NUMERICAL MODEL FOR RANDOM WAVES ON PERMEABLE COASTAL STRUCTURES

by

Andojo Wurjanto

and

Nobuhisa Kobayashi

RESEARCH REPORT NO. CACR-92-02

March 1992

CENTER FOR APPLIED COASTAL RESEARCH

Department of Civil Engineering
University of Delaware
Newark, Delaware 19716

ABSTRACT

A computer program called PBREAK is presented in this report. PBREAK is an extended version of our previous computer program RBREAK to simulate the flow inside a permeable underlayer of arbitrary thickness as well as the flow above a rough permeable slope of arbitrary geometry. This report explains the computational aspects of the computer program PBREAK together with the input and output for the computation made for each of three model test runs.

PBREAK may be used for the design of rough permeable coastal structures against normally-incident random waves. Any incident wave train can be specified as input by a user at the seaward boundary of the computation domain. PBREAK computes the reflected wave train at the seaward boundary from which the reflected wave spectrum can be computed. PBREAK computes the water-line oscillation on the seaward slope of the structure from which the run-up spectrum and exceedance probability can be computed. In addition to the equations of mass and momentum used to compute the flow field, equations of energy are used in PBREAK to estimate the spatial variations of energy dissipation rates inside the permeable underlayer and above the rough permeable slope. Furthermore, PBREAK computes the hydraulic stability of individual armor units under the action of the computed flow. It should be stated that PBREAK in its present form does not allow wave overtopping and transmission.

Three examples are presented herein and may be used to get familiarized with PBREAK. The limitations and capabilities of PBREAK are discussed in separate reports and papers. Users of PBREAK are strongly recommended to read these publications quoted in this report so that the degree of accuracy of the computed results may be inferred. It is also recommended to calibrate and verify PBREAK if it is to be applied to other problems. It should be emphasized that PBREAK has been compared only with three model test runs so far.

Acknowledgement

This is the fourth and last report resulting from research sponsored by the Coastal Engineering Research Center, U.S. Army Engineer Waterways Experiment Station under Contract No. DACW39-90-K-0006-P001. The authors would like to thank D. L. Ward, J. P. Ahrens and D. D. Davidson for their support and patience during this study to develop numerical models for the design of impermeable and permeable coastal structures against incident random waves.

Contents

FIGURES

FIGURE		PAGE

TABLES

PART I

INTRODUCTION

• 1.1 •

BACKGROUND

Kobayashi and Wurjanto (1989c) developed a computer program called IBREAK, which may be used for the design of rough or smooth impermeable coastal structures of arbitrary geometry against normally-incident monochromatic or transient waves. IBREAK was initially used to simulate irregular waves on the slope of a coastal structure since any incident wave train can be specified as input to IBREAK at the seaward boundary of the computation domain. However, the irregular waterline oscillation on the slope was found to cause numerical difficulties and stoppage during the computation of a sufficient duration for a stationary random sea. The constant time step size Δt for the explicit finite difference method used in IBREAK was reduced to overcome the numerical difficulties. This increased the computation time considerably but did not always work.

Wurjanto and Kobayashi (1991) developed a computer program called RBREAK, which is an improved version of IBREAK with an automated adjustment procedure. The adjustment procedure reduces the time step size Δt for portions of the computation with numerical difficulties. Since the portions with numerical difficulties are not known in advance, the time-marching computation needs to be reversed to an earlier time level before the initiation of the current numerical difficulty and then resumed from the reversed time level using a smaller value of Δt. To reduce the computation time, the value of Δt needs to be increased after overcoming the current numerical difficulty. This automated adjustment procedure has been essential for making successful computations for incident random waves of sufficient durations in an efficient manner. The subroutines required for the spectral and time series analyses for random waves were reported separately by Cox, Kobayashi, and Wurjanto (1991).

Kobayashi, Cox, and Wurjanto (1990) showed that RBREAK could predict the time series and spectral characteristics of irregular wave reflection and run-up on a 1:3 rough impermeable slope. Kobayashi and Wurjanto (1992a) showed that RBREAK could also be used to predict shoreline oscillations on natural beaches, which are normally dominated by low-frequency wave components (Guza and Thornton 1982; Holman and Sallenger 1985). On the other hand, Wise, Kobayashi, and Wurjanto (1991) attempted to predict the net cross-shore sediment transport in the surf and swash zones using the depth-averaged velocities computed by RBREAK. Their attempt has indicated that RBREAK will need to be expanded to account for the vertical variation of the horizontal fluid velocity, if it is to be applied to predict the net cross-shore sediment transport.

The computer program RBREAK is applicable only to impermeable structures and beaches. As a result, it can not be applied to highly permeable structures such as berm breakwaters (e.g., Baird and Hall 1984), reef breakwaters (Ahrens 1989), dynamic revetments (Ahrens 1990) and bermed revetments (Ahrens and Ward 1991). Since viscous effects inside the permeable structures are generally not in similitude between small-scale physical models and prototypes, numerical models may alternatively be calibrated using model tests and applied to simulate the viscous and turbulent flow resistance inside the permeable structure in an approximate manner.

An initial attempt to include the permeability effect in RBREAK was made assuming a thin permeable underlayer. The assumption of the thin permeable underlayer allows one to neglect the region landward of the waterline on the rough slope and the inertia terms in the horizontal momentum equation for the flow inside the thin permeable underlayer. However, this previous numerical model has turned out to be of limited practical use since the permeability effects of the thin permeable underlayer have been found to be minor or negligible. The previous numerical model and computed results based on the assumption of the thin permeable underlayer are summarized in the next section since the computer program PBREAK presented herein should be applicable to a permeable underlayer of arbitrary thickness including the thin permeable underlayer assumed in the following.

• 1.2 •

PREVIOUS WORK FOR THIN PERMEABLE UNDERLAYER

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

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

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

The numerical model based on the assumption of a thin permeable underlayer was found to be inapplicable to three test runs conducted for a 1:3 rough permeable slope with a thick permeable underlayer (Kobayashi, Cox, and Wurjanto 1991). The computed results did not satisfy the time-averaged equation of mass conservation mainly because the previous model did not account for water storage in the region landward of the waterline on the slope. These three test runs corresponded to the three test runs for the 1:3 rough impermeable slope conducted by Kobayashi, Cox, and Wurjanto (1990) except for the presence of the thick permeable underlayer. The improved numerical model for a permeable underlayer with arbitrary thickness has also been compared with the same three test runs of Kobayashi, Cox, and Wurjanto (1991). The compared results presented in Appendices B, C, and D show fairly good agreement.

• 1.3 •

Outline of Report

This report is written in such a way that a user should be able to run PBREAK using this report alone. Part II of this report concisely explains the equations and numerical procedures used in the computer program PBREAK. The derivations of the equations are given separately in the report of Wurjanto and Kobayashi (1992a) so as to reduce the number of equations presented in this report. The previous model of Kobayashi and Wurjanto (1990) based on the assumption of a thin permeable underlayer neglected the region landward of the waterline on the rough slope (Region 3 in Figure 1 in the next section) and the inertia terms in the horizontal momentum equation for the flow in the thin permeable underlayer (Region 2 in Figure 1 in the next section). The computer program PBREAK in its present form does not allow wave overtopping and transmission unlike the previous programs IBREAK and RBREAK. The user may be able to modify the landward boundary condition used in PBREAK to allow wave overtopping or transmission where reference may be made to Kobayashi and Wurjanto (1989a) for wave overtopping and Kobayashi and Wurjanto (1989b) for wave transmission. On the other hand, the hydraulic stability analysis of armor units presented in Part II of this report is the same as that used by Kobayashi and Wurjanto (1990) who neglected the direct effects of the mass and momentum fluxes between the flow above the rough slope and the flow inside the permeable underlayer. This is because the present numerical model with the two flow regions is not expected to predict the local mass and momentum fluxes between the two flow regions very accurately. Moreover, the slope profile change due to armor dislodgement is not analyzed in this report, although this is an important design consideration for berm breakwaters and dynamic revetments.

The computer program PBREAK listed in Appendix A is explained in detail in Part III of this report. PBREAK employs a semi-automated adjustment procedure to overcome numerical difficulties in an efficient manner without making the computer program too long. This procedure combines the user's judgement with the automated adjustment procedure used in RBREAK, which may not be successful always. Sufficient explanation and guidelines are given in the report so that the user will be able to make a successful computation after just a few trials. The input and output data files are explained in detail so that the user will be able to use PBREAK without examining the computer program line by line. The subroutines, parameters and variables in PBREAK are explained in such a way that the user will be able to comprehend the computer program and modify it if necessary or desired. PBREAK has been compared only with the three model test runs of Kobayashi, Cox, and Wurjanto (1991). Various extensions of PBREAK including wave overtopping and transmission are possible in the future.

Appendices B, C, and D present the input and output for the computation made for each of the three model test runs together with the comparisons between the computed and measured time series. The computed results presented in these appendices indicate the extensive information produced by PBREAK. The computed time series are also analyzed using spectral methods and explained physically in the separate report of Wurjanto and Kobayashi (1992a). The condensed version of this report will be published in the papers of Wurjanto and Kobayashi (1992b) and Kobayashi and Wurjanto (1992b). Finally, Appendix E explains the contents of the disk accompanying this report, while Appendix F summarizes the computation time for each of the examples presented in Appendices B, C, and D.

PART II

NUMERICAL METHOD

• 2.1 •

GOVERNING EQUATIONS

The wave motion on a permeable slope is computed for the normally incident wave train specified at the seaward boundary of the computation domain as shown in Figure 1a. The symbols shown in the figure are explained in the following where the prime indicates the dimensional variables.

x' = horizontal coordinate taken to be positive landward with $x' = 0$ at the seaward boundary
z' = vertical coordinate taken to be positive upward with $z' = 0$ at the still water level (SWL)
z_b' = elevation above SWL of the upper boundary of the permeable underlayer, referred to as the *upper slope*
z_p' = elevation above SWL of the impermeable slope that constitutes the lower boundary of the permeable underlayer, referred to as the *lower slope*
x_s' = x'-coordinate of the waterline on the upper slope
x_w' = x'-coordinate of the waterline on the lower slope
d_t' = water depth below SWL at the seaward boundary
d' = water depth below SWL above the upper slope
θ' = local angle of the upper slope
θ_p' = local angle of the lower slope
η' = elevation above SWL of the free surface or internal water table
h' = water depth above the upper slope = $d' + \eta'$
u' = depth-averaged horizontal velocity above the upper slope
q_b' = volume influx per unit horizontal area into the permeable underlayer
b' = local vertical thickness of the permeable underlayer = $z_b' - z_p'$
u_p' = vertically-averaged horizontal discharge velocity in the permeable underlayer
h_p' = vertical depth of the flow in the permeable underlayer as defined in Eq. 1

$$h_p' = \begin{cases} b' & \text{for } 0 \le x' \le x_s' \\ \eta' - z_p' & \text{for } x_s' \le x' \le x_w' \end{cases} \tag{1}$$

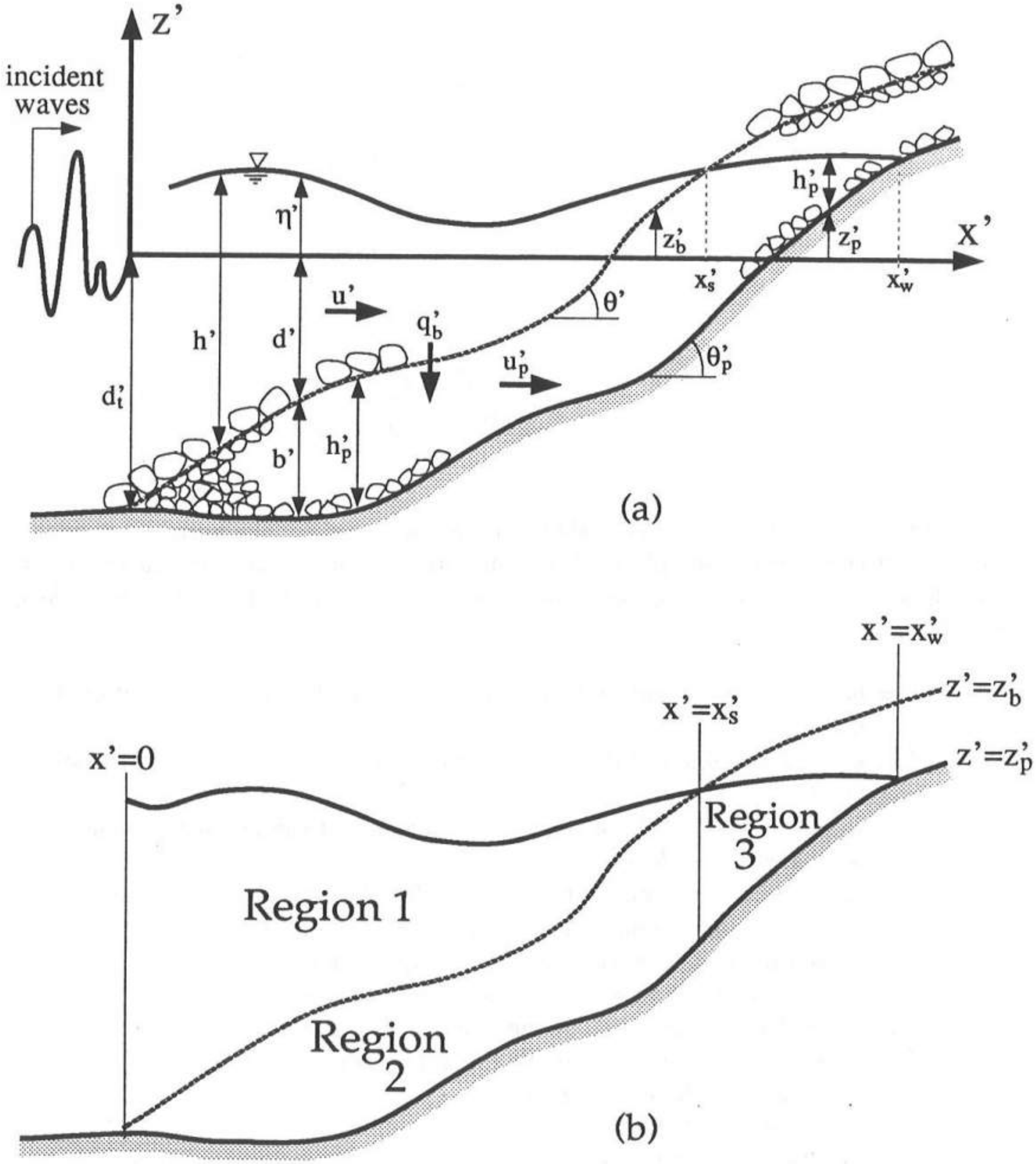


Figure 1: Definition sketch for numerical model for random waves on permeable coastal structures.

The terms *upper slope* and *lower slope* have been introduced above. In addition, the following terms are formally defined for clarity and simplicity.

- *Permeable underlayer* is the layer of homogeneous permeable materials lying on top of the impermeable slope, *not* including the armor layer. It is noted that the permeable underlayer is separated from the armor layer by the upper slope, $z' = z'_b$, and from the impermeable slope by the lower slope, $z' = z'_p$.
- *Armor layer* is the layer of stones placed on top of the upper slope $z' = z'_b$. The effects of the armor layer on the flow is simply modelled by the constant friction factor f'.
- *Region 1* is the region bounded vertically by the free surface of the flow and the upper slope, and horizontally by $x' = 0$ and $x' = x'_s$.
- *Region 2* is the region bounded vertically by the upper and lower slopes, and horizontally by $x' = 0$ and $x' = x'_s$.
- *Region 3* is the region bounded vertically by the internal water table and the lower slope, and horizontally by $x' = x'_s$ and $x' = x'_w$.
- *External flow* is the flow in the Region 1.
- *Internal flow* is the flow in the Regions 2 and 3.
- *Upper waterline* is the waterline on the *upper* slope. The location of this waterline is marked by $x' = x'_s$.
- *Lower waterline* is the waterline on the *lower* slope. The location of this waterline is marked by $x' = x'_w$.
- *Free surface* is used to describe the interface between the external flow and the air.
- *Internal water table* refers to the phreatic surface in Region 3.

The three regions defined above are illustrated in Figure 1b. Each region is governed by a distinct set of governing equations, and will be treated separately in this numerical model. In this report, the governing equations and other equations underlying the numerical model will be presented without derivation. The derivation of many of the equations involved in this numerical model can be found in Wurjanto and Kobayashi (1992a). It may also be necessary to refer to the papers and reports discussing waves on permeable slopes outlined in Part I.

The continuity and horizontal momentum equations for each region are:

REGION 1:

$$\frac{\partial h'}{\partial t'} + \frac{\partial}{\partial x'}\left(h'u'\right) = -q'_b \tag{2}$$

$$\frac{\partial}{\partial t'}\left(h'u'\right) + \frac{\partial}{\partial x'}\left(h'u'^2\right) = -gh'\frac{\partial \eta'}{\partial x'} - \frac{1}{2}f'|u'|u' - u'_b q'_b \tag{3}$$

REGION 2:

$$\frac{\partial}{\partial x'}\left(h'_p u'_p\right) = q'_b \tag{4}$$

$$\frac{\partial}{\partial t'}\left(h'_p u'_p\right) + \frac{1}{n_p}\frac{\partial}{\partial x'}\left(h'_p u'^2_p\right) - u'_b q'_b = -g n_p h'_p \frac{\partial \eta'}{\partial x'} - n_p h'_p\left(\alpha' + \beta'|u'_p|\right)u'_p \tag{5}$$

REGION 3:

$$\frac{\partial \eta'}{\partial t'} + \frac{1}{n_p}\frac{\partial}{\partial x'}\left(h'_p u'_p\right) = 0 \tag{6}$$

$$\frac{\partial}{\partial t'}\left(h'_p u'_p\right) + \frac{1}{n_p}\frac{\partial}{\partial x'}\left(h'_p u'^2_p\right) = -g n_p h'_p \frac{\partial \eta'}{\partial x'} - n_p h'_p\left(\alpha' + \beta'|u'_p|\right)u'_p \tag{7}$$

with

t' = time
g = gravitational acceleration
f' = constant friction factor associated with the armor layer
n_p = porosity of the permeable underlayer
u'_b = horizontal fluid velocity between the upper and lower slopes which is assumed to be given by

$$u'_b = \begin{cases} u' & \text{for } q'_b \geq 0 \\ \frac{u'_p}{n_p} & \text{for } q'_b < 0 \end{cases} \tag{8}$$

α' = coefficient expressing the laminar flow resistance
β' = coefficient expressing the turbulent flow resistance

The coefficients α' and β' are empirically given by Madsen and White (1976):

$$\alpha' = \frac{\alpha_o\left(1-n_p\right)^3 \nu}{\left(n_p d'_p\right)^2} \tag{9}$$

$$\beta' = \frac{\beta_o\left(1-n_p\right)}{n_p^3 d'_p} \tag{10}$$

in which

d'_p = representative diameter of the materials composing the permeable underlayer
ν = kinematic viscosity of the fluid
α_o = dimensionless constant in the range $780 \leq \alpha_o \leq 1500$
β_o = dimensionless constant in the range $1.8 \leq \beta_o \leq 3.6$

In this numerical model, the vertical thickness of the permeable underlayer b' is assumed to be zero at the seaward boundary $x' = 0$, although this assumption could be modified.

The following dimensionless variables and parameters are introduced to normalize Eqs. 2 through 10:

$$t = \frac{t'}{T'} \tag{11}$$

$$x = \frac{x'}{T'\sqrt{gH'}} \quad ; \quad x_s = \frac{x'_s}{T'\sqrt{gH'}} \quad ; \quad x_w = \frac{x'_w}{T'\sqrt{gH'}} \tag{12}$$

$$z = \frac{z'}{H'} \quad ; \quad z_p = \frac{z'_p}{H'} \quad ; \quad z_b = \frac{z'_b}{H'} \tag{13}$$

$$\eta = \frac{\eta'}{H'} \tag{14}$$

$$b = \frac{b'}{H'} \tag{15}$$

$$h = \frac{h'}{H'} \quad ; \quad h_p = \frac{h'_p}{H'} = \begin{cases} b & \text{for } 0 \leq x \leq x_s \\ \eta - z_p & \text{for } x_s \leq x \leq x_w \end{cases} \tag{16}$$

$$d = \frac{d'}{H'} \quad ; \quad d_t = \frac{d'_t}{H'} \tag{17}$$

$$u = \frac{u'}{\sqrt{gH'}} \quad ; \quad u_b = \frac{u'_b}{\sqrt{gH'}} \tag{18}$$

$$u_p = \frac{u'_p}{p_q\sqrt{gH'}} \tag{19}$$

$$q_b = \frac{q'_b}{p_q H'/T'} \tag{20}$$

$$\theta = \sigma \tan\theta' \quad ; \quad \theta_p = \sigma \tan\theta'_p \tag{21}$$

$$\sigma = T'\sqrt{\frac{g}{H'}} \tag{22}$$

$$f = \frac{1}{2}\sigma f' \tag{23}$$

where H' and T' are the reference wave height and period, respectively, to be specified by the user. These reference wave parameters can be taken as the height and period used to characterize the incident waves for a particular problem. Substituting the physical variables in the governing equations 2 through 7 with their corresponding dimensionless variables using Eqs. 11 through 23 yields the following normalized governing equations:

REGION 1:

$$\frac{\partial h}{\partial t} + \frac{\partial m}{\partial x} + p_q q_b = 0 \tag{24}$$

$$\frac{\partial m}{\partial t} + \frac{\partial}{\partial x}\left(mu + \frac{1}{2}h^2\right) + \theta h + f|u|u + p_q u_b q_b = 0 \tag{25}$$

REGION 2:

$$\frac{\partial}{\partial x}(h_p u_p) = q_b \tag{26}$$

$$p_u \frac{\partial}{\partial t}(h_p u_p) + p_u^2 \frac{\partial}{\partial x}\left(h_p u_p^2\right) - p_u u_b q_b = -h_p \frac{\partial \eta}{\partial x} - h_p u_p (\mu + |u_p|) \tag{27}$$

REGION 3:

$$\frac{\partial \eta}{\partial t} + p_u \frac{\partial}{\partial x}(h_p u_p) = 0 \tag{28}$$

$$p_u \frac{\partial}{\partial t}(h_p u_p) + p_u^2 \frac{\partial}{\partial x}\left(h_p u_p^2\right) + h_p \frac{\partial \eta}{\partial x} + h_p u_p (\mu + |u_p|) = 0 \tag{29}$$

where

$$m = hu \tag{30}$$

$$u_b = \begin{cases} u & \text{for } q_b \geq 0 \\ p_u u_p & \text{for } q_b < 0 \end{cases} \tag{31}$$

$$p_q = n_p p_u \tag{32}$$

$$p_u = \sqrt{\frac{n_p}{\beta_o (1 - n_p)} \cdot \frac{d'_p}{T' \sqrt{gH'}}} \tag{33}$$

$$\mu = \frac{\alpha_o (1 - n_p)^2 \nu}{\beta_o p_u d'_p \sqrt{gH'}} \tag{34}$$

In terms of the normalized coordinate system, the upper slope is located at

$$z_b = \int_0^x \theta \, dx - d_t \quad ; \quad x \geq 0 \tag{35}$$

and the lower slope is at

$$z_p = \int_0^x \theta_p \, dx - d_t \quad ; \quad x \geq 0 \tag{36}$$

For a uniform upper slope Eq. 35 reduces to the following equation.

$$z_b = \theta x - d_t \quad ; \quad x \geq 0, \text{ constant } \theta \tag{37}$$

The following vectors are introduced to express the governing equations 24 through 29 in a vector form.

$$\mathbf{U}^{[1]} = \begin{pmatrix} m \\ h \end{pmatrix} \tag{38}$$

$$\mathbf{F}^{[1]} = \begin{pmatrix} mu + \frac{1}{2}h^2 \\ m \end{pmatrix} \tag{39}$$

$$\mathbf{G}^{[1]} = \begin{pmatrix} \theta h + f|u|u + p_q u_b q_b \\ p_q q_b \end{pmatrix} \tag{40}$$

$$\mathbf{U}^{[2]} = (m_p) \tag{41}$$

$$\mathbf{F}^{[2]} = \left(\frac{m_p^2}{h_p}\right) \tag{42}$$

$$\mathbf{G}^{[2]} = \left(- p_u u_b q_b + h_p \frac{\partial h}{\partial x} + \theta h_p + \frac{m_p}{p_u}\left[\mu + \frac{|m_p|}{p_u h_p}\right]\right) \tag{43}$$

$$\mathbf{U}^{[3]} = \begin{pmatrix} m_p \\ h_p \end{pmatrix} \tag{44}$$

$$\mathbf{F}^{[3]} = \begin{pmatrix} \frac{m_p^2}{h_p} + \frac{1}{2}h_p^2 \\ m_p \end{pmatrix} \tag{45}$$

$$\mathbf{G}^{[3]} = \begin{pmatrix} \theta_p h_p + \frac{\mu m_p}{p_u} + \frac{|m_p| m_p}{p_u^2 h_p} \\ 0 \end{pmatrix} \tag{46}$$

with

$$m_p = p_u h_p u_p \tag{47}$$

The normalized governing equations 24, 25, and 27 through 29 can now be written in a single expression as follows:

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

where the bracketed superscript $[k]$ denotes the *region* where the component equation k applies. It is noted that each of the vectors $\mathbf{U}^{[2]}$, $\mathbf{F}^{[2]}$, and $\mathbf{G}^{[2]}$ has only one element.

Eq. 26, which is not included in Eq. 48, is used to express the normalized volume influx q_b in terms of one of the variables involved in Eq. 48.

$$q_b = \frac{1}{p_u}\frac{\partial m_p}{\partial x} = \frac{1}{p_u}\frac{\partial \mathbf{U}^{[2]}}{\partial x} \quad \text{for REGIONS 1 and 2} \tag{49}$$

It is obvious that the quantity q_b does not exist in Region 3. However, it will be useful in Sections 2.7 and 2.8 to have the explicit mathematical expression for q_b for Region 3.

$$q_b = 0 \quad \text{for REGION 3} \tag{50}$$

The following quantities are introduced because they appear in the computer program PBREAK as explained in Part III.

$$L_0 = \frac{L_0'}{d_t'} \quad ; \quad L_0' = \frac{g(T')^2}{2\pi} \tag{51}$$

$$L = \frac{L'}{d_t'} \tag{52}$$

$$\xi = \sqrt{\frac{L_0'}{H'}} \tan\theta_\xi' = \frac{1}{\sqrt{2\pi}}\sigma \tan\theta_\xi' \tag{53}$$

$$U_r = \frac{H'(L')^2}{(d_t')^3} = \frac{L^2}{d_t} \tag{54}$$

where

L_0' = dimensional linear wavelength in deep water
L' = dimensional linear wavelength at the seaward boundary
ξ = surf similarity parameter
θ_ξ' = representative slope used to define the surf similarity parameter
U_r = Ursell number at the seaward boundary
d_t' = water depth below SWL at the seaward boundary
d_t is given in Eq. 17
σ is given by Eq. 22

• 2.2 •

Overview of the Numerical Method

The problem formulated in Section 2.1 is one dimensional in space x and unsteady in time t. A numerical solution of the problem is sought at the discrete grid points as shown in Figure 2. Hereafter, the space and time refer to the normalized independent variables x and t, respectively, for the computation performed in terms of normalized variables.

The computational grid points are separated in space by an equal space interval Δx, and in time by a time increment Δt. Over the course of the computation Δx is held constant while Δt may vary. The computational nodes are spatially numbered from 1 to w where the node number 1 is assigned to the node at the seaward boundary ($x = 0$) and the node number w is for the node associated with the *lower* waterline. The subscript j will be used in the rest of this report to exclusively denote the spatial node number. The node j=1,2,3,. . .,w is associated with the location $x_j = (j-1)\Delta x$, as shown in Figure 2 for j=1,2,3,4. In relation with the integer w, introduced here is the integer s that denotes the node number associated with the *upper* waterline. The nature of the problem dictates that the integers s and w vary with time. The locations of the upper and lower waterlines are controlled by the normalized water depth h and internal flow depth h_p, respectively. The integers s and w are taken to satisfy the following requirements:

$$h_s > \quad \delta_s \quad \geq h_{s+1} \tag{55}$$

$$h_w > \quad \delta_w \quad \geq h_{w+1} \tag{56}$$

where δ_s and δ_w are small positive values specified by the user to define the computational upper and lower waterlines.

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

$$\left.\begin{array}{l} \eta = 0 \\ u = 0 \\ u_p = 0 \end{array}\right\} \text{ at } t = 0 \text{ for the region } x \geq 0 \tag{57}$$

From the known initial conditions, and with a time-varying boundary condition prescribed at the seaward boundary, the computation marches in time up to the specified time level. Figure 2 illustrates an arbitrary "present" time level t and the "next" time level $t^* = t + \Delta t$. At the present time level, the values of the dependent variables at all the computation nodes are known. From the known values at the present time level, the unknown values at the next time level are computed in the following order: $\boxed{\text{Region 1}} \longrightarrow \boxed{\text{Region 2}} \longrightarrow \boxed{\text{Region 3}}$ using the numerical methods to be described in the next three sections. Throughout this report, the asterisk denotes the value at the next time level $t^* = t + \Delta t$. For example, the vector $\mathbf{U}$ for Region 3 at the node j at the next time level $t^* = t + \Delta t$ is denoted by $\mathbf{U}_j^{[3]*}$, and at the present time level t by $\mathbf{U}_j^{[3]}$ without any asterisk.

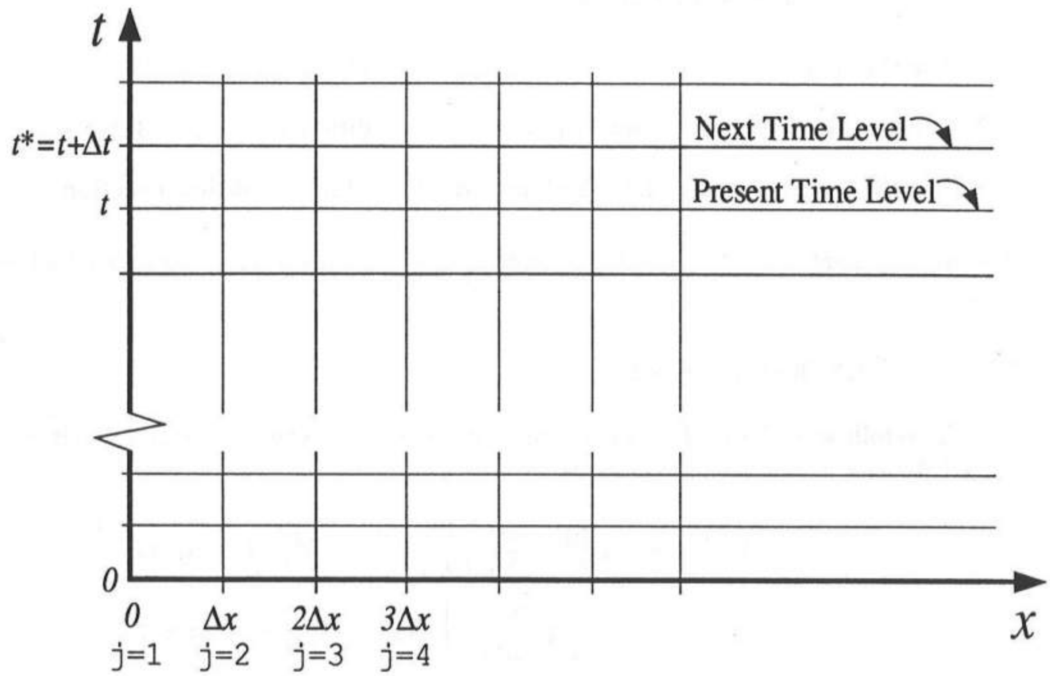


Figure 2: Diagram of the computation grid.

• 2.3 •

Numerical Method for Region 1

The flow in Region 1 is solved by extending the explicit dissipative Lax-Wendroff method adopted from Kobayashi and Wurjanto (1990) with considerable modifications to allow a thick permeable underlayer and simplify the landward boundary condition.

The marching scheme from the present time level t to the next time level $t^* = t + \Delta t$ for Region 1 is divided into three parts:

1. For the nodes j=2,3,...,$(s-2)$: the general scheme (Section 2.3.1).
2. For the node j=1: the seaward boundary condition (Section 2.3.2).
3. For the nodes $j=(s-1),s$: the landward boundary condition (Section 2.3.3).

In addition, a separate Section 2.3.4 discusses wave run-up, run-down, and set-up on the upper slope.

2.3.1 General Scheme

The following finite difference equation, whose accuracy is second order in space and time, is used for the nodes j=2,...,$(s-2)$:

$$\begin{aligned} \mathbf{U}_j^{[1]*} &= \mathbf{U}_j^{[1]} - \frac{\Delta t}{\Delta x}\left[\frac{1}{2}\left(\mathbf{F}_{j+1}^{[1]} - \mathbf{F}_{j-1}^{[1]}\right) + \Delta x\ \mathbf{G}_j^{[1]}\right] \\ &+ \frac{1}{2}\left(\frac{\Delta t}{\Delta x}\right)^2 \left[\ (\mathbf{H}_j - \mathbf{H}_{j-1}) - \Delta x\ \mathbf{S}_j\right] \\ &+ \mathbf{D}_j - \frac{n_p}{2}\frac{(\Delta t)^2}{\Delta x}\mathbf{P}_j \end{aligned} \tag{58}$$

where the unknown quantities at the next time level on the left hand side is expressed in terms of the known quantities at the present time level on the right hand side. It is reminded here that the asterisk indicates the next time level $t^* = t + \Delta t$, the bracketed superscript [1] indicates Region 1, the subscript j denotes the spatial node number, and the quantities written without an asterisk are of the present time level t.

The vector $\mathbf{H}$ in Eq. 58 is given by:

$$\mathbf{H}_j = \frac{1}{2}\left[\mathbf{A}_{j+1} + \mathbf{A}_j\right]\left[\left(\mathbf{F}_{j+1}^{[1]} - \mathbf{F}_j^{[1]}\right) + \frac{\Delta x}{2}\left(\mathbf{G}_{j+1}^{[1]} + \mathbf{G}_j^{[1]}\right)\right] \tag{59}$$

with

$$\mathbf{A}_j = \begin{pmatrix} 2\dfrac{m_j}{h_j} & \left(h_j - \dfrac{m_j^2}{h_j^2}\right) \\ 1 & 0 \end{pmatrix} \tag{60}$$

The vector **S** in Eq. 58 is given by:

$$\mathbf{S}_j = \begin{pmatrix} \Delta x \; e_j - \frac{1}{2}\theta_j \left(m_{j+1} - m_{j-1}\right) \\ \\ 0 \end{pmatrix} \tag{61}$$

with

$$\begin{aligned} e_j \;\; &= \;\; 2f\frac{|u_j|}{h_j}\left[\left(u_j^2 - h_j\right)\frac{h_{j+1} - h_{j-1}}{2\Delta x}\right. \\ &\qquad \left. - \, u_j \frac{m_{j+1} - m_{j-1}}{2\Delta x} - \theta_j h_j - f|u_j|u_j\right] \end{aligned} \tag{62}$$

The vector $\mathbf{D}_j$ in Eq. 58 represents the additional term for damping high frequency parasitic waves, which tend to appear at the rear of a breaking wave, and is given by:

$$\mathbf{D}_j = \frac{1}{2}\frac{\Delta t}{\Delta x}\left[\mathbf{Q}_j\left(\mathbf{U}_{j+1}^{[1]} - \mathbf{U}_j^{[1]}\right) - \mathbf{Q}_{j-1}\left(\mathbf{U}_j^{[1]} - \mathbf{U}_{j-1}^{[1]}\right)\right] \tag{63}$$

with

$$\mathbf{Q}_j = (q_1)_j \, \mathbf{I} + \frac{1}{2}(q_2)_j \left(\mathbf{A}_j + \mathbf{A}_{j+1}\right) \tag{64}$$

where $\mathbf{I}$ = unit matrix, and the coefficients $(q_1)_j$ and $(q_2)_j$ are given by:

$$\begin{aligned} (q_1)_j \;\; &= \;\; -\frac{1}{2\left(c_{j+1} + c_j\right)}\left[\epsilon_1|\varphi_{j+1} - \varphi_j|\left(\psi_{j+1} + \psi_j\right)\right. \\ &\qquad \left. - \, \epsilon_2|\psi_{j+1} - \psi_j|\left(\varphi_{j+1} + \varphi_j\right)\right] \end{aligned} \tag{65}$$

$$(q_2)_j \;\; = \;\; \frac{1}{\left(c_{j+1} + c_j\right)}\left[\epsilon_1|\varphi_{j+1} - \varphi_j| - \epsilon_2|\psi_{j+1} - \psi_j|\right] \tag{66}$$

with

$$c = \sqrt{h} \quad ; \quad \varphi = u + c \quad ; \quad \psi = u - c \tag{67}$$

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

The vector $\mathbf{P}_j$ in Eq. 58 is given by:

$$\mathbf{P}_j = \begin{pmatrix} \frac{f\,|u_j|}{h_j}\left[u_j - (u_b)_j\right]\Delta\mathbf{U}_j^{[2]} - \frac{\theta_j}{2}\Delta\mathbf{U}_j^{[2]} - \tilde{P}_j \\ \\ -\left(\mathbf{B}_j - \mathbf{B}_{j-1}\right) \end{pmatrix} \tag{68}$$

with

$$\tilde{P}_j = \begin{cases} \frac{1}{h_j}\left\{ m_j\left(\mathbf{B}_j - \mathbf{B}_{j-1}\right) + \frac{1}{2}\Delta\mathbf{U}_j^{[2]}\left[\frac{\left(F_1^{[1]}\right)_{j+1} - \left(F_1^{[1]}\right)_{j-1}}{2\Delta x} + \left(G_1^{[1]}\right)_j \right.\right. \\ \qquad\left.\left. -u_j\frac{\left(F_2^{[1]}\right)_{j+1} - \left(F_2^{[1]}\right)_{j-1}}{2\Delta x} - u_j\left(G_2^{[1]}\right)_j\right]\right\} & \text{for } q_b \geq 0 \\ \\ \frac{1}{(h_p)_j}\left\{ (m_p)_j\left(\mathbf{B}_j - \mathbf{B}_{j-1}\right) + \frac{1}{2}\Delta\mathbf{U}_j^{[2]}\left[\frac{\mathbf{F}_{j+1}^{[2]} - \mathbf{F}_{j-1}^{[2]}}{2\Delta x} + \mathbf{G}_j^{[2]}\right]\right\} & \text{for } q_b < 0 \end{cases} \tag{69}$$

and

$$\mathbf{B}_j = \frac{1}{\Delta x}\left(\mathbf{F}_{j+1}^{[2]} - \mathbf{F}_j^{[2]}\right) + \frac{1}{2}\left(\mathbf{G}_{j+1}^{[2]} + \mathbf{G}_j^{[2]}\right) \tag{70}$$

The symbol $\Delta\mathbf{U}_j^{[2]}$ appearing in Eqs. 68 and 69 represents the following difference:

$$\Delta\mathbf{U}_j^{[k]} = \mathbf{U}_{j+1}^{[k]} - \mathbf{U}_{j-1}^{[k]} \quad ; \quad k = 1,\ 2, \text{ or } 3 \tag{71}$$

where the bracketed superscript $[k]$ denotes the region k.

The symbols $F_1^{[1]}$, $F_2^{[1]}$, $G_1^{[1]}$, and $G_2^{[1]}$ in Eq. 69 are variants of the following generic notation:

$$X_i^{[k]} = \text{the } i\text{-th element of the vector } \mathbf{X}^{[k]} \text{ of Region } k \tag{72}$$

with i being 1 or 2; k being 1 or 3; and X being U, F, or G, corresponding to the vector $\mathbf{U}$, $\mathbf{F}$, or $\mathbf{G}$, respectively, where the quantity U will appear in the subsequent sections.

The normalized volume influx q_b may be evaluated using a central finite difference form of Eq. 49.

$$(q_b)_j = \frac{1}{2\Delta x p_u}\Delta\mathbf{U}_j^{[2]} \tag{73}$$

The numerical stability criterion for this explicit finite difference method is adopted from Packwood (1980):

$$\frac{\Delta t}{\Delta x} < \frac{\left[\sqrt{\left(1 + \frac{1}{4}\epsilon^2\right)} - \frac{1}{2}\epsilon\right]}{|u_m| + c_m} \tag{74}$$

which requires that

$$|u_j| < \frac{\Delta x}{\Delta t} \quad \text{at any time } t \text{ for any node } j \tag{75}$$

where

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

To facilitate the use of Eq. 74, the numerical stability indicator Ω is introduced

$$\Omega = \frac{\Delta x}{\Delta t} \frac{\left[\sqrt{\left(1+\frac{1}{4}\epsilon^2\right)} - \frac{1}{2}\epsilon\right]}{1+\sqrt{d_t}} \tag{76}$$

where the normalized water depth d_t below SWL at the seaward boundary is defined in Eq. 17. If the following approximations are made

$$\begin{aligned} u_m &\simeq 1 \\ c_m &\simeq \sqrt{d_t} \end{aligned}$$

the numerical stability criterion given by Eq. 74 can be expressed as:

$$\Omega > 1 \tag{77}$$

Eq. 77 will actually be used in lieu of Eq. 74 as a basis to determine the time step size Δt for given space interval Δx. Eq. 75 will also be used to determine whether the numerical stability criterion is violated or not. The numerical method for each of the three regions can be considered as a variant of the Lax-Wendroff method (*e.g.*, Richtmyer and Morton 1967). Thus, the numerical stability criterion established for Region 1 should also prevail for Regions 2 and 3 as well. Moreover, the numerical stability criteria for nonlinear problems including the present problem are approximate in nature. For this reason, no numerical stability criterion is imposed for Regions 2 and 3.

2.3.2 Seaward Boundary Condition

In order to derive an appropriate seaward boundary condition for Region 1, Eqs. 24 and 25 are expressed in the following characteristic forms (Kobayashi and Wurjanto 1990):

$$\frac{\partial \alpha}{\partial t} + (u+c)\frac{\partial \alpha}{\partial x} = -\theta - \frac{f|u|u}{h} + \frac{p_q q_b (u-c-u_b)}{h} \quad ; \quad \frac{dx}{dt} = u + c \tag{78}$$

$$\frac{\partial \beta}{\partial t} + (u-c)\frac{\partial \beta}{\partial x} = \theta + \frac{f|u|u}{h} - \frac{p_q q_b (u+c-u_b)}{h} \quad ; \quad \frac{dx}{dt} = u - c \tag{79}$$

with c already defined in Eq. 67, and α and β given as follows:

$$\alpha = u + 2c \quad ; \quad \beta = -u + 2c \tag{80}$$

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

$$h = d_t + \eta_i(t) + \eta_r(t) \quad \text{at } x = 0 \tag{81}$$

where η_i and η_r are the free surface variations normalized by H' at $x = 0$ due to the incident and reflected waves, respectively. The incident wave train is specified by prescribing the variation of η_i for $t \geq 0$. The normalized reflected wave train η_r is approximately expressed in terms of the seaward advancing characteristic β at $x = 0$:

$$\eta_r(t) \simeq \frac{1}{2}\sqrt{d_t}\,\beta(t) - d_t \quad \text{at } x = 0 \tag{82}$$

The values of the total water depth h_1^* and the depth-averaged horizontal velocity u_1^* at the next time level $t^* = t + \Delta t$ and at the seaward boundary $x = 0$ are computed as follows. Eq. 81 yields the following equation:

$$h_1^* = d_t + \eta_i^* + \eta_r^* \tag{83}$$

in which the unknown η_r^* is computed using Eq. 82:

$$\eta_r^* \simeq \frac{1}{2}\sqrt{d_t}\,\beta_1^* - d_t \tag{84}$$

The characteristic β_1^* in Eq. 84 is evaluated using the following first-order finite difference equation obtained from Eq. 79 with $f = 0$:

$$\beta_1^* = \beta_1 - \frac{\Delta t}{\Delta x}(u_1 - c_1)(\beta_2 - \beta_1) + \Delta t\,\theta_1 - \frac{\Delta t}{\Delta x}\frac{n_p}{h_1}\left(\mathbf{U}_2^{[2]} - \mathbf{U}_1^{[2]}\right)\left(u_1 - (u_b)_1 + c_1\right) \tag{85}$$

where all the terms on the right hand side of Eq. 85 are of the present time level t and known. Use has been made of Eq. 49 to derive Eq. 85.

The value of u_1^* is computed using Eq. 80:

$$u_1^* = 2\sqrt{h_1^*} - \beta_1^* \tag{86}$$

where β_1^* and h_1^* are computed using Eqs. 85 and 83, respectively.

2.3.3 Landward Boundary Condition

The landward boundary for Region 1 is at the moving waterline on the upper slope where the water depth h is theoretically zero. Computationally, the landward boundary for Region 1 is located at the node s satisfying Eq. 55. The procedure to determine the waterline node s^* at the next time level t^* based on the values at the present time t is detailed below:

1. Compute
   $h_{s+1} = (2h_s - h_{s-1})$ and
   $u_{s+1} = (2u_s - u_{s-1})$.

If $h_{s+1} < 0$, set $h_{s+1} = 0$ and $u_{s+1} = 0$.
Obtain $m_{s+1} = h_{s+1}u_{s+1}$.

2. Compute h_j^* and m_j^* at the time t^* for the nodes $j = (s-1)$ and $j = s$, using Eq. 58 without the damping term $\mathbf{D}_j$.

3. If $h_{s-1}^* \leq \delta_s$, abort the computation because the waterline should not move more than Δx due to the numerical stability criterion given by Eq. 75 where the kinematic condition requires that the normalized horizontal velocity of the waterline equals u_s.
Otherwise proceed.

4. If $h_s^* > h_{s-1}^*$, compute
$\widehat{h_s^*} = (2h_{s-1}^* - h_{s-2}^*)$ and
$\widehat{u_s^*} = (2u_{s-1}^* - u_{s-2}^*)$.
If $\widehat{h_s^*} \leq 0$, set $s^* = (s-1)$ and go to Step 7.
If $\widehat{h_s^*} > 0$, ensure that the water depth and the magnitude of the horizontal fluid velocity near the waterline decrease landward:
If $\widehat{h_s^*} > h_{s-1}^*$, set $h_s^* = 0.9h_{s-1}^*$. Otherwise set $h_s^* = \widehat{h_s^*}$.
If $|\widehat{u_s^*}| > |u_{s-1}^*|$, set $u_s^* = 0.9u_{s-1}^*$. Otherwise set $u_s^* = \widehat{u_s^*}$.
Obtain $m_s^* = h_s^* u_s^*$.

5. If $h_s^* \leq \delta_s$, set $s^* = (s-1)$ and go to Step 7.
Otherwise, compute
$h_{s+1}^* = (2h_s^* - h_{s-1}^*)$,
$u_{s+1}^* = (2u_s^* - u_{s-1}^*)$, and
$m_{s+1}^* = h_{s+1}^* u_{s+1}^*$.
The use of linear extrapolation to find h_{s+1}^* and u_{s+1}^* and the measures taken in Step 4 should ensure that the water depth and the magnitude of the horizontal fluid velocity near the waterline decrease landward.

6. If $h_{s+1}^* \leq \delta_s$, set $s^* = s$.
Otherwise, set $s^* = (s+1)$.
Go to Step 7.

7. After $s*$ is obtained, set $h_j^* = 0$, $u_j^* = 0$, and $m_j^* = 0$ for $j \geq (s^* + 1)$ since no water is present on the upper slope landward of the upper waterline node.

2.3.4 Run-up, Run-down, and Set-up on the Upper Slope

Once the normalized water depth h at a given time is known as a function of x, the normalized free surface elevation $Z_r^U = (Z_r^U)'/H'$, where the physical water depth on the upper slope equals a specified value δ_r', can be computed as long as $\delta_r = (\delta_r'/H') > \delta_s$. The use of the physical depth δ_r' may be related to the use of a waterline meter to measure the waterline oscillation on the upper slope. The specified depth δ_r' can be regarded as the vertical distance between the waterline meter and the upper slope, while the corresponding elevation $(Z_r^U)'$ is the elevation above SWL of the intersection between the waterline meter and the free surface. The computed oscillations of $Z_r^U(t)$ for different values of δ_r' can be used to examine the sensitivity to δ_r' of wave *run-up*, *run-down*, and *set-up* on the upper slope. Wave run-up and run-down on a slope are normally defined as the maximum and minimum elevations relative to SWL reached by up-rushing and down-rushing water on the slope, respectively. Wave set-up is defined here as the mean of the oscillating waterline elevation relative to SWL. The normalized run-up R^U, run-down R_d^U, and set-up $\overline{Z_r^U}$ on the upper slope for given δ_r' are obtained from the computed oscillation of $Z_r^U(t)$.

• 2.4 •

NUMERICAL METHOD FOR REGION 2

At each time-marching step, Region 2 is computed after the computation for Region 1 is completed. In consequence:

- The landward boundary of the Region 2 at the next time level $t^* = t + \Delta t$ is the upper waterline node s^* that has been determined from the preceding computation of Region 1.
- When the computation for Region 2 needs values from Region 1, the values at the time t^* are used since the flow in Region 2 is mostly forced by the flow in Region 1. The variables of Region 1 involved in the analysis in Section 2.4 appear with an asterisk indicating the time t^*.

The flow in Region 2 is solved using the MacCormack method (MacCormack 1969), which is in the class of "two-step Lax-Wendroff methods". The two steps in this method are one of a forward space difference and another of a backward space difference, which are interchangeable (Roache 1982, p. 253; Anderson *et al.*, 1984, p. 103). The "backward-forward" setting used by Fennema and Chaudhry (1986) is adopted here. Their use of the term *predictor* step referring to the *first* step and the term *corrector* step for the *second* step is also adopted here. The primary reason for the use of this backward-forward setting is to compute the value at the node s^* in the predictor step.

PREDICTOR STEP:

$$\dot{\mathbf{U}}_j^{[2]} = \mathbf{U}_j^{[2]} - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_j^{[2]} - \mathbf{F}_{j-1}^{[2]}\right) - \Delta t \mathbf{G}_j^{[2]} \tag{87}$$

CORRECTOR STEP:

$$\ddot{\mathbf{U}}_j^{[2]} = \dot{\mathbf{U}}_j^{[2]} - \frac{\Delta t}{\Delta x}\left(\dot{\mathbf{F}}_{j+1}^{[2]} - \dot{\mathbf{F}}_j^{[2]}\right) - \Delta t \dot{\mathbf{G}}_j^{[2]} \tag{88}$$

The value of the vector $\mathbf{U}$ for Region 2 at the next time level $t^* = t + \Delta t$ is then given by:

$$\mathbf{U}_j^{[2]*} = \frac{1}{2}\left(\mathbf{U}_j^{[2]} + \ddot{\mathbf{U}}_j^{[2]}\right) \tag{89}$$

The vector $\mathbf{G}_j^{[2]}$ in Eq. 87 is computed using Eq. 43 together with Eq. 49 in which the spatial derivatives involved are discretized using backward differences to be consistent with the predictor step

$$\mathbf{G}_j^{[2]} = -(\hat{u}_b)_j \frac{\mathbf{U}_j^{[2]} - \mathbf{U}_{j-1}^{[2]}}{\Delta x} + (h_p)_j \frac{\left(U_2^{[1]*}\right)_j - \left(U_2^{[1]*}\right)_{j-1}}{\Delta x} + \theta_j (h_p)_j + \frac{\mathbf{U}_j^{[2]}}{p_u}\left(\mu + \frac{\left|\mathbf{U}_j^{[2]}\right|}{p_u (h_p)_j}\right) \quad (90)$$

where $U_2^{[1]}$ is defined in Eq. 72. The quantities written with an overhead dot ($\dot{}$) in Eq. 88 are evaluated after the predictor step but before the corrector step, as detailed below.

Eq. 41 yields

$$\dot{\mathbf{U}}_j^{[2]} = (\dot{m}_p)_j \quad (91)$$

Using Eqs. 42 and 43, the vectors $\dot{\mathbf{F}}_j^{[2]}$ and $\dot{\mathbf{G}}_j^{[2]}$ in Eq. 88 are computed from

$$\dot{\mathbf{F}}_j^{[2]} = \frac{(\dot{m}_p)_j^2}{(h_p)_j} = \frac{\left(\dot{\mathbf{U}}_j^{[2]}\right)^2}{(h_p)_j} \quad (92)$$

$$\dot{\mathbf{G}}_j^{[2]} = -(\dot{u}_b)_j \frac{\dot{\mathbf{U}}_{j+1}^{[2]} - \dot{\mathbf{U}}_j^{[2]}}{\Delta x} + (h_p)_j \frac{\left(U_2^{[1]*}\right)_{j+1} - \left(U_2^{[1]*}\right)_j}{\Delta x} + \theta_j (h_p)_j + \frac{\dot{\mathbf{U}}_j^{[2]}}{p_u}\left(\mu + \frac{\left|\dot{\mathbf{U}}_j^{[2]}\right|}{p_u (h_p)_j}\right) \quad (93)$$

where the forward differences consistent with the corrector step are used to represent the spatial derivatives involved therein. It is noted that $(h_p)_j$ in Region 2 is equal to b_j and given as can be seen from Eq. 16. The quantities $(\hat{u}_b)_j$ in Eq. 90 and $(\dot{u}_b)_j$ in Eq. 93 are based on Eq. 31 with Eq. 47 and given as follows:

$$(\hat{u}_b)_j = \begin{cases} u_j^* & \text{for } (q_b)_j \geq 0 \\ \dfrac{(m_p)_j}{(h_p)_j} & \text{for } (q_b)_j < 0 \end{cases} \quad (94)$$

$$(\dot{u}_b)_j = \begin{cases} u_j^* & \text{for } (\dot{q}_b)_j \geq 0 \\ \dfrac{(\dot{m}_p)_j}{(h_p)_j} & \text{for } (\dot{q}_b)_j < 0 \end{cases} \quad (95)$$

where q_b is given by Eq. 73 and $\dot{q}_b$ is computed from

$$(\dot{q}_b)_j = \frac{1}{2\Delta x p_u} \Delta \dot{\mathbf{U}}_j^{[2]} \quad (96)$$

Because of the assumption of zero thickness of the permeable underlayer at the seaward boundary

$$b = 0 \text{ at } x = 0 \quad (97)$$

the boundary condition of no lateral flux at the seaward boundary ($x = 0$; $j = 1$) of Region 2 can be imposed:

$$m_p = 0 \text{ at } x = 0 \text{ for } t \geq 0 \tag{98}$$

which can be rewritten in the discrete form:

$$(m_p)_1 = 0 \qquad \text{at all time levels} \tag{99}$$

Thus,

$$\mathbf{U}_1^{[2]*} = \ddot{\mathbf{U}}_1^{[2]} = \dot{\mathbf{U}}_1^{[2]} = \mathbf{U}_1^{[2]} = 0 \tag{100}$$

The time-marching from the present time level t to the next time level $t^* = t + \Delta t$ is performed according to the procedure described below:

1. For the node j=1, $\mathbf{U}_j^{[2]*}$ is given by Eq. 100.

2. For Region 2, the thickness of the permeable underlayer is given by

$$(h_p)_j = b_j \quad \text{for } j = 1, 2, \ldots, s \tag{101}$$

   where b used in Eq. 16 is the normalized permeable underlayer thickness.
   If $s^* > s$, that is, $s^* = s + 1$, then:

   (a) Set

$$(h_p)_j = b_j \quad \text{for } j = s^* \tag{102}$$

   (b) The normalized flux m_p is defined by Eq. 47 for both Regions 2 and 3. When $s^* > s$, the node s^* which formerly located in Region 3 is now the border node between Regions 2 and 3. Hence, set

$$\left.\begin{aligned} \mathbf{U}_j^{[2]} &= (m_p)_j = \left(U_1^{[3]}\right)_j \\ (u_p)_j &= \frac{\left(U_1^{[3]}\right)_j}{p_u\,(h_p)_j} \end{aligned}\right\} \quad \text{for } j = s^* \tag{103}$$

   where $\left(U_1^{[3]}\right)_j$ is defined by Eq. 72 together with Eq. 44.

3. Compute $\dot{\mathbf{U}}_j^{[2]}$ using the predictor step Eq. 87 for j=2,3,...,s^*.

4. For the node j=2,3,...,($s^* - 1$), compute $\mathbf{U}_j^{[2]*}$ using Eqs. 88 and 89.

5. For the nodes j=s^*, let

$$\widehat{\mathbf{U}}_j^{[2]} = \dot{\mathbf{U}}_j^{[2]} \quad \text{for } j = s^* \tag{104}$$

   where the right hand side of Eq. 104 has been computed in Step 3. The final value of $\mathbf{U}_j^{[2]*}$ for $j = s^*$ will be described in Step 6 of Section 2.5.1 in relation with the seaward boundary condition of Region 3.

• 2.5 •

NUMERICAL METHOD FOR REGION 3

Region 3 is similar to Region 1 except that Region 3 is inside the permeable underlayer as shown in Figure 1. The seaward boundary for Region 3 at the next time level $t^* = t + \Delta t$ is the upper waterline node s^*, which has been determined from the preceding computation of Region 1.

The MacCormack method is used to solve Eq. 48 for the flow in Region 3. The "backward-forward" setting is used for Region 3. This is the same setting that has been applied to Region 2.

PREDICTOR STEP:

$$\dot{\mathbf{U}}_j^{[3]} = \mathbf{U}_j^{[3]} - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_j^{[3]} - \mathbf{F}_{j-1}^{[3]}\right) - \Delta t \mathbf{G}_j^{[3]} \tag{105}$$

CORRECTOR STEP:

$$\ddot{\mathbf{U}}_j^{[3]} = \dot{\mathbf{U}}_j^{[3]} - \frac{\Delta t}{\Delta x}\left(\dot{\mathbf{F}}_{j+1}^{[3]} - \dot{\mathbf{F}}_j^{[3]}\right) - \Delta t \dot{\mathbf{G}}_j^{[3]} \tag{106}$$

The value of the vector $\mathbf{U}$ for Region 3 at the next time level $t^* = t + \Delta t$ is given by:

$$\mathbf{U}_j^{[3]*} = \frac{1}{2}\left(\mathbf{U}_j^{[3]} + \ddot{\mathbf{U}}_j^{[3]}\right) \tag{107}$$

where $\mathbf{F}^{[3]}$ and $\mathbf{G}^{[3]}$ given by Eqs. 45 and 46 are fairly simple functions of $\mathbf{U}^{[3]}$ given by Eq. 44. The quantities written with an overhead dot ($\dot{}$) in Eq. 106 are evaluated after the predictor step where the predictor step produces

$$\dot{\mathbf{U}}_j^{[3]} = \begin{pmatrix} (\dot{m}_p)_j \\ \\ \left(\dot{h}_p\right)_j \end{pmatrix} \quad \text{at node } j \tag{108}$$

from which the vectors $\dot{\mathbf{F}}_j^{[3]}$ and $\dot{\mathbf{G}}_j^{[3]}$ can be calculated straightforwardly using Eq. 45 and 46 since there is no derivative term involved in these two equations.

The description of the scheme for marching the computation from the present time level t to the next time level $t^* = t + \Delta t$ for Region 3 is divided into two parts: (1) the general scheme for the nodes j=(s^*),(s^*+1),...,$(w-1)$ to be explained in Section 2.5.1, and (2) the landward boundary condition to find the new node w^* to be described in Section 2.5.2. The symbol s^* denotes the upper waterline node at the next time level $t^* = t + \Delta t$, w the lower waterline node at the present time level t, and w^* the lower waterline node at the next time level $t^* = t + \Delta t$. Section 2.5.3 discusses wave run-up, run-down, and set-up on the lower slope.

2.5.1 GENERAL SCHEME

1. Following the adjustment for the value of $(h_p)_j$ for j=s^* taken in Step 2 in Section 2.4, the vector $\mathbf{U}^{[3]}$ at the present time level t is now

$$\mathbf{U}_j^{[3]} = \begin{pmatrix} (m_p)_j \\ \\ b_j \end{pmatrix} \quad \text{for } j = s^* \tag{109}$$

2. Compute $\dot{\mathbf{U}}_j^{[3]}$ using the predictor step Eq. 105 for j=(s^*+1),(s^*+2),...,w. This step, along with the predictor step for Region 2, is illustrated in Figure 3a where the black dots and open circles correspond to the nodes in Regions 2 and 3, respectively, and the open square represents the special treatment for the node j=s^*.

3. For j=s^* set

$$\dot{\mathbf{U}}_j^{[3]} = \begin{pmatrix} \widehat{\mathbf{U}}_j^{[2]} \\ \\ b_j \end{pmatrix} \quad \text{for } j = s^* \tag{110}$$

where $\widehat{\mathbf{U}}_j^{[2]}$ for $j = s^*$ is given by Eq. 104 from Region 2.

4. Compute $\ddot{\mathbf{U}}_j^{[3]}$ using the corrector step Eq. 106 for j=(s^*),(s^*+1),...,$(w-1)$. This step, along with the corrector step for Region 2 is illustrated in Figure 3b.

5. Compute $\mathbf{U}_j^{[3]*}$ using Eq. 107 for j=(s^*),(s^*+1),...,$(w-1)$, as illustrated in Figure 3c.

6. The final value of $\mathbf{U}_j^{[2]*}$ for $j = s^*$ (see Step 5 of Section 2.4) is given by:

$$\mathbf{U}_j^{[2]*} = \left(U_1^{[3]*}\right)_j \quad \text{for } j = s^* \tag{111}$$

where $\left(U_1^{[3]*}\right)_j$ is the first element of the vector $\mathbf{U}_j^{[3]*}$ computed in Step 5.

2.5.2 Landward Boundary Condition

The landward boundary for Region 3 is at the moving waterline on the lower slope where the water depth h_p is essentially zero. Computationally, the landward boundary node for Region 3 is located using Eq. 56 and the following steps:

1. Compute
   $(h_p)_{w+1} = 2\,(h_p)_w - (h_p)_{w-1}$ and
   $(u_p)_{w+1} = 2\,(u_p)_w - (u_p)_{w-1}$.
   If $(h_p)_{w+1} < 0$, set $(h_p)_{w+1} = 0$ and $(u_p)_{w+1} = 0$.
   Obtain $(m_p)_{w+1} = p_u\,(h_p)_{w+1}\,(u_p)_{w+1}$.

