero. In the inner zone, $\overline{D}_B$ is computed using the energy equation (5) with $\overline{D}_f = 0$ for $f_b = 0$.

Figs. 7–11 compare the measured and computed cross-shore variations of $\overline{\eta}$, H_{rms}, s and K for tests 1–5, respectively. The variations of $\overline{\eta}$ and H_{rms} computed by BJS model are also plotted in these figures where BJS model does not predict the skewness s and the kurtosis K of the free surface elevation. The bottom elevation $z_b(x)$ above and below SWL is shown in the first and second panels, respectively, in Figs. 7–11 to show the effects of the beach profile on the wave setup $\overline{\eta}$ and the root-mean-square wave height H_{rms}. The data points from repeated runs in each test are presented without averaging to indicate the degree of the data scatter which was apparent in the swash zone because of the difficulty in measuring small water depth accurately (Kobayashi et al. 1997, 1998).

For tests 1–3 shown in Figs. 7–9, breaker types on the 1:16 smooth slope varied from mostly spilling breakers for test 1 to predominantly plunging breakers for test 3. Correspondingly, the inner zone became wider from test 1 to test 3 where $(x_s - x_i) = 0.9$, 3.0 and 4.7 m for test 1, 2 and 3, respectively, in Table 1. Comparing CSHORE and BJS model, the computed variations of $\overline{\eta}$ and H_{rms} in the outer zone $x < x_i$ are practically the same in view of the larger uncertainty associated with the empirical formula (21) with (22)–(24). No attempt is made to calibrate γ to improve the agreement for H_{rms} in the outer zone for test 2. In the inner zone $x > x_i$, CSHORE is capable of predicting the larger increase of the wave setup $\overline{\eta}$ and the more gradual decrease of the wave height H_{rms} in the inner zone.

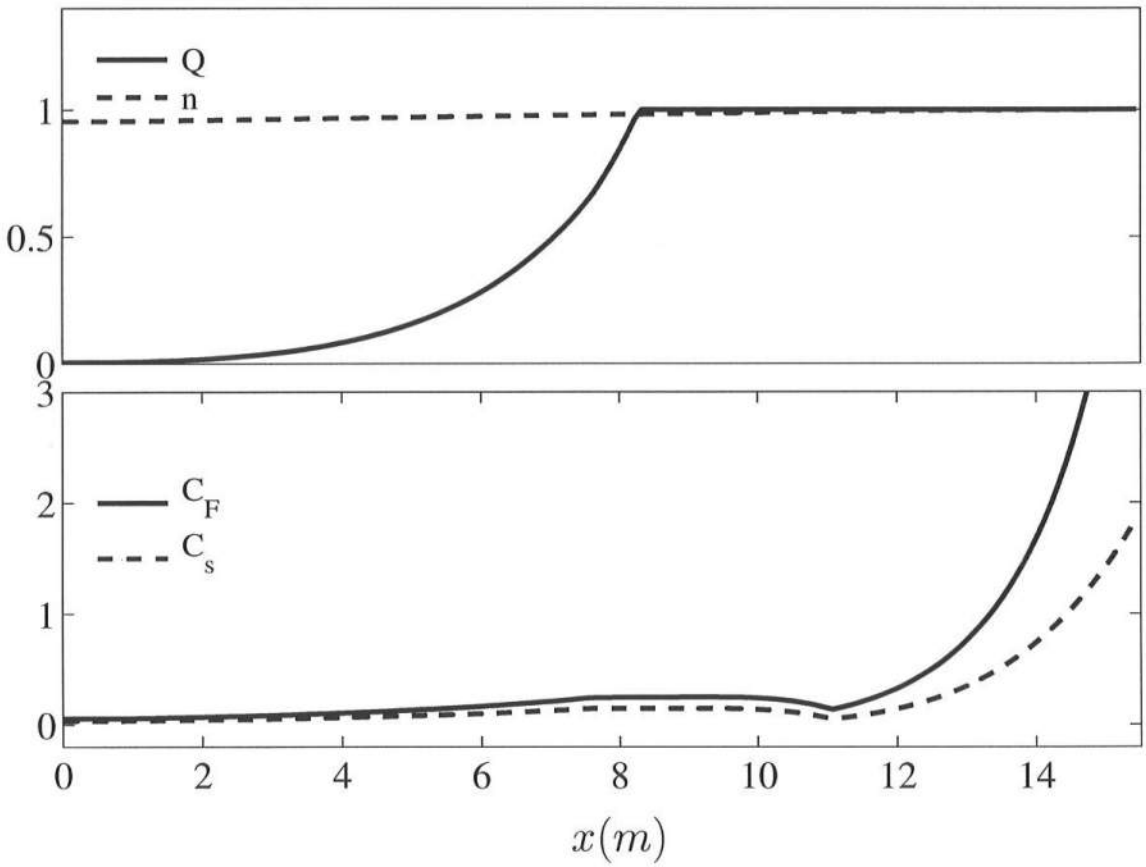

Figure 5: Computed Cross-Shore Variations of n, Q, C_s and C_F for Test 3.

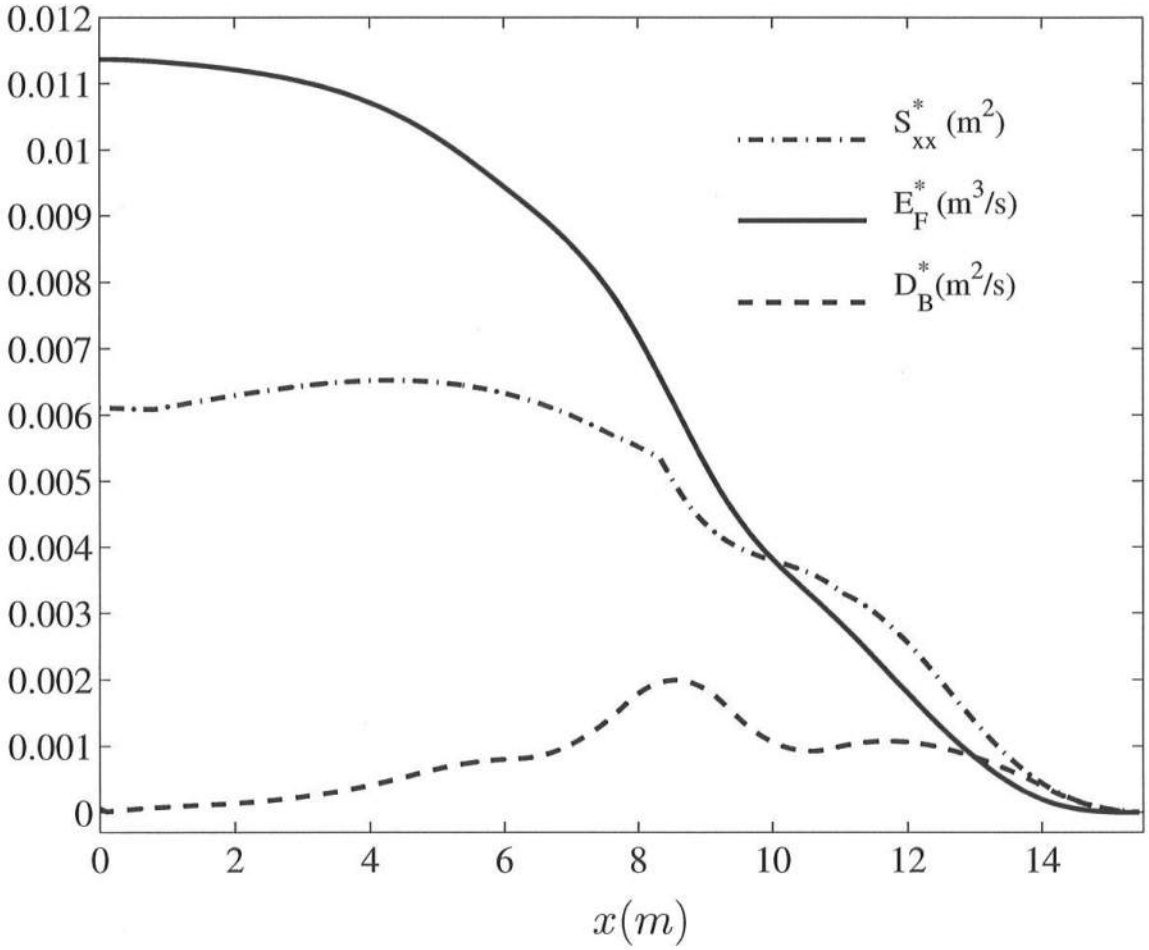

Figure 6: Computed Cross-Shore Variations of S^*_{xx}, E^*_F, and D^*_B for Test 3.

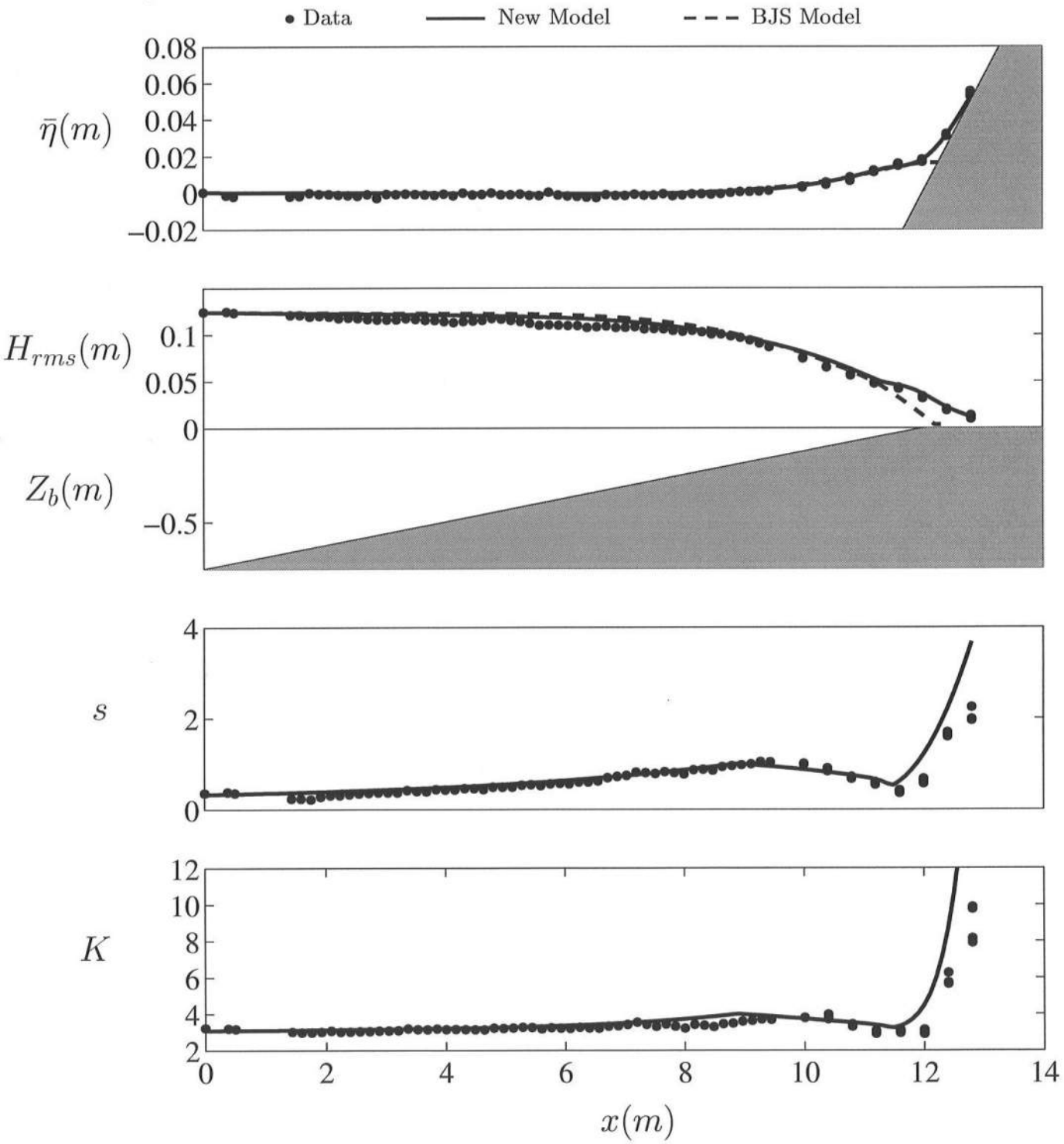

Figure 7: Measured and Computed Setup $\bar{\eta}$, Height H_{rms}, Skewness s, and Kurtosis K for Test 1.

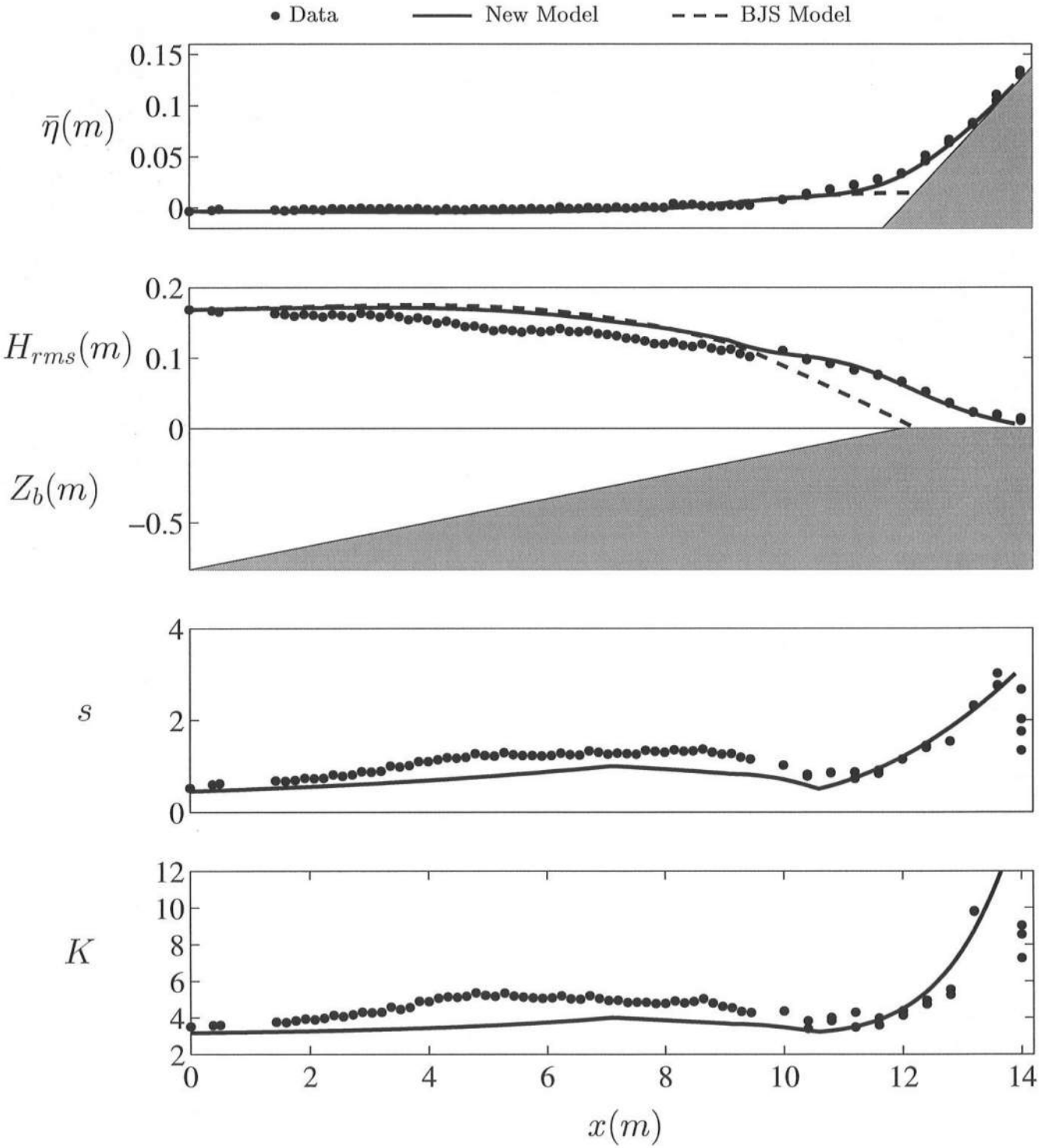

Figure 8: Measured and Computed Setup $\bar{\eta}$, Height H_{rms}, Skewness s, and Kurtosis K for Test 2.

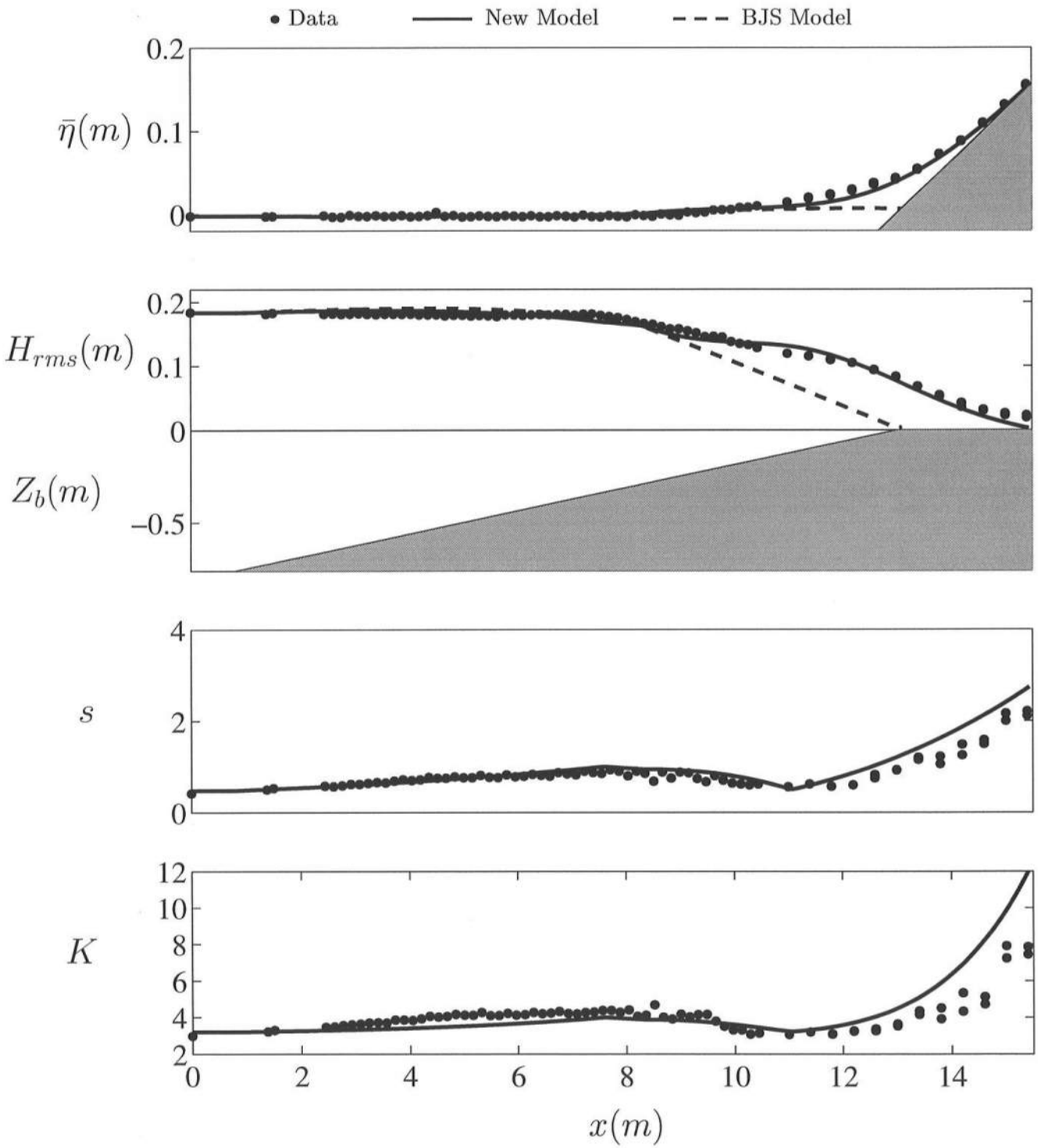

Figure 9: Measured and Computed Setup $\bar{\eta}$, Height H_{rms}, Skewness s, and Kurtosis K for Test 3.