2. Compute $(h_p)_j^*$ and $(m_p)_j^*$ at the time t^* for the node $j = w$, using Eq. 107

3. If $(h_p)_{w-1}^* \le \delta_w$, abort the computation.
   Otherwise proceed.

4. If $(h_p)_w^* > (h_p)_{w-1}^*$, compute
   $\widehat{(h_p)}_w^* = 2\,(h_p)_{w-1}^* - (h_p)_{w-2}^*$ and
   $\widehat{(u_p)}_w^* = 2\,(u_p)_{w-1}^* - (u_p)_{w-2}^*$.
   If $\widehat{(h_p)}_w^* \le 0$, set $w^* = (w-1)$ and go to Step 7.
   If $\widehat{(h_p)}_w^* > 0$, ensure that the water depth and the magnitude of u_p near the waterline decrease landward:

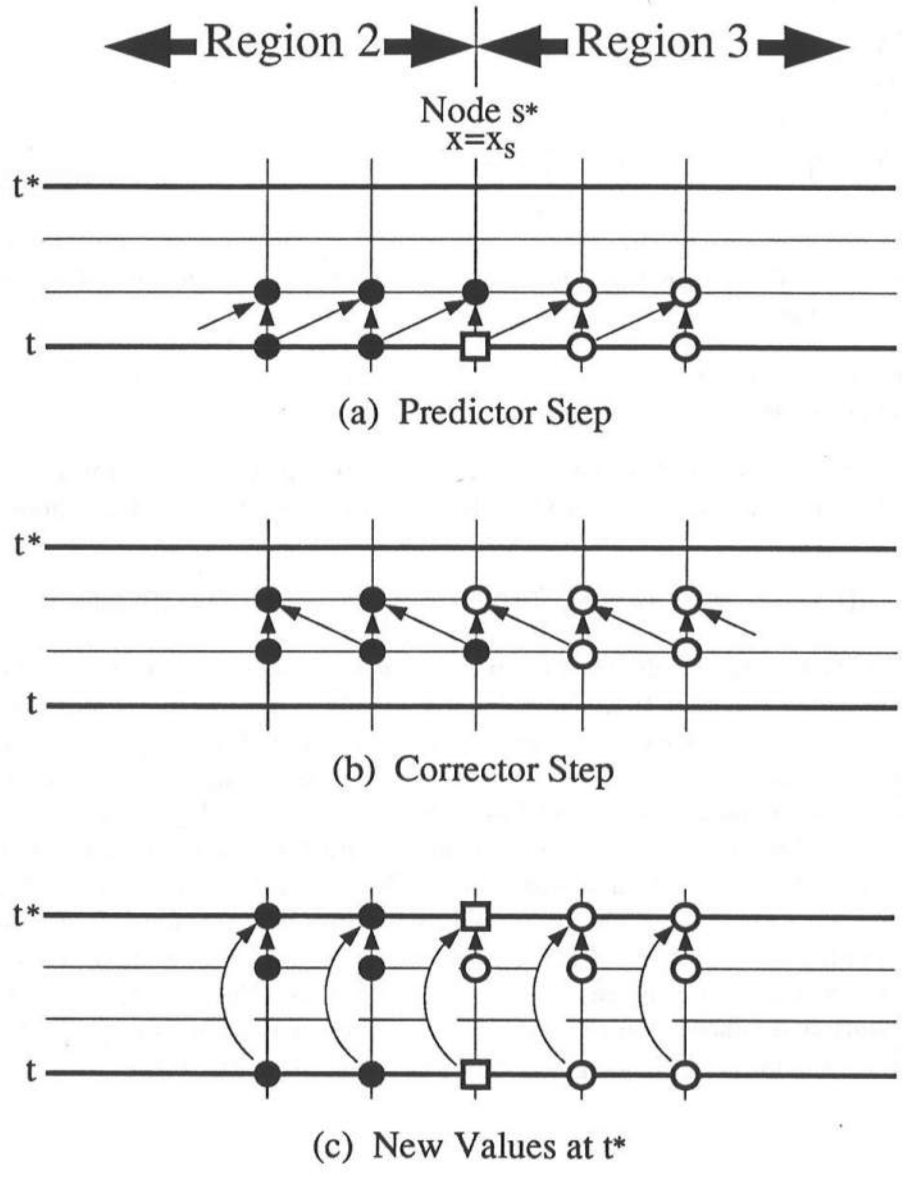


Figure 3: Diagram of time marching computation near the border between Regions 2 and 3.

If $\widehat{(h_p)}^*_w > (h_p)^*_{w-1}$, set $(h_p)^*_w = 0.9\,(h_p)^*_{w-1}$. Otherwise set $(h_p)^*_w = \widehat{(h_p)}^*_w$.
If $|\widehat{(u_p)}^*_w| > |\,(u_p)^*_{w-1}\,|$, set $(u_p)^*_w = 0.9\,(u_p)^*_{w-1}$. Otherwise set $(u_p)^*_w = \widehat{(u_p)}^*_w$.
Obtain $(m_p)^*_w = p_u\,(h_p)^*_w\,(u_p)^*_w$.

5. If $(h_p)^*_w \le \delta_w$, set $w^* = (w-1)$ and go to Step 7.
   Otherwise, compute
   $(h_p)^*_{w+1} = 2\,(h_p)^*_w - (h_p)^*_{w-1}$,
   $(u_p)^*_{w+1} = 2\,(u_p)^*_w - (u_p)^*_{w-1}$, and
   $(m_p)^*_{w+1} = p_u\,(h_p)^*_{w+1}\,(u_p)^*_{w+1}$.
   The use of linear extrapolation to find $(h_p)^*_{w+1}$ and $(u_p)^*_{w+1}$ and the measures taken in Step 4 should ensure that the water depth and the magnitude of u_p near the waterline decrease landward.

6. If $(h_p)^*_{w+1} \le \delta_w$, set $w^* = w$.
   Otherwise, set $w^* = (w+1)$.

7. After $w*$ is obtained, set $(h_p)^*_j = 0$, $(u_p)^*_j = 0$, and $(m_p)^*_j = 0$ for $j \ge (w^*+1)$ since no water is present on the lower slope landward of the lower waterline node.

2.5.3 Run-up, Run-down, and Set-up on the Lower Slope

After the normalized depth of the internal flow, h_p, at the given time is known as a function of x, the normalized water table elevation $Z^L_r = (Z^L_r)'/H'$, where the physical water depth on the lower slope equals a specified value δ'_r, can be computed as long as $\delta_r = (\delta'_r/H') > \delta_w$. The use of the physical depth δ'_r might be related to the use of a hypothetical meter to measure the internal water table oscillation on the lower slope. The specified depth δ'_r can be regarded as the vertical distance between the hypothetical meter and the lower slope, while the corresponding elevation $(Z^L_r)'$ is the elevation above SWL of the intersection between the hypothetical meter and the internal water table. The computed oscillations of $Z^L_r(t)$ for different values of δ'_r can be used to examine the sensitivity to δ'_r of wave run-up, run-down, and set-up on the lower slope. The terms run-up, run-down, and set-up used here follow the definitions given in Section 2.3.4 for the waterline oscillation on the upper slope. The normalized run-up R^L, run-down R^L_d, and set-up $\overline{Z^L_r}$ on the lower slope for given δ'_r are obtained from the computed oscillation of $Z^L_r(t)$.

• 2.6 •
Statistical Calculations

The *statistical calculations* in this report imply the calculations of the time-averaged (mean) and the extreme (maximum and minimum) values of computed time series. The statistical calculations are performed over the time span from t_{min} to t_{max}, where t_{min} and t_{max} are specified by the user and explained more below. In the subsequent sections, the time-averaged (mean) value will be denoted by an overbar, whereas the maximum and minimum values are indicated by the subscripts $_{max}$ and $_{min}$.

The following definition of time average is adopted in this report.

$$\overline{V} = \frac{1}{t_{max} - t_{min}} \int_{t_{min}}^{t_{max}} V(t) dt \qquad (112)$$

where

V = computed time-varying quantity
t_{min} = normalized time when the statistical calculation begins
t_{max} = normalized time when the statistical calculation ends

It is noted that t_{max} also denotes the computation duration starting from $t = 0$. The statistical calculations will be performed over the entire computation duration if $t_{min} = 0$.

• 2.7 •
Mass Balance

Utilizing Eqs. 32 and 47, Eqs. 26 and 28 may be rewritten as follows:

$$p_q q_b - n_p \frac{\partial m_p}{\partial x} = 0 \qquad (113)$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial m_p}{\partial x} = 0 \qquad (114)$$

Integrating Eqs. 24, 113, and 114 using the definition of the time average given by Eq. 112 yields the following time-averaged equations of mass:

Region 1:

$$\frac{d\overline{m}}{dx} + p_q \overline{q_b} + \frac{h(t_{max}) - h(t_{min})}{t_{max} - t_{min}} = 0 \qquad (115)$$

Region 2:

$$p_q \overline{q_b} - n_p \frac{d\overline{m_p}}{dx} = 0 \qquad (116)$$

Region 3:

$$\frac{d\overline{m_p}}{dx} + \frac{\eta(t_{max}) - \eta(t_{min})}{t_{max} - t_{min}} = 0 \qquad (117)$$

For random waves, the computation duration is typically long and $(t_{max} - t_{min})$ is much greater than unity. The third term of Eq. 115 and the second term of Eq. 117 are then negligible. For such cases, Eq. 115 may be reduced to Eq. 118 for Region 1 whereas Eq. 117 may be simplified and combined with Eq. 116 to form Eq. 119 as shown below:

Region 1:

$$\frac{d\overline{m}}{dx} + p_q \overline{q_b} = 0 \quad \text{for } (t_{max} - t_{min}) >> 1 \qquad (118)$$

REGIONS 2+3:

$$p_q\overline{q_b} - n_p\frac{d\overline{m_p}}{dx} = 0 \qquad \text{for } (t_{max} - t_{min}) >> 1 \tag{119}$$

It is noted that the volume influx q_b for Region 3 is zero as given by Eq. 50.

• 2.8 •
WAVE ENERGY BALANCE

The normalized equations of mass and x-momentum given by Eqs. 24 through 29 are used to compute the flow fields in the three regions of the computation domain. The normalized energy equations corresponding to those equations are derived by Wurjanto and Kobayashi (1992a) and expressed as follows:

REGION 1:

$$\frac{\partial E}{\partial t} + \frac{\partial F}{\partial x} = -D - D_p \tag{120}$$

with

$$E = \begin{cases} \frac{1}{2}(hu^2 + \eta^2) & \text{for } h \geq \eta \\ \frac{1}{2}\left[hu^2 + \eta^2 - (h-\eta)^2\right] & \text{for } h < \eta \end{cases} \tag{121}$$

$$F = hu\left(\frac{1}{2}u^2 + \eta\right) \tag{122}$$

$$D_p = p_q q_b\left(\frac{1}{2}u_b^2 + \eta\right) \tag{123}$$

REGION 2:

$$\frac{\partial E_p}{\partial t} + \frac{\partial F_p}{\partial x} = D_p - D_r \tag{124}$$

with

$$E_p = \begin{cases} \frac{1}{2}n_p p_u^2 u_p^2 h_p & \text{for } z_b \leq 0 \\ \frac{1}{2}n_p\left(p_u^2 u_p^2 h_p + z_b^2\right) & \text{for } z_b > 0 \end{cases} \tag{125}$$

$$F_p = p_q u_p h_p\left(\frac{1}{2}p_u^2 u_p^2 + \eta\right) \tag{126}$$

$$D_r = p_q\left(|u_p| + \mu\right)u_p^2 h_p \tag{127}$$

REGION 3:

$$\frac{\partial E_p}{\partial t} + \frac{\partial F_p}{\partial x} = -D_r \tag{128}$$

with

$$E_p = \begin{cases} \frac{1}{2} n_p \left(p_u^2 u_p^2 h_p + \eta^2 \right) & \text{for } z_p \le 0 \\ \frac{1}{2} n_p \left(p_u^2 u_p^2 h_p + \eta^2 - z_p^2 \right) & \text{for } z_p > 0 \end{cases} \tag{129}$$

$$D_p = 0 \qquad \text{for REGION 3} \tag{130}$$

The symbols representing the energy quantities involved in the above equations are explained below:

- E = normalized specific energy, defined as the sum of kinetic and potential energy per unit horizontal area, of the external flow
- F = normalized horizontal energy flux per unit width of the external flow
- D_p = normalized vertical energy flux per unit horizontal area from the external flow to the internal flow
- D = normalized rate of energy dissipation per unit horizontal area of the external flow
- E_p = normalized specific energy of the internal flow
- F_p = normalized horizontal energy flux per unit width of the internal flow
- D_r = normalized rate of energy dissipation per unit horizontal area due to the laminar and turbulent flow resistance

The dimensional counterparts of the normalized specific energy, horizontal energy flux, and rate of energy dissipation listed above can be obtained by multiplying the normalized quantities by $(\rho g H'^2)$, $(\rho g \sqrt{gH'} H'^2)$, and $(\rho g H'^2/T')$, respectively. The energy flux D_p' is normalized by $(\rho g H'^2/T')$. The symbol ρ represents the fluid density, which is assumed to be constant neglecting air bubbles. The normalized potential energy is taken to be relative to the normalized potential energy at the time $t = 0$ when the incident wave train arrives at $x = 0$ and no wave action exists in the region of $x \ge 0$.

Integrating Eqs. 120, 124, and 128 using the definition of the time average given by Eq. 112 yields the following time-averaged equations of energy:

REGION 1:

$$\overline{D} = -\frac{d\overline{F}}{dx} - \overline{D_p} - \frac{E(t_{max}) - E(t_{min})}{t_{max} - t_{min}} \tag{131}$$

REGION 2:

$$\frac{d\overline{F_p}}{dx} + \overline{D_r} - \overline{D_p} + \frac{E_p(t_{max}) - E_p(t_{min})}{t_{max} - t_{min}} = 0 \tag{132}$$

REGION 3:

$$\frac{d\overline{F_p}}{dx} + \overline{D_r} + \frac{E_p(t_{max}) - E_p(t_{min})}{t_{max} - t_{min}} = 0 \tag{133}$$

where the normalized time levels t_{min} and t_{max} have been introduced in Section 2.6.

In this numerical model, Eq. 131 is used to estimate the normalized time-averaged rate of energy dissipation per unit horizontal area of the external flow, $\overline{D}$.

Similar to the discussion in Section 2.7 regarding mass balance, for a computation of long duration with a large value of $(t_{max} - t_{min})$, the terms involving the integral limits t_{min} and

t_{max} in Eqs. 131 through 133 are negligible. For such cases, Eqs. 132 and 133 may be combined and written as

$$\frac{d\overline{F_p}}{dx} + \overline{D_r} - \overline{D_p} = 0 \qquad \text{for Regions 2+3} \tag{134}$$

It is noted that the energy flux D_p for Region 3 is zero as stated in Eq. 130.

• 2.9 •

Hydraulic Stability of Armor Units

The hydraulic stability of armor units is analyzed using the computed flow field of Region 1. The following procedure is the same as that of Kobayashi and Wurjanto (1990) for the case of a *thin* permeable underlayer. The permeability in the present problem is taken into account through the fluid acceleration expressed as

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -\frac{\partial h}{\partial x} - \theta - \frac{f|u|u}{h} + \frac{p_q q_b(u - u_b)}{h} \tag{135}$$

where use is made of Eqs. 24 and 25. The horizontal velocity u is also affected by the presence of the permeable underlayer since Eqs. 24 and 25 include the mass and momentum fluxes into and out of the thick permeable underlayer. No attempt has been made in this simplified approach to consider the direct effects of the water flowing in and out of the thick permeable underlayer.

The drag, lift, and inertia forces acting on individual armor units may be expressed in terms of the fluid velocity u and acceleration du/dt of the flow of Region 1. Kobayashi and Otta (1987) expressed the stability condition against sliding or rolling of an armor unit located on the slope with its local slope angle θ', as shown in Figure 1, in the following form:

$$|N_s + E_1| + E_2 N_s \leq E_3 \tag{136}$$

which is applicable for the case of $u \neq 0$. The stability number N_s in Eq. 136 is defined as:

$$N_s = \frac{H'}{[W'/(\rho s_g)]^{1/3}(s_g - 1)} \tag{137}$$

where

H' = reference wave height used for the normalization of the governing equations
s_g = specific gravity of the armor unit whose unit mass is given by ρs_g
W' = median mass of the armor unit

If the stability number N_s required for the hydraulic stability of armor units is known, the required mass W' can be found using Eq. 137 for given H'. E_1, E_2, and E_3 in Eq. 136 are defined as

$$E_1 = \frac{2C_3^{2/3}}{C_2 C_D|u|u}\left[\frac{C_M}{(s_g - 1)\sigma}\ \frac{du}{dt} - \sin\theta'\right] \tag{138}$$

$$E_2 = \frac{C_L\ \tan\phi}{C_D} \tag{139}$$

$$E_3 = \frac{2C_3^{2/3}}{C_2 C_D u^2}\ \cos\theta'\ \tan\phi \tag{140}$$

where

C_D = drag coefficient
C_L = lift coefficient
C_M = inertia coefficient
C_2 = area coefficient
C_3 = volume coefficient
ϕ = frictional angle of armor units

Eq. 136 can be solved in terms of N_s

$$N_s \leq N_R(t,x) = \begin{cases} (E_1 + E_3)/(E_2 - 1) & \text{if } E_1 < 0,\ E_2 > 1, \text{ and } E_3 < (-E_1 E_2) \\ \\ (E_3 - E_1)/(E_2 + 1) & \text{otherwise} \end{cases} \tag{141}$$

where N_R = dimensionless function expressing the degree of the armor unit stability as a function of t and x. For the computation, Eq. 141 is used if $|u| \geq 10^{-3}$ and N_R is set to be $N_R = 1000$ if $|u| < 10^{-3}$.

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

$$\left| \frac{C_M}{(s_g - 1)\sigma} \ \frac{du}{dt} - \sin\theta' \right| \leq \cos\theta' \ \tan\phi \tag{142}$$

The condition given by Eq. 142 is satisfied if the normalized fluid acceleration remain within the following lower and upper bounds

$$\sigma \ a_{min} \ \leq \ \frac{du}{dt} \ \leq \ \sigma \ a_{max} \tag{143}$$

with

$$a_{min} \geq \ - \frac{s_g - 1}{C_M} \ \frac{sin(\phi - \theta')}{cos\phi} \quad ; \quad a_{max} \leq \frac{s_g - 1}{C_M} \ \frac{sin(\phi + \theta')}{cos\phi} \tag{144}$$

In terms of the dimensional variables, Eq. 143 can be rewritten as

$$g a_{min} \leq \frac{du'}{dt'} \leq g a_{max} \tag{145}$$

where g is the gravitational acceleration. The dimensionless parameters a_{min} and a_{max} need to be chosen so as to satisfy the conditions given by Eq. 144.

The local stability number $N_{sx}(x)$ for initiation of armor movement at given location x is defined as the minimum value of $N_R(t,x)$ at the same location for a specified duration. If $N_s \leq N_{sx}(x)$, the armor unit located at given x will not move during the specified duration. The critical stability number N_{sc} for initiation of armor movement is defined as the minimum value of $N_{sx}(x)$ with respect to x in the computation domain. If $N_s \leq N_{sc}$, no armor units in the computation domain will move during the specified duration. In the computer program PBREAK, the duration of the armor stability computation is taken to be identical to the duration from $t = t_{min}$ to $t = t_{max}$ used for the statistical calculations as described in Section 2.6.

PART III

COMPUTER PROGRAM PBREAK

• 3.1 •

MODES OF COMPUTATION

This section provides an overview of the ways to execute the computer program PBREAK in light of numerical problems that may be encountered. The computer program PBREAK may be seen as a system composed of the five subsystems:

- Hydrodynamics: Sections 2.2 through 2.5.
- Statistics: Section 2.6.
- Mass and energy balances: Sections 2.7 and 2.8.
- Armor stability: Section 2.9.
- Documentation: this refers to the parts of the program that write output data and essential information.

Learning from our experiences in developing the computer program RBREAK (Wurjanto and Kobayashi 1991), numerical problems in the *hydrodynamic* computation of the computer program PBREAK are anticipated.

We have identified six computation parameters which are instrumental to the numerical stability of PBREAK computations. Section 3.7 lists the six instrumental computation parameters, which hereafter will be referred to as "ICPs" for brevity. Numerical problems can be minimized by adjusting these ICPs during the hydrodynamic computation. For this reason, the computation duration is divided into smaller divisions as explained in the following:

- Computation is from the initial time $t = 0$ to the normalized terminal time $t = t_{max}$. It is noted that in terms of the normalized time t, the reference wave period is unity.
- The computation duration is divided into *computation units* which are identified by the integer `M` in the computer program PBREAK. Each computation unit is one reference wave period long except the last unit, which may be shorter than one reference wave period.
- Depending on the numerical problems encountered, *computation units* may be grouped into a number of computation *sequences*. Each sequence consists of a number of sets of five computation units except the last sequence, which can be of any length.

- The values of the ICPs may vary during the computation. Some ICPs are allowed to vary by the sequence, and others may vary by the computation unit, or even within each computation unit as will be explained in Section 3.7.

The question is, how to adjust the ICPs. In developing remedial measures to overcome potential numerical problems listed in Section 3.8, we have considered an automated adjustment procedure. To be effective but also flexible, however, we have eventually adopted a semi-automated remedial procedure that under certain circumstances calls for a manual intervention by the user.

For an easy reference, the part of the computer program PBREAK that performs the adjustment of the ICPs is called the *diagnostic* part. Only the hydrodynamic subsystem is involved in the diagnostic part. The diagnostic part is designed to assign a set of the ICPs that lead to a successful computation of each computation unit. After such a set of the ICPs has been found for a computation unit, computation for all the five subsystems for that unit can be performed. The *actual* part is the term used for the part of the computer program PBREAK that performs computation for all the five subsystems with the ICPs provided by the diagnostic part.

In any case, the actual part for a computation unit must be preceded by its corresponding diagnostic part. There are two options in handling the succession of the *diagnostic-actual* parts as depicted in Figure 4 and explained further below.

1. *By the whole computation duration.*

   In this method, the diagnostic part is first completed for the *whole* computation duration from $t = 0$ to $t = t_{max}$. The diagnostic part in this method needs to start with the computation mode `ICOMP=1` in the computer program PBREAK. This first sequence may break down before t_{max} is reached. The user may then either

   - proceed to the next sequence $\Longrightarrow$ select the computation mode `ICOMP=2`, or
   - start all over again from the initial time $t = 0 \Longrightarrow$ select the computation mode `ICOMP=1`.

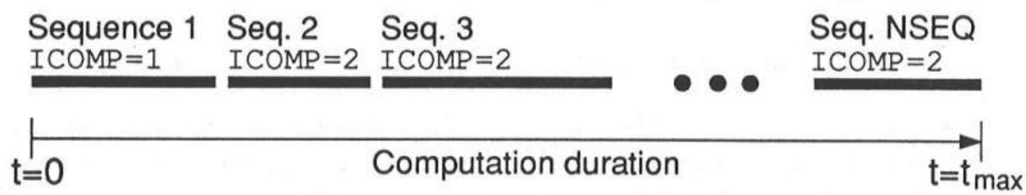
   In either way, the user will need to modify some or all of the ICP values in the new computation.
   A computation with `ICOMP=2` may break down before t_{max} is reached, in which case the user will have the options similar to those described above for `ICOMP=1`. It is now clear that the option `ICOMP=2` will allow the user to overcome numerical difficulties in sequence manually.

   After the diagnostic part has been completed from $t = 0$ to $t = t_{max}$, the actual part is executed from $t = 0$ to $t = t_{max}$. The actual part in this method is performed using the computation mode `ICOMP=3`, which is allowed only after the diagnostic part has been completed for the whole computation duration. A computation with `ICOMP=3` should not encounter any hydrodynamic-related numerical problem if its corresponding diagnostic part (`ICOMP=1,2`) has been run properly.

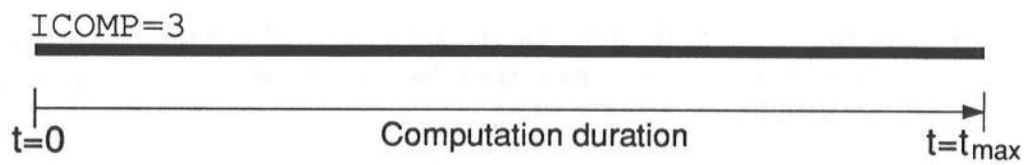
Options in Handling the Succession Diagnostic-Actual Parts of PBREAK

Option 1: By the whole computation duration

- Step A: Execute the diagnostic part from t=0 to $t=t_{max}$



- Step B: Execute the actual part from t=0 to $t=t_{max}$



Option 2: By the computation unit

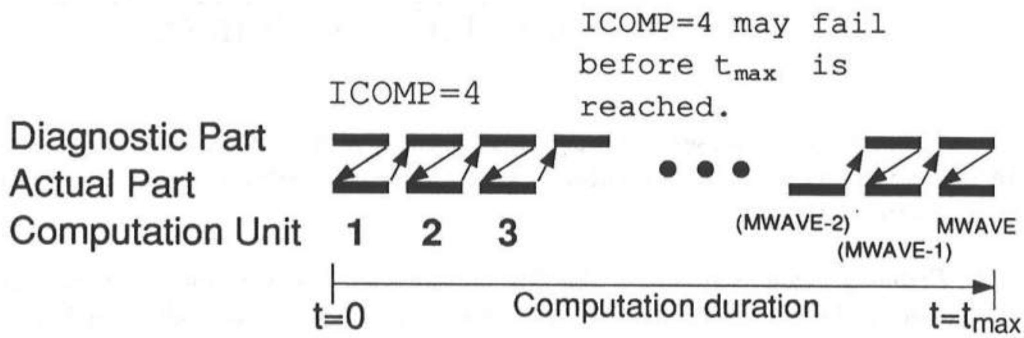


Figure 4

2. *By the computation unit.*

   In this method, upon completion of the diagnostic part of a computation unit, the actual part for that unit is performed immediately. The computation then proceeds to the next computation unit where the succession of the diagnostic-actual parts is repeated. This method is used by specifying the computation mode `ICOMP=4`. A computation with `ICOMP=4` may break down before t_{max} is reached. There is no remedial measure for a failed computation with `ICOMP=4`. The user will have to start all over again from the initial time $t = 0$.
   If `ICOMP=4` is chosen again, the user will face the same risk of computation failure. For this reason, we do not recommend the option `ICOMP=4` for simulating the case involving irregular waves of long duration.

We caution the user regarding the following aspects of computation:

- Our experience in working with two different computer systems suggests that computation sequencing (`ICOMP=1,2`) is sensitive to the floating-point processor unit (FPU). Executing PBREAK on different machines may require different sequencing of the diagnostic part.

- In light of this sensitivity, the diagnostic (`ICOMP=1,2`) and actual (`ICOMP=3`) parts of a PBREAK computation should be executed on the *same* machine.

- Since the values of the ICPs for the actual part of a PBREAK computation have been provided by the diagnostic part, the values of ICPs specified in the primary input data file are not used for the actual part.

• 3.2 •

INPUT DATA FILES FOR PBREAK

This section discusses the input data files for a PBREAK computation and how the input items must be arranged in those files. Two data files are *initially* needed to execute the computer program PBREAK.

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

2. The second input data file containing *input wave train.* The *input wave train* is the prescribed time series of the *incident* wave profile at the seaward boundary. We emphasize the word *incident* here because PBREAK accepts only *incident* waves as input at the seaward boundary. It is also noted that PBREAK does not compute incident regular wave profiles at the seaward boundary. In *any* case, the input wave train must be prepared by the user prior to executing PBREAK. It may be worth mentioning here that the computer program RBREAK (Wurjanto and Kobayashi 1991) includes regular wave generators. A user could hence utilize RBREAK to generate a regular wave train.

For a computation with ICOMP=1 or ICOMP=4, the above two files are all that the user needs, as depicted in Figures 5 and 7. For an ICOMP=2 or ICOMP=3, on the other hand, additional input data files are needed as depicted in Figures 5 and 6. However, the user does not have to be concerned about these additional input files since they are self-managed by PBREAK as long as the procedure for the sequential computation depicted in Figure 4 is properly followed. What the user needs to do between the computation sequences is to manually adjust one or more of the ICPs in the primary input data file. This manual adjustment is described in concept in Section 3.7 and explained further by examples in Appendices B and C.

There are two input items that the user has to enter interactively at the beginning of a PBREAK computation: **(1)** the option ICOMP described in Section 3.1 and **(2)** the name of the primary input data file. The name of the second input data file is specified in the primary input data file. The names of the first and second input data files are to be read by PBREAK as the variables **FINP1** and **FINP2**, respectively. These two names should consist of no more than ten characters each.

In the following Section 3.2.1 lists the input items in the primary input data file in the order they are read by Subroutine 04 **INPUT1** of PBREAK, and Section 3.2.2 explains the contents of the second input data file. A list of the subroutines of the computer program PBREAK is presented in Section 3.4.

3.2.1 Input Items in the Primary Input Data File

The primary input data file has a header in which a user may write pertinent information or comments. The header is read by Subroutine 04 **INPUT1** in the following order.

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

where **NLINES** is the number of lines containing the user's comments and **COMMEN** is a character array. Thus, the first line in the primary input data file contains the integer to be read as **NLINES** by Subroutine 04 **INPUT1**. The **NLINES** lines containing the user's comments follow immediately. These comments are then written as the header of the output files **ODOC** and **OMSG**. Input parameters and variables are then read in the order to be described below. In the following list: **(1)** **READ** statements are listed on the left, **(2)** Notes written on the right show where the parameter(s) or variable(s) read are explained, and **(3)** Sequential numerals in the middle are provided for convenience. The following list includes six comment lines which are put in the primary input data file to make it easier for the user to visually separate distinct groups of input items. The six comment lines are read as **COMMEN(1)**. The examples featured in Appendices B, C, and D include the presentation of the respective primary input data files.

Diagram of Files in an ICOMP=1,2 Computation

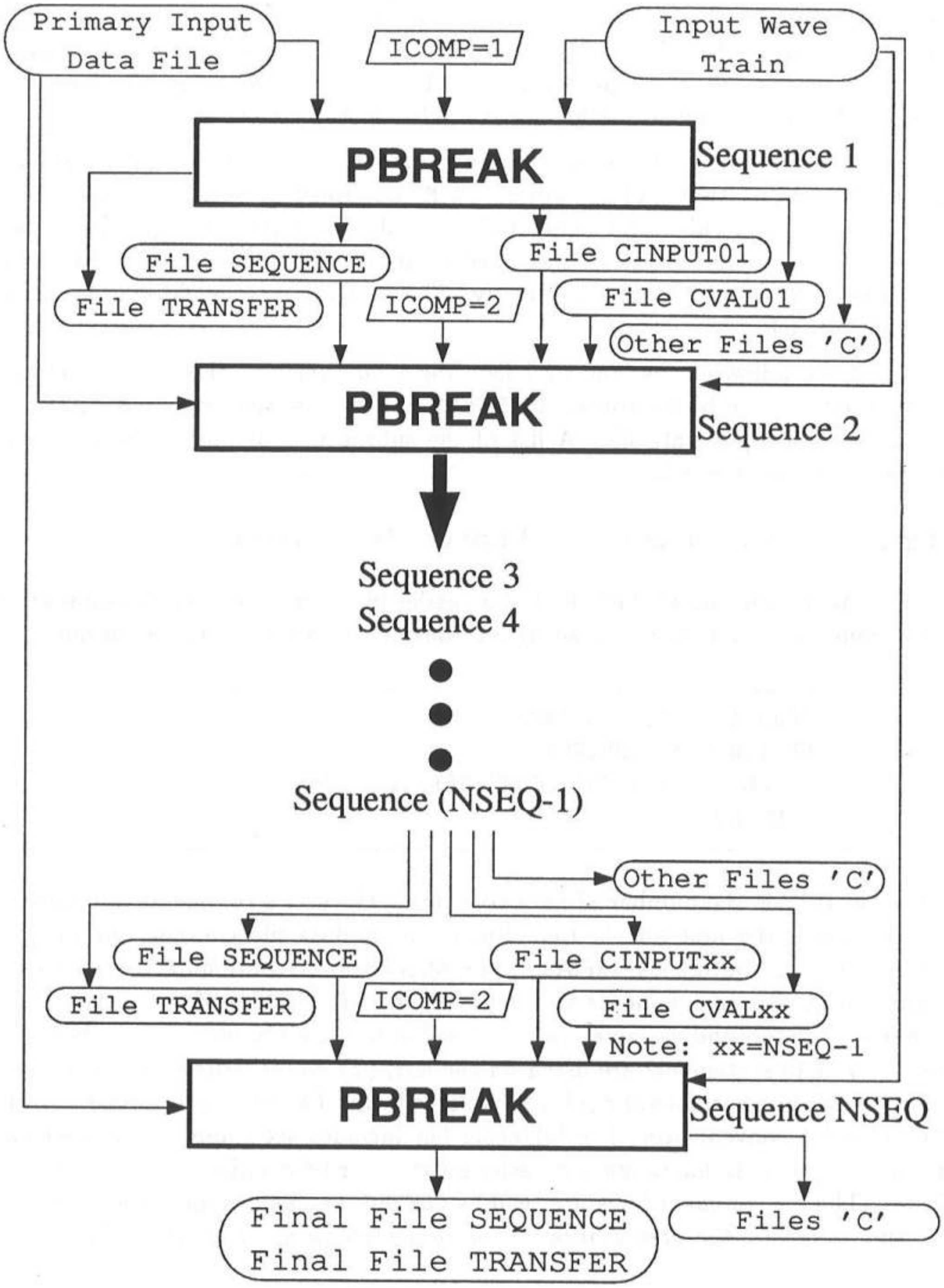


Figure 5

Diagram of Files in an ICOMP=3 Computation

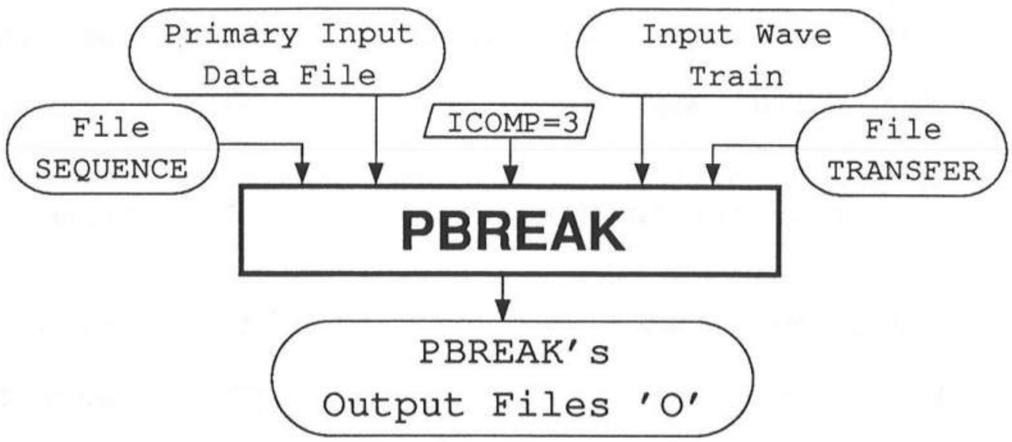


Figure 6

Diagram of Files in an ICOMP=4 Computation

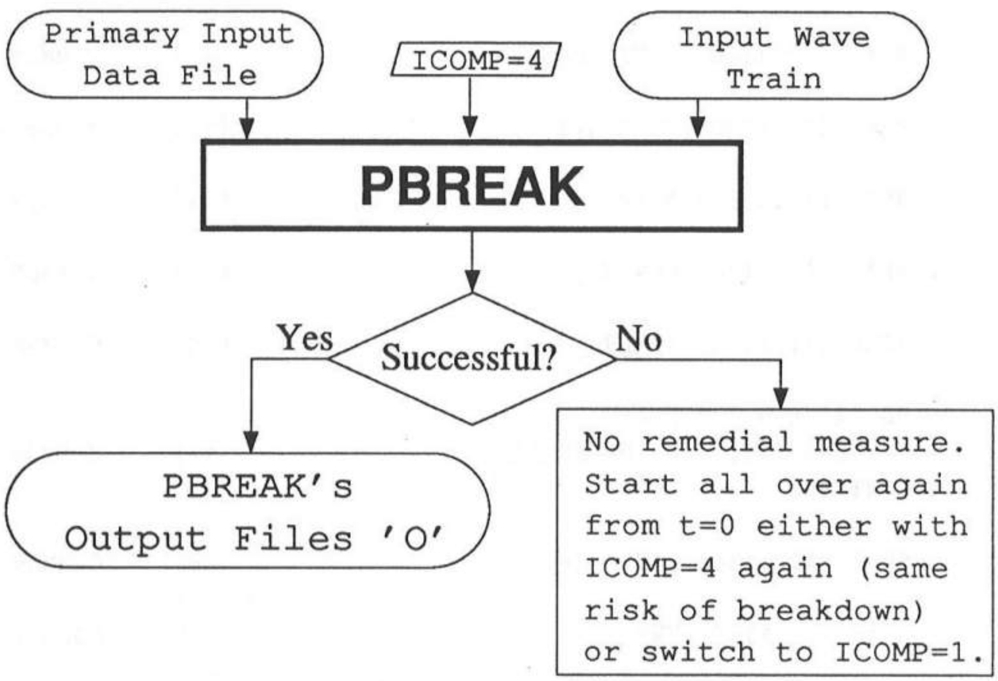


Figure 7

```
    READ (11,1130) ISYST                        [1]     COMMON /OPTION/

    READ (11,1130) IBOT                         [2]     COMMON /OPTION/

    READ (11,1110) MSAVA1,MSAVA2,NTIMES         [3]     COMMON /CPAR1/

    READ (11,1110) NNODB                        [4]     COMMON /SAVB/

    IF (NNODB.GT.0) THEN
       READ (11,1110) (NODB(I),I=1,NNODB)       [5]     COMMON /SAVB/
    ENDIF

    READ (11,5000) FINP2                        [6]     Section 3.2

    READ (11,1150) TMAX                         [7]     COMMON /CPAR2/

    READ (11,1120) COMMEN(1)                    [8]     A comment line

    READ (11,1110) INITS                        [9]     COMMON /CPAR1/

    READ (11,1110) MULTIF                       [10]    COMMON /CPAR1/

    READ (11,1150) DELTAS,DELTAW                [11]    COMMON /CPAR2/

    READ (11,1150) EPSI1,EPSI2                  [12]    COMMON /CPAR2/

    READ (11,1120) COMMEN(1)                    [13]    A comment line

    READ (11,1110) MSTAT                        [14]    COMMON /CPAR1/

    READ (11,1110) NRATE                        [15]    COMMON /CPAR1/

    READ (11,1110) NDELR                        [16]    COMMON /IWLINE/

    DO 120 L = 1,NDELR
       READ (11,1150) DELRP(L)                  [17]    COMMON /DWLINE/
120 CONTINUE

    READ (11,1150) HREFP,TP                     [18]    COMMON /WAVE1/

    READ (11,1150) FWP                          [19]    COMMON /BOT1/

    READ (11,1150) DSEAP                        [20]    COMMON /BOT1/

    READ (11,1150) TSLOPS                       [21]    COMMON /BOT2/
```

```
      READ (11,1110) NUSEG                       [22]     COMMON /BOT3/

      IF (IBOT.EQ.1) THEN
         DO 130 K = 1,NUSEG
            READ (11,1150) WUSEG(K),TUSEG(K)     [23]     COMMON /BOT4/
130      CONTINUE
      ELSE
         DO 140 K = 1,NUSEG+1
            READ (11,1150) XUSEG(K),ZUSEG(K)     [24]     COMMON /BOT4/
140      CONTINUE
      ENDIF

      READ (11,1110) NLSEG                       [25]     COMMON /BOT3/

      IF (IBOT.EQ.1) THEN
         DO 150 K = 1,NLSEG
            READ (11,1150) WLSEG(K),TLSEG(K)     [26]     COMMON /BOT4/
150      CONTINUE
      ELSE
         DO 160 K = 1,NLSEG+1
            READ (11,1150) XLSEG(K),ZLSEG(K)     [27]     COMMON /BOT4/
160      CONTINUE
      ENDIF

      READ (11,1120) COMMEN(1)                   [28]     A comment line

      READ (11,1130) ISALT                       [29]     COMMON /OPTION/

      READ (11,1150) TEMPER                      [30]     COMMON /ULAYER/

      READ (11,1150) DPP                         [31]     COMMON /ULAYER/

      READ (11,1150) PORO                        [32]     COMMON /ULAYER/

      READ (11,1150) ALPA0,BETA0                 [33]     COMMON /ULAYER/

      READ (11,1120) COMMEN(1)                   [34]     A comment line

      READ (11,1120) COMMEN(1)                   [35]     A comment line

      READ (11,1150) SG,TANPHI                   [36]     COMMON /ARMOR1/

      READ (11,1150) C2,C3                       [37]     COMMON /ARMOR1/

      READ (11,1150) CD,CL                       [38]     COMMON /ARMOR1/

      READ (11,1150) CM                          [39]     COMMON /ARMOR1/
```

```
READ (11,1150) AMAX,AMIN                [40]    COMMON /ARMOR1/

READ (11,1120) COMMEN(1)                [41]    A comment line
```

The six FORMATs used to read the above input items are listed below.

```
1110 FORMAT (3I8)
1120 FORMAT (15A5)
1130 FORMAT (2I1)
1140 FORMAT (5I6)
1150 FORMAT (2F13.6)
5000 FORMAT (A10)
```

3.2.2 THE SECOND INPUT DATA FILE

The input wave train $\eta_i(t)$ must be prescribed in the *normalized* form, where $\eta_i = \eta_i'/H'$; $t = t'/T'$; H' = reference wave height read as HREFP; and T' = reference wave period read as TP. The input wave train is read by Subroutine 05 INPUT2 as follows:

```
     READ (12,5000) CHAR
     READ (12,9000) NDATA
     READ (12,8000) (ETA(I),I=1,NDATA)
5000 FORMAT (A)
8000 FORMAT (5D15.6)
9000 FORMAT (I8)
```

where

CHAR = a dummy character variable
NDATA = number of data points
ETA = array of size NDATA prescribing the free surface elevation η_i at the seaward boundary

The first READ statement is to read a comment line containing the user's comment. It is *required* that this one comment line exists in the second input data file. If the user does not wish to put any comment, then the first line of the second input data file must be left blank. The data points of ETA read in double precision are equally spaced in time and NDATA is given by

$$\text{NDATA} = \frac{\text{TMAX}}{\delta t} + 1 \qquad (146)$$

where

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

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

The time increment of the input wave train, δt, is normally much larger than the finite difference time step Δt for the stable numerical computation discussed in Part II. The computer program PBREAK performs a simple linear interpolation of the input wave train to get the appropriate value at each time level during the time-marching computation. The interpolation is carried out in Subroutine 36 SEABC.

• 3.3 •

MAIN PROGRAM PBREAK

The main program lists all the important parameters and variables in the COMMON blocks. These parameters and variables are described in Section 3.5. The main program coordinates tasks which are actually executed by subroutines. A list of the subroutines of the computer program PBREAK is presented in Section 3.4. The tasks of the main program can be categorized into four groups.

1. Reading Input Data.
2. Groundwork.
3. Time-marching Computation.
4. Finishing.

The preceding Section 3.2 explains how the computer program PBREAK *reads the input data*. *Groundwork* consists of calculation of the permeable slope geometry and nodal locations, computation of needed parameters, and specification of appropriate initial conditions. *Finishing* job includes documentation of results or, if the sequential procedure is opted, preparation of values to be transferred to the subsequent computation.

The *time-marching computation* depends on the computation mode, ICOMP, selected by the user, as explained to some extent in Section 3.1. The basic procedure for time-marching from one time level to the next is described in Sections 2.2 through 2.5. The diagnostic part of PBREAK is primarily managed by the main program, which has the capability of self-adjusting the computation parameters INVT, DELTAS, and DELTAW explained in COMMON /CPAR1/ and /CPAR2/. This self-adjusting mechanism may fail before the terminal time t_{max} is reached. When a failure occurs in a computation with ICOMP=4, there is no remedial measure. For a computation with ICOMP=1 or ICOMP=2, a failure calls for the user to execute the next sequence of computation using different values of the ICPs. This self-adjusting mechanism is not involved in a computation with ICOMP=3 where all the needed ICPs have been set by its corresponding diagnostic part. See Sections 3.1 and 3.7 for the procedure to execute PBREAK, and Section 3.9 for the self-adjusting mechanism embedded in PBREAK.

• 3.4 •

SUBROUTINES

Program PBREAK consists of a main program, a block data, and 56 subroutines. The 56 subroutines are numbered as listed in Table 1, which indicates the starting page of each subroutine in this report. Each of the subroutines are explained concisely in this section following this format: [Number] – [**NAME**] – [Description], where the [Number] refers to the numerical order assigned to the subroutine in the computer program PBREAK.

01 `OPENIO` manages opening of files.

02 **`MAIN2`** executes the actual time-marching for a computation unit after a working set of the instrumental computation parameters has been established by the diagnostic part.

03 **`KERNEL`** executes subsystems of PBREAK.

04 **`INPUT1`** reads most of input data.

05 **`INPUT2`** reads information on the input wave train.

06 `CVISCO` calculates kinematic viscosity of water using tables from Fischer *et al.* (1979).

07 **`BOTTOM`** calculates normalized structure geometry and the space interval Δx.

08 `BOTA` aids Subroutine 07 **`BOTTOM`** in completing physical segment data not specified as input.

09 `BOTB` aids Subroutine 07 **`BOTTOM`** in calculating physical structure geometry at each node

10 **`PARAM`** calculates parameters.

11 `INIT` specifies the initial conditions at $t = 0$.

12 **`INITSQ`** initializes values at the beginning of a computation with **`ICOMP=2`**.

13 **`INITM`** initializes values at the beginning of a computation unit.

14 **`SAVEM`** saves values at the end of each successful computation unit.

15 **`SAVE5`** saves values at the end of each set of five computation units.

16 **`WRITER`** writes information to be passed on to the next sequence of computation.

17 **`READER`** reads information passed on by the preceding sequence of computation.

18 **`WRSEQ`** writes sequential information into file **`SEQUENCE`** at the end of a computation with `ICOMP=1` or `ICOMP=2`.

19 CINVT assigns a value to INVT and determines NEND for each computation unit where INVT is the number of time steps in one reference wave period and NEND is the final time level for a computation unit. NEND is always equal to INVT except for the last computation unit.

20 CINVTD calculates computation parameters that are dependent on INVT.

21 MARCH1 marches the computation for Region 1 from the present time level to the next.

22 MARCH2 marches the computation for Region 2 from the present time level to the next.

23 MARCH3 marches the computation for Region 3 from the present time level to the next.

24 ROOTH calculates the square root of the water depth h unless there is negative h (IPROB=1).

25 XTRAP1 estimates hydrodynamic quantities of Region 1 at node (IS+1) by linear extrapolation where IS is the upper waterline node s satisfying Eq. 55.

26 XTRAP3 estimates hydrodynamic quantities of Region 3 at node (IW+1) by linear extrapolation where IW is the lower waterline node w satisfying Eq. 56.

27 CQBUB computes QB=q_b and UB=u_b.

28 MA1 computes the elements of the first row of Matrix $\mathbf{A}$ (2x2) given by Eq. 60.

29 VFONE computes the elements of vector $\mathbf{F}^{[1]}$ (2x1) defined by Eq. 39.

30 VGONE computes the elements of vector $\mathbf{G}^{[1]}$ (2x1) defined by Eq. 40.

31 VH computes the elements of vector $\mathbf{H}$ (2x1) given by Eq. 59.

32 VS1 computes the first element of vector $\mathbf{S}$ (2x1) given by Eq. 61.

33 VD computes the elements of vector $\mathbf{D}$ (2x1) given by Eq. 63.

34 VP computes the elements of vector $\mathbf{P}$ (2x1) given by Eq. 68.

35 VUONE computes the elements of vector $\mathbf{U}^{[1]}$ (2x1) defined by Eqs. 38 and given by Eqs. 58.

36 SEABC handles the seaward boundary condition for Region 1 at node 1 as described in Section 2.3.2.

37 LANBC1 handles the landward boundary condition for Region 1 as described in Section 2.3.3.

38 LANBC3 handles the landward boundary condition for Region 3 as described in Section 2.5.2.

39 NUMSTA checks if the numerical stability criterion given by Eq. 75 is violated.

40 VFTWO computes the single element of vector $\mathbf{F}^{[2]}$ (1x1) defined by Eq. 42 and given by Eq. 92.

41 VGTWO computes the single element of vector $\mathbf{G}^{[2]}$ (1x1) defined by Eq. 43 and given by Eqs. 90 or 93.

42 VUTWO computes the single element of vector $\mathbf{U}^{[2]}$ (1x1) defined by Eq. 41 and given by Eqs. 87 or 88.

43 VFTRE1 computes the first element of vector $\mathbf{F}^{[3]}$ (2x1) defined by Eq. 45.

44 VGTRE1 computes the first element of vector $\mathbf{G}^{[3]}$ (2x1) defined by Eq. 46.

45 VUTRE computes the elements of vector $\mathbf{U}^{[3]}$ (2x1) defined by Eq. 44 and given by Eqs. 105 or 106.

46 RUNUP computes run-up heights on the upper and lower slopes discussed in Sections 2.3.4 and 2.5.3.

47 STAT computes statistical values (*i.e.*, mean, maximum, and minimum) with respect to the time span from $t = t_{min}$ to $t = t_{max}$ as explained in Section 2.6.

48 ENERGY computes energy quantities discussed in Section 2.8.

49 SAVET saves the values at $t = t_{min}$ and $t = t_{max}$ used in Sections 2.7 and 2.8.

50 STABNO computes the stability number of armor units explained in Section 2.9.

51 DOC1 writes essential information before time-marching computation.

52 DOC2 stores computed results at designated time levels during time-marching computation.

53 DOC3 documents the results after time-marching computation.

54 NSI calculates the numerical stability indicator Ω defined by Eq. 76.

55 OPENF opens a specified file.

56 STOPP executes a programmed stop.

Table 1: Subroutines of program PBREAK.

No.	Subroutine	Listing Starts on Page	No.	Subroutine	Listing Starts on Page
01	OPENIO	97	29	VFONE	136
02	MAIN2	98	30	VGONE	136
03	KERNEL	100	31	VH	136
04	INPUT1	100	32	VS1	137
05	INPUT2	105	33	VD	138
06	CVISCO	106	34	VP	139
07	BOTTOM	107	35	VUONE	140
08	BOTA	109	36	SEABC	140
09	BOTB	109	37	LANBC1	141
10	PARAM	110	38	LANBC3	142
11	INIT	112	39	NUMSTA	144
12	INITSQ	115	40	VFTWO	145
13	INITM	117	41	VGTWO	145
14	SAVEM	118	42	VUTWO	146
15	SAVE5	119	43	VFTRE1	147
16	WRITER	120	44	VGTRE1	147
17	READER	121	45	VUTRE	148
18	WRSEQ	122	46	RUNUP	148
19	CINVT	122	47	STAT	150
20	CINVTD	124	48	ENERGY	153
21	MARCH1	124	49	SAVET	155
22	MARCH2	127	50	STABNO	157
23	MARCH3	129	51	DOC1	160
24	ROOTH	132	52	DOC2	164
25	XTRAP1	133	53	DOC3	166
26	XTRAP3	134	54	NSI	172
27	CQBUB	134	55	OPENF	172
28	MA1	135	56	STOPP	172

• 3.5 •

PARAMETERS AND VARIABLES IN COMMON BLOCKS

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

/CONSTA/ contains constants.

PI = π = 3.141592...

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

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

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

IBOT indicates the style of input data for the structure geometry. IBOT=1 for the width and slope of linear segments, IBOT=2 for the locations of the seaward end points of linear segments.
See Figure 8 for clarification.

ISALT indicates whether the fluid is fresh water (ISALT=0) or sea water (ISALT=1), because this affects the viscosity of the fluid.

/CPAR1/ contains integer computation parameters.

JE = landward edge node.

INITS = number of the spatial nodes at which the *upper* slope is located below SWL.

INITW = number of the spatial nodes at which the *lower* slope is located below SWL.

MWAVE = number of computation units.

MINVT = the smallest value satisfying the numerical stability criterion Eq. 77 and being divisible by NRATE. MINVT constitutes the minimum allowable value of INVT.

MULTIF = an integer multiplication factor such that (MULTIF*MINVT) constitutes the minimum value of INVT in a sequence of computation.

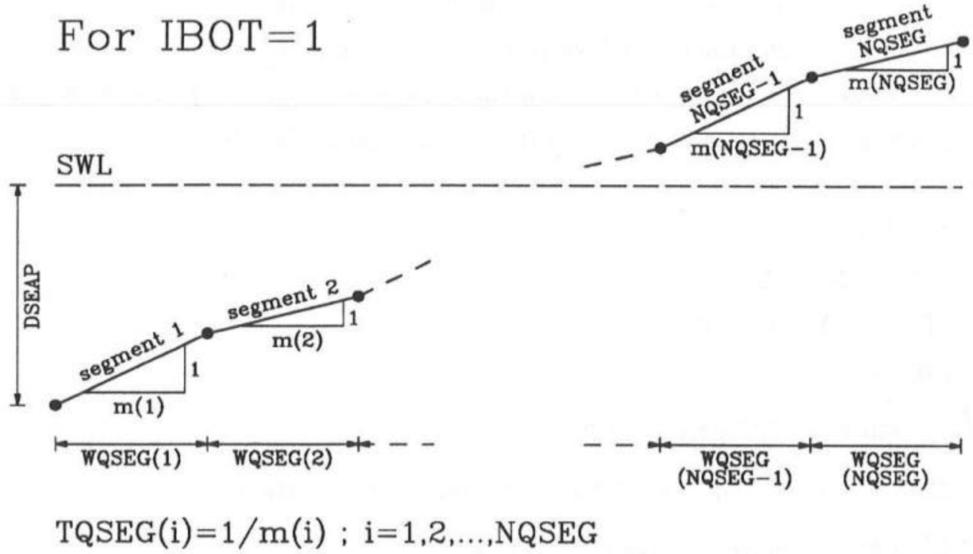
INVT = number of time steps in one wave period = $1/\Delta t$, where Δt is the time step.

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

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

NHOP = INVT/NRATE

MSAVA1, MSAVA2, NTIMES: the spatial variations of six hydrodynamic quantities are stored NTIMES (>1) times at equal intervals per wave period from the computation unit MSAVA1 to MSAVA2, inclusive. See the related explanation in Section 3.6.3.



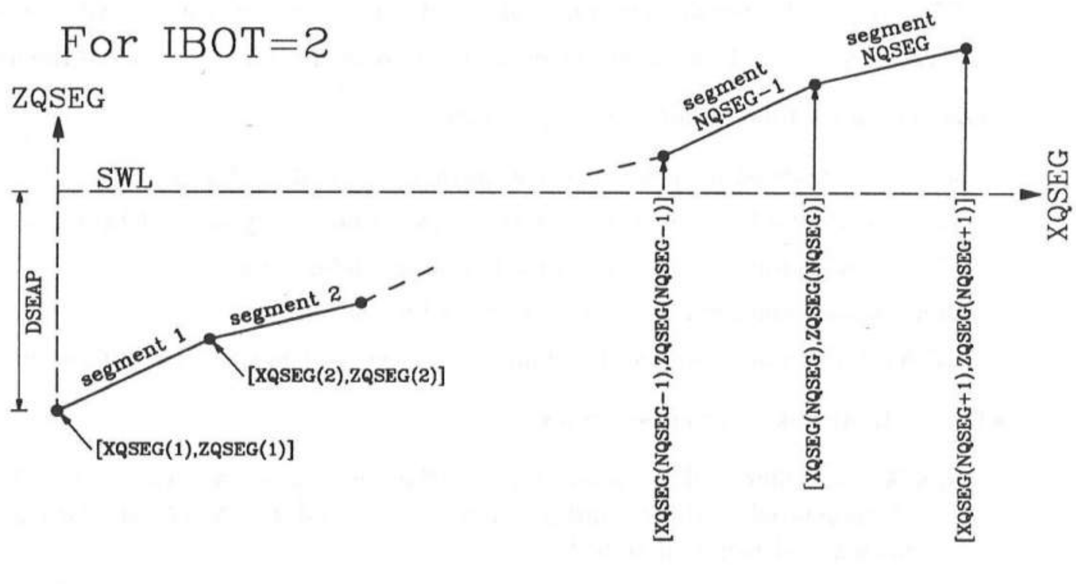


Figure 8: Input for dimensional slope geometry for IBOT=1 and IBOT=2 where Q=U for the upper slope and Q=L for the lower slope.

/CPAR2/ contains double precision computation parameters.

TMAX = normalized duration of the computation.

DELTAS = the parameter δ_s in Eq. 55.

DELTAW = the parameter δ_w in Eq. 56.

EPSI1 = the parameter ϵ_1 involved in Eqs. 65 and 66.

EPSI2 = the parameter ϵ_2 involved in Eqs. 65 and 66.

X = constant space interval Δx in the computation grid depicted in Figure 2.

T = time step Δt in the computation grid depicted in Figure 2.

TX = $\Delta t / \Delta x$

XT = $\Delta x / \Delta t$

TTX = $(\Delta t)^2 / \Delta x$

TTXX = $(\Delta t)^2 / (\Delta x)^2$

TWOX = $2\Delta x$

/TEMPO/ contains TIME described below.

TIME = normalized present time t during a computation.

/WAVE1/ contains physical wave parameters.

HREFP = physical reference wave height H' introduced in Section 2.1. HREFP is specified in *meters* when SI is used or in *feet* when USCS is used.

TP = physical reference wave period T' introduced in Section 2.1. TP is in *seconds*.

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

/WAVE2/ contains dimensionless wave parameters.

WLO = normalized deep-water wavelength L_0 defined by Eq. 51.

WL = normalized wavelength L at the seaward boundary defined by Eq. 52.

UR = Ursell number at the seaward boundary defined by Eq. 54.

SURF = surf similarity parameter defined by Eq. 53.

SIGMA = ratio between the horizontal and vertical length scales defined by Eq. 22.

/IWT1/ contains NDATA described below.

NDATA = number of data points representing the input wave train. The first data point is associated with the initial time $t = 0$, and the NDATA-th data point with the normalized terminal time $t = t_{max}$.

/IWT2/ contains information on the input wave train.

ETA = array of size NDATA prescribing the normalized elevation η_i of the incident wave profile at the seaward boundary.

ETAMA = maximum value of η_i out of the NDATA points.

ETAMI = minimum value of η_i out of the NDATA points.

/BOT1/ contains physical properties of the upper slope.

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

FWP = constant friction factor f' used in Eq. 3.

/BOT2/ contains dimensionless properties of the upper slope.

DSEA = normalized depth d_t defined by Eq. 17.

DSEA2 = $\sqrt{d_t}$

FW = normalized friction factor f defined by Eq. 23.

TSLOPS = tangent of slope $tan\theta'_\xi$ used to define the surf similarity parameter ξ in Eq. 53.

/BOT3/ contains integers related to linear segments specifying the upper and lower slopes. See also the related explanations in COMMONs /OPTION/ and /BOT4/ and Figure 8.

NUSEG = number of linear segments of different inclinations used to specify the upper slope.

NLSEG = number of linear segments of different inclinations used to specify the lower slope.

/BOT4/ contains dimensional quantities associated with the linear segments of the upper and lower slopes. See also the related explanations in COMMONs /OPTION/ and /BOT3/ and Figure 8. All quantities listed below are segmental properties. The indices i and k refer to the segment number of the upper and lower slopes, respectively.
For the option IBOT=1, pairs of [WUSEG(i),TUSEG(i)] and [WLSEG(k),TLSEG(k)] are specified and i=1,2,...,NUSEG and k=1,2,...,NLSEG.
For the option IBOT=2, pairs of [XUSEG(i),ZUSEG(i)] and [XLSEG(k),ZLSEG(k)] are specified and i=1,2,...,(NUSEG+1) and k=1,2,...,(NLSEG+1).

- For the *upper* slope:

  WUSEG(i) = physical horizontal width of the segment i.

  TUSEG(i) = tangent of the slope of the segment i.

  XUSEG(i) = physical horizontal distance from the seaward boundary located at $x' = 0$ to the seaward end of the segment i.

  ZUSEG(i) = physical z'-coordinate of the seaward end of the segment i which is positive if the end point is located above SWL.

- For the *lower* slope:

  WLSEG(k) = physical horizontal width of the segment k.

  TLSEG(k) = tangent of the slope of the segment k.

  XLSEG(k) = physical horizontal distance from the seaward boundary located at $x' = 0$ to the seaward end of the segment k.

  ZLSEG(k) = physical z'-coordinate of the seaward end of the segment k which is positive if the end point is located above SWL.

/BOT5/ contains vectors related to the normalized geometry of the upper and lower slopes.

XUL(j) = normalized x-coordinate of the node j.

ZU(j) = normalized z-coordinate of the upper slope, z_b, at the node j where z_b is given by Eq. 35.

TU(j) = dimensionless gradient θ of the upper slope at the node j where θ is given by Eq. 21.

ZL(j) = normalized z-coordinate of the lower slope, z_p, at the node j where z_p is given by Eq. 36.

TL(j) = dimensionless gradient θ_p of the lower slope at the node j where θ_p is given by Eq. 21.

/ULAYER/ contains properties of the permeable underlayer.

DPP = representative diameter d'_p of the materials composing the permeable underlayer. DPP is specified in *meters* when SI is used or in *feet* when USCS is used.

TEMPER = temperature of water. TEMPER is specified in degrees *Celcius* when SI is used or in degrees *Fahrenheit* when USCS is used.

VISCO = kinematic viscosity of water depending on its temperature and salinity.

ALPAP = coefficient α' given by Eq. 9.

BETAP = coefficient β' given by Eq. 10.

ALPAO = dimensionless constant α_o involved in Eq. 9.

BETAO = dimensionless constant β_o involved in Eq. 10.

PORO = porosity n_p of the permeable underlayer.

PARPU = dimensionless parameter p_u defined by Eq. 33

PARPQ = dimensionless parameter p_q defined by Eq. 32

PARMU = dimensionless parameter μ defined by Eq. 34

/HYDROE/ contains the hydrodynamic quantities of the *external* flow.

UONE = vector $\mathbf{U}^{[1]}$ defined by Eq. 38. The first element of $\mathbf{U}^{[1]}$ at the node j is denoted by UONE(1,j)=m_j where m is given by Eq. 30. The second element of $\mathbf{U}^{[1]}$ at the node j is denoted by UONE(2,j)=h_j where h is defined by Eq. 16.

U = normalized depth-averaged horizontal velocity u defined by Eq. 18.

UB = normalized velocity u_b given by Eq. 31.

C = $\sqrt{h}$.

DUDT = horizontal fluid acceleration du/dt given by Eq. 135.

/HYDROI/ contains the hydrodynamic quantities of the *internal* flow.

HP = normalized depth of the internal flow h_p given by Eq. 16

UP = normalized vertically-averaged horizontal discharge velocity u_p given by Eq. 19.

AMP = normalized volume flux m_p defined by Eq. 47.

/HYDROG/ contains the hydrodynamic quantities related to *both* external and internal flows.

ELEV = normalized elevation η above SWL of the free surface or internal water table.

QB = normalized volume influx q_b per unit horizontal area into the permeable underlayer given by Eq. 49.

/VECONE/ contains the vectors and matrices of Region 1.

FONE = vector $\mathbf{F}^{[1]}$ defined by Eq. 39. The first and second elements of $\mathbf{F}^{[1]}$ at the node j are denoted by FONE(1,j) and FONE(2,j), respectively.

GONE = vector $\mathbf{G}^{[1]}$ defined by Eq. 40. The first and second elements of $\mathbf{G}^{[1]}$ at the node j are denoted by GONE(1,j) and GONE(2,j), respectively.

A1 = a vector containing of the first row of the matrix $\mathbf{A}$ where $\mathbf{A}$ is given by Eq. 60. The variables A1(1,j) and A1(2,j) denotes the elements A_{11} and A_{12}, respectively of the matrix $\mathbf{A}_j$.

H = vector $\mathbf{H}$ given by Eq. 59.
H(1,j) = the first element at the node j.
H(2,j) = the second element at the node j.

S1 = the first element of the vector $\mathbf{S}$ where $\mathbf{S}$ is given by Eq. 61.

D = vector $\mathbf{D}$ given by Eq. 63.
D(1,j) = the first element at the node j.
D(2,j) = the second element at the node j.

P = vector $\mathbf{P}$ given by Eq. 68.
P(1,j) = the first element at the node j.
P(2,j) = the second element at the node j.

/VECTWO/ contains the vectors of Region 2.

FTWO = vector $\mathbf{F}^{[2]}$ defined by Eq. 42.

GTWO = vector $\mathbf{G}^{[2]}$ defined by Eq. 43.

/VECTRE/ contains the vectors of Region 3.

UTRE = vector $\mathbf{U}^{[3]}$ defined by Eq. 44.
UTRE(1,j) = the first element at the node j.
UTRE(2,j) = the second element at the node j.

FTRE1(j) = the first element at the node j of the vector $\mathbf{F}^{[3]}$ defined by Eq. 45.

GTRE1(j) = the first element at the node j of the vector $\mathbf{G}^{[3]}$ defined by Eq. 46.

/ETASEA/ contains the surface elevations at the seaward boundary.

ETAI = normalized elevation η_i of the incident waves at the seaward boundary.

ETAR = normalized elevation η_r of the reflected waves at the seaward boundary where η_r is approximated by Eq. 82.

ESEA = an array of six statistical quantities related to η_i and η_r. Statistical quantities are computed over the time span from t_{min} to t_{max} as explained in Section 2.6.
ESEA(1) = $\overline{\eta_i}$.
ESEA(2) = $\overline{\eta_r}$.
ESEA(3) = a dummy variable used in the process of calculating $\overline{\eta_i}$.
ESEA(4) = a dummy variable used in the process of calculating $\overline{\eta_r}$.

ESEA(5) = maximum value of η_r.
ESEA(6) = minimum value of η_r.

/IWLINE/ contains integer quantities associated with the movement of the computed upper and lower waterlines. See related explanations in COMMON /DWLINE/ below.

NDELR = number of different values of the physical water depth δ_r' defining the waterlines for calculating the normalized run-up heights Z_r^U and Z_r^L.

IS = upper waterline node s satisfying Eq. 55.

ISOLD retains the value of IS at the previous time level.

IW = lower waterline node w satisfying Eq. 56.

ISMIN, ISMAX = minimum, maximum value of IS over the time span from t_{min} to t_{max}.

IWMIN, IWMAX = minimum, maximum value of IW over the time span from t_{min} to t_{max}.

/DWLINE/ contains double precision quantities associated with the movement of the computed upper and lower waterlines.

DELRP(i) = the i-th specified value of δ_r'.

DELR(i) = the i-th value of $\delta_r = (\delta_r'/H')$. It is noted that the *first* value of δ_r is associated with the node JSTAB explained in COMMON /ARMOR4/.

RU(1,i) = Z_r^U associated with the i-th value of δ_r as explained in Section 2.3.4.

RU(2,i) = Z_r^L associated with the i-th value of δ_r as explained in Section 2.5.3.

/STATV/ contains variables representing the mean, maximum, and minimum values of hydrodynamic quantities over the time span from $t = t_{min}$ to $t = t_{max}$.
The index k represents a quantity as listed below.

k	QUANTITY
1	m
2	u
3	q_b
4	η
5	u_p
6	m_p
>6	dummy variable

VMEAN(k,j) = mean value for the quantity k with k=1-12 at the node j.

VMAX(k,j) = maximum value for the quantity k with k=1-6 at the node j.

VMIN(k,j) = minimum value for the quantity k with k=1-6 at the node j.

/STATR/ contains variables representing the mean, maximum, and minimum values of the elevations of waterlines over the time span from $t = t_{min}$ to $t = t_{max}$.

RMEAN(1,i) = mean value of the i-th value of Z_r^U, that is, RU(1,i).

RMEAN(2,i) = mean value of the i-th value of Z_r^L, that is, RU(2,i).

RMEAN(3,i) and RMEAN(4,i) are dummy variables used in the process of calculating RMEAN(1,i) and RMEAN(2,i), respectively.

RMAX(1,i) = maximum value of the i-th value of Z_r^U.

RMAX(2,i) = maximum value of the i-th value of Z_r^L.

RMIN(1,i) = minimum value of the i-th value of Z_r^U.

RMIN(2,i) = minimum value of the i-th value of Z_r^L.

/STATT/ contains the timing of statistical calculations.

TSTAT1 = t_{min} as explained in Section 2.6.

TSTAT2 = t_{max} as explained in Section 2.6.

TSPAN = $(t_{max} - t_{min})$.

/STATI/ contains the upper and lower waterline nodes at the beginning and end of statistical calculations.

IST1 saves the value of the upper waterline node IS at the normalized time TSTAT1=t_{min}.

IST2 saves the value of the upper waterline node IS at the normalized time TSTAT2=t_{max}.

IWT1 saves the value of the lower waterline node IW at the normalized time TSTAT1=t_{min}.

IWT2 saves the value of the lower waterline node IW at the normalized time TSTAT2=t_{max}.

/ENERG/ contains quantities related to wave energy. The index j in the following indicates the node j. The notations for the energy quantities appearing in the following are described in Section 2.8.

EEXTMN(1,j) = $\overline{E}$.

EEXTMN(2,j) = $\overline{F}$.

EEXTMN(3,j) = $\overline{D_p}$.

EEXTMN(4,j) = $\overline{D}$.

EINTMN(1,j) = $\overline{E_p}$.

EINTMN(2,j) = $\overline{F_p}$.

EINTMN(3,j) = $\overline{D_r}$.

EINTMN(4,j) = $\overline{D_p}$.

EEXT(k,j) is a dummy variable used in the process of computing EEXTMN(k,j) with k=1,2,3.

EINT(k,j) is a dummy variable used in the process of computing EINTMN(k,j) with k=1,2,3,4.

EEXT(4,j) = E at the node j at $t = t_{min}$ involved in Eq. 131.

EEXT(5,j) = E at the node j at $t = t_{max}$ involved in Eq. 131.

EINT(5,j) = E_p at the node j at $t = t_{min}$ involved in Eqs. 132 and 133.

EINT(6,j) = E_p at the node j at $t = t_{max}$ involved in Eqs. 132 and 133.

/MASSB/ contains quantities to check mass balance. The index j in the following indicates the node j.

VMASS(1,j) = h at the node j at $t = t_{min}$ involved in Eq. 115.

VMASS(2,j) = h at the node j at $t = t_{max}$ involved in Eq. 115.

VMASS(3,j) = η at the node j at $t = t_{min}$ involved in Eq. 117.

VMASS(4,j) = η at the node j at $t = t_{max}$ involved in Eq. 117.

/ARMOR1/ contains the input parameters for armor stability.

SG = specific gravity s_g of the armor unit.

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

C2 = area coefficient C_2.

C3 = volume coefficient C_3.

CD = drag coefficient C_D.

CL = lift coefficient C_L.

CM = inertia coefficient C_M.

AMAX = parameter a_{min} involved in Eq. 143.

AMIN = parameter a_{max} involved in Eq. 143.

/ARMOR2/ contains the computed parameters for armor stability.

CSTAB1 = $[2C_3^{2/3}/(C_2C_D)]$ used in Eq. 138.

CSTAB2 = $[C_M/\,[(s_g-1)\sigma]]$ used in Eqs. 138 and 142.

AMAXS = $(\sigma\, a_{max})$ used in Eq. 143.

AMINS = $(\sigma\, a_{min})$ used in Eq. 143.

E2 = parameter E_2 defined by Eq. 139.

SSLOPE = $\sin\theta'$ where θ' is the local angle of the upper slope.

CTAN = value of $(cos\theta'\ \tan\phi)$.

E3PRE = value of $[2C_3^{2/3}\ \cos\theta'\ \tan\phi/(C_2C_D)]$ used in Eq. 140.

/ARMOR3/ contains the stability numbers and their timing, and the spatial variations at the time of the critical armor stability.

SNR(j) = stability function N_R at the node j where N_R is given by Eq. 141.

SNSX(j) = local stability number N_{sx} at the node j where N_{sx} is defined as the minimum value of $N_R(t,x)$ at the node j with respect to $t = t_{min}$ to t_{max}.

TSNSX(j) = normalized time when SNSX(j) occurs.

ATMIN(i,j) saves the following quantities at node j at the time of the critical armor stability:
i=1 for N_R
i=2 for u
i=3 for du/dt
i=4 for q_b
i=5 for η
i=6 for u_p
i=7 for m_p

SNSC = critical stability number N_{sc} discussed in Section 2.9.

TSNSC = normalized time when N_{sc} occurs = time of the critical armor stability.

/ARMOR4/ contains the waterline nodes related to armor stability computation.

JSTAB = node number associated with the *first* value of δ_r. The computation of armor stability is performed for the nodes j=1,2,. . .,JSTAB.

JSNSC = node number where the critical stability number N_{sc} occurs.

JSTABM = maximum value of JSTAB over the time span from $t = t_{min}$ to $t = t_{max}$.

JSATM = value of JSTAB at the time of the critical armor stability.

ISATM = upper waterline node IS at the time of the critical armor stability.

IWATM = lower waterline node IW at the time of the critical armor stability.

/SAVB/ contains information related to the nodes for which the computed time series of the hydrodynamic quantities m and h are stored during the time-marching computation.

NNODB = number of nodes.

NODB(i) = i-th node number.

/DIAGV1/ contains integers used to save values to allow the sequential computation of the diagnostic part of PBREAK.

ISOSAV and ISSAV save the upper waterline nodes ISOLD and IS as explained in COMMON /IWLINE/.

IWSAV saves the lower waterline node IW.

M5SAV saves the computation unit number at the end of the set of preceding five computation units.

/DIAGV2/ contains double precision variables used to save values to allow the sequential computation of the diagnostic part of PBREAK.

DSSAV saves the value of δ_s.

DWSAV saves the value of δ_w.

U1SAV saves the values of vector $\mathbf{U}^{[1]}$.

U3SAV saves the values of vector $\mathbf{U}^{[3]}$.

USAV saves the values of the normalized velocity u.

ESAV saves the values of the normalized surface elevation η.

HPSAV saves the values of the normalized internal flow depth h_p.

UPSAV saves the values of the normalized velocity u_p.

AMPSAV saves the values of the normalized flux m_p.

/SEQI/ contains integers specific to each sequence of the diagnostic part of PBREAK.

MSEQ(k) = computation unit number at the end of the sequence k.

MULTIQ retains the values of MULTIF of the sequence k where MULTIF is explained in COMMON /CPAR1/

/SEQD/ contains double precision values specific to each sequence of the diagnostic part of PBREAK.

EPSIQ(k,1) retains the value of the damping coefficient ϵ_1 of the sequence k.

EPSIQ(k,2) retains the value of the damping coefficient ϵ_2 of the sequence k.

/SEQC/ carries the filenames specified in the BLOCK DATA of PBREAK. These filenames are used in the sequential procedure of the diagnostic part of PBREAK. See the BLOCK DATA starting on page 96 for the filenames represented by the character variables FDOC, FMSG, FPUT, and FVAL. The contents of those files are briefly described in the subsequent Section 3.6.

• 3.6 •

OUTPUT PRODUCED BY PBREAK

This section lists the output files produced by the computer program PBREAK. The contents of important output files are explained in details with references being made to Section 3.5 which has already explained the parameters and variables involved. We distinguish two different kinds of PBREAK's output files as follows:

1. A computation with ICOMP=1 or ICOMP=2 produces the following output files.
   - File SEQUENCE contains parameters and values related to the sequential procedure.
   - File TRANSFER contains the ICPs (instrumental computation parameters) to be transferred to the actual part (ICOMP=3) after the diagnostic part is completed.
   - File CINPUTxx contains the values of the ICPs to be passed from the sequence xx to the subsequent sequence (xx+1).
   - File CVALxx contains the values needed to specify the initial conditions for the sequence (xx+1) after the sequence xx is completed.
     This file is not created if the sequence xx reaches t_{max}, which means the diagnostic part has been completed. When this happens, the following message will be displayed on screen.

     ```
     TMAX reached.  Diagnostic part completed.
     May now proceed with actual part (ICOMP=3).
     ```

   - File CDOCxx contains a concise documentation summarizing the computation of the sequence xx.
   - File CMSGxx contains messages issued during the computation of the sequence xx.

   The first four files listed above are self-managed by PBREAK and the user does not need to examine them. File CDOCxx lists, among other things, the ICPs established by the sequence xx. File CMSGxx documents numerical problems encountered by the sequence xx.

2. A computation with ICOMP=3 or a successful computation with ICOMP=4 produces the following output files whose names start with the letter "O" for easy identification.

   - File ODOC contains a summary of the input data and computed results. The contents of this file can be seen in the examples featured in Appendices B through D.
   - File OMSG contains warning and error messages explained in Sections 3.8 and 3.10. A computation with ICOMP=3 should generate no error message.
   - File OENERG contains the spatial variations of wave energy quantities.
   - File OMASS contains the spatial variations of quantities related to mass balance.
   - File OSPACE contains the spatial variations of instantaneous hydrodynamic quantities.
   - File OSTAB contains the spatial variations of quantities related to armor stability.
   - File OSTAT contains the spatial variations of statistical quantities.
   - File ORUNUPL contains the time series of quantities related to the waterline oscillation on the lower slope.
   - File ORUNUPU contains the time series of quantities related to the waterline oscillation on the upper slope.
   - File OSEAWAV contains the time series of hydrodynamic quantities at the seaward boundary.
   - File O1SAVBxx contains the time series of the normalized flux m where xx=01 for the file containing the first 100 waves, xx=02 for the file containing the second 100 waves, and so on.
   - File O2SAVBxx contains the time series of the normalized water depth h where xx=01 for the file containing the first 100 waves, xx=02 for the file containing the second 100 waves, and so on.

Except for the files ODOC and OMSG, the contents of each file with the letter "O" will be detailed further in Sections 3.6.1 through 3.6.9. There are three common FORMATs used to write data points in the output files "O":

```
7000 FORMAT (4F18.9)
8000 FORMAT (5D15.6)
9000 FORMAT (8I8)
```

Besides, a number of comment lines are written in the output files containing spatial variations so as to make them more readable to the user. For brevity, each comment line will be represented simply by the character variable CHAR under 5000 FORMAT (A) in the following.

3.6.1 FILE OENERG

Writer: Subroutine 53 DOC3
Order:

```
      WRITE (53,5000) CHAR
      WRITE (53,9000) ISMIN,IWMIN,ISMAX,IWMAX
      WRITE (53,5000) CHAR
      WRITE (53,9000) IST1,IST2,IWT1,IWT2
      WRITE (53,5000) CHAR
      WRITE (53,9000) ISMAX,IWMAX,JE
      WRITE (53,5000) CHAR
      WRITE (53,7000) XUL(ISMAX),XUL(IWMAX),XUL(JE)
      WRITE (53,5000) CHAR
      WRITE (53,7000) TSTAT1,TSTAT2
      WRITE (53,5000) CHAR
      WRITE (53,7000) X
      DO 531 K = 1,4
         WRITE (53,5000) CHAR
         WRITE (53,8000) (EEXTMN(K,J),J=1,IWMAX)
  531 CONTINUE
      DO 532 K = 4,5
         WRITE (53,5000) CHAR
         WRITE (53,8000) (EEXT(K,J),J=1,IWMAX)
  532 CONTINUE
      DO 533 K = 1,4
         WRITE (53,5000) CHAR
         WRITE (53,8000) (EINTMN(K,J),J=1,IWMAX)
  533 CONTINUE
      DO 534 K = 5,6
         WRITE (53,5000) CHAR
         WRITE (53,8000) (EINT(K,J),J=1,IWMAX)
  534 CONTINUE
```

Reference for explanation:

VARIABLE	DESCRIPTION IN SECTION 3.5
ISMIN,IWMIN,ISMAX,IWMAX	COMMON /IWLINE/
IST1,IST2,IWT1,IWT2	COMMON /STATI/
JE	COMMON /CPAR1/
XUL	COMMON /BOT5/
TSTAT1,TSTAT2	COMMON /STATT/
X	COMMON /CPAR2/
EEXTMN,EEXT,EINTMN,EINT	COMMON /ENERG/

3.6.2 File OMASS

Writer: Subroutine 53 DOC3
Order:

```
      WRITE (54,5000) CHAR
      WRITE (54,9000) ISMIN,IWMIN,ISMAX,IWMAX
      WRITE (54,5000) CHAR
      WRITE (54,9000) IST1,IST2,IWT1,IWT2
      WRITE (54,5000) CHAR
      WRITE (54,9000) ISMAX,IWMAX,JE
      WRITE (54,5000) CHAR
      WRITE (54,7000) XUL(ISMAX),XUL(IWMAX),XUL(JE)
      WRITE (54,5000) CHAR
      WRITE (54,7000) TSTAT1,TSTAT2
      WRITE (54,5000) CHAR
      WRITE (54,7000) X,PORO,PARPQ
      WRITE (54,5000) CHAR
      WRITE (54,8000) (VMEAN(1,J),J=1,IWMAX)
      WRITE (54,5000) CHAR
      WRITE (54,8000) (VMEAN(6,J),J=1,IWMAX)
      WRITE (54,5000) CHAR
      WRITE (54,8000) (VMEAN(3,J),J=1,IWMAX)
      DO 541 K = 1,4
         WRITE (54,5000) CHAR
         WRITE (54,8000) (VMASS(K,J),J=1,IWMAX)
  541 CONTINUE
```

Reference for explanation:

Variable	Description in Section 3.5
ISMIN,IWMIN,ISMAX,IWMAX	COMMON /IWLINE/
IST1,IST2,IWT1,IWT2	COMMON /STATI/
JE	COMMON /CPAR1/
XUL	COMMON /BOT5/
TSTAT1,TSTAT2	COMMON /STATT/
X	COMMON /CPAR2/
PORO,PARPQ	COMMON /ULAYER/
VMEAN	COMMON /STATV/
VMASS	COMMON /MASSB/

3.6.3 FILE OSPACE

Writer: Subroutine 51 DOC1
Order:

```
WRITE (52,5000) CHAR
WRITE (52,9000) INITS,INITW,JE
WRITE (52,5000) CHAR
WRITE (52,7000) XUL(INITS),XUL(INITW),XUL(JE)
```

Writer: Subroutine 52 DOC2
Order: The following is written [NTIMESx(MSAVA2-MSAVA1+1)] times as explained in COMMON /CPAR1/

```
WRITE (52,8000) TIME
WRITE (52,9000) IS,IW
WRITE (52,8000) (ELEV(J),J=1,IW)
WRITE (52,8000) (U(J), J=1,IS)
WRITE (52,8000) (UB(J), J=1,IS)
WRITE (52,8000) (QB(J), J=1,IS)
WRITE (52,8000) (UP(J), J=1,IW)
WRITE (52,8000) (AMP(J), J=1,IW)
```

Reference for explanation:

VARIABLE	DESCRIPTION IN SECTION 3.5
INITS,INITW,JE, MSAVA1,MSAVA2,NTIMES	COMMON /CPAR1/
XUL	COMMON /BOT5/
TIME	COMMON /TEMPO/
IS,IW	COMMON /IWLINE/
ELEV,QB	COMMON /HYDROG/
U,UB	COMMON /HYDROE/
UP,AMP	COMMON /HYDROI/

3.6.4 FILE OSTAB

Writer: Subroutine 53 DOC3
Order:

```
      WRITE (55,5000) CHAR
      WRITE (55,9000) ISMAX,IWMAX,JE
      WRITE (55,5000) CHAR
      WRITE (55,9000) JSTABM,JSATM,ISATM,IWATM
      WRITE (55,5000) CHAR
      WRITE (55,7000) XUL(ISMAX),XUL(IWMAX),XUL(JE)
      WRITE (55,5000) CHAR
      WRITE (55,7000) XUL(JSTABM),XUL(JSATM),XUL(ISATM),XUL(IWATM)
      WRITE (55,5000) CHAR
      WRITE (55,7000) ZU(JSTABM),ZU(JSATM),ZU(ISATM),ZU(IWATM)
      WRITE (55,5000) CHAR
      WRITE (55,7000) ZU(1)
      WRITE (55,5000) CHAR
      WRITE (55,7000) TSTAT1,TSTAT2
      WRITE (55,5000) CHAR
      WRITE (55,8000) (SNSX(J),J=1,IWMAX)
      WRITE (55,5000) CHAR
      WRITE (55,8000) (TSNSX(J),J=1,IWMAX)
      DO 551 K = 1,7
         WRITE (55,5000) CHAR
         WRITE (55,8000) (ATMIN(K,J),J=1,IWMAX)
  551 CONTINUE
```

Reference for explanation:

VARIABLE	DESCRIPTION IN SECTION 3.5
ISMAX,IWMAX	COMMON /IWLINE/
JE	COMMON /CPAR1/
JSTABM,JSATM,ISATM,IWATM	COMMON /ARMOR4/
XUL,ZU	COMMON /BOT5/
TSTAT1,TSTAT2	COMMON /STATT/
SNSX,TSNSX,ATMIN	COMMON /ARMOR3/

3.6.5 File OSTAT

Writer: Subroutine 53 DOC3
Order:

```
      WRITE (51,5000) CHAR
      WRITE (51,9000) INITS,INITW,JE
      WRITE (51,5000) CHAR
      WRITE (51,9000) ISMIN,IWMIN,ISMAX,IWMAX
      WRITE (51,5000) CHAR
      WRITE (51,9000) IST1,IST2,IWT1,IWT2
      WRITE (51,5000) CHAR
      WRITE (51,7000) TSTAT1,TSTAT2
      WRITE (51,5000) CHAR
      WRITE (51,8000) (XUL(J),J=1,JE)
      WRITE (51,5000) CHAR
      WRITE (51,8000) (ZU(J),J=1,JE)
      WRITE (51,5000) CHAR
      WRITE (51,8000) (ZL(J),J=1,JE)
      DO 511 K = 1,6
         WRITE (51,5000) CHAR
         WRITE (51,8000) (VMAX(K,J), J=1,IWMAX)
         WRITE (51,5000) CHAR
         WRITE (51,8000) (VMEAN(K,J),J=1,IWMAX)
         WRITE (51,5000) CHAR
         WRITE (51,8000) (VMIN(K,J), J=1,IWMAX)
  511 CONTINUE
```

Reference for explanation:

Variable	Description in Section 3.5
INITS,INITW,JE	COMMON /CPAR1/
ISMIN,IWMIN,ISMAX,IWMAX	COMMON /IWLINE/
IST1,IST2,IWT1,IWT2	COMMON /STATI/
TSTAT1,TSTAT2	COMMON /STATT/
XUL,ZU,ZL	COMMON /BOT5/
VMAX,VMEAN,VMIN	COMMON /STATV/

3.6.6 FILE ORUNUPL

Writer: Subroutine 52 DOC2
Order: The following is written at the rate of NRATE data points per wave period.

```
DBLEW = DBLE(IW)
WRITE (63,8000) DBLEW,(RU(2,L),L=1,NDELR)
```

Reference for explanation:

VARIABLE	DESCRIPTION IN SECTION 3.5
IW,NDELR	COMMON /IWLINE/
RU	COMMON /DWLINE/

3.6.7 FILE ORUNUPU

Writer: Subroutine 52 DOC2
Order: The following is written at the rate of NRATE data points per wave period.

```
DBLES = DBLE(IS)
WRITE (62,8000) DBLES,(RU(1,L),L=1,NDELR)
```

Reference for explanation:

VARIABLE	DESCRIPTION IN SECTION 3.5
IS,NDELR	COMMON /IWLINE/
RU	COMMON /DWLINE/

3.6.8 File OSEAWAV

Writer: Subroutine 52 DOC2
Order: The following is written at the rate of NRATE data points per wave period.

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

Reference for explanation:

Variable	Description in Section 3.5
TIME	COMMON /TEMPO/
ETAI,ETAR	COMMON /ETASEA/
UONE	COMMON /HYDROE/

3.6.9 File 01SAVBxx and 02SAVBxx

Writer: Subroutine 52 DOC2
Order: The following is written at the rate of NRATE data points per wave period.

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

Reference for explanation:

Variable	Description in Section 3.5
NNODB,NODB	COMMON /SAVB/
UONE	COMMON /HYDROE/

• 3.7 •

INSTRUMENTAL COMPUTATION PARAMETERS

The six instrumental computation parameters (ICPs) mentioned in Section 3.1 are listed below.

1. INITS = number of the spatial nodes at which the *upper* slope is located below SWL.
2. INVT = number of time steps in one reference wave period.
3. EPSI1 = damping coefficient ϵ_1 involved in Eqs. 65 and 66.
4. EPSI2 = damping coefficient ϵ_2 involved in Eqs. 65 and 66.
5. DELTAS = normalized water depth δ_s defining the computational *upper* waterline (see Eq. 55).
6. DELTAW = normalized water depth δ_w defining the computational *lower* waterline (see Eq. 56).

It should be stated that the computed results of practical importance should not be affected by the particular values of these parameters used in the computation. As a result, the values of these parameters should be selected to produce a successful computation within an appropriate range of each of these parameters.

Section 3.1 has mentioned that numerical problems can be minimized by adjusting these ICPs during the hydrodynamic computation. Actually, not all the ICPs are allowed to vary during a computation. As explained in Section 3.1, a computation may consist of a number of sequences, and a sequence consists of a number of computation units. The values of the ICPs are selected as follows:

- INITS is fixed during a computation. INITS should be chosen so large that the spacing between two adjacent nodes is small enough to resolve the spatial variation of the wave motion. The range of INITS=100 to 400 has been used previously. INITS=200 is used in the examples given in the Appendices B through D.
- INVT is not specified by the user since it is harder to specify the value of this parameter. The value of INVT is determined by the computer program PBREAK as follows:
  - INVT must satisfy the numerical stability criterion Eq. 77.
  - INVT must be divisible by NRATE as explained in COMMON /CPAR1/.
  - MINVT is the smallest value satisfying the above two criteria. MINVT constitutes the minimum allowable value of INVT.
  - MULTIF is an integer multiplication factor such that (MULTIF*MINVT) constitutes the minimum value of INVT in a *sequence* of computation.

It is the parameter MULTIF that the user specifies. MULTIF=1 has been used to minimize the computation time. It is suggested to increase MULTIF from unity if the computed results show unrealistic spikes and kinks. The value of MULTIF is fixed during a *sequence* of computation, but the value of INVT will be increased incrementally for the computation units with numerical problems by the diagnostic part of PBREAK. INVT is constant during a computation unit.

- EPSI1 and EPSI2 are fixed during a *sequence* of computation. The range of EPSI1= EPSI2=1-2 has been used previously.
- DELTAS and DELTAW may vary within a computation unit as deemed necessary by the diagnostic part of PBREAK. The range of DELTAS=DELTAW=0.001-0.003 has been used as input to PBREAK. The increase of these parameters tends to reduce numerical difficulties. As mentioned in Section 3.3, the self-adjusting mechanism of PBREAK has the capability of adjusting the computation parameters INVT, DELTAS, and DELTAW. This self-adjusting mechanism is explained in Section 3.9.

In the primary input data, the user specifies the above six ICPs, with MULTIF specified *in lieu* of INVT. The specification of these ICPs to execute PBREAK can be outlined as follows:

- For a computation with ICOMP=4, the user specifies one set of the ICPs in a single primary input data file.
- For a computation with ICOMP=3, the ICPs specified in the primary input data file are not used since all the ICPs needed for a successful computation have been adjusted by its corresponding diagnostic part.
- For a computation with ICOMP=1:
  1. Specify one set of the ICPs in the *first* primary input data file of the diagnostic part of PBREAK.
  2. If this computation with ICOMP=1 successfully reaches the terminal time t_{max}, then the diagnostic part is completed. Go to Step 6 as also explained in Figure 4.
  3. If this computation with ICOMP=1 breaks down before t_{max} is reached, the user may either go back to $t = 0$ (Step 1) or proceed with the subsequent computation with ICOMP=2 (Step 4).
  4. Specify another set of the ICPs in the *next* primary input data file for the subsequent computation with ICOMP=2.
  5. If the subsequent computation with ICOMP=2 reaches t_{max}, go to Step 6. Otherwise, repeat Step 4.
  6. Proceed to the actual part (ICOMP=3) of PBREAK. The user may use any of the primary input data files that have been used in the diagnostic part since the ICPs specified in the primary input data file are not used in the computation with ICOMP=3.

• 3.8 •

REMEDIABLE NUMERICAL PROBLEMS

Listed in the following are five numerical problems that may occur in the hydrodynamic subsystem of PBREAK and the messages issued when they occur. These five numerical problems are remediable by the self-adjusting mechanism of PBREAK or by the user's intervention. Other numerical problems which are not remediable are listed in Section 3.10 along with other fatal computational problems. The messages presented in the following are self-explanatory. If necessary, the user should examine the subroutine associated with each problem where the computer program PBREAK is presented in Appendix A.

3.8.1 `IPROB=1`

Problem: Negative water depth in Region 1.
Messages issued:

```
::  IPROB=1 from Subr.  24 ROOTH ::
    Computation unit            M = ...
    Upper waterline node       IS = ...
    Computation parameter INVT = ...
    Time of occurrence       TIME = ...
    Indicators              OMEGA = ...
                  ITRY, ICOUNT = ...
     Node          h
      .            .
      .            .
      .            .
```

where OMEGA=Ω given by Eq. 76; ITRY = number of tries within the computation unit M to overcome numerical problems; and ICOUNT = number of nodes where this problem occurs.

3.8.2 IPROB=2

Problem: Numerical stability criterion Eq. 75 violated.
Messages issued:

```
:: IPROB=2 from Subr.  39 NUMSTA ::
   Computation unit           M = ...
   Upper waterline node      IS = ...
   Computation parameter INVT = ...
   Time of occurrence      TIME = ...
   Delta x / Delta t         XT = ...
   Indicators             OMEGA = ...
                  ITRY, ICOUNT = ...
    Node          u
     .            .
     .            .
     .            .
```

3.8.3 IPROB=3

Problem: $h_{s-1} < \frac{1}{2}\delta_s$, that is, the water depth h at the node (s-1) is too small.
Messages issued:

```
:: IPROB=3 from Subr.  21 MARCH1 ::
   Computation unit              M = ...
   Upper waterline node         IS = ...
   Computation parameters     INVT = ...
                            DELTAS = ...
   Water depth at (IS-1)           = ...
   Time of occurrence         TIME = ...
   Indicators                OMEGA = ...
                              ITRY = ...
```

3.8.4 IPROB=4

Problem: Negative water depth in Region 3.
Messages issued:

```
:: IPROB=4 from Subr.  23 MARCH3 ::
   Computation unit           M = ...
   Lower waterline node      IW = ...
   Computation parameter INVT = ...
   Time of occurrence      TIME = ...
   Indicators             OMEGA = ...
                  ITRY, ICOUNT = ...
    Node          hp
     .             .
     .             .
     .             .
```

3.8.5 IPROB=5

Problem: $(h_p)_{w-1} < \frac{1}{2}\delta_w$, that is, the water depth h_p at the node (w-1) is too small.
Messages issued:

```
:: IPROB=5 from Subr.  23 MARCH3 ::
   Computation unit             M = ...
   Lower waterline node        IW = ...
   Computation parameters    INVT = ...
                           DELTAW = ...
   Water depth at (IW-1)          = ...
   Time of occurrence        TIME = ...
   Indicators               OMEGA = ...
                             ITRY = ...
```

• 3.9 •

SELF-ADJUSTING MECHANISM

Based on the ICPs specified by the user in the primary input data file, the computer program PBREAK (1) determines the initial values for the computation parameters INVT, DELTAS, and DELTAW for a computation unit and (2) adjust them when the unit encounters a remediable numerical problem. These tasks are performed for each computation unit according to the following guidelines:

1. Initially

   - Set INVT = MULTIF*MINVT.
   - Use the value of DELTAS at the end of the previous computation unit.
   - Use the value of DELTAW at the end of the previous computation unit.

2. If $\frac{1}{2}$(DELTAS)$\leq h_{s-1} \leq$ (DELTAS), set a new value of DELTAS which satisfies the following:

   - DELTAS $\simeq 0.95h_{s-1}$.
   - DELTAS is divisible by 0.000001.

   Computation then proceeds to the next time level where the time level within the computation unit M is indicated by the integer N in the computer program PBREAK.
   Otherwise, that is, if $h_{s-1} < \frac{1}{2}$(DELTAS), set IPROB=3.

3. If $\frac{1}{2}$(DELTAW)$\leq (h_p)_{w-1} \leq$ (DELTAW), set a new value of DELTAW which satisfies the following:

   - DELTAW $\simeq 0.95(h_p)_{w-1}$.
   - DELTAW is divisible by 0.000001.

   Computation then proceeds to the next time level.
   Otherwise, that is, if $(h_p)_{w-1} < \frac{1}{2}$(DELTAW), set IPROB=5.

4. If any of the remediable numerical problems listed in Section 3.8 (IPROB=1,2,3,4,5) is encountered, set a new value of INVT which satisfies the following:

   - New value of INVT $\simeq$ 1.5 times the old value.
   - INVT is divisible by NRATE.

   Computation is then re-started from the beginning of the unit using the new value of INVT and the values of DELTAS and DELTAW at the end of the previous computation unit. The number of trials for each computation unit is counted by the integer ITRY.

5. It can happen that INVT has been increased several times but remediable numerical problems still occur. There is an upper limit for the value of INVT. The limit is set in terms of the numerical stability indicator OMEGA=Ω given by Eq. 77 which implies that Ω increases with INVT=$(1/\Delta t)$. Whenever $\Omega > 10$, computation is terminated. When this happens

and this computation is made with ICOMP=1 or ICOMP=2, then computation may be continued to the next sequence. The values needed as the initial conditions for the next sequence of computation are saved. These values are at the designated stage in the present sequence as explained below.

6. At any one time, the values at the ends of the two nearest sets of preceding five computation units are saved. The values to be passed to the next sequence of computation are the values at the farthest end. This is because numerical problems may have initiated before a few computation units.
   Example:

   > If a computation breaks down during computation unit 32, the values at the end of computation unit 25 are passed to the next sequence. The next sequence thus begins at computation unit 26.

Reference for more explanation regarding the variables involved in the above discussion:

VARIABLE	DESCRIPTION
INVT,MULTIF,MINVT,NRATE	See COMMON /CPAR1/ in Section 3.5
DELTAS,DELTAW	See COMMON /CPAR2/ in Section 3.5
h_{s-1}=UONE(2,IS-1)	See COMMON /HYDROE/ in Section 3.5
$(h_p)_{w-1}$=UTRE(2,IW-1)	See COMMON /VECTRE/ in Section 3.5
IS,IW	See COMMON /IWLINE/ in Section 3.5
IPROB	See Section 3.8
ICOMP	See Section 3.1

• 3.10 •

Error Messages

Computation of PBREAK is terminated immediately following any of the error messages listed below. Preceding each message in the list is the circumstance under which the message is issued.

1. Upper waterline movement reaches the landward end node (IPROB=6).

```
:: IPROB=6 from Subr.  21 MARCH1 ::
   Upper waterline movement reaches the landward end node
   Computation unit             M = ...
   Upper waterline node        IS = ...
   Landward end node           JE = ...
   Computation parameters    INVT = ...
                           DELTAS = ...
   Time of occurrence        TIME = ...
   Indicators               OMEGA = ...
                             ITRY = ...
>> Possibility Number 1:
   Slope is not long enough to accommodate upper waterline
       movement.  Specify longer slope and start computing
       from t=0.
>> Possibility Number 2:
   If slope has been specified long enough, this waterline
       movement may not be realistic.  Keep the current
       specification of slope geometry, and repeat this
       sequence using different computation parameters.
>> Computation aborted.
```

It is noted that the computer program PBREAK in its present form does not allow wave overtopping and transmission unlike the computer program IBREAK (Kobayashi and Wurjanto 1989c) and RBREAK (Wurjanto and Kobayashi 1991) developed for impermeable structures.

2. Lower waterline movement reaches the landward end node (IPROB=7).

```
:: IPROB=7 from Subr.  23 MARCH3 ::
   Lower waterline movement reaches the landward end node
   Computation unit            M = ...
   Lower waterline node       IW = ...
   Landward end node          JE = ...
   Computation parameters INVT = ...
                          DELTAW = ...
   Time of occurrence       TIME = ...
   Indicators              OMEGA = ...
                            ITRY = ...
   >> Possibility Number 1:
   Slope is not long enough to accommodate lower waterline
       movement.  Specify longer slope and start computing
       from t=0.
   >> Possibility Number 2:
   If slope has been specified long enough, this waterline
       movement may not be realistic.  Keep the current
       specification of slope geometry, and repeat this
       sequence using different computation parameters.
   >> Computation aborted.
```

3. The specified value of ICOMP is outside the range $1 \leq$ICOMP≤ 4.

```
Choice of ICOMP out of range.
Programmed Stop.
```

where the programmed stop is executed in Subroutine 56 STOPP.

4. The self-adjusting mechanism of PBREAK needs a larger upper limit of the FORTRAN PARAMETER MXDIV where MXDIV is the maximum number of divisions having different values of DELTAS or DELTAW in a computation unit.

```
Need larger MXDIV in the main program.
Programmed Stop.
```

5. There is a mismatch between the diagnostic (ICOMP=1,2) and actual (ICOMP=3) parts of a PBREAK execution. This error should *not* happen if PBREAK is executed properly following the procedure described in Sections 3.1 and 3.7.

```
Mismatch between M and MREAD in ICOMP=3.
File TRANSFER might have been altered.
Programmed Stop.
```

6. An execution of the actual part (ICOMP=3) of PBREAK is attempted before the preceding diagnostic part has been completed.

```
DIAGNOSTIC part incomplete.  ACTUAL part can not run.
Programmed Stop.
```

7. There is a mismatch between the diagnostic (ICOMP=1,2) and actual (ICOMP=3) parts of a PBREAK execution since IPROB=0 in Subroutine 02 MAIN2 for the computation with ICOMP=3.

```
IPROB not zero from MAIN2.
Programmed Stop.
```

8. The computation parameter MSAVA2<MSAVA1 as explained in COMMON /CPAR1/.

```
MSAVA2 must be .GE. MSAVA1.
Programmed Stop.
```

9. The specified temperature of water is beyond the range 0-100 °C in Subroutine 06 CVISCO.

```
Temperature out of range.
Programmed Stop.
```

10. If the specified upper slope is longer than the specified lower slope, IPROB=7 may occur during computation. As a result, computation is aborted before time-marching computation.

```
Upper slope can not extend beyond lower slope.
Programmed Stop.
```

11. The upper slope is submerged where PBREAK does not allow wave transmission.

```
Upper slope always below SWL. PBREAK can not run.
Programmed Stop.
```

12. The FORTRAN **PARAMETER MXNOD** is too small for the specified structure where **MXNOD** is the maximum number of spatial nodes and **MXNOD**=1000 in the present form of PBREAK.

```
Slope too long for present setting of MXNOD.
Programmed Stop.
```

13. The lower slope is submerged where PBREAK does not allow wave transmission.

```
Lower slope always below SWL, unacceptable in PBREAK.
Programmed Stop.
```

14. The upper waterline moves more than Δx over one time step Δt, which is a violation of the numerical stability criterion Eq. 75.

```
Upper waterline moved more than one node.
Programmed Stop.
```

15. The lower waterline moves more than Δx over one time step Δt, which is a violation of the adopted numerical stability criterion.

```
Lower waterline moved more than one node.
Programmed Stop.
```

PART IV

CONCLUSIONS AND RECOMMENDATIONS

The computer program PBREAK simulates the interaction of normally-incident irregular waves with a permeable coastal structure in the manner similar to hydraulic model tests in a wave flume. PBREAK has been evaluated using three irregular wave test runs for irregular wave reflection and run-up on a 1:3 rough slope with a thick permeable underlayer (Kobayashi, Cox, and Wurjanto 1991). Users of PBREAK are strongly recommended to read the report of Wurjanto and Kobayashi (1992a) which presents more physical interpretations of the computed results using additional spectral and time series analyses. Users will need the subroutines reported by Cox, Kobayashi, and Wurjanto (1991) to perform these additional analyses if the subroutines are not available to users. The report of Wurjanto and Kobayashi (1992a) also presents the comparisons of the computed results for permeable and impermeable slopes using PBREAK and RBREAK to examine the permeability effects on irregular wave run-up, reflection and armor stability.

PBREAK will need to be modified to predict irregular wave overtopping and transmission. The hydraulic stability and movement of armor units under the action of irregular waves have been compared with available data using the related numerical model limited to a thin permeable underlayer as described by Kobayashi, Wurjanto, and Cox (1990a,b). The computed hydraulic stability and movement of armor units under the action of irregular waves are normally difficult to interpret since the corresponding laboratory data are typically analyzed in terms of the profile change of the slope protected with armor units. Users of PBREAK will need to evaluate and calibrate PBREAK if it is to be applied to the problems for which PBREAK has not been evaluated.

The numerical model PBREAK is based on the approximate governing equations that are time-dependent but one-dimensional normal to a long coastal structure with the assumption of uniformity along the structure. This model is probably the simplest time- dependent model that can be applied for breaking and broken waves on a permeable slope. The limited verification of PBREAK in Appendices B, C, and D indicates that PBREAK may predict some quantities accurately but other quantities only marginally. As a result, correct interpretations of the output of PBREAK are essential to avoid misuse of PBREAK. It will eventually be necessary to develop unsteady two-dimensional and three-dimensional models to improve the accuracy of the predictions, although computational efforts will increase considerably.

REFERENCES

Ahrens, J.P., 1989. Stability of reef breakwaters. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **115** (2), 221-234.

Ahrens, J.P., 1990. Dynamic revetments. *Proc. 22nd Coastal Engineering Conference*, ASCE, **2**, 1837-1850.

Ahrens, J.P., and Ward, D.L., 1991. Performance of bermed revetments. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **117** (5), 534-540.

Anderson, D.A., Tannehill, J.C., and Pletcher, R.H., 1984. *Computational Fluid Mechanics and Heat Transfer*. Hemisphere Publishing Company, New York.

Baird, W.F., and Hall, K.R., 1984. The design of breakwaters using quarried stones. *Proc. 19th Coastal Engineering Conference*, ASCE, **1**, 1024-1031.

Cox, D.T., Kobayashi, N., and Wurjanto, A., 1991. Computer programs for spectral and time series analyses for random waves. *Research Report No. CACR-91-06*, Center for Applied Coastal Research, University of Delaware.

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

Fischer, H.B., List, E.J, Koh, R.C.Y., Imberger, J., and Brooks, N.H., 1979. *Mixing in Inland and Coastal Waters*. Academic Press, New York.

Guza, R.T., and Thornton, E.B., 1982. Swash oscillations on a natural beach. *J. Geophysical Research*, **87** (C1), 483-491.

Holman, R.A., and Sallenger, A.H., Jr., 1985. Set-up and swash on a natural beach. *J. Geophysical Research*, **90** (C1), 945-953.

Kobayashi, N., and Otta, A.K., 1987. Hydraulic stability analysis of armor units. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **113** (2), 171-186.

Kobayashi, N., Otta, A.K., and Roy, I., 1987. Wave reflection and run-up on rough slopes. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **113** (3), 282-298.

Kobayashi, N., and Greenwald, J.H., 1988. Waterline oscillation and riprap movement. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **114** (3), 281-296.

Kobayashi, N., and Wurjanto, A., 1989a. Wave overtopping on coastal structures. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **115** (2), 235-251.

Kobayashi, N., and Wurjanto, A., 1989b. Wave transmission over submerged breakwaters. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **115** (5), 662-680.

Kobayashi, N., and Wurjanto, A., 1989c. Numerical model for design of impermeable coastal structures. *Research Report No. CE-89-75*, Center for Applied Coastal Research, University of Delaware.

Kobayashi, N., and Wurjanto, A., 1989d. Armor stability on rough permeable slopes of marine structures. *Proc. 23rd Congress of International Association for Hydraulic Research*, Ottawa, Canada, **C**, 407-414.

Kobayashi, N., Cox, D.T., and Wurjanto, A., 1990. Irregular wave reflection and run-up on rough impermeable slopes. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **116** (6), 708-726.

Kobayashi, N., and Wurjanto, A., 1990. Numerical model for waves on rough permeable slopes. *J. Coastal Research*, Special Issue No. **7** on Rational Design of Mound Structures, 149-166.

Kobayashi, N., Wurjanto, A., and Cox, D.T., 1990a. Irregular waves on rough permeable slopes. *J. Coastal Research*, Special Issue No. **7** on Rational Design of Mound Structures, 167-184.

Kobayashi, N., Wurjanto, A., and Cox, D.T., 1990b. Rock slopes under irregular wave attack. *Proc. 22nd Coastal Engineering Conference*, ASCE, 1306-1319.

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

Kobayashi, N., and Wurjanto, A., 1992a. Irregular wave set-up and run-up on beaches. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **118** (4) (to appear).

Kobayashi, N., and Wurjanto, A., 1992b. Irregular wave interaction with permeable slopes. *Proceedings of 23rd Coastal Engineering Conference*, ASCE (accepted).

Madsen, O.S., and White, S.M., 1976. Reflection and transmission characteristics of porous rubble mound breakwaters. *Miscellaneous Report 76-5.* U.S. Army Corps of Engineers Coastal Engineering Research Center.

Packwood, A.R., 1980. Surf and run-up on beaches. *Ph.D. Dissertation.* School of Mathematics, University of Bristol, United Kingdom.

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

Roache, P.J., 1982. *Computational Fluid Dynamics.* Revised Printing. Hermosa Publishers, Albuquerque, New Mexico.

Van der Meer, J.W., 1988. Deterministic and probabilistic design of breakwater armor layers. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE, **114** (1), 66-80.

Wise, R.A., Kobayashi, N., and Wurjanto, A., 1991. Cross-shore sediment transport under irregular waves in surf zones. *Proceedings of Coastal Sediments '91*, ASCE, pp. 658-673.

Wurjanto, A., and Kobayashi, N., 1991. Numerical model for random waves on impermeable coastal structures and beaches. *Research Report No. CACR-91-05*, Center for Applied Coastal Research, University of Delaware.

Wurjanto, A., and Kobayashi, N., 1992a. Irregular waves interaction with permeable slopes of coastal structures. *Research Report No. CACR-92-03*, Center for Applied Coastal Research, University of Delaware.

Wurjanto, A., and Kobayashi, N., 1992b. Irregular wave reflection and run-up on permeable slopes. *J. Waterway, Port, Coastal, and Ocean Engineering*, ASCE (submitted).