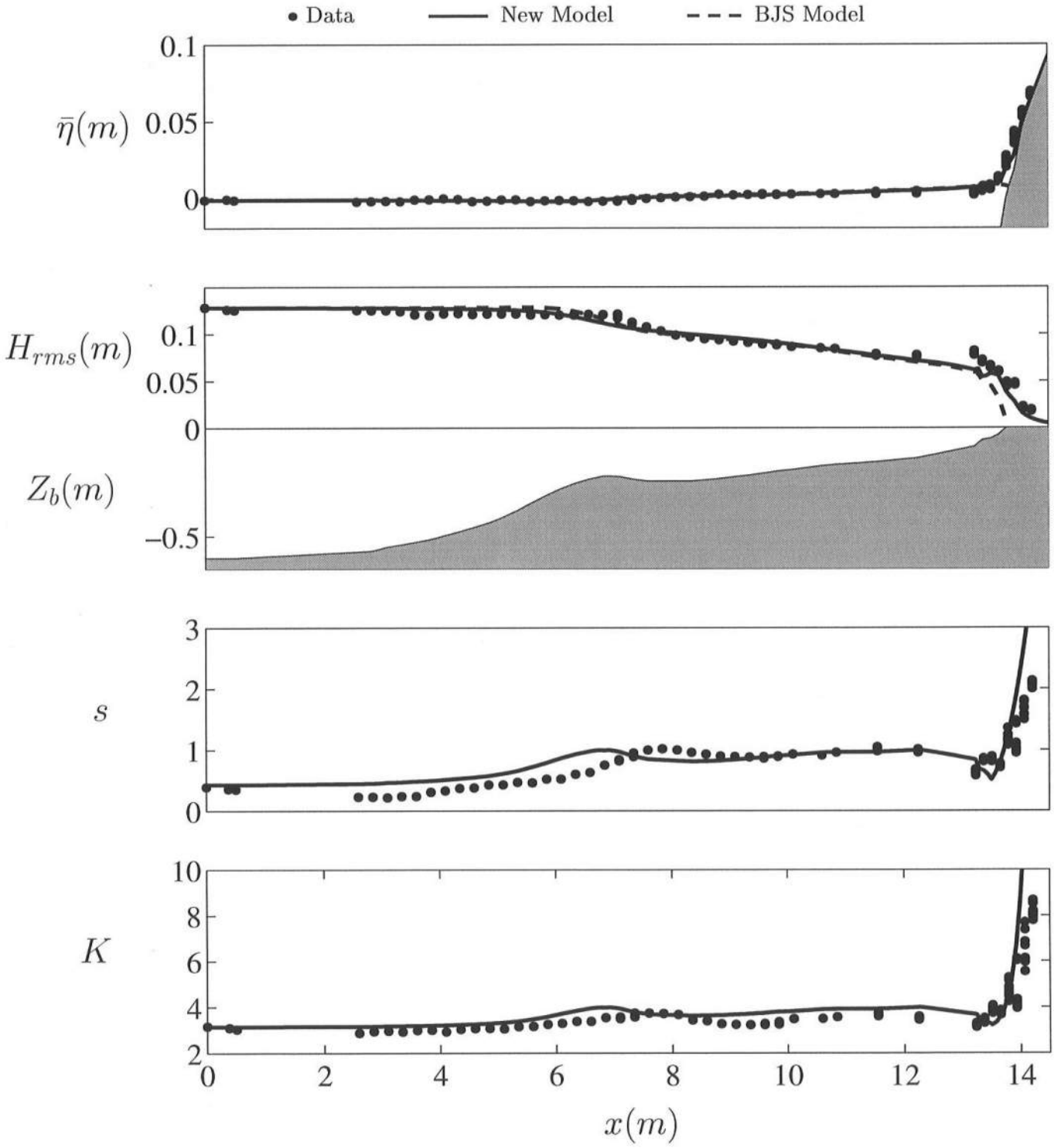

Figure 10: Measured and Computed Setup $\bar{\eta}$, Height H_{rms}, Skewness s, and Kurtosis K for Test 4.

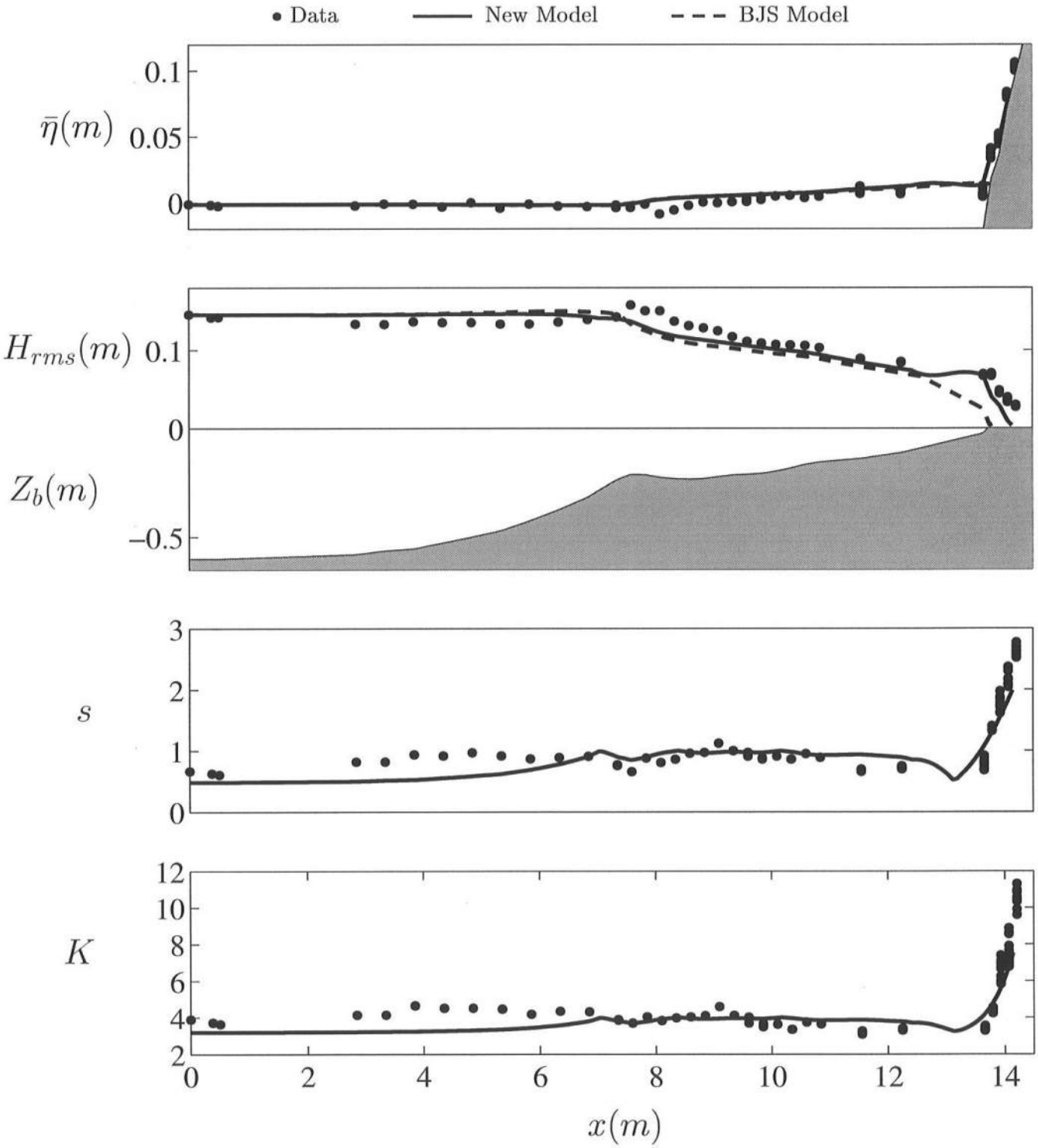

Figure 11: Measured and Computed Setup $\bar{\eta}$, Height H_{rms}, Skewness s, and Kurtosis K for Test 5.

CHSORE also predicts the overall variations of s and K from outside the surf zone to the swash zone, but the empirical formulas (31) and (32) are too simple to yield a good agreement for all the tests.

For tests 4 and 5 shown in Figs. 10 and 11, incident waves shoaled and broke on the small bar at the edge of the terrace. Plunging breakers at the terrace edge were very intense in test 5. Wave breaking was reduced on the terrace before incident waves broke again in the swash zone. BJS model is capable of predicting this wave transformation across the terrace except for the detailed variations of H_{rms} at the terrace edge. The differences between CSHORE and BJS model are limited essentially in the narrow inner zone where $(x_s - x_i) = 0.5$ and 1.3 m for tests 4 and 5, respectively, in Table 1. CSHORE allows the extension of BJS model into the lower swash zone. CSHORE also predicts the overall variations of s and K across the terraced and barred beaches.

Tests 1–5 are limited to beaches without a coastal structure. Kennedy et al. (1997) conducted two tests and measured the cross-shore variations of H_{rms} on a 1:35 slope in front of a stone revetment with a 1:1.5 slope. They compared BJS model with the data and obtained good agreement where reflection coefficients were less than about 40%. CSHORE is expected to yield similar agreement with these two tests because the differences between BJD model and CSHORE are practically limited to the region near the still waterline as shown in Figs. 7–11.

2.4 Conclusions

A time-averaged model is developed to predict the cross-shore variations of the mean, standard deviation, skewness and kurtosis of the free surface elevation from outside the surf zone to the lower swash zone. This time-averaged model derived from the time-dependent continuity, momentum and energy equations which were used successfully to predict irregular wave runup on beaches and coastal structures includes nonlinear corrections terms in the cross-shore radiation stress and energy flux. The correction terms involving the skewness and kurtosis are shown to be important in very shallow water. The time-averaging of the time-dependent equations reduces computation time considerably but creates a closure problem. The energy dissipation rate due to wave breaking is estimated using an existing empirical formula in the outer zone. In the inner zone near the still waterline, a new empirical formula for $H_* = H_{rms}/\overline{h}$ is proposed to describe the landward increase of H_*. In addition, simple empirical formulas are proposed to express the skewness and kurtosis as a function of H_*.

The developed model is compared with three irregular wave tests on a 1:16 smooth impermeable slope and two tests on quasi-equilibrium terraced and barred beaches. The major improvements of the new model in comparison to existing models are that it is capable of predicting the wave setup and root-mean-square wave height near the still waterline and the overall cross-shore variations of the skewness and kurtosis from outside the surf zone to the swash zone. Since the new empirical formulas are developed using the same five tests, the new model will need to be verified using additional tests on beaches and coastal structures. Coupling of the new wave model with a cross-shore sediment transport model may make

it feasible to predict the erosion and recovery near the still water shoreline. Furthermore, this model may be coupled with the empirical formulas by Melby and Kobayashi (1998) to predict damage progression and variability on rubble mound breakwaters. It may also be possible to extend the model to predict irregular wave runup and overtopping on coastal structures.

3 COMPUTER PROGRAM CSHORE

The computer program CSHORE is explained hereafter including the computational details which have been excluded in Section 2 for the sake of readability. The main program performs the computation marching landward from the seaward boundary node located at $x = 0$ with the aid of subroutines arranged in numerical order in the computer program. Double precision mode is used throughout the program which is written in standard FORTRAN-77. The metric units (m and s) are used in CSHORE.

3.1 Main Program

Before the computation marching landward from $x = 0$, CSHORE performs the following groundwork:

- The name of the primary file containing all the input is read as follows:

```
      WRITE(*,*) 'Name of Primary Input-Data-File?'
      READ(*,5000) FINMIN
5000  FORMAT(A10)
```

- Subroutine 1 OPENER (FINMIN) is called to open the input and output files.
- Subroutine 2 INPUT is called to read all the input from the input data file, FINMIN, including $\overline{\eta}$ = wave setup at $x = 0$; H_{rms} = root-mean-square wave height at $x = 0$; and T_p = spectral peak period which is assumed to be invariant in the cross-shore direction. The node number j increases landward with $j = 1$ at $x = 0$.
- Subroutine 3 BOTTOM is called to obtain Δx = constant nodal spacing; x_j = x-coordinate of node j; $(z_b)_j$ = bottom elevation at node j which is positive above SWL; $0.5(f_b)_j$ = half of the bottom friction factor f_b at node j; $(dz_b/dx)_j$ = bottom slope at node j; and j_{max} = upper limit of j corresponding to the number of cross-shore nodes involved in the specified bottom geometry.
- Subroutine 4 PARAM is called to obtain the constants π, 2π, $\sqrt{8}$ and $g = 9.81\ m/s^2$ as well as the deep-water linear wave number $k_o = 2\pi/L_o$ with $L_o = gT_p^2/2\pi$.
- Compute the standard deviation of the free surface elevation, $\sigma = H_{rms}/\sqrt{8}$, and the mean water depth, $\overline{h} = (\overline{\eta} - z_b)$, at node $j = 1$.

- Subroutine 5 LWAVE is called to compute the finite-depth adjustment parameter n given by (15) and the linear wave phase velocity C_p based on T_p at node $j = 1$.

- Subroutine 6 SKEWKU is called to compute the skewness s and the kurtosis K for the known value of $H_* = H_{rms}/\overline{h}$ at node $j = 1$ using (31) and (32), respectively.

- Subroutine 7 CSFFSE is called to compute the nonlinear correction terms C_s and C_F given by (13) and (18), respectively, with $\sigma_* = \sigma/\overline{h}$ at node $j = 1$ as well as the values of F_s and F_E at node $j = 1$ where F_s and F_E are defined as

$$F_s = 2n - \frac{1}{2} + C_s \tag{33}$$

$$F_E = nC_p(1 + C_F) \tag{34}$$

Correspondingly, (11) and (16) are simplified as

$$S^*_{xx} = \frac{S_{xx}}{\rho g} = \sigma^2 F_s \tag{35}$$

$$E^*_F = \frac{\overline{E}_F}{\rho g} = \sigma^2 F_E \tag{36}$$

The values of S^*_{xx} and E^*_F at node $j = 1$ are obtained now.

- Subroutine 8 GBANGF is called if the bottom friction factor $f_b > 0$ at node $j = 1$ to compute G_b and G_f given by (14) and (19), respectively. In the computer program CSHORE, (12) and (17) are rewritten as

$$\tau^*_b = \frac{\overline{\tau}_b}{\rho g} = \frac{1}{2} f_b \, G_b \, \sigma^2_* \, \overline{h} \tag{37}$$

$$D^*_f = \frac{\overline{D}_f}{\rho g} = \frac{1}{2} f_b \, G_f \, \sigma^3_* \, \sqrt{g\overline{h}} \, \overline{h} \tag{38}$$

On the other hand, $\tau^*_b = 0$ and $D^*_f = 0$ if $f_b = 0$. The values of τ^*_b and D^*_f at node $j = 1$ are obtained.

- Subroutine 9 DBREAK is called to compute Q = local fraction of breaking waves given by (22) and $\overline{D}_B$ = energy dissipation rate due to wave breaking given by (21) which is rewritten as

$$D^*_B = \frac{\overline{D}_B}{\rho g} = \frac{\alpha}{4} \, Q \, f_p \, H^2_m \tag{39}$$

which yields the value of D^*_B at node $j = 1$.

The computation marching landward in the outer zone $0 \leq x \leq x_i$ is performed using (26) and (27) which are rewritten as

$$\overline{\eta}_{j+1} = \overline{\eta}_j - (\overline{h}_{j+1} + \overline{h}_j)^{-1} \left\{ 2 \left[(S^*_{xx})_{j+1} - (S^*_{xx})_j \right] + \Delta x \left[(\tau^*_b)_{j+1} + (\tau^*_b)_j \right] \right\} \quad (40)$$

$$(E^*_F)_{j+1} = (E^*_F)_j - \frac{\Delta x}{2} \left[(D^*_f)_{j+1} + (D^*_f)_j + (D^*_B)_{j+1} + (D^*_B)_j \right] \quad (41)$$

For the known values of $\overline{\eta}_j$, $\overline{h}_j$, $(S^*_{xx})_j$, $(\tau^*_b)_j$, $(E^*_F)_j, (D^*_f)_j$, and $(D^*_B)_j$ at node j, the unknown values of these variables at node $(j+1)$ are computed using an iteration method. It is required that $JP1 = (j+1) \leq j_{max}$. If $JP1 > j_{max}$, an error message is written and the computation stops.

The initial estimates of σ_{j+1} and $\overline{h}_{j+1}$ are obtained in the following manner. First, (41) with $(D^*_f)_{j+1} \simeq (D^*_f)_j$ and $(D^*_B)_{j+1} \simeq (D^*_B)_j$ is used to obtain $(E^*_F)_{j+1}$. Using (36), $\sigma^2_{j+1} \simeq (E^*_F)_{j+1}/(F_E)_j$ which must be positive or zero. Otherwise, an error message is written and the computation stops. Second, (35) is used to estimate $(S^*_{xx})_{j+1} \simeq \sigma^2_{j+1}(F_s)_j$. Third, (40) with $\overline{h}_{j+1} \simeq \overline{h}_j$ and $(\tau^*_b)_{j+1} \simeq (\tau^*_b)_j$ yields $\overline{\eta}_{j+1}$ and $\overline{h}_{j+1} = [\overline{\eta}_{j+1} - (z_b)_{j+1}]$. The estimated values of σ_{j+1} and $\overline{h}_{j+1}$ are denoted by σ_{ite} and $\overline{h}_{ite}$. These initial estimates are improved iteratively where the maximum number of iterations allowed in the main program is specified by MAXITE which is taken to be 100 in the DATA statement in the main program.

The iteration algorithm for the outer zone consists of the following steps:

- Subroutine 5 LWAVE is called to find the values of n and C_p corresponding to $\overline{h}_{ite}$.
- Subroutine 6 SKEWKU is called to find the values of s and K corresponding to $H_* = (H_{rms})_{ite}/\overline{h}_{ite}$ with $(H_{rms})_{ite} = \sqrt{8}\,\sigma_{ite}$.
- Subroutine 7 CSFFSE is called to obtain the values of C_s and C_F corresponding to $\sigma_* = \sigma_{ite}/\overline{h}_{ite}$ as well as F_s and F_E given by (33) and (34), respectively.
- Subroutine 8 GBANGF is called if $f_b > 0$ at node $(j+1)$ to obtain G_b and G_f. The values of τ^*_b and D^*_f at node $(j+1)$ are computed using (37) and (38), respectively.
- Subroutine 9 DBREAK is called to find the values of Q and D^*_B given by (39) at node $(j+1)$.
- Eq. (41) is used to find $\sigma^2_{j+1} = (E^*_F)_{j+1}/F_E$. If $\sigma^2_{j+1} < 0$, an error message is written and the computation stops.
- Eq. (35) yields $(S^*_{xx})_{j+1} = \sigma^2_{j+1}\,F_s$. Eq. (40) with $\overline{h}_{j+1} = \overline{h}_{ite}$ is then used to compute $\overline{\eta}_{j+1}$ and $\overline{h}_{j+1} = [\overline{\eta}_{j+1} - (z_b)_{j+1}]$.
- If $|\sigma_{j+1} - \sigma_{ite}| < \epsilon$ and $|\overline{h}_{j+1} - \overline{h}_{ite}| < \epsilon$, the iteration is regarded to have converged where $\epsilon = 10^{-5}m$ is specified in the DATA statement in the main program. Otherwise,

the iteration is continued using $(\sigma_{j+1} + \sigma_{ite})/2$ and $(\overline{h}_{j+1} + \overline{h}_{ite})/2$ as the new values of σ_{ite} and $\overline{h}_{ite}$ to accelerate the convergence. If the iteration does not converge after MAXITE iterations, an error message is written and the computation stops.

- The converged values of σ_{j+1} and $\overline{h}_{j+1}$ are used to compute the corresponding values of $H_{rms} = \sqrt{8}\ \sigma$, $\sigma_* = \sigma/\overline{h}$, $\overline{\eta} = (\overline{h} + z_b)$, S^*_{xx}, E^*_F, τ^*_b, D^*_f, Q and D^*_B at node $(j+1)$ in a manner similar to the computation of these variables at node $j = 1$ for the known values of σ_1 and $\overline{h}_1$. If $\overline{h}_{j+1} < \epsilon$ or $(j+1) = j_{max}$, the landward limit of the computation is reached and this is the end of the marching computation.

- If $Q_{j+1} = 1.0$ and $(dz_b/dx)_{jj} > 0$ for $jj = (j+1), \ldots, j_{max}$, the inner zone is reached. The landward boundary of the outer zone is taken as $x_i = x_{j+1}$ and its nodal location $JXI = (j+1)$ is stored. Otherwise, the computation marches to the next landward node in the outer zone.

On the other hand, the computation marching landward in the inner zone $x > x_i$ is performed using (30) with (29). Comparison of (29) with $\sigma_* = \sigma/\overline{h}$ and (35) with (13) and (33) indicates

$$P = \sigma_*^2 F_s = S^*_{xx} \overline{h}^{-2} \tag{42}$$

In light of (37), it is convenient to introduce

$$R = \frac{1}{2} f_b\ G_b\ \sigma_*^2 = \tau^*_b \overline{h}^{-1} \tag{43}$$

In the main program, $SXXH2 = P$ and $TBH = R$ are used because of (42) and (43), respectively. For the computation, (30) is rewritten as

$$\overline{h}_{j+1} = (3P_{j+1} + C_1)^{-1}[(P_{j+1} + C_2)\overline{h}_j - C_3] \tag{44}$$

with

$$C_1 = P_j + 2 \tag{45}$$

$$C_2 = 3P_j + 2 \tag{46}$$

$$C_3 = 2[(z_b)_{j+1} - (z_b)_j] + \Delta x(R_{j+1} + R_j) \tag{47}$$

For the known values of $\overline{h}_j$, P_j and R_j at node j starting from $j = JXI$, (44) is solved iteratively to find the values of $\overline{h}_{j+1}$, P_{j+1} and R_{j+1} at node $(j+1)$. The bottom elevations $(z_b)_j$ and $(z_b)_{j+1}$ are known from the bathymetry specified as input. The values of P_j and R_j at $j = JXI$ are obtained using (42) and (43) with the known values of S^*_{xx} and τ^*_b at node $j = JXI$.

The iteration algorithm for the inner zone consists of the following steps:

- For the known location of x_{j+1} at node $(j+1)$, (25) is used to find the value of $H_* = H_{rms}/\overline{h}$ at node $(j+1)$ where the landward boundary location x_i of the outer zone has been obtained at the end of the outer zone computation and the still waterline location x_s has been obtained from the bathymetry specified as input in Subroutine 3 BOTTOM. The value of γ given by (24) has been calculated in Subroutine 9 DBREAK before the landward marching computation when $j = 1$. The empirical values of $\gamma_s = 2$ and $\beta = 2.2$ are specified in the DATA statement in the main program.

- For the calculated value of $H_* = H_{rms}/\overline{h}$ at node $(j+1)$, Subroutine 6 SKEWKU is called to obtain the values of s and K at node $(j+1)$ using (31) and (32), respectively. Then, the values of C_s and C_F at node $(j+1)$ are computed using (13) and (18), respectively, where $\sigma_* = \sigma/\overline{h} = H_*/\sqrt{8}$.

- Subroutine 8 GBANGF is called if $f_b > 0$ at node $(j+1)$ to compute the values of G_b and G_f at node $(j+1)$ where G_b and G_f depend on s and σ_* only. The value of R_{j+1} is calculated using (43) where $R_{j+1} = 0$ if $f_b = 0$.

- The values of C_1, C_2 and C_3 given by (45), (46) and (47), respectively, are hence obtained before (44) is solved in the following iteration method.

- The initial estimate of $\overline{h}_{j+1}$ is taken as $\overline{h}_{ite} = \overline{h}_j$. Subroutine 5 LWAVE is called to find the value of n_{j+1} corresponding to $\overline{h}_{ite}$. The values of F_s and P at node $(j+1)$ are computed using (33) and (42), respectively, and (44) yields $\overline{h}_{j+1}$. If $\overline{h}_{j+1} < 0$, $\overline{h}_{j+1} = 0$ is set and the landward marching computation is terminated. If $|\overline{h}_{j+1} - \overline{h}_{ite}| < \epsilon$, the iteration is regarded to have converged. Otherwise, the iteration is continued using $\overline{h}_{j+1}$ as the new value of $\overline{h}_{ite}$. This iteration method converges rapidly because n_{j+1} is essentially unity in shallow water and does not depend on $\overline{h}_{ite}$ much. An error message is written if the iteration does not converge after MAXITE iterations.

- After the iteration converges, the values of $\sigma = \overline{h}\sigma_*$, $H_{rms} = \sqrt{8}\,\sigma$ and $\overline{\eta} = (\overline{h} + z_b)$ at node $(j+1)$ are calculated. If $\overline{h}_{j+1} < \epsilon$ or $(j+1) = j_{max}$, the landward limit of the computation is reached and this is the end of the marching computation. Subroutine 5 LWAVE is called to find the values of n and C_p for the converged value of $\overline{h}_{j+1}$. Subroutine 7 CSFFSE is then called to find the values of F_s and F_E given by (33) and (34) at node $(j+1)$. The values of S^*_{xx} and E^*_F at node $(j+1)$ are calculated using (35) and (36). The values of P and τ^*_b at node $(j+1)$ are calculated using (42) and (43) where R_{j+1} has been obtained before the iteration. The value of D^*_f at node $(j+1)$ is calculated using (38) and $Q_{j+1} = 1.0$ is set in the inner zone. Then the computation marches to the next landward node in the inner zone.

After the landward marching computation is completed, the most landward node with the mean water depth $\overline{h} \geq \epsilon$ is identified by its node location JR where $\overline{h}_{JR} \geq \epsilon$ but $\overline{h}_{JR+1} < \epsilon$. The x-coordinate and bottom elevation at node JR are denoted by $XR = x_{JR}$ and $ZR = (z_b)_{JR}$. The energy dissipation rate due to wave breaking in the inner zone with $j = (JXI+1), \ldots, JR$ is computed using the energy equation (5) which is rewritten as

$$D_B^* = -\frac{d}{dx}(E_F^*) - D_f^* \tag{48}$$

where use is made of (36), (38) and (39). Eq. (48) is approximated by

$$(D_B^*)_j = -(2\Delta x)^{-1}\left[(E_F^*)_{j+1} - (E_F^*)_{j-1}\right] - (D_f^*)_j \tag{49}$$

except for $j = JR$ for which use is made of

$$(D_B^*)_j = -(\Delta x)^{-1}\left[(E_F^*)_j - (E_F^*)_{j-1}\right] - (D_f^*)_j \qquad \text{for } j = JR \tag{50}$$

The computed cross-shore variation of D_B^* in the inner zone tends to exhibit rapid spatial oscillations because the small irregularities in the computed cross-shore variation of E_F^* are amplified greatly by the finite difference approximation of the gradient of E_F^* in (48). The small irregularities in E_F^* given by (36) with (34) and (18) may have been caused partly by the adopted empirical formula (31) for the skewness s which does not vary smoothly with $H_* = H_{rms}/\overline{h}$ as shown in Fig. 2. Furthermore, the cross-shore variation of Q is not smooth in the vicinity of $x = x_i$ as shown in Fig. 5. As a result, the computed cross-shore variation of E_F^* for the domain $0 \leq x \leq XR$ is smoothed using cubic spline interpolation with a reduced number of the computed values of E_F^*. Subroutines 10 SPLINE and SPLINT from Numerical Recipes (Press et al. 1986) are called to obtain the second derivatives of the interpolating function at the specified points and then to find the cubic-spline interpolated values of E_F^* at $j = 1, 2, \ldots, JR$. The computed and smoothed cross-shore variations of E_F^* are practically indistinguishable visually for the five tests in this report. The interpolated values of E_F^* are used to compute $(D_B^*)_j$ using (49) and (50) where $(D_B^*)_1$ at $j = 1$ is not smoothed. The smoothed D_B^* is then used to reset JR so that the computed results only in the region of the smoothed $D_B^* \geq 0$ are stored in the output files. The smoothed cross-shore variation of $(D_B^*)_j$ is practically the same as the computed variation of D_B^* in the outer zone and represents the overall variation of D_B^* in the inner zone as will be shown in Section 3.5. It is emphasized that this smoothing procedure is necessary only for D_B^* in the inner zone which is not used in the computation of the rest of the variables in the inner zone.

Finally, Subroutine 11 OUTPUT is called to store the input and computed results.

3.2 Subroutines

All the subroutines called in the main program are explained in the numerical order used to arrange the sequence of the subroutines.

3.2.1 Subroutine OPENER

This subroutine opens the input and output files listed in Table 2.

The names of the output files start with the letter "O" for easy identification. The computed variables are stored with the nodal location x_j to plot the cross-shore variations

Table 2: Summary of Input and Output Files

Unit	File	Description
11	FINMIN	input file containing all the required input to CSHORE
21	ODOC	output containing a summary of input and output
22	OSETUP	x_j, $(z_b)_j$, $\overline{\eta}_j$ and $\overline{h}_j$ for $j = 1, 2, \ldots, JR$
23	OWAVEHT	x_j, $(z_b)_j$, $(H_{rms})_j$ and $(\sigma_*)_j$ for $j = 1, 2, \ldots, JR$
24	OSKEW	x_j and s_j for $j = 1, 2, \ldots, JR$
25	OCURTO	x_j and K_j for $j = 1, 2, \ldots, JR$
26	ODISSIP	x_j, Q_j, $(G_b)_j$ and $(G_f)_j$ for $j = 1, 2, \ldots, JR$
27	ONONLIN	x_j, n_j, $(C_s)_j$ and $(C_F)_j$ for $j = 1, 2, \ldots, JR$
28	OITER	x_j and $NUMITE(j)$ for $j = 1, 2, \ldots, JR$
29	OMOMENT	x_j, $(S^*_{xx})_j$ and $(\tau^*_b)_j$ for $j = 1, 2, \ldots, JR$
30	OENERGY	x_j, $(E^*_F)_j$, $(D^*_B)_j$ and $(D^*_f)_j$ for $j = 1, 2, \ldots, JR$
31	OENERSM	x_j and smoothed $(E^*_F)_j$ and $(D^*_B)_j$ for $j = 1, 2, \ldots, JR$
40	OMESSG	output file containing all messages written during the computation

of the computed variables. NUMITE(j) is the number of iterations at node j performed to solve (40) and (41) or (44). This information can be used to check the efficiency of the iteration methods and improve these methods if necessary.

3.2.2 Subroutine INPUT

This subroutine reads all input required for CSHORE from the input file, FINMIN. The input file is read in the following sequence:

- A header used by a user to identify a specific input file is read first

```
      READ(11,1110) NLINES
      DO 110 I=1, NLINES
         READ(11,1120) (COMMEN(J), J=1,14)
         WRITE(21,1120) (COMMEN(J), J=1,14)
         WRITE(*,*) (COMMEN(J), J=1,14)
  110 CONTINUE
 1110 FORMAT(I8)
 1120 FORMAT(14A5)
```

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

- The seaward boundary conditions required for CSHORE are read second

```
      READ(11,1150) TP, HRMS(1), WSETUP(1)
 1150 FORMAT(3D13.6)
```

where TP = spectral peak period T_p in seconds at $x = 0$; HRMS(1) = root-mean-square wave height H_{rms} in meters at $x = 0$; and WSETUP(1) = wave setup (+) or set-down (−) $\overline{\eta}$ in meters at $x = 0$.

- The number of spatial nodes, JSWL, along the bottom below SWL is read third

```
      READ(11,1110) JSWL
 1110 FORMAT(I8)
```

The constant nodal spacing Δx is obtained in Subroutine 3 BOTTOM using $\Delta x = x_s/\text{JSWL}$ where x_s is the cross-shore distance between the seaward boundary $x = 0$ and the still waterline defined as $z_b = 0$ at $x = x_s$. The integer JSWL must be chosen to be so large that the computed cross-shore variations of the time-averaged quantities do not depend on Δx. For the five tests compared in this report, use is made of $\Delta x \simeq 0.1$ m on the order of H_{rms}.

- The bathymetry in the computation domain consisting of linear segments of different inclination and roughness is read last

```
      READ(11,1110) NBINP
      READ(11,1150) XBINP(1), ZBINP(1)
      DO 140 J=2, NBINP
         READ(11,1150) XBINP(J), ZBINP(J), FBINP(J-1)
 140  CONTINUE
 1110 FORMAT(I8)
 1150 FORMAT(3D13.6)
```

NBINP is the number of points used to describe the input bathymetry where the points are connected by straight lines in Subroutine 3 BOTTOM. The maximum number of input bottom points allowed in CSHORE is specified by the integer NB in the PARAMETER statement where NB = 100 in the computer program listed in Appendix. If NBINP > NB, an error message is written and the computation stops. The number of linear segments is given by (NBINP-1). XBINP(1) is the x-coordinate of the seaward boundary which is taken as $x = 0$ in CSHORE. ZBINP(1) is the z-coordinate (positive above SWL) of the bottom elevation at the seaward boundary. For J = 2, ..., NBINP, XBINP(J) and ZBINP(J) are the x and z-coordinates of the landward point of linear segment (J-1), whereas FBINP(J-1) is the bottom friction factor f_b associated with linear segment (J-1). In this way, it will be possible to include smooth (low f_b) and rough (high f_b) slope segments in the computation domain. XBINP(NPINP) is the

landward limit of the bathymetry which must be landward of the computation limit based on $\overline{h} = \epsilon$. XBINP(J) and ZBINP(J) with J = 1, 2, ..., NBINP must be given in meters.

3.2.3 Subroutine BOTTOM

This subroutine calculates the x and z-coordinates of spatial nodes located on the bottom from the input bathymetry read in Subroutine 2 INPUT. First, the x-coordinate of the still waterline, XS = x_s, is found from the intersection between the still water level (SWL) at $z = 0$ and the linear segment crossing SWL. If no intersection exists, an error message is written and the computation stops. The constant nodal spacing, DX = Δx, is then given by $\Delta x = x_s/\text{JSWL}$ where the integer JSWL has been specified as input. It is possible to modify CSHORE to allow the cross-shore variation of the nodal spacing but this will increase the required input. Since CSHORE is very efficient computationally, use has been made of the constant spacing Δx on the order of H_{rms}.

The maximum node number, JMAX = j_{max}, for the most landward node is determined such that $(j_{max} - 1) \leq [\text{XBINP(NBINP)}/[\Delta x] < j_{max}$. The maximum number of nodes allowed in CSHORE is specified by the integer NN where NN = 2000 is given in the PARAMETER statement in the computer program listed in Appendix. If $j_{max} >$ NN, an error message is written and the computation stops. The x-coordinate of node j is given by $x_j = (j-1)\Delta x$ for $j = 1, 2, \ldots, j_{max}$. After finding the location of node j on the input bathymetry consisting of linear segments, the bottom elevation $(z_b)_j$, the bottom slope $(dz_b/dx)_j$ and the bottom friction factor $(f_b)_j$ at node j are obtained. CSHORE uses FB2(j) = $(f_b/2)_j$ in light of (37), (38) and (43). Finally, the landward boundary location x_i of the outer zone is tentatively set as $x_i = x_s$ until x_i is found in the main program where $x_i < x_s$ in (25).

3.2.4 Subroutine PARAM

The subroutine calculates the following constants and parameter: PI = $\pi = 3.14159$; TWOPI = $2\,\pi$; GRAV = $g = 9.81$ m/s^2; SQR8 = $\sqrt{8}$; and WKPO = $k_o = (2\pi)^2/(gT_p^2)$ = deep-water wave number based on the spectral peak period T_p.

3.2.5 Subroutine LWAVE

This subroutine calculates linear wave quantities for the known mean water depth, WD = $\overline{h}$, and the spectral peak period T_p specified as input. The dispersion relationship based on linear wave theory is written in the following form (Goda 1985)

$$(k_p\overline{h}) - (k_o\overline{h})/\tanh(k_p\overline{h}) = 0 \tag{51}$$

which is solved using the Newton-Raphson iteration method to obtain the wave number, WKP = k_p, based on T_p. The initial estimate of k_p in (51) is taken as the value of k_p

computed on the previous call of this subroutine except for the first call at the seaward boundary node $j = 1$ for which use is made of $k_p\overline{h} \simeq (k_o\overline{h})/[\tanh(k_o\overline{h})]^{1/2}$. After k_p is obtained, the value of WN $= n$ is calculated using (15) and the phase velocity CP $= C_p$ is given by $C_p = 2\pi/(k_p\, T_p)$.

3.2.6 Subroutine SKEWKU

The subroutine calculates the empirical values of the skewness, SKEW $= s$, and the kurtosis, CURTO $= K$, for the known value of HSTA $= H_*$. To facilitate modifications for future applications, the empirical formula (31) is rewritten as

$$s = A\, H_* \qquad \text{for} \quad s \le Y1 \tag{52a}$$

$$s = A\, Y1 - B(H_* - Y1) \qquad \text{for} \quad Y1 < s \le Y2 \tag{52b}$$

$$s = A\, Y1 - B(Y2 - Y1) + C(H_* - Y2) \qquad \text{for} \quad Y2 < s \tag{52c}$$

where $A = 2.0$, $B = 1.0$, $C = 0.7$, $Y1 = 0.5$ and $Y2 = 1.0$ are specified in the DATA statement in this subroutine. After s is obtained, the corresponding value of K is calculated using (32).

3.2.7 Subroutine CSFFSE

The subroutine computes the nonlinear correction terms CS $= C_s$ and CF $= C_F$ given by (13) and (18), respectively, for the known value of SSTA $= \sigma_*$ in this subroutine, where the values of s and K are transmitted through the COMMON statement. The corresponding values of F_s and F_E are then calculated using (33) and (34), respectively.

3.2.8 Subroutine GBANGF

This subroutine computes the values of G_b and G_f given by (14) and (19), respectively, for the known values of SIGSTA $= \sigma_*$ and SKEW $= s$. To compute G_b and G_f, (14) and (19) are rearranged as

$$G_b = 2I_b - (1 + \sigma_*^2) \tag{53}$$

$$G_f = 2I_f + 3\sigma_* + \sigma_*^3 - s \tag{54}$$

with

$$I_b = \int_{\sigma_*}^{\infty} (\eta_* - \sigma_*)^2 f(\eta_*) d\eta_* \tag{55}$$

$$I_f = \int_{\sigma_*}^{\infty} (\eta_* - \sigma_*)^3 f(\eta_*) d\eta_* \tag{56}$$

where use is made of

$$\overline{\eta_*} = \int_{-\infty}^{\infty} \eta_* \, f(\eta_*) d\,\eta_* = 0 \tag{57}$$

$$\overline{\eta_*^2} = \int_{-\infty}^{\infty} \eta_*^2 \, f(\eta_*) d\,\eta_* = 1 \tag{58}$$

$$\overline{\eta_*^3} = \int_{-\infty}^{\infty} \eta_*^3 \, f(\eta_*) d\,\eta_* = s \tag{59}$$

The probability density function $f(\eta_*)$ is assumed to be expressed by the exponential gamma distribution (Kobayashi et al. 1998)

$$f(\eta_*) = [\Gamma(a)]^{-1} \sqrt{\psi'(a)} \exp(-ay - e^{-y}) \tag{60}$$

with

$$y = \sqrt{\psi'(a)}\, \eta_* - \psi(a) \tag{61}$$

where a = shape parameter; Γ = gamma function; ψ = digamma function; and ψ' = trigamma function. The relationship between s and a is given by

$$s = -\psi''(a)[\psi'(a)]^{-1.5} \tag{62}$$

where ψ'' = tetragamma function. The gamma and related functions are explained in Abramowitz and Stegun (1972) and tabulated by Gran (1992).

For the skewness $s = 0$, (60) reduces to the Gaussian distribution

$$f(\eta_*) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\eta_*^2}{2}\right) \qquad \text{for} \quad s = 0 \tag{63}$$

Substitution of (63) into (55) and (56) yields

$$I_b = \frac{1}{2}(1 + \sigma_*^2)\, \mathrm{erfc}\left(\frac{\sigma_*}{\sqrt{2}}\right) - \frac{\sigma_*}{\sqrt{2\pi}} \exp\left(-\frac{\sigma_*^2}{2}\right) \qquad \text{for} \quad s = 0 \tag{64}$$

$$I_f = \frac{2 + \sigma_*^2}{\sqrt{2\pi}} \exp\left(-\frac{\sigma_*^2}{2}\right) - \frac{\sigma_*}{2}(3 + \sigma_*^2)\, \mathrm{erfc}\left(\frac{\sigma_*}{\sqrt{2}}\right) \qquad \text{for} \quad s = 0 \tag{65}$$

where erfc = complementary error function.

For the skewness $s = 2$, (60) becomes the exponential distribution

$$f(\eta_*) = \exp(-\eta_* - 1) \qquad \text{for} \quad s = 2 \tag{66}$$

which is limited for the range $\eta_* \geq -1$. Substitution of (66) into (55) and (56) yields

$$I_b = 2\exp(-\sigma_* - 1) \qquad \text{for} \quad s = 2 \tag{67}$$

$$I_f = 6\exp(-\sigma_* - 1) \qquad \text{for} \quad s = 2 \tag{68}$$

For the range $0 < s < 2$, (55) and (56) need to be integrated numerically. To find a for the known value of s, (62) is rewritten as (Gran 1992)

$$\ln s = F(a) = \ln[-\psi''(a)] - 1.5 \ln[\psi'(a)] \tag{69}$$

To solve (69) using the Newton-Raphson iteration method, the first derivative of $F(a)$ with respect to a is required

$$\frac{dF}{da} = \frac{\psi'''(a)}{\psi''(a)} - \frac{3}{2} \frac{\psi''(a)}{\psi'(a)} \tag{70}$$

where ψ''' = pentagamma function. The initial estimate of a for the iteration is taken as $a \simeq 1.16(2 - s)$ for $s \geq 1$ and $a \simeq (0.5 + s^{-2})$ for $s < 1$ on the basis of the relationship between a and s plotted in Fig. 1 of Kobayashi et al. (1998). The iteration is continued until $|\ln s - F(a)| <$ DELTA where DELTA $= 10^{-6}$ is specified in the DATA statement in this subroutine.

The gamma and related functions for the given value of a are computed as follows. First, the gamma function is computed using Function GAMMLN(a) $= \ln[\Gamma(a)]$ for $a > 1$ listed in Numerical Recipes (Press et al. 1986) where use is made of $\ln[\Gamma(a)] = \ln[\Gamma(a + 1)] - \ln a$ for $0 < a \leq 1$. The poly-gamma functions $\psi(a)$, $\psi'(a)$, $\psi''(a)$ and $\psi'''(a)$ for $a \geq 10$ are computed using Functions DIGAMM, TRIGAM, TETRAG and PENTAG, respectively, which are based on the expressions given by Gran (1992). For $a < 10$, the following recurrence formulas are applied

$$\psi(a) = \psi(x) - \sum_{i=1}^{10-n} (x - i)^{-1} \tag{71}$$

$$\psi'(a) = \psi'(x) + \sum_{i=1}^{10-n} (x - i)^{-2} \tag{72}$$

$$\psi''(a) = \psi''(x) - \sum_{i=1}^{10-n} 2(x - i)^{-3} \tag{73}$$

$$\psi'''(a) = \psi'''(x) + \sum_{i=1}^{10-n} 6(x - i)^{-4} \tag{74}$$

$$\tag{75}$$

where n is the integer satisfying $(a - 1) < n \leq a$ and $x = (10 + a - n)$ is in the range $10 \leq x < 11$.

Eqs. (55) and (56) with (60) are integrated numerically using the extended Simpson's rule [e.g., Press et al. (1986)] where the upper limit of the integration in (55) and (56) is replaced by $(5+4s)$ because $f(\eta_*) \simeq 10^{-6}$ at $\eta_* \simeq (5+4s)$ in (63) and (66). The odd number of points used for the numerical integration is given by NUM where NUM=31 is found to be sufficient and given in the DATA statement in this subroutine.