Appendix A

Listing of Computer Program PBREAK

```
C
C     #######  #######  #######  ########  ######   ##     ##
C     ##    ## ##    ## ##    ## ##       ##    ##  ##    ##
C     ##    ## ##    ## ##    ## ##       ##    ##  ##   ##
C     #######  #######  #######  #######  ########  ## ##
C     ##       ##    ## ##  ##   ##       ##    ##  ######
C     ##       ##    ## ##   ##  ##       ##    ##  ###  ##
C     ##       #######  ##    ## ######## ##    ##  ##     ##
C
C                     Numerical Simulation of
C              Random Waves on Permeable Breakwaters
C
C Developed by Andojo Wurjanto under Supervision of Nobuhisa Kobayashi
C                Center for Applied Coastal Research
C                  Department of Civil Engineering
C          University of Delaware, Newark, Delaware 19716
C                           March  1992
C
C >>>>>>>>>>>>>>>>>>>>>>>>>>  GENERAL NOTES  <<<<<<<<<<<<<<<<<<<<<<<<<<
C
C  For a successful PBREAK computation, needed is a "working" combi-
C    nation of the following computation parameters:
C    1. INITS  = number of spatial nodes at which the upper slope is
C                located below SWL
C    2. INVT   = number of time steps in one reference wave period
C    3. DELTAS = normalized water depth defining the computational
C                upper waterline
C    4. DELTAW = normalized water depth defining the computational
C                lower waterline
C    5. EPSI1 and EPSI2 = damping coefficients incorporated in the
C                dissipative term in the numerical scheme
C  It is not likely that the user would find a single combination
C    that works for the whole duration of a computation. Therefore,
C    all of the above parameters are allowed to vary during a com-
C    putation, with the exception of INITS, which must be constant.
C  PBREAK has a limited capability of self-adjusting INVT, DELTAS,
C    and DELTAW. This mechanism, however, may not always be counted
C    on to lead to a successful computation.
C  Failure may occur for cases involving irregular waves of long
C    duration. For this reason, PBREAK has been written in such a
C    way that when a failure occurs:
C    . most of what has been successfully computed is saved
C    . necessary information at the end of the successful part is
C      stored to be prescibed as the initial condition for the next
C      sequence of computation
C    . the user resumes the next sequence of computation using a
C      different set of the computation parameters
C  Thus, a computation may consist of several sequences. Efforts
C    have been made to provide a simple transition from one sequence
C    to the next.
C  It is imperative that all the sequences of a computation be
C    performed on the same machine.
C
C  PBREAK consists of two parts:
C    1. The "Diagnostic" Part.
```

```
C          Contained mainly in the main program, this part manages
C          . the self-adjusting mechanism and
C          . the mechanism for the sequential transition.
C       2. The "Actual" Part.
C          Executed by Subroutine 02 MAIN2, this part performs the
C          "actual" computation after a working set of the instrumental
C          computation parameters has been established by the
C          diagnostic part.
C
C     The user is given the following options in executing PBREAK:
C       . ICOMP=1 executes only the diagnostic part from time t=0.
C       . ICOMP=2 executes a continuation of a previously incomplete
C         computation of the diagnostic part.
C       . ICOMP=3 executes only the actual part from t=0.
C         This option must be preceded by a complete execution of the
C         diagnostic part.
C       . ICOMP=4 executes both the diagnostic and actual parts from
C         t=0 in one sweep.
C     Sequential computation is inherent in the options ICOMP=1 and
C       ICOMP=2. If ICOMP=1 or ICOMP=2 does not complete its job,
C       the user may either
C       . start with ICOMP=1 again using different computation
C         parameters, or
C       . proceed with ICOMP=2.
C       It is recommended that the user execute ICOMP=1 in the first
C       place for the reason that this option does not waste computer
C       resources in case the computation fails along the way.
C     The options ICOMP=3 and ICOMP=4 are not sequential.
C       . ICOMP=3 should not fail if the associated diagnostic part has
C         been completed on the same machine.
C       . ICOMP=4 may result in failure during a computation of long
C         duration. ICOMP=4 may be used for cases involving regular
C         waves or irregular waves of short duration.
C
C     PBREAK breaks computation duration into "computation units". Each
C       computation unit is one reference wave period long except for
C       the last unit, the length of which may be less than one
C       reference wave period.
C     The number of computation units in a sequence of computation is
C       chosen to be 5*IM where IM is an integer multiplication factor
C       greater than or equal to unity, except for the last sequence,
C       which may consist of any number of computation units.
C
C >>>>>>>>>>>>>>>>>>>>>>>  MAIN PROGRAM PBREAK  <<<<<<<<<<<<<<<<<<<<<<<
C
      PROGRAM PBREAK
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C ... PARAMETERS
C
C       >> Used in the main program and subprograms:
C        . MXNOD = max. number of spatial nodes
C        . MXTIM = max. number of data points in the input time series
C        . MXRUN = max. number of user-specified water depths defining
```

```
C                waterlines for run-up computation
C        . MXSEG = max. number of linear segments specifying upper and
C                lower slopes
C        . MXSAV = max. number of spatial nodes for which temporal
C                variations are to be saved
C       >> Used only in the main program:
C        . MXDIV = max. number of divisions having different values of
C                DELTAS or DELTAW in a computation unit
C
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      PARAMETER (MXDIV=100)
C
C ... TYPES AND DIMENSIONS OF VARIABLES
C
      CHARACTER*10 FINP1,FINP2
      CHARACTER*10 FDOC,FMSG,FPUT,FVAL
      CHARACTER*49 MSGS(0:4)
      DIMENSION NDIV(MXDIV),IDIV(MXDIV)
      DIMENSION SDIV(MXDIV),WDIV(MXDIV)
C
C ... COMMON BLOCKS
C
C     /CONSTA/ contains basic constants
C     /OPTION/ contains identifiers specifying user's options
C     /CPAR1/ & /CPAR2/ contain computation parameters
C     /TEMPO/  contains the present time in time-marching scheme
C     /WAVE1/  contains physical wave parameters
C     /WAVE2/  contains dimensionless wave parameters
C     /IWT1/ & /IWT2/ contain information on the input wave train
C     /BOT1/   contains physical properties of bottom
C     /BOT2/   contains dimensionless properties of bottom
C     /BOT3/ & /BOT4/ contain info on physical structure geometry
C     /BOT5/   contains information on normalized structure geometry
C     /ULAYER/ contains properties of the permeable underlayer
C     /HYDROE/ contains hydrodynamic quantities of external flow
C     /HYDROI/ contains hydrodynamic quantities of internal flow
C     /HYDROG/ contains hydrodynamic quantities related to both
C              external and internal flow
C     /VECONE/ contains vectors and matrices of Region 1
C     /VECTWO/ contains vectors of Region 2
C     /VECTRE/ contains vectors of Region 3
C     /ETASEA/ contains surface elevations at the seaward boundary
C     /IWLINE/ & /DWLINE/ contain quantities associated with the
C              movements of computational upper and lower waterlines;
C              "I" is for integers, "D" for real*8
C     /STATV/ & /STATR/ contain variables representing mean, maximum,
C              and minimum values
C     /STATT/  contains timing of statistical calculations
C     /STATI/  contains upper and lower waterline nodes at the
C              beginning and the end of statistical calculations
C     /ENERG/  contains quantities related to wave energy
C     /MASSB/  contains quantities to check mass balance
C     /ARMOR1/ contains input parameters for armor stability
C     /ARMOR2/ contains computed parameters for armor stability
C     /ARMOR3/ contains stability numbers and their timing, and
```

```
C             spatial variations at the time of critical stability
C     /ARMOR4/ contains waterline nodes related to armor stability
C     /SAVB/   contains nodal information for saving "B"
C     /DIAGV1/ and /DIAGV2/ contain indicators and saving
C              variables for diagnostic purposes
C     /SEQI/, /SEQD/, and /SEQC/ contain information related to
C              sequential computation for ICOMP=1 and ICOMP=2;
C              "I" is for integers, "D" for real*8, "C" for character
C
      COMMON /CONSTA/ PI,GRAV
      COMMON /OPTION/ ISYST,IBOT,ISALT
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /TEMPO/  TIME
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  WLO,WL,UR,SURF,SIGMA
      COMMON /IWT1/   NDATA
      COMMON /IWT2/   ETAMA,ETAMI,ETA(MXTIM)
      COMMON /BOT1/   DSEAP,FWP
      COMMON /BOT2/   DSEA,DSEA2,FW,TSLOPS
      COMMON /BOT3/   NUSEG,NLSEG
      COMMON /BOT4/   WUSEG(MXSEG),TUSEG(MXSEG),
     2                XUSEG(MXSEG),ZUSEG(MXSEG),
     3                WLSEG(MXSEG),TLSEG(MXSEG),
     4                XLSEG(MXSEG),ZLSEG(MXSEG)
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
      COMMON /VECTWO/ FTWO(MXNOD),GTWO(MXNOD)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /ETASEA/ ETAI,ETAR,ESEA(6)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /DWLINE/ DELRP(MXRUN),DELR(MXRUN),RU(2,MXRUN)
      COMMON /STATV/  VMEAN(12,MXNOD),VMAX(6,MXNOD),VMIN(6,MXNOD)
      COMMON /STATR/  RMEAN(4,MXRUN),RMAX(2,MXRUN),RMIN(2,MXRUN)
      COMMON /STATT/  TSTAT1,TSTAT2,TSPAN
      COMMON /STATI/  IST1,IST2,IWT1,IWT2
      COMMON /ENERG/  EEXTMN(4,MXNOD),EEXT(5,MXNOD),
     2                EINTMN(4,MXNOD),EINT(6,MXNOD)
      COMMON /MASSB/  VMASS(4,MXNOD)
      COMMON /ARMOR1/ SG,TANPHI,C2,C3,CD,CL,CM,AMAX,AMIN
      COMMON /ARMOR2/ CSTAB1,CSTAB2,AMAXS,AMINS,E2,
     +                SSLOPE(MXNOD),CTAN(MXNOD),E3PRE(MXNOD)
      COMMON /ARMOR3/ SNR(MXNOD),SNSX(MXNOD),TSNSX(MXNOD),
     +                ATMIN(7,MXNOD),SNSC,TSNSC
      COMMON /ARMOR4/ JSTAB,JSNSC,JSTABM,JSATM,ISATM,IWATM
```

```
      COMMON /SAVB/   NNODB,NODB(MXSAV)
      COMMON /DIAGV1/ ISOSAV(3),ISSAV(3),IWSAV(3),M5SAV(2)
      COMMON /DIAGV2/ DSSAV(3),DWSAV(3),U1SAV(6,MXNOD),U3SAV(6,MXNOD),
     2                USAV(3,MXNOD),ESAV(3,MXNOD),
     3                HPSAV(3,MXNOD),UPSAV(3,MXNOD),AMPSAV(3,MXNOD)
      COMMON /SEQI/   MSEQ(0:40),MULTIQ(40)
      COMMON /SEQD/   EPSIQ(2,40)
      COMMON /SEQC/   FDOC(40),FMSG(40),FPUT(40),FVAL(40)
C
C     MBEGIN=1 corresponding to t=0 unless ICOMP=2
      DATA MBEGIN /1/
      DATA MSGS /' Job to be performed:                               ',
     1          '     DIAGNOSTIC part only, from the beginning t=0',
     2          '     DIAGNOSTIC part only,                          ',
     3          '     ACTUAL part only                               ',
     4          '     BOTH diagnostic and actual parts               '/
C
C             NOTE
C             ------------------------------------------------------
C             In the comment lines throughout this program, a subrou-
C             tines is customarily referred to by its number followed
C             by its name, with the number written between brackets.
C             For example, (01) OPENIO refers to Subroutine 01 OPENIO.
C
C             READ INPUT AND OPEN FILES
C             ------------------------------------------------------
C             (56) STOPP  executes a programmed stop
C             (01) OPENIO manages opening of files
C             (55) OPENF  opens a specified file
C             (04) INPUT1 reads data from the primary input data file
C             (05) INPUT2 reads the input wave train
C             ------------------------------------------------------
      WRITE (*,1010)
      READ  (*,*) ICOMP
      IF (ICOMP.LT.1.OR.ICOMP.GT.4) CALL STOPP (1,1)
      WRITE (*,1020) MSGS(0),MSGS(ICOMP)
      IF (ICOMP.EQ.2) THEN
        CALL OPENF (1,19,'SEQUENCE  ','OLD')
        READ (19,9000) ISEQ
        WRITE (*,1025) ISEQ+1
      ENDIF
      WRITE (*,1030)
      READ  (*,5000) FINP1
      CALL OPENIO (ICOMP,ISEQ,FINP1)
      CALL INPUT1 (FINP2)
      CALL OPENF  (1,12,FINP2,'OLD')
      CALL INPUT2
C
C             GROUND WORK
C             ------------------------------------------------------
C             (07) BOTTOM computes normalized structure geometry
C             (10) PARAM  calculates parameters
C             (12) INITSQ specifies initial condition for ICOMP=2
C             (11) INIT   specifies initial condition for ICOMP.NE.2
C             (51) DOC1   writes essential information
```

```
C            (18) WRSEQ  writes sequential information
C            ---------------------------------------------------
      CALL BOTTOM
      CALL PARAM
      IF (ICOMP.EQ.2) THEN
        CALL INITSQ (ISEQ,MBEGIN)
      ELSE
        CALL INIT
      ENDIF
      CALL DOC1 (ICOMP,ISEQ)
      IF (ICOMP.NE.3) THEN
        WRITE (*,1040)
        READ  (*,*) IYES
        IF (IYES.NE.1) THEN
          WRITE (*,1050)
          IF (ICOMP.EQ.2) THEN
            WRITE (*,1060) ISEQ,FINP1,ISEQ
            CALL WRSEQ (ISEQ-1,0)
          ENDIF
          STOP
        ENDIF
      ENDIF
      WRITE (*,1070)
C
C            SELF-ADJUSTING MECHANISM
C            ---------------------------------------------------
C            This mechanism self-adjusts INVT, DELTAS, DELTAW
C            For explanation on these parameters, see GENERAL NOTES
C              at the beginning of this main program
C            Loop with bigger number nests to one with smaller number
C            . 510 represents one computation unit indicated by M
C            . 520 constitutes one try for a computation unit,
C                  indicated by ITRY
C            . 530 performs time-marching from one time level to the
C                  next, indicated by N
C            ---------------------------------------------------
      IPROB = 0
      M = MBEGIN-1
  510 IF (M.LT.MWAVE) THEN
C       --------> 510 Begins <--------------------------------------
        M    = M+1
        ITRY = 0
  520   CONTINUE
C         ------> 520 begins <--------------------------------------
C              . (13) INITM initializes values at the beginning of a
C                computation unit
C              . (19) CINVT assigns value to INVT & determines NEND
C              . (20) CINVTD calculates computation parameters that
C                are dependent on INVT
C              . NEND = final time level for a computation unit
          ITRY = ITRY+1
          IF (ICOMP.EQ.3) THEN
            NEND = 0
          ELSE
            IF (IPROB.NE.0) CALL INITM
```

```
          CALL CINVT (M,ITRY,IPROB,NEND,ICOMP,ISEQ)
          CALL CINVTD
          KDIV = 0
        ENDIF
        IPROB = 0
        N     = 0
  530   IF (IPROB.EQ.0.AND.N.LT.NEND) THEN
C         ----> 530 begins <---------------------------------------
C               . TIME = normalized time
C               . (20+y) MARCHy executes time-marching for Region y
C                 with y=1,2,3
C               . IPROB=33 & IPROB=55 indicates a change of value of
C                 DELTAS & DELTAW, respectively. KDIV, NDIV, IDIV,
C                 SDIV, and WDIV record information associated with
C                 this change.
          N = N+1
          TIME = DBLE(M-1) + DBLE(N)/DBLE(INVT)
          CALL MARCH1 (0,M,ITRY,IPROB,DVALUE)
C               ::
                IF (IPROB.EQ.33) THEN
                  KDIV = KDIV+1
                  IF (KDIV.GT.MXDIV) CALL STOPP (2,2)
                  NDIV(KDIV) = N
                  IDIV(KDIV) = 33
                  SDIV(KDIV) = DVALUE
                  WDIV(KDIV) = DELTAW
                  IPROB=0
                ENDIF
C               ::
          IF (IPROB.EQ.0) THEN
            CALL MARCH2
            CALL MARCH3 (0,M,ITRY,IPROB,DVALUE)
C               ::
                IF (IPROB.EQ.55) THEN
                  KDIV = KDIV+1
                  IF (KDIV.GT.MXDIV) CALL STOPP (2,2)
                  NDIV(KDIV) = N
                  IDIV(KDIV) = 55
                  SDIV(KDIV) = DELTAS
                  WDIV(KDIV) = DVALUE
                  IPROB=0
                ENDIF
C               ::
          ENDIF
C         ----> 530 Ends <-----------------------------------------
        GOTO 530
        ENDIF
C       ------> 520 Ends <-----------------------------------------
      IF (IPROB.NE.0) GOTO 520
C               . (01) MAIN2 manages the actual time-marching for
C                 computation unit M after a set of working
C                 combinations of (INVT,DELTAS,DELTAW) has been
C                 established
C               . (54) NSI calculates numerical stability indicator
C                 OMEGA
```

```
C                . For explanation on ICOMP, see GENERAL NOTES at the
C                  beginning of this main program
      IF (ICOMP.EQ.3) THEN
        READ (14,4910) MREAD,INVT,KDIV,EPSI1,EPSI2
        DO 120 IDV = 1,KDIV
          READ (14,4920) NDIV(IDV),IDIV(IDV),SDIV(IDV),WDIV(IDV)
  120   CONTINUE
        CALL NSI (OMEGA,INVT)
        WRITE (98,9810) M,INVT,OMEGA,NDIV(1),SDIV(1),WDIV(1)
        DO 130 IDV = 2,KDIV
          WRITE (98,9820) NDIV(IDV),SDIV(IDV),WDIV(IDV)
  130   CONTINUE
        IF (M.NE.MREAD) CALL STOPP (3,4)
        CALL CINVTD
        CALL MAIN2 (M,KDIV,NDIV,IDIV,SDIV,WDIV)
        WRITE (*,1080) M
        IDUM = MOD(M,5)
        IF (IDUM.EQ.0) THEN
          DO 140 ISQ = 1,ISEQ-1
            IF (M.EQ.MSEQ(ISQ)) THEN
              CALL READER (ISQ,MRD,MURD,DSSV,DWSV,ESV1,ESV2)
              WRITE  (*,9910) MRD+1
              WRITE (99,9910) MRD+1
              GOTO 910
            ENDIF
  140     CONTINUE
        ENDIF
  910   CONTINUE
      ELSE
C                 ::
                  KDIV = KDIV+1
                  IF (KDIV.GT.MXDIV) CALL STOPP (2,2)
                  NDIV(KDIV) = NEND
                  IDIV(KDIV) = 0
                  SDIV(KDIV) = DELTAS
                  WDIV(KDIV) = DELTAW
C                 ::
        WRITE  (*,1090) M,ITRY,INVT,DELTAS,DELTAW
        CALL NSI (OMEGA,INVT)
        WRITE (98,9830) M,ITRY,INVT,OMEGA,NDIV(1),SDIV(1),WDIV(1)
        DO 150 IDV = 2,KDIV
          WRITE (98,9840) NDIV(IDV),SDIV(IDV),WDIV(IDV)
  150   CONTINUE
        IF (ICOMP.LE.2) THEN
          DO 170 NUNIT = 48,49
            WRITE (NUNIT,4910) M,INVT,KDIV,EPSI1,EPSI2
            DO 160 IDV = 1,KDIV
              WRITE (NUNIT,4920) NDIV(IDV),IDIV(IDV),
     +                           SDIV(IDV),WDIV(IDV)
  160       CONTINUE
  170     CONTINUE
        ELSE
          CALL INITM
          CALL MAIN2 (M,KDIV,NDIV,IDIV,SDIV,WDIV)
          WRITE (*,1080) M
```

```
          ENDIF
          CALL SAVEM
          IDUM = MOD(M,5)
          IF (IDUM.EQ.0.AND.ICOMP.LE.2) CALL SAVE5 (1,1,M)
        ENDIF
C       --------> 510 Ends <-----------------------------------------
      GOTO 510
      ENDIF
C
C             FINISHING
C             -----------------------------------------------------
C             (18) WRSEQ writes sequential information
C             (53) DOC3  documents results after time-marching
C                        computation
      IF (ICOMP.EQ.3) THEN
        WRITE (98,9850)
      ELSE
        WRITE (98,9860)
      ENDIF
      IF (ICOMP.LE.2) THEN
        MSEQ(ISEQ)     = MWAVE
        MULTIQ(ISEQ)   = MULTIF
        EPSIQ(1,ISEQ) = EPSI1
        EPSIQ(2,ISEQ) = EPSI2
        CALL WRSEQ (ISEQ,1)
        WRITE (*,1012)
      ELSE
        CALL DOC3
        WRITE (*,1014)
      ENDIF
C
C             FORMATS
C             -----------------------------------------------------
 1010 FORMAT (' PBREAK begins ...'/
     2        ' Select job: 1 = diagnostic only'/
     3 '             2 = continuation of 1 or 2'/
     4 '             3 = actual only (diagnostic already done)'/
     5 '             4 = diagnostic & actual')
 1012 FORMAT (' TMAX reached. Diagnostic part completed.' /
     +        ' May now proceed with actual part (ICOMP=3).')
 1014 FORMAT (' TMAX reached. PBREAK computation completed.')
 1020 FORMAT (A49/A49)
 1025 FORMAT ('     Sequence number',I3)
 1030 FORMAT (' Enter name of primary input data file')
 1040 FORMAT (' Time-marching computation is about to begin'/
     +        ' Proceed? 1=yes')
 1050 FORMAT (' :: At the user's request'/
     +      '   PBREAK was aborted before time-marching computation')
 1060 FORMAT ('    The sequence just aborted was number  ',I8/
     2       '    Primary input data file used was   "',A,'"'/
     3       '    Please make changes on that file if so desired'/
     4       '    The next sequential computation will be'/
     5       '                                sequence number',I7)
 1070 FORMAT (' Time-marching computation begins ...')
 1080 FORMAT ('         :: Finished unit ',I8)
```

```
 1090 FORMAT (' M,ITRY,INVT,DELTAS,DELTAW=',3I8,2F10.6)
 5000 FORMAT (A10)
 4910 FORMAT (3I8,2D18.9)
 4920 FORMAT (2I8,2D18.9)
 9000 FORMAT (8I8)
 9810 FORMAT (2I8,F8.1,I8,2F10.6)
 9820 FORMAT (24X,I8,2F10.6)
 9830 FORMAT (3I8,F8.1,I8,2F10.6)
 9840 FORMAT (32X,I8,2F10.6)
 9850 FORMAT (54(1H-))
 9860 FORMAT (62(1H-))
 9910 FORMAT (' :: Quantities initialized using stored values'/
     +        '               at the beginning of computation unit',I8)
C
      STOP
      END
C
C .................. END OF MAIN PROGRAM PBREAK ..................
C >>>>>>>>>>>>>>>>>>>>>>>>>>>> BLOCK DATA <<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
      BLOCK DATA
C
      CHARACTER*10 FDOC,FMSG,FPUT,FVAL
      COMMON /SEQC/  FDOC(40),FMSG(40),FPUT(40),FVAL(40)
      DATA FDOC /'CDOC01    ','CDOC02    ','CDOC03    ','CDOC04    ',
     2           'CDOC05    ','CDOC06    ','CDOC07    ','CDOC08    ',
     3           'CDOC09    ','CDOC10    ','CDOC11    ','CDOC12    ',
     4           'CDOC13    ','CDOC14    ','CDOC15    ','CDOC16    ',
     5           'CDOC17    ','CDOC18    ','CDOC19    ','CDOC20    ',
     6           'CDOC21    ','CDOC22    ','CDOC23    ','CDOC24    ',
     7           'CDOC25    ','CDOC26    ','CDOC27    ','CDOC28    ',
     8           'CDOC29    ','CDOC30    ','CDOC31    ','CDOC32    ',
     9           'CDOC33    ','CDOC34    ','CDOC35    ','CDOC36    ',
     +           'CDOC37    ','CDOC38    ','CDOC39    ','CDOC40    '/
      DATA FMSG /'CMSG01    ','CMSG02    ','CMSG03    ','CMSG04    ',
     2           'CMSG05    ','CMSG06    ','CMSG07    ','CMSG08    ',
     3           'CMSG09    ','CMSG10    ','CMSG11    ','CMSG12    ',
     4           'CMSG13    ','CMSG14    ','CMSG15    ','CMSG16    ',
     5           'CMSG17    ','CMSG18    ','CMSG19    ','CMSG20    ',
     6           'CMSG21    ','CMSG22    ','CMSG23    ','CMSG24    ',
     7           'CMSG25    ','CMSG26    ','CMSG27    ','CMSG28    ',
     8           'CMSG29    ','CMSG30    ','CMSG31    ','CMSG32    ',
     9           'CMSG33    ','CMSG34    ','CMSG35    ','CMSG36    ',
     +           'CMSG37    ','CMSG38    ','CMSG39    ','CMSG40    '/
      DATA FPUT /'CINPUT01  ','CINPUT02  ','CINPUT03  ','CINPUT04  ',
     2           'CINPUT05  ','CINPUT06  ','CINPUT07  ','CINPUT08  ',
     3           'CINPUT09  ','CINPUT10  ','CINPUT11  ','CINPUT12  ',
     4           'CINPUT13  ','CINPUT14  ','CINPUT15  ','CINPUT16  ',
     5           'CINPUT17  ','CINPUT18  ','CINPUT19  ','CINPUT20  ',
     6           'CINPUT21  ','CINPUT22  ','CINPUT23  ','CINPUT24  ',
     7           'CINPUT25  ','CINPUT26  ','CINPUT27  ','CINPUT28  ',
     8           'CINPUT29  ','CINPUT30  ','CINPUT31  ','CINPUT32  ',
     9           'CINPUT33  ','CINPUT34  ','CINPUT35  ','CINPUT36  ',
     +           'CINPUT37  ','CINPUT38  ','CINPUT39  ','CINPUT40  '/
      DATA FVAL /'CVAL01    ','CVAL02    ','CVAL03    ','CVAL04    ',
```

```
     2             'CVAL05    ','CVAL06    ','CVAL07    ','CVAL08    ',
     3             'CVAL09    ','CVAL10    ','CVAL11    ','CVAL12    ',
     4             'CVAL13    ','CVAL14    ','CVAL15    ','CVAL16    ',
     5             'CVAL17    ','CVAL18    ','CVAL19    ','CVAL20    ',
     6             'CVAL21    ','CVAL22    ','CVAL23    ','CVAL24    ',
     7             'CVAL25    ','CVAL26    ','CVAL27    ','CVAL28    ',
     8             'CVAL29    ','CVAL30    ','CVAL31    ','CVAL32    ',
     9             'CVAL33    ','CVAL34    ','CVAL35    ','CVAL36    ',
     +             'CVAL37    ','CVAL38    ','CVAL39    ','CVAL40    '/
      END
C
C ....................... END OF BLOCK DATA .......................
C >>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 01 OPENIO <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 01 OPENIO opens files
C
      SUBROUTINE OPENIO (ICOMP,ISEQ,FINP1)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*10 FINP1
      CHARACTER*10 FDOC,FMSG,FPUT,FVAL
      COMMON /SEQI/  MSEQ(0:40),MULTIQ(40)
      COMMON /SEQD/  EPSIQ(2,40)
      COMMON /SEQC/  FDOC(40),FMSG(40),FPUT(40),FVAL(40)
C
      CALL OPENF (1,11,FINP1,'OLD')
      IF (ICOMP.EQ.1) THEN
        ISEQ = 1
C       ::          Non-destructive file opening is to avoid
C                   accidental loss of data
        CALL OPENF (1,48,FPUT(ISEQ)  ,'NEW')
        CALL OPENF (1,98,FDOC(ISEQ)  ,'NEW')
        CALL OPENF (1,99,FMSG(ISEQ)  ,'NEW')
        CALL OPENF (1,49,'TRANSFER  ','NEW')
      ELSEIF (ICOMP.EQ.2) THEN
        CALL OPENF (1,47,FPUT(ISEQ),'OLD')
C       ::          Destructive file opening is to provide simple
C                   transfer of information from one sequence of
C                   computation to the next
        CALL OPENF (999,48,FPUT(ISEQ+1),'999')
        CALL OPENF (999,98,FDOC(ISEQ+1),'999')
        CALL OPENF (999,99,FMSG(ISEQ+1),'999')
        CALL OPENF (999,49,'TRANSFER  ','999')
      ELSE
        IF (ICOMP.EQ.3) THEN
          CALL OPENF (1,14,'TRANSFER  ','OLD')
          CALL OPENF (1,19,'SEQUENCE  ','OLD')
          READ (19,9000) ISEQ,INDICA
          IF (INDICA.NE.1) CALL STOPP (5,5)
        ENDIF
        CALL OPENF (1,98,'ODOC      ','NEW')
        CALL OPENF (1,99,'OMSG      ','NEW')
        CALL OPENF (1,51,'OSTAT     ','NEW')
        CALL OPENF (1,52,'OSPACE    ','NEW')
        CALL OPENF (1,53,'OENERG    ','NEW')
```

```
        CALL OPENF (1,54,'OMASS     ','NEW')
        CALL OPENF (1,55,'OSTAB     ','NEW')
        CALL OPENF (1,61,'OSEAWAV   ','NEW')
        CALL OPENF (1,62,'ORUNUPU   ','NEW')
        CALL OPENF (1,63,'ORUNUPL   ','NEW')
      ENDIF
      IF (ICOMP.EQ.2.OR.ICOMP.EQ.3) THEN
        READ (19,9000) (MSEQ(I),   I=1,ISEQ)
        READ (19,9000) (MULTIQ(I), I=1,ISEQ)
        READ (19,8800) (EPSIQ(1,I),I=1,ISEQ)
        READ (19,8800) (EPSIQ(2,I),I=1,ISEQ)
        CLOSE (19)
      ENDIF
      IF (ICOMP.EQ.2) ISEQ=ISEQ+1
 8800 FORMAT (4D18.9)
 9000 FORMAT (8I8)
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 01 OPENIO ...................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 02 MAIN2  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 02 MAIN2 executes the actual time-marching for
C     computation unit M after a working set of the instrumental
C     computation parameters has been established by the
C     diagnostic part
C
      SUBROUTINE MAIN2 (M,KDIV,NDIV,IDIV,SDIV,WDIV)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION NDIV(KDIV),IDIV(KDIV)
      DIMENSION SDIV(KDIV),WDIV(KDIV)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /TEMPO/  TIME
      NSAV=0
      IF (M.GT.MSTAT)      CALL STAT  (1)
      IF (M.EQ.(MSTAT+1)) CALL SAVET (1)
C
C            Notes
C            ---------------------------------------------------------
C          . A computation unit will be divided into more than one
C            divisions if DELTAS and/or DELTAW vary within the unit,
C            each division having constant DELTAS and DELTAW
C          . KDIV = number of divisions
C          . Complication in this routine is due to the fact that the
C            change in the value of DELTAS or DELTAW occurs in the
C            middle of the time-marching procedure
C          . If DELTAS and DELTAW do not vary, KDIV=1, proceed
C            directly to the second block
C          . The first block is executed if KDIV>1
C          . (47) STAT   calculates statistical quantities
```

```
C           . (03) KERNEL executes subsystems of PBREAK
C
C             First block
C             ----------------------------------------------------------
      DO 550 IDV = 1,KDIV-1
        NBEGIN = NSAV+1
        NFINAL = NDIV(IDV)
        DELTAS = SDIV(IDV)
        DELTAW = WDIV(IDV)
        NSAV   = NFINAL
        DO 540 N = NBEGIN,NFINAL-1
          TIME = DBLE(M-1) + DBLE(N)/DBLE(INVT)
          CALL MARCH1 (1,M,777,IPROB,0.D+00)
          IF (IPROB.NE.0) CALL STOPP (6,6)
          CALL MARCH2
          CALL MARCH3 (1,M,777,IPROB,0.D+00)
          IF (IPROB.NE.0) CALL STOPP (6,6)
          CALL KERNEL (M,N)
  540   CONTINUE
        N = NFINAL
        TIME   = DBLE(M-1) + DBLE(N)/DBLE(INVT)
        MODE   = IDIV(IDV)
        DVALUE = SDIV(IDV+1)
        CALL MARCH1 (MODE,M,777,IPROB,DVALUE)
        IF (IPROB.NE.0) CALL STOPP (6,6)
        CALL MARCH2
        MODE   = IDIV(IDV)
        DVALUE = WDIV(IDV+1)
        CALL MARCH3 (MODE,M,777,IPROB,DVALUE)
        IF (IPROB.NE.0) CALL STOPP (6,6)
        CALL KERNEL (M,N)
  550 CONTINUE
C
C             Second block
C             ----------------------------------------------------------
      NBEGIN = NSAV+1
      NTOP   = NDIV(KDIV)
      DELTAS = SDIV(KDIV)
      DELTAW = WDIV(KDIV)
      DO 560 N = NBEGIN,NTOP
        TIME = DBLE(M-1) + DBLE(N)/DBLE(INVT)
        CALL MARCH1 (1,M,777,IPROB,0.D+00)
        IF (IPROB.NE.0) CALL STOPP (6,6)
        CALL MARCH2
        CALL MARCH3 (1,M,777,IPROB,0.D+00)
        IF (IPROB.NE.0) CALL STOPP (6,6)
        CALL KERNEL (M,N)
  560 CONTINUE
C
      IF (M.GT.MSTAT) CALL STAT  (3)
      IF (M.EQ.MWAVE) CALL SAVET (2)
C
      RETURN
      END
C
```

```
C .................. END OF SUBROUTINE 02 MAIN2 ..................
C >>>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 03 KERNEL <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 03 KERNEL executes subsystems of PBREAK
C
      SUBROUTINE KERNEL (M,N)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /SAVB/   NNODB,NODB(MXSAV)
C
C          . (46) RUNUP computes run-up on the upper and lower slopes
C          . (47) STAT computes statistical values: mean, maximum,
C            and minimum values
C          . (48) ENERGY computes energy quantities in (47) STAT
C          . (50) STABNO computes stability number of armor units
C          . (52) DOC2 (MODE,M) stores computed results at designated
C            time during time-marching computation
C               : MODE=1 stores standard time series
C               : MODE=2 stores "A"
C               : MODE=3 opens and closes files for storing "B"
C               : MODE=4 stores "B"
C          . See (52) DOC2 for an explanation on "A" and "B"
C          ----------------------------------------------------------
      CALL RUNUP
      CALL CQBUB
      IF (M.GT.MSTAT) THEN
        CALL STAT (2)
        CALL STABNO
      ENDIF
      IDUMO = MOD(N,NHOP)
      IF (IDUMO.EQ.0) THEN
        CALL DOC2 (1,M)
        IF (NNODB.GT.0) THEN
          IDUM1 = MOD(M,100)
          IF (IDUM1.EQ.1.AND.N.EQ.NHOP) CALL DOC2 (3,M)
          CALL DOC2 (4,M)
        ENDIF
      ENDIF
      IF (M.GE.MSAVA1.AND.M.LE.MSAVA2) THEN
        DO 100 IA = 1,NTIMES
          NSAVA = IA*INVT/NTIMES
          IF (N.EQ.NSAVA) CALL DOC2 (2,M)
  100   CONTINUE
      ENDIF
      RETURN
      END
C
C .................. END OF SUBROUTINE 03 KERNEL ..................
C >>>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 04 INPUT1 <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 04 INPUT1 reads most of input data
C
```

```
      SUBROUTINE INPUT1 (FINP2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      CHARACTER*10 FINP2
      CHARACTER*5  COMMEN(15)
      COMMON /CONSTA/ PI,GRAV
      COMMON /OPTION/ ISYST,IBOT,ISALT
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  WLO,WL,UR,SURF,SIGMA
      COMMON /BOT1/   DSEAP,FWP
      COMMON /BOT2/   DSEA,DSEA2,FW,TSLOPS
      COMMON /BOT3/   NUSEG,NLSEG
      COMMON /BOT4/   WUSEG(MXSEG),TUSEG(MXSEG),
     2                XUSEG(MXSEG),ZUSEG(MXSEG),
     3                WLSEG(MXSEG),TLSEG(MXSEG),
     4                XLSEG(MXSEG),ZLSEG(MXSEG)
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /DWLINE/ DELRP(MXRUN),DELR(MXRUN),RU(2,MXRUN)
      COMMON /ARMOR1/ SG,TANPHI,C2,C3,CD,CL,CM,AMAX,AMIN
      COMMON /SAVB/   NNODB,NODB(MXSAV)
C
C           COMMENT LINES
C           ----------------------------------------------------------
C           NLINES = number of comment lines preceding primary
C                    input data
C           Comment lines of the primary input data file are read and
C           then written as the heading of the output files 'ODOC' &
C           'OMSG' (ICOMP=3 or ICOMP=4) or 'CDOCxx' & 'CMSGxx'
C           (ICOMP=1 or ICOMP=2) where xx is the sequence number
C           ----------------------------------------------------------
      READ  (11,1110) NLINES
      DO 110 L = 1,NLINES
        READ  (11,1120) (COMMEN(I),I=1,15)
        WRITE (98,1120) (COMMEN(I),I=1,15)
        WRITE (99,1120) (COMMEN(I),I=1,15)
  110 CONTINUE
C
C           OPTIONS
C           ----------------------------------------------------------
C           ISYST=1: International System of Units (SI) is used
C                =2: US Customary System of Units (USCS) is used
C           IBOT =1: "Type 1" bottom data (width-slope)
C                =2: "Type 2" bottom data (coordinates)
C           MSAVA1, MSAVA2, AND NTIMES:
C                "A" is saved NTIMES (>1) times at equal intervals per
C                reference wave period from computation unit MSAVA1 to
```

```
C              MSAVA2, inclusive
C           NNODB = number of nodes for which "B" is to be saved
C           NODB(I) = I-th node number for which "B" is to be saved
C           See (52) DOC2 for an explanation on "A" and "B"
C           ------------------------------------------------------------
C
      READ  (11,1130) ISYST
      READ  (11,1130) IBOT
      READ  (11,1110) MSAVA1,MSAVA2,NTIMES
      READ  (11,1110) NNODB
      IF (NNODB.GT.0) READ (11,1110) (NODB(I),I=1,NNODB)
      IF (MSAVA2.LT.MSAVA1) CALL STOPP (7,7)
C
C           INPUT WAVE TRAIN
C           ------------------------------------------------------------
C           FINP2 = name of the file containing the input wave train
C           TMAX  = normalized length of the input wave train
C                 = normalized duration of computation
C           MWAVE = number of computation units
C           ------------------------------------------------------------
      READ  (11,5000) FINP2
      READ  (11,1150) TMAX
      ITMAX = INT(TMAX)
      RES   = DMOD(TMAX,1.D+00)
      IF (DABS(RES).LT.1.D-08) THEN
        MWAVE = ITMAX
      ELSE
        MWAVE = ITMAX+1
      ENDIF
C
C           INSTRUMENTAL COMPUTATION PARAMETERS
C           The computation parameters in this block
C           . are instrumental to the success or failure of the
C             computation,
C           . user-adjustable, and
C           . may be varied from one sequence of computation to
C             another, except for INITS, which must be constant
C             for the whole computation duration
C           ------------------------------------------------------------
C           INITS  = number of spatial nodes at which the upper slope
C                    is located below SWL
C           MULTIF = multiplication factor associated with INVT, where
C                    INVT is the number of time steps in one reference
C                    wave period [See (10) PARAM for more explanation]
C           DELTAS & DELTAW as input are initial values for normalized
C                  water depth defining computational upper and lower
C                  waterlines, respectively
C           EPSI1,EPSI2 = damping coefficients
C           ------------------------------------------------------------
      READ  (11,1120) COMMEN(1)
      READ  (11,1110) INITS
      READ  (11,1110) MULTIF
      READ  (11,1150) DELTAS,DELTAW
      READ  (11,1150) EPSI1,EPSI2
      READ  (11,1120) COMMEN(1)
```

```
C
C          ADDITIONAL COMPUTATION PARAMETERS
C          -------------------------------------------------------
C          MSTAT:  Statistical calculations are performed excluding
C                  the first MSTAT computation units
C          NRATE = rate (data points per wave period) of storing
C                  output time series
C          NDELR = number of "DELRP"s to be specified
C          DELRP = physical water depth associated with visual or
C                  measured waterline
C                  . in millimeters if ISYST=1 (SI)
C                  . in inches      if ISYST=2 (USCS)
C          -------------------------------------------------------
      READ  (11,1110) MSTAT
      READ  (11,1110) NRATE
      READ  (11,1110) NDELR
      DO 120 L = 1,NDELR
        READ (11,1150) DELRP(L)
  120 CONTINUE
C
C          CONSTANTS
C          -------------------------------------------------------
C          PI   = 3.141592...
C          GRAV = gravitational acceleration
C                 . in m/sec**2  if ISYST=1 (SI)
C                 . in ft/sec**2 if ISYST=2 (USCS)
C          -------------------------------------------------------
      PI = 4.D+00*DATAN(1.D+00)
      IF (ISYST.EQ.1) THEN
        GRAV = 9.81D+00
      ELSE
        GRAV = 32.2D+00
      ENDIF
C
C          WAVE PROPERTIES AND FRICTION FACTOR
C          -------------------------------------------------------
C          HREFP = physical reference wave height
C                  . in meters if ISYST=1 (SI)
C                  . in feet   if ISYST=2 (USCS)
C          TP    = physical reference wave period, in seconds
C          HREFP and TP are used to normalize the governing
C                  equations
C          SIGMA is the ratio between horizontal and vertical length
C                  scales
C          FWP   = bottom friction factor
C          -------------------------------------------------------
      READ  (11,1150) HREFP,TP
      READ  (11,1150) FWP
      SIGMA = TP*DSQRT(GRAV/HREFP)
C
C          STRUCTURE GEOMETRY
C          -------------------------------------------------------
C          DSEAP and DSEA = physical & normalized water depth below
C            SWL at seaward boundary
C          TSLOPS = tangent of slope used to define
```

```
C          "surf similarity parameter"
C         There are two slopes: the upper slope and the lower slope.
C          Each slope is divided into segments of different
C          inclination. In the following, x=U for the upper slope,
C          x=L for the lower slope.
C         NxSEG  = number of segments
C         For segment i starting from the seaward boundary:
C          WxSEG(i)  = physical horizontal width
C          TxSLOP(i) = tangent of slope (+ upslope, - downslope)
C          XxSEG(i)  = physical x-coordinate of the segment's
C                      seaward-end (+ landward of seaward boundary)
C          ZxSEG(i)  = physical z-coordinate of the segment's
C                      seaward-end (+ above SWL)
C         DSEAP,WxSEG,XxSEG,ZxSEG are in meters if ISYST=1 (SI),
C                                    feet   if ISYST=2 (USCS)
C         ----------------------------------------------------------
      READ  (11,1150) DSEAP
      READ  (11,1150) TSLOPS
C                              :: Upper slope ::
      READ  (11,1110) NUSEG
      IF (IBOT.EQ.1) THEN
        DO 130 K = 1,NUSEG
          READ (11,1150) WUSEG(K),TUSEG(K)
  130   CONTINUE
      ELSE
        DO 140 K = 1,NUSEG+1
          READ (11,1150) XUSEG(K),ZUSEG(K)
  140   CONTINUE
      ENDIF
C                              :: Lower slope ::
      READ  (11,1110) NLSEG
      IF (IBOT.EQ.1) THEN
        DO 150 K = 1,NLSEG
          READ (11,1150) WLSEG(K),TLSEG(K)
  150   CONTINUE
      ELSE
        DO 160 K = 1,NLSEG+1
          READ (11,1150) XLSEG(K),ZLSEG(K)
  160   CONTINUE
      ENDIF
C
C         PROPERTIES OF WATER AND PERMEABLE UNDERLAYER
C         ----------------------------------------------------------
C         ISALT=0: fresh water
C              =1: sea water
C         TEMPER = temperature of water,
C                  in degrees Celcius    if ISYST=1 (SI)
C                     degrees Fahrenheit if ISYST=2 (USCS)
C         DPP = physical representative diameter of permeable
C               underlayer materials, in meters if ISYST=1 (SI)
C                                         feet  if ISYST=2 (USCS)
C         PORO = porosity of the permeable underlayer
C         ALPAO,BETAO = dimensionless constants
C         VISCO = kinematic viscosity of water
C         Call to (06) CVISCO is to calculate VISCO
```

```
C         ----------------------------------------------------------
      READ  (11,1120) COMMEN(1)
      READ  (11,1130) ISALT
      READ  (11,1150) TEMPER
      READ  (11,1150) DPP
      READ  (11,1150) PORO
      READ  (11,1150) ALPA0,BETA0
      READ  (11,1120) COMMEN(1)
      CALL CVISCO (ISYST,ISALT,TEMPER,VISCO)
C
C         ARMOR STABILITY
C         ----------------------------------------------------------
C         SG = specific gravity
C         C2 = area coefficient
C         C3 = volume coefficient
C         CD = drag coefficient
C         CL = lift coefficient
C         CM = inertia coefficient
C         TANPHI = armor friction factor
C         AMAX & AMIN = upper & lower bounds of horizontal fluid
C              acceleration of the external flow, normalized by
C              gravitational acceleration
C         ----------------------------------------------------------
      READ  (11,1120) COMMEN(1)
      READ  (11,1150) SG,TANPHI
      READ  (11,1150) C2,C3
      READ  (11,1150) CD,CL
      READ  (11,1150) CM
      READ  (11,1150) AMAX,AMIN
      READ  (11,1120) COMMEN(1)
C
C          FORMATS
C          ---------------------------------------------------------
 1110 FORMAT (3I8)
 1120 FORMAT (15A5)
 1130 FORMAT (2I1)
 1140 FORMAT (5I6)
 1150 FORMAT (2F13.6)
 5000 FORMAT (A10)
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 04 INPUT1 ..................
C >>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 05 INPUT2  <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 05 INPUT2 reads information on the input wave train
C
      SUBROUTINE INPUT2
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      CHARACTER CHAR*1
      COMMON /IWT1/  NDATA
      COMMON /IWT2/  ETAMA,ETAMI,ETA(MXTIM)
```

```
C
C             READ DATA POINTS
C             ------------------------------------------------------
      READ (12,5000) CHAR
      READ (12,9000) NDATA
      READ (12,8000) (ETA(I),I=1,NDATA)
 5000 FORMAT (A)
 8000 FORMAT (5D15.6)
 9000 FORMAT (I8)
C
C             FIND EXTREME VALUES
C             ------------------------------------------------------
C             ETA = given time series of free surface profile at
C                   seaward boundary
C             ETAMA,ETAMI = maximum,minimum of ETA
C             ------------------------------------------------------
      ETAMA = -1.D+03
      ETAMI =  1.D+03
      DO 100 I = 1,NDATA
        IF (ETA(I).GT.ETAMA) ETAMA=ETA(I)
        IF (ETA(I).LT.ETAMI) ETAMI=ETA(I)
  100 CONTINUE
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 05 INPUT2 ...................
C >>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 06 CVISCO  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 06 CVISCO calculates kinematic viscosity of water
C     using Tables B.1 and B.2 from the book:
C       Fischer, H.B., et al., 1979.
C       Mixing in Inland and Coastal Waters.
C       Academic Press.
C
      SUBROUTINE CVISCO (ISYST,ISALT,TEMPER,VISCO)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(16),Y(16),YSEA(6)
      DATA X      / 0.D+00, 5.D+00,10.D+00,15.D+00,20.D+00,
     +              25.D+00,30.D+00,35.D+00,40.D+00,45.D+00,
     +              50.D+00,60.D+00,70.D+00,80.D+00,90.D+00,
     +             100.D+00/
      DATA Y      /1.787D+00,1.519D+00,1.307D+00,1.140D+00,1.004D+00,
     +              0.893D+00,0.801D+00,0.724D+00,0.658D+00,0.602D+00,
     +              0.553D+00,0.475D+00,0.413D+00,0.365D+00,0.326D+00,
     +              0.294D+00/
      DATA YSEA   /1.80D+00,1.60D+00,1.40D+00,1.20D+00,1.10D+00,
     +              0.94D+00/
      DATA ORDER,CONVER /1.D-06,.3048D+00/
      DATA IMAX,XMAX /16,1.D+02/
      IF (ISALT.EQ.1) THEN
        IMAX =  6
        XMAX = 25.D+00
        DO 100 I = 1,IMAX
```

```
           Y(I) = YSEA(I)
  100    CONTINUE
       ENDIF
       IF (ISYST.EQ.2) TEMPER=(TEMPER-32.D+00)/1.8D+00
       IF (TEMPER.LT.0.D+00.OR.TEMPER.GT.XMAX) CALL STOPP (8,8)
       I = 1
  900 CONTINUE
         I    = I+1
         XDIF = TEMPER-X(I)
       IF (XDIF.GT.0.D+00) GOTO 900
       VISCO = Y(I) + XDIF*(Y(I)-Y(I-1))/(X(I)-X(I-1))
       IF (ISYST.EQ.2) THEN
         VISCO  = VISCO/CONVER**2
         TEMPER = TEMPER*1.8D+00 + 32.D+00
       ENDIF
       VISCO = ORDER*VISCO
C
       RETURN
       END
C
C .................. END OF SUBROUTINE 06 CVISCO ..................
C >>>>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 07 BOTTOM <<<<<<<<<<<<<<<<<<<<<<<<
C
C      Subroutine 07 BOTTOM calculates normalized structure geometry
C      and the space interval between two adjacent nodes [delta x]
C
       SUBROUTINE BOTTOM
C
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
       COMMON /CONSTA/ PI,GRAV
       COMMON /OPTION/ ISYST,IBOT,ISALT
       COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
      +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
       COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
      +                X,T,TX,XT,TTX,TTXX,TWOX
       COMMON /WAVE1/  HREFP,TP,WLOP
       COMMON /WAVE2/  WLO,WL,UR,SURF,SIGMA
       COMMON /BOT1/   DSEAP,FWP
       COMMON /BOT3/   NUSEG,NLSEG
       COMMON /BOT4/   WUSEG(MXSEG),TUSEG(MXSEG),
      2                XUSEG(MXSEG),ZUSEG(MXSEG),
      3                WLSEG(MXSEG),TLSEG(MXSEG),
      4                XLSEG(MXSEG),ZLSEG(MXSEG)
       COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
      +                ZL(MXNOD),TL(MXNOD)
C
C            COMPLETE SEGMENT DATA NOT SPECIFIED AS INPUT
C            (Dimensional)
C            Accomplished by calling (08) BOTA
C            ------------------------------------------------------
C            DSEAP = water depth below SWL at seaward boundary
C            There are two slopes: the upper slope and the lower slope.
C              Each slope is divided into segments of different
C              inclination. In the following, x=U for the upper slope,
```

```
C            x=L for the lower slope.
C          NxSEG  = number of segments
C          For segment i starting from the seaward boundary:
C            WxSEG(i)  = physical horizontal width
C            TxSLOP(i) = tangent of slope (+ upslope, - downslope)
C            XxSEG(i)  = physical x-coordinate of the segment's
C                        seaward-end (+ landward of seaward boundary)
C            ZxSEG(i)  = physical z-coordinate of the segment's
C                        seaward-end (+ above SWL)
C          DSEAP,WxSEG,XxSEG,ZxSEG are in meters if ISYST=1 (SI),
C                                        feet   if ISYST=2 (USCS)
C          ---------------------------------------------------------
      CALL BOTA (IBOT,DSEAP,NUSEG,WUSEG,TUSEG,XUSEG,ZUSEG)
      CALL BOTA (IBOT,DSEAP,NLSEG,WLSEG,TLSEG,XLSEG,ZLSEG)
      IF (XLSEG(NLSEG+1).LT.XUSEG(NUSEG+1)) CALL STOPP (9,9)
C
C          CALCULATE SPACE INTERVAL X BETWEEN TWO ADJACENT NODES
C          (Dimensional)
C          ---------------------------------------------------------
C          INITS = number of spatial nodes at which the upper slope
C                  is located below SWL
C          ---------------------------------------------------------
      K = 0
  910 CONTINUE
        IF (K.EQ.NUSEG) CALL STOPP (10,10)
        K = K+1
        CROSS = ZUSEG(K)*ZUSEG(K+1)
      IF (CROSS.GT.0.D+00) GOTO 910
      WINITS = XUSEG(K+1) - ZUSEG(K+1)/TUSEG(K)
      X      = WINITS/DBLE(INITS)
C
C          CALCULATE STRUCTURE GEOMETRY AT EACH NODE
C          (Dimensional)
C          Accomplished by calling (09) BOTB
C          ---------------------------------------------------------
C          JE = landward edge node
C          At node j:
C          . XUL(j) = x-coordinate of upper and lower slopes
C          . ZU(j),ZL(j) = z-coordinate of  upper,lower slope
C          . TU(j),TL(j) = tangent of local upper,lower slope
C          ---------------------------------------------------------
      DUM = XUSEG(NUSEG+1)/X
      JE  = INT(DUM)+1
      IF (JE.GT.MXNOD) CALL STOPP (11,11)
      CALL BOTB (X,NUSEG,TUSEG,XUSEG,ZUSEG,JE,ZU,TU)
      CALL BOTB (X,NLSEG,TLSEG,XLSEG,ZLSEG,JE,ZL,TL)
C
C          NORMALIZATION
C          ---------------------------------------------------------
C          Note that TU(j) & TL(j) are no longer "tangent" of slopes
C          ---------------------------------------------------------
      DUM  = TP*DSQRT(GRAV*HREFP)
      X    = X/DUM
      DIST = -X
      DO 100 J = 1,JE
```

```
        DIST   = DIST + X
        XUL(J) = DIST
        ZU(J)  = ZU(J)/HREFP
        ZL(J)  = ZL(J)/HREFP
        TU(J)  = TU(J)*SIGMA
        TL(J)  = TL(J)*SIGMA
  100 CONTINUE
C
C          FIND INITW = number of spatial nodes at which the lower
C                       slope is located below SWL
C          --------------------------------------------------------
      J = 0
  920 CONTINUE
        IF (J.EQ.JE) CALL STOPP (12,12)
        J = J+1
        CROSS = ZL(J)*ZL(J+1)
      IF (CROSS.GT.0.D+00) GOTO 920
      INITW = J
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 07 BOTTOM ..................
C >>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 08 BOTA  <<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 08 BOTA aids Subroutine 07 BOTTOM in completing
C     physical segment data not specified as input
C
      SUBROUTINE BOTA (IBOT,DSEAW,NSEG,WSEG,TSEG,XSEG,ZSEG)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION WSEG(NSEG),TSEG(NSEG),XSEG(NSEG+1),ZSEG(NSEG+1)
      IF (IBOT.EQ.1) THEN
        DCUM    = 0.D+00
        XSEG(1) = 0.D+00
        ZSEG(1) = -DSEAW
        DO 110 K = 2,NSEG+1
          DCUM    = DCUM + WSEG(K-1)*TSEG(K-1)
          XSEG(K) = XSEG(K-1) + WSEG(K-1)
          ZSEG(K) = -DSEAW + DCUM
  110   CONTINUE
      ELSE
        DO 120 K = 1,NSEG
          TSEG(K) = (ZSEG(K+1)-ZSEG(K))/(XSEG(K+1)-XSEG(K))
  120   CONTINUE
      ENDIF
C
      RETURN
      END
C
C ................... END OF SUBROUTINE 08 BOTA ....................
C >>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 09 BOTB  <<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 09 BOTB aids Subroutine 07 BOTTOM in calculating
C     physical structure geometry at each node
```

```
C
      SUBROUTINE BOTB (X,NSEG,TSEG,XSEG,ZSEG,JE,ZNODE,TNODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION TSEG(NSEG),XSEG(NSEG+1),ZSEG(NSEG+1)
      DIMENSION ZNODE(JE),TNODE(JE)
C
      DIST = -X
      K    = 1
      XCUM = XSEG(K+1)
      DO 100 J = 1,JE
        DIST = DIST + X
        IF (DIST.GT.XCUM.AND.K.LT.NSEG) THEN
  900     CONTINUE
            K    = K+1
            XCUM = XSEG(K+1)
          IF (DIST.GT.XCUM.AND.K.LT.NSEG) GOTO 900
        ENDIF
        ZNODE(J) = ZSEG(K) + (DIST-XSEG(K))*TSEG(K)
        TNODE(J) = TSEG(K)
  100 CONTINUE
C
      RETURN
      END
C
C .................... END OF SUBROUTINE 09 BOTB ....................
C >>>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 10 PARAM  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 10 PARAM calculates parameters
C
      SUBROUTINE PARAM
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CONSTA/ PI,GRAV
      COMMON /OPTION/ ISYST,IBOT,ISALT
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  WLO,WL,UR,SURF,SIGMA
      COMMON /BOT1/   DSEAP,FWP
      COMMON /BOT2/   DSEA,DSEA2,FW,TSLOPS
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPAO,BETAO,PORO,PARPU,PARPQ,PARMU
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /DWLINE/ DELRP(MXRUN),DELR(MXRUN),RU(2,MXRUN)
      COMMON /ARMOR1/ SG,TANPHI,C2,C3,CD,CL,CM,AMAX,AMIN
      COMMON /ARMOR2/ CSTAB1,CSTAB2,AMAXS,AMINS,E2,
     +                SSLOPE(MXNOD),CTAN(MXNOD),E3PRE(MXNOD)
      COMMON /ARMOR3/ SNR(MXNOD),SNSX(MXNOD),TSNSX(MXNOD),
     +                ATMIN(7,MXNOD),SNSC,TSNSC
```

```fortran
C
C          PARAMETERS RELATED TO WAVE AND SLOPE CHARACTERISTICS
C          ----------------------------------------------------------
C          DSEAP = physical water depth below SWL at seaward boundary
C          HREFP = physical reference wave height
C          SURF  = surf similarity parameter
C          WLOP & WLO = physical & normalized deep-water wavelengths
C          FWP  & FW  = physical & normalized slope friction factors
C          WL = normalized linear wavelength at seaward boundary
C          UR = Ursell number at seaward boundary based on linear
C               wavelength
C          ----------------------------------------------------------
      DSEA  = DSEAP/HREFP
      DSEA2 = DSQRT(DSEA)
      SURF  = SIGMA*TSLOPS/DSQRT(2.D+00*PI)
      WLOP  = GRAV*(TP*TP)/(2.D+00*PI)
      WLO   = WLOP/DSEAP
      FW    = .5D+00*FWP*SIGMA
      TWOPI = 2.D+00*PI
      WL    = WLO
      FUN1  = WL - WLO*DTANH(TWOPI/WL)
  900 IF (DABS(FUN1).GT.1.D-04) THEN
        FUN2 = 1.D+00 + WLO*TWOPI/(WL*DCOSH(TWOPI/WL))**2
        WL   = WL - FUN1/FUN2
        FUN1 = WL - WLO*DTANH(TWOPI/WL)
      GOTO 900
      ENDIF
      UR = WL*WL/DSEA
C
C          PARAMETERS RELATED TO PERMEABILITY
C          ----------------------------------------------------------
C          ALPAP = coefficient [alpha prime] expressing the laminar
C                  flow resistance
C          BETAP = coefficient [beta prime] expressing the turbulent
C                  flow resistance
C          ALPA0,BETA0 = dimensionless constants related to ALPAP
C                  and BETAP, respectively
C          PORO  = porosity of the permeable underlayer
C          VISCO = kinematic viscosity of water
C          DPP   = physical representative diameter of permeable
C                  underlayer materials
C          PARPU,PARPQ,PARMU = dimensionless parameters that arise in
C                  consequence of normalizing the governing equations
C          ----------------------------------------------------------
      PORO1 = 1.D+00-PORO
      ALPAP = ALPA0*PORO1**3*VISCO/(DPP**2*PORO**2)
      BETAP = BETA0*PORO1/(DPP*PORO**3)
      UPS   = PORO*DPP
      DOWNS = BETA0*PORO1*TP*DSQRT(GRAV*HREFP)
      PARPU = DSQRT(UPS/DOWNS)
      PARPQ = PORO*PARPU
      UPS   = ALPA0*PORO1**2*VISCO
      DOWNS = BETA0*PARPU*DPP*DSQRT(GRAV*HREFP)
      PARMU = UPS/DOWNS
C
```

```
C           COMPUTATION PARAMETERS
C           --------------------------------------------------------
C        :: Related to the user-defined waterlines:
C           . DELRP & DELR = physical & normalized water depths
C             associated with the user-defined waterlines
C        :: Related to time-stepping:
C           . INVT = number of time steps in one reference wave period
C           . INVT must + satisfy a numerical stability criterion
C                       + be divisible by NRATE for storing purposes
C           . MINVT = smallest value satisfying the above criteria
C                   = minimum allowable value of INVT
C           --------------------------------------------------------
      DO 110 L = 1,NDELR
        IF (ISYST.EQ.1) THEN
          DELR(L) = DELRP(L)/(1.D+03*HREFP)
        ELSE
          DELR(L) = DELRP(L)/(12.D+00*HREFP)
        ENDIF
  110 CONTINUE
      EPSI   = DMAX1(EPSI1,EPSI2)
      TERMA  = 1.D+00 + EPSI*EPSI/4.D+00
      TERMB  = DSQRT(TERMA) - EPSI/2.D+00
      TERMC  = (1.D+00+DSEA2)/(TERMB*X)
      NTERMA = INT(TERMC) + 1
      NTERMB = NTERMA/NRATE + 1
      MINVT  = NTERMB*NRATE
C
C           COMPUTATION PARAMETERS
C           --------------------------------------------------------
C           TSLOPE,CSLOPE,SSLOPE = tangent,cosine,sine of local slope
C           TANPHI = armor friction factor
C           --------------------------------------------------------
      SG1 = SG-1.D+00
      CSTAB1 = 2.D+00*C3**(2.D+00/3.D+00)/(C2*CD)
      CSTAB2 = CM/(SG1*SIGMA)
      E2     = CL*TANPHI/CD
      AMAXS  = AMAX*SIGMA
      AMINS  = AMIN*SIGMA
      DO 210 J = 1,JE
        TSLOPE    = TU(J)/SIGMA
        ANGLE     = DATAN(TSLOPE)
        CSLOPE    = DCOS(ANGLE)
        SSLOPE(J) = DSIN(ANGLE)
        CTAN(J)   = CSLOPE*TANPHI
        E3PRE(J)  = CSTAB1*CTAN(J)
  210 CONTINUE
C
      RETURN
      END
C
C .................... END OF SUBROUTINE 10 PARAM ...................
C >>>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 11 INIT <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 11 INIT specifies initial condition at t=0
C
```

```
      SUBROUTINE INIT
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /ETASEA/ ETAI,ETAR,ESEA(6)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /STATV/  VMEAN(12,MXNOD),VMAX(6,MXNOD),VMIN(6,MXNOD)
      COMMON /STATR/  RMEAN(4,MXRUN),RMAX(2,MXRUN),RMIN(2,MXRUN)
      COMMON /ENERG/  EEXTMN(4,MXNOD),EEXT(5,MXNOD),
     2                EINTMN(4,MXNOD),EINT(6,MXNOD)
      COMMON /MASSB/  VMASS(4,MXNOD)
      COMMON /ARMOR3/ SNR(MXNOD),SNSX(MXNOD),TSNSX(MXNOD),
     +                ATMIN(7,MXNOD),SNSC,TSNSC
      COMMON /ARMOR4/ JSTAB,JSNSC,JSTABM,JSATM,ISATM,IWATM
C
C          HYDRODYNAMIC VARIABLES
C          At node j:
C          -----------------------------------------------------
C       :: EXTERNAL FLOW:
C          UONE(1,j) = volume flux
C          UONE(2,j) = total water depth
C          U(j)      = depth-averaged horizontal velocity
C          UB(j)     = horizontal velocity between the two slopes
C          ELEV(j) = free surface elevation above SWL
C       :: INTERNAL FLOW:
C          HP(j)     = vertical thickness of underlayer flow
C                    = UTRE(2,j)
C          UP(j)     = vertically-averaged horizontal discharge
C                      velocity in the permeable underlayer
C          ELEV(j) = internal water table elevation above SWL
C          AMP(J)  = PARPU*HP(j)*UP(j)
C          PARPU = a dimensionless parameter related to permeability,
C                      defined in (10) PARAM
C       :: INTERACTION:
C          QB(j)     = volume influx per unit horizontal area into the
C                      permeable underlayer
C          -----------------------------------------------------
      DO 110 J = 1,JE
        UONE(1,J) = 0.D+00
        U(J)    = 0.D+00
        UB(J)   = U(J)
        UP(J)   = 0.D+00
        QB(J)   = 0.D+00
        AMP(J) = 0.D+00
```

```
          IF (J.LE.INITW) THEN
            ELEV(J) = 0.D+00
          ELSE
            ELEV(J) = ZL(J)
          ENDIF
          IF (J.LE.INITS) THEN
            UONE(2,J) = -ZU(J)
          ELSE
            UONE(2,J) = 0.D+00
          ENDIF
          IF (J.LE.INITS) THEN
            HP(J) = ZU(J)-ZL(J)
          ELSEIF (J.LE.INITW) THEN
            HP(J) = -ZL(J)
          ELSE
            HP(J) = 0.D+00
          ENDIF
          UTRE(1,J) = AMP(J)
          UTRE(2,J) = HP(J)
  110 CONTINUE
C
C           WATERLINE NODES
C           --------------------------------------------------------------
      ISOLD  = INITS
      IS     = INITS
      IW     = INITW
      ISMIN  = INITS
      IWMIN  = INITW
      ISMAX  = INITS
      IWMAX  = INITW
      JSTAB  = INITS
      JSTABM = INITS
C
C           SAVING VARIABLES
C           --------------------------------------------------------------
      CALL SAVEM
      CALL SAVE5 (0,1,0)
      CALL SAVE5 (1,0,0)
C
C           STATISTICAL QUANTITIES
C           See (47) STAT for a description of statistical quantities
C           --------------------------------------------------------------
      DO 140 J = 1,JE
        DO 120 K = 1,6
          VMEAN(K,J) = 0.D+00
  120   CONTINUE
        DO 130 K = 1,6
          VMAX(K,J) = -1.D+06
          VMIN(K,J) =  1.D+06
  130   CONTINUE
  140 CONTINUE
      DO 160 K = 1,2
        DO 150 L= 1,NDELR
          RMEAN(K,L) =  0.D+00
          RMAX(K,L)  = -1.D+06
```

```
        RMIN(K,L)  =  1.D+06
  150   CONTINUE
        ESEA(K) = 0.D+00
  160 CONTINUE
      ESEA(5) = -1.D+06
      ESEA(6) =  1.D+06
C
C           ENERGY QUANTITIES
C           See (48) ENERGY for a description of energy quantities
C           ------------------------------------------------------
      DO 200 J = 1,JE
        DO 170 K = 1,4
          EEXTMN(K,J) = 0.D+00
          EINTMN(K,J) = 0.D+00
  170   CONTINUE
        DO 180 K = 1,5
          EEXT(K,J) = 0.D+00
  180   CONTINUE
        DO 190 K = 1,6
          EINT(K,J) = 0.D+00
  190   CONTINUE
  200 CONTINUE
C
C           MASS BALANCE
C           See (49) SAVET for a description of quantities needed to
C           check mass balance
C           ------------------------------------------------------
      DO 220 J = 1,JE
        DO 210 K = 1,4
          VMASS(K,J) = 0.D+00
  210   CONTINUE
  220 CONTINUE
C
C           ARMOR STABILITY
C           See (50) STABNO for a description of quantities related to
C           armor stability
C           ------------------------------------------------------
      SNSC  = 1.D+03
      TSNSC = 0.D+00
      DO 240 J = 1,JE
        SNR(J)   = 0.D+00
        SNSX(J)  = 1.D+03
        TSNSX(J) = 0.D+00
        DO 230 K = 1,7
          ATMIN(K,J) = 0.D+00
  230   CONTINUE
  240 CONTINUE
C
      RETURN
      END
C
C .................... END OF SUBROUTINE 11 INIT ....................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 12 INITSQ  <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 12 INITSQ initializes values at the beginning of an
```

```
C     ICOMP=2 computation
C
      SUBROUTINE INITSQ (ISEQ,MBEGIN)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /SEQI/   MSEQ(0:40),MULTIQ(40)
C
C           Read information written by the preceding sequence
C           using (17) READER
C           -----------------------------------------------------
C           Note: MBEGIN=1 unless ICOMP=2
      CALL READER (ISEQ-1,MRD,MURD,DSSV,DWSV,ESV1,ESV2)
      MBEGIN = MRD+1
C
C           Make adjustments on DELTAS and DELTAW if applicable
C           -----------------------------------------------------
      IF (UONE(2,IS).LE.DELTAS) THEN
        DUM1   = .95*UONE(2,IS)
        DUM2   = DUM1/.000001D+00
        IDUM   = INT(DUM2)+1
        DVAL   = DBLE(IDUM)*.000001D+00
        DELTAS = DMAX1(DVAL,DSSV)
      ENDIF
      IF (UTRE(2,IW).LE.DELTAW) THEN
        DUM1   = .95*UTRE(2,IW)
        DUM2   = DUM1/.000001D+00
        IDUM   = INT(DUM2)+1
        DVAL   = DBLE(IDUM)*.000001D+00
        DELTAW = DMAX1(DVAL,DWSV)
      ENDIF
      WRITE  (*,9910) ISEQ,MBEGIN,MURD,DSSV,DWSV,ESV1,ESV2
      WRITE (99,9910) ISEQ,MBEGIN,MURD,DSSV,DWSV,ESV1,ESV2
      WRITE  (*,9920) MULTIF,DELTAS,DELTAW,EPSI1,EPSI2
      WRITE (99,9920) MULTIF,DELTAS,DELTAW,EPSI1,EPSI2
C
C           Transfer information:
C           read from Unit 47, transfer to Units 48 and 49.
C           Example: For sequence number 4
C            . Unit 48 = file 'CINPUT04' created for this sequence 4
C            . Unit 47 = file 'CINPUT03' prepared by sequence 3
C            . Unit 49 = file 'TRANSFER' (independent
C                                          of sequence number)
C           Since the purpose of this block is only to transfer
C           information from one file to two other files, this does
C           not affect the computation of this sequence.
C           For simplicity data points are read and written using
```

```
C           dummy variables.
C           ----------------------------------------------------------
      DO 120 M = 1,MRD
        READ  (47,4910) IAA,IBB,ICC,DAA,DBB
        WRITE (48,4910) IAA,IBB,ICC,DAA,DBB
        WRITE (49,4910) IAA,IBB,ICC,DAA,DBB
        DO 110 I = 1,ICC
          READ  (47,4920) IDD,IEE,DCC,DDD
          WRITE (48,4920) IDD,IEE,DCC,DDD
          WRITE (49,4920) IDD,IEE,DCC,DDD
  110   CONTINUE
  120 CONTINUE
      CLOSE (47)
C
C          SAVING VARIABLES
C          -----------------------------------------------------------
      CALL SAVEM
      CALL SAVE5 (0,1,MRD)
      CALL SAVE5 (1,0,MRD)
C
C           Formats
C           ----------------------------------------------------------
 4910 FORMAT (3I8,2D18.9)
 4920 FORMAT (2I8,2D18.9)
 9000 FORMAT (8I8)
 9910 FORMAT (/':: Initializing sequence       ISEQ =',I8/
     +         '          at beginning of unit MBEGIN =',I8/
     +         '  Instrumental computation parameters LAST USED:'/
     +         '        MULTIF    DELTAS    DELTAW    EPSI1    EPSI2'/
     +         3x,I10,2F10.6,2F9.2)
 9920 FORMAT ( '  Instrumental computation parameters NOW IN USE:'/
     +         '        MULTIF    DELTAS    DELTAW    EPSI1    EPSI2'/
     +         3x,I10,2F10.6,2F9.2)
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 12 INITSQ ..................
C >>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 13 INITM  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 13 INITM initializes values at the beginning of a
C     computation unit
C
      SUBROUTINE INITM
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
```

```
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /DIAGV1/ ISOSAV(3),ISSAV(3),IWSAV(3),M5SAV(2)
      COMMON /DIAGV2/ DSSAV(3),DWSAV(3),U1SAV(6,MXNOD),U3SAV(6,MXNOD),
     2                USAV(3,MXNOD),ESAV(3,MXNOD),
     3                HPSAV(3,MXNOD),UPSAV(3,MXNOD),AMPSAV(3,MXNOD)
      ISOLD  = ISOSAV(1)
      IS     = ISSAV(1)
      IW     = IWSAV(1)
      DELTAS = DSSAV(1)
      DELTAW = DWSAV(1)
      DO 110 J = 1,JE
        UONE(1,J) = U1SAV(1,J)
        UONE(2,J) = U1SAV(2,J)
        UTRE(1,J) = U3SAV(1,J)
        UTRE(2,J) = U3SAV(2,J)
        U(J)      = USAV(1,J)
        ELEV(J)   = ESAV(1,J)
        HP(J)     = HPSAV(1,J)
        UP(J)     = UPSAV(1,J)
        AMP(J)    = AMPSAV(1,J)
  110 CONTINUE
      RETURN
      END
C
C .................. END OF SUBROUTINE 13 INITM ..................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 14 SAVEM  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 14 SAVEM saves values at the end of each successful
C     computation unit
C
      SUBROUTINE SAVEM
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /DIAGV1/ ISOSAV(3),ISSAV(3),IWSAV(3),M5SAV(2)
      COMMON /DIAGV2/ DSSAV(3),DWSAV(3),U1SAV(6,MXNOD),U3SAV(6,MXNOD),
     2                USAV(3,MXNOD),ESAV(3,MXNOD),
     3                HPSAV(3,MXNOD),UPSAV(3,MXNOD),AMPSAV(3,MXNOD)
      ISOSAV(1) = ISOLD
      ISSAV(1)  = IS
      IWSAV(1)  = IW
      DSSAV(1)  = DELTAS
      DWSAV(1)  = DELTAW
      DO 110 J = 1,JE
```

```
        U1SAV(1,J)  = UONE(1,J)
        U1SAV(2,J)  = UONE(2,J)
        U3SAV(1,J)  = UTRE(1,J)
        U3SAV(2,J)  = UTRE(2,J)
        USAV(1,J)   = U(J)
        ESAV(1,J)   = ELEV(J)
        HPSAV(1,J)  = HP(J)
        UPSAV(1,J)  = UP(J)
        AMPSAV(1,J) = AMP(J)
  110 CONTINUE
      RETURN
      END
C
C ................... END OF SUBROUTINE 14 SAVEM ..................
C >>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 15 SAVE5  <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 15 SAVE5 saves values at the end of each set of five
C     computation units
C
      SUBROUTINE SAVE5 (MODA,MODB,M)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /DIAGV1/ ISOSAV(3),ISSAV(3),IWSAV(3),M5SAV(2)
      COMMON /DIAGV2/ DSSAV(3),DWSAV(3),U1SAV(6,MXNOD),U3SAV(6,MXNOD),
     2                USAV(3,MXNOD),ESAV(3,MXNOD),
     3                HPSAV(3,MXNOD),UPSAV(3,MXNOD),AMPSAV(3,MXNOD)
C
      IF (MODA.EQ.1) THEN
        M5SAV(2)  = M5SAV(1)
        ISOSAV(3) = ISOSAV(2)
        ISSAV(3)  = ISSAV(2)
        IWSAV(3)  = IWSAV(2)
        DSSAV(3)  = DSSAV(2)
        DWSAV(3)  = DWSAV(2)
        DO 110 J = 1,JE
          U1SAV(5,J)  = U1SAV(3,J)
          U1SAV(6,J)  = U1SAV(4,J)
          U3SAV(5,J)  = U3SAV(3,J)
          U3SAV(6,J)  = U3SAV(4,J)
          USAV(3,J)   = USAV(2,J)
          ESAV(3,J)   = ESAV(2,J)
          HPSAV(3,J)  = HPSAV(2,J)
          AMPSAV(3,J) = AMPSAV(2,J)
          UPSAV(3,J)  = UPSAV(2,J)
```

```
  110   CONTINUE
      ENDIF
      IF (MODB.EQ.1) THEN
        M5SAV(1)  = M
        ISOSAV(2) = ISOLD
        ISSAV(2)  = IS
        IWSAV(2)  = IW
        DSSAV(2)  = DELTAS
        DWSAV(2)  = DELTAW
        DO 120 J = 1,JE
          U1SAV(3,J)  = UONE(1,J)
          U1SAV(4,J)  = UONE(2,J)
          U3SAV(3,J)  = UTRE(1,J)
          U3SAV(4,J)  = UTRE(2,J)
          USAV(2,J)   = U(J)
          ESAV(2,J)   = ELEV(J)
          HPSAV(2,J)  = HP(J)
          AMPSAV(2,J) = AMP(J)
          UPSAV(2,J)  = UP(J)
  120   CONTINUE
      ENDIF
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 15 SAVE5 .................
C >>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 16 WRITER  <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 16 WRITER writes information to be passed on to the
C     next sequence of computation
C
      SUBROUTINE WRITER (M,ISEQ)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      CHARACTER*10 FDOC,FMSG,FPUT,FVAL
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /DIAGV1/ ISOSAV(3),ISSAV(3),IWSAV(3),M5SAV(2)
      COMMON /DIAGV2/ DSSAV(3),DWSAV(3),U1SAV(6,MXNOD),U3SAV(6,MXNOD),
     2                USAV(3,MXNOD),ESAV(3,MXNOD),
     3                HPSAV(3,MXNOD),UPSAV(3,MXNOD),AMPSAV(3,MXNOD)
      COMMON /SEQI/   MSEQ(0:40),MULTIQ(40)
      COMMON /SEQD/   EPSIQ(2,40)
      COMMON /SEQC/   FDOC(40),FMSG(40),FPUT(40),FVAL(40)
C
      MFAR = M-MSEQ(ISEQ-1)
      IF (MFAR.GT.10) THEN
        MSEQ(ISEQ)    = M5SAV(2)
        MULTIQ(ISEQ)  = MULTIF
        EPSIQ(1,ISEQ) = EPSI1
        EPSIQ(2,ISEQ) = EPSI2
        CALL WRSEQ (ISEQ,0)
```

```
      ENDIF
C
C     At any one time, the values at the ends of the two nearest sets
C     of five past computation units are saved. The values to be
C     passed to the next sequence of computation are the values at
C     the farthest end.
C     Example: If computation breaks down during computation unit 32,
C              it is the values at the end of computation unit 25
C              which are passed to the next sequence.
C              The next sequence thus begins at computation unit 26.
C
      CALL OPENF (999,46,FVAL(ISEQ),'999')
      WRITE (46,9000) M5SAV(2)
      WRITE (46,9000) MULTIF
      WRITE (46,8800) DSSAV(3),DWSAV(3),EPSI1,EPSI2
      WRITE (46,9000) ISOSAV(3),ISSAV(3),IWSAV(3)
      WRITE (46,9000) JE
      WRITE (46,8800) (U1SAV(5,J),J=1,JE)
      WRITE (46,8800) (U1SAV(6,J),J=1,JE)
      WRITE (46,8800) (U3SAV(5,J),J=1,JE)
      WRITE (46,8800) (U3SAV(6,J),J=1,JE)
      WRITE (46,8800) (USAV(3,J),J=1,JE)
      WRITE (46,8800) (ESAV(3,J),J=1,JE)
      WRITE (46,8800) (HPSAV(3,J),J=1,JE)
      WRITE (46,8800) (UPSAV(3,J),J=1,JE)
      WRITE (46,8800) (AMPSAV(3,J),J=1,JE)
      CLOSE (46)
 8800 FORMAT (4D18.9)
 9000 FORMAT (8I8)
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 16 WRITER  ..................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 17 READER  <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 17 READER reads information passed on by the
C     preceding sequence of computation
C
      SUBROUTINE READER (ISQ,MRD,MURD,DSSV,DWSV,ESV1,ESV2)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      CHARACTER*10 FDOC,FMSG,FPUT,FVAL
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /SEQC/   FDOC(40),FMSG(40),FPUT(40),FVAL(40)
```

```
C
      CALL OPENF (1,46,FVAL(ISQ),'OLD')
      READ (46,9000) MRD
      READ (46,9000) MURD
      READ (46,8800) DSSV,DWSV,ESV1,ESV2
      READ (46,9000) ISOLD,IS,IW
      READ (46,9000) JERD
      READ (46,8800) (UONE(1,J),J=1,JE)
      READ (46,8800) (UONE(2,J),J=1,JE)
      READ (46,8800) (UTRE(1,J),J=1,JE)
      READ (46,8800) (UTRE(2,J),J=1,JE)
      READ (46,8800) (U(J),J=1,JE)
      READ (46,8800) (ELEV(J),J=1,JE)
      READ (46,8800) (HP(J),J=1,JE)
      READ (46,8800) (UP(J),J=1,JE)
      READ (46,8800) (AMP(J),J=1,JE)
      CLOSE (46)
 8800 FORMAT (4D18.9)
 9000 FORMAT (8I8)
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 17 READER ..................
C >>>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 18 WRSEQ  <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 18 WRSEQ writes sequential information into file
C     'SEQUENCE' at the end of an ICOMP=1 or ICOMP=2 computation
C
      SUBROUTINE WRSEQ (ISQ,INDICA)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /SEQI/  MSEQ(0:40),MULTIQ(40)
      COMMON /SEQD/  EPSIQ(2,40)
      CALL OPENF (999,19,'SEQUENCE  ','999')
      WRITE (19,9000) ISQ,INDICA
      WRITE (19,9000) (MSEQ(I),   I=1,ISQ)
      WRITE (19,9000) (MULTIQ(I), I=1,ISQ)
      WRITE (19,8800) (EPSIQ(1,I),I=1,ISQ)
      WRITE (19,8800) (EPSIQ(2,I),I=1,ISQ)
      CLOSE (19)
 8800 FORMAT (4D18.9)
 9000 FORMAT (8I8)
      RETURN
      END
C
C .................. END OF SUBROUTINE 18 WRSEQ ..................
C >>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 19 CINVT  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 19 CINVT assigns value to INVT and determines NEND
C     for each computation unit
C
      SUBROUTINE CINVT (M,ITRY,IPROB,NEND,ICOMP,ISEQ)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
```

```
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
C
C          ASSIGN VALUE TO INVT
C          INVT = number of time steps in one reference wave period
C          --------------------------------------------------------
      IF (ITRY.EQ.1) THEN
        INVT = MULTIF*MINVT
      ELSE
        NDUM1 = INVT/(2*NRATE)
        NDUM2 = NDUM1*NRATE
        INVT  = INVT + NDUM2
        CALL NSI (OMEGA,INVT)
C       ***          ***
C       *** Failure ***
C       ***          ***
        IF (OMEGA.GT.10.OR.IPROB.GE.6) THEN
          IF (ICOMP.LE.2) CALL WRITER (M,ISEQ)
          WRITE  (*,9910) OMEGA,M,ITRY,INVT,DELTAS,DELTAW
          WRITE (99,9910) OMEGA,M,ITRY,INVT,DELTAS,DELTAW
          STOP
        ENDIF
      ENDIF
C
C          DETERMINE NEND
C          NEND = number of time steps for computation unit M
C          --------------------------------------------------------
      IF (M.LT.MWAVE) THEN
        NEND = INVT
      ELSE
        RES = DMOD(TMAX,1.D+00)
        IF (DABS(RES).LT.1.D-08) THEN
          NEND = INVT
        ELSE
          DEND = RES*DBLE(INVT)
          NEND = INT(DEND)
        ENDIF
      ENDIF
C
C          FORMATS
C          --------------------------------------------------------
 9910 FORMAT (/':: Notice of FAILURE from Subr. 19 CINVT ::'/
     +  '    Computation was aborted because OMEGA exceeds 10.'/
     +        '    Numer. stab. indicator   OMEGA =',F11.2/
     +        '    Computation unit             M =',I8/
     +        '    Trial number              ITRY =',I8/
     +        '    Computation parameters    INVT =',I8/
     +        '                            DELTAS =',F18.9/
     +        '                            DELTAW =',F18.9/
     +        '    Programmed Stop.')
C
      RETURN
      END
C
```

```
C .................. END OF SUBROUTINE 19 CINVT ..................
C >>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 20 CINVTD <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 20 CINVTD calculates computation parameters that are
C     dependent on INVT
C
      SUBROUTINE CINVTD (M,ITRY,NEND)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
C        T = constant time step
C        X = constant grid spacing between two adjacent nodes
      T    = 1.D+00/DBLE(INVT)
      TX   = T/X
      XT   = X/T
      TTX  = T*T/X
      TTXX = T*T/(X*X)
      TWOX = 2.D+00*X
      NHOP = INVT/NRATE
      RETURN
      END
C
C ................. END OF SUBROUTINE 20 CINVTD ..................
C >>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 21 MARCH1 <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 21 MARCH1 marches the computation for Region 1 from
C     the present time level to the next. More specifically, this
C     subroutine computes the following values at the next time
C     level:
C     . the upper waterline node IS
C     . the hydrodynamic quantities UONE, U, and ELEV at all nodes
C
C     MODE=0 indicates call from the main program
C     MODE>0 indicates call from (01) MAIN2
C
      SUBROUTINE MARCH1 (MODE,M,ITRY,IPROB,DVALUE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /TEMPO/  TIME
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
C
C     Subroutine   To compute
C     ---------------------------------------------------------------
```

```
C       (24) ROOTH   the square root of the water depth h
C       (25) XTRAP1  the vector UONE and velocity U at node (IS+1)
C       (27) CQBUB   quantities QB and UB
C       (28) MA1     the elements of the first row of Matrix A (2x2)
C       (29) VFONE   the elements of vector FONE (2x1)
C       (30) VGONE   the elements of vector GONE (2x1)
C       (31) VH      the elements of vector H (2x1)
C       (32) VS1     the first element of vector S (2x1)
C       (33) VD      the elements of vector D (2x1)
C       (40) VFTWO   the single element of vector FTWO (2x1)
C       (41) VGTWO   the single element of vector GTWO (2x1)
C       (34) VP      the elements of vector P (2x1)
C       (35) VUONE   the elements of vector UONE (2x1)
C       (36) SEABC   the hydrodynamic quantities at node 1
C       (37) LANBC1  new upper waterline node IS* and the hydrodynamic
C                    quantities at node IS*
C       (39) NUMSTA  checks if numerical stability criterion is violated
C       -------------------------------------------------------------
C                 :: Compute UONE at the next time level for nodes 2
C                    through IS (the value at node IS is tentative)
        CALL ROOTH (M,ITRY,IPROB)
C       ***
C       *** Premature RETURN because IPROB=1
C       ***
        IF (IPROB.NE.0) RETURN
        CALL XTRAP1
        CALL CQBUB
        CALL MA1   (1,IS+1)
        CALL VFONE (1,IS+1)
        CALL VGONE (1,IS+1)
        CALL VH    (1,IS)
        CALL VS1   (2,IS)
        CALL VD    (1,IS-2,IS)
        CALL VFTWO
        CALL VGTWO (0,IS+1)
        CALL VP    (2,IS)
        CALL VUONE (2,IS)
C                 :: Check for off-limit seaward movement of the upper
C                    waterline
        UX    = UONE(2,IS-1)
        RATIO = ABS(UX)/DELTAS
        IF (UX.LE.DELTAS) THEN
          IF (UX.GT.0.D+00.AND.RATIO.GE..5D+00) THEN
            IF (MODE.EQ.33) THEN
              DELTAS = DVALUE
            ELSE
              IPROB  = 33
              DUM1   = .95*UX
              DUM2   = DUM1/.000001D+00
              IDUM   = INT(DUM2)+1
              DVALUE = DELTAS
              DELTAS = DBLE(IDUM)*.000001D+00
              CALL NSI (OMEGA,INVT)
              WRITE  (*,9910) M,IS,TIME,DVALUE,DELTAS,INVT,OMEGA,ITRY
              WRITE (99,9910) M,IS,TIME,DVALUE,DELTAS,INVT,OMEGA,ITRY
```

```
            IF (ITRY.EQ.777) CALL STOPP (6,6)
          ENDIF
        ELSE
          IPROB=3
          CALL NSI (OMEGA,INVT)
          WRITE  (*,9920) M,IS,INVT,DELTAS,UX,TIME,OMEGA,ITRY
          WRITE (99,9920) M,IS,INVT,DELTAS,UX,TIME,OMEGA,ITRY
C         ***
C         *** Premature RETURN because IPROB=3
C         ***
          RETURN
        ENDIF
      ENDIF
C               :: Compute vector UONE and velocity U at node 1
      CALL SEABC
C               :: Compute velocity U at nodes 2 through IS
      DO 110 J = 2,IS
        U(J) = UONE(1,J)/UONE(2,J)
  110 CONTINUE
C               :: Find IS at the next time level and complete
C                  computation of the hydrodynamic quantities
C               :: IS before LANBC1 is called is of the PRESENT time
C                  level, after LANBC1 is called is of the NEXT time
C                  level
      CALL LANBC1
      IF (IS.GE.JE) THEN
        CALL NSI (OMEGA,INVT)
        WRITE  (*,9930) M,IS,JE,INVT,DELTAS,TIME,OMEGA,ITRY
        WRITE (99,9930) M,IS,JE,INVT,DELTAS,TIME,OMEGA,ITRY
C       ***
C       *** Abort computation
C       ***
        STOP
      ENDIF
      DO 120 J = 1,IS
        ELEV(J) = UONE(2,J)+ZU(J)
  120 CONTINUE
      DO 130 J = IS+1,JE
        UONE(1,J) = 0.D+00
        UONE(2,J) = 0.D+00
        U(J)      = 0.D+00
  130 CONTINUE
C               :: Check if the numer. stability criterion is violated
      CALL NUMSTA (M,ITRY,IPROB)
C
 9910 FORMAT (/':: IPROB=33 from Subr. 21 MARCH1 ::'/
     2        '   Computation unit                M =',I8/
     3        '   Upper waterline node           IS =',I8/
     4        '   Time of occurrence           TIME =',F18.9/
     5        '   Change in DELTAS: . Old DELTAS =',F18.9/
     6        '                     . New DELTAS =',F18.9/
     7        '   Computation parameter        INVT =',I8/
     8        '   Indicators                  OMEGA =',F11.2/
     9        '                                ITRY =',I8)
 9920 FORMAT (/':: IPROB=3 from Subr. 21 MARCH1 ::'/
```

```
     2          '  Computation unit                M =',I8/
     3          '  Upper waterline node           IS =',I8/
     4          '  Computation parameters       INVT =',I8/
     5          '                             DELTAS =',F18.9/
     6          '  Water depth at (IS-1)             =',F18.9/
     7          '  Time of occurrence           TIME =',F18.9/
     8          '  Indicators                  OMEGA =',F11.2/
     9          '                               ITRY =',I8)
 9930 FORMAT (/':: IPROB=6 from Subr. 21 MARCH1 ::'/
     2 '  Upper waterline movement reaches the landward end node'/
     3          '  Computation unit                M =',I8/
     4          '  Upper waterline node           IS =',I8/
     5          '  Landward end node              JE =',I8/
     6          '  Computation parameters       INVT =',I8/
     7          '                             DELTAS =',F18.9/
     8          '  Time of occurrence           TIME =',F18.9/
     9          '  Indicators                  OMEGA =',F11.2/
     +          '                               ITRY =',I8/
     1 '>> Possibility Number 1:'/
     2 '  Slope is not long enough to accomodate upper waterline'/
     3 '      movement. Specify longer slope and start computing'/
     4 '      from t=0.'/
     5 '>> Possibility Number 2:'/
     6 '  If slope has been specified long enough, this waterline'/
     7 '      movement may not be realistic. Keep the current'/
     8 '      specification of slope geometry, and repeat this'/
     9 '      sequence using different computation parameters.'/
     + '>> Computation aborted.')
C     ***
C     *** This RETURN may either be
C     *** . normal RETURN, which implies IPROB=0 or IPROB=33, or
C     *** . RETURN with problem IPROB=2 from (39) NUMSTA
C     ***
      RETURN
      END
C
C .................. END OF SUBROUTINE 21 MARCH1 ...................
C >>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 22 MARCH2 <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 22 MARCH2 marches the computation for Region 2 from
C     the present time level to the next. More specifically, this
C     subroutine computes the hydrodynamic quantities AMP and HP at
C     the next time level.
C
      SUBROUTINE MARCH2
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPAO,BETAO,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
```

```
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
      COMMON /VECTWO/ FTWO(MXNOD),GTWO(MXNOD)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
C
C     The first element of vector UTRE is identical to quantity AMP,
C     its second element to quantity HP. Arrangement in this program
C     is made as follows:
C     . In this Subr. 22 MARCH2, quantity AMP for the nodes of
C       Region 2 marches to the next time level, while its PRESENT
C       values are being held by the first element of vector UTRE.
C     . In the subsequent Subr. 23 MARCH3, vector UTRE for the nodes
C       of Region 3 marches to the next time level, while its PRESENT
C       values are being held by quantities AMP and HP.
C
C     Subroutine   To compute
C     -----------------------------------------------------------------
C     (27) CQBUB   quantities QB and UB
C     (40) VFTWO   the single element of vector FTWO (2x1)
C     (41) VGTWO   the single element of vector GTWO (2x1)
C     (42) VUTWO   predictor/corrector values of quantity AMP for
C                  Region 2
C     -----------------------------------------------------------------
C               :: Special treatment for the border node between
C                  Regions 2 & 3
      IF (IS.GT.ISOLD) THEN
        HP(IS)   = ZU(IS)-ZL(IS)
        AMP(IS)  = UTRE(1,IS)