Figs. 12 and 13 show G_b and G_f as a function of σ_* for $s = 0$, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8 and 2.0, respectively. G_b decreases monotonically with the increase of σ_* for

given s and with the decrease of s for given σ_*. On the other hand, the variations of G_f with respect to σ_* and s are more complex as shown in Fig. 13. G_f increases with the increase of s for given $\sigma_* \simeq 0$ but the trend is opposite for given $\sigma_* \simeq 1$. These figures indicate $-1.85 \leq G_b \leq 0.47$ and $1.60 \leq G_f \leq 4.18$ for $0 \leq \sigma_* \leq 1$ and $0 \leq s \leq 2$.

For the actual computation of G_b and G_f in this subroutine, (64) and (65) for $s = 0$ are used for $s \leq$ SMIN, whereas (67) and (68) for $s = 2$ are used for $s \geq$ SMAX. The values of SMAX = 1.99 and SMIN = 0.15 are given in the DATA statement in this subroutine. These limits are necessary because $\Gamma(a)$, $\psi(a)$, $\psi'(a)$, $\psi''(a)$ and $\psi'''(a)$ become infinite at $a = 0$ corresponding to $s = 2$, whereas a becomes infinite at $s = 0$. Figs. 12 and 13 show that G_b and G_f vary little with s within the ranges $0 \leq s \leq 0.15$ and $1.99 \leq x \leq 2.0$. The complementary error function in (64) and (65) is computed using Function ERFCC given in Numerical Recipes (Press et al. 1986). Furthermore, for the case of $s \geq$ SMAX, G_b and G_f are computed using (53) and (54) with (67) and (68) where $s = 2$ and $\sigma_* = 1$ are assumed to be consistent with the exponential distribution whose lower limit imposed by the bottom elevation (Kobayashi et al. 1998). These approximations affect only G_b and G_f involved in τ_b^* and D_f^* given by (37) and (38). The comparisons shown in Figs. 7–11 are based on the bottom friction factor $f_b = 0$ for which $\tau_b^* = 0$ and $D_f^* = 0$ for any G_b and G_f.

3.2.9 Subroutine DBREAK

This subroutine computes the local fraction of breaking waves, QBREAK=Q, and the corresponding energy dissipation rate, DBSTA=D_B^*, for the known root-mean-square wave height, WHRMS = H_{rms}, and mean water depth, $D = \overline{h}$. When this subroutine is called at node $j = 1$ before the landward marching computation, the empirical breaker parameter, GAMMA = γ, is computed using (24). The deep-water, root-mean-square wave height H_{rmso} based on linear wave shoaling theory is given by [e.g., Shore Protection Manual (1984)]

$$H_{rmso} = H_{rms} \left\{ \tanh(k_p \, \overline{h}) \left[1 + \frac{2k_p \, \overline{h}}{\sinh(2k_p \, \overline{h})} \right] \right\}^{1/2} \tag{76}$$

where k_p = linear wave number based on the spectral peak period T_p at the seaward boundary $x = 0$.

For the computation of Q_j and $(D_B^*)_j$ at node j in the outer zone with $j = 1, 2, \ldots,$ JXI, the local depth-limited wave height H_m is calculated using (23). Eq. (22) is rewritten as

$$1 - Q + \left(\frac{H_{rms}}{H_m} \right)^2 \ln Q = 0 \tag{77}$$

If $(H_{rms}/H_m)^2 < 0.99999$, (77) is solved to find Q using the Newton-Raphson iteration method starting from $Q \simeq (H_{rms}/H_m)^2/2$. Otherwise, $Q = 1$ is set because $Q = 1$ for $H_{rms} = H_m$ in the formula by Battjes and Stive (1985). The value of D_B^* is calculated using (39) where $f_p = T_p^{-1}$ has been calculated in Subroutine 2 INPUT and $\alpha = 1.0$ is specified in the DATA statement, ALPHA = 1.0, in this subroutine.

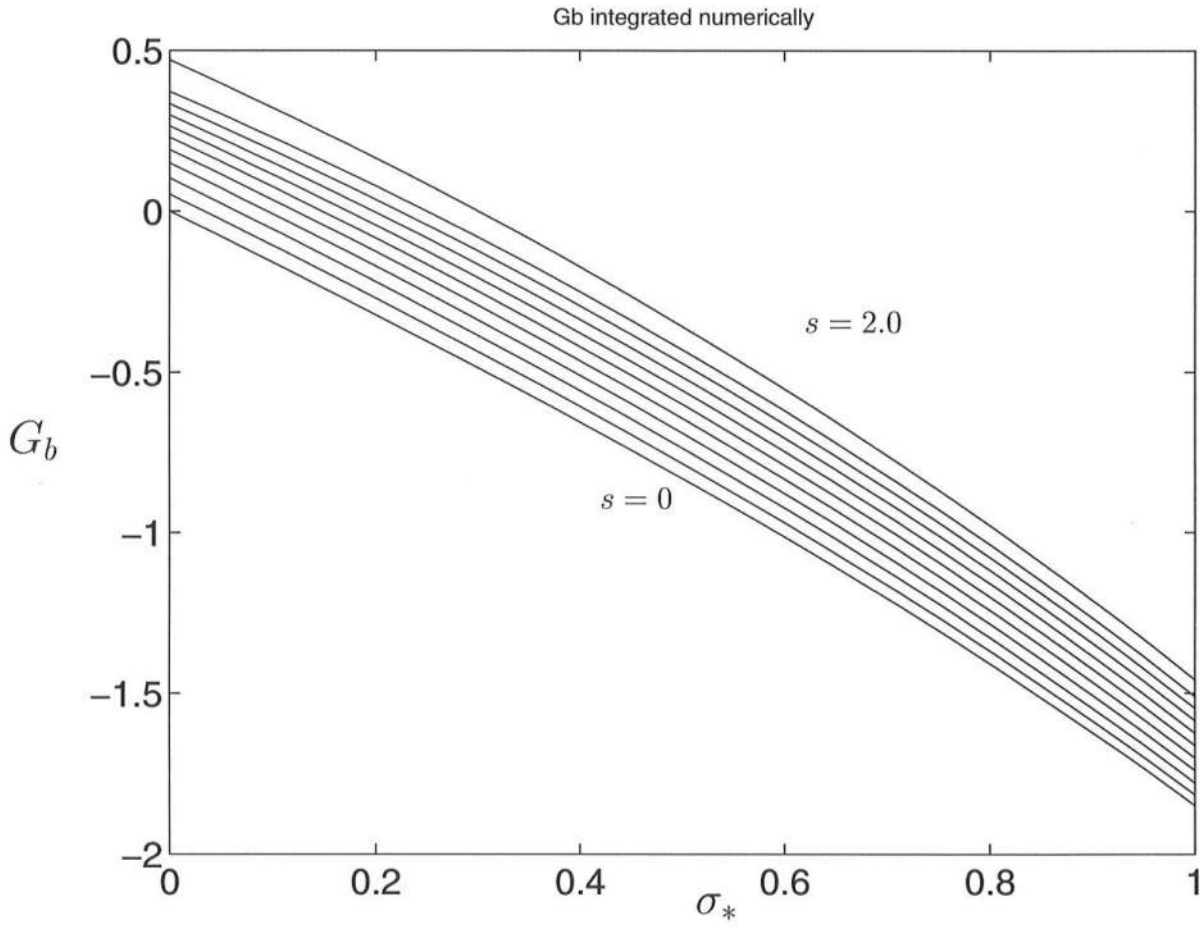

Figure 12: G_b as a Function of σ_* for $s = 0,\ 0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0,\ 1.2,\ 1.4,\ 1.6,\ 1.8$ and 2.0.

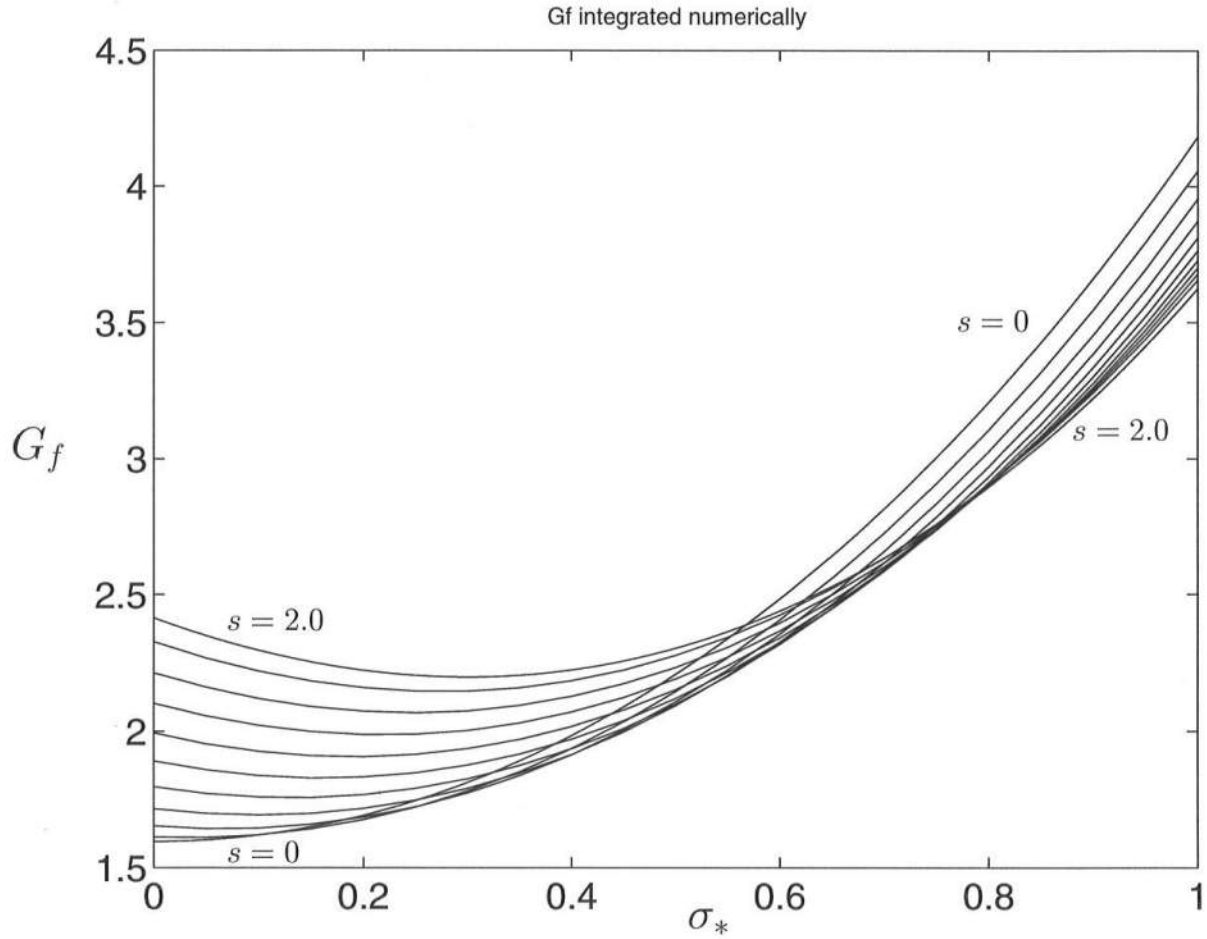

Figure 13: G_f as a Function of σ_* for $s = 0,\ 0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0,\ 1.2,\ 1.4,\ 1.6,\ 1.8$ and 2.0.

3.2.10 Subroutines SPLINE and SPLINT

These subroutines for cubic spline interpolation, which are double-precision versions of SPLINE and SPLINT listed in Numerical Recipes (Press et al. 1986), are used to smooth small irregularities in the computed cross-shore variation of E_F^* after the landward marching computation as explained in Section 3.1. These subroutines are explained clearly by Press et al. (1986) and their explanations are not repeated here.

3.2.11 Subroutine OUTPUT

This subroutine stores the input and computed results in the output files listed in Table 2.

First, the following quantities are stored in File 21 ODOC whose printout will be presented in Section 3.5:

```
      WRITE(21,1000) TP, FP, HRMS(1), WSETUP(1)
      WRITE(21,1100) 0.D0- ZBINP(1), NBINP-1, JSWL, DX, JMAX
      WRITE(21,1200) XBINP(1), ZBINP(1)
      DO 140 J=2, NBINP
         WRITE(21,1200) XBINP(J), ZBINP(J), FBINP(J-1)
140   CONTINUE
      WRITE(21,1300) ALPHA, GAMMA, GAMMAS, BETA, XS, XI
      WRITE(21,1400) EPS1, MAXITE
      WRITE(21,1450) JR, JXI, JMAX, XR, ZR, H(JR)
```

where TP $= T_p$ = spectral peak period in seconds; FP $= f_p$ = spectral peak frequency given by $f_p = T_p^{-1}$; HRM(1) = root-mean-square wave height H_{rms} at the seaward boundary $x = 0$ in meters; WSETUP(1) = wave setup (+) or set-down (−) $\overline{\eta}$ at $x = 0$ in meters; $-ZBINP(1)$ = still water depth at $x = 0$ in meters; (NBINP-1) = number of linear segments used to describe the input bathymetry; JSWL = number of spatial nodes below SWL used to compute $\Delta x = x_s$/JSWL; DX $= \Delta x$ = constant nodal spacing (m) used for the finite difference method; JMAX = maximum landward node corresponding to the landward extent of the input bathymetry; XBINP(1) = x-coordinate (m) of the seaward boundary taken as $x = 0$; ZBINP(1) = z-coordinate (+ above SWL) of the seaward boundary in meters; XBINP(J) = x-coordinate (m) of the landward end point of linear segment $(J-1)$; ZBINP(J) = z-coordinate (m) of the bottom point at $x =$ XBINP(J); FBINP(J-1) = bottom friction factor f_b of linear segment $(J-1)$; ALPHA $= \alpha$ = empirical parameter in (39) specified as $\alpha = 1$ in Subroutine 9 DBREAK; GAMMA $= \gamma$ = empirical parameter given by (24) and computed in Subroutine 9 DBREAK; GAMMAS $= \gamma_s$ and BETA $= \beta$ are the empirical parameters introduced in (25) and specified as $\gamma_s = 2$ and $\beta = 2.2$ in the main program; XS $= x_s$ = x-coordinate (m) of the still waterline on the input bathymetry; XI $= x_i$ = x-coordinate (m) of the landward limit of the outer zone; EPS1 $= \epsilon$ = allowable error for the converged mean water depth (m) for the iterations in the main program which specifies $\epsilon = 10^{-5}$ m; MAXITE = maximum number of iterations allowed for each step during the

landward marching computation where MAXITE = 100 is specified in the main program; JR = most landward node of the computation based on the smoothed $D_B^* \geq 0$; JXI = most landward node in the outer zone; JAX = maximum landward node on the input bathymetry; XR = x-coordinate (m) of node JR; ZR = z-coordinate (m) of node JR; and H(JR) = mean water depth $\overline{h}$ (m) at node JR.

The computed cross-shore variations are stored in Files 22 OSETUP, 23 OWAVEHT, 24 OSKEW, 25 OCURTO, 26 ODISSIP, 27 ONONLIN, 28 OITER, 29 OMOMENT, 30 OENERGY, and 31 OENERSM as follows:

```
DO 160 J=1, JR
      WRITE(22,1500) XB(J), ZB(J), WSETUP(J), H(J)
      WRITE(23,1500) XB(J), ZB(J), HRMS(J), SIGSTA(J)
      WRITE(24,1500) XB(J), SKEW(J)
      WRITE(25,1500) XB(J), CURTO(J)
      IF (FB2(J).GT.0.D0) THEN
      WRITE(26,1500) XB(J), QBREAK(J), GB(J), GF(J)
      ELSE
      WRITE(26,1500) XB(J), QBREAK(J)
      ENDIF
      WRITE(27,1500) XB(J), WN(J), CS(J), CF(J)
      WRITE(28,1510) XB(J), NUMITE(J)
      WRITE(29,1500) XB(J), SXXSTA(J), TBSTA(J)
      WRITE(30,1500) XB(J), EFSTA(J), DBSTA(J), DFSTA(J)
      WRITE(31,1500) XB(J), EFSM(J), DBSM(J)
160   CONTINUE
1500  FORMAT(4F17.9)
1510  FORMAT(F13.6,I8)
```

where J = node number; XB = x = cross-shore coordinate (m) of node J; ZB = z_b = bottom elevation (m) at node J; WSETUP = $\overline{\eta}$ = wave setup or set-down (m); $H = \overline{h}$ = mean water depth (m); HRMS = H_{rms} = root-mean-square wave height (m); SIGSTA = $\sigma_* = \sigma/\overline{h}$ with $\sigma = H_{rms}/\sqrt{8}$; SKEW = s = free surface skewness; CURTO = K = free surface kurtosis; $QBREAK = Q$ = local fraction of breaking waves; GB = G_b = dimensionless parameter related to the bottom shear stress $\overline{\tau}_b$; GF = G_f = dimensionless parameter related to the energy dissipation rate due to bottom friction, $\overline{D}_f$, where GB and GF are computed only if the bottom friction factor $f_b > 0$; WN = n = finite-depth adjustment parameter with $n = 1$ in shallow water; CS = C_s = nonlinear correction term for the cross-shore radiation stress S_{xx}; CF = C_F = nonlinear correction term for the cross-shore energy flux $\overline{E}_F$; NUMITE = number of iterations made to obtain the converged solution at node J; SXXSTA = $S_{xx}^* = S_{xx}/\rho\ g(m^2)$; TBSTA = $\tau_b^* = \overline{\tau}_b/\rho\ g(m)$; EFSTA = $E_F^* = \overline{E}_F/\rho\ g(m^3/s)$; DBSTA = $D_B^* = \overline{D}_B/\rho\ g(m^2/s)$ with $\overline{D}_B$ = energy dissipation rate due to wave breaking; DFSTA = $D_f^* = \overline{D}_f/\rho\ g(m^2/s)$; EFSM = smoothed $E_F^*(m^3/s)$ which must be essentially the same as EFSTA apart from small irregularities; and DBSM = smoothed D_B^* computed

using the smoothed E_F^*.

3.3 Common Statements

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

/PERIOD/ contains the spectral peak period and related parameters.
TP = T_p = spectral peak period specified as input.
FP = f_p = spectral peak frequency given by $f_p = T_p^{-1}$.
WKPO = k_o = deep-water wave number given by $k_o = (2\pi)^2/(gT_p^2)$.

/PREDIC/ contains the unknown variables predicted by CSHORE.
HRMS(j) = root-mean-square wave H_{rms} at node j.
SIGMA(j) = standard deviation σ of the free surface elevation at node j.
H(j) = mean water depth $\overline{h}$ at node j.
WSETUP(j) = wave setup or setdown $\overline{\eta}$ at node j.
SIGSTA(j) = ratio, $\sigma_* = \sigma/\overline{h}$, at node j.

/BINPUT/ contains the input bathymetry consisting of linear segments.
XBINP(k) = x-coordinate of the seaward and of segment k.
ZBINP(k) = z-coordinate of the seaward end of segment k.
FBINP(k) = bottom friction factor f_b associated with segment k.
NBINP = number of points used to specify the input bathymetry.
JSWL = number of nodes below SWL which is the same as the number of nodal spacings between the seaward boundary and the still waterline.

/BPROFL/ contains the bathymetry used for the computation.
DX = constant nodal spacing Δx.
XB(j) = x-coordinate of node j.
ZB(j) = bottom elevation z_b at node j.
FB2(j) = value of $f_b/2$ at node j.
DZBDX(j) = bottom slope, dz_b/dx, at node j.
JMAX = maximum node number at the landward limit of the input bathymetry.

/CONSTA/ contains constants.
GRAV = gravitational acceleration, $g = 9.81 m/s^2$.
SQR8 = constant value of $\sqrt{8}$ used in $H_{rms} = \sqrt{8}\sigma$.
PI = constant value of $\pi = 3.14159$.
TWOPI = constant value of 2π.

/LINEAR/ contains local quantities based on linear wave theory.

WKP = wave number k_p based on the spectral peak period which is calculated at each node.
CP = phase velocity C_p based on the spectral peak period given by $C_p = 2\pi/(k_p T_p)$.
WN(j) = finite-depth adjustment parameter n at node j given by (15) such that $n\,C_p$ is the group velocity.

/**NONLIN**/ contains the skewness and kurtosis which are not included in linear models.
SKEW(j) = free surface skewness s at node j.
CURTO(j) = free surface kurtosis K at node j.

/**FRICTN**/ contains dimensionless parameters related to bottom friction.
GB(j) = parameter G_b at node j given by (14).
GF(j) = parameter G_f at node j given by (19).

/**WBREAK**/ contains quantities related to wave breaking.
ALPHA = empirical parameter α in (21) where $\alpha = 1$ is specified in Subroutine 9 DBREAK.
GAMMA = empirical parameter γ computed using (24) in Subroutine 9 DBREAK.
QBREAK(j) = local fraction Q of breaking waves at node j computed using (22) in the outer zone whereas $Q = 1$ in the inner zone.
DBSTA(j) = computed value of $D_B^* = \overline{D}_B/\rho\,g$ at node j where $\overline{D}_B$ = energy dissipation rate due to wave breaking.
DBSM(j) = smoothed value of D_B^* at node j obtained after the landward marching computation.