        UP(IS)   = UTRE(1,IS)/(PARPU*HP(IS))
        UTRE(2,IS) = HP(IS)
      ENDIF
C               :: MODE=1: Predictor step
C                      =2: Corrector step
      DO 120 MODE = 1,2
        JEND = IS+1-MODE
        CALL CQBUB
        CALL VFTWO
        CALL VGTWO (MODE,JEND)
        CALL VUTWO (MODE,JEND)
  120 CONTINUE
C               :: Final values of quantity AMP
      AMP(1) = 0.D+00
      DO 130 J = 2,IS-1
        AMP(J)     = (UTRE(1,J)+AMP(J))/2.D+00
        UTRE(1,J) = AMP(J)
        UP(J)      = AMP(J)/(PARPU*HP(J))
  130 CONTINUE
C               :: At this point
C                . AMP(IS) is at PREDICTOR value
C                . UTRE(1,IS) holds the PRESENT value of AMP(IS)
C
      RETURN
      END
```

```
C
C ................. END OF SUBROUTINE 22 MARCH2 ..................
C >>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 23 MARCH3 <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 23 MARCH3 marches the computation for Region 3 from
C     the present time level to the next. More specifically, this
C     subroutine computes the following values at the next time
C     level:
C     . the lower waterline node IW
C     . the hydrodynamic quantities UTRE, UP, and ELEV at all nodes
C
      SUBROUTINE MARCH3 (MODD,M,ITRY,IPROB,DVALUE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      DIMENSION DNEG(MXNOD)
      INTEGER   JNEG(MXNOD)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /TEMPO/  TIME
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
      COMMON /VECTWO/ FTWO(MXNOD),GTWO(MXNOD)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
C
C     The first element of vector UTRE is identical to quantity AMP,
C     its second element to quantity HP. Arrangement in this program
C     is made as follows:
C     . In the previous Subr. 22 MARCH2, quantity AMP for the nodes of
C       Region 2 marches to the next time level, while its PRESENT
C       values are being held by the first element of vector UTRE.
C     . In this Subr. 23 MARCH3, vector UTRE for the nodes of Region 3
C       marches to the next time level, while its PRESENT values are
C       being held by quantities AMP and HP.
C
C     Subroutine  To compute
C     ----------------------------------------------------------------
C     (26) XTRAP3  the vector UTRE and velocity UP at node (IW+1)
C     (43) VFTRE1  the first element of vector F[3] (2x1)
C     (44) VGTRE1  the first element of vector G[3] (2x1)
C     (45) VUTRE   predictor/corrector values of vector UTRE (2x1)
C     (38) LANBC3  new lower waterline node IW* and the hydrodynamic
C                  quantities at node IW*
C     ----------------------------------------------------------------
```

```
C                :: Save the PRESENT value of AMP(IS) as AMPIS
C                 . AMP(IS) itself is at PREDICTOR value at this stage
      AMPIS = UTRE(1,IS)
C                :: MODE=1: Predictor step
C                       =2: Corrector step
      CALL XTRAP3
      DO 110 MODE = 1,2
        CALL VFTRE1
        CALL VGTRE1
        CALL VUTRE (MODE)
  110 CONTINUE
C                :: Compute values of vector UTRE (2x1) at the next
C                   time level before new lower waterline node is
C                   determined
      UTRE(1,IS) = (AMPIS+UTRE(1,IS))/2.D+00
      DO 120 J = IS+1,IW
        UTRE(1,J) = (AMP(J)+UTRE(1,J))/2.D+00
        UTRE(2,J) = (HP(J) +UTRE(2,J))/2.D+00
  120 CONTINUE
      DO 130 J = IS,IW
        UP(J) = UTRE(1,J)/(PARPU*UTRE(2,J))
  130 CONTINUE
C                :: Check for negative water depth hp
      ICOUNT = 0
      DO 140 J = IS+1,IW
        IF (UTRE(2,J).LT.0.D+00) THEN
          ICOUNT = ICOUNT+1
          JNEG(ICOUNT) = J
          DNEG(ICOUNT) = UTRE(2,J)
         ENDIF
  140 CONTINUE
      IF (ICOUNT.GT.0) THEN
        IPROB=4
        CALL NSI (OMEGA,INVT)
        WRITE  (*,9910) M,IW,INVT,TIME,OMEGA,ITRY,ICOUNT
        WRITE (99,9910) M,IW,INVT,TIME,OMEGA,ITRY,ICOUNT
        WRITE (99,9920)
        DO 150 I = 1,ICOUNT
          WRITE (99,9930) JNEG(I),DNEG(I)
  150   CONTINUE
C       ***
C       *** Premature RETURN because IPROB=4
C       ***
        RETURN
      ENDIF
C                :: Check for off-limit seaward movement of the lower
C                   waterline
      UX    = UTRE(2,IW-1)
      RATIO = ABS(UX)/DELTAW
      IF (UX.LE.DELTAW) THEN
        IF (UX.GT.0.D+00.AND.RATIO.GE..5D+00) THEN
          IF (MODD.EQ.55) THEN
            DELTAW = DVALUE
          ELSE
            IPROB  = 55
```

```
            DUM1   = .95*UX
            DUM2   = DUM1/.000001D+00
            IDUM   = INT(DUM2)+1
            DVALUE = DELTAW
            DELTAW = DBLE(IDUM)*.000001D+00
            CALL NSI (OMEGA,INVT)
            WRITE  (*,9940) M,IW,TIME,DVALUE,DELTAW,INVT,OMEGA,ITRY
            WRITE (99,9940) M,IW,TIME,DVALUE,DELTAW,INVT,OMEGA,ITRY
            IF (ITRY.EQ.777) CALL STOPP (6,6)
          ENDIF
        ELSE
          IPROB=5
          CALL NSI (OMEGA,INVT)
          WRITE  (*,9950) M,IW,INVT,DELTAW,UX,TIME,OMEGA,ITRY
          WRITE (99,9950) M,IW,INVT,DELTAW,UX,TIME,OMEGA,ITRY
C         ***
C         *** Premature RETURN because IPROB=5
C         ***
          RETURN
        ENDIF
      ENDIF
C                :: Determine new lower waterline node and complete
C                   computation for Region 3
      CALL LANBC3
      IF (IW.GE.JE) THEN
        CALL NSI (OMEGA,INVT)
        WRITE  (*,9960) M,IW,JE,INVT,DELTAW,TIME,OMEGA,ITRY
        WRITE (99,9960) M,IW,JE,INVT,DELTAW,TIME,OMEGA,ITRY
C       ***
C       *** Abort computation because IPROB=7
C       ***
        STOP
      ENDIF
      DO 160 J = IS,IW
        AMP(J) = UTRE(1,J)
  160 CONTINUE
      DO 170 J = IS+1,IW
        HP(J)   = UTRE(2,J)
        ELEV(J) = UTRE(2,J)+ZL(J)
  170 CONTINUE
      DO 180 J = IW+1,JE
        ELEV(J) = ZL(J)
  180 CONTINUE
C                :: Formats
 9910 FORMAT (/':: IPROB=4 from Subr. 23 MARCH3 ::'/
     +          '  Computation unit              M =',I8/
     +          '  Lower waterline node         IW =',I8/
     +          '  Computation parameter      INVT =',I8/
     +          '  Time of occurrence         TIME =',F18.9/
     +          '  Indicators                OMEGA =',F11.2/
     +          '                     ITRY, ICOUNT =',2I8)
 9920 FORMAT ( '   Node',12x,'hp')
 9930 FORMAT (I8,4x,D15.6)
 9940 FORMAT (/':: IPROB=55 from Subr. 23 MARCH3 ::'/
     +          '  Computation unit              M =',I8/
```

```
     +          '  Lower waterline node          IW =',I8/
     +          '  Time of occurrence          TIME =',F18.9/
     +          '  Change in DELTAW: . Old DELTAW =',F18.9/
     +          '                    . New DELTAW =',F18.9/
     +          '  Computation parameter       INVT =',I8/
     +          '  Indicators                 OMEGA =',F11.2/
     +          '                              ITRY =',I8)
 9950 FORMAT (/':: IPROB=5 from Subr. 23 MARCH3 ::'/
     +          '  Computation unit               M =',I8/
     +          '  Lower waterline node          IW =',I8/
     +          '  Computation parameters      INVT =',I8/
     +          '                            DELTAW =',F18.9/
     +          '  Water depth at (IW-1)            =',F18.9/
     +          '  Time of occurrence          TIME =',F18.9/
     +          '  Indicators                 OMEGA =',F11.2/
     +          '                              ITRY =',I8)
 9960 FORMAT (/':: IPROB=7 from Subr. 23 MARCH3 ::'/
     2 '    Lower waterline movement reaches the landward end node'/
     3          '  Computation unit               M =',I8/
     4          '  Lower waterline node          IW =',I8/
     5          '  Landward end node             JE =',I8/
     6          '  Computation parameters      INVT =',I8/
     7          '                            DELTAW =',F18.9/
     8          '  Time of occurrence          TIME =',F18.9/
     9          '  Indicators                 OMEGA =',F11.2/
     +          '                              ITRY =',I8/
     1 '>> Possibility Number 1:'/
     2 '    Slope is not long enough to accomodate lower waterline'/
     3 '        movement. Specify longer slope and start computing'/
     4 '        from t=0.'/
     5 '>> Possibility Number 2:'/
     6 '    If slope has been specified long enough, this waterline'/
     7 '        movement may not be realistic. Keep the current'/
     8 '        specification of slope geometry, and repeat this'/
     9 '        sequence using different computation parameters.'/
     + '>> Computation aborted.')
C     ***
C     *** Normal RETURN implying IPROB=0 or IPROB=55
C     ***
      RETURN
      END
C
C ................. END OF SUBROUTINE 23 MARCH3 ..................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 24 ROOTH  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 24 ROOTH calculates the square root of the water
C     depth h unless there is negative h (IPROB=1)
C
      SUBROUTINE ROOTH (M,ITRY,IPROB)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      DIMENSION DNEG(MXNOD)
      INTEGER   JNEG(MXNOD)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
```

```
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /TEMPO/  TIME
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      ICOUNT = 0
      DO 110 J = 1,IS
        IF (UONE(2,J).LT.0.D+00) THEN
          ICOUNT = ICOUNT+1
          JNEG(ICOUNT) = J
          DNEG(ICOUNT) = UONE(2,J)
         ENDIF
  110 CONTINUE
      IF (ICOUNT.EQ.0) THEN
        DO 120 J = 1,IS
          C(J) = DSQRT(UONE(2,J))
  120   CONTINUE
      ELSE
        IPROB=1
        CALL NSI (OMEGA,INVT)
        WRITE  (*,9910) M,IS,INVT,TIME,OMEGA,ITRY,ICOUNT
        WRITE (99,9910) M,IS,INVT,TIME,OMEGA,ITRY,ICOUNT
        WRITE (99,9920)
        DO 130 I = 1,ICOUNT
          WRITE (99,9930) JNEG(I),DNEG(I)
  130   CONTINUE
      ENDIF
 9910 FORMAT (/':: IPROB=1 from Subr. 24 ROOTH ::'/
     +         '  Computation unit            M =',I8/
     +         '  Upper waterline node       IS =',I8/
     +         '  Computation parameter    INVT =',I8/
     +         '  Time of occurrence       TIME =',F18.9/
     +         '  Indicators              OMEGA =',F11.2/
     +         '                   ITRY, ICOUNT =',2I8)
 9920 FORMAT ( '    Node',12x,'h')
 9930 FORMAT (I8,4x,D15.6)
      RETURN
      END
C
C .................. END OF SUBROUTINE 24 ROOTH ..................
C >>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 25 XTRAP1  <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 25 XTRAP1 estimates vector UONE, velocities U and UB,
C     and flux QB at node (IS+1) by linear extrapolation
C
      SUBROUTINE XTRAP1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
```

```
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      UONE(2,IS+1) = 2.D+00*UONE(2,IS) - UONE(2,IS-1)
      U(IS+1)      = 2.D+00*U(IS)      - U(IS-1)
      IF (UONE(2,IS+1).LE.0.D+00) THEN
        UONE(2,IS+1) = 0.D+00
        U(IS+1)      = 0.D+00
        QB(IS+1)     = 0.D+00
        UB(IS+1)     = 0.D+00
      ELSE
        QB(IS+1) = (AMP(IS+1)-AMP(IS))/(X*PARPU)
        IF (QB(IS+1).GE.0.D+00) THEN
          UB(IS+1) = 0.D+00
        ELSE
          HPVAL    = ZU(IS+1)-ZL(IS+1)
          UB(IS+1) = AMP(IS+1)/HPVAL
        ENDIF
      ENDIF
      C(IS+1)      = DSQRT(UONE(2,IS+1))
      UONE(1,IS+1) = UONE(2,IS+1)*U(IS+1)
      RETURN
      END
C
C .................. END OF SUBROUTINE 25 XTRAP1 ...................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 26 XTRAP3  <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 26 XTRAP3 estimates vector UTRE and velocity UP at
C     node (IW+1) by linear extrapolation
C
      SUBROUTINE XTRAP3
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      UTRE(2,IW+1) = 2.D+00*UTRE(2,IW) - UTRE(2,IW-1)
      UP(IW+1)     = 2.D+00*UP(IW)     - UP(IW-1)
      IF (UTRE(2,IW+1).LT.0.D+00) THEN
        UTRE(2,IW+1) = 0.D+00
        UP(IW+1)     = 0.D+00
      ENDIF
      UTRE(1,IW+1) = PARPU*UTRE(2,IW+1)*UP(IW+1)
      RETURN
      END
C
C .................. END OF SUBROUTINE 26 XTRAP3 ...................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 27 CQBUB  <<<<<<<<<<<<<<<<<<<<<<<
```

```
C
C     Subroutine 27 CQBUB computes quantities QB and UB
C
      SUBROUTINE CQBUB
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
C
      QB(1)  = 0.D+00
      QB(IS) = (AMP(IS)-AMP(IS-1))/(X*PARPU)
      DO 110 J = 2,IS-1
        QB(J) = (AMP(J+1)-AMP(J-1))/(TWOX*PARPU)
  110 CONTINUE
C
      UB(1) = 0.D+00
      DO 120 J = 2,IS
        IF (QB(J).GE.0.D+00) THEN
          UB(J) = U(J)
        ELSE
          UB(J) = AMP(J)/HP(J)
        ENDIF
  120 CONTINUE
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 27 CQBUB ..................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 28 MA1  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 28 MA1 computes the elements of the first row of
C     Matrix A (2x2)
C
      SUBROUTINE MA1 (JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
      DO 100 J = JBEGIN,JEND
        A1(1,J) = 2.D+00*U(J)
        A1(2,J) = UONE(2,J)-U(J)*U(J)
  100 CONTINUE
      RETURN
      END
```

```fortran
C
C .................... END OF SUBROUTINE 28 MA1 ....................
C >>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 29 VFONE <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 29 VFONE computes the elements of vector FONE (2x1)
C
      SUBROUTINE VFONE (JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
      DO 100 J = JBEGIN,JEND
        FONE(1,J) = UONE(1,J)*U(J) + UONE(2,J)**2/2.D+00
        FONE(2,J) = UONE(1,J)
  100 CONTINUE
      RETURN
      END
C
C .................. END OF SUBROUTINE 29 VFONE ..................
C >>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 30 VGONE <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 30 VGONE computes the elements of vector GONE (2x1)
C
      SUBROUTINE VGONE (JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /BOT2/   DSEA,DSEA2,FW,TSLOPS
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPAO,BETAO,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
      DO 100 J = JBEGIN,JEND
        GONE(1,J)=TU(J)*UONE(2,J)+FW*DABS(U(J))*U(J)+PARPQ*UB(J)*QB(J)
        GONE(2,J)=PARPQ*QB(J)
  100 CONTINUE
      RETURN
      END
C
C .................. END OF SUBROUTINE 30 VGONE ..................
C >>>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 31 VH <<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 31 VH computes the elements of vector H (2x1)
C
      SUBROUTINE VH (JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      DIMENSION FG(2),ALOCAL(2,2)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
      DO 140 J = JBEGIN,JEND
        DO 110 L = 1,2
          FG(L)=FONE(L,J+1)-FONE(L,J)+X*(GONE(L,J+1)+GONE(L,J))/2.D+00
  110   CONTINUE
        ALOCAL(1,1) = (A1(1,J+1)+A1(1,J))/2.D+00
        ALOCAL(1,2) = (A1(2,J+1)+A1(2,J))/2.D+00
        ALOCAL(2,1) = 1.D+00
        ALOCAL(2,2) = 0.D+00
        DO 130 L = 1,2
          H(L,J) = 0.D+00
          DO 120 I = 1,2
            H(L,J) = H(L,J) + ALOCAL(L,I)*FG(I)
  120     CONTINUE
  130   CONTINUE
  140 CONTINUE
      RETURN
      END
C
C ..................... END OF SUBROUTINE 31 VH ....................
C >>>>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 32 VS1 <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 32 VS1 computes the first element of vector S (2x1)
C
      SUBROUTINE VS1 (JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEA2,FW,TSLOPS
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
      DO 100 J = JBEGIN,JEND
        TERMA = (U(J)*U(J)-UONE(2,J))*(UONE(2,J+1)-UONE(2,J-1))/TWOX
        TERMB = U(J)*(UONE(1,J+1)-UONE(1,J-1))/TWOX
        TERMC = TU(J)*UONE(2,J)
        TERMD = FW*DABS(U(J))*U(J)
        TERME = 2.D+00*FW*DABS(U(J))/UONE(2,J)
        ESMALL = TERME*(TERMA-TERMB-TERMC-TERMD)
        S1(J)  = X*ESMALL - TU(J)*(UONE(1,J+1)-UONE(1,J-1))/2.D+00
  100 CONTINUE
      RETURN
      END
C
C .................... END OF SUBROUTINE 32 VS1 ....................
```

```
C >>>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 33 VD  <<<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 33 VD computes the elements of vector D (2x1)
C
      SUBROUTINE VD (JBEGIN,JDAM,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      DIMENSION PHI(0:1),PSI(0:1),UNITM(2,2),ALOCAL(2,2),Q(2,2)
      DIMENSION QU(2,MXNOD)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
      DATA UNITM /1.D+00,0.D+00,0.D+00,1.D+00/
C
      DO 160 J = JBEGIN,JDAM
        DO 110 I = 0,1
          PHI(I) = U(J+I) + C(J+I)
          PSI(I) = U(J+I) - C(J+I)
  110   CONTINUE
        TERMA = DABS(PHI(1)-PHI(0))
        TERMB = DABS(PSI(1)-PSI(0))
        TERMC = DABS(TERMA)*(PSI(1)+PSI(0))
        TERMD = (PHI(1)+PHI(0))*DABS(TERMB)
        TERME = C(J+1)+C(J)
        QSUB1 = -(EPSI1*TERMC-EPSI2*TERMD)/(2.D+00*TERME)
        QSUB2 =  (EPSI1*TERMA-EPSI2*TERMB)/TERME
        ALOCAL(1,1) = (A1(1,J)+A1(1,J+1))/2.D+00
        ALOCAL(1,2) = (A1(2,J)+A1(2,J+1))/2.D+00
        ALOCAL(2,1) = 1.D+00
        ALOCAL(2,2) = 0.D+00
        DO 130 K = 1,2
          DO 120 L = 1,2
            Q(K,L) = QSUB1*UNITM(K,L) + QSUB2*ALOCAL(K,L)
  120     CONTINUE
  130   CONTINUE
        DO 150 K = 1,2
          QU(K,J) = 0.D+00
          DO 140 L = 1,2
            QU(K,J) = QU(K,J) + Q(K,L)*(UONE(L,J+1)-UONE(L,J))
  140     CONTINUE
  150   CONTINUE
  160 CONTINUE
C
      DO 180 J = JBEGIN+1,JDAM
        DO 170 K = 1,2
          D(K,J) = TX*(QU(K,J)-QU(K,J-1))/2.D+00
  170   CONTINUE
  180 CONTINUE
      DO 200 J = JDAM+1,JEND
        DO 190 K = 1,2
          D(K,J) = 0.D+00
```

```
  190   CONTINUE
  200 CONTINUE
C
      RETURN
      END
C
C .................... END OF SUBROUTINE 33 VD ....................
C >>>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 34 VP  <<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 34 VP computes the elements of vector P (2x1)
C
      SUBROUTINE VP (JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      DIMENSION B(-1:0)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEA2,FW,TSLOPS
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
      COMMON /VECTWO/ FTWO(MXNOD),GTWO(MXNOD)
C
      DO 120 J = JBEGIN,JEND
        DO 110 I = -1,0
          TERMA = FTWO(J+I+1)-FTWO(J+I)
          TERMB = GTWO(J+I+1)+GTWO(J+I)
          B(I)  = TERMA/X + TERMB/2.D+00
  110   CONTINUE
        P(2,J) = B(-1) - B(0)
C       ------
        TERMA = AMP(J+1)-AMP(J-1)
        TERMB = FW*DABS(U(J))*(U(J)-UB(J))*TERMA/UONE(2,J)
        TERMC = TU(J)*TERMA/2.D+00
        IF (QB(J).GE.0.D+00) THEN
          TERMD  = -UONE(1,J)*P(2,J)
          TERME  = (FONE(1,J+1)-FONE(1,J-1))/TWOX + GONE(1,J)
          TERMF  = (FONE(2,J+1)-FONE(2,J-1))/TWOX + GONE(2,J)
          TERMG  = TERME - U(J)*TERMF
          TERMH  = TERMD + TERMA*TERMG/2.D+00
          PTILDE = TERMH/UONE(2,J)
        ELSE
          TERMD  = -AMP(J)*P(2,J)
          TERME  = (FTWO(J+1)-FTWO(J-1))/TWOX + GTWO(J)
          TERMF  = TERMD + TERMA*TERME/2.D+00
          PTILDE = TERMF/HP(J)
        ENDIF
        P(1,J) = TERMB - TERMC - PTILDE
C       ------
```

```
  120 CONTINUE
C
      RETURN
      END
C
C .................... END OF SUBROUTINE 34 VP  .....................
C >>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 35 VUONE  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 35 VUONE computes the elements of vector UONE (2x1)
C
      SUBROUTINE VUONE (JBEGIN,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      DIMENSION SLOCAL(2)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
C
      DO 120 J = JBEGIN,JEND
        SLOCAL(1) = S1(J)
        SLOCAL(2) = 0.D+00
        DO 110 K = 1,2
          TERMA = TX*((FONE(K,J+1)-FONE(K,J-1))/2.D+00+X*GONE(K,J))
          TERMB = TTXX*((H(K,J)-H(K,J-1))-X*SLOCAL(K))/2.D+00
          TERMC = TTX*PORO*P(K,J)/2.D+00
          UONE(K,J) = UONE(K,J) - TERMA + TERMB + D(K,J) - TERMC
  110   CONTINUE
  120 CONTINUE
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 35 VUONE  ...................
C >>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 36 SEABC  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 36 SEABC handles the seaward boundary condition
C     for Region 1 at node 1
C
      SUBROUTINE SEABC
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /TEMPO/  TIME
      COMMON /IWT1/   NDATA
      COMMON /IWT2/   ETAMA,ETAMI,ETA(MXTIM)
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
```

```
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /BOT2/   DSEA,DSEA2,FW,TSLOPS
      COMMON /ETASEA/ ETAI,ETAR,ESEA(6)
C
C           ESTIMATE ETAR
C           ------------------------------------------------------
C         . ETAR = surface elevation associated with reflected wave at
C           seaward boundary
C         . BETA1,BETA2 = seaward-advancing characteristics [beta] at
C           nodes 1 and 2, respectively, at the PRESENT time level
C         . BSTAR = seaward-advancing characteristics [beta] at node 1
C           at the NEXT time level
C           ------------------------------------------------------
      BETA1 = -U(1)+2.D+00*C(1)
      BETA2 = -U(2)+2.D+00*C(2)
      TERMP = TX*PORO*(AMP(2)-AMP(1))*(U(1)-UB(1)+C(1))/UONE(2,1)
      BSTAR = BETA1 - TX*(U(1)-C(1))*(BETA2-BETA1) + T*TU(1) - TERMP
      ETAR  = BSTAR*DSEA2/2.D+00 - DSEA
C
C           VALUES AT NODE ONE
C           ------------------------------------------------------
C           Incident wave train is specified by user. The specified
C           train ETA has NDATA points from the normalized time t=0
C           to t=TMAX. Interpolate the specified train to obtain
C           the value at t=TIME.
C           ------------------------------------------------------
      DJJ  = DBLE(NDATA-1)*TIME/TMAX
      JJ   = INT(DJJ)
      ETA1 = ETA(JJ+1)
      ETA2 = ETA(JJ+2)
      DEL  = DJJ - DBLE(JJ)
      ETAI = ETA1 + DEL*(ETA2-ETA1)
      UONE(2,1) = DSEA+ETAI+ETAR
      U(1) = 2.D+00*DSQRT(UONE(2,1))-BSTAR
      UONE(1,1) = UONE(2,1)*U(1)
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 36 SEABC  ..................
C >>>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 37 LANBC1  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 37 LANBC1 handles the landward boundary condition
C     for Region 1
C
      SUBROUTINE LANBC1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
```

```
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
C
C     The following statement has no significance in the computation
C     procedure. It is put here only to avoid annoying compilation
C     warning from "too smart" a compiler.
      ISNEW = IS
C
      IF (UONE(2,IS).GT.UONE(2,IS-1)) THEN
        HHAT = 2.D+00*UONE(2,IS-1) - UONE(2,IS-2)
        UHAT = 2.D+00*U(IS-1)      - U(IS-2)
        IF (HHAT.LE.0.D+00) THEN
          ISNEW = IS-1
          GOTO 900
        ELSE
          IF (HHAT.GT.UONE(2,IS-1)) THEN
            UONE(2,IS) = .9*UONE(2,IS-1)
          ELSE
            UONE(2,IS) = HHAT
          ENDIF
          IF (DABS(UHAT).GT.DABS(U(IS-1))) THEN
            U(IS) = .9*U(IS-1)
          ELSE
            U(IS) = UHAT
          ENDIF
          UONE(1,IS) = UONE(2,IS)*U(IS)
        ENDIF
      ENDIF
C
      IF (UONE(2,IS).LE.DELTAS) THEN
        ISNEW = IS-1
      ELSE
        UONE(2,IS+1) = 2.D+00*UONE(2,IS) - UONE(2,IS-1)
        U(IS+1)      = 2.D+00*U(IS)      - U(IS-1)
        UONE(1,IS+1) = UONE(2,IS+1)*U(IS+1)
        IF (UONE(2,IS+1).LE.DELTAS) THEN
          ISNEW = IS
        ELSE
          ISNEW = IS+1
        ENDIF
      ENDIF
C
  900 CONTINUE
      IF (IABS(ISNEW-IS).GT.1) CALL STOPP (13,13)
      ISOLD = IS
      IS    = ISNEW
C
      RETURN
      END
C
C ................. END OF SUBROUTINE 37 LANBC1 ..................
C >>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 38 LANBC3  <<<<<<<<<<<<<<<<<<<<<<<<
C
```

```
C     Subroutine 38 LANBC3 handles the landward boundary condition
C     for Region 3
C
      SUBROUTINE LANBC3
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
C
C     The following statement has no significance in the computation
C     procedure. It is put here only to avoid annoying compilation
C     warning from "too smart" a compiler.
      IWNEW = IW
C
      IF (UTRE(2,IW).GT.UTRE(2,IW-1)) THEN
        HPHAT = 2.D+00*UTRE(2,IW-1) - UTRE(2,IW-2)
        UPHAT = 2.D+00*UP(IW-1)     - UP(IW-2)
        IF (HPHAT.LE.0.D+00) THEN
          IWNEW = IW-1
          GOTO 900
        ELSE
          IF (HPHAT.GT.UTRE(2,IW-1)) THEN
            UTRE(2,IW) = .9*UTRE(2,IW-1)
          ELSE
            UTRE(2,IW) = HPHAT
          ENDIF
          IF (DABS(UPHAT).GT.DABS(UP(IW-1))) THEN
            UP(IW) = .9*UP(IW-1)
          ELSE
            UP(IW) = UPHAT
          ENDIF
          UTRE(1,IW) = PARPU*UTRE(2,IW)*UP(IW)
        ENDIF
      ENDIF
C
      IF (UTRE(2,IW).LE.DELTAW) THEN
        IWNEW = IW-1
      ELSE
        UTRE(2,IW+1) = 2.D+00*UTRE(2,IW) - UTRE(2,IW-1)
        UP(IW+1)     = 2.D+00*UP(IW)     - UP(IW-1)
        UTRE(1,IW+1) = PARPU*UTRE(2,IW+1)*UP(IW+1)
        IF (UTRE(2,IW+1).LE.DELTAW) THEN
          IWNEW = IW
        ELSE
          IWNEW = IW+1
        ENDIF
      ENDIF
C
  900 CONTINUE
```

```
      IF (IABS(IWNEW-IW).GT.1) CALL STOPP (14,14)
      IW = IWNEW
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 38 LANBC3 ..................
C >>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 39 NUMSTA  <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 39 NUMSTA checks if numerical stability criterion is
C     violated
C
      SUBROUTINE NUMSTA (M,ITRY,IPROB)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      DIMENSION DBIG(MXNOD)
      INTEGER   JBIG(MXNOD)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /TEMPO/  TIME
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
C
      ICOUNT = 0
      DO 110 J = 1,IS
        IF (DABS(U(J)).GT.XT) THEN
          ICOUNT=ICOUNT+1
          JBIG(ICOUNT) = J
          DBIG(ICOUNT) = U(J)
        ENDIF
  110 CONTINUE
      IF (ICOUNT.GT.0) THEN
        IPROB=2
        CALL NSI (OMEGA,INVT)
        WRITE  (*,9910) M,IS,INVT,TIME,XT,OMEGA,ITRY,ICOUNT
        WRITE (99,9910) M,IS,INVT,TIME,XT,OMEGA,ITRY,ICOUNT
        WRITE (99,9920)
        DO 120 I = 1,ICOUNT
          WRITE (99,9930) JBIG(I),DBIG(I)
  120   CONTINUE
      ENDIF
 9910 FORMAT (/':: IPROB=2 from Subr. 39 NUMSTA ::'/
     +        '   Computation unit               M =',I8/
     +        '   Upper waterline node          IS =',I8/
     +        '   Computation parameter       INVT =',I8/
     +        '   Time of occurrence          TIME =',F18.9/
     +        '   Delta x / Delta t             XT =',F18.9/
     +        '   Indicators                 OMEGA =',F11.2/
     +        '                       ITRY, ICOUNT =',2I8)
 9920 FORMAT ( '    Node',12x,'u')
 9930 FORMAT (I8,4x,D15.6)
```

```
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 39 NUMSTA ..................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 40 VFTWO  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 40 VFTWO computes the single element of vector FTWO
C
      SUBROUTINE VFTWO
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /VECTWO/ FTWO(MXNOD),GTWO(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      FTWO(1) = 0.D+00
      DO 100 J = 2,IS
        FTWO(J) = AMP(J)**2/HP(J)
  100 CONTINUE
      HPVAL       = ZU(IS+1)-ZL(IS+1)
      FTWO(IS+1) = AMP(IS+1)**2/HPVAL
      RETURN
      END
C
C .................. END OF SUBROUTINE 40 VFTWO ..................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 41 VGTWO  <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 41 VGTWO computes the single element of vector GTWO
C
      SUBROUTINE VGTWO (MODE,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /VECTWO/ FTWO(MXNOD),GTWO(MXNOD)
      GTWO(1) = 0.D+00
      DO 100 J = 2,JEND
C       :: MODE=1 is for computation of Region 2, predictor step
C             =2 is for computation of Region 2, corrector step
C             =0 is for computation of Region 1 before vector P (2x1)
```

```
C                        is computed
        IF (MODE.EQ.1) THEN
          TERMA = -UB(J)*(AMP(J)-AMP(J-1))/X
          TERMB = HP(J)*(UONE(2,J)-UONE(2,J-1))/X
          HPVAL = HP(J)
        ELSEIF (MODE.EQ.2) THEN
          TERMA = -UB(J)*(AMP(J+1)-AMP(J))/X
          TERMB = HP(J)*(UONE(2,J+1)-UONE(2,J))/X
          HPVAL = HP(J)
        ELSEIF (MODE.EQ.0) THEN
          IF (J.LT.JEND) THEN
            TERMA = -UB(J)*(AMP(J+1)-AMP(J-1))/TWOX
            TERMB = HP(J)*(UONE(2,J+1)-UONE(2,J-1))/TWOX
            HPVAL = HP(J)
          ELSE
            HPVAL = ZU(J)-ZL(J)
            TERMA = -UB(J)*(AMP(J)-AMP(J-1))/X
            TERMB = HPVAL*(UONE(2,J)-UONE(2,J-1))/X
          ENDIF
        ELSE
          WRITE (*,*) ' Error in Subr. 41 VGTWO.'
          STOP
        ENDIF
        TERMC = TU(J)*HPVAL
        TERMD = AMP(J)/PARPU
        TERME = PARMU + DABS(AMP(J))/(PARPU*HPVAL)
        GTWO(J) = TERMA + TERMB + TERMC + TERMD*TERME
  100 CONTINUE
      RETURN
      END
C
C .................. END OF SUBROUTINE 41 VGTWO ..................
C >>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 42 VUTWO <<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 42 VUTWO computes quantity AMP for Region 2
C
      SUBROUTINE VUTWO (MODE,JEND)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /VECTWO/ FTWO(MXNOD),GTWO(MXNOD)
      IF (MODE.EQ.1) THEN
        DO 110 J = 2,JEND
          AMP(J) = AMP(J) - TX*(FTWO(J)-FTWO(J-1)) - T*GTWO(J)
  110   CONTINUE
      ELSE
        DO 120 J = 2,JEND
          AMP(J) = AMP(J) - TX*(FTWO(J+1)-FTWO(J)) - T*GTWO(J)
  120   CONTINUE
      ENDIF
      RETURN
      END
```

```
C
C .................. END OF SUBROUTINE 42 VUTWO ..................
C >>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 43 VFTRE1 <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 43 VFTRE1 computes the first element of vector F[3]
C     (2x1).
C     Note that the second element of vector F[3] is identical to the
C     first element of vector UTRE. Only the latter is represented in
C     this program in order to minimize memory requirements.
C
      SUBROUTINE VFTRE1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      DO 100 J = IS,IW+1
        IF (DABS(UTRE(2,J)).GT.1.D-06) THEN
          FTRE1(J) = UTRE(1,J)**2/UTRE(2,J) + UTRE(2,J)**2/2.D+00
        ELSE
          FTRE1(J) = 0.D+00
        ENDIF
  100 CONTINUE
      RETURN
      END
C
C .................. END OF SUBROUTINE 43 VFTRE1 ..................
C >>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 44 VGTRE1 <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 44 VGTRE1 computes the first element of vector G[3]
C     (2x1).
C     Note that the second element of vector G[3] is zero.
C
      SUBROUTINE VGTRE1
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      DO 100 J = IS,IW+1
        TERMA = TL(J)*UTRE(2,J)
        TERMB = PARMU*UTRE(1,J)/PARPU
        IF (DABS(UTRE(2,J)).GT.1.D-06) THEN
          TERMC = DABS(UTRE(1,J))*UTRE(1,J)/(PARPU**2*UTRE(2,J))
        ELSE
          TERMC = 0.D+00
        ENDIF
        GTRE1(J) = TERMA + TERMB + TERMC
  100 CONTINUE
      RETURN
      END
```

```
C
C .................. END OF SUBROUTINE 44 VGTRE1 ...................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 45 VUTRE  <<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 45 VUTRE computes the elements of vector UTRE (2x1)
C
      SUBROUTINE VUTRE (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
C
C     Note that the second element of vector F[3] is identical to the
C     first element of vector UTRE. Only the latter is represented in
C     this program in order to minimize memory requirements.
C
      IF (MODE.EQ.1) THEN
        DO 110 J = IS+1,IW+1
          UTRE(2,J) = UTRE(2,J)-TX*(UTRE(1,J)-UTRE(1,J-1))
          UTRE(1,J) = UTRE(1,J)-TX*(FTRE1(J)-FTRE1(J-1))-T*GTRE1(J)
  110   CONTINUE
C       :: The PREDICTOR value of AMP(IS) computed from Region 2
C          is adopted as that of UTRE(1,IS) for Region 3
        UTRE(1,IS) = AMP(IS)
      ELSE
C       :: The value of UTRE(2,IS) does not change
        DO 120 J = IS+1,IW
          UTRE(2,J) = UTRE(2,J)-TX*(UTRE(1,J+1)-UTRE(1,J))
  120   CONTINUE
        DO 130 J = IS,IW
          UTRE(1,J) = UTRE(1,J)-TX*(FTRE1(J+1)-FTRE1(J))-T*GTRE1(J)
  130   CONTINUE
      ENDIF
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 45 VUTRE ...................
C >>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 46 RUNUP  <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 46 RUNUP computes run-up heights on the upper and
C     lower slope
C
      SUBROUTINE RUNUP
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
```

```
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /DWLINE/ DELRP(MXRUN),DELR(MXRUN),RU(2,MXRUN)
      COMMON /ARMOR4/ JSTAB,JSNSC,JSTABM,JSATM,ISATM,IWATM
C
C            DELR(i) = i-th user-defined water depth for computation
C                      run-up heights
C            RU(1,i) = free surface elevation where the water depth
C                      above the upper slope equals DELR(i)
C            RU(2,i) = internal water table elevation where the water
C                      depth above the lower slope equals DELR(i)
C            NDELR   = number of DELRs
C
C            Run-up on the UPPER slope
C            ------------------------------------------------------
      DO 110 L = 1,NDELR
        INDIC = 0
        K = -1
  910   CONTINUE
          K = K + 1
          IF (UONE(2,IS-K).GE.DELR(L)) THEN
            INDIC = 1
            NRUN1 = IS-K
            NRUN2 = IS-K+1
            DEL1  = UONE(2,IS-K)
            DEL2  = UONE(2,IS-K+1)
            RUN   = (ZU(NRUN2)-ZU(NRUN1))*(DEL1-DELR(L))
            RUN   = RUN/(DEL1-DEL2)
            RUN   = RUN + ZU(NRUN1)
            RU(1,L) = RUN + DELR(L)
            IF (L.EQ.1) JSTAB=NRUN1
          ENDIF
        IF (INDIC.EQ.0) GOTO 910
  110 CONTINUE
C
C            Run-up on the LOWER slope
C            ------------------------------------------------------
      DO 120 L = 1,NDELR
        INDIC = 0
        K = -1
  920   CONTINUE
          K = K + 1
          IF (UTRE(2,IW-K).GE.DELR(L)) THEN
            INDIC = 1
            NRUN1 = IW-K
            NRUN2 = IW-K+1
            DEL1  = UTRE(2,IW-K)
            DEL2  = UTRE(2,IW-K+1)
            RUN   = (ZL(NRUN2)-ZL(NRUN1))*(DEL1-DELR(L))
            RUN   = RUN/(DEL1-DEL2)
            RUN   = RUN + ZL(NRUN1)
            RU(2,L) = RUN + DELR(L)
          ENDIF
        IF (INDIC.EQ.0) GOTO 920
  120 CONTINUE
```

```
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 46 RUNUP ..................
C >>>>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 47 STAT <<<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 47 STAT computes statistical values: mean, maximum,
C     and minimum with respect to the time interval
C     from t = (MSTAT + delta t)
C       to t = TMAX
C     This subroutine also performs part of the calculation of energy
C     quantities
C
      SUBROUTINE STAT (MODE)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /ETASEA/ ETAI,ETAR,ESEA(6)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /DWLINE/ DELRP(MXRUN),DELR(MXRUN),RU(2,MXRUN)
      COMMON /STATV/  VMEAN(12,MXNOD),VMAX(6,MXNOD),VMIN(6,MXNOD)
      COMMON /STATR/  RMEAN(4,MXRUN),RMAX(2,MXRUN),RMIN(2,MXRUN)
      COMMON /ENERG/  EEXTMN(4,MXNOD),EEXT(5,MXNOD),
     2                EINTMN(4,MXNOD),EINT(6,MXNOD)
      COMMON /MASSB/  VMASS(4,MXNOD)
      COMMON /ARMOR4/ JSTAB,JSNSC,JSTABM,JSATM,ISATM,IWATM
C
C     Spatially varying quantities at node j:
C       VMEAN(1,j) = mean value of UONE(1,j)
C       VMEAN(2,j) = mean value of U(j)
C       VMEAN(3,j) = mean value of QB(j)
C       VMEAN(4,j) = mean value of ELEV(j)
C       VMEAN(5,j) = mean value of UP(j)
C       VMEAN(6,j) = mean value of AMP(j)
C       VMEAN(6+k,j) = dummy for VMEAN(k,j) where k=1,2,...,6
C       VMAX(1,j) = maximum value of UONE(1,j)
C       VMAX(2,j) = maximum value of U(j)
C       VMAX(3,j) = maximum value of QB(j)
C       VMAX(4,j) = maximum value of ELEV(j)
C       VMAX(5,j) = maximum value of UP(j)
C       VMAX(6,j) = maximum value of AMP(j)
C       VMIN(1,j) = minimum value of UONE(1,j)
C       VMIN(2,j) = minimum value of U(j)
C       VMIN(3,j) = minimum value of QB(j)
C       VMIN(4,j) = minimum value of ELEV(j)
C       VMIN(5,j) = minimum value of UP(j)
```

```
C         VMIN(6,j) = minimum value of AMP(j)
C       Run-up heights (k=1,2,...,NDELR in the following):
C         RMEAN(1,k) = mean value of RU(1,k)
C         RMEAN(2,k) = mean value of RU(2,k)
C         RMEAN(3,k) = dummy for RMEAN(1,k)
C         RMEAN(4,k) = dummy for RMEAN(2,k)
C         RMAX(1,k)  = maximum value of RU(1,k)
C         RMAX(2,k)  = maximum value of RU(2,k)
C         RMIN(1,k)  = minimum value of RU(1,k)
C         RMIN(2,k)  = minimum value of RU(2,k)
C       Elevations at the seaward boundary:
C         ESEA(1) = mean value of ETAI
C         ESEA(2) = mean value of ETAR
C         ESEA(3) = dummy for ESEA(1)
C         ESEA(4) = dummy for ESEA(2)
C         ESEA(5) = maximum value of ETAR
C         ESEA(6) = minimum value of ETAR
C       See (48) ENERGY for a description of energy quantities
C       See (49) SAVET  for a description of quantities needed to check
C                       mass balance
C
        IF (MODE.EQ.1) THEN
C         ::  MODE=1 is to initialize dummy variables representing
C             mean values at the beginning of a computation unit
C         --------------------------------------------------------------
          DO 120 J = 1,JE
            DO 110 K = 7,12
              VMEAN(K,J) = 0.D+00
  110       CONTINUE
  120     CONTINUE
          DO 140 K = 3,4
            DO 130 L= 1,NDELR
              RMEAN(K,L) = 0.D+00
  130       CONTINUE
            ESEA(K) = 0.D+00
  140     CONTINUE
          DO 160 J = 1,JE
            DO 150 K = 1,3
              EEXT(K,J) = 0.D+00
              EINT(K,J) = 0.D+00
  150       CONTINUE
            EINT(4,J) = 0.D+00
  160     CONTINUE
C
        ELSEIF (MODE.EQ.2) THEN
C         ::  MODE=2 is to do statistical operations
C         --------------------------------------------------------------
          DO 210 J = 1,IS
            VMEAN(7,J) = VMEAN(7,J) + UONE(1,J)
            VMEAN(8,J) = VMEAN(8,J) + U(J)
            VMEAN(9,J) = VMEAN(9,J) + QB(J)
            IF  (UONE(1,J).GT.VMAX(1,J)) VMAX(1,J)=UONE(1,J)
            IF        (U(J).GT.VMAX(2,J)) VMAX(2,J)=U(J)
            IF       (QB(J).GT.VMAX(3,J)) VMAX(3,J)=QB(J)
            IF  (UONE(1,J).LT.VMIN(1,J)) VMIN(1,J)=UONE(1,J)
```

```
            IF        (U(J).LT.VMIN(2,J)) VMIN(2,J)=U(J)
            IF       (QB(J).LT.VMIN(3,J)) VMIN(3,J)=QB(J)
  210     CONTINUE
          DO 220 J = 1,JE
            VMEAN(10,J) = VMEAN(10,J)  + ELEV(J)
            IF (ELEV(J).GT.VMAX(4,J)) VMAX(4,J)=ELEV(J)
            IF (ELEV(J).LT.VMIN(4,J)) VMIN(4,J)=ELEV(J)
  220     CONTINUE
          DO 230 J = 1,IW
            VMEAN(11,J) = VMEAN(11,J) + UP(J)
            VMEAN(12,J) = VMEAN(12,J) + AMP(J)
            IF  (UP(J).GT.VMAX(5,J)) VMAX(5,J)=UP(J)
            IF (AMP(J).GT.VMAX(6,J)) VMAX(6,J)=AMP(J)
            IF  (UP(J).LT.VMIN(5,J)) VMIN(5,J)=UP(J)
            IF (AMP(J).LT.VMIN(6,J)) VMIN(6,J)=AMP(J)
  230     CONTINUE
          DO 250 K = 1,2
            KPLUS2 = K+2
            DO 240 L = 1,NDELR
              RMEAN(KPLUS2,L) = RMEAN(KPLUS2,L) + RU(K,L)
              IF (RU(K,L).GT.RMAX(K,L)) RMAX(K,L)=RU(K,L)
              IF (RU(K,L).LT.RMIN(K,L)) RMIN(K,L)=RU(K,L)
  240       CONTINUE
  250     CONTINUE
          ESEA(3) = ESEA(3) + ETAI
          ESEA(4) = ESEA(4) + ETAR
          IF (ETAR.GT.ESEA(5)) ESEA(5)=ETAR
          IF (ETAR.LT.ESEA(6)) ESEA(6)=ETAR
          IF (IS.LT.ISMIN) ISMIN=IS
          IF (IW.LT.IWMIN) IWMIN=IW
          IF (IS.GT.ISMAX) ISMAX=IS
          IF (IW.GT.IWMAX) IWMAX=IW
          IF (JSTAB.GT.JSTABM) JSTABM=JSTAB
C         ::  Mean values of energy quantities are treated separately
C             in (48) ENERGY
          CALL ENERGY
C
        ELSEIF (MODE.EQ.3) THEN
C         ::  MODE=3 is to recapitulate variables representing mean
C             values at the end of a computation unit
C         -----------------------------------------------------------
          FACTOR = DBLE(INVT)*(TMAX-DBLE(MSTAT))
          DO 320 K = 1,6
            KPLUS6 = K+6
            DO 310 J = 1,JE
              VMEAN(K,J) = VMEAN(K,J) + VMEAN(KPLUS6,J)/FACTOR
  310       CONTINUE
  320     CONTINUE
          DO 340 K = 1,2
            KPLUS2 = K+2
            DO 330 L = 1,NDELR
              RMEAN(K,L) = RMEAN(K,L) + RMEAN(KPLUS2,L)/FACTOR
  330       CONTINUE
            ESEA(K) = ESEA(K) + ESEA(KPLUS2)/FACTOR
  340     CONTINUE
```

```
        DO 360 J = 1,JE
          DO 350 K = 1,3
            EEXTMN(K,J) = EEXTMN(K,J) + EEXT(K,J)/FACTOR
            EINTMN(K,J) = EINTMN(K,J) + EINT(K,J)/FACTOR
  350     CONTINUE
          EINTMN(4,J) = EINTMN(4,J) + EINT(4,J)/FACTOR
  360   CONTINUE
C     --------------------------------------------------------------
      ENDIF
C
      RETURN
      END
C
C .................... END OF SUBROUTINE 47 STAT ....................
C >>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 48 ENERGY <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 48 ENERGY computes energy quantities
C
      SUBROUTINE ENERGY
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      DIMENSION EEX(3),EIN(4)
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /STATT/  TSTAT1,TSTAT2,TSPAN
      COMMON /STATI/  IST1,IST2,IWT1,IWT2
      COMMON /ENERG/  EEXTMN(4,MXNOD),EEXT(5,MXNOD),
     2                EINTMN(4,MXNOD),EINT(6,MXNOD)
C
C     TSTAT1 = normalized time when statistical calculations begin
C     TSTAT2 = normalized time when statistical calculations end
C     In the comment lines in this subroutine, j indicates node number
C
C           REGION 1 (External Flow)
C           ---------------------------------------------------------
C           EEX(i) with i=1,2,3 is instantaneous quantity
C           EEXTMN(i,j) with i=1,2,3,4 is time-averaged quantity
C           EEX(1),EEXTMN(1,j)
C             = normalized specific (kinetic + potential) energy per
C               unit horizontal area
C           EEX(2),EEXTMN(2,j)
C             = normalized energy flux per unit width
C           EEX(3),EEXTMN(3,j)
C             = normalized energy flux per unit horizontal area into
C               the permeable underlayer
C           EEXTMN(4,j)
C             = normalized rate of energy dissipation per unit
```

```
C              horizontal area
C           EEXT(i,j) with i=1,2,3 is a dummy for EEXTMN(i,j)
C           EEXT(4,j) saves the value of EEX(1) at t=TSTAT1
C           EEXT(5,j) saves the value of EEX(1) at t=TSTAT2
C           EEXT(4,j) and EEXT(5,j) are processed by (49) SAVET
C           -------------------------------------------------------
      DO 120 J = 1,IS
        EEX(1) = (UONE(1,J)*U(J) + ELEV(J)*ELEV(J))/2.D+00
        IF (UONE(2,J).LT.ELEV(J)) THEN
          EEX(1) = EEX(1) - (UONE(2,J)-ELEV(J))**2/2.D+00
        ENDIF
        EEX(2) = UONE(1,J)*(U(J)*U(J)/2.D+00+ELEV(J))
        EEX(3) = PARPQ*QB(J)*(UB(J)*UB(J)/2.D+00+ELEV(J))
        DO 110 K = 1,3
          EEXT(K,J) = EEXT(K,J) + EEX(K)
  110   CONTINUE
  120 CONTINUE
C
C           REGIONS 2+3 (Internal Flow)
C           -------------------------------------------------------
C           EIN(i) with i=1,2,3,4 is instantaneous quantity
C           EINTMN(i,j) with i=1,2,3,4 is time-averaged quantity
C           EIN(1),EINTMN(1,j) = Ep
C           EIN(2),EINTMN(2,j) = Fp
C           EIN(3),EINTMN(3,j) = Dr
C           EIN(4),EINTMN(4,j) = Dp
C           EINT(i,j) with i=1,2,3,4 is a dummy for EINTMN(i,j)
C           EINT(5,j) saves the value of EIN(1) at t=TSTAT1
C           EINT(6,j) saves the value of EIN(1) at t=TSTAT2
C           EINT(5,j) and EINT(6,j) are processed by (49) SAVET
C           -------------------------------------------------------
      DO 220 J = 1,IW
        PU2UP2 = PARPU*PARPU*UP(J)*UP(J)
        EIN(2) = PARPQ*UP(J)*HP(J)*(PU2UP2/2.D+00+ELEV(J))
        EIN(3) = PARPQ*UP(J)*UP(J)*HP(J)*(DABS(UP(J))+PARMU)
        DO 210 K = 2,3
          EINT(K,J) = EINT(K,J) + EIN(K)
  210   CONTINUE
  220 CONTINUE
      DO 240 J = 1,IS
        PU2UP2 = PARPU*PARPU*UP(J)*UP(J)
        EIN(1) = PORO*PU2UP2*HP(J)/2.D+00
        IF (ZU(J).GT.0.D+00) THEN
          EIN(1) = EIN(1) + PORO*ZU(J)*ZU(J)/2.D+00
        ENDIF
        EIN(4) = PARPQ*QB(J)*(UB(J)*UB(J)/2.D+00+ELEV(J))
        DO 230 K = 1,4,3
          EINT(K,J) = EINT(K,J) + EIN(K)
  230   CONTINUE
  240 CONTINUE
      DO 250 J = IS+1,IW
        PU2UP2 = PARPU*PARPU*UP(J)*UP(J)
        EIN(1) = PORO*(PU2UP2*HP(J)+ELEV(J)*ELEV(J))/2.D+00
        IF (ZL(J).GT.0.D+00) THEN
          EIN(1) = EIN(1) - PORO*ZL(J)*ZL(J)/2.D+00
```

```
        ENDIF
        EINT(1,J) = EINT(1,J) + EIN(1)
  250 CONTINUE
C
      RETURN
      END
C
C ..................  END OF SUBROUTINE 48 ENERGY  ..................
C >>>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 49 SAVET  <<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 49 SAVET saves values at t=TSTAT1 and t=TSTAT2 where
C     . TSTAT1 = normalized time when statistical calculations begin
C     . TSTAT2 = normalized time when statistical calculations end
C
      SUBROUTINE SAVET (ICALL)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      DIMENSION DER(MXNOD)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /TEMPO/  TIME
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /STATT/  TSTAT1,TSTAT2,TSPAN
      COMMON /STATI/  IST1,IST2,IWT1,IWT2
      COMMON /ENERG/  EEXTMN(4,MXNOD),EEXT(5,MXNOD),
     2                EINTMN(4,MXNOD),EINT(6,MXNOD)
      COMMON /MASSB/  VMASS(4,MXNOD)
C
C     In the comment lines in this subroutine, j indicates node number
C
C     ::  Wave energy: See (48) ENERGY
C     ::  Mass balance:
C         VMASS(1,j) = UONE(2,j) at t=TSTAT1
C         VMASS(2,j) = UONE(2,j) at t=TSTAT2
C         VMASS(3,j) = ELEV(j) at t=TSTAT1
C         VMASS(4,j) = ELEV(j) at t=TSTAT2
C
      IF (ICALL.EQ.1) THEN
C       ::  ICALL=1 saves values at t=TSTAT1
C       ---------------------------------------------------------------
        TSTAT1 = TIME
        IST1   = IS
        IWT1   = IW
C       ::  Energy of External Flow
        DO 110 J = 1,IS
          EEX1 = (UONE(1,J)*U(J) + ELEV(J)*ELEV(J))/2.D+00
```

```
        IF (UONE(2,J).LT.ELEV(J)) THEN
          EEX1 = EEX1 - (UONE(2,J)-ELEV(J))**2/2.D+00
        ENDIF
        EEXT(4,J) = EEX1
  110 CONTINUE
C     :: Energy of Internal Flow
      DO 210 J = 1,IS
        PU2UP2  = PARPU*PARPU*UP(J)*UP(J)
        EIN1 = PORO*PU2UP2*HP(J)/2.D+00
        IF (ZU(J).GT.0.D+00) THEN
          EIN1 = EIN1 + PORO*ZU(J)*ZU(J)/2.D+00
        ENDIF
        EINT(5,J) = EIN1
  210 CONTINUE
      DO 220 J = IS+1,IW
        PU2UP2  = PARPU*PARPU*UP(J)*UP(J)
        EIN1 = PORO*(PU2UP2*HP(J)+ELEV(J)*ELEV(J))/2.D+00
        IF (ZL(J).GT.0.D+00) THEN
          EIN1 = EIN1 - PORO*ZL(J)*ZL(J)/2.D+00
        ENDIF
        EINT(5,J) = EIN1
  220 CONTINUE
C     :: Mass Balance
      DO 230 J = 1,IS
        VMASS(1,J) = UONE(2,J)
  230 CONTINUE
      DO 240 J = 1,IW
        VMASS(3,J) = ELEV(J)
  240 CONTINUE
C     -------------------------------------------------------------
    ELSEIF (ICALL.EQ.2) THEN
C     :: ICALL=2 saves values at t=TSTAT2
C     -------------------------------------------------------------
      TSTAT2 = TIME
      IST2   = IS
      IWT2   = IW
      TSPAN  = TSTAT2-TSTAT1
C     :: Energy of External Flow
      DO 310 J = 1,IS
        EEX1 = (UONE(1,J)*U(J) + ELEV(J)*ELEV(J))/2.D+00
        IF (UONE(2,J).LT.ELEV(J)) THEN
          EEX1 = EEX1 - (UONE(2,J)-ELEV(J))**2/2.D+00
        ENDIF
        EEXT(5,J) = EEX1
  310 CONTINUE
      DER(1)      = (EEXTMN(2,2)-EEXTMN(2,1))/X
      DER(ISMAX) = (EEXTMN(2,ISMAX)-EEXTMN(2,ISMAX-1))/X
      DO 320 J = 2,ISMAX-1
        DER(J) = (EEXTMN(2,J+1)-EEXTMN(2,J-1))/TWOX
  320 CONTINUE
      DO 330 J = 1,ISMAX
        TERM        = (EEXT(5,J)-EEXT(4,J))/TSPAN
        EEXTMN(4,J) = -DER(J)-EEXTMN(3,J)-TERM
  330 CONTINUE
C     :: Wave Energy of Internal Flow
```

```
         DO 410 J = 1,IS
           PU2UP2  = PARPU*PARPU*UP(J)*UP(J)
           EIN1 = PORO*PU2UP2*HP(J)/2.D+00
           IF (ZU(J).GT.0.D+00) THEN
             EIN1 = EIN1 + PORO*ZU(J)*ZU(J)/2.D+00
           ENDIF
           EINT(6,J) = EIN1
  410    CONTINUE
         DO 420 J = IS+1,IW
           PU2UP2  = PARPU*PARPU*UP(J)*UP(J)
           EIN1 = PORO*(PU2UP2*HP(J)+ELEV(J)*ELEV(J))/2.D+00
           IF (ZL(J).GT.0.D+00) THEN
             EIN1 = EIN1 - PORO*ZL(J)*ZL(J)/2.D+00
           ENDIF
           EINT(6,J) = EIN1
  420    CONTINUE
C        :: Mass Balance
         DO 430 J = 1,IS
           VMASS(2,J) = UONE(2,J)
  430    CONTINUE
         DO 440 J = 1,IW
           VMASS(4,J) = ELEV(J)
  440    CONTINUE
C        ---------------------------------------------------------------
      ENDIF
C
      RETURN
      END
C
C.................. END OF SUBROUTINE 49 SAVET ..................
C >>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 50 STABNO  <<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 50 STABNO computes stability number of armor units
C
      SUBROUTINE STABNO
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      DIMENSION DER(MXNOD)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /TEMPO/  TIME
      COMMON /BOT2/   DSEA,DSEA2,FW,TSLOPS
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /ARMOR1/ SG,TANPHI,C2,C3,CD,CL,CM,AMAX,AMIN
      COMMON /ARMOR2/ CSTAB1,CSTAB2,AMAXS,AMINS,E2,
     +                SSLOPE(MXNOD),CTAN(MXNOD),E3PRE(MXNOD)
```

```
      COMMON /ARMOR3/ SNR(MXNOD),SNSX(MXNOD),TSNSX(MXNOD),
     +                ATMIN(7,MXNOD),SNSC,TSNSC
      COMMON /ARMOR4/ JSTAB,JSNSC,JSTABM,JSATM,ISATM,IWATM
C
C           FLUID ACCELERATION
C           -----------------------------------------------------
      DER(1)  = (UONE(2,2)-UONE(2,1))/X
      DER(IS) = (UONE(2,IS)-UONE(2,IS-1))/X
      DO 110 J = 2,IS-1
        DER(J) = (UONE(2,J+1)-UONE(2,J-1))/TWOX
  110 CONTINUE
      DO 120 J = 1,IS
        TERM = PARPQ*QB(J)*(U(J)-UB(J)) - FW*U(J)*DABS(U(J))
        DUDT(J) = -DER(J) - TU(J) + TERM/UONE(2,J)
  120 CONTINUE
C
C           STABILITY NUMBER SNR
C           -----------------------------------------------------
C         . SNR(j) = stability number against rolling/sliding at
C           node j
C         . SNR is computed for j=1,2,...,JSTAB where JSTAB is
C           defined in (46) RUNUP
C           -----------------------------------------------------
      DO 130 J = 1,JSTAB
C
        IF (DABS(U(J)).LT.1.D-03) THEN
C         ::          Avoid having very small velocity values
C                     SNR=1000 indicates very stable units
          SNR(J) = 1.D+03
        ELSE
C         ::          Impose lower and upper bounds of fluid
C                     acceleration
          IF (DUDT(J).GT.AMAXS) DUDT(J)=AMAXS
          IF (DUDT(J).LT.AMINS) DUDT(J)=AMINS
C         ::          SNR=-1000 indicates that AMAX and AMIN
C                     specified in (04) INPUT1 needs to be
C                     modified
          VALUE = CSTAB2*DUDT(J)-SSLOPE(J)
          IF (DABS(VALUE).GT.CTAN(J)) THEN
            SNR(J) = -1.D+03
            WRITE  (*,9910) TIME,J,IS,IW
            WRITE (99,9910) TIME,J,IS,IW
            STOP
          ENDIF
C         ::          Compute SNR
          E1   = VALUE*CSTAB1/(U(J)*DABS(U(J)))
          E3   = E3PRE(J)/(U(J)*U(J))
          E1E2 = -E1*E2
          IF (E1.LT.0.D+00.AND.E2.GT.1.D+00.AND.E3.LT.E1E2) THEN
            SNR(J) = (E3+E1)/(E2-1.D+00)
          ELSE
            SNR(J) = (E3-E1)/(E2+1.D+00)
          ENDIF
        ENDIF
C
```

```
  130 CONTINUE
C
C         LOCAL STABILITY NUMBER SNSX
C         -----------------------------------------------------------
C         SNSX(j)  = local stability number
C                  = minimum of SNR at node j
C         TSNSX(j) = normalized time when SNSX(j) occurs
C         -----------------------------------------------------------
      DO 140 J = 1,JSTAB
        IF (SNR(J).LT.SNSX(J)) THEN
          SNSX(J)  = SNR(J)
          TSNSX(J) = TIME
        ENDIF
  140 CONTINUE
C
C         CRITICAL STABILITY NUMBER SNSC
C         -----------------------------------------------------------
C         SNSC  = critical stability number
C               = minimum of SNSX along the slope
C         TSNSC = normalized time when SNSC occurs
C         JSNSC = node number where SNSC occurs
C         At the time of critical stability, i.e., t=TSNSC:
C         . ATMIN(i,j) saves spatial variations, i=1,2,...,7
C           [see below for what each i represents]
C         . JSATM,ISATM,IWATM save the value of JSTAB,IS,IW
C         -----------------------------------------------------------
      INDI = 0
      DO 150 J = 1,JSTAB
        IF (SNSX(J).LT.SNSC) THEN
          SNSC  = SNSX(J)
          JSNSC = J
          TSNSC = TSNSX(J)
          INDI  = 1
        ENDIF
  150 CONTINUE
      IF (INDI.EQ.1) THEN
        JSATM = JSTAB
        ISATM = IS
        IWATM = IW
        DO 160 J = 1,JSATM
          ATMIN(1,J) = SNR(J)
  160   CONTINUE
        DO 170 J = 1,ISATM
          ATMIN(2,J) = U(J)
          ATMIN(3,J) = DUDT(J)
          ATMIN(4,J) = QB(J)
  170   CONTINUE
        DO 180 J = 1,IWATM
          ATMIN(5,J) = ELEV(J)
          ATMIN(6,J) = UP(J)
          ATMIN(7,J) = AMP(J)
  180   CONTINUE
      ENDIF
C
 9910 FORMAT (/':: FAILURE in Subroutine 50 STABNO ::'/
```

```
     +          '  Armor stability impossible'/
     +          '  Time of occurrence    TIME =',F18.9/
     +          '  Troubled node            J =',I8/
     +          '  Upper waterline node    IS =',I8/
     +          '  Lower waterline node    IW =',I8)
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 50 STABNO ...................
C >>>>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 51 DOC1 <<<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 51 DOC1 writes essential information before
C     time-marching computation
C
      SUBROUTINE DOC1 (ICOMP,ISEQ)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      CHARACTER UL*7,UT*10,UV*10
      CHARACTER*58 MSGS(12)
      COMMON /CONSTA/ PI,GRAV
      COMMON /OPTION/ ISYST,IBOT,ISALT
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  WLO,WL,UR,SURF,SIGMA
      COMMON /IWT1/   NDATA
      COMMON /IWT2/   ETAMA,ETAMI,ETA(MXTIM)
      COMMON /BOT1/   DSEAP,FWP
      COMMON /BOT2/   DSEA,DSEA2,FW,TSLOPS
      COMMON /BOT3/   NUSEG,NLSEG
      COMMON /BOT4/   WUSEG(MXSEG),TUSEG(MXSEG),
     2                XUSEG(MXSEG),ZUSEG(MXSEG),
     3                WLSEG(MXSEG),TLSEG(MXSEG),
     4                XLSEG(MXSEG),ZLSEG(MXSEG)
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /DWLINE/ DELRP(MXRUN),DELR(MXRUN),RU(2,MXRUN)
      COMMON /ARMOR1/ SG,TANPHI,C2,C3,CD,CL,CM,AMAX,AMIN
      COMMON /SEQI/   MSEQ(0:40),MULTIQ(40)
      COMMON /SEQD/   EPSIQ(2,40)
      DATA MSGS /
     1 'MULTIF = integer multiplication factor',
     2 'MULTIF*MINVT = minimum value of INVT used in a sequence',
     3 'INVT   = number of time steps in one reference wave period',
     4 'MINVT  = minimum allowable value of INVT',
     5 'EPSI1, EPSI2 = damping coefficients',
     6 'OMEGA  = numerical stability indicator',
     7 'NDIV   = time level in a computation unit when DELTAS or',
```

```
     8 '         DELTAW changes value',
     9 'DELTAS = normalized water depth defining the computational',
     + '         upper waterline',
     1 'DELTAW = normalized water depth defining the computational',
     2 '         lower waterline'/
C
C        SYSTEM OF UNITS
C        --------------------------------------------------------
      IF (ISYST.EQ.1) THEN
        UL = ' meters'
        UT = ' degrees C'
        UV = ' m.m/sec  '
      ELSE
        UL = ' feet   '
        UT = ' degrees F'
        UV = ' ft.ft/sec'
      ENDIF
C
C        WAVE CONDITION
C        --------------------------------------------------------
      WRITE (98,9811) NDATA,ETAMA,ETAMI,TP,HREFP,UL,DSEAP,UL,
     +             DSEA,WL,UR,SURF,SIGMA
 9811 FORMAT (/'WAVE CONDITION'//
     2        'Incident wave at seaward boundary given as input'/
     3        'Number of data points     NDATA =',I8/
     4        'Norm. maximum surface elev.     = ',F14.6/
     5        'Norm. minimum surface elev.     = ',F14.6/
     6        'Reference wave period           = ',F14.6,' sec.'/
     7        'Reference wave height           = ',F14.6,A7/
     8        'Depth at seaward boundary       = ',F14.6,A7/
     9        'Norm. depth at seaw. boundary   = ',F11.3/
     +        'Normalized wave length          = ',F11.3/
     1        'Ursell number                   = ',F11.3/
     2        'Surf similarity parameter       = ',F11.3/
     3        '"Sigma"                         = ',F11.3)
C
C        PROPERTIES OF THE SLOPES
C        --------------------------------------------------------
C        File 52 = 'BPSPACE'
C        XUL(j) = x-coordinate of upper and lower slope at node j
C        ZU(j),ZL(j) = z-coordinate of upper,lower slope at node j
C        --------------------------------------------------------
      WRITE (98,9821) FWP,FW,NUSEG,UL,UL,UL
      WRITE (98,9822)(K,WUSEG(K),TUSEG(K),XUSEG(K),ZUSEG(K),K=1,NUSEG)
      WRITE (98,9823)(K,XUSEG(K),ZUSEG(K),K=NUSEG+1,NUSEG+1)
      WRITE (98,9824)
      WRITE (98,9825) NLSEG,UL,UL,UL
      WRITE (98,9822)(K,WLSEG(K),TLSEG(K),XLSEG(K),ZLSEG(K),K=1,NLSEG)
      WRITE (98,9823)(K,XLSEG(K),ZLSEG(K),K=NLSEG+1,NLSEG+1)
      WRITE (98,9824)
 9821 FORMAT (/'PROPERTIES OF THE UPPER SLOPE'//
     2        'Friction factor           = ',F15.6/
     3        'Norm. friction factor     = ',F15.6/
     4        'Number of segments        = ',I8/56(1H-)/
     5     ' SEGMENT    WUSEG(I)    TUSEG(I)    XUSEG(I)     ZUSEG(I)'/
```

```
     6    '        I',5X,A7,12X,2(5X,A7)/56(1H-))
 9822 FORMAT (I8,4F12.6)
 9823 FORMAT (I8,24X,2F12.6)
 9824 FORMAT (56(1H-))
 9825 FORMAT (/'GEOMETRY OF THE LOWER SLOPE'//
     1         'Number of Segments          = ',I8/56(1H-)/
     2    ' SEGMENT   WLSEG(I)   TLSEG(I)   XLSEG(I)    ZLSEG(I)'/
     3    '        I',5X,A7,12X,2(5X,A7)/56(1H-))
C
C          PROPERTIES OF THE PERMEABLE UNDERLAYER and
C          PARAMETERS RELATED TO PERMEABILITY
C          ----------------------------------------------------
      WRITE (98,9831)
      IF (ISALT.EQ.0) THEN
        WRITE (98,9832) TEMPER,UT
      ELSE
        WRITE (98,9833) TEMPER,UT
      ENDIF
      VISCO = 1.D+06*VISCO
      WRITE (98,9834) VISCO,UV,DPP,UL,PORO,ALPAO,BETAO,
     +                ALPAP,BETAP,PARPU,PARPQ,PARMU
      VISCO = 1.D-06*VISCO
 9831 FORMAT (/'PROPERTIES OF THE PERMEABLE UNDERLAYER and'/
     1         'PARAMETERS RELATED TO PERMEABILITY'/)
 9832 FORMAT ('Fresh water at temperature         TEMPER = ',
     +         F15.3,A10)
 9833 FORMAT ('Sea water at temperature           TEMPER = ',
     +         F15.3,A10)
 9834 FORMAT ('Kinematic viscosity of water x 1.D+06'/
     2  '                     = 1.D+06 x VISCO = ',F18.6,A10/
     3  'Representative diameter of permeable'/
     4  '                underlayer materials  DPP = ',F15.3,A7/
     5  'Porosity of the permeable underlayer  PORO = ',F15.3/
     6  'Dimensionless constants              ALPAO = ',F15.3/
     7  '                                     BETAO = ',F15.3/
     8  'Coefficient expressing the laminar flow resistance'/
     9  '                      ALPAP [alpha prime] = ',F15.3/
     +  'Coefficient expressing the turbulent flow resistance'/
     1  '                       BETAP [beta prime] = ',F15.3/
     2  'Dimensionless parameters that arise in consequence'/
     3  '   of normalizing the governing equations:'/
     4  '                                    PARPU = ',D22.6/
     5  '                       PARPQ = PORO*PARPU = ',D22.6/
     6  '                                    PARMU = ',D22.6)
C          ARMOR STABILITY
C          ----------------------------------------------------
      WRITE (98,9841) SG,TANPHI,C2,C3,CD,CL,CM,AMAX,AMIN
 9841 FORMAT (/'INPUT PARAMETERS FOR ARMOR STABILITY'//
     +         'Specific gravity             SG = ',F9.3/
     +         'Armor friction Factor    TANPHI = ',F9.3/
     +         'Area coefficient             C2 = ',F9.3/
     +         'Volume coefficient           C3 = ',F9.3/
     +         'Drag coefficient             CD = ',F9.3/
     +         'Lift coefficient             CL = ',F9.3/
     +         'Inertia coefficient          CM = ',F9.3/
```

```
     +        'Upper and lower bounds of du/dt'/
     +        '         (normalized by gravity)'/
     +        '                          AMAX = ',F9.3/
     +        '                          AMIN = ',F9.3)
C
C        COMPUTATION PARAMETERS
C        -------------------------------------------------------------
      WRITE (98,9851) JE,INITS,INITW,MINVT,NRATE,X,TMAX
      WRITE (98,9852) MSTAT,NTIMES,MSAVA1,MSAVA2
      IF (ICOMP.EQ.4) WRITE (98,9853) MULTIF,EPSI1,EPSI2
      WRITE (98,9854)
      WRITE (98,9855) INITS,XUL(INITS),INITW,XUL(INITW),JE,XUL(JE)
 9851 FORMAT (/'COMPUTATION PARAMETERS'//
     2 'Total number of spatial nodes        JE = ',I8/
     3 'No. of nodes with upper slope below SWL'/
     4 '                                  INITS = ',I8/
     5 'No. of nodes with lower slope below SWL'/
     6 '                                  INITW = ',I8/
     7 'Minimum allowable value of INVT'/
     8 '                                = MINVT = ',I8/
     9 '  where INVT is the number of time steps'/
     + '  in one reference wave period'/
     1 'Computed time series are stored at rate NRATE'/
     2 '  per reference wave period with NRATE = ',I8/
     3 'Normalized delta x        X = ',D22.6/
     4 'Computation duration  TMAX = ',F18.6,' ref. wave periods')
 9852 FORMAT (
     2 'Statistical calculations are performed excluding'/
     3 '     the first MSTAT computation units'/
     4 '     with                       MSTAT = ',I8/
     5 'Instantaneous spatial variations are stored NTIMES times'/
     6 '     at equal intervals per reference wave period from'/
     7 '     computation unit MSAVA1 to MSAVA2, inclusive,'/
     8 '     with                      NTIMES = ',I8/
     9 '                               MSAVA1 = ',I8/
     + '                               MSAVA2 = ',I8)
 9853 FORMAT (
     2 'Integer multiplication factor   MULTIF = ',I8/
     3 'Minimum value of INVT used = MULTIF*MINVT'/
     4 'Damping coefficients             EPSI1 = ',F15.6/
     5 '                                 EPSI2 = ',F15.6)
 9854 FORMAT ('Other parameters are given in COMPUTATION LOG below')
 9855 FORMAT ('Positioning:'/
     2 '. Node       1          is at x =          0.000000'/
     3 '. Node',I8,' (INITS) is at x = ',F15.6/
     4 '. Node',I8,' (INITW) is at x = ',F15.6/
     5 '. Node',I8,' (JE)    is at x = ',F15.6)
C
C        COMPUTATION LOG
C        -------------------------------------------------------------
      MSEQ(0)=0
      WRITE (98,9861)
      IF (ICOMP.LT.4) THEN
        DO 110 I = 1,12
          WRITE (98,9862) MSGS(I)
```

```
  110   CONTINUE
        WRITE (98,9863)
        DO 120 I = 1,ISEQ-1
          WRITE (98,9864) I,MSEQ(I-1)+1,MULTIQ(I),
     +                    EPSIQ(1,I),EPSIQ(2,I)
          WRITE (98,9865) MSEQ(I)
  120   CONTINUE
        IF (ICOMP.EQ.3) THEN
          WRITE (98,9864) ISEQ,MSEQ(ISEQ-1)+1,MULTIQ(ISEQ),
     +                    EPSIQ(1,ISEQ),EPSIQ(2,ISEQ)
          WRITE (98,9865) MWAVE
        ELSE
          WRITE (98,9864) ISEQ,MSEQ(ISEQ-1)+1,MULTIF,EPSI1,EPSI2
          WRITE (98,9866)
        ENDIF
        WRITE (98,9867)
      ENDIF
      IF (ICOMP.EQ.3) THEN
        WRITE (98,9868)
      ELSE
        WRITE (98,9869)
      ENDIF
 9861 FORMAT (/'COMPUTATION LOG'/)
 9862 FORMAT (A)
 9863 FORMAT (/56(1H-)/
     2 ' Sequence  Computation  MULTIF     EPSI1     EPSI2'/
     + '   Number         Units'/56(1H-))
 9864 FORMAT (I10,6X,2I9,2F10.2)
 9865 FORMAT (15X,'to',I8)
 9866 FORMAT (15X,'to      ?')
 9867 FORMAT (56(1H-)/)
 9868 FORMAT (54(1H-)/
     2 '    Comp.   INVT   OMEGA    NDIV    DELTAS    DELTAW'/
     3 '    Unit'/54(1H-))
 9869 FORMAT (62(1H-)/
     2 '    Comp.   # of    INVT   OMEGA    NDIV    DELTAS    DELTAW'/
     3 '    Unit   Tries'/62(1H-))
C
C         HEADER FOR FILE 52 'OSPACE' [See (52) DOC2]
C         ------------------------------------------------------------
      IF (ICOMP.GE.3) THEN
        WRITE (52,*) '   INITS   INITW      JE'
        WRITE (52,9000) INITS,INITW,JE
        WRITE (52,5210)
        WRITE (52,7000) XUL(INITS),XUL(INITW),XUL(JE)
      ENDIF
 5210 FORMAT (8X,'x at INITS       x at INITW          x at JE')
 7000 FORMAT (4F18.9)
 9000 FORMAT (8I8)
C
      RETURN
      END
C
C ................... END OF SUBROUTINE 51 DOC1 ....................
C >>>>>>>>>>>>>>>>>>>>>>>>> SUBROUTINE 52 DOC2 <<<<<<<<<<<<<<<<<<<<<<<<<
```

```
C
C    Subroutine 52 DOC2 stores computed results at designated time
C    during time-marching computation
C
     SUBROUTINE DOC2 (MODE,M)
C
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
     PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
     CHARACTER*10 F1SAVB(20),F2SAVB(20)
     DIMENSION VONE(MXSAV),VTWO(MXSAV)
     COMMON /TEMPO/  TIME
     COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
    +                DUDT(MXNOD)
     COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
     COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
     COMMON /ETASEA/ ETAI,ETAR,ESEA(6)
     COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
     COMMON /DWLINE/ DELRP(MXRUN),DELR(MXRUN),RU(2,MXRUN)
     COMMON /SAVB/   NNODB,NODB(MXSAV)
     DATA F1SAVB /
    1 'O1SAVB01  ','O1SAVB02  ','O1SAVB03  ','O1SAVB04  ',
    2 'O1SAVB05  ','O1SAVB06  ','O1SAVB07  ','O1SAVB08  ',
    3 'O1SAVB09  ','O1SAVB10  ','O1SAVB11  ','O1SAVB12  ',
    4 'O1SAVB13  ','O1SAVB14  ','O1SAVB15  ','O1SAVB16  ',
    5 'O1SAVB17  ','O1SAVB18  ','O1SAVB19  ','O1SAVB20  '/
     DATA F2SAVB /
    1 'O2SAVB01  ','O2SAVB02  ','O2SAVB03  ','O2SAVB04  ',
    2 'O2SAVB05  ','O2SAVB06  ','O2SAVB07  ','O2SAVB08  ',
    3 'O2SAVB09  ','O2SAVB10  ','O2SAVB11  ','O2SAVB12  ',
    4 'O2SAVB13  ','O2SAVB14  ','O2SAVB15  ','O2SAVB16  ',
    5 'O2SAVB17  ','O2SAVB18  ','O2SAVB19  ','O2SAVB20  '/
C
     IF (MODE.EQ.1) THEN
C      ------------------------------------------------------------
C      :: MODE=1 writes standard time series, i.e.,
C          . elevations at the seaward boundary
C          . run-up heights on the upper and lower slopes
C          File 61 = 'OSEAWAV'
C          File 62 = 'ORUNUPU'
C          File 63 = 'ORUNUPL'
       DBLES = DBLE(IS)
       DBLEW = DBLE(IW)
       WRITE (61,8000) TIME,ETAI,ETAR,UONE(1,1)
       WRITE (62,8000) DBLES,(RU(1,L),L=1,NDELR)
       WRITE (63,8000) DBLEW,(RU(2,L),L=1,NDELR)
C
     ELSEIF (MODE.EQ.2) THEN
C      ------------------------------------------------------------
C      :: MODE=2 stores "A"
C          "A" = spatial variations of hydrodynamic quantities
C          File 52 = 'OSPACE'
       WRITE (52,8000) TIME
       WRITE (52,9000) IS,IW
       WRITE (52,8000) (ELEV(J),J=1,IW)
       WRITE (52,8000) (U(J),   J=1,IS)
```

```
        WRITE (52,8000) (UB(J),  J=1,IS)
        WRITE (52,8000) (QB(J),  J=1,IS)
        WRITE (52,8000) (UP(J),  J=1,IW)
        WRITE (52,8000) (AMP(J), J=1,IW)
C
      ELSEIF (MODE.EQ.3) THEN
C       ---------------------------------------------------------------
C       ::  MODE=3 opens and closes output files for storing "B"
C           (see MODE=4 below)
        MPACK = M/100 + 1
        IF (M.GT.1) THEN
          CLOSE (71)
          CLOSE (72)
        ENDIF
        CALL OPENF (1,71,F1SAVB(MPACK),'NEW')
        CALL OPENF (1,72,F2SAVB(MPACK),'NEW')
C
      ELSEIF (MODE.EQ.4) THEN
C       ---------------------------------------------------------------
C       ::  MODE=4 stores "B"
C           "B" = time series of volume flux and total water depth of
C                 Region 1 at specified nodes
C           File 71 ('01SAVBxx') stores volume flux of Region 1
C           File 72 ('02SAVBxx') stores total water depth of Region 1
C           where xx=01 contains the first 100 waves
C                 xx=02 contains the second 100 waves
C                 and so on
C           Timing for storing of "B" is identical to that of standard
C                 time series (see MODE=1) and can be retrieved from
C                 File 61 'OSEAWAV'
        DO 110 I = 1,NNODB
          J = NODB(I)
          VONE(I) = UONE(1,J)
          VTWO(I) = UONE(2,J)
  110   CONTINUE
        WRITE (71,8000) (VONE(I),I=1,NNODB)
        WRITE (72,8000) (VTWO(I),I=1,NNODB)
      ENDIF
 8000 FORMAT (5D15.6)
 9000 FORMAT (8I8)
C
      RETURN
      END
C
C ................... END OF SUBROUTINE 52 DOC2 ....................
C >>>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 53 DOC3  <<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 53 DOC3 documents results after time-marching
C     computation
C
      SUBROUTINE DOC3
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MXNOD=1000,MXTIM=30000,MXRUN=5,MXSEG=50,MXSAV=50)
      CHARACTER UL*7
```

```
      CHARACTER VTITLE(6)*32,ETITLE(12)*32,MTITLE(4)*32,STITLE(9)*32
      COMMON /OPTION/ ISYST,IBOT,ISALT
      COMMON /CPAR1/  JE,INITS,INITW,MWAVE,MINVT,MULTIF,INVT,
     +                MSTAT,NRATE,NHOP,MSAVA1,MSAVA2,NTIMES
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /WAVE1/  HREFP,TP,WLOP
      COMMON /WAVE2/  WLO,WL,UR,SURF,SIGMA
      COMMON /IWT1/   NDATA
      COMMON /IWT2/   ETAMA,ETAMI,ETA(MXTIM)
      COMMON /BOT1/   DSEAP,FWP
      COMMON /BOT2/   DSEA,DSEA2,FW,TSLOPS
      COMMON /BOT3/   NUSEG,NLSEG
      COMMON /BOT4/   WUSEG(MXSEG),TUSEG(MXSEG),
     2                XUSEG(MXSEG),ZUSEG(MXSEG),
     3                WLSEG(MXSEG),TLSEG(MXSEG),
     4                XLSEG(MXSEG),ZLSEG(MXSEG)
      COMMON /BOT5/   XUL(MXNOD),ZU(MXNOD),TU(MXNOD),
     +                ZL(MXNOD),TL(MXNOD)
      COMMON /ULAYER/ DPP,TEMPER,VISCO,ALPAP,BETAP,
     +                ALPA0,BETA0,PORO,PARPU,PARPQ,PARMU
      COMMON /HYDROE/ UONE(2,MXNOD),U(MXNOD),UB(MXNOD),C(MXNOD),
     +                DUDT(MXNOD)
      COMMON /HYDROI/ HP(MXNOD),UP(MXNOD),AMP(MXNOD)
      COMMON /HYDROG/ ELEV(MXNOD),QB(MXNOD)
      COMMON /VECONE/ FONE(2,MXNOD),GONE(2,MXNOD),A1(2,MXNOD),
     +                H(2,MXNOD),S1(MXNOD),D(2,MXNOD),P(2,MXNOD)
      COMMON /VECTWO/ FTWO(MXNOD),GTWO(MXNOD)
      COMMON /VECTRE/ UTRE(2,MXNOD),FTRE1(MXNOD),GTRE1(MXNOD)
      COMMON /ETASEA/ ETAI,ETAR,ESEA(6)
      COMMON /IWLINE/ NDELR,ISOLD,IS,IW,ISMIN,IWMIN,ISMAX,IWMAX
      COMMON /DWLINE/ DELRP(MXRUN),DELR(MXRUN),RU(2,MXRUN)
      COMMON /STATV/  VMEAN(12,MXNOD),VMAX(6,MXNOD),VMIN(6,MXNOD)
      COMMON /STATR/  RMEAN(4,MXRUN),RMAX(2,MXRUN),RMIN(2,MXRUN)
      COMMON /STATT/  TSTAT1,TSTAT2,TSPAN
      COMMON /STATI/  IST1,IST2,IWT1,IWT2
      COMMON /ENERG/  EEXTMN(4,MXNOD),EEXT(5,MXNOD),
     2                EINTMN(4,MXNOD),EINT(6,MXNOD)
      COMMON /MASSB/  VMASS(4,MXNOD)
      COMMON /ARMOR3/ SNR(MXNOD),SNSX(MXNOD),TSNSX(MXNOD),
     +                ATMIN(7,MXNOD),SNSC,TSNSC
      COMMON /ARMOR4/ JSTAB,JSNSC,JSTABM,JSATM,ISATM,IWATM
      DATA VTITLE /'Volume flux in Region 1      h.u',
     2            'Velocity in Region 1          U',
     3            'Volume flux into underlayer  QB',
     4            'Surface elevation          ELEV',
     5            'Underlayer discharge velocity UP',
     6            'Underlayer quantity         AMP'/
      DATA ETITLE /'Mean E  of external flow        ',
     2            'Mean F  of external flow        ',
     3            'Mean Dp of external flow        ',
     4            'Mean D  of external flow        ',
     5            'Specific energy E at t=TSTAT1   ',
     6            'Specific energy E at t=TSTAT2   ',
     7            'Mean Ep of internal flow        ',
```

```
     8            'Mean Fp of internal flow       ',
     9            'Mean Dr of internal flow       ',
     +            'Mean Dp of internal flow       ',
     1            'Ep of internal flow at t=TSTAT1 ',
     2            'Ep of internal flow at t=TSTAT2 '/
      DATA MTITLE /'UONE(2,j)=h(j) at t=TSTAT1     ',
     2            'UONE(2,j)=h(j) at t=TSTAT2     ',
     3            'ELEV(j) at t=TSTAT1            ',
     4            'ELEV(j) at t=TSTAT2            '/
      DATA STITLE /'Local stability number SNSX(j) ',
     2            'Time of occurrence for SNSX(j) ',
     3            ' SNR(j) at critical stability ',
     4            '   U(j) at critical stability ',
     5            'DUDT(j) at critical stability ',
     6            '  QB(j) at critical stability ',
     7            'ELEV(j) at critical stability ',
     8            '  UP(j) at critical stability ',
     9            ' AMP(j) at critical stability '/
C
C        SYSTEM OF UNITS
C        -------------------------------------------------------------
      IF (ISYST.EQ.1) THEN
        UL = '   [mm]'
      ELSE
        UL = ' [inch]'
      ENDIF
C
C        STATISTICAL CALCULATIONS
C        -------------------------------------------------------------
        WRITE (98,9811) TSTAT1,TSTAT2,TSPAN,IST1,IWT1,IST2,IWT2
 9811 FORMAT (/'STATISTICAL CALCULATIONS'//
     2 'Statistical calculations begins at    t=TSTAT1 =',F15.6/
     3 'Statistical calculations ends at      t=TSTAT2 =',F15.6/
     4 'Duration of statistical calculations     TSPAN =',F15.6/
     5 'Computational waterlines at t=TSTAT1:'/
     6 '                            . Upper slope: IST1 =',I8/
     7 '                            . Lower slope: IWT1 =',I8/
     8 'Computational waterlines at t=TSTAT2:'/
     9 '                            . Upper slope: IST2 =',I8/
     + '                            . Lower slope: IWT2 =',I8)
C
C        ELEVATIONS AT SEAWARD BOUNDARY
C        -------------------------------------------------------------
        WRITE (98,9821) (ESEA(I),I=1,2),(ESEA(I),I=5,6)
 9821 FORMAT (/'ELEVATIONS AT SEAWARD BOUNDARY'//
     2 'Time-averaged elevation due to incident waves =',F15.6/
     3 'Elevations due to reflected waves:'/
     4 '                                  . Time-averaged =',F15.6/
     5 '                                  . Maximum       =',F15.6/
     6 '                                  . Minimum       =',F15.6)
C
C        RUN-UP, RUN-DOWN, SET-UP
C        -------------------------------------------------------------
      WRITE (98,9831) ISMIN,IWMIN,ISMAX,IWMAX,UL
      WRITE (98,9832)(L,DELRP(L),RMEAN(1,L),RMAX(1,L),RMIN(1,L),L=1,1)
```

```
      DO 981 L = 2,NDELR
        WRITE (98,9834) L,DELRP(L),RMEAN(1,L),RMAX(1,L),RMIN(1,L)
  981 CONTINUE
      WRITE (98,9835)
      WRITE (98,9833)(L,DELRP(L),RMEAN(2,L),RMAX(2,L),RMIN(2,L),L=1,1)
      DO 982 L = 2,NDELR
        WRITE (98,9834) L,DELRP(L),RMEAN(2,L),RMAX(2,L),RMIN(2,L)
  982 CONTINUE
      WRITE (98,9835)
 9831 FORMAT (/'RUN-UP, RUN-DOWN, SET-UP'//
     2'Smallest nodes reached by computational waterlines:'/
     3'                          . Upper slope: ISMIN =',I8/
     4'                          . Lower slope: IWMIN =',I8/
     5'Largest nodes reached by computational waterlines:'/
     6'                          . Upper slope: ISMAX =',I8/
     7'                          . Lower slope: IWMAX =',I8/61(1H-)/
     8' Slope    i  Deltar(i)    Setup(i)    Runup(i)   Rundown(i)'/
     9'               ',A7,/61(1H-))
 9832 FORMAT (' Upper',I5,2X,F9.3,3(3X,F9.3))
 9833 FORMAT (' Lower',I5,2X,F9.3,3(3X,F9.3))
 9834 FORMAT (I11,2X,F9.3,3(3X,F9.3))
 9835 FORMAT (61(1H-))
C
C           STATISTICS OF HYDRODYNAMIC QUANTITIES
C           -------------------------------------------------------
C           File 51 = 'OSTAT'
C           JE    = landward edge node
C           ISMAX = the largest node reached by computational upper
C                   waterline
C           IWMAX = the largest node reached by computational lower
C                   waterline
C           See (47) STAT for a description of statistical quantities
C           -------------------------------------------------------
      WRITE (51,*)   '  INITS   INITW      JE'
      WRITE (51,9000) INITS,INITW,JE
      WRITE (51,*)   '  ISMIN   IWMIN   ISMAX   IWMAX'
      WRITE (51,9000) ISMIN,IWMIN,ISMAX,IWMAX
      WRITE (51,*)   '   IST1    IST2    IWT1    IWT2'
      WRITE (51,9000) IST1,IST2,IWT1,IWT2
      WRITE (51,*) '           TSTAT1             TSTAT2'
      WRITE (51,7000) TSTAT1,TSTAT2
      WRITE (51,*)   '"Abscisca of nodes".'
      WRITE (51,8000) (XUL(J),J=1,JE)
      WRITE (51,*)   '"Ordinates of upper slope".'
      WRITE (51,8000)  (ZU(J),J=1,JE)
      WRITE (51,*)   '"Ordinates of lower slope".'
      WRITE (51,8000)  (ZL(J),J=1,JE)
      DO 511 K = 1,6
        WRITE (51,5010) VTITLE(K)
        WRITE (51,8000) (VMAX(K,J), J=1,IWMAX)
        WRITE (51,5020) VTITLE(K)
        WRITE (51,8000) (VMEAN(K,J),J=1,IWMAX)
        WRITE (51,5030) VTITLE(K)
        WRITE (51,8000) (VMIN(K,J), J=1,IWMAX)
  511 CONTINUE
```

```
C
C          WAVE ENERGY
C          ---------------------------------------------------------
C          File 53 = 'OENERG'
C          See (48) ENERGY for a description of energy quantities
C          ---------------------------------------------------------
      WRITE (53,*) '   ISMIN   IWMIN   ISMAX   IWMAX'
      WRITE (53,9000) ISMIN,IWMIN,ISMAX,IWMAX
      WRITE (53,*) '    IST1    IST2    IWT1    IWT2'
      WRITE (53,9000) IST1,IST2,IWT1,IWT2
      WRITE (53,*) '   ISMAX   IWMAX      JE'
      WRITE (53,9000) ISMAX,IWMAX,JE
      WRITE (53,5310)
      WRITE (53,7000) XUL(ISMAX),XUL(IWMAX),XUL(JE)
      WRITE (53,*) '          TSTAT1              TSTAT2'
      WRITE (53,7000) TSTAT1,TSTAT2
      WRITE (53,*) '         Delta x'
      WRITE (53,7000) X
      DO 531 K = 1,4
        WRITE (53,5050) ETITLE(K)
        WRITE (53,8000) (EEXTMN(K,J),J=1,IWMAX)
  531 CONTINUE
      DO 532 K = 4,5
        WRITE (53,5050) ETITLE(K+1)
        WRITE (53,8000) (EEXT(K,J),J=1,IWMAX)
  532 CONTINUE
      DO 533 K = 1,4
        WRITE (53,5050) ETITLE(K+6)
        WRITE (53,8000) (EINTMN(K,J),J=1,IWMAX)
  533 CONTINUE
      DO 534 K = 5,6
        WRITE (53,5050) ETITLE(K+6)
        WRITE (53,8000) (EINT(K,J),J=1,IWMAX)
  534 CONTINUE
 5310 FORMAT (8X,'x at ISMAX         x at IWMAX          x at JE')
C
C          MASS BALANCE
C          ---------------------------------------------------------
C          File 54 = 'OMASS'
C          See (49) SAVET for a description of quantities needed to
C          check mass balance
C          ---------------------------------------------------------
      WRITE (54,*) '   ISMIN   IWMIN   ISMAX   IWMAX'
      WRITE (54,9000) ISMIN,IWMIN,ISMAX,IWMAX
      WRITE (54,*) '    IST1    IST2    IWT1    IWT2'
      WRITE (54,9000) IST1,IST2,IWT1,IWT2
      WRITE (54,*) '   ISMAX   IWMAX      JE'
      WRITE (54,9000) ISMAX,IWMAX,JE
      WRITE (54,5410)
      WRITE (54,7000) XUL(ISMAX),XUL(IWMAX),XUL(JE)
      WRITE (54,*) '          TSTAT1              TSTAT2'
      WRITE (54,7000) TSTAT1,TSTAT2
      WRITE (54,5420)
      WRITE (54,7000) X,PORO,PARPQ
      WRITE (54,5020) VTITLE(1)
```

```
      WRITE (54,8000) (VMEAN(1,J),J=1,IWMAX)
      WRITE (54,5020) VTITLE(6)
      WRITE (54,8000) (VMEAN(6,J),J=1,IWMAX)
      WRITE (54,5020) VTITLE(3)
      WRITE (54,8000) (VMEAN(3,J),J=1,IWMAX)
      DO 541 K = 1,4
        WRITE (54,5050) MTITLE(K)
        WRITE (54,8000) (VMASS(K,J),J=1,IWMAX)
  541 CONTINUE
 5410 FORMAT (8X,'x at ISMAX        x at IWMAX        x at JE')
 5420 FORMAT (11X,'Delta x  Porosity np=PORO   Param. pq=PARPQ')
C
C           ARMOR STABILITY
C           ------------------------------------------------------------
C           File 55 = 'OSTAB'
C           See (50) STABNO for a description of quantities related to
C           armor stability
C           ------------------------------------------------------------
      WRITE (98,9841) JSTABM,SNSC,TSNSC,JSNSC,XUL(JSNSC),ZU(JSNSC)
      WRITE (98,9842) JSATM,ISATM,IWATM
      WRITE (55,*) '  ISMAX   IWMAX     JE'
      WRITE (55,9000) ISMAX,IWMAX,JE
      WRITE (55,*) '  JSTABM   JSATM   ISATM   IWATM'
      WRITE (55,9000) JSTABM,JSATM,ISATM,IWATM
      WRITE (55,5510)
      WRITE (55,7000) XUL(ISMAX),XUL(IWMAX),XUL(JE)
      WRITE (55,5520)
      WRITE (55,7000) XUL(JSTABM),XUL(JSATM),XUL(ISATM),XUL(IWATM)
      WRITE (55,5530)
      WRITE (55,7000) ZU(JSTABM),ZU(JSATM),ZU(ISATM),ZU(IWATM)
      WRITE (55,*) '     zb at node 1'
      WRITE (55,7000) ZU(1)
      WRITE (55,*) '           TSTAT1           TSTAT2'
      WRITE (55,7000) TSTAT1,TSTAT2
      WRITE (55,5050) STITLE(1)
      WRITE (55,8000) (SNSX(J),J=1,IWMAX)
      WRITE (55,5050) STITLE(2)
      WRITE (55,8000) (TSNSX(J),J=1,IWMAX)
      DO 551 K = 1,7
        WRITE (55,5050) STITLE(K+2)
        WRITE (55,8000) (ATMIN(K,J),J=1,IWMAX)
  551 CONTINUE
 9841 FORMAT (/'ARMOR STABILITY'//
     +         'Largest node number for which armor stability'/
     +         '     computation is performed    JSTABM = ',I8/
     +         'Critical stability number Nsc      SNSC = ',F12.3/
     +         '  occurring at normalized time    TSNSC = ',F15.6/
     +         '                  at node number   JSNSC = ',I8/
     +         '                                x at JSNSC = ',F15.6/
     +         '                               zb at JSNSC = ',F15.6)
 9842 FORMAT ( 'At the time of critical stability:'/
     +         ' . Armor stability computation is'/
     +         '   performed up to node           JSATM = ',I8/
     +         ' . Upper waterline node           ISATM = ',I8/
     +         ' . Lower waterline node           IWATM = ',I8)
```

```
 5510 FORMAT (8X,'x at ISMAX        x at IWMAX          x at JE')
 5520 FORMAT ( '       x at JSTABM       x at JSATM',
     +         '        x at ISATM        x at IWATM')
 5530 FORMAT ( '      zb at JSTABM      zb at JSATM',
     +         '       zb at ISATM      zb at IWATM')
C
C          GENERAL FORMATS
C          ------------------------------------------------------------
 5010 FORMAT ('"',A32,' maximum".')
 5020 FORMAT ('"',A32,'    mean".')
 5030 FORMAT ('"',A32,' minimum".')
 5050 FORMAT ('"',A32,'".')
 7000 FORMAT (4F18.9)
 8000 FORMAT (5D15.6)
 9000 FORMAT (8I8)
C
      RETURN
      END
C
C .................. END OF SUBROUTINE 53 DOC3 ...................
C >>>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 54 NSI  <<<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 54 NSI calculates numerical stability indicator OMEGA
C
      SUBROUTINE NSI (OMEGA,INVT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CPAR2/  TMAX,DELTAS,DELTAW,EPSI1,EPSI2,
     +                X,T,TX,XT,TTX,TTXX,TWOX
      COMMON /BOT2/   DSEA,DSEA2,FW,TSLOPS
      EPSI  = DMAX1(EPSI1,EPSI2)
      DUM1  = 1.D+00 + EPSI*EPSI/4.D+00
      DUM2  = DSQRT(DUM1) - EPSI/2.D+00
      OMEGA = DUM2*X*DBLE(INVT)/(1.D+00+DSEA2)
      RETURN
      END
C
C ................... END OF SUBROUTINE 54 NSI ...................
C >>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 55 OPENF  <<<<<<<<<<<<<<<<<<<<<<<<
C
C     Subroutine 55 OPENF opens a specified file in either
C     non-destructive (MODE=1) or destructive (MODE.NE.1) mode
C
      SUBROUTINE OPENF (MODE,NUNIT,FNAME,FSTAT)
      CHARACTER FNAME*10,FSTAT*3
      IF (MODE.EQ.1) THEN
        OPEN (UNIT=NUNIT,FILE=FNAME,STATUS=FSTAT,ACCESS='SEQUENTIAL')
      ELSE
        OPEN (UNIT=NUNIT,FILE=FNAME)
      ENDIF
      RETURN
      END
C
C .................. END OF SUBROUTINE 55 OPENF ..................
C >>>>>>>>>>>>>>>>>>>>>>>>  SUBROUTINE 56 STOPP  <<<<<<<<<<<<<<<<<<<<<<<<
C
```

```
C     Subroutine 56 STOPP executes a programmed stop
C
      SUBROUTINE STOPP (IBEGIN,IEND)
C
      CHARACTER*55 MSG(14)
      DATA MSG /
     1 ' Choice of ICOMP out of range.',
     2 ' Need larger MXDIV in the main program.',
     3 ' Mismatch between M and MREAD in ICOMP=3.',
     4 ' File TRANSFER might have been altered.',
     5 ' DIAGNOSTIC part incomplete. ACTUAL part can not run.',
     6 ' IPROB not zero from MAIN2.',
     7 ' MSAVA2 must be .GE. MSAVA1.',
     8 ' Temperature out of range.',
     9 ' Upper slope can not extend beyond lower slope.',
     + ' Upper slope always below SWL. PBREAK can not run.',
     1 ' Slope too long for present setting of MXNOD.',
     2 ' Lower slope always below SWL, unacceptable in PBREAK.',
     3 ' Upper waterline moved more than one node.',
     4 ' Lower waterline moved more than one node.'/
      DO 100 I = IBEGIN,IEND
        WRITE  (*,9910) MSG(I)
        WRITE (99,9910) MSG(I)
  100 CONTINUE
      WRITE  (*,9920)
      WRITE (99,9920)
 9910 FORMAT (/A55)
 9920 FORMAT (' Programmed Stop.')
C
      STOP
      END
C
C .................. END OF SUBROUTINE 56 STOPP  ..................
```