/**BRKNEW**/ contains new parameters for wave breaking in the inner zone.
GAMMAS = empirical value of γ_s corresponding to $H_* = H_{rms}/\overline{h}$ at the still waterline where $\gamma_s = 2$ is specified in the main program.
BETA = empirical parameter β in (25) where $\beta = 2.2$ is specified in the main program.
XS = x-coordinate x_s of the still waterline used in (25).
XI = x-coordinate x_i of the landward limit of the outer zone.

/**MOMENT**/ contains terms involved in the cross-shore momentum equation.
CS(j) = nonlinear correction term C_s at node j given by (13).
FS = parameter F_s defined by (33) at each node.
SXXSTA(j) = computed value of $S_{xx}^* = S_{xx}/\rho\,g$ at node j where S_{xx} = cross-shore radiation stress.
TBSTA(j) = computed value of $\tau_b^* = \overline{\tau}_b/\rho\,g$ at node j where $\overline{\tau}_b$ = corss-shore bottom shear stress.

/**ENERGY**/ contains terms involved in the corss-shore energy equation.
CF(j) = nonlinear correction term C_F at node j given by (18).
FE = parameter F_E defined by (34) at each node.
EFSTA(j) = computed value of $E_F^* = \overline{E}_F/\rho\,g$ at node j where $\overline{E}_F$ = cross-shore energy flux.
DFSTA(j) = computed value of $D_f^* = \overline{D}_f/\rho\,g$ at node j where $\overline{D}_f$ = energy dissipation rate due to bottom friction.
EFSM(j) = smoothed value of E_F^* at node j obtained after the landward marching computation.

/**ITERAT**/ contains parameters for the iteration method in CSHORE.

EPS1 = allowable error ϵ for the converged mean water depth at each node where $\epsilon = 10^{-5}$ m is specified in the main program.
MAXITE = maximum number of iterations allowed at each node where MAXITE = 100 is specified in the main program.
NUMITE(j) = number of the iterations performed at node j which can not exceed MAXITE.

/**RUNUP**/ contains parameters for the computational landward limit.
XR = x-coordinate of the landward limit of the smoothed $D_B^* \geq 0$.
ZR = bottom elevation z_b at x = XR.
JR = node number corresponding to x = XR.

The size of the vectors containing the quantities at node j is spceified by the integer NN where it is required that NN $\geq$ JMAX. The size of the vectors containing the quantitites for the input bathymetry is specified by the integer NB where it is required that NB $\geq$ NBINP. In the computer program CSHORE listed in Appendix, NN = 2000 and NB = 100 in the PARAMETER statements in the main program and subroutines.

3.4 Input

The input required for CSHORE has been explained in Section 3.2.2 for Subroutine 2 INPUT. Two examples from the input files used for the comparisons of CSHORE with tests 1–5 in Section 2.3 are presented in the following.

Table 3: Input File for Test 3.

```
3                                                    <-- NLINES
-------------------------------------------------------------------------
Comparision of CSHORE with Test 3                     August, 1998
-------------------------------------------------------------------------
4.7          .1840        -.0024                     <--TP, HRMS(1),WSETUP(1)
128                                                  <--JSWL
3                                                    <--NBINP
0.           -0.762
0.808        -0.762        0.00
22.6          0.6          0.00
```

Table 3 lists the input file for the comparison of CSHORE with test 3. For test 3, T_p = 4.7 s, H_{rms} = 0.184 m at $x = 0$, and $\overline{\eta}$ = − 0.0024 m as listed in Table 1. The number of constant nodal spacings between the seaward boundary $x = 0$ and the still waterline located at x_s = 13 m is taken as JSWL = 128. The nodal spacing given by $\Delta x = x_s/\text{JSWL}$ is hence Δx = 0.102 m. The number of points used to describe the input bathymetry is NBINP = 3. The first linear segment is horizontal and located at z_b = − 0.762 m in the still water depth d = 0.762 m. The horizontal length of this segment is 0.808 m. The slope of the second linear segment is (0.6 + 0.762)/(22.6 - 0.808) = 1/16. The landward end at x = 22.6 m and z_b = 0.6 m of the input bathymetry is taken well above SWL to ensure that the computed mean depth $\overline{h}$ becomes less than $\epsilon = 10^{-5}$ m sufficiently landward of the landward end of the 1:16 slope. The bottom friction factor f_b for these smooth impermeable segments is taken to be $f_b = 0$ as explained in relation to Fig. 4.

On the other hand, Table 4 lists the input file for the comparison of CSHORE with test 5. For test 5, T_p = 2.8 s, H_{rms} = 0.1459 m, $\overline{\eta}$ = − 0.0012 m and d = 0.60 m at $x = 0$. The use of JSWL = 201 results in Δx = 0.068 m and gives a sufficient spatial resolution near the still waterline as shown in Fig. 11. The measured equilibrium beach profile shown in Fig. 11 is represented by NBINP = 34 points and 33 linear segments. The bottom friction factor $f_b = 0$ for all the segments.

3.5 Output

The output produced by CSHORE has been explained in Section 3.2.11 for Subroutine 11 OUTPUT. For example, the printouts of File 21 ODOC for the comparisons of CSHORE with tests 3 and 5 are listed in Tables 5 and 6, respectively, where the corresponding input files have been listed in Tables 3 and 4. The printout of File 21 ODOC is self-explanatory where the symbols have been explained in Section 3.2.11.

On the other hand, the output files containing the computed cross-shore variations are listed in Table 2 and explained in Section 3.2.11. Files OSETUP, OWAVEHT, OSKEW and OCURTO for tests 1–5 have been used to plot Figs. 7–11 where $\overline{h} = (\overline{\eta} - z_b)$ and $\sigma_* = \sigma/\overline{h}$ with $\sigma = H_{rms}/\sqrt{8}$ have not been plotted in these figures for brevity. Files ODISSIP and ONONLIN for test 3 have been used to plot Fig. 5 where GB = G_b and GF = G_f are not computed for the bottom friction factor $f_b = 0$ because (37) and (38) yield $\tau_b^* = 0$ and $D_f^* = 0$ for $f_b = 0$ for any G_b and G_f.

The number of iterations, NUMITE(j), performed at node j is stored in File OITER. Fig. 14 shows the cross-shore variations of NUMITE for tests 1–5. Fig. 14 indicates that the iteration methods adopted in the outer and inner zones normally converge within several iterations.

Finally, Files OMOMENT, OENERGY and OENERSM for test 3 have been used to plot Fig. 6 where $\tau_b^* = 0$ and $D_f^* = 0$ for $f_b = 0$ specified as input. The smoothed cross-shore variations of E_F^* and D_B^* have been plotted in Fig. 6. Fig. 15 compares the computed and smoothed cross-shore variations of E_F^* and D_B^* for test 3 where the smoothing procedure after the landward marching computation has been explained in relation to Eqs. (48)–(50).

Table 4: Input File for Test 5.

```
3                                                    <-- NLINES
-------------------------------------------------------------------------
Comparision of CSHORE with Test 5                     August, 1998
-------------------------------------------------------------------------
2.8          .1459        -0.0012          <--TP, HRMS(1) WSETUP(1)
201                                        <--JSWL
34                                         <--NBINP
0.           -0.6000      0.00
0.3800       -0.6000      0.00
0.5000       -0.6000      0.00
2.8500       -0.5810      0.00
3.3500       -0.5640      0.00
3.8500       -0.5550      0.00
4.3500       -0.5280      0.00
4.8500       -0.5010      0.00
5.3500       -0.4720      0.00
5.8500       -0.4260      0.00
6.3500       -0.3750      0.00
6.8500       -0.3170      0.00
7.3500       -0.2360      0.00
7.6000       -0.2110      0.00
7.8500       -0.2130      0.00
8.1000       -0.2250      0.00
8.3500       -0.2310      0.00
8.6000       -0.2330      0.00
8.8500       -0.2310      0.00
9.1000       -0.2220      0.00
9.3500       -0.2140      0.00
9.6000       -0.2110      0.00
9.8500       -0.2070      0.00
10.1000      -0.1970      0.00
10.3500      -0.1830      0.00
10.6000      -0.1660      0.00
10.8500      -0.1550      0.00
11.5500      -0.1390      0.00
12.2500      -0.1120      0.00
13.6500      -0.0239      0.00
13.7900      0.0170       0.00
13.9300      0.0352       0.00
14.0700      0.0711       0.00
16.0700      0.4287       0.00
```

Table 5: Output File ODOC for Test 3

```
------------------------------------------------------------------
Comparision of CSHORE with Test 3              August, 1998
------------------------------------------------------------------

INPUT WAVE PROPERTIES:
Peak wave period (sec)              = 0.470000E+01
Peak frequency (1/sec)              = 0.212766E+00
Root-mean-square wave height
          at seaward boundary (m) = 0.184000E+00
Wave setup at seaward boundary (m) =-0.240000E-02

INPUT BOTTOM GEOMETRY
Depth at seaward boundary (m)       =     0.762000
Number of linear segments           =       2
Number of spatial nodes  below
   SWL used to find DX              =     128
Node spaceing, DX (m)               =     0.101563
Maximum landward node          JMAX =     223

       X (m)            Zb (m)         Friction factor
     0.000000         -0.762000
     0.808000         -0.762000          0.000000
    22.600000          0.600000          0.000000

EMPIRICAL PARAMETERS FOR WAVE BREAKING
Alpha  =      1.000000
Gamma  =      0.558245
Gammas =      2.000000
Beta   =      2.200000
Xs(m)  =     13.000000
Xi(m)  =      8.328125

ITERATION PARAMETERS
Allowable relative error in iterated depth(m)  =     0.000010
Maximum iterations allowed =       100

WAVE RUNUP OR COMPUTATION LIMIT
Most landward node of computation          JR =     153
Most landward node in outer zone          JXI =      83
             in comparision with         JMAX =     223
X-coordinate of JR (m)                     XR =    15.437500
Z-coordinate of JR (m)                     ZR =     0.152344
Mean water depth of this node (m)        H(JR) =    0.000672
```

Table 6: Output File ODOC for Test 5.

```
------------------------------------------------------------------------
Comparision of CSHORE with Test 5              August, 1998
------------------------------------------------------------------------

INPUT WAVE PROPERTIES:
Peak wave period (sec)              = 0.280000E+01
Peak frequency (1/sec)              = 0.357143E+00
Root-mean-square wave height
          at seaward boundary (m) = 0.145900E+00
Wave setup at seaward boundary (m) =-0.120000E-02

INPUT BOTTOM GEOMETRY
Depth at seaward boundary (m)       =     0.600000
Number of linear segments           =      33
Number of spatial nodes  below
   SWL used to find DX              =     201
Node spaceing, DX (m)               =     0.068317
Maximum landward node          JMAX =     236

      X (m)          Zb (m)        Friction factor
    0.000000       -0.600000
    0.380000       -0.600000          0.000000
    0.500000       -0.600000          0.000000
    2.850000       -0.581000          0.000000
    3.350000       -0.564000          0.000000
    3.850000       -0.555000          0.000000
    4.350000       -0.528000          0.000000
    4.850000       -0.501000          0.000000
    5.350000       -0.472000          0.000000
    5.850000       -0.426000          0.000000
    6.350000       -0.375000          0.000000
    6.850000       -0.317000          0.000000
    7.350000       -0.236000          0.000000
    7.600000       -0.211000          0.000000
    7.850000       -0.213000          0.000000
    8.100000       -0.225000          0.000000
    8.350000       -0.231000          0.000000
    8.600000       -0.233000          0.000000
    8.850000       -0.231000          0.000000
    9.100000       -0.222000          0.000000
    9.350000       -0.214000          0.000000
    9.600000       -0.211000          0.000000
```

Table 6: Continued

```
     9.850000          -0.207000          0.000000
    10.100000          -0.197000          0.000000
    10.350000          -0.183000          0.000000
    10.600000          -0.166000          0.000000
    10.850000          -0.155000          0.000000
    11.550000          -0.139000          0.000000
    12.250000          -0.112000          0.000000
    13.650000          -0.023900          0.000000
    13.790000           0.017000          0.000000
    13.930000           0.035200          0.000000
    14.070000           0.071100          0.000000
    16.070000           0.428700          0.000000

EMPIRICAL PARAMETERS FOR WAVE BREAKING
Alpha  =       1.000000
Gamma  =       0.646164
Gammas =       2.000000
Beta   =       2.200000
Xs(m)  =      13.731809
Xi(m)  =      12.433778

ITERATION PARAMETERS
Allowable relative error in iterated depth(m)  =     0.000010
Maximum iterations allowed =       100

WAVE RUNUP OR COMPUTATION LIMIT
Most landward node of computation            JR =     208
Most landward node in outer zone            JXI =     183
             in comparision with           JMAX =     236
X-coordinate of JR (m)                       XR =    14.141714
Z-coordinate of JR (m)                       ZR =     0.083922
Mean water depth of this node (m)          H(JR) =    0.001112
```

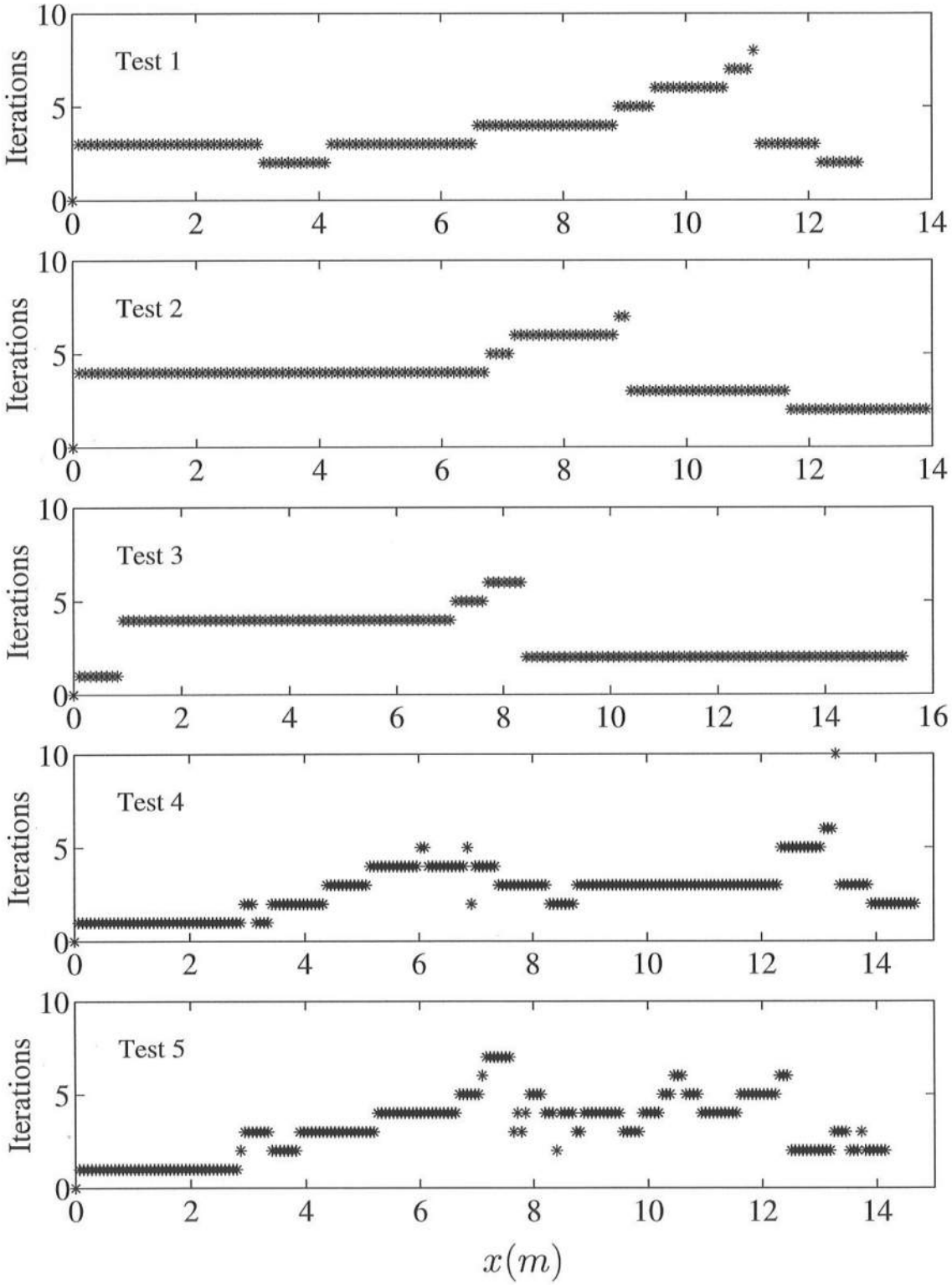

Figure 14: Cross-Shore Variations of Number of Iterations Performed for Tests 1-5.

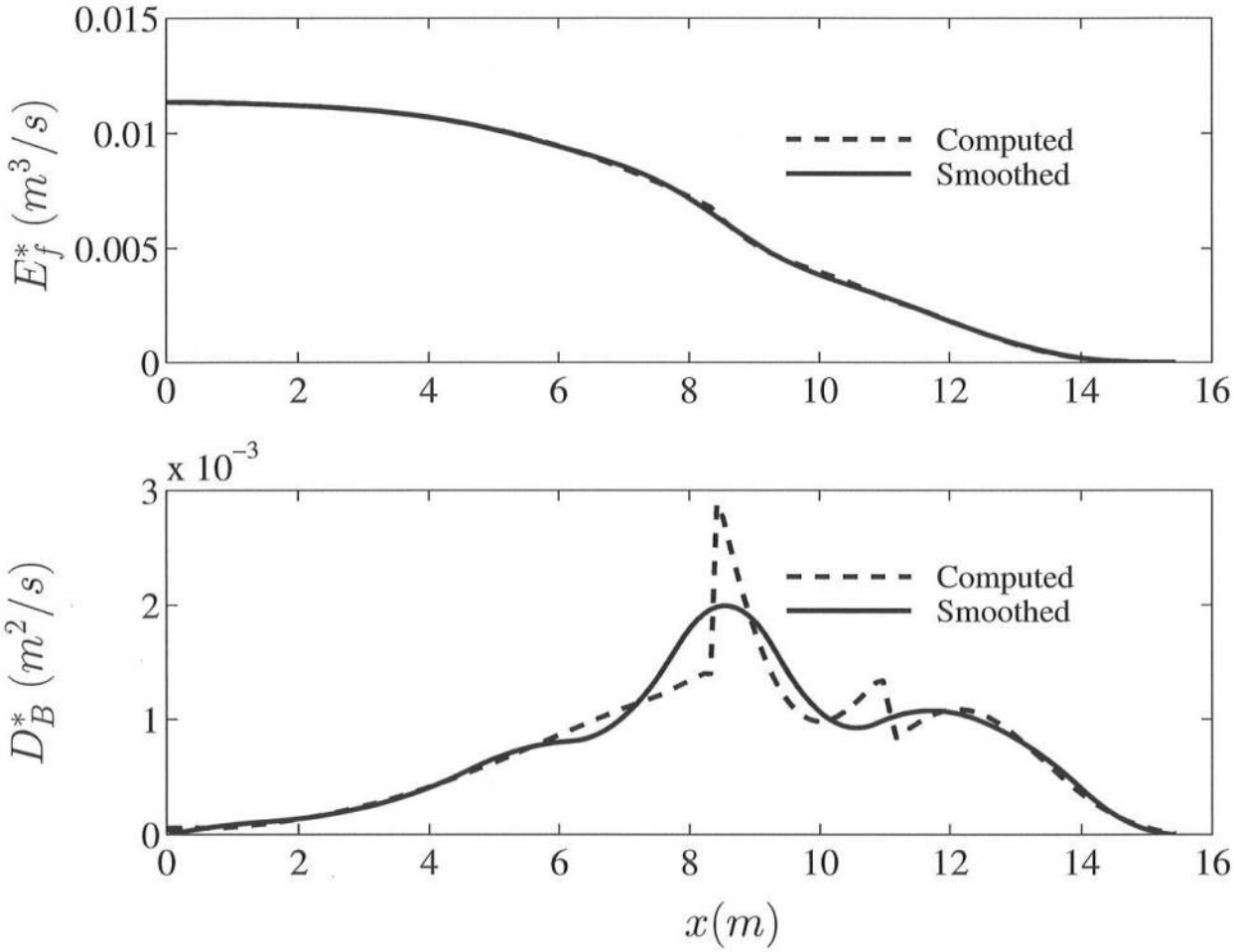

Figure 15: Comparisons of Computed and Smoothed Cross-Shore Variations of E_f^* and D_B^* for Test 3.

The computed and smoothed cross-shore variations of E_F^* are practically the same except for small irregularities in the inner zone $x > x_i = 8.3$ m listed in Tables 1 and 5. The cross-shore variation of D_B^* computed using (49) and (50) with the computed E_F^* exhibits rapid spatial oscillations caused by the small irregularities in the computed E_F^*. The cross-shore variation of D_B^* calculated using the smoothed E_F^* does not exhibit rapid spatial oscillations in the inner zone and appears more realistic. In short, the smoothing procedure is applied to minimize the oscillations of D_B^* in the inner zone without modifying the rest of the computed results by CSHORE.

4 REFERENCES

Abramowitz, M., and Stegun, I.A. (1972). *Handbook of mathematical functions.* Dover Publications, New York, N.Y.

Baquerizo, A., Losada, M.A., Smith, J.M., and Kobayashi, N. (1997). "Cross-shore variation of wave reflection from beaches." *J. Wtrwy., Port, Coast. and Oc. Engrg.*, ASCE, 123(5), 274–279.

Battjes, J.A., and Janssen, J.P.F.M. (1978). "Energy loss and set-up due to breaking of random waves." *Proc., 16th Coast. Engrg. Conf.*, ASCE, New York, N.Y., 1, 569–587.

Battjes, J.A., and Stive, M.J.F. (1985). "Calibration and verification of a dissipation model for random breaking waves." *J. Geophys. Res.*, 90(C5), 9159–9167.

Bouws, E., Gunther, H., Rosenthal, W., and Vincent, C.L. (1985). "Similarity of the wind wave spectrum in finite depth water. 1. Spectral Form." *J. Geophys. Res.*, 90(C1), 975–986.

Bowen, A.J., Inman, D.L., and Simmons, V.P. (1968). "Wave set-down and set-up." *J. Geophys. Res.*, 73(8), 2569–2577.

Cox, D.T., Kobayashi, N., and Kriebel, D.L. (1994). "Numerical model verification using SUPERTANK data in surf and swash zones." *Proc., Coast. Dynamics '94*, ASCE, New York, N.Y., 248–262.

Dally, W.R. (1992). "Random breaking waves: Field verification of a wave-by-wave algorithm for engineering application." *Coast. Engrg.*, 16, 369–397.

Goda, Y. (1985). *Random seas and design of maritime structures.* University of Tokyo Press, Tokyo, Japan.

Gran, S. (1992). *A course in ocean engineering.* Elsevier, New York, N.Y.

Guza, R.T., and Thornton, E.B. (1980). "Local and shoaled comparisons of sea surface elevations, pressures, and velocities." *J. Geophys. Res.*, 85(C3), 1524–1530.

Hedegaard, I.B., Roelvink, J.A., Southgate, H., Pechon, P., Nicholson, J., and Hamm, L. (1992). "Intercomparison of coastal profile models." *Proc., 23rd Coast. Engrg. Conf.*, ASCE, New York, N.Y., 2, 2108–2121.

Hudson, R.Y. (1959). "Laboratory investigation of rubble-mound breakwaters." *J. Wtrwy., Port, Coastal and Oc. Engrg.*, ASCE, 85(3), 93–121.

Kennedy, D.L., Rathbun, J.R., Cox, D.T., and Edge, B.L. (1997). "Irregular wave transformation and undertow for coastal structures in a surf zone." *Proc., Waves'97*, ASCE, Reston, Va., 2, 1115–1128.

Kobayashi, N. (1995). "Numerical models for design of inclined structures." *Wave forces on inclined and vertical wall structures*, ASCE, New York, N.Y., 118–139.

Kobayashi, N. (1998). "Wave runup and overtopping on beaches and coastal structures." *Advances in coastal and ocean engineering*, World Scientific, River Edge, N.J., 5 (in press).

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

Kobayashi, N., Herrman, M.N., Johnson, B.D., and Orzech, M.D. (1998). "Probability distribution of surface elevation in surf and swash zones." *J. Wtrwy., Port, Coast. and Oc. Engrg.*, ASCE, 124(3), 99–107.

Kobayashi, N., and Johnson, B.D. (1998). "Cross-shore variations of wave setup and height in swash and surf zone." *J. Wtrwy., Port, Coast. and Oc. Engrg.*, ASCE (submitted).

Kobayashi, N., Orzech, M.D., Johnson, B.D., and Herrman, M.N. (1997). "Probability modeling of surf zone and swash dynamics." *Proc., Waves '97*, ASCE, Reston, Va., 1, 107–121.

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

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

Kriebel, D.L. (1990). "Advances in numerical modeling of dune erosion." *Proc., 22nd Coast. Engrg. Conf.*, ASCE, New York, N.Y., 3, 2304–2317.

Kriebel, D.L. (1994). "Swash zone wave characteristic from SUPERTANK." *Proc., 24th Coast. Engrg. Conf.*, ASCE, New York, N.Y., 2, 2207–2221.

Lippmann, T.C., Brookins, A.H., and Thornton, E.B. (1996). "Wave energy transformation on natural profiles." *Coast. Engrg.*, 27, 1–20.

Melby, J.A., and Kobayashi, N. (1998). "Progression and variability of damage on rubble mound breakwaters." *J. Wtrwy., Port, Coast. and Oc. Engrg.*, ASCE, 124(6) (in press).

Nairn, R.B., and Southgate, H.N. (1993). "Deterministic profile modelling of nearshore processes. Part 2. Sediment transport and beach profile development." *Coast. Engrg.*, 19, 57–96.

Ochi, M.K., and Wang, W.C. (1984). "Non-Gaussian characteristics of coastal waves." *Proc., 19th Coast. Engrg. Conf.*, ASCE, New York, N.Y., 1, 516–531.

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, U.K.

Raubenheimer, B., and Guza, R.T. (1996). "Observations and predictions of run-up." *J. Geophys. Res.*, 101(C10), 25575–25587.

Raubenheimer, B., Guza, R.T., and Elgar, S. (1996). "Wave transformation across the inner surf zone." *J. Geophys. Res.*, 101(C10), 25589–25597.

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

Shore Protection Manual. (1984). Coast. Engrg. Res. Ctr., U.S. Army Engr. Wtrwy. Experiment Station, U.S. Government Printing Office, Washington, D.C.

Svendsen, I.A. (1984). "Mass flux and undertow in a surf zone." *Coast. Engrg.*, 8, 347–365.

Thornton, E.B., and Guza, R.T. (1983). "Transformation of wave height distribution." *J. Geophys. Res.*, 88(C10), 5925–5938.

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

APPENDIX: Listing of Computer Program CSHORE

- Main Program
- Subroutine 1 OPENER
- Subroutine 2 INPUT
- Subroutine 3 BOTTOM
- Subroutins 4 PARAM
- Subroutine 5 LWAVE
- Subroutine 6 SKEWKU
- Subroutine 7 CSFFSE
- Subroutine 8 GBANGF

  Subroutine INTGRL
  Function GAMMLN
  Function DIGAMM
  Function TRIGAM
  Function TETRAG
  Function PENTAG
  Function ERFCC