APPENDIX B

EXAMPLE: RUN P1

Kobayashi, Cox, and Wurjanto (1991) reported small scale laboratory experiments in which impermeable and permeable slopes were subjected to irregular waves attack. Their three permeable runs, identified as Runs P1, P2, and P3, are simulated using PBREAK in this report. This appendix shows the computed results from Run P1.

• INPUT DATA •

The experiment procedures are discussed in the following so that the input data to the PBREAK computation can be understood.

- **Bottom geometry.** The upper and lower slopes specified for the PBREAK computation of Runs P1, P2, and P3 are depicted in Figure B-1.
  - The seaward boundary for the numerical computation is taken to be at the toe of the slope, where the water depth is 0.4m $\Longrightarrow$ DSEAP=0.4m under ISYST=1 (International System of Units).
  - The *upper* slope is uniform with an inclination of 1:3. A single linear segment is sufficient to represent the upper slope $\Longrightarrow$ NUSEG=1. IBOT=1 (width-slope representation) is selected. TUSEG(1)=$\frac{1}{3}$=0.333333. Since the slope is uniform, TSLOPS=TUSEG(1)= 0.333333. The horizontal width WUSEG(1)=2.4m specified for the PBREAK computation should be sufficient for the movement of the upper and lower waterlines without any wave overtopping or transmission. It is noted that PBREAK will inform the user during the computation if the specified slope is not long enough.
  - Two segments are needed to represent the *lower* slope $\Longrightarrow$ NLSEG=2.
    The first segment is horizontal [TLSEG(1)=0] and 0.566049m long [WLSEG(1)= 0.566049m]. This horizontal width corresponds to the permeable underlayer thickness of 0.179m measured perpendicular to the upper slope. The actual thickness of the permeable underlayer was 0.2m, but the top one layer of stones is considered an armor layer, whose thickness is taken to be equal to the median diameter of the underlayer materials, 0.021m. In PBREAK the armor layer is represented by the roughness of the upper slope to be discussed later.
    The second segment is parallel to the upper slope $\Longrightarrow$ TLSEG(2)=0.333333. It is required that the total horizontal width of the lower slope be larger than that of the upper slope. WLSEG(2) is taken to be 1.835m so that the total horizontal width of the lower slope is slightly larger than 2.4m.

- **Slope roughness.** The top one layer of stones is considered an armor layer, whose roughness effect on the flow is represented by the friction factor f'=FWP. To estimate the value of FWP, use may be made of the rough *impermeable* slope tests of Kobayashi and Greenwald (1988) where the friction factor FWP for impermeable structures was calibrated over a wide range of the surf similarity parameter ξ for regular waves. The value of ξ for Run P1 using irregular waves is 1.72, which suggests FWP=.05 as a first approximation.

- **The wave train at the toe of the slope.** In the experiment of Kobayashi, Cox, and Wurjanto (1991), wave gages recorded the free surface elevations at the toe and seaward of the slope. The recorded total waves were separated into the incident wave $\eta_i(t)$ and the reflected waves $\eta_r(t)$ at the toe of the slope. The duration of the wave trains in Run P1 was 185 seconds.

  - The measured *incident* wave train is used as the input wave train. FINP2=tkcwp1 in the primary input data file specifies the name of the file containing the input wave train. The wave train is plotted in Figure B-2 and is stored in the file `appenb.iwt` on the disk accompanying this report.
  - The significant wave height of the input wave train is taken to be the reference wave height $\Longrightarrow$ HREFP=0.0685m. The mean period of the input wave train is taken to be the reference wave period $\Longrightarrow$ TP=1.082s [reported as 1.08s in Kobayashi, Cox, and Wurjanto (1991)].
  - The normalized computation duration TMAX is obtained by normalizing the duration of the input wave train (185s) by the reference wave period (1.082s) $\Longrightarrow$ TMAX= 170.979667.

- **Water depths defining waterlines.** Four water depths defining upper and lower waterlines are specified in this computation $\Longrightarrow$ NDELR=4. The first DELRP is reserved for the armor stability computation to be discussed later. The second DELRP corresponds to the water depth used for the measurement of the upper waterline, DELRP(2)=27.5mm. The third and fourth DELRPs may be of interest if the user wishes to examine the sensitivity of the computed waterlines to the parameter DELRP.

- **Properties of permeable underlayer.** Fresh water of 20° Celcius was used in the experiment $\Longrightarrow$ ISALT=0 and TEMPER=20. The characteristic diameter of the permeable materials is taken to be the median diameter $\Longrightarrow$ DPP=0.021m. PORO=0.48 represents the porosity. The dimensionless constants α_o and β_o are taken to be 1140 and 2.7, respectively, of the ranges $780 \leq \alpha_o \leq 1500$ and $1.8 \leq \beta_o \leq 3.6$ as explained in relation to Eqs. 9 and 10.

- **Armor stability parameters.** The following parameters are specified after Kobayashi and Greenwald (1988) and Kobayashi, Cox, and Wurjanto (1991): the specific density of the gravel SG=2.7; the area coefficient C2=0.9; volume coefficient C3=0.6; drag coefficient CD=0.5; inertia coefficient CM=1.5; $tan\phi$=TANPHI=1.191754 (ϕ=50°); the upper and lower limits for the normalized fluid acceleration, AMAX=1.0 and AMIN=–0.8. The lift coefficient is simply taken as CL=0.3 which is a representative value on the basis of the calibration for *impermeable* slopes reported by Kobayashi and Greenwald (1988).
  The landward limit of the armor stability computation is determined by the water depth DELRP(1), which is taken to be equal to the median diameter of the stones $\Longrightarrow$ DELRP(1)= 21mm.

The instrumental computation parameters that include INITS, MULTIF, DELTAS, DELTAW, EPSI1, and EPSI2 are chosen as follows.

- INITS=200. This corresponds to the grid spacing $\Delta x'$=0.006m, which is sufficiently small for the characteristic wave with H'= 0.0685m and linear wavelength L'=1.66m at the seaward boundary (T'=1.082s).
- The *diagnostic* part of Run P1 consists of two sequences. Listed in the following are the instrumental computation parameters specified in the primary input data files for the two sequences.

Sequence	Computation Units	INITS	MULTIF	DELTAS	DELTAW	EPSI1	EPSI2
1	1-45	200	1	.001000	.001000	1.00	1.00
2	46-171	200	1	.001000	.002000	1.00	1.00

Notes:

- Of the six instrumental parameters, only INITS must be kept constant throughout the duration of the computation; the others may vary. MULTIF, EPSI1, and EPSI2 are constant within a sequence. DELTAS and DELTAW may vary within a computation unit. In light of the variability of MULTIF, it is worth mentioning here that the computation parameter INVT is constant within a computation unit. See the explanation regarding MULTIF and INVT in Section 3.7.
- The sequential procedure for this computation will be explained further in a later section under the heading "SEQUENTIAL PROCEDURE".

The remaining input parameters for this example are:

- MSAVA1=162, MSAVA2=167, NTIMES=4. The spatial variations of the hydrodynamic quantities η, u, q_b, u_b, u_p, and m_p are stored at the normalized time t = 161.25, 161.50, 161.75, ..., 167.00. In all, there are 24 instantaneous variations stored for each of the six quantities.
- NNODB=5. The time series of m=UONE(1,j$_B$) and h=UONE(2,j$_B$) at the five specified nodes j$_B$ = NODB = 1, 51, 101, 151, and 201 are stored during the computation.
- MSTAT=0. The statistical calculations cover the entire computation duration, 0< t ≤TMAX.
- NRATE=50. The rate of storing computed time series is 50 data points per wave period.

This run is executed following the option number 1 in Figure 4 in Section 3.1. Thus, there are two stages of computation, that is, the *diagnostic* (ICOMP=1,2) part and the *actual* (ICOMP=3) part. The primary input data file for the *actual* part of PBREAK computation for Run P1 is presented in Table B-1 and stored as the file appenb.inp on the accompanying disk. It is noted that the values of the instrumental computation parameters specified in *this* file have no effect on the computation of the *actual* (ICOMP=3) part since the values needed are provided by the file TRANSFER produced by the *diagnostic* part. The values of the instrumental computation parameters used for the two sequences of the diagnostic part have been discussed and tabulated above. It is also noted that all the computations made for this report were performed on SUN MICROSYSTEM SPARC2 machines. Our experience in working on an IBM-3090 mainframe and the SPARC2 suggests that PBREAK is sensitive to the floating point processor. Performing PBREAK computations on different machines might require different sequencing of the diagnostic part, although the computed essential results should be the same.

The file ODOC containing a concise documentation produced by PBREAK for Run P1 is presented in Table B-2 and stored on the accompanying disk under the name `appenb.doc`. The user who is interested in examining the actual values of the instrumental computation parameters used in the computation may find the comprehensive information in this file ODOC under the heading COMPUTATION LOG.

• FIGURES •

Figures for Run P1 that can be obtained directly from the PBREAK's output files without additional analyses, such as spectral analyses, are presented in this appendix. For further reference, the user may read Part I of this report and the RBREAK report by Wurjanto and Kobayashi (1991) that summarize papers and reports discussing physical and probabilistic interpretations of the computed results using time series and spectral analyses.

In addition to the plots of the computed results, two measured quantities are also plotted to be compared with their respective computed results: the reflected wave train at the seaward boundary $\eta_r(t)$ and the run-up height on the upper slope $Z_r^U(t)$. The index j in the following list is used to indicate the spatial node j located at x=(j−1)Δx.

- Figure B-3 shows the computed and measured time series of the reflected wave train at the seaward boundary, η_r=ETAR. The computed data points are retrieved from the file OSEAWAV, column 3.

- Figure B-4 shows a portion of the computed time series of the surface elevation η at four different locations listed below. The data points are retrieved from the files O2SAVB01 and O2SAVB02. Some processing is needed since these files actually contain the time series of the total water depth h instead of η. The equation $\eta(x,t) = h(x,t) - d(x,t)$ relates h and η where d=-ZB(j) is the normalized water depth below the still water level above the upper slope. The information on ZB(j) is obtained from the file OSTAT.

x	NODE NUMBER	COLUMN NUMBER IN FILES O2SAVB01 & O2SAVB02
0.34	51	2
0.68	101	3
1.01	151	4
1.35	201	5

- Figure B-5 shows the computed and measured time-varying elevation of the *upper* waterline Z_r^U=RUNUPU(1,2) corresponding to the physical water depth δ'_r=27.5mm. The computed data points are retrieved from the file ORUNUPU, column 3.

- Figure B-6 shows the computed time series of the elevation of the *lower* waterline Z_r^L= RUNUPL(2,2) corresponding to the physical water depth δ_r'=27.5mm. The computed data points are retrieved from the file ORUNUPL, column 3.

- Figures B-7 through B-9 show the spatial variations of the instantaneous hydrodynamic quantities η=ELEV(j), u=U(j), q_b=QB(j), u_b=UB(j), u_p=UP(j), and m_p=AMP(j) at six of the specified 24 time levels. The data points are from the file OSPACE.

- Figures B-10 through B-15 show the mean, maximum, and minimum values of the hydrodynamic quantities η, u, m, q_b, u_p, and m_p over the entire computation duration, $0< t \leq$TMAX. The data points are obtained from the file OSTAT.