- Subroutine 9 DBREAK
- Subroutines 10 SPLINE and SPLINT
- Subroutine 11 OUTPUT

```
C
C      #######    #######   ##    ##     ######    #######    ########
C     ##          ##        ##    ##    ##    ##   ##    ##   ##
C     ##          ##        ##    ##    ##    ##   ##    ##   ##
C     ##          #######   ########    ##    ##   #######    #######
C     ##                ##  ##    ##    ##    ##   ##  ##     ##
C     ##                ##  ##    ##    ##    ##   ##   ##    ##
C      #######   #######    ##    ##     ######    ##    ##   ########
C
C
C
C
C             Nobuhisa Kobayashi and Bradley D. Johnson
C               Center for Applied Coastal Research
C            University of Delaware, Newark, Delaware 19716
C                          August, 1998
C
C ##########################  GENERAL NOTES  ############################
C
C  The purpose of each of 11 subroutines arranged in numerical order
C  is described in each subroutine and where it is called.
C
C  All COMMON statements appear in the Main Program. Description of
C  each COMMON statement is given only in Main Program.
C
C #00#######################  MAIN PROGRAM  #############################
C
C     Main program marches from the offshore boundary node to the
C     shoreline using subroutines.

C
      PROGRAM CSHORE
      IMPLICIT NONE
      INTEGER NN, NB
      PARAMETER (NN=2000, NB=100)
      DOUBLE PRECISION TP, FP, WKPO
      DOUBLE PRECISION HRMS, SIGMA, SIGMA2, H, WSETUP, SIGSTA
      DOUBLE PRECISION XBINP, ZBINP, FBINP
      DOUBLE PRECISION DX, XB, ZB, FB2, DZBDX
      DOUBLE PRECISION GRAV, SQR8, PI, TWOPI
      DOUBLE PRECISION WKP, CP, WN
      DOUBLE PRECISION SKEW, CURTO
      DOUBLE PRECISION GB, GF
      DOUBLE PRECISION ALPHA, GAMMA, QBREAK, DBSTA, DBSM
      DOUBLE PRECISION GAMMAS, BETA, XS, XI
      DOUBLE PRECISION CS, FS, SXXSTA, TBSTA
      DOUBLE PRECISION CF, FE, EFSTA, DFSTA, EFSM
      DOUBLE PRECISION EPS1
      DOUBLE PRECISION XR, ZR
      DOUBLE PRECISION DUM, C1, C2, C3
      DOUBLE PRECISION SIGITE, HITE, HRMITE, HSTA, SXXH2(NN), TBH(NN)
      DOUBLE PRECISION ESIGMA, EH
      DOUBLE PRECISION XBDEC(NN), EFDEC(NN), EF2(NN)
      INTEGER MAXITE, NUMITE
      INTEGER JSWL, NBINP, JMAX, JR, JXI
      INTEGER J, JJ, ICHECK, JP1, ITE, COUNT
      CHARACTER*10 FINMIN
```

```
C
C
C
C ... COMMONs
C
C       Name    Contents
C     ------------------------------------------------------------
C     /PERIOD/  Quantities at the spectral peak period
C     /PREDIC/  Unknowns predicted by CSHORE
C     /BINPUT/  Input bottom geometry
C     /BPROFL/  Discritized bottom geometry
C     /CONSTA/  Constants
C     /LINEAR/  Linear wave values
C     /NONLIN/  Skewness and kurtosis
C     /FRICTN/  Dimensionless parameters related to bottom friction
C     /WBREAK/  Wave breaking quantities and constants
C     /BRKNEW/  New wave breaking parameters in inner zone
C     /MOMENT/  Terms in momentum equation
C     /ENERGY/  Terms in energy equation
C     /ITERAT/  Iteration loop parameters
C     /RUNUP/   Parameters for landward computation limit
C
      COMMON /PERIOD/ TP, FP, WKPO
      COMMON /PREDIC/ HRMS(NN), SIGMA(NN), H(NN), WSETUP(NN), SIGSTA(NN)
      COMMON /BINPUT/ XBINP(NB), ZBINP(NB), FBINP(NB), NBINP, JSWL
      COMMON /BPROFL/ DX, XB(NN), ZB(NN), FB2(NN), DZBDX(NN), JMAX
      COMMON /CONSTA/ GRAV, SQR8, PI, TWOPI
      COMMON /LINEAR/ WKP, CP, WN(NN)
      COMMON /NONLIN/ SKEW(NN), CURTO(NN)
      COMMON /FRICTN/ GB(NN), GF(NN)
      COMMON /WBREAK/ ALPHA, GAMMA, QBREAK(NN), DBSTA(NN), DBSM(NN)
      COMMON /BRKNEW/ GAMMAS, BETA, XS, XI
      COMMON /MOMENT/ CS(NN), FS, SXXSTA(NN), TBSTA(NN)
      COMMON /ENERGY/ CF(NN), FE, EFSTA(NN), DFSTA(NN), EFSM(NN)
      COMMON /ITERAT/ EPS1, MAXITE, NUMITE(NN)
      COMMON /RUNUP/  XR, ZR, JR
      DATA EPS1, MAXITE /1D-5, 100/
      DATA GAMMAS, BETA /2.D0, 2.2D0/
C
      WRITE (*,*) 'Name of Primary Input-Data-File?'
      READ  (*,5000) FINMIN
 5000 FORMAT (A10)
C
C     Subr. 1 OPENER opens input and output files.
      CALL OPENER (FINMIN)
C     Subr. 2 INPUT gets input wave and bathymetry information
C                     from  the input file, FINMIN.
      CALL INPUT
C
C ... PREPARATIONS FOR TIME MARCHING COMPUTATION
C
C     Subr. 3 BOTTOM computes bathymetry at each node.
      CALL BOTTOM
C     Subr. 4 PARAM  calculates constants and parameters.
      CALL PARAM
C
```

```
      SIGMA(1) = HRMS(1)/SQR8
      H(1) = WSETUP(1) - ZB(1)
C
C     Subr. 5 LWAVE returns the linear wave number and ratio of group
C                   velocity to phase velocity for the peak frequency.
      CALL LWAVE(1, H(1))
C
C     Subr. 6 SKEWKU returns skewness and kurtosis of the free surface
C                   using empirical formulas.
      CALL SKEWKU(1,HRMS(1)/H(1))
C
C     Subr. 7 CSFFSE computes CS, CF, FS, and FE involved in
C                   cross-shore radiation stress and energy flux.
      SIGSTA(1) = SIGMA(1)/H(1)
      CALL CSFFSE(1,SIGSTA(1))

C
      SIGMA2 =  SIGMA(1)**2.D0
      SXXSTA(1) = SIGMA2*FS
      EFSTA(1) = SIGMA2*FE
C
C     Subr. 8 GBANGF returns numerically integrated values for the
C                   Gb and Gf factors used in calculating bottom
C                   shear stress and bottom dissipation if bottom
C                   friction coefficient is positive.
      IF(FB2(1).GT.0.D0) THEN
        CALL GBANGF(1)
        TBSTA(1) = FB2(1)*GB(1)*SIGSTA(1)**2.D0*H(1)
        DFSTA(1) = FB2(1)*GF(1)*SIGSTA(1)**3.D0*DSQRT(GRAV*H(1))*H(1)
      ELSE
        TBSTA(1) = 0.D0
        DFSTA(1) = 0.D0
      ENDIF
C
C     Subr. 9 DBREAK computes the fraction of breaking waves and
C                   the associated wave energy dissipation and
C                   returns DBSTA(1).
C
      CALL DBREAK(1, HRMS(1), H(1))
C
C ---------------------- MARCHING COMPUTATION --------------------------
C
C Computation marching landward from seaward boundary, J = 1
      NUMITE(1) = 0
      J = 0
 100  J = J + 1
      JP1 = J + 1
      IF(JP1.GT.JMAX) THEN
         WRITE(*,2900) JMAX
         WRITE(40,2900) JMAX
         STOP
      ENDIF
 2900 FORMAT('ERROR: JP1 is greater than JMAX = ',I3)
C
      IF(XB(JP1).LE.XI) THEN
      DUM = (EFSTA(J) - DX*(DFSTA(J) + DBSTA(J)))/FE
C
```

```
      IF(DUM.LE.0.D0) THEN
         WRITE(*,2901)J
         WRITE(40,2901)J
         STOP
      ENDIF
 2901 FORMAT(/'ERROR: '/
     +   'Square of sigma is negative at node ',I3)
C
      SIGITE  = DSQRT(DUM)
 110  SXXSTA(JP1) = FS*SIGITE**2.D0
C
      WSETUP(JP1) = WSETUP(J) -
     +                (SXXSTA(JP1) - SXXSTA(J) + TBSTA(J)*DX)/H(J)
      HITE = WSETUP(JP1) - ZB(JP1)
C
C Begin iteration for adopted implicit finite difference method
C
      DO 200 ITE = 1, MAXITE
C
         CALL LWAVE(JP1, HITE)
         HRMITE = SIGITE*SQR8
         SIGSTA(JP1) = SIGITE/HITE
C
         CALL SKEWKU(JP1,HRMITE/HITE)
C
         CALL CSFFSE(JP1,SIGSTA(JP1))
C
         IF(FB2(JP1).GT.0.D0) THEN
           CALL GBANGF(JP1)
           TBSTA(JP1)=FB2(JP1)*GB(JP1)*SIGSTA(JP1)**2.D0*HITE
           DFSTA(JP1)=FB2(JP1)*GF(JP1)*SIGSTA(JP1)**3.D0 *
     +           DSQRT(GRAV*HITE)*HITE
         ELSE
           TBSTA(JP1) = 0.D0
           DFSTA(JP1) = 0.D0
         ENDIF
C

         CALL DBREAK(JP1, HRMITE, HITE)
C
         DUM = (EFSTA(J) - DX/2.D0*( DFSTA(JP1) + DBSTA(JP1) +
     +                               DFSTA(J) +   DBSTA(J) ))/FE
         IF(DUM.LE.0.D0) THEN
             WRITE(*,2901)J
             WRITE(40,2901)J
             STOP
         ENDIF
C
         SIGMA(JP1) = DSQRT(DUM)
         SXXSTA(JP1) = FS*SIGMA(JP1)**2.D0
C
         WSETUP(JP1) = WSETUP(J) - (2.D0* (SXXSTA(JP1)-SXXSTA(J)) +
     +                  DX*(TBSTA(JP1)+TBSTA(J)))/(HITE+H(J))
         H(JP1) = WSETUP(JP1) - ZB(JP1)
C
C Check for convergence
C
```

```
          ESIGMA = DABS(SIGMA(JP1) - SIGITE)
          EH = DABS(H(JP1) - HITE)
C
          IF(ESIGMA.LT.EPS1.AND.EH.LT.EPS1) GO TO 210
C
C Averages of new and previous values are used to accelerate convergence
          SIGITE =  0.5D0*(SIGMA(JP1) + SIGITE)
          HITE = 0.5D0*(H(JP1) + HITE)
C
 200   CONTINUE
C
       WRITE(*,2903) MAXITE, EPS1, JP1
       WRITE(40,2903) MAXITE, EPS1, JP1
       STOP
 2903  FORMAT(/'ERROR: Convergence was not reached after MAXITE = ',I4/
      +          ' iterations with relative error EPS1 = ',E7.5/
      +          'at node JP1 = ',I4)
C
 210   NUMITE(JP1) = ITE
       SIGSTA(JP1) = SIGMA(JP1)/H(JP1)
       HRMS(JP1) = SQR8*SIGMA(JP1)
       WSETUP(JP1) = H(JP1) + ZB(JP1)
       IF(H(JP1).LT.EPS1.OR.JP1.EQ.JMAX) GO TO 400
       CALL LWAVE(JP1, H(JP1))
       CALL SKEWKU(JP1, HRMS(JP1)/H(JP1))
       CALL CSFFSE(JP1,SIGSTA(JP1))
       SIGMA2 = SIGMA(JP1)**2.D0
       SXXSTA(JP1) = SIGMA2*FS
       EFSTA(JP1) = SIGMA2*FE
       IF(FB2(JP1).GT.0.D0) THEN
         CALL GBANGF(JP1)
         TBSTA(JP1)=FB2(JP1)*GB(JP1)*SIGSTA(JP1)**2.D0*H(JP1)
         DFSTA(JP1)=FB2(JP1)*GF(JP1)*SIGSTA(JP1)**3.D0 *
      +           DSQRT(GRAV*H(JP1))*H(JP1)
       ELSE
         TBSTA(JP1) = 0.D0
         DFSTA(JP1) = 0.D0
       ENDIF
       CALL DBREAK(JP1,HRMS(JP1),H(JP1))
C
C      Check whether the inner zone is reached
       IF(QBREAK(JP1).EQ.1.0D0) THEN
          ICHECK = 0
          DO 220 JJ = JP1, JMAX
            IF(DZBDX(JJ).LE.0.D0) ICHECK = ICHECK + 1
 220      CONTINUE
          IF(ICHECK.EQ.0) THEN
             XI = XB(JP1)
             JXI = JP1
          ENDIF
       ENDIF
C
       GO TO 100
C
       ENDIF
C*****************************End of IF(XB(JP1).LE.XI)**********************
C
```

```
      IF(XB(JP1).GT.XI) THEN
C
C     If XB(J) = X(I), compute SXXH2(J) and TBH(J)
      IF(XB(J).EQ.XI) THEN
         SXXH2(J) = SXXSTA(J)/H(J)**2.D0
         TBH(J) = TBSTA(J)/H(J)
      ENDIF
C
C Empirical formula for HSTA = HRMS/H for region for XB.GE.XI
C
      HSTA = GAMMA + (GAMMAS - GAMMA)*((XB(JP1) - XI)/(XS - XI))**BETA
      SIGSTA(JP1) = HSTA/SQR8
      CALL SKEWKU(JP1, HSTA)
      DUM = SIGSTA(JP1)**2.D0
      CS(JP1) = SIGSTA(JP1)*SKEW(JP1) - DUM
      CF(JP1) = 1.5D0*SKEW(JP1)*SIGSTA(JP1)*(1.D0 - DUM) +
     +          0.5D0*DUM*(CURTO(JP1)- 5.D0) + DUM**2.D0
      IF(FB2(JP1).GT.0.D0) THEN
        CALL GBANGF(JP1)
        TBH(JP1) = FB2(JP1)*GB(JP1)*DUM
      ELSE
        TBH(JP1) = 0.D0
      ENDIF
      C1 = SXXH2(J) + 2.D0
      C2 = 3.D0*SXXH2(J) + 2.D0
      C3 = 2.D0*(ZB(JP1) - ZB(J)) + DX*(TBH(JP1) + TBH(J))
      HITE = H(J)
      DO 300 ITE = 1, MAXITE
         CALL LWAVE(JP1, HITE)
         FS = 2.D0*WN(JP1) - 0.5D0 + CS(JP1)
         SXXH2(JP1) = DUM*FS
         H(JP1)=  (3.D0*SXXH2(JP1)+C1)**(-1.D0) *
     +                  ((SXXH2(JP1)+C2)*H(J)-C3)
         IF(H(JP1).LE.0.D0) THEN
            H(JP1) = 0.D0
            GO TO 310
         ENDIF
         IF(DABS((H(JP1) - HITE)).LT.EPS1) GO TO 310
         HITE = H(JP1)
 300  CONTINUE
C
      WRITE(*,2903) MAXITE, EPS1, JP1
      WRITE(40,2903) MAXITE, EPS1, JP1
      STOP
C
 310  NUMITE(JP1) = ITE
      SIGMA(JP1) = H(JP1)*SIGSTA(JP1)
      HRMS(JP1) = SQR8*SIGMA(JP1)
      WSETUP(JP1) = H(JP1) + ZB(JP1)
      IF(H(JP1).LT.EPS1.OR.JP1.EQ.JMAX) GO TO 400
C
      CALL LWAVE(JP1, H(JP1))
      CALL CSFFSE(JP1,SIGSTA(JP1))
      SIGMA2 = SIGMA(JP1)**2.D0
      SXXSTA(JP1) = SIGMA2*FS
      SXXH2(JP1) = SXXSTA(JP1)/H(JP1)**2.D0
      EFSTA(JP1) = SIGMA2*FE
```

```
      TBSTA(JP1)=TBH(JP1)*H(JP1)
      IF(FB2(JP1).GT.0.D0) THEN
        DFSTA(JP1)=FB2(JP1)*GF(JP1)*SIGSTA(JP1)**3.D0 *
     +           DSQRT(GRAV*H(JP1))*H(JP1)
      ELSE
        DFSTA(JP1) = 0.D0
      ENDIF
C
      QBREAK(JP1) = 1.D0
C
      GO TO 100
C
      ENDIF
C**************************End of IF(XB(JP1).GT.XI)**********************
C
C     The iteration has converged but H(JP1) is less then EPS1(m) or
C     JP1 = JMAX.  Regard this location as the landward limit of
C     computation.
C
 400  JR = J
      XR = XB(J)
      ZR = ZB(J)
C
C     Calculate the energy dissipation due to breaking
C
      DO 430 J = JXI+1, JR-1
         DBSTA(J) =  - (1.D0/(2.D0*DX))*(EFSTA(J+1) - EFSTA(J-1))
     +               - DFSTA(J)
 430  CONTINUE
         DBSTA(JR) =  - (1.D0/(DX))*(EFSTA(JR) - EFSTA(JR-1))
     +               - DFSTA(JR)
C
C     Decimate computed energy flux to smooth out small irregularities
C
      COUNT = 0
      DO 440 J = 1,JR
         IF((DFLOAT(COUNT)*XR/10.D0 - XB(J)).LE.DX/2.D0) THEN
           COUNT = COUNT+1
           XBDEC(COUNT) = XB(J)
           EFDEC(COUNT) = EFSTA(J)
         ENDIF
 440  CONTINUE
C
C     Fit a cubic spline using Subroutines SPLINE and SPLINT
C     in Subrs. 10 to decimated energy flux
C
      CALL SPLINE(XBDEC, EFDEC, COUNT, 0.D0, 0.D0, EF2)
      DO 450 J = 1,JR
        CALL SPLINT(XBDEC, EFDEC, EF2, COUNT, XB(J), EFSM(J))
 450  CONTINUE
C
C     Calculate the energy dissipation due to breaking based on the
C     the smoothed energy flux and limit the computed results to
C     region of non-negative dissipation.
C
```

```
      DO 460 J = 2,JR-1
         DBSM(J) =  - (1.D0/(2.D0*DX))*(EFSM(J+1) - EFSM(J-1))
     +              - DFSTA(J)
 460  CONTINUE
         DBSM(1) = DBSTA(1)
         DBSM(JR) = -(1.D0/DX)*(EFSM(JR) - EFSM(JR-1))
     +              - DFSTA(JR)
      DO 470 J = JXI+1,JR
         IF(DBSM(J).LT.0.D0) THEN
            JR = J-1
            XR = XB(JR)
            ZR = ZB(JR)
            GO TO 480
         ENDIF
 470  CONTINUE
C
C     Subr. 11 OUTPUT stores input and computed results
C
 480  CALL OUTPUT(JXI)
C
      END
C -00------------------  END OF MAIN PROGRAM  ----------------------
C #01####################  SUBROUTINE OPENER  ########################
C
C     This subroutine opens all input and output files
C
      SUBROUTINE OPENER(FINMIN)
C
      IMPLICIT NONE
      CHARACTER*10 FINMIN
C
      OPEN (UNIT=11,FILE=FINMIN,STATUS='OLD',ACCESS='SEQUENTIAL')
      OPEN (UNIT=21,FILE='ODOC',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN (UNIT=22,FILE='OSETUP',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN (UNIT=23,FILE='OWAVEHT',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN (UNIT=24,FILE='OSKEW',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN (UNIT=25,FILE='OCURTO',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN (UNIT=26,FILE='ODISSIP',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN (UNIT=27,FILE='ONONLIN',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN (UNIT=28,FILE='OITER',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN (UNIT=29,FILE='OMOMENT',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN (UNIT=30,FILE='OENERGY',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN (UNIT=31,FILE='OENERSM',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
      OPEN (UNIT=40,FILE='OMESSG',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
C
      RETURN
      END
C
C -01-----------------  END OF SUBROUTINE OPENER  -------------------
C #02####################  SUBROUTINE INPUT  ########################
C
C     This subroutine reads data from primary input data file
C
      SUBROUTINE INPUT
C
      IMPLICIT NONE
      INTEGER NN, NB
```

```
      PARAMETER (NN=2000, NB=100)
      DOUBLE PRECISION TP, FP, WKPO
      DOUBLE PRECISION HRMS, SIGMA, H, WSETUP, SIGSTA
      DOUBLE PRECISION XBINP, ZBINP, FBINP
      INTEGER NLINES, JSWL, NBINP, J, I
      CHARACTER*5 COMMEN(14)
C
      COMMON /PERIOD/ TP, FP, WKPO
      COMMON /PREDIC/ HRMS(NN), SIGMA(NN), H(NN), WSETUP(NN), SIGSTA(NN)
      COMMON /BINPUT/ XBINP(NB), ZBINP(NB), FBINP(NB), NBINP, JSWL
C
C ..... COMMENT LINES
C           NLINES = number of comment lines preceding input data
      READ (11,1110) NLINES
C
      DO 110 I = 1,NLINES
        READ  (11,1120) (COMMEN(J),J=1,14)
        WRITE (21,1120) (COMMEN(J),J=1,14)
        WRITE (*,*) (COMMEN(J),J=1,14)
 110  CONTINUE
C
C ..... INPUT WAVE PROPERTIES
C           TP        = spectral peak period in seconds
C           HRMS(1)   = root mean square wave height at seaward
C                       boundary in meters
C           WSETUP(1) = wave setup at seaward boundary in meters
C
      READ (11,1150) TP, HRMS(1), WSETUP(1)
      FP = 1.D0/TP
C
C ..... COMPUTATIONAL INPUT DATA
C           JSWL = number of spatial nodes along the bottom below
C                  SWL used to determine nodal spacing DX for
C                  given bottom geometry.
C           Note :  JSWL should be so large that delta x between two
C                   adjacent nodes is sufficiently small.
C
      READ (11,1110) JSWL
C
C ..... BOTTOM GEOMETRY
C           The bottom geometry is divided into segments of
C           different inclination and roughness starting from
C           seaward boundary.
C           NBINP     = number of input bottom points
C           XBINP(J)  = horizontal distance to input bottom point (J)
C                       in meters where XBINP(1) = 0 at the
C                       seaward boundary
C           ZBINP(J)  = dimensional vertical coordinate (+ above SWL)
C                       of input bottom point (J) in meters
C           FBINP(J)  = bottom friction factor for segment between
C                       points (J) and (J+1)
C

      READ (11,1110) NBINP
      IF(NBINP.GT.NB) THEN
         WRITE(*,2900) NBINP, NB
         WRITE(40,2900) NBINP, NB
```

```
         STOP
      ENDIF
 2900 FORMAT(/'Number of Input Bottom Nodes NBINP = ',I8,' ;NB = ',I8/
     +       'Increase PARAMETER NB.')
C
C     Point J = 1 has no corresponding friction factor.
      READ (11,1150) XBINP(1), ZBINP(1)
      XBINP(1) = 0.D0
      DO 140 J = 2,NBINP
         READ (11,1150) XBINP(J), ZBINP(J), FBINP(J-1)
  140 CONTINUE
C
      CLOSE (11)
 1110 FORMAT (I8)
 1120 FORMAT (14A5)
 1150 FORMAT (3D13.6)
C
      END
C
C -02-----------------  END OF SUBROUTINE INPUT  --------------------
C #03####################  SUBROUTINE BOTTOM  ########################
C
C     This subroutine calculates the bottom geometry and
C     DX between two adjacent nodes
C
      SUBROUTINE BOTTOM
C
      IMPLICIT NONE
      INTEGER NN, NB
      PARAMETER (NN=2000, NB=100)
      DOUBLE PRECISION SLOPE(NB)
      DOUBLE PRECISION XBINP, ZBINP, FBINP
      DOUBLE PRECISION DX, XB, ZB, FB2, DZBDX
      DOUBLE PRECISION GAMMAS, BETA, XS, XI
      DOUBLE PRECISION CROSS, DUM, DIST, XCUM
      INTEGER JSWL, NBINP, K, J, JMAX
C
      COMMON /BINPUT/ XBINP(NB), ZBINP(NB), FBINP(NB), NBINP, JSWL
      COMMON /BPROFL/ DX, XB(NN), ZB(NN), FB2(NN), DZBDX(NN), JMAX
      COMMON /BRKNEW/ GAMMAS, BETA, XS, XI
C
C     XS =  dimensional horizontal distance between
C             seaward boundary and initial shoreline at SWL
C
C     The structure geometry is divided into segments of different
C     inclination and roughness.
C       NBINP = number of input bottom points
C     For segments starting from the seaward boundary:
C       SLOPE(K)  = slope of segment K(+ upslope, - downslope)
C       FBINP(K)  = bottom friction factor
C       XBINP(K)  = dimensional horizontal distance from seaward boundary
C                   to the seaward-end of segment K
C       ZBINP(K)  = dimensional vertical coordinate (+ above SWL)
C                   at the seaward-end of segment K
C
        DO 120 K = 1,NBINP-1
          SLOPE(K) = (ZBINP(K+1)-ZBINP(K))/(XBINP(K+1)-XBINP(K))
```

```
 120    CONTINUE
C
C ... CALCULATE GRID SPACING DX BETWEEN TWO ADJACENT NODES
C
C     The value of JSWL specified as input corresponds to
C     number of nodes along the bottom below SWL.
C
        K = 0
  900   CONTINUE
         IF (K.EQ.NBINP) THEN
          WRITE(*,2900)
          WRITE(40,2900)
          STOP
         ENDIF
        K = K+1
        CROSS = ZBINP(K)*ZBINP(K+1)
        IF (CROSS.GT.0.D0) GOTO 900
        XS  = XBINP(K+1) - ZBINP(K+1)/SLOPE(K)
        DX    = XS/DFLOAT(JSWL)
 2900   FORMAT(/'Bottom is always below SWL.'/
     +          'There is no still water shoreline.')

C
C ... CALCULATE BOTTOM GEOMETRY AT EACH NODE
C
C     JMAX = landward edge node corresponding to maximum node number
C     XB(J)= horizontal coordinate of node j where XB(1) = 0
C     ZB(J)= vertical coordinate of bottom at node j (+ above SWL)
C     SLOPE(K) = tangent of local slope of segment K
C
        DUM = XBINP(NBINP)/DX
        JMAX  = INT(DUM)+1
      IF (JMAX.GT.NN) THEN
        WRITE  (*,2910) JMAX,NN
        WRITE (40,2910) JMAX,NN
        STOP
      ENDIF
 2910 FORMAT (/' End Node =',I8,'; NN =',I8/
     +         ' Bottom length is too long.'/
     +         ' Cut it, or change PARAMETER NN.')
C
      DIST = -DX
      K    = 1
      XCUM = XBINP(K+1)
      DO 140 J = 1,JMAX
        DIST = DIST + DX
        IF (DIST.GT.XCUM.AND.K.LT.NBINP) THEN
          K    = K+1
          XCUM = XBINP(K+1)
        ENDIF
        ZB(J) = ZBINP(K) + (DIST-XBINP(K))*SLOPE(K)
        XB(J) = DIST
        DZBDX(J) = SLOPE(K)
        FB2(J) = 1.D0/2.D0*FBINP(K)
  140 CONTINUE
C
C     Set XI = XS until XI is found in Main Program.
```

```
      XI = XS
C
      RETURN
      END
C
C -03---------------  END OF SUBROUTINE BOTTOM  --------------------
C #04#####################  SUBROUTINE PARAM  ########################
C
C     This subroutine calculates parameters used in other subroutines
C
      SUBROUTINE PARAM
C
      IMPLICIT NONE
      DOUBLE PRECISION TP, FP, WKPO
      DOUBLE PRECISION GRAV, SQR8, PI, TWOPI
      COMMON /PERIOD/ TP, FP, WKPO
      COMMON /CONSTA/ GRAV, SQR8, PI, TWOPI
C
C
C ... CONSTANTS and PARAMETER
C
C     PI      = 3.14159
C     TWOPI   = 2.D0 * PI
C     GRAV    = acceleration due to gravity