QUANTITY	NAME OF VARIABLE IN PBREAK		
	MEAN	MAXIMUM	MINIMUM
η	VMEAN(4,j)	VMAX(4,j)	VMIN(4,j)
u	VMEAN(2,j)	VMAX(2,j)	VMIN(2,j)
m	VMEAN(1,j)	VMAX(1,j)	VMIN(1,j)
q_b	VMEAN(3,j)	VMAX(3,j)	VMIN(3,j)
u_p	VMEAN(5,j)	VMAX(5,j)	VMIN(5,j)
m_p	VMEAN(6,j)	VMAX(6,j)	VMIN(6,j)

- Figures B-16 and B-17 show how well PBREAK satisfies the mass balances for Region 1 and Regions 2+3 expressed by Eq. 118 and 119, respectively. The solid line in each figure represents the sum of the other quantities shown. The sum must be zero if PBREAK satisfies the mass balance exactly. These figures indicate that there are slight errors in the computed mass balance partly because of the difficulty in predicting relatively small time-averaged quantities near the waterline at SWL accurately. The data points for the quantities $\overline{m}$, $\overline{q_b}$, $\overline{m_p}$ can be retrieved from either the file OMASS or OSTAT.

  - $\overline{m}$ = VMEAN(1,j)
  - $\overline{q_b}$ = VMEAN(3,j)
  - $\overline{m_p}$ = VMEAN(6,j)

  These basic quantities must be processed outside PBREAK to obtain the quantities shown in Figures B-16 and B-17. The values of Δx, p_q, and n_p required to do the processing can be found in either the file OMASS or ODOC.

- Figures B-18 and B-19 show the spatial variations of the time-averaged energy quantities. The data points are obtained from the file OENERG.

  - $\overline{E}$ = EEXTMN(1,j)
  - $\overline{F}$ = EEXTMN(2,j)
  - $\overline{D_p}$ = EEXTMN(3,j)
  - $\overline{D}$ = EEXTMN(4,j)
  - $\overline{E_p}$ = EINTMN(1,j)

- $\overline{F_p}$ = EINTMN(2,j)
- $\overline{D_r}$ = EINTMN(3,j)
- $\overline{D_p}$ = EINTMN(4,j)

- Figure B-20 shows how well PBREAK satisfies the energy balance for Regions 2+3 expressed by Eq. 134. There are slight errors in the computed energy balance as has been the case with the computed mass balance. The quantities $\overline{F_p}$, $\overline{D_r}$, and $\overline{D_p}$ are retrieved from the file OENERG (see the explanation for Figures B-18 and B-19). The derivative $d\overline{F_p}/dx$ must be processed outside PBREAK. The value of Δx to calculate $d\overline{F_p}/dx$ is also available in the file OENERG.

- Figures B-21 and B-22 show the spatial variations related to armor stability where z_b given by Eq. 35 is used as the spatial variable instead of the usual x-coordinate. With this method of presentation, the location of the still water level $z_b = 0$ is clear, which is an important elevation for the armor stability analysis.

  - Figure B-21 shows the spatial variation of the local stability number N_{sx}= SNSX(j). The critical stability number N_{sc}=1.155 occurred at the node number 156 at the time t=163.434. More information on the minimum armor stability can be found in the file ODOC presented in Table B-2.
  - Figure B-22 shows the spatial variations of various quantities at the time of the critical armor stability.
    - η = ATMIN(5,j)
    - u = ATMIN(2,j)
    - du/dt = ATMIN(3,j)
    - q_b = ATMIN(4,j)
    - u_p = ATMIN(6,j)
    - N_R = ATMIN(1,j)

    It is noted that the landward extent for the armor stability computation is limited by DELRP(1)= 21mm specified as input.

• Sequential Procedure •

The first leg of the diagnostic part of this PBREAK simulation was the computation with ICOMP=1. Two input data files were needed. The primary input data file for the computation with ICOMP=1 was identical to that presented in Table B-1. The file containing the input wave train is included in the accompanying disk under the name appenb.iwt. This second input data file was needed in all the legs of computation. The computation with ICOMP=1 broke down in the computation unit 54 because of a remediable numerical problem IPROB=5. The last 19 lines of messages written in the file CMSG01 are shown below.

```
:: IPROB=5 from Subr. 23 MARCH3 ::
   Computation unit                M =        54
   Lower waterline node           IW =       299
   Computation parameters       INVT =      6150
                              DELTAW =          0.001000000
   Water depth at (IW-1)             =          0.000359297
   Time of occurrence           TIME =         53.698699187
   Indicators                  OMEGA =          7.53
                                ITRY =         6

:: Notice of FAILURE from Subr. 19 CINVT ::
   Computation was aborted because OMEGA exceeds 10.
   Numer. stab. indicator      OMEGA =         11.26
   Computation unit                M =        54
   Trial number                 ITRY =         7
   Computation parameters       INVT =      9200
                              DELTAS =          0.001000000
                              DELTAW =          0.001000000
   Programmed Stop.
```

Because of the nature of the problem IPROB=5, we suspected that the cause of the breakdown was too small a value of DELTAW. So, for the subsequent computation with ICOMP=2, we increased the value of DELTAW to 0.002. The primary input data file for this computation with ICOMP=2 differs from that of the computation with ICOMP=1 only in the value of DELTAW. In addition to the primary and second input data files, the files CINPUT01, CVAL01, and SEQUENCE were needed to execute the subsequent computation with ICOMP=2. However, we did not need to do anything as far as the last three files were concerned because they were self-managed by PBREAK. It turned out that only one computation with ICOMP=2 was needed to complete the diagnostic part. This single computation with ICOMP=2, which started at the computation unit 46 as self-determined by PBREAK, ran successfully all the way up to the normalized terminal time t_{max}.

Upon the completion of the diagnostic part, the actual part with ICOMP=3 was then computed. The primary input data file for the actual part is presented in Table B-1. The computation with ICOMP=3 also needed as input the files SEQUENCE and TRANSFER created by the preceding diagnostic part. The resulting file ODOC is presented in Table B-2. The output data points are plotted in Figures B-2 through B-22.

Table B-1: Primary input data file for PBREAK computation of Run P1.

```
        4                    --> NLINES
-------------------------------------------------------------------------
             PBREAK          *** Kobayashi, Cox, and Wurjanto 1991
                             *** Run P1
-------------------------------------------------------------------------
1                            --> ISYST
1                            --> IBOT
      162       167        4 --> MSAVA1,MSAVA2,NTIMES
        5                    --> NNODB
        1        51      101 --> NODB(1)-(2)-(3)
      151       201          --> NODB(4)-(5)
tkcwp1                       --> FINP2
    170.979667               --> TMAX(normalized)
------ Block of instrumental computation parameters begins
      200                    --> INITS
        1                    --> MULTIF
       .001000       .001000 --> DELTAS,DELTAW(normalized)
      1.000000      1.000000 --> EPSI1,EPSI2
------ Block ends
        0                    --> MSTAT
       50                    --> NRATE
        4                    --> NDELR
     21.000000               --> DELRP(1)(millimeters)
     27.500000               -->      (2)
     15.000000               -->      (3)
      5.000000               -->      (4)
       .068500      1.082000 --> HREFP(meters),TP(seconds)
       .050000               --> FWP
       .400000               --> DSEAP(meters)
       .333333               --> TSLOPS
        1                    --> NUSEG
      2.400000       .333333 --> WUSEG(1)(meters),TUSEG(1)
        2                    --> NLSEG
       .566049       .000000 --> WLSEG(1)(meters),TLSEG(1)
      1.835000       .333333 -->      (2)             (2)
------ Block of permeable underlayer data begins
0                            --> ISALT
     20.000000               --> TEMPER(celcius)
       .021000               --> DPP(meters)
       .480000               --> PORO
   1140.000000      2.700000 --> ALPA0,BETA0
------ Block ends
------ Block of armor stability data begins
      2.700000      1.191754 --> SG,TANPHI
       .900000       .600000 --> C2,C3
       .500000       .300000 --> CD,CL
      1.500000               --> CM
      1.000000      -.800000 --> AMAX,AMIN
------ Block ends
```

Table B-2: File ODOC produced by PBREAK computation of Run P1.

```
--------------------------------------------------------------------
           PBREAK            *** Kobayashi, Cox, and Wurjanto 1991
                             *** Run P1
--------------------------------------------------------------------


WAVE CONDITION

Incident wave at seaward boundary given as input
Number of data points   NDATA =    5921
Norm. maximum surface elev.    =     0.836300
Norm. minimum surface elev.    =    -0.852300
Reference wave period          =     1.082000 sec.
Reference wave height          =     0.068500 meters
Depth at seaward boundary      =     0.400000 meters
Norm. depth at seaw. boundary  =     5.839
Normalized wave length         =     4.148
Ursell number                  =     2.947
Surf similarity parameter      =     1.722
"Sigma"                        =    12.948

PROPERTIES OF THE UPPER SLOPE

Friction factor          =       0.050000
Norm. friction factor    =       0.323710
Number of segments       =       1
-------------------------------------------------------
 SEGMENT   WUSEG(I)   TUSEG(I)   XUSEG(I)   ZUSEG(I)
    I       meters                meters     meters
-------------------------------------------------------
    1     2.400000   0.333333   0.000000  -0.400000
    2                           2.400000   0.399999
-------------------------------------------------------

GEOMETRY OF THE LOWER SLOPE

Number of Segments       =       2
-------------------------------------------------------
 SEGMENT   WLSEG(I)   TLSEG(I)   XLSEG(I)   ZLSEG(I)
    I       meters                meters     meters
-------------------------------------------------------
    1     0.566049   0.000000   0.000000  -0.400000
    2     1.835000   0.333333   0.566049  -0.400000
    3                           2.401049   0.211666
-------------------------------------------------------

PROPERTIES OF THE PERMEABLE UNDERLAYER and
PARAMETERS RELATED TO PERMEABILITY

Fresh water at temperature           TEMPER =          20.000 degrees C
Kinematic viscosity of water x 1.D+06
                    = 1.D+06 x VISCO =                  1.004000 m.m/sec
Representative diameter of permeable
                underlayer materials   DPP =            0.021 meters
Porosity of the permeable underlayer  PORO =            0.480
```

Table B-2 continued.

```
Dimensionless constants                  ALPA0 =          1140.000
                                         BETA0 =             2.700
Coefficient expressing the laminar flow resistance
                       ALPAP [alpha prime] =                 1.584
Coefficient expressing the turbulent flow resistance
                        BETAP [beta prime] =               604.539
Dimensionless parameters that arise in consequence
    of normalizing the governing equations:
                                         PARPU =                0.899691D-01
                        PARPQ = PORO*PARPU =                0.431851D-01
                                         PARMU =                0.740098D-01

INPUT PARAMETERS FOR ARMOR STABILITY

Specific gravity             SG =     2.700
Armor friction Factor    TANPHI =     1.192
Area coefficient             C2 =     0.900
Volume coefficient           C3 =     0.600
Drag coefficient             CD =     0.500
Lift coefficient             CL =     0.300
Inertia coefficient          CM =     1.500
Upper and lower bounds of du/dt
        (normalized by gravity)
                           AMAX =     1.000
                           AMIN =    -0.800

COMPUTATION PARAMETERS

Total number of spatial nodes        JE =       400
No. of nodes with upper slope below SWL
                                  INITS =       200
No. of nodes with lower slope below SWL
                                  INITW =       295
Minimum allowable value of INVT
                                = MINVT =       850
  where INVT is the number of time steps
  in one reference wave period
Computed time series are stored at rate NRATE
  per reference wave period with NRATE =         50
Normalized delta x          X =          0.676464D-02
Computation duration  TMAX =          170.979667 ref. wave periods
Statistical calculations are performed excluding
     the first MSTAT computation units
     with                         MSTAT =         0
Instantaneous spatial variations are stored NTIMES times
     at equal intervals per reference wave period from
     computation unit MSAVA1 to MSAVA2, inclusive,
     with                        NTIMES =         4
                                 MSAVA1 =       162
                                 MSAVA2 =       167
Other parameters are given in COMPUTATION LOG below
Positioning:
. Node      1         is at x =        0.000000
. Node    200 (INITS) is at x =        1.346163
. Node    295 (INITW) is at x =        1.988804
. Node    400 (JE)    is at x =        2.699091
```

Table B-2 continued.

COMPUTATION LOG

MULTIF = integer multiplication factor
MULTIF*MINVT = minimum value of INVT used in a sequence
INVT = number of time steps in one reference wave period
MINVT = minimum allowable value of INVT
EPSI1, EPSI2 = damping coefficients
OMEGA = numerical stability indicator
NDIV = time level in a computation unit when DELTAS or DELTAW changes value
DELTAS = normalized water depth defining the computational upper waterline
DELTAW = normalized water depth defining the computational lower waterline

Sequence Number	Computation Units	MULTIF	EPSI1	EPSI2
1	1 to 45	1	1.00	1.00
2	46 to 171	1	1.00	1.00

Comp. Unit	INVT	OMEGA	NDIV	DELTAS	DELTAW
1	850	1.0	850	0.001000	0.001000
2	850	1.0	850	0.001000	0.001000
3	850	1.0	850	0.001000	0.001000
4	850	1.0	850	0.001000	0.001000
5	850	1.0	850	0.001000	0.001000
6	850	1.0	850	0.001000	0.001000
7	850	1.0	850	0.001000	0.001000
8	850	1.0	850	0.001000	0.001000
9	850	1.0	850	0.001000	0.001000
10	850	1.0	850	0.001000	0.001000
11	850	1.0	850	0.001000	0.001000
12	850	1.0	850	0.001000	0.001000
13	850	1.0	850	0.001000	0.001000
14	850	1.0	850	0.001000	0.001000
15	850	1.0	850	0.001000	0.001000
16	850	1.0	850	0.001000	0.001000
17	850	1.0	850	0.001000	0.001000
18	850	1.0	850	0.001000	0.001000
19	850	1.0	850	0.001000	0.001000
20	850	1.0	850	0.001000	0.001000
21	850	1.0	850	0.001000	0.001000
22	850	1.0	850	0.001000	0.001000
23	850	1.0	850	0.001000	0.001000
24	850	1.0	850	0.001000	0.001000
25	850	1.0	850	0.001000	0.001000
26	850	1.0	850	0.001000	0.001000

Table B-2 continued.

167	27	850	1.0	850	0.001000	0.001000
168	28	850	1.0	850	0.001000	0.001000
169	29	850	1.0	850	0.001000	0.001000
170	30	850	1.0	850	0.001000	0.001000
171	31	850	1.0	850	0.001000	0.001000
172	32	850	1.0	850	0.001000	0.001000
173	33	850	1.0	850	0.001000	0.001000
174	34	850	1.0	850	0.001000	0.001000
175	35	850	1.0	850	0.001000	0.001000
176	36	850	1.0	850	0.001000	0.001000
177	37	850	1.0	850	0.001000	0.001000
178	38	850	1.0	850	0.001000	0.001000
179	39	850	1.0	850	0.001000	0.001000
180	40	850	1.0	850	0.001000	0.001000
181	41	850	1.0	850	0.001000	0.001000
182	42	850	1.0	850	0.001000	0.001000
183	43	850	1.0	850	0.001000	0.001000
184	44	850	1.0	850	0.001000	0.001000
185	45	850	1.0	850	0.001000	0.001000
186	46	850	1.0	850	0.001000	0.002000
187	47	850	1.0	850	0.001000	0.002000
188	48	850	1.0	850	0.001000	0.002000
189	49	850	1.0	850	0.001000	0.002000
190	50	850	1.0	850	0.001000	0.002000
191	51	850	1.0	850	0.001000	0.002000
192	52	850	1.0	850	0.001000	0.002000
193	53	850	1.0	850	0.001000	0.002000
194	54	850	1.0	850	0.001000	0.002000
195	55	850	1.0	850	0.001000	0.002000
196	56	850	1.0	850	0.001000	0.002000
197	57	850	1.0	850	0.001000	0.002000
198	58	850	1.0	850	0.001000	0.002000
199	59	850	1.0	850	0.001000	0.002000
200	60	850	1.0	850	0.001000	0.002000
201	61	850	1.0	850	0.001000	0.002000
202	62	850	1.0	850	0.001000	0.002000
203	63	850	1.0	850	0.001000	0.002000
204	64	850	1.0	850	0.001000	0.002000
205	65	850	1.0	850	0.001000	0.002000
206	66	850	1.0	850	0.001000	0.002000
207	67	850	1.0	850	0.001000	0.002000
208	68	850	1.0	850	0.001000	0.002000
209	69	850	1.0	850	0.001000	0.002000
210	70	850	1.0	850	0.001000	0.002000
211	71	850	1.0	850	0.001000	0.002000
212	72	850	1.0	850	0.001000	0.002000
213	73	850	1.0	850	0.001000	0.002000
214	74	850	1.0	850	0.001000	0.002000
215	75	850	1.0	850	0.001000	0.002000
216	76	850	1.0	850	0.001000	0.002000
217	77	850	1.0	850	0.001000	0.002000
218	78	850	1.0	850	0.001000	0.002000
219	79	850	1.0	850	0.001000	0.002000
220	80	850	1.0	850	0.001000	0.002000
221	81	850	1.0	850	0.001000	0.002000
222	82	850	1.0	850	0.001000	0.002000

Table B-2 continued.

223	83	850	1.0	850	0.001000	0.002000
224	84	850	1.0	850	0.001000	0.002000
225	85	850	1.0	850	0.001000	0.002000
226	86	850	1.0	850	0.001000	0.002000
227	87	850	1.0	850	0.001000	0.002000
228	88	850	1.0	850	0.001000	0.002000
229	89	850	1.0	850	0.001000	0.002000
230	90	850	1.0	850	0.001000	0.002000
231	91	850	1.0	850	0.001000	0.002000
232	92	850	1.0	850	0.001000	0.002000
233	93	850	1.0	850	0.001000	0.002000
234	94	850	1.0	850	0.001000	0.002000
235	95	850	1.0	850	0.001000	0.002000
236	96	850	1.0	850	0.001000	0.002000
237	97	850	1.0	850	0.001000	0.002000
238	98	850	1.0	850	0.001000	0.002000
239	99	850	1.0	850	0.001000	0.002000
240	100	850	1.0	850	0.001000	0.002000
241	101	850	1.0	850	0.001000	0.002000
242	102	850	1.0	850	0.001000	0.002000
243	103	850	1.0	850	0.001000	0.002000
244	104	850	1.0	850	0.001000	0.002000
245	105	850	1.0	850	0.001000	0.002000
246	106	850	1.0	850	0.001000	0.002000
247	107	850	1.0	850	0.001000	0.002000
248	108	850	1.0	850	0.001000	0.002000
249	109	850	1.0	850	0.001000	0.002000
250	110	850	1.0	850	0.001000	0.002000
251	111	850	1.0	850	0.001000	0.002000
252	112	850	1.0	850	0.001000	0.002000
253	113	850	1.0	850	0.001000	0.002000
254	114	850	1.0	850	0.001000	0.002000
255	115	850	1.0	850	0.001000	0.002000
256	116	850	1.0	850	0.001000	0.002000
257	117	850	1.0	850	0.001000	0.002000
258	118	850	1.0	850	0.001000	0.002000
259	119	850	1.0	850	0.001000	0.002000
260	120	850	1.0	850	0.001000	0.002000
261	121	850	1.0	850	0.001000	0.002000
262	122	850	1.0	850	0.001000	0.002000
263	123	850	1.0	850	0.001000	0.002000
264	124	850	1.0	850	0.001000	0.002000
265	125	850	1.0	850	0.001000	0.002000
266	126	850	1.0	850	0.001000	0.002000
267	127	850	1.0	850	0.001000	0.002000
268	128	850	1.0	850	0.001000	0.002000
269	129	850	1.0	850	0.001000	0.002000
270	130	850	1.0	850	0.001000	0.002000
271	131	850	1.0	850	0.001000	0.002000
272	132	850	1.0	850	0.001000	0.002000
273	133	850	1.0	850	0.001000	0.002000
274	134	850	1.0	850	0.001000	0.002000
275	135	850	1.0	850	0.001000	0.002000
276	136	850	1.0	850	0.001000	0.002000
277	137	850	1.0	850	0.001000	0.002000
278	138	850	1.0	850	0.001000	0.002000

```
      139    850    1.0    850  0.001000  0.002000
      140    850    1.0    850  0.001000  0.002000
      141    850    1.0    850  0.001000  0.002000
      142    850    1.0    850  0.001000  0.002000
      143    850    1.0    850  0.001000  0.002000
      144    850    1.0    850  0.001000  0.002000
      145    850    1.0    850  0.001000  0.002000
      146    850    1.0    850  0.001000  0.002000
      147    850    1.0    850  0.001000  0.002000
      148    850    1.0    850  0.001000  0.002000
      149    850    1.0    850  0.001000  0.002000
      150    850    1.0    850  0.001000  0.002000
      151    850    1.0    850  0.001000  0.002000
      152    850    1.0    850  0.001000  0.002000
      153    850    1.0    850  0.001000  0.002000
      154    850    1.0    850  0.001000  0.002000
      155    850    1.0    850  0.001000  0.002000
      156    850    1.0    850  0.001000  0.002000
      157    850    1.0    850  0.001000  0.002000
      158    850    1.0    850  0.001000  0.002000
      159    850    1.0    850  0.001000  0.002000
      160    850    1.0    850  0.001000  0.002000
      161    850    1.0    850  0.001000  0.002000
      162    850    1.0    850  0.001000  0.002000
      163    850    1.0    850  0.001000  0.002000
      164    850    1.0    850  0.001000  0.002000
      165    850    1.0    850  0.001000  0.002000
      166    850    1.0    850  0.001000  0.002000
      167    850    1.0    850  0.001000  0.002000
      168    850    1.0    850  0.001000  0.002000
      169    850    1.0    850  0.001000  0.002000
      170    850    1.0    850  0.001000  0.002000
      171    850    1.0    832  0.001000  0.002000
-------------------------------------------------------

STATISTICAL CALCULATIONS

Statistical calculations begins at    t=TSTAT1 =      0.000000
Statistical calculations ends at      t=TSTAT2 =    170.978824
Duration of statistical calculations     TSPAN =    170.978824
Computational waterlines at t=TSTAT1:
                          . Upper slope: IST1 =    200
                          . Lower slope: IWT1 =    295
Computational waterlines at t=TSTAT2:
                          . Upper slope: IST2 =    209
                          . Lower slope: IWT2 =    296

ELEVATIONS AT SEAWARD BOUNDARY

Time-averaged elevation due to incident waves =      0.000003
Elevations due to reflected waves:
                              . Time-averaged =      0.011700
                              . Maximum       =      0.105245
                              . Minimum       =     -0.101129
```

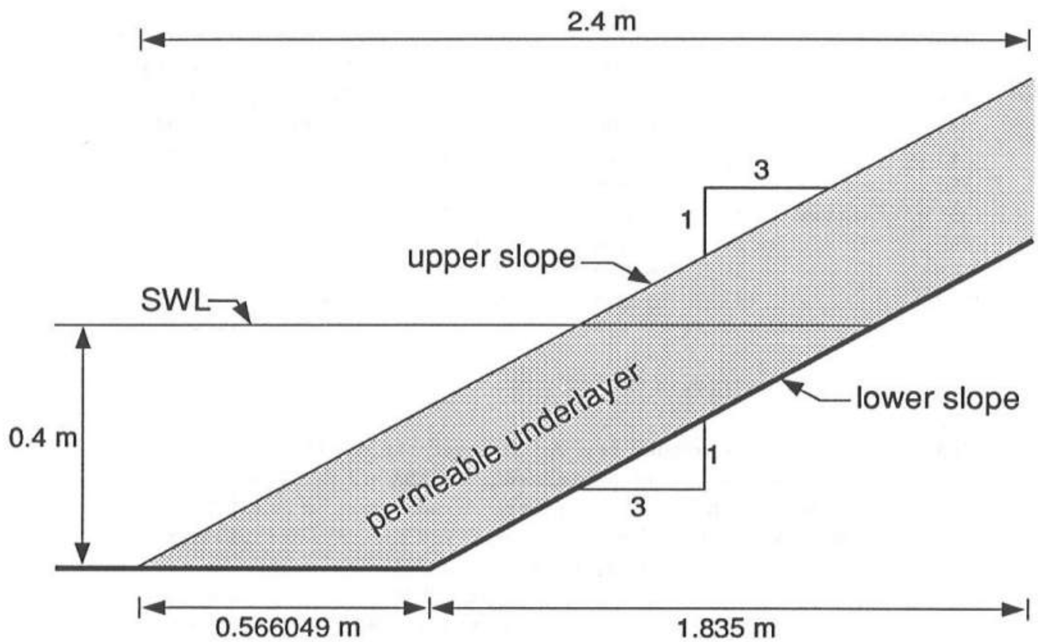
```

RUN-UP, RUN-DOWN, SET-UP

Smallest nodes reached by computational waterlines:
                         . Upper slope: ISMIN =      180
                         . Lower slope: IWMIN =      292
Largest nodes reached by computational waterlines:
                         . Upper slope: ISMAX =      235
                         . Lower slope: IWMAX =      312
-------------------------------------------------------------
 Slope    i   Deltar(i)    Setup(i)    Runup(i)   Rundown(i)
                [mm]
-------------------------------------------------------------
 Upper    1    21.000       0.069       1.020       -0.951
          2    27.500       0.061       1.007       -0.973
          3    15.000       0.075       1.025       -0.913
          4     5.000       0.084       1.017       -0.761
-------------------------------------------------------------
 Lower    1    21.000       0.094       0.489       -0.090
          2    27.500       0.094       0.494       -0.093
          3    15.000       0.095       0.487       -0.088
          4     5.000       0.095       0.491       -0.088
-------------------------------------------------------------


ARMOR STABILITY

Largest node number for which armor stability
     computation is performed   JSTABM =        225
Critical stability number Nsc      SNSC =          1.155
  occurring at normalized time    TSNSC =        163.434118
               at node number     JSNSC =        156
                              x at JSNSC =          1.048519
                             zb at JSNSC =         -1.313869
At the time of critical stability:
 . Armor stability computation is
   performed up to node           JSATM =        171
 . Upper waterline node           ISATM =        202
 . Lower waterline node           IWATM =        302
```



Permeable Structure of Kobayashi, Cox, and Wurjanto (1991)

Figure B-1

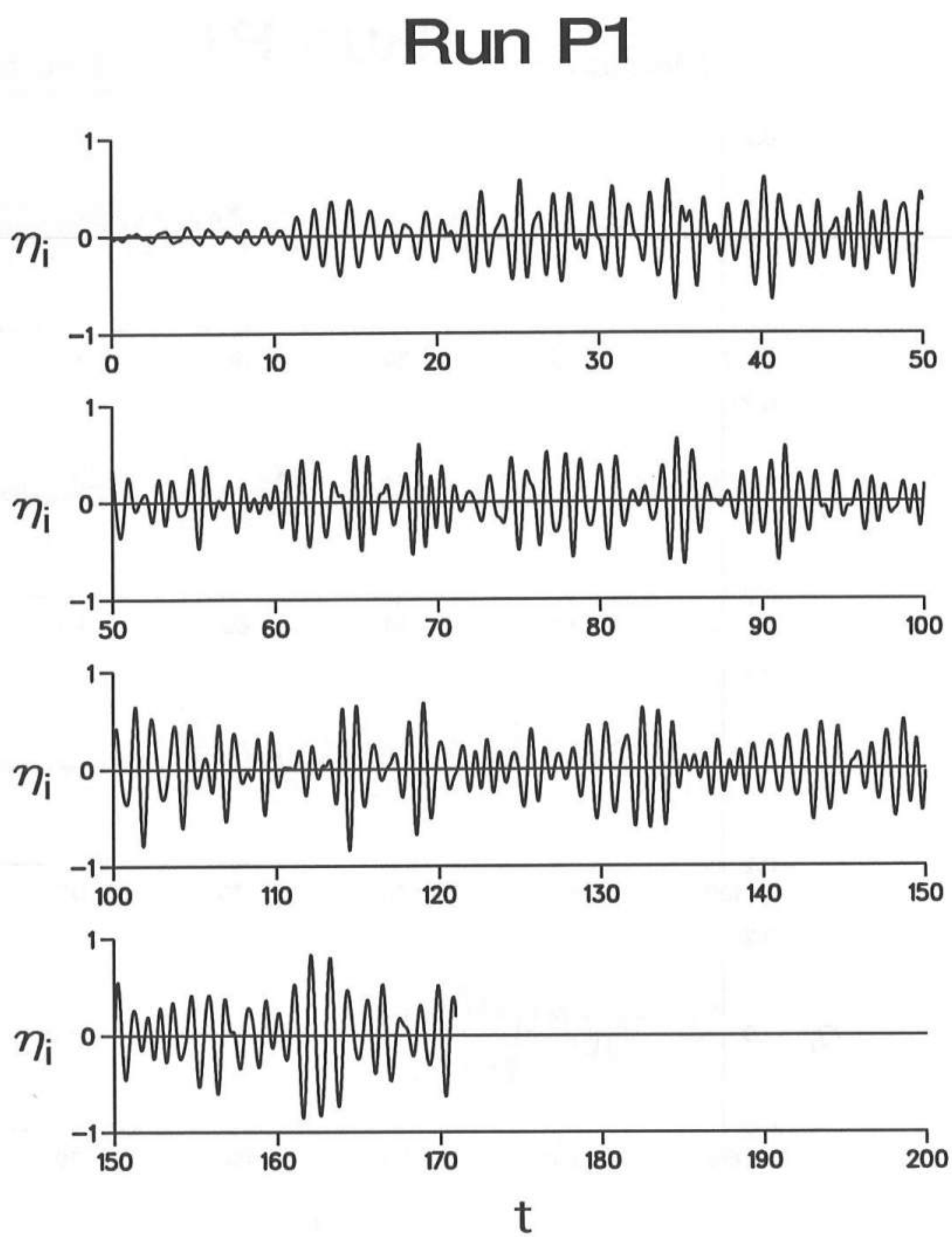

Run P1
1
0
-1
η_i
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
t

Figure B-2

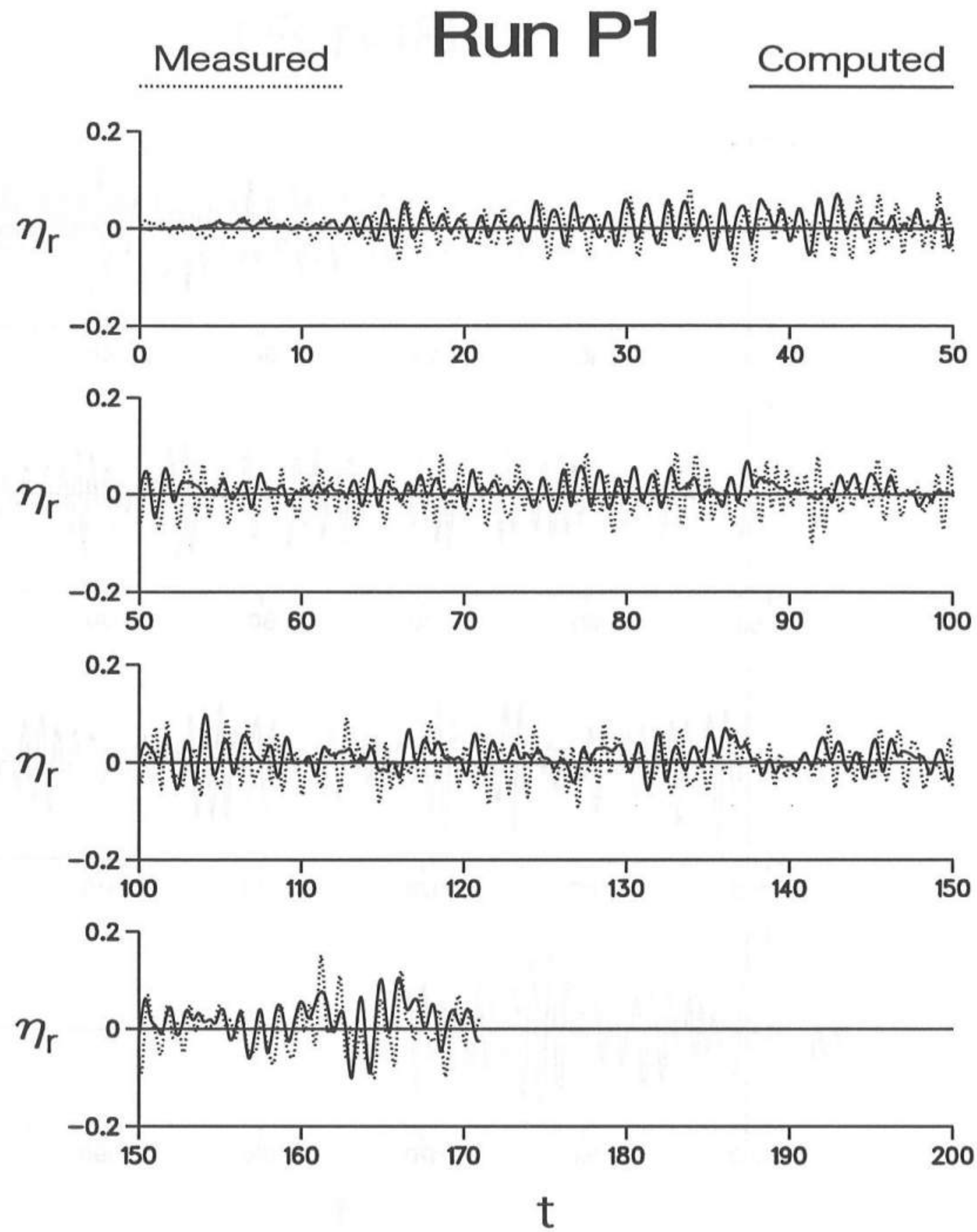

Run P1
Measured
Computed
0.2
0
−0.2
η_r
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
t

Figure B-3

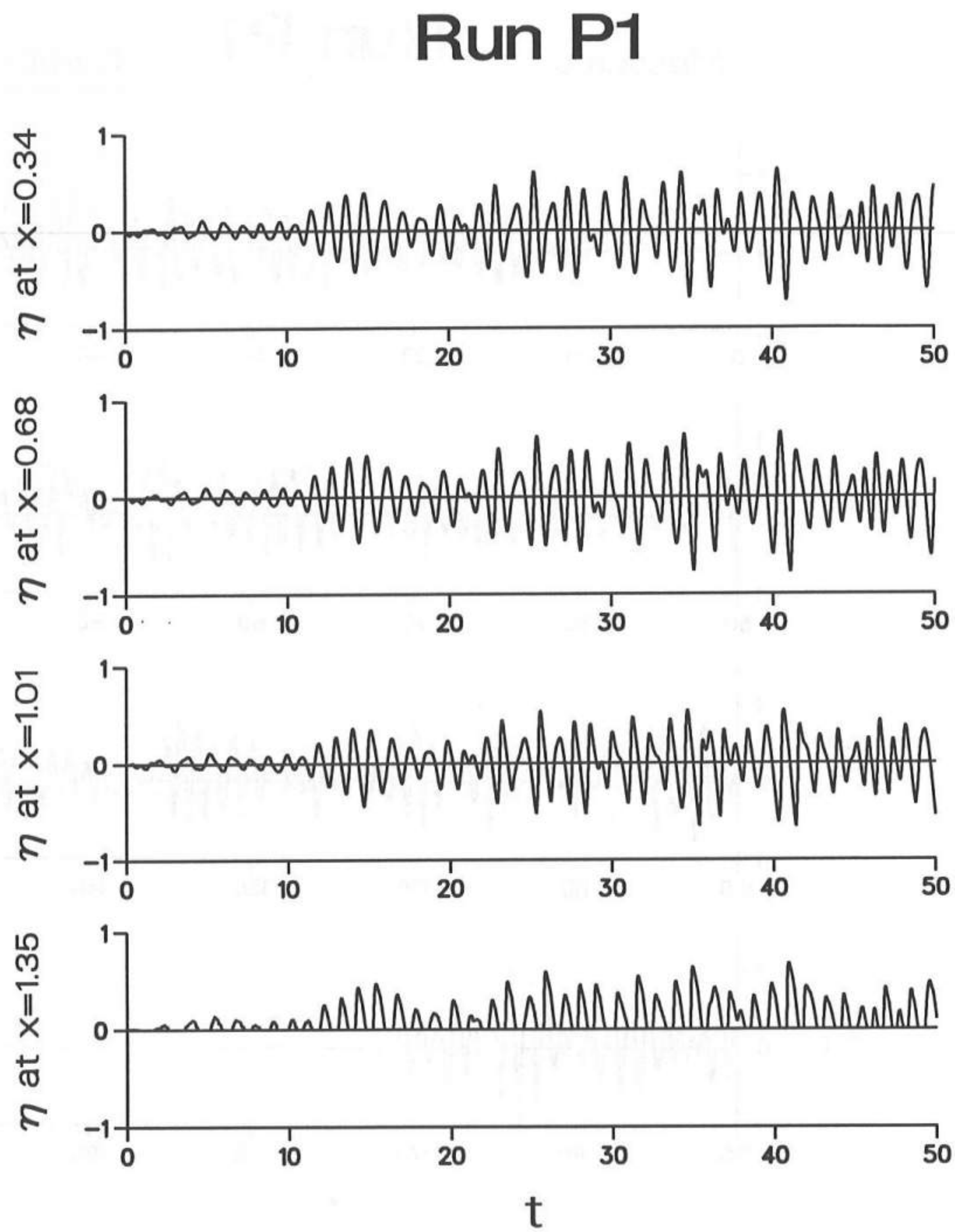


Figure B-4

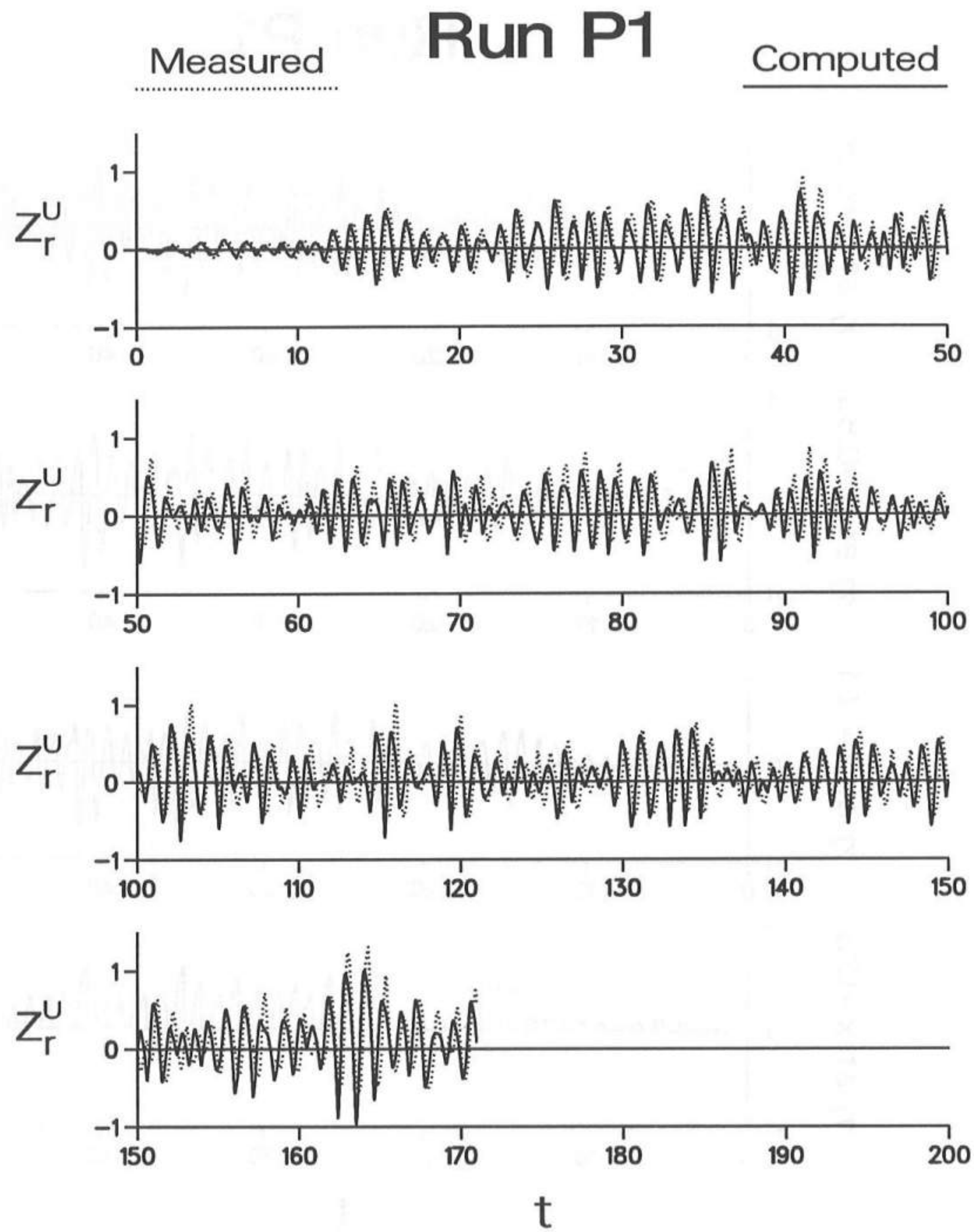

Run P1
Measured
Computed
Z_r^U
1
0
−1
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
t

Figure B-5

Run P1

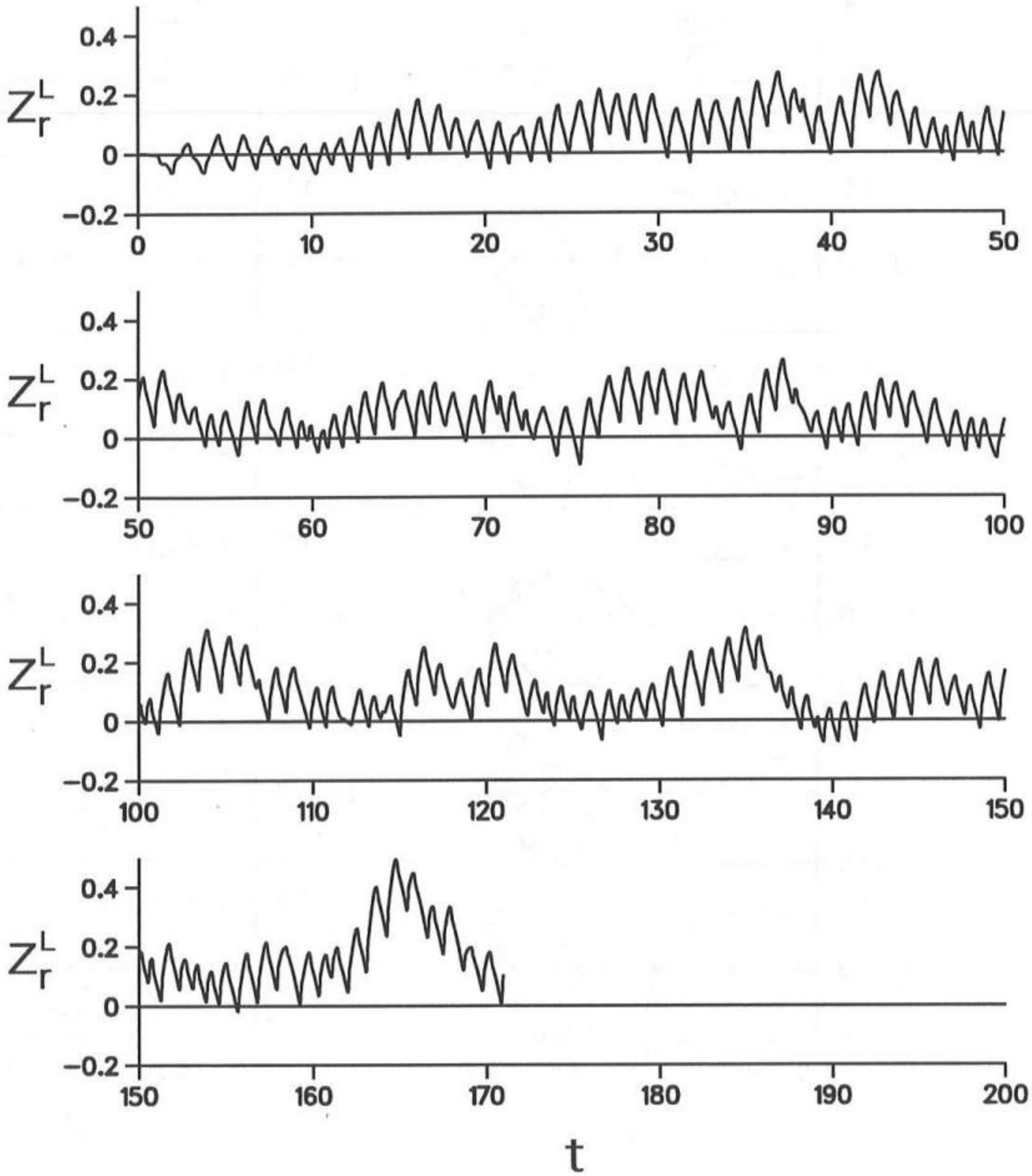


Figure B-6

Run P1

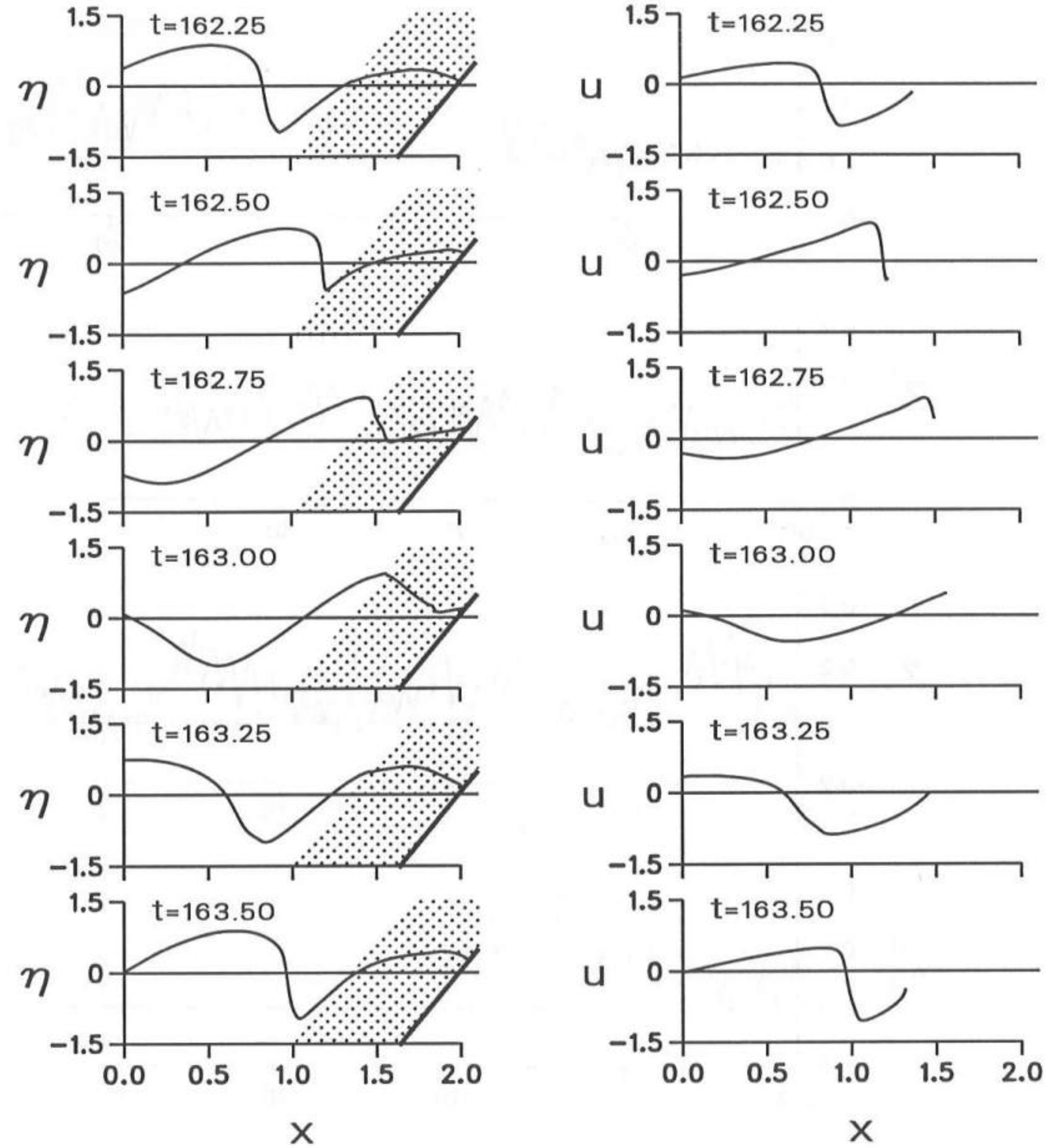


Figure B-7

Run P1

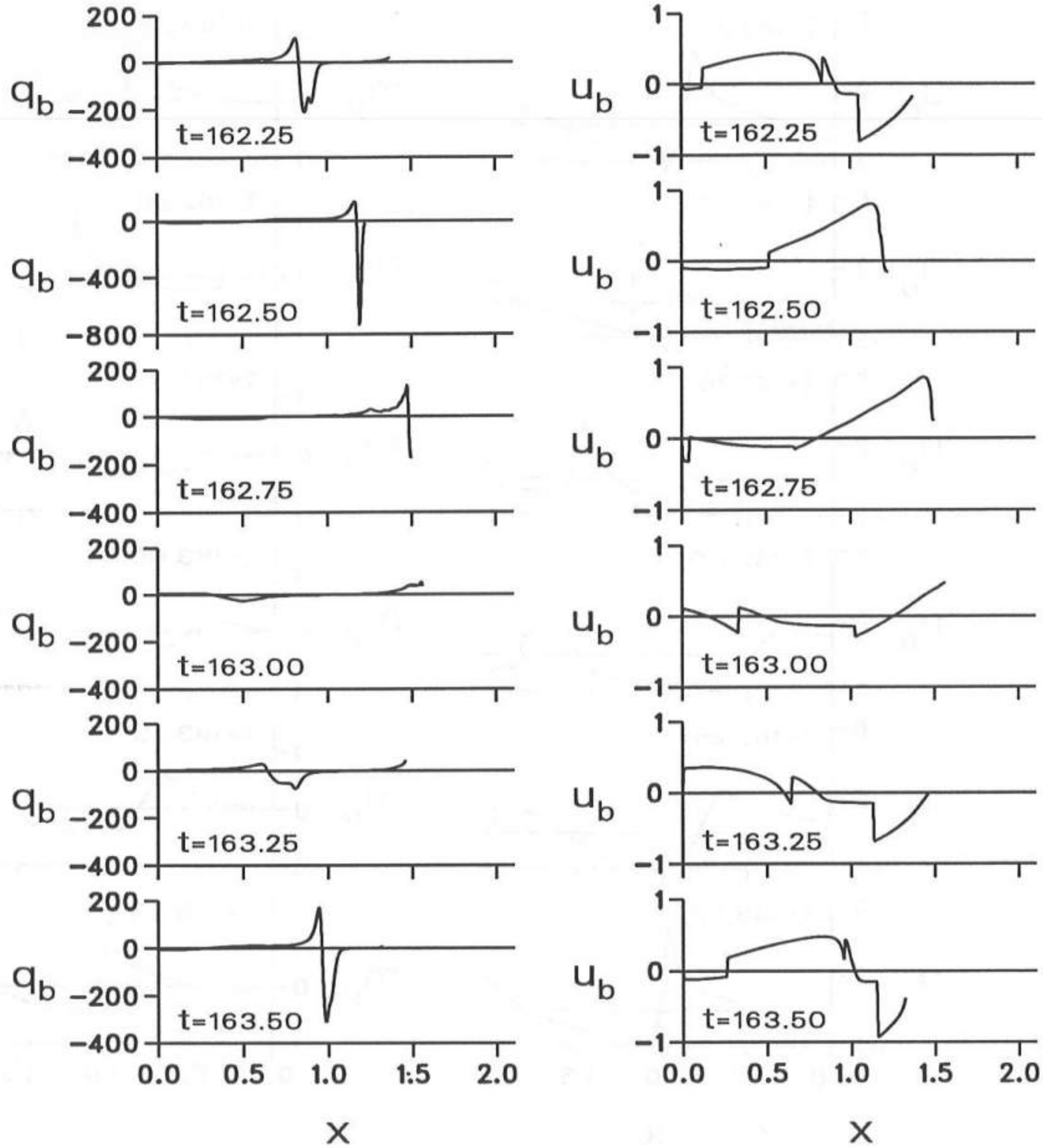


Figure B-8

Run P1

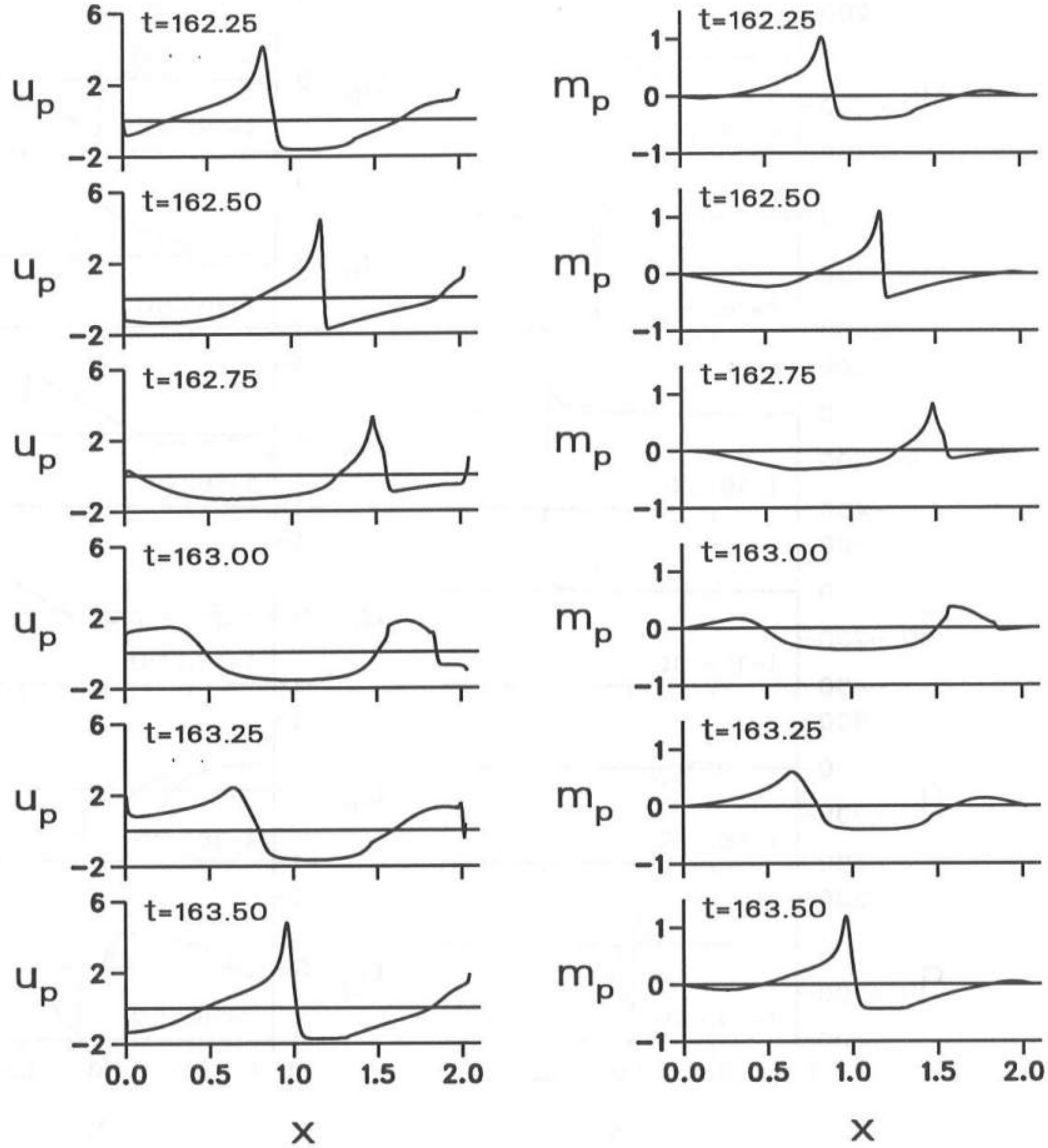


Figure B-9

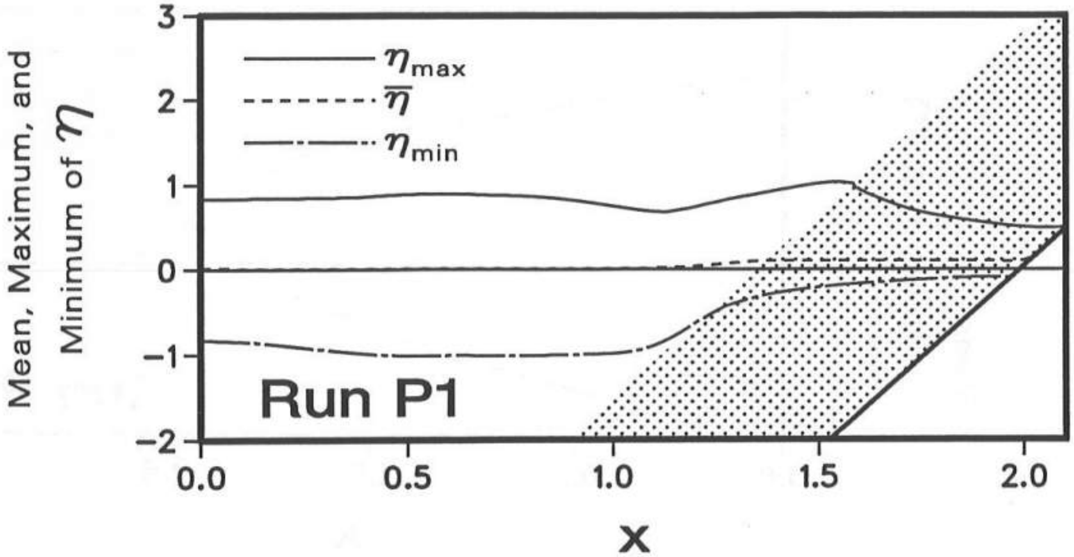


Figure B-10

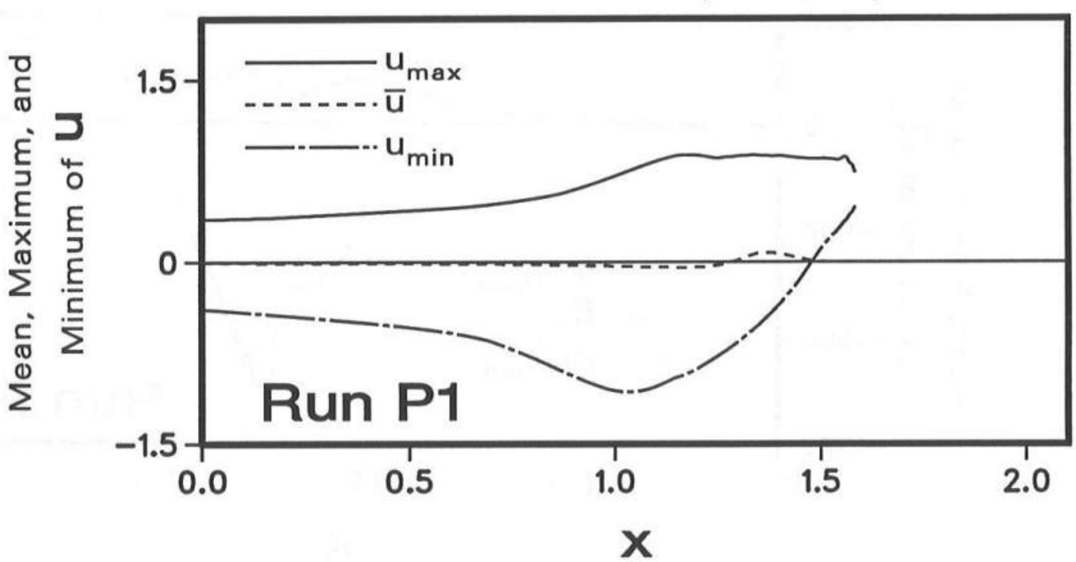


Figure B-11

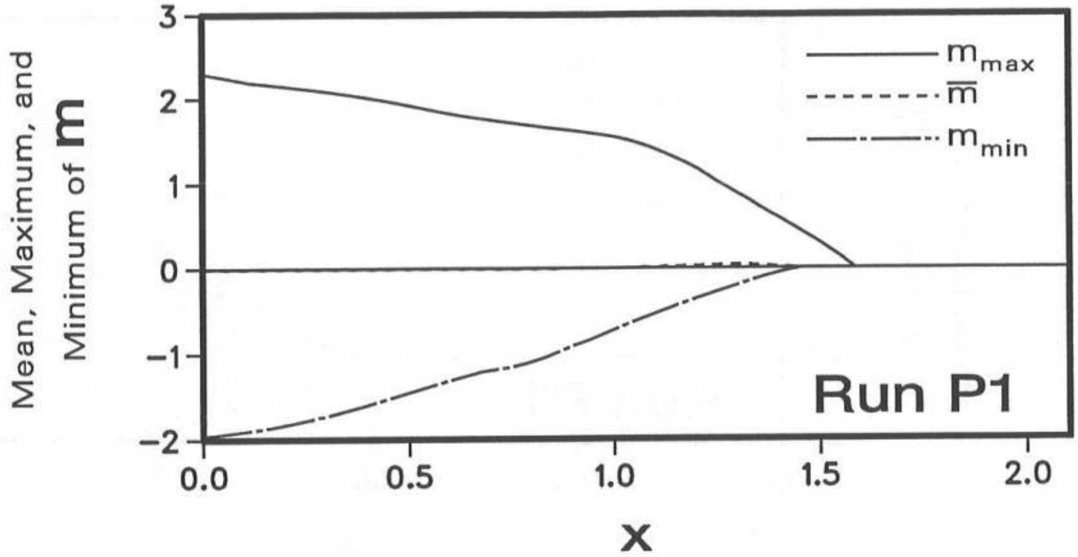


Figure B-12

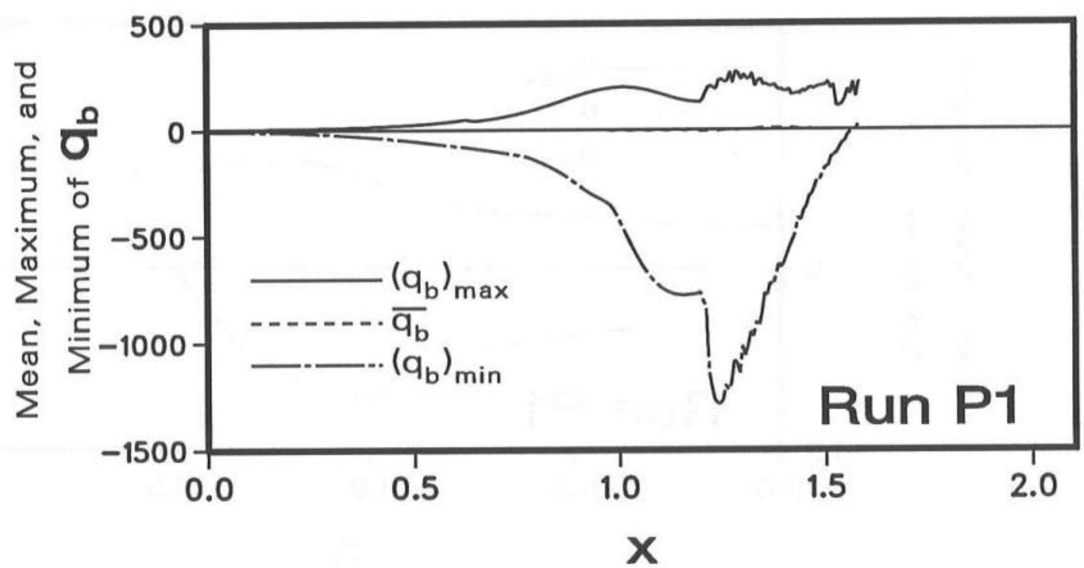


Figure B-13

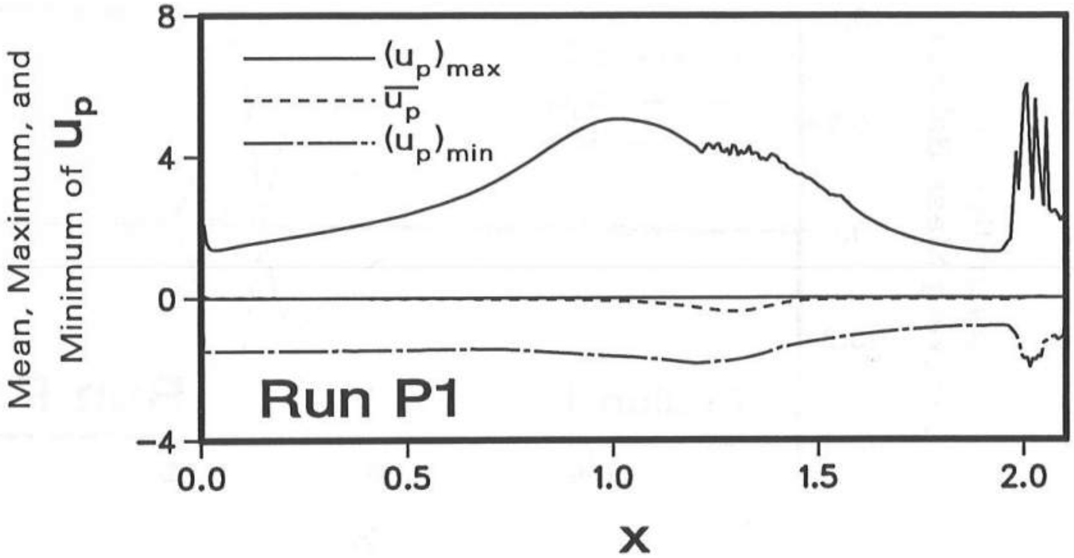


Figure B-14

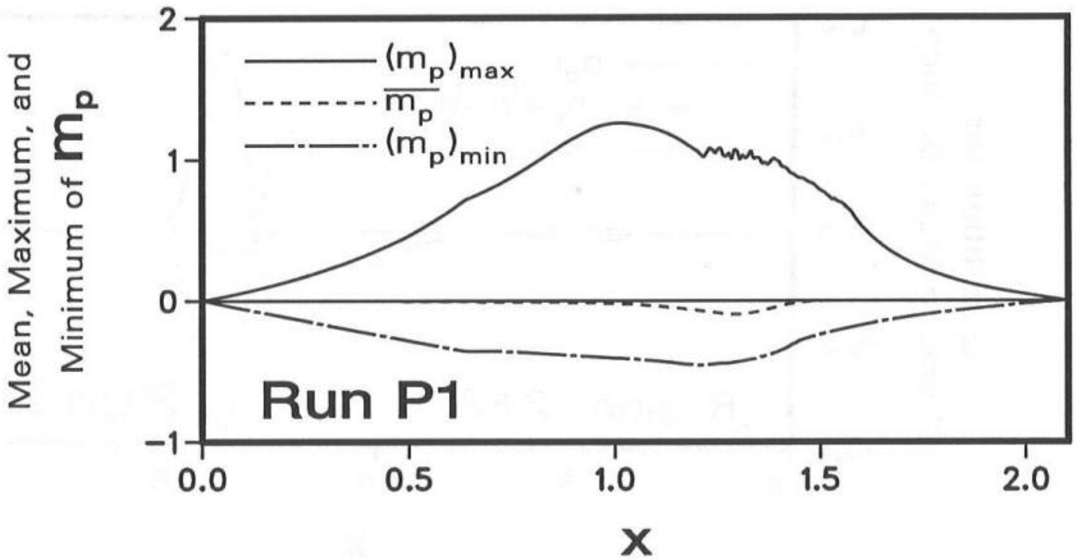


Figure B-15

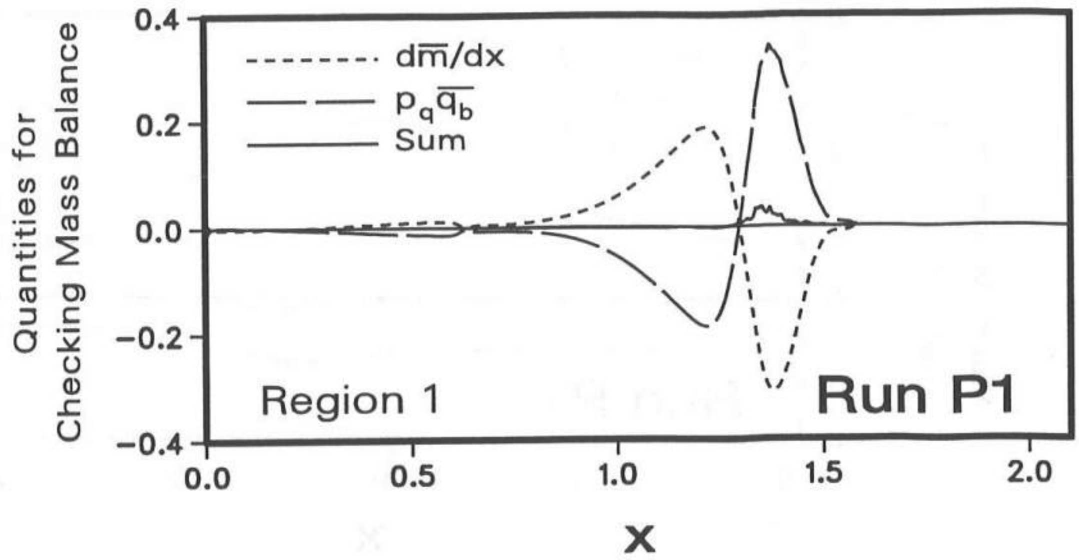


Figure B-16

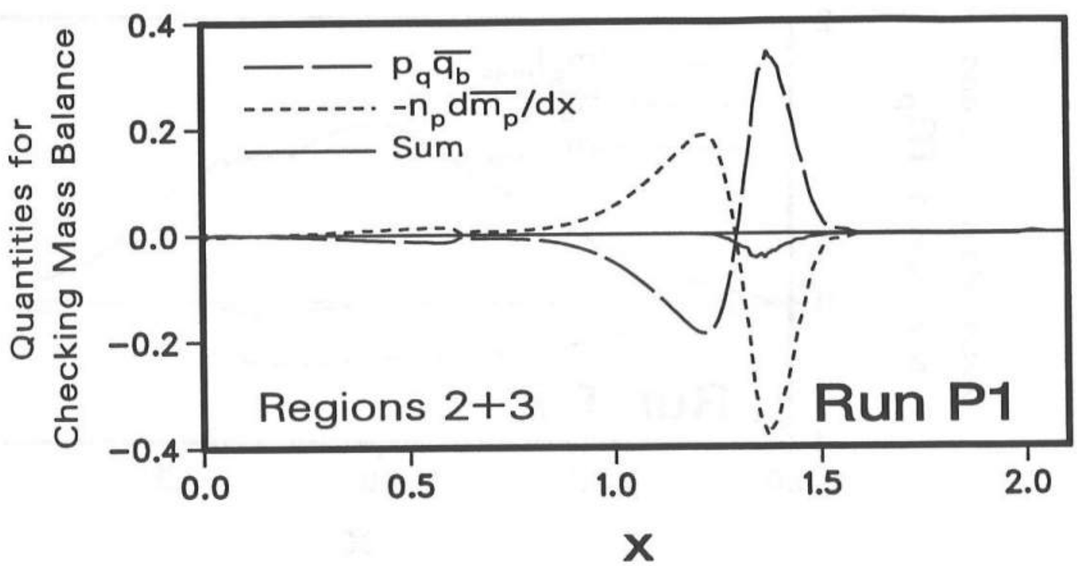


Figure B-17

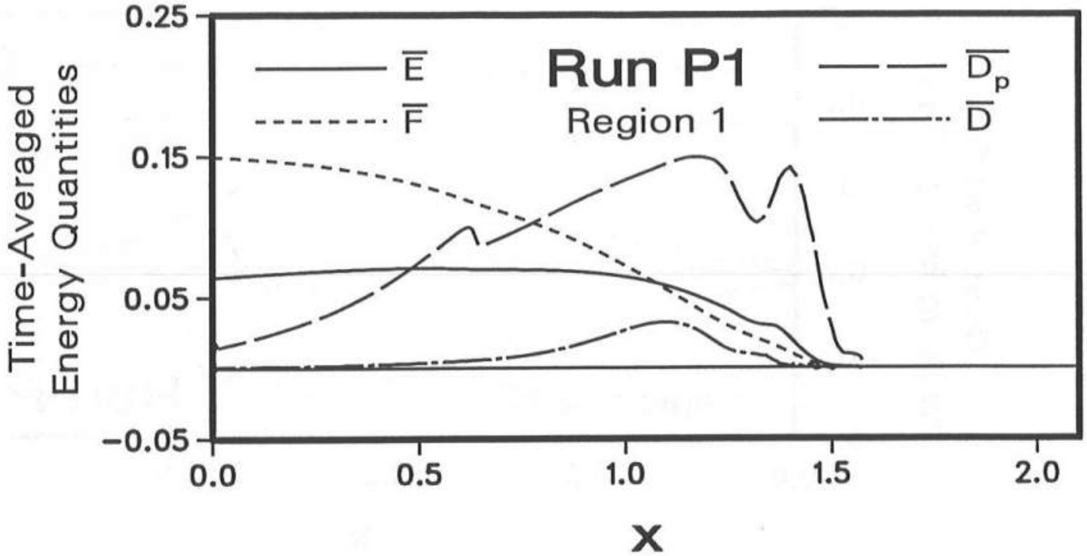


Figure B-18

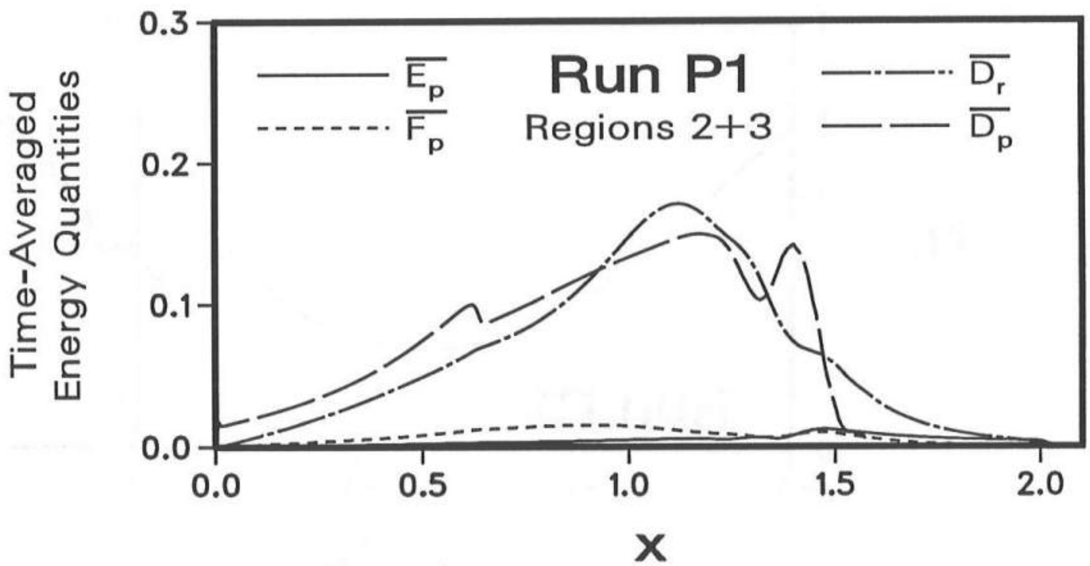


Figure B-19

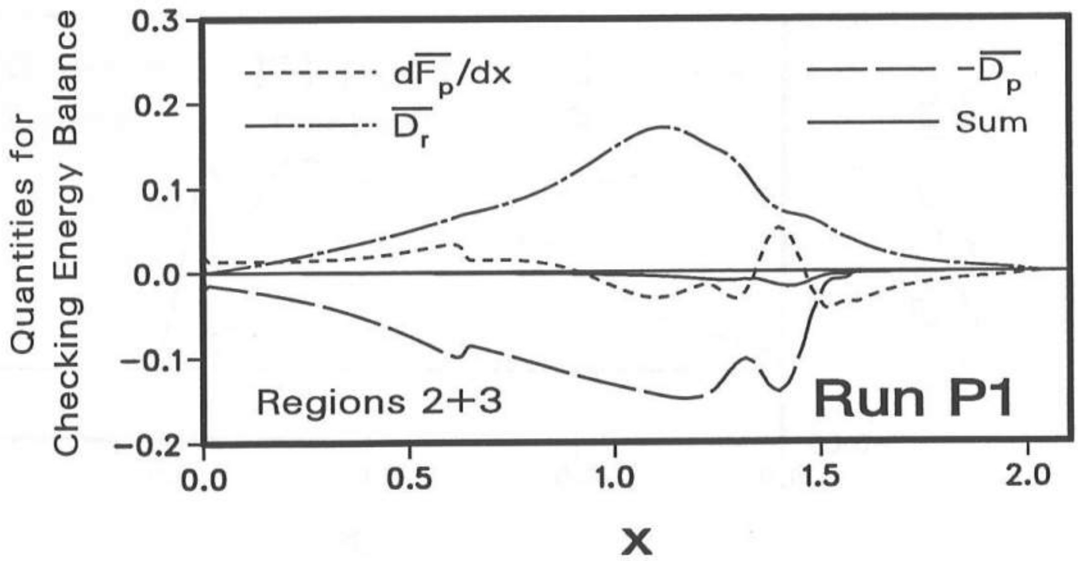


Figure B-20

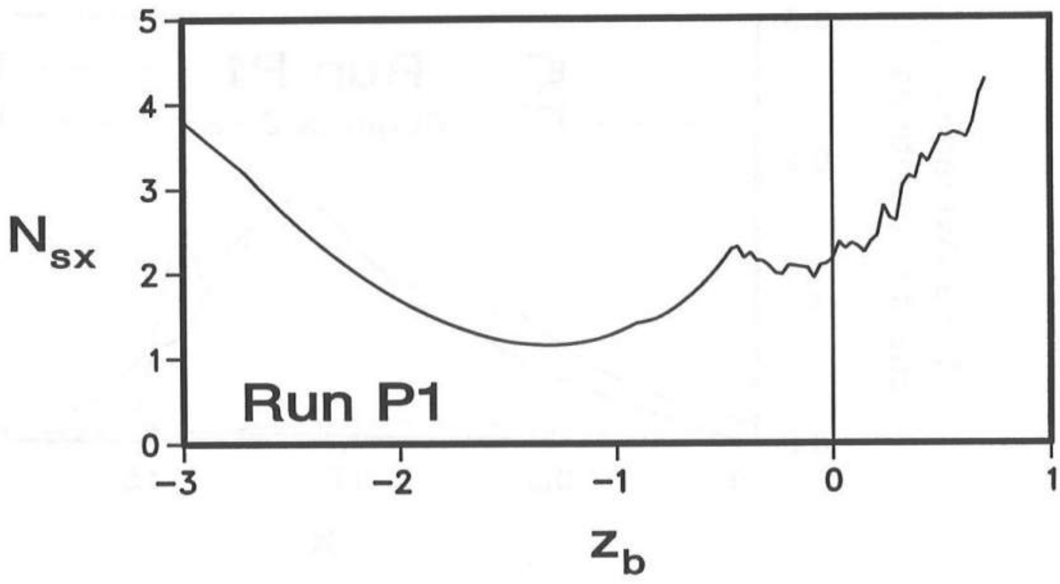


Figure B-21

Run P1

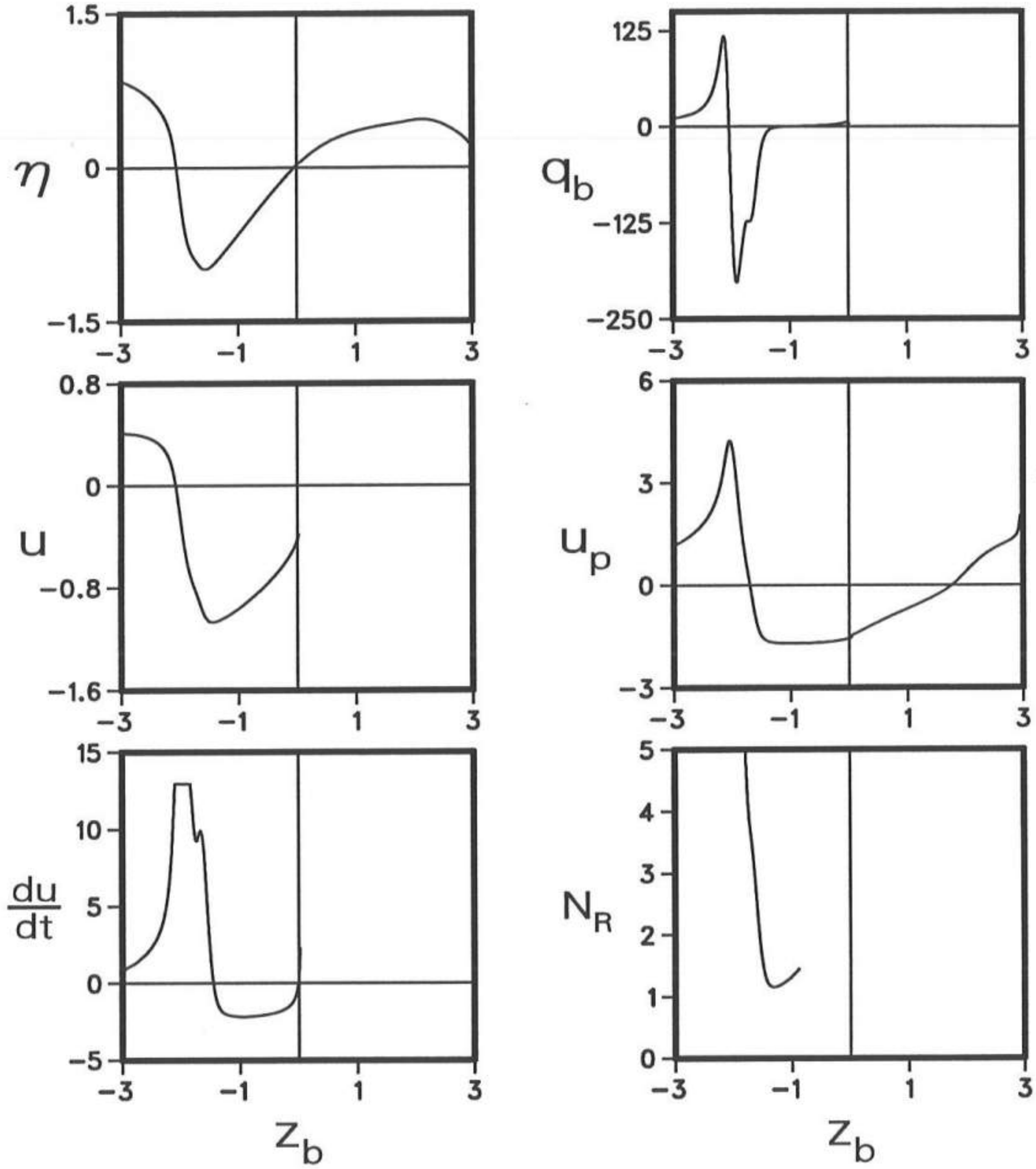


Figure B-22

APPENDIX C

EXAMPLE: RUN P2

This appendix shows the results of the PBREAK simulation for Run P2 of Kobayashi, Cox, and Wurjanto (1991). The structure for Run P2 was identical to that of Run P1 and is depicted in Figure B-1 in Appendix B. The experiment procedure was also similar.

• INPUT DATA •

Assuming that the user has become familiar with the example of Run P1 in Appendix B, in the following we only describe the input parameters and values that are different from Run P1.

- **Input wave train.** The duration of the input wave train for Run P2 was 365 seconds. The train is plotted in Figure C-1 and stored in the file `appenc.iwt` on the accompanying disk.
  - The name of the file containing the input wave train is `FINP2=tkcwp2` for this run.
  - `HREFP`=0.0535m is the significant wave height of the input wave train, whereas `TP`= 1.357s [reported as 1.36s in Kobayashi, Cox, and Wurjanto (1991)] is the mean period.
  - The normalized computation duration `TMAX` = 365s/1.357s = 268.975682.
- The *diagnostic* part of Run P2 consists of three sequences. Listed in the following are the instrumental computation parameters specified in the primary input data files for the three sequences.

Sequence	Computation Units	INITS	MULTIF	DELTAS	DELTAW	EPSI1	EPSI2
1	1-5	200	1	.001000	.001000	1.00	1.00
2	6-220	200	1	.001000	.002000	1.00	1.00
3	221-269	200	1	.001000	.002000	1.00	1.00

It is worth to mention again here that this simulation was performed on a SUN MICROSYSTEM SPARC2 machine. An identical simulation performed on a different machine may require different sequencing of the diagnostic part. The sequential procedure for this computation will be explained further in a later section under the heading "SEQUENTIAL PROCEDURE".

- MSAVA1=125, MSAVA2=130, NTIMES=4. The spatial variations of the hydrodynamic quantities η, u, q_b, u_b, u_p, and m_p are stored at the normalized time t = 124.25, 124.50, 124.75, ..., 130.00. In all, there are 24 instantaneous variations stored for each of the six quantities.

The primary input data file for the *actual* part of the PBREAK computation for Run P2 is presented in Table C-1 and stored as the file `appenc.inp` on the accompanying disk. The file ODOC containing a concise documentation produced by PBREAK for Run P2 is presented in Table C-2 and stored on the accompanying disk under the name `appenc.doc`.

• FIGURES •

Each figure presented in this appendix (Run P2) has a counterpart in Appendix B (Run P1). Since the output from PBREAK for Runs P1 and P2 is organized in the same way, and assuming the familiarity with the example of Run P1 in Appendix B, explanations on the names of variables and output files are not duplicated in this appendix. The figures presented in this appendix are listed in Table C-3.

• SEQUENTIAL PROCEDURE •

The file containing the input wave train for Run P2 is included in the accompanying disk under the name `appenc.iwt`. The computation with ICOMP=1 used the primary input data file identical to that presented in Table C-1. This computation with ICOMP=1 broke down very early in the computation unit 12 because of a remediable numerical problem IPROB=4. The last 19 lines of messages written in the file CMSG01 are shown below.

```
:: IPROB=4 from Subr. 23 MARCH3 ::
   Computation unit               M =      12
   Lower waterline node          IW =     301
   Computation parameter       INVT =    7500
   Time of occurrence          TIME =      11.404266667
   Indicators                 OMEGA =      7.58
                      ITRY, ICOUNT =      6          1
    Node          hp
     299       -0.770511D-05

:: Notice of FAILURE from Subr. 19 CINVT ::
   Computation was aborted because OMEGA exceeds 10.
   Numer. stab. indicator     OMEGA =      11.36
   Computation unit               M =      12
   Trial number                ITRY =       7
   Computation parameters      INVT =    11250
                             DELTAS =        0.001000000
                             DELTAW =        0.001000000
   Programmed Stop.
```

We thought that the cause of the problem was also too small a value of DELTAW, similar to what happened to the computation with ICOMP=1 for Run P1. For the subsequent computation with ICOMP=2, we specified DELTAW=0.002. This computation with ICOMP=2 started at the computation unit 6 (self-determined by PBREAK) and ran successfully until the computation unit 230 when the following messages appeared:

```
:: IPROB=4 from Subr. 23 MARCH3 ::
   Computation unit              M =      230
   Lower waterline node         IW =      302
   Computation parameter      INVT =     7500
   Time of occurrence         TIME =      229.075333333
   Indicators                OMEGA =        7.58
                     ITRY, ICOUNT =        6          1
    Node          hp
     300      -0.847656D-05

:: Notice of FAILURE from Subr. 19 CINVT ::
   Computation was aborted because OMEGA exceeds 10.
   Numer. stab. indicator    OMEGA =         11.36
   Computation unit              M =      230
   Trial number               ITRY =        7
   Computation parameters     INVT =    11250
                            DELTAS =          0.001000000
                            DELTAW =          0.001301000
   Programmed Stop.
```

Thus, we had to execute another computation with ICOMP=2, which was to start at the computation unit 221. We learned from the file CDOC02 that the value of DELTAW in the sequence 2 had been reduced from 0.002 to 0.001301 during the computation unit 137 (see page 216). Again, we thought that the cause of the problem was too small a value of DELTAW. For the subsequent computation with ICOMP=2 for the sequence 3, we specified DELTAW=0.002 again, which was larger than DELTAW=0.001301 when IPROB=4 occurred at M=230 as well as at the end of the computation unit 220. Thus, the primary input data file for the sequences 2 and 3 were in fact identical. The sequence 3 successfully reached the normalized terminal time t_{max}, and thus completed the diagnostic part of Run P2. The actual part with ICOMP=3 was then computed using the primary input data file presented in Table C-1.

Table C-1: Primary input data file for PBREAK computation of Run P2.

```
        4                     --> NLINES
---------------------------------------------------------------------
              PBREAK          *** Kobayashi, Cox, and Wurjanto 1991
                              *** Run P2
---------------------------------------------------------------------
1                             --> ISYST
1                             --> IBOT
      125      130        4   --> MSAVA1,MSAVA2,NTIMES
        5                     --> NNODB
        1       51      101   --> NODB(1)-(2)-(3)
      151      201            --> NODB(4)-(5)
tkcwp2                        --> FINP2
    268.975682                --> TMAX(normalized)
------ Block of instrumental computation parameters begins
      200                     --> INITS
        1                     --> MULTIF
       .001000      .001000   --> DELTAS,DELTAW(normalized)
      1.000000     1.000000   --> EPSI1,EPSI2
------ Block ends
        0                     --> MSTAT
       50                     --> NRATE
        4                     --> NDELR
     21.000000                --> DELRP(1)(millimeters)
     27.500000                -->      (2)
     15.000000                -->      (3)
      5.000000                -->      (4)
       .053500     1.357000   --> HREFP(meters),TP(seconds)
       .050000                --> FWP
       .400000                --> DSEAP(meters)
       .333333                --> TSLOPS
        1                     --> NUSEG
      2.400000      .333333   --> WUSEG(1)(meters),TUSEG(1)
        2                     --> NLSEG
       .566049      .000000   --> WLSEG(1)(meters),TLSEG(1)
      1.835000      .333333   -->      (2)              (2)
------ Block of permeable underlayer data begins
0                             --> ISALT
     20.000000                --> TEMPER(celcius)
       .021000                --> DPP(meters)
       .480000                --> PORO
   1140.000000     2.700000   --> ALPA0,BETA0
------ Block ends
------ Block of armor stability data begins
      2.700000     1.191754   --> SG,TANPHI
       .900000      .600000   --> C2,C3
       .500000      .300000   --> CD,CL
      1.500000                --> CM
      1.000000     -.800000   --> AMAX,AMIN
------ Block ends
```

Table C-2: File ODOC produced by PBREAK computation of Run P2.

```
-------------------------------------------------------------------
          PBREAK          *** Kobayashi, Cox, and Wurjanto 1991
                          *** Run P2
-------------------------------------------------------------------

WAVE CONDITION

Incident wave at seaward boundary given as input
Number of data points    NDATA =    7301
Norm. maximum surface elev.    =       0.797500
Norm. minimum surface elev.    =      -0.798600
Reference wave period          =       1.357000 sec.
Reference wave height          =       0.053500 meters
Depth at seaward boundary      =       0.400000 meters
Norm. depth at seaw. boundary  =       7.477
Normalized wave length         =       5.740
Ursell number                  =       4.407
Surf similarity parameter      =       2.444
"Sigma"                        =      18.375

PROPERTIES OF THE UPPER SLOPE

Friction factor            =    0.050000
Norm. friction factor      =    0.459386
Number of segments         =    1
------------------------------------------------------
 SEGMENT    WUSEG(I)   TUSEG(I)   XUSEG(I)   ZUSEG(I)
       I     meters                 meters     meters
------------------------------------------------------
       1   2.400000   0.333333   0.000000  -0.400000
       2                         2.400000   0.399999
------------------------------------------------------

GEOMETRY OF THE LOWER SLOPE

Number of Segments         =    2
------------------------------------------------------
 SEGMENT    WLSEG(I)   TLSEG(I)   XLSEG(I)   ZLSEG(I)
       I     meters                 meters     meters
------------------------------------------------------
       1   0.566049   0.000000   0.000000  -0.400000
       2   1.835000   0.333333   0.566049  -0.400000
       3                         2.401049   0.211666
------------------------------------------------------

PROPERTIES OF THE PERMEABLE UNDERLAYER and
PARAMETERS RELATED TO PERMEABILITY

Fresh water at temperature          TEMPER =           20.000 degrees C
Kinematic viscosity of water x 1.D+06
                        = 1.D+06 x VISCO =            1.004000 m.m/sec
Representative diameter of permeable
                underlayer materials  DPP =            0.021 meters
Porosity of the permeable underlayer  PORO =           0.480
```

```
Dimensionless constants                 ALPA0 =        1140.000
                                        BETA0 =           2.700
Coefficient expressing the laminar flow resistance
                      ALPAP [alpha prime] =               1.584
Coefficient expressing the turbulent flow resistance
                       BETAP [beta prime] =             604.539
Dimensionless parameters that arise in consequence
    of normalizing the governing equations:
                                        PARPU =           0.854577D-01
                       PARPQ = PORO*PARPU =               0.410197D-01
                                        PARMU =           0.881657D-01

INPUT PARAMETERS FOR ARMOR STABILITY

Specific gravity           SG =     2.700
Armor friction Factor  TANPHI =     1.192
Area coefficient           C2 =     0.900
Volume coefficient         C3 =     0.600
Drag coefficient           CD =     0.500
Lift coefficient           CL =     0.300
Inertia coefficient        CM =     1.500
Upper and lower bounds of du/dt
        (normalized by gravity)
                         AMAX =     1.000
                         AMIN =    -0.800

COMPUTATION PARAMETERS

Total number of spatial nodes        JE =      400
No. of nodes with upper slope below SWL
                                  INITS =      200
No. of nodes with lower slope below SWL
                                  INITW =      295
Minimum allowable value of INVT
                                = MINVT =     1000
  where INVT is the number of time steps
  in one reference wave period
Computed time series are stored at rate NRATE
  per reference wave period with NRATE =        50
Normalized delta x          X =         0.610324D-02
Computation duration  TMAX =          268.975682 ref. wave periods
Statistical calculations are performed excluding
     the first MSTAT computation units
     with                         MSTAT =        0
Instantaneous spatial variations are stored NTIMES times
     at equal intervals per reference wave period from
     computation unit MSAVA1 to MSAVA2, inclusive,
     with                        NTIMES =        4
                                 MSAVA1 =      125
                                 MSAVA2 =      130
Other parameters are given in COMPUTATION LOG below
Positioning:
. Node        1         is at x =       0.000000
. Node      200 (INITS) is at x =       1.214545
. Node      295 (INITW) is at x =       1.794353
. Node      400 (JE)    is at x =       2.435193
```

```
COMPUTATION LOG

MULTIF = integer multiplication factor
MULTIF*MINVT = minimum value of INVT used in a sequence
INVT   = number of time steps in one reference wave period
MINVT  = minimum allowable value of INVT
EPSI1, EPSI2 = damping coefficients
OMEGA  = numerical stability indicator
NDIV   = time level in a computation unit when DELTAS or
         DELTAW changes value
DELTAS = normalized water depth defining the computational
         upper waterline
DELTAW = normalized water depth defining the computational
         lower waterline

--------------------------------------------------------
  Sequence   Computation  MULTIF     EPSI1      EPSI2
   Number       Units
--------------------------------------------------------
        1             1        1      1.00       1.00
              to      5
        2             6        1      1.00       1.00
              to    220
        3           221        1      1.00       1.00
              to    269
--------------------------------------------------------

------------------------------------------------------
   Comp.   INVT   OMEGA    NDIV   DELTAS    DELTAW
   Unit
------------------------------------------------------
       1   1000     1.0    1000  0.001000  0.001000
       2   1000     1.0    1000  0.001000  0.001000
       3   1000     1.0    1000  0.001000  0.001000
       4   1000     1.0    1000  0.001000  0.001000
       5   1000     1.0    1000  0.001000  0.001000
       6   1000     1.0    1000  0.001000  0.002000
       7   1000     1.0    1000  0.001000  0.002000
       8   1000     1.0    1000  0.001000  0.002000
       9   1000     1.0    1000  0.001000  0.002000
      10   1000     1.0    1000  0.001000  0.002000
      11   1000     1.0    1000  0.001000  0.002000
      12   1000     1.0    1000  0.001000  0.002000
      13   1000     1.0    1000  0.001000  0.002000
      14   1000     1.0    1000  0.001000  0.002000
      15   1000     1.0    1000  0.001000  0.002000
      16   1000     1.0    1000  0.001000  0.002000
      17   1000     1.0    1000  0.001000  0.002000
      18   1000     1.0    1000  0.001000  0.002000
      19   1000     1.0    1000  0.001000  0.002000
      20   1000     1.0    1000  0.001000  0.002000
      21   1000     1.0    1000  0.001000  0.002000
      22   1000     1.0    1000  0.001000  0.002000
      23   1000     1.0    1000  0.001000  0.002000
      24   1000     1.0    1000  0.001000  0.002000
```

Table C-2 continued.

167	25	1000	1.0	1000	0.001000	0.002000
168	26	1000	1.0	1000	0.001000	0.002000
169	27	1000	1.0	1000	0.001000	0.002000
170	28	1000	1.0	1000	0.001000	0.002000
171	29	1000	1.0	1000	0.001000	0.002000
172	30	1000	1.0	1000	0.001000	0.002000
173	31	1000	1.0	1000	0.001000	0.002000
174	32	1000	1.0	1000	0.001000	0.002000
175	33	1000	1.0	1000	0.001000	0.002000
176	34	1000	1.0	1000	0.001000	0.002000
177	35	1000	1.0	1000	0.001000	0.002000
178	36	1000	1.0	1000	0.001000	0.002000
179	37	1000	1.0	1000	0.001000	0.002000
180	38	1000	1.0	1000	0.001000	0.002000
181	39	1000	1.0	1000	0.001000	0.002000
182	40	1000	1.0	1000	0.001000	0.002000
183	41	1000	1.0	1000	0.001000	0.002000
184	42	1000	1.0	1000	0.001000	0.002000
185	43	1000	1.0	1000	0.001000	0.002000
186	44	1000	1.0	1000	0.001000	0.002000
187	45	1000	1.0	1000	0.001000	0.002000
188	46	1000	1.0	1000	0.001000	0.002000
189	47	1000	1.0	1000	0.001000	0.002000
190	48	1000	1.0	1000	0.001000	0.002000
191	49	1000	1.0	1000	0.001000	0.002000
192	50	1000	1.0	1000	0.001000	0.002000
193	51	1000	1.0	1000	0.001000	0.002000
194	52	1000	1.0	1000	0.001000	0.002000
195	53	1000	1.0	1000	0.001000	0.002000
196	54	1000	1.0	1000	0.001000	0.002000
197	55	1000	1.0	1000	0.001000	0.002000
198	56	1000	1.0	1000	0.001000	0.002000
199	57	1000	1.0	1000	0.001000	0.002000
200	58	1000	1.0	1000	0.001000	0.002000
201	59	1000	1.0	1000	0.001000	0.002000
202	60	1000	1.0	1000	0.001000	0.002000
203	61	1000	1.0	1000	0.001000	0.002000
204	62	1000	1.0	1000	0.001000	0.002000
205	63	1000	1.0	1000	0.001000	0.002000
206	64	1000	1.0	1000	0.001000	0.002000
207	65	1000	1.0	1000	0.001000	0.002000
208	66	1000	1.0	1000	0.001000	0.002000
209	67	1000	1.0	1000	0.001000	0.002000
210	68	1000	1.0	1000	0.001000	0.002000
211	69	1000	1.0	1000	0.001000	0.002000
212	70	1000	1.0	1000	0.001000	0.002000
213	71	1000	1.0	1000	0.001000	0.002000
214	72	1000	1.0	1000	0.001000	0.002000
215	73	1000	1.0	1000	0.001000	0.002000
216	74	1000	1.0	1000	0.001000	0.002000
217	75	1000	1.0	1000	0.001000	0.002000
218	76	1000	1.0	1000	0.001000	0.002000
219	77	1000	1.0	1000	0.001000	0.002000
220	78	1000	1.0	1000	0.001000	0.002000
221	79	1000	1.0	1000	0.001000	0.002000
222	80	1000	1.0	1000	0.001000	0.002000

Table C-2 continued.

223	81	1000	1.0	1000	0.001000	0.002000
224	82	1000	1.0	1000	0.001000	0.002000
225	83	1000	1.0	1000	0.001000	0.002000
226	84	1000	1.0	1000	0.001000	0.002000
227	85	1000	1.0	1000	0.001000	0.002000
228	86	1000	1.0	1000	0.001000	0.002000
229	87	1000	1.0	1000	0.001000	0.002000
230	88	1000	1.0	1000	0.001000	0.002000
231	89	1000	1.0	1000	0.001000	0.002000
232	90	1000	1.0	1000	0.001000	0.002000
233	91	1000	1.0	1000	0.001000	0.002000
234	92	1000	1.0	1000	0.001000	0.002000
235	93	1000	1.0	1000	0.001000	0.002000
236	94	1000	1.0	1000	0.001000	0.002000
237	95	1000	1.0	1000	0.001000	0.002000
238	96	1000	1.0	1000	0.001000	0.002000
239	97	1000	1.0	1000	0.001000	0.002000
240	98	1000	1.0	1000	0.001000	0.002000
241	99	1000	1.0	1000	0.001000	0.002000
242	100	1000	1.0	1000	0.001000	0.002000
243	101	1000	1.0	1000	0.001000	0.002000
244	102	1000	1.0	1000	0.001000	0.002000
245	103	1000	1.0	1000	0.001000	0.002000
246	104	1000	1.0	1000	0.001000	0.002000
247	105	1000	1.0	1000	0.001000	0.002000
248	106	1000	1.0	1000	0.001000	0.002000
249	107	1000	1.0	1000	0.001000	0.002000
250	108	1000	1.0	1000	0.001000	0.002000
251	109	1000	1.0	1000	0.001000	0.002000
252	110	1000	1.0	1000	0.001000	0.002000
253	111	1000	1.0	1000	0.001000	0.002000
254	112	1000	1.0	1000	0.001000	0.002000
255	113	1000	1.0	1000	0.001000	0.002000
256	114	1000	1.0	1000	0.001000	0.002000
257	115	1000	1.0	1000	0.001000	0.002000
258	116	1000	1.0	1000	0.001000	0.002000
259	117	1000	1.0	1000	0.001000	0.002000
260	118	1000	1.0	1000	0.001000	0.002000
261	119	1000	1.0	1000	0.001000	0.002000
262	120	1000	1.0	1000	0.001000	0.002000
263	121	1000	1.0	1000	0.001000	0.002000
264	122	1000	1.0	1000	0.001000	0.002000
265	123	1000	1.0	1000	0.001000	0.002000
266	124	1000	1.0	1000	0.001000	0.002000
267	125	1000	1.0	1000	0.001000	0.002000
268	126	1000	1.0	1000	0.001000	0.002000
269	127	1000	1.0	1000	0.001000	0.002000
270	128	1000	1.0	1000	0.001000	0.002000
271	129	1000	1.0	1000	0.001000	0.002000
272	130	1000	1.0	1000	0.001000	0.002000
273	131	1000	1.0	1000	0.001000	0.002000
274	132	1000	1.0	1000	0.001000	0.002000
275	133	1000	1.0	1000	0.001000	0.002000
276	134	1000	1.0	1000	0.001000	0.002000
277	135	1000	1.0	1000	0.001000	0.002000

Table C-2 continued.

278	136	1000	1.0	1000	0.001000	0.002000
279	137	1000	1.0	725	0.001000	0.002000
280				1000	0.001000	0.001301
281	138	1000	1.0	1000	0.001000	0.001301
282	139	1000	1.0	1000	0.001000	0.001301
283	140	1000	1.0	1000	0.001000	0.001301
284	141	1000	1.0	1000	0.001000	0.001301
285	142	1000	1.0	1000	0.001000	0.001301
286	143	1000	1.0	1000	0.001000	0.001301
287	144	1000	1.0	1000	0.001000	0.001301
288	145	1000	1.0	1000	0.001000	0.001301
289	146	1000	1.0	1000	0.001000	0.001301
290	147	1000	1.0	1000	0.001000	0.001301
291	148	1000	1.0	1000	0.001000	0.001301
292	149	1000	1.0	1000	0.001000	0.001301
293	150	1000	1.0	1000	0.001000	0.001301
294	151	1000	1.0	1000	0.001000	0.001301
295	152	1000	1.0	1000	0.001000	0.001301
296	153	1000	1.0	1000	0.001000	0.001301
297	154	1000	1.0	1000	0.001000	0.001301
298	155	1000	1.0	1000	0.001000	0.001301
299	156	1000	1.0	1000	0.001000	0.001301
300	157	1000	1.0	1000	0.001000	0.001301
301	158	1000	1.0	1000	0.001000	0.001301
302	159	1000	1.0	1000	0.001000	0.001301
303	160	1000	1.0	1000	0.001000	0.001301
304	161	1000	1.0	1000	0.001000	0.001301
305	162	1000	1.0	1000	0.001000	0.001301
306	163	1000	1.0	1000	0.001000	0.001301
307	164	1000	1.0	1000	0.001000	0.001301
308	165	1000	1.0	1000	0.001000	0.001301
309	166	1000	1.0	1000	0.001000	0.001301
310	167	1000	1.0	1000	0.001000	0.001301
311	168	1000	1.0	1000	0.001000	0.001301
312	169	1000	1.0	1000	0.001000	0.001301
313	170	1000	1.0	1000	0.001000	0.001301
314	171	1000	1.0	1000	0.001000	0.001301
315	172	1000	1.0	1000	0.001000	0.001301
316	173	1000	1.0	1000	0.001000	0.001301
317	174	1000	1.0	1000	0.001000	0.001301
318	175	1000	1.0	1000	0.001000	0.001301
319	176	1000	1.0	1000	0.001000	0.001301
320	177	1000	1.0	1000	0.001000	0.001301
321	178	1000	1.0	1000	0.001000	0.001301
322	179	1000	1.0	1000	0.001000	0.001301
323	180	1000	1.0	1000	0.001000	0.001301
324	181	1000	1.0	1000	0.001000	0.001301
325	182	1000	1.0	1000	0.001000	0.001301
326	183	1000	1.0	1000	0.001000	0.001301
327	184	1000	1.0	1000	0.001000	0.001301
328	185	1000	1.0	1000	0.001000	0.001301
329	186	1000	1.0	1000	0.001000	0.001301
330	187	1000	1.0	1000	0.001000	0.001301
331	188	1000	1.0	1000	0.001000	0.001301
332	189	1000	1.0	1000	0.001000	0.001301
333	190	1000	1.0	1000	0.001000	0.001301

Table C-2 continued.

334	191	1000	1.0	1000	0.001000	0.001301
335	192	1000	1.0	1000	0.001000	0.001301
336	193	1000	1.0	1000	0.001000	0.001301
337	194	1000	1.0	1000	0.001000	0.001301
338	195	1000	1.0	1000	0.001000	0.001301
339	196	1000	1.0	1000	0.001000	0.001301
340	197	1000	1.0	1000	0.001000	0.001301
341	198	1000	1.0	1000	0.001000	0.001301
342	199	1000	1.0	1000	0.001000	0.001301
343	200	1000	1.0	1000	0.001000	0.001301
344	201	1000	1.0	1000	0.001000	0.001301
345	202	1000	1.0	1000	0.001000	0.001301
346	203	1000	1.0	1000	0.001000	0.001301
347	204	1000	1.0	1000	0.001000	0.001301
348	205	1000	1.0	1000	0.001000	0.001301
349	206	1000	1.0	1000	0.001000	0.001301
350	207	1000	1.0	1000	0.001000	0.001301
351	208	1000	1.0	1000	0.001000	0.001301
352	209	1000	1.0	1000	0.001000	0.001301
353	210	1000	1.0	1000	0.001000	0.001301
354	211	1000	1.0	1000	0.001000	0.001301
355	212	1000	1.0	1000	0.001000	0.001301
356	213	1000	1.0	1000	0.001000	0.001301
357	214	1000	1.0	1000	0.001000	0.001301
358	215	1000	1.0	1000	0.001000	0.001301
359	216	1000	1.0	1000	0.001000	0.001301
360	217	1000	1.0	1000	0.001000	0.001301
361	218	1000	1.0	1000	0.001000	0.001301
362	219	1000	1.0	1000	0.001000	0.001301
363	220	1000	1.0	1000	0.001000	0.001301
364	221	1000	1.0	1000	0.001000	0.002000
365	222	1000	1.0	1000	0.001000	0.002000
366	223	1000	1.0	1000	0.001000	0.002000
367	224	1000	1.0	1000	0.001000	0.002000
368	225	1000	1.0	1000	0.001000	0.002000
369	226	1000	1.0	1000	0.001000	0.002000
370	227	1000	1.0	1000	0.001000	0.002000
371	228	1000	1.0	1000	0.001000	0.002000
372	229	1000	1.0	1000	0.001000	0.002000
373	230	1000	1.0	1000	0.001000	0.002000
374	231	1000	1.0	1000	0.001000	0.002000
375	232	1000	1.0	1000	0.001000	0.002000
376	233	1000	1.0	1000	0.001000	0.002000
377	234	1000	1.0	1000	0.001000	0.002000
378	235	1000	1.0	1000	0.001000	0.002000
379	236	1000	1.0	1000	0.001000	0.002000
380	237	1000	1.0	1000	0.001000	0.002000
381	238	1000	1.0	1000	0.001000	0.002000
382	239	1000	1.0	1000	0.001000	0.002000
383	240	1000	1.0	1000	0.001000	0.002000
384	241	1000	1.0	1000	0.001000	0.002000
385	242	1000	1.0	1000	0.001000	0.002000
386	243	1000	1.0	1000	0.001000	0.002000
387	244	1000	1.0	1000	0.001000	0.002000
388	245	1000	1.0	1000	0.001000	0.002000
389	246	1000	1.0	1000	0.001000	0.002000

Table C-2 continued.

```
     247  1000    1.0    1000  0.001000  0.002000
     248  1000    1.0    1000  0.001000  0.002000
     249  1000    1.0    1000  0.001000  0.002000
     250  1000    1.0    1000  0.001000  0.002000
     251  1000    1.0    1000  0.001000  0.002000
     252  1000    1.0    1000  0.001000  0.002000
     253  1000    1.0    1000  0.001000  0.002000
     254  1000    1.0    1000  0.001000  0.002000
     255  1000    1.0    1000  0.001000  0.002000
     256  1000    1.0    1000  0.001000  0.002000
     257  1000    1.0    1000  0.001000  0.002000
     258  1000    1.0    1000  0.001000  0.002000
     259  1000    1.0    1000  0.001000  0.002000
     260  1000    1.0    1000  0.001000  0.002000
     261  1000    1.0    1000  0.001000  0.002000
     262  1000    1.0    1000  0.001000  0.002000
     263  1000    1.0    1000  0.001000  0.002000
     264  1000    1.0    1000  0.001000  0.002000
     265  1000    1.0    1000  0.001000  0.002000
     266  1000    1.0    1000  0.001000  0.002000
     267  1000    1.0    1000  0.001000  0.002000
     268  1000    1.0    1000  0.001000  0.002000
     269  1000    1.0     975  0.001000  0.002000
-----------------------------------------------------------

STATISTICAL CALCULATIONS

Statistical calculations begins at    t=TSTAT1 =       0.000000
Statistical calculations ends at      t=TSTAT2 =     268.975000
Duration of statistical calculations     TSPAN =     268.975000
Computational waterlines at t=TSTAT1:
                         . Upper slope: IST1 =     200
                         . Lower slope: IWT1 =     295
Computational waterlines at t=TSTAT2:
                         . Upper slope: IST2 =     199
                         . Lower slope: IWT2 =     301

ELEVATIONS AT SEAWARD BOUNDARY

Time-averaged elevation due to incident waves =       0.000001
Elevations due to reflected waves:
                             . Time-averaged =       0.011220
                             . Maximum       =       0.145005
                             . Minimum       =      -0.157810
```

Table C-2 continued.

```
RUN-UP, RUN-DOWN, SET-UP

Smallest nodes reached by computational waterlines:
                         . Upper slope: ISMIN =     174
                         . Lower slope: IWMIN =     288
Largest nodes reached by computational waterlines:
                         . Upper slope: ISMAX =     230
                         . Lower slope: IWMAX =     309
------------------------------------------------------------
 Slope    i   Deltar(i)    Setup(i)    Runup(i)  Rundown(i)
                [mm]
------------------------------------------------------------
 Upper    1    21.000       0.053       1.131      -1.191
          2    27.500       0.047       1.114      -1.188
          3    15.000       0.059       1.139      -1.177
          4     5.000       0.070       1.126      -1.073
------------------------------------------------------------
 Lower    1    21.000       0.096       0.497      -0.246
          2    27.500       0.096       0.500      -0.249
          3    15.000       0.097       0.497      -0.245
          4     5.000       0.097       0.503      -0.247
------------------------------------------------------------

ARMOR STABILITY

Largest node number for which armor stability
     computation is performed   JSTABM =     220
Critical stability number Nsc     SNSC =       1.485
  occurring at normalized time   TSNSC =     197.657000
                at node number   JSNSC =     165
                      x at JSNSC =       1.000931
                     zb at JSNSC =      -1.345794
At the time of critical stability:
  . Armor stability computation is
    performed up to node          JSATM =     167
  . Upper waterline node          ISATM =     195
  . Lower waterline node          IWATM =     301
```

Table C-3: List of figures for Run P2 presented in Appendix C.

FIGURE	DESCRIPTION	COUNTERPART IN APPENDIX B (RUN P1)
C-1	Incident wave train η_i at the seaward boundary	Figure B-2
C-2	Reflected wave train η_r at the seaward boundary, computed and measured	Figure B-3
C-3	Time series of surface elevation at four different locations: $x = 0.31, 0.61, 0.92, 1.22$	Figure B-4
C-4	Time series of run-up height on the *upper* slope Z_r^U, computed and measured	Figure B-5
C-5	Time series of run-up height on the *lower* slope Z_r^L	Figure B-6
C-6	Instantaneous spatial variations of η and u	Figure B-7
C-7	Instantaneous spatial variations of q_b and u_b	Figure B-8
C-8	Instantaneous spatial variations of u_p and m_p	Figure B-9
C-9	Mean, maximum, and minimum values of η	Figure B-10
C-10	Mean, maximum, and minimum values of u	Figure B-11
C-11	Mean, maximum, and minimum values of m	Figure B-12
C-12	Mean, maximum, and minimum values of q_b	Figure B-13
C-13	Mean, maximum, and minimum values of u_p	Figure B-14
C-14	Mean, maximum, and minimum values of m_p	Figure B-15
C-15	Mass balance for Region 1	Figure B-16
C-16	Mass balance for Regions 2+3	Figure B-17
C-17	Time-averaged energy quantities for Region 1	Figure B-18
C-18	Time-averaged energy quantities for Regions 2+3	Figure B-19
C-19	Energy balance for Regions 2+3	Figure B-20
C-20	Spatial variation of the local stability number N_{sx}	Figure B-21
C-21	Spatial variations of various quantities at the time of the critical armor stability	Figure B-22

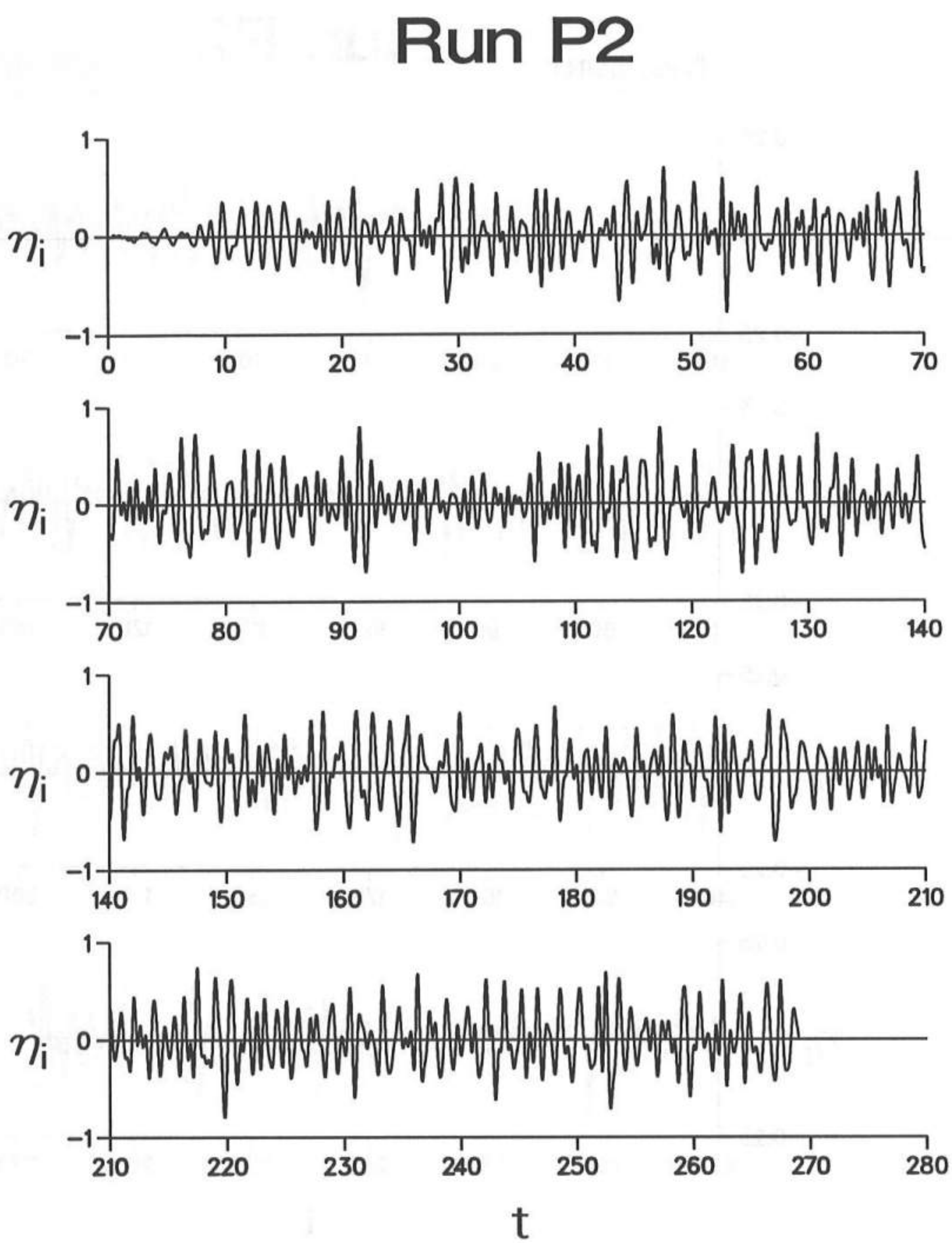

Run P2
η_i
1
0
−1
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
t

Figure C-1

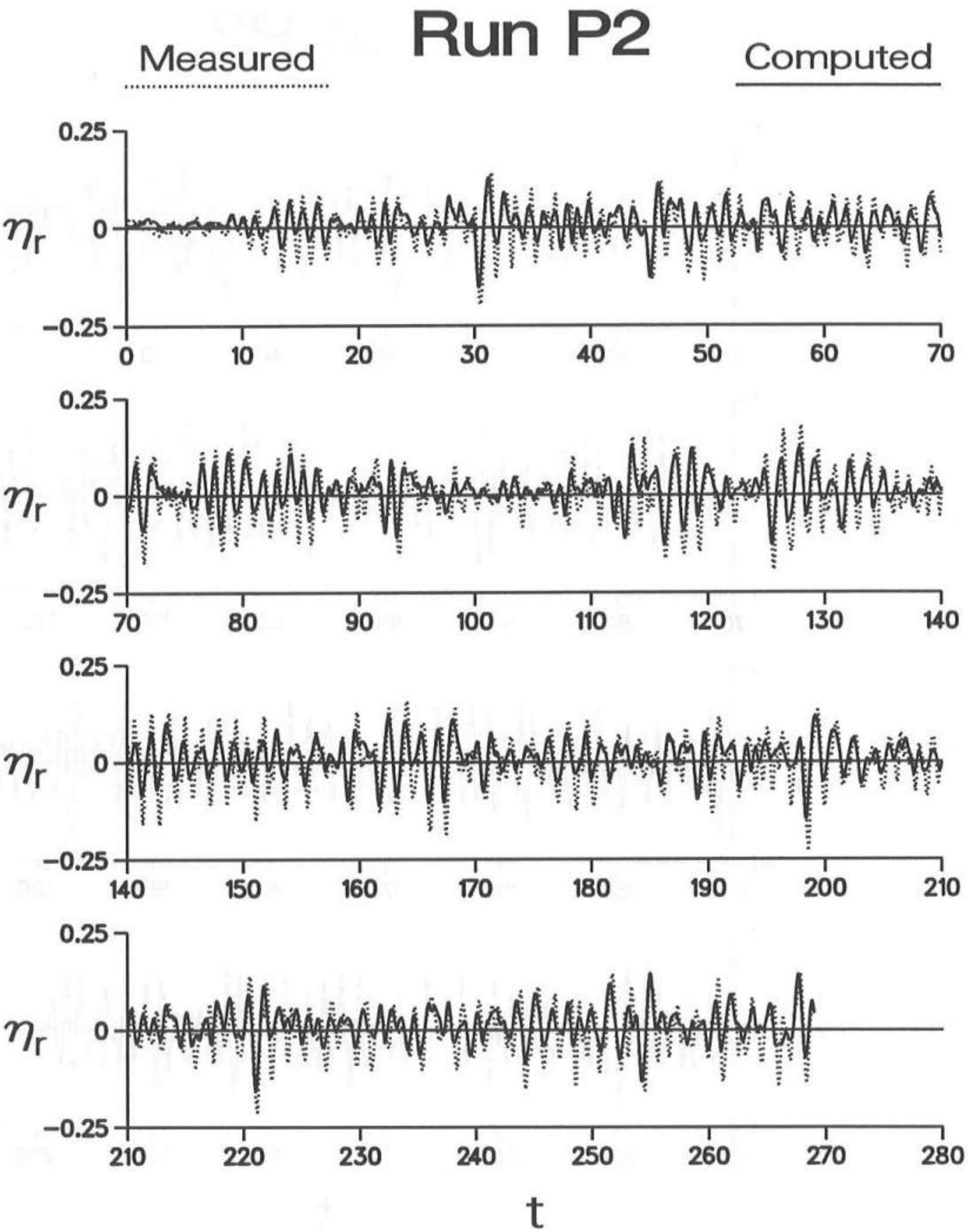

Run P2
Measured
Computed
0.25
0
−0.25
η_r
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
t

Figure C-2

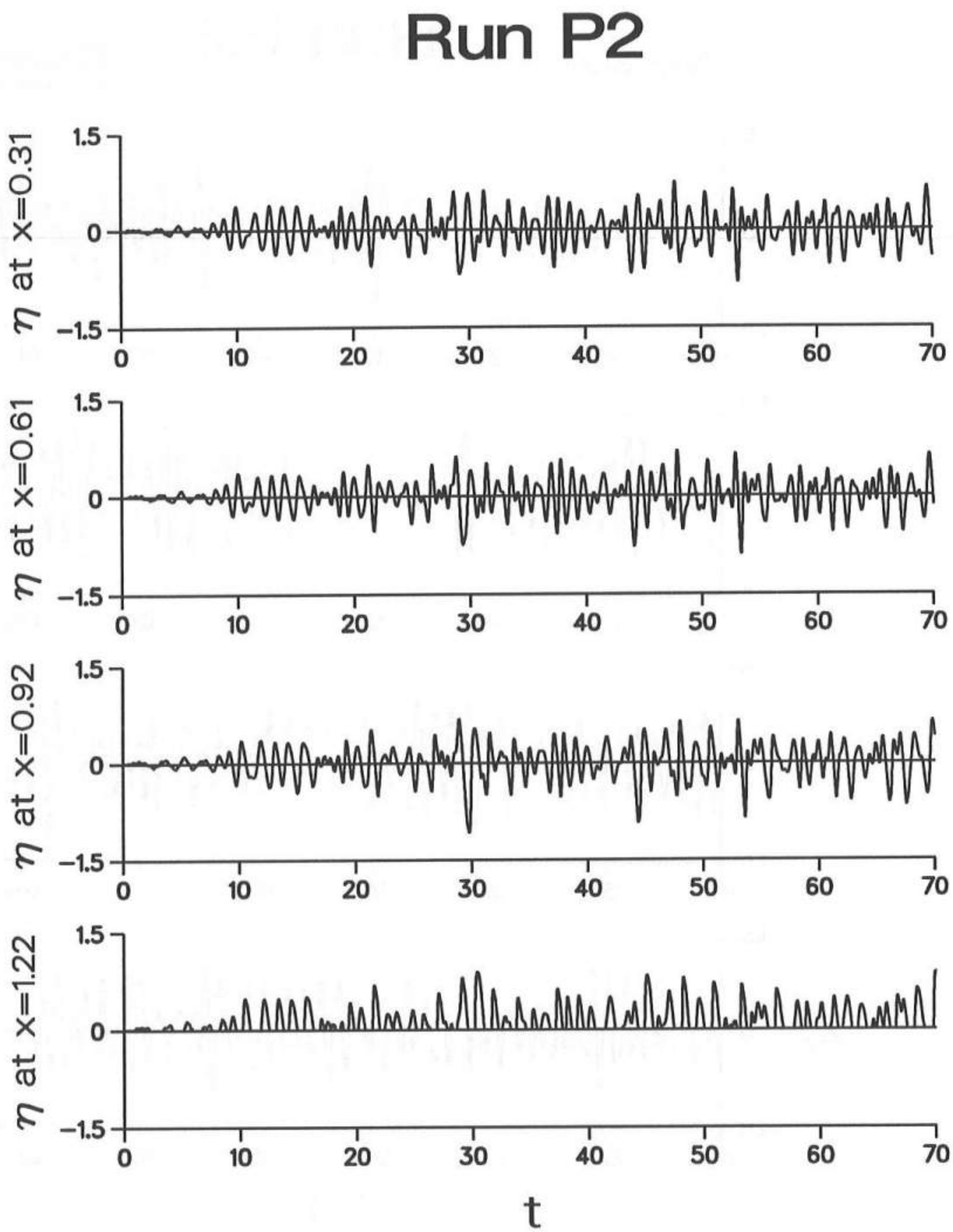