C     SQR8    = Sqrt(8)
C     WKPO    = deep water wave number for the peak period
C
      PI = 3.14159D0
      TWOPI = 2.D0*PI
      GRAV = 9.81D0
      SQR8 = DSQRT(8.D0)
      WKPO = (TWOPI)**2.D0/(GRAV*TP**2.D0)
C
      RETURN
      END
C
C -04----------------  END OF SUBROUTINE PARAM  --------------------
C #05#####################  SUBROUTINE LWAVE  ########################
C
C     This subroutine calculates quantities based on linear wave theory
C
      SUBROUTINE LWAVE(J, WD)
C
      IMPLICIT NONE
      INTEGER NN
      PARAMETER (NN=2000)
      DOUBLE PRECISION X, XNEW, D, WD, COTH
      INTEGER J
      DOUBLE PRECISION TP, FP, WKPO
      DOUBLE PRECISION GRAV, SQR8, PI, TWOPI
      DOUBLE PRECISION WKP, CP, WN
      COMMON /PERIOD/ TP, FP, WKPO
      COMMON /CONSTA/ GRAV, SQR8, PI, TWOPI
      COMMON /LINEAR/ WKP, CP, WN(NN)
C
C
C ... LINEAR WAVE PARAMETERS
```

```
C
C       TP  = peak period specified as input
C       FP  = peak frequency
C       CP  = phase velocity of peak frequency
C       WN  = ratio of group velocity to phase velocity
C
      D = WD*WKPO
      IF(J.EQ.1) THEN
        X = D/DSQRT(DTANH(D))
      ELSE
        X = WKP*WD
      ENDIF
 10   COTH = 1.D0/DTANH(X)
      XNEW = X - (X-D*COTH)/(1.D0+D*(COTH**2.D0-1.D0))
      IF (DABS(XNEW - X).GT.1.D-7) THEN
        X = XNEW
        GOTO 10
      ENDIF
      WKP = X/WD
      WN(J) = 0.5D0*(1.D0 + 2.D0*X/(DSINH(2.D0*X)))
      CP = TWOPI/WKP/TP
C
      RETURN
      END
C
C -05-----------------  END OF SUBROUTINE  LWAVE ---------------------
C #06######################  SUBROUTINE SKEWKU  ##########################
C
C     This subroutine calculates skewness and kurtosis for given HSTA
C
      SUBROUTINE SKEWKU(J,HSTA)
C
      IMPLICIT NONE
      INTEGER NN
      PARAMETER (NN=2000)
      DOUBLE PRECISION SKEW, CURTO, HSTA
      DOUBLE PRECISION A, B, C, Y1, Y2
      INTEGER J
C
      COMMON /NONLIN/ SKEW(NN), CURTO(NN)
      DATA A, B, C, Y1, Y2 / 2.D0, 1.D0, 0.7D0, 0.5D0, 1.D0/
C
C
C ... SKEWNESS AND KURTOSIS OF THE FREE SURFACE
C
C     HSTA = ratio of root-mean-square wave height to mean water depth
C     SKEW(J)  = skewness of the free surface
C     CURTO(J) = kurtosis of the free surface
C
      IF(HSTA.LE.Y1) THEN
        SKEW(J) = A*HSTA
      ELSEIF(HSTA.LE.Y2) THEN
        SKEW(J) = A*Y1 - B*(HSTA - Y1)
      ELSE
        SKEW(J) = A*Y1 - B*(Y2 - Y1) + C*(HSTA - Y2)
      ENDIF
      CURTO(J) = 3.D0 + SKEW(J)**2.2D0
```

```
C
      RETURN
      END
C
C -06-----------------  END OF SUBROUTINE SKEWKU  ---------------------
C #07#####################  SUBROUTINE CSFFSE  ########################
C
C     This subroutine computes CS, CF, FS, and FE
C
      SUBROUTINE CSFFSE(J,SSTA)
      IMPLICIT NONE
      INTEGER NN
      PARAMETER (NN=2000)
      DOUBLE PRECISION WKP, CP, WN
      DOUBLE PRECISION SKEW, CURTO
      DOUBLE PRECISION CS, FS, SXXSTA, TBSTA
      DOUBLE PRECISION CF, FE, EFSTA, DFSTA, EFSM
      DOUBLE PRECISION SSTA
      INTEGER J
C
      COMMON /LINEAR/ WKP, CP, WN(NN)
      COMMON /NONLIN/ SKEW(NN), CURTO(NN)
      COMMON /MOMENT/ CS(NN), FS, SXXSTA(NN), TBSTA(NN)
      COMMON /ENERGY/ CF(NN), FE, EFSTA(NN), DFSTA(NN), EFSM(NN)
C
         CS(J) = SSTA*SKEW(J) - SSTA**2.D0
         CF(J) = 1.5D0*SKEW(J)*SSTA*(1.D0 - SSTA**2.D0) +
     +           0.5D0*SSTA**2.D0*(CURTO(J) - 5.D0) + SSTA**4.D0
         FS = (2.D0*WN(J) - 0.5D0) + CS(J)
         FE = WN(J)*CP*(1.D0 + CF(J))
C

         RETURN
         END
C
C -07-----------------  END OF SUBROUTINE CSFFSE  ---------------------
C #08#####################  SUBROUTINE GBANGF  ########################
C
C     This subroutine calculates GB and GF related to bottom friction
C
      SUBROUTINE GBANGF(J)
C
      IMPLICIT NONE
      INTEGER NN
      PARAMETER (NN=2000)
      DOUBLE PRECISION HRMS, SIGMA, H, WSETUP, SIGSTA
      DOUBLE PRECISION GRAV, SQR8, PI, TWOPI
      DOUBLE PRECISION SKEW, CURTO
      DOUBLE PRECISION GB, GF
      DOUBLE PRECISION IB, IF
      DOUBLE PRECISION A, AA, XX, G1, G2, G3, G4
      DOUBLE PRECISION GAMMLN, DIGAMM, TRIGAM, TETRAG, PENTAG
      DOUBLE PRECISION GAM, DIM, TRI, TET, PEN
      DOUBLE PRECISION C1, C2, C3, F, F1, ERROR, DELTA
      DOUBLE PRECISION ONE, TWO, SMAX, SMIN, ERFCC
      INTEGER J, NUM, N, I
C
```

```
      COMMON /PREDIC/ HRMS(NN), SIGMA(NN), H(NN), WSETUP(NN), SIGSTA(NN)
      COMMON /CONSTA/ GRAV, SQR8, PI, TWOPI
      COMMON /NONLIN/ SKEW(NN), CURTO(NN)
      COMMON /FRICTN/ GB(NN), GF(NN)
      DATA DELTA, NUM /1.D-6, 31/
      DATA ONE, TWO, SMAX, SMIN /1.0D0, 2.0D0, 1.99D0, 0.15D0/
C
C
C     GB(J)  = GB at node J
C     GF(J)  = GF at node J
C
C
        IF(SKEW(J).GE.SMAX) THEN
C
C ...EXPONENTIAL DISTRIBUTION
C
C     Assume SIGSTA(J) = ONE
C            SKEW(J) = TWO
C
      IB = TWO*DEXP(-ONE - ONE)
      IF = 6.D0*DEXP(-ONE - ONE)
      GB(J) = -(ONE + ONE**TWO) + TWO*IB
      GF(J) = 3.D0*ONE + ONE**3.D0 - TWO + TWO*IF
C
C
        ELSEIF(SKEW(J).GT.SMIN) THEN
C
C ...EXPONENTIAL GAMMA DISTRIBUTION
C
C     Determine 'a' parameter based on approximate relation between s and a
      IF(SKEW(J).GE.1.D0) THEN
        A = 1.16D0*(2.D0 - SKEW(J))
      ELSE
        A = 0.5D0 + 1.D0/SKEW(J)**2.D0
      ENDIF
C
 300      G1 = 0.D0
          G2 = 0.D0
          G3 = 0.D0
          G4 = 0.D0
C_____________________________________________________________________
C
          IF (A.LE.1.D0) THEN
            GAM = DEXP(GAMMLN(A + 1.D0) - DLOG(A))
          ELSE
            GAM = DEXP(GAMMLN(A))
          ENDIF
C_____________________________________________________________________
C
          IF (A.GE.10.D0) THEN
             XX = A
          ELSE
             N = IDINT(A)
             XX = 10.D0 + (A - DFLOAT(N))
             DO 30 I = 1,(10-N)
               AA = XX - FLOAT(I)
               G1 = G1 - 1.D0/AA
```

```
              G2 = G2 + 1.D0/AA**2.D0
              G3 = G3 - 2.D0/AA**3.D0
              G4 = G4 + 6.D0/AA**4.D0
 30         CONTINUE
          ENDIF
C_____________________________________________________________________
C
          DIM = G1 + DIGAMM(XX)
          TRI = G2 + TRIGAM(XX)
          TET = G3 + TETRAG(XX)
          PEN = G4 + PENTAG(XX)
C
          F = DLOG(-TET) - 1.5D0*DLOG(TRI)
          F1 = PEN/TET - 1.5D0*TET/TRI
          A = A + (DLOG(SKEW(J)) - F )/F1
          ERROR = DABS(DLOG(SKEW(J)) - F)
          IF(ERROR.GT.DELTA) GOTO 300
C
          C1 = DSQRT(TRI)/GAM
          C2 = DSQRT(TRI)
          C3 = DIM
C
          CALL INTGRL(SIGSTA(J),SKEW(J),A,C1,C2,C3,NUM,IB,IF)
C
          GB(J) = -(ONE+SIGSTA(J)**TWO) + TWO*IB
          GF(J) = 3.D0*SIGSTA(J) + SIGSTA(J)**3.D0 - SKEW(J) + TWO*IF
        ELSE
C
C ...GAUSSIAN DISTRIBUTION
C
        C1 = ERFCC(SIGSTA(J)/DSQRT(TWO))
        C2 = DEXP(-SIGSTA(J)**TWO/TWO)
        IB = 0.5D0*(ONE+SIGSTA(J)**TWO)*C1 -
     +       SIGSTA(J)/DSQRT(TWOPI)*C2
        IF=(TWO+SIGSTA(J)**TWO)/DSQRT(TWOPI)*C2 -
     +   SIGSTA(J)/TWO*(3.D0+SIGSTA(J)**TWO)*C1
          GB(J) = -(ONE+SIGSTA(J)**TWO) + TWO*IB
          GF(J) = 3.D0*SIGSTA(J) + SIGSTA(J)**3.D0 - SKEW(J) + TWO*IF

        ENDIF
C
C
      RETURN
      END
C
C*****************************************************************************
      SUBROUTINE INTGRL(SIGSTAR,SKEW,A,C1,C2,C3,NUM,IB,IF)
      IMPLICIT NONE
      INTEGER NP
      PARAMETER (NP = 1000)
      DOUBLE PRECISION SIGSTAR, SKEW, A, C1, C2, C3, IB, IF
      DOUBLE PRECISION UPPER, DFS, ETAST, Y
       DOUBLE PRECISION FIB(NP), FIF(NP), SEIB, SOIB, SEIF, SOIF
      INTEGER I, NUM
      UPPER = 5.D0 + 4.D0 * SKEW
      DFS = (UPPER - SIGSTAR)/FLOAT((NUM-1))
      DO 100 I = 1,NUM
```

```
       ETAST = SIGSTAR + DFS*FLOAT((I-1))
       Y = C2*ETAST - C3
       FIB(I) = (ETAST - SIGSTAR)**2.D0*C1*DEXP(-A*Y - DEXP(-Y))
       FIF(I) = (ETAST - SIGSTAR)**3.D0*C1*DEXP(-A*Y - DEXP(-Y))
 100   CONTINUE
       SEIB = FIB(2)
       SOIB = 0.D0
       SEIF = FIF(2)
       SOIF = 0.D0
      DO 200 I = 2,(NUM-1)/2
       SEIB = SEIB + FIB(I*2)
       SOIB = SOIB + FIB(I*2 - 1)
       SEIF = SEIF + FIF(I*2)
       SOIF = SOIF + FIF(I*2 - 1)
 200   CONTINUE
      IB = DFS/3.D0* (FIB(1) + 4.D0*SEIB + 2.D0*SOIB + FIB(NUM))
      IF = DFS/3.D0* (FIF(1) + 4.D0*SEIF + 2.D0*SOIF + FIF(NUM))
      RETURN
      END
C*********************************************************************
      FUNCTION GAMMLN(XX)
      DOUBLE PRECISION COF(6),STP,HALF,ONE,FPF,X,TMP,SER,XX,GAMMLN
      DATA COF,STP /76.18009173D0,-86.50532033D0,24.01409822D0,
     +   -1.231739516D0,.120858003D-2,-.536382D-5,2.50662827465D0/
      DATA HALF,ONE,FPF/0.5D0,1.0D0,5.5D0/
        X = XX - ONE
        TMP = X + FPF
        TMP = (X+HALF)*DLOG(TMP) - TMP
        SER = ONE
        DO 11 J = 1,6
          X = X + ONE
          SER = SER + COF(J)/X
 11     CONTINUE
        GAMMLN = TMP + DLOG(STP*SER)
      RETURN
      END
C*********************************************************************
      FUNCTION DIGAMM(XX)
      DOUBLE PRECISION XX, ONE, TWO, ONEXX, OXXTWO, DIGAMM
      DATA ONE,TWO/1.D0,2.D0/
      ONEXX = ONE/XX
      OXXTWO = ONE/XX**TWO
      DIGAMM = DLOG(XX) - ONEXX*(ONE/TWO + ONEXX*(ONE/12.D0 -
     +     OXXTWO*(ONE/120.D0 - OXXTWO*(ONE/252.D0 -
     +     OXXTWO* (ONE/240.D0 - OXXTWO/132.D0)))))
      RETURN
      END
C*********************************************************************
      FUNCTION TRIGAM(XX)
      DOUBLE PRECISION XX, ONE, TWO, ONEXX, OXXTWO, TRIGAM
      DATA ONE,TWO/1.D0,2.D0/
      ONEXX = ONE/XX
      OXXTWO = ONE/XX**TWO
      TRIGAM = ONEXX*(ONE + ONEXX*(ONE/TWO + ONEXX*(ONE/6.D0 -
     +     OXXTWO*(ONE/30.D0 - OXXTWO*(ONE/42.D0 -
     +     OXXTWO*(ONE/30.D0 - OXXTWO*5.D0/66.D0))))))
      RETURN
```

```
      END
C*********************************************************************
      FUNCTION TETRAG(XX)
      DOUBLE PRECISION XX, ONE, TWO, ONEXX, OXXTWO, TETRAG
      DATA ONE,TWO/1.D0,2.D0/
      ONEXX = ONE/XX
      OXXTWO = ONE/XX**TWO
      TETRAG = -OXXTWO*(ONE + ONEXX*(ONE + ONEXX*(ONE/TWO -
     +     OXXTWO*(ONE/6.D0 - OXXTWO*(ONE/6.D0 -
     +     OXXTWO*(3.D0/10.D0 - OXXTWO*5.D0/6.D0))))))
      RETURN
      END
C*********************************************************************
      FUNCTION PENTAG(XX)
      DOUBLE PRECISION XX, ONE, TWO, THR, ONEXX, OXXTWO, PENTAG
      DATA ONE,TWO,THR/1.D0,2.D0,3.D0/
      ONEXX = ONE/XX
      OXXTWO = ONE/XX**TWO
      PENTAG = ONE/XX**THR*(TWO + ONEXX*(THR + ONEXX*(TWO -
     +     OXXTWO*(ONE - OXXTWO*(4.D0/THR -
     +     OXXTWO*(THR - OXXTWO*10.D0))))))
      RETURN
      END
C*********************************************************************
      FUNCTION ERFCC(X)
      DOUBLE PRECISION X, Z, T, ERFCC
      Z=DABS(X)
      T=1.D0/(1.D0+0.5D0*Z)
      ERFCC=T*DEXP(-Z*Z-1.26551223D0+T*(1.00002368D0+T*(.37409196D0+
     * T*(.09678418D0+T*(-.18628806D0+T*(.27886807D0+T*(-1.13520398D0+
     * T*(1.48851587D0+T*(-.82215223D0+T*.17087277D0)))))))))
      IF (X.LT.0.D0) ERFCC=2.D0-ERFCC
      RETURN
      END
C*********************************************************************
C -08----------------  END OF SUBROUTINE GBANGF  --------------------
C #09#####################  SUBROUTINE DBREAK  ########################
C
C     This subroutine calculates QBREAK and DBSTA for wave breaking in
C     region of XB < XI
C
      SUBROUTINE DBREAK(J, WHRMS, D)
C
      IMPLICIT NONE
      INTEGER NN, NB
      PARAMETER (NN=2000, NB=100)
      DOUBLE PRECISION WHRMS, D, HM, B, QBOLD, S0
      DOUBLE PRECISION TP, FP, WKPO
      DOUBLE PRECISION WKP, CP, WN
      DOUBLE PRECISION ALPHA, GAMMA,  QBREAK, DBSTA, DBSM
      DOUBLE PRECISION GRAV, SQR8, PI, TWOPI
      INTEGER J
      COMMON /PERIOD/ TP, FP, WKPO
      COMMON /LINEAR/ WKP, CP, WN(NN)
      COMMON /WBREAK/ ALPHA, GAMMA, QBREAK(NN), DBSTA(NN), DBSM(NN)
      COMMON /CONSTA/ GRAV, SQR8, PI, TWOPI
      DATA ALPHA/1.D0/
```

```
C
      IF(J.EQ.1) THEN
        S0 = WHRMS*TWOPI/(GRAV*TP**2.D0) *
     +      DSQRT(DTANH(WKP*D)*(1.D0 + 2.D0*WKP*D/DSINH(2.D0*WKP*D)))
        GAMMA = 0.5D0+0.4D0*DTANH(33.D0*S0)
      ENDIF
C
C ... FRACTION OF BREAKING WAVES AND ASSOCIATED DISSIPATION
C
C     QBREAK(J) = Fraction of breaking waves at node J
C     DBSTA(J)  = Time averaged normalized energy dissipation due to wave
C                 breaking at node J
C
      HM = 0.88D0/WKP*DTANH(GAMMA*WKP*D/0.88D0)
C
        B = (WHRMS/HM)**2.D0
      IF(B.LT.0.99999D0) THEN
        QBOLD = B/2.D0
10      QBREAK(J) = QBOLD - (1.D0-QBOLD + B*DLOG(QBOLD))/(B/QBOLD-1.D0)
        IF(QBREAK(J).LE.0.D0) QBREAK(J) = QBOLD/2.D0
        IF(DABS(QBREAK(J)-QBOLD).GT.1D-6) THEN
          QBOLD = QBREAK(J)
          GOTO 10
        ENDIF
      ELSE
        QBREAK(J) = 1.D0
      ENDIF
C
      DBSTA(J) = 0.25D0*ALPHA*QBREAK(J)*FP*HM**2.D0
C
      RETURN
      END
C
C -09-----------------  END OF SUBROUTINE DBREAK  ---------------------
C #10################  SUBROUTINES SPLINE and SPLINT ###################
C
C     These subroutines for cubic spline interpolation are double
C     precision versions of those listed in Numerical Recipies
C
      SUBROUTINE SPLINE(X,Y,N,YP1,YPN,Y2)
C
      IMPLICIT NONE
      INTEGER N, NMAX
      PARAMETER (NMAX=100)
      DOUBLE PRECISION YP1, YPN, X(N), Y(N), Y2(N), U(NMAX)
      DOUBLE PRECISION SIG, P, QN, UN
      INTEGER K, I
      IF (YP1.GT..99D30) THEN
        Y2(1)=0.D0
        U(1)=0.D0
      ELSE
        Y2(1)=-0.5D0
        U(1)=(3.D0/(X(2)-X(1)))*((Y(2)-Y(1))/(X(2)-X(1))-YP1)
      ENDIF
      DO 110 I=2,N-1
        SIG=(X(I)-X(I-1))/(X(I+1)-X(I-1))
        P=SIG*Y2(I-1)+2.D0
```

```
        Y2(I)=(SIG-1.D0)/P
        U(I)=(6.D0*((Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))
     *       /(X(I)-X(I-1)))/(X(I+1)-X(I-1))-SIG*U(I-1))/P
 110    CONTINUE
        IF (YPN.GT..99D30) THEN
          QN=0.D0
          UN=0.D0
        ELSE
          QN=0.5D0
          UN=(3.D0/(X(N)-X(N-1)))*(YPN-(Y(N)-Y(N-1))/(X(N)-X(N-1)))
        ENDIF
        Y2(N)=(UN-QN*U(N-1))/(QN*Y2(N-1)+1.D0)
        DO 120 K=N-1,1,-1
          Y2(K)=Y2(K)*Y2(K+1)+U(K)
 120    CONTINUE
        RETURN
        END
C*********************************************************************
        SUBROUTINE SPLINT(XA,YA,Y2A,N,X,Y)
        IMPLICIT NONE
        INTEGER N, K, KHI, KLO
        DOUBLE PRECISION XA(N),YA(N),Y2A(N)
        DOUBLE PRECISION  X, Y, H, A, B
        KLO=1
        KHI=N
 110    IF (KHI-KLO.GT.1) THEN
          K=(KHI+KLO)/2
          IF(XA(K).GT.X)THEN
            KHI=K
          ELSE
            KLO=K
          ENDIF
        GOTO 110
        ENDIF
        H=XA(KHI)-XA(KLO)
        IF (H.EQ.0.D0) PAUSE 'Bad XA input.'
        A=(XA(KHI)-X)/H
        B=(X-XA(KLO))/H
        Y=A*YA(KLO)+B*YA(KHI)+
     +       ((A**3-A)*Y2A(KLO)+(B**3-B)*Y2A(KHI))*(H**2)/6.D0
        RETURN
        END
C -10-----------  END OF SUBROUTINES SPLINE AND SPLINT-----------------
C #11######################  SUBROUTINE OUTPUT  ########################
C
C     This subroutine stores computed and input quantities
C
        SUBROUTINE OUTPUT(JXI)
C
        IMPLICIT NONE
        INTEGER NN, NB
        PARAMETER (NN=2000, NB=100)
        DOUBLE PRECISION TP, FP, WKPO
        DOUBLE PRECISION HRMS, SIGMA, H, WSETUP, SIGSTA
        DOUBLE PRECISION XBINP, ZBINP, FBINP
        DOUBLE PRECISION DX, XB, ZB, FB2, DZBDX
        DOUBLE PRECISION WKP, CP, WN
```

```
      DOUBLE PRECISION SKEW, CURTO
      DOUBLE PRECISION GB, GF
      DOUBLE PRECISION ALPHA, GAMMA, QBREAK, DBSTA, DBSM
      DOUBLE PRECISION GAMMAS, BETA, XS, XI
      DOUBLE PRECISION CS, FS, SXXSTA, TBSTA
      DOUBLE PRECISION CF, FE, EFSTA, DFSTA, EFSM
      DOUBLE PRECISION EPS1
      DOUBLE PRECISION XR, ZR
      INTEGER MAXITE, NUMITE
      INTEGER J, JSWL, NBINP, JMAX, JR, JXI
C
      COMMON /PERIOD/ TP, FP, WKPO
      COMMON /PREDIC/ HRMS(NN), SIGMA(NN), H(NN), WSETUP(NN), SIGSTA(NN)
      COMMON /BINPUT/ XBINP(NB), ZBINP(NB), FBINP(NB), NBINP, JSWL
      COMMON /BPROFL/ DX, XB(NN), ZB(NN), FB2(NN), DZBDX(NN),  JMAX
      COMMON /LINEAR/ WKP, CP, WN(NN)
      COMMON /NONLIN/ SKEW(NN), CURTO(NN)
      COMMON /FRICTN/ GB(NN), GF(NN)
      COMMON /WBREAK/ ALPHA, GAMMA, QBREAK(NN), DBSTA(NN), DBSM(NN)
      COMMON /BRKNEW/ GAMMAS, BETA, XS, XI
      COMMON /MOMENT/ CS(NN), FS, SXXSTA(NN), TBSTA(NN)
      COMMON /ENERGY/ CF(NN), FE, EFSTA(NN), DFSTA(NN), EFSM(NN)
      COMMON /ITERAT/ EPS1, MAXITE, NUMITE(NN)
      COMMON /RUNUP/  XR, ZR, JR
C
C
C
C ......... INPUT WAVE PROPERTIES
C           TP        = spectral peak period in seconds
C           HRMS(1)   = root mean square wave height at seaward
C                       boundary in meters
C           WSETUP(1) = wave setup at seaward boundary in meters
C
      WRITE (21,1000) TP, FP, HRMS(1), WSETUP(1)
 1000 FORMAT (/ 'INPUT WAVE PROPERTIES:'/
     +          'Peak wave period (sec)              =',E13.6/
     +          'Peak frequency (1/sec)              =',E13.6/
     +          'Root-mean-square wave height '/
     +          '           at seaward boundary (m) =',E13.6/
     +          'Wave setup at seaward boundary (m) =',E13.6)
C
C ......... OUTPUT BOTTOM GEOMETRY
C           The bottom geometry is divided into segments of
C           different inclination and roughness starting from
C           seaward boundary.
C           NBINP     = number of segments
C           XBINP(J)  = horizontal distance from seaward boundary
C                       to landward-end of segment (J-1) in meters
C           ZBINP(J)  = dimensional vertical coordinate (+ above SWL)
C                       of the landward end of segment (J-1) in meters
C           FBINP(J) = bottom friction factor
      WRITE (21,1100)  0.D0-ZBINP(1), NBINP-1, JSWL, DX, JMAX
C
 1100 FORMAT (/'INPUT BOTTOM GEOMETRY'/
     +          'Depth at seaward boundary (m)       =',F13.6/
     +          'Number of linear segments           =',I8/
     +          'Number of spatial nodes  below'/
```

```
     +          '   SWL used to find DX              =',I8/
     +          'Node spaceing, DX (m)               =',F13.6/
     +          'Maximum landward node          JMAX =',I8//
     +          '       X (m)            Zb (m)          Friction factor')
      WRITE (21,1200) XBINP(1), ZBINP(1)
      DO 140 J = 2,NBINP
        WRITE (21,1200) XBINP(J), ZBINP(J), FBINP(J-1)
 140  CONTINUE
C
 1200 FORMAT(3(F13.6,5X))
C
C..... EMPIRICAL PARAMETERS FOR WAVE BREAKING
C
      WRITE (21,1300) ALPHA, GAMMA, GAMMAS, BETA, XS, XI
C
 1300 FORMAT(/'EMPIRICAL PARAMETERS FOR WAVE BREAKING'/
     +        'Alpha  = ',F13.6/
     +        'Gamma  = ',F13.6/
     +        'Gammas = ',F13.6/
     +        'Beta   = ',F13.6/
     +        'Xs(m)  = ',F13.6/
     +        'Xi(m)  = ',F13.6)
C
C..... ITERATION PARAMETERS
C
      WRITE (21,1400) EPS1, MAXITE
C
 1400 FORMAT(/'ITERATION PARAMETERS'/
     +    'Allowable relative error in iterated depth(m)  =',F13.6/
     +    'Maximum iterations allowed =  ',I8)
C
C.....WAVE RUNUP OR COMPUTATION LIMIT
C
      WRITE(21,1450) JR, JXI, JMAX, XR, ZR, H(JR)
C
 1450 FORMAT(/'WAVE RUNUP OR COMPUTATION LIMIT'/
     +    'Most landward node of computation          JR =',I8/
     +    'Most landward node in outer zone         JXI =',I8/
     +    '              in comparision with       JMAX =',I8/
     +    'X-coordinate of JR (m)                    XR =',F13.6/
     +    'Z-coordinate of JR (m)                    ZR =',F13.6/
     +    'Mean water depth of this node (m)      H(JR) =',F13.6/)
C
C..... WAVE PROPERTIES AT EACH NODE
C
c      OPEN (UNIT=11,FILE=FINMIN,STATUS='OLD',ACCESS='SEQUENTIAL')
c      OPEN (UNIT=21,FILE='ODOC',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
c      OPEN (UNIT=22,FILE='OSETUP',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
c      OPEN (UNIT=23,FILE='OWAVEHT',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
c      OPEN (UNIT=24,FILE='OSKEW',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
c      OPEN (UNIT=25,FILE='OCURTO',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
c      OPEN (UNIT=26,FILE='ODISSIP',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
c      OPEN (UNIT=27,FILE='ONONLIN',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
c      OPEN (UNIT=28,FILE='OITER',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
c      OPEN (UNIT=29,FILE='OMOMENT',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
c      OPEN (UNIT=30,FILE='OENERGY',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
c      OPEN (UNIT=31,FILE='OENERSM',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
```

```
c       OPEN (UNIT=40,FILE='OMESSG',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
        DO 160 J = 1, JR
          WRITE(22,1500) XB(J),ZB(J),WSETUP(J), H(J)
          WRITE(23,1500) XB(J),ZB(J),HRMS(J),SIGSTA(J)
          WRITE(24,1500) XB(J),SKEW(J)
          WRITE(25,1500) XB(J),CURTO(J)
          IF(FB2(J).GT.0.D0) THEN
             WRITE(26,1500) XB(J),QBREAK(J),GB(J),GF(J)
          ELSE
             WRITE(26,1500) XB(J),QBREAK(J)
          ENDIF
          WRITE(27,1500) XB(J),WN(J),CS(J),CF(J)
          WRITE(28,1510) XB(J),NUMITE(J)
          WRITE(29,1500) XB(J),SXXSTA(J),TBSTA(J)
          WRITE(30,1500) XB(J),EFSTA(J),DBSTA(J),DFSTA(J)
          WRITE(31,1500) XB(J),EFSM(J),DBSM(J)
 160    CONTINUE
 1500   FORMAT(4F17.9)
 1510   FORMAT(F13.6,I8)
        RETURN
        END
C
C -11-----------------  END OF SUBROUTINE OUTPUT  ----------------------
```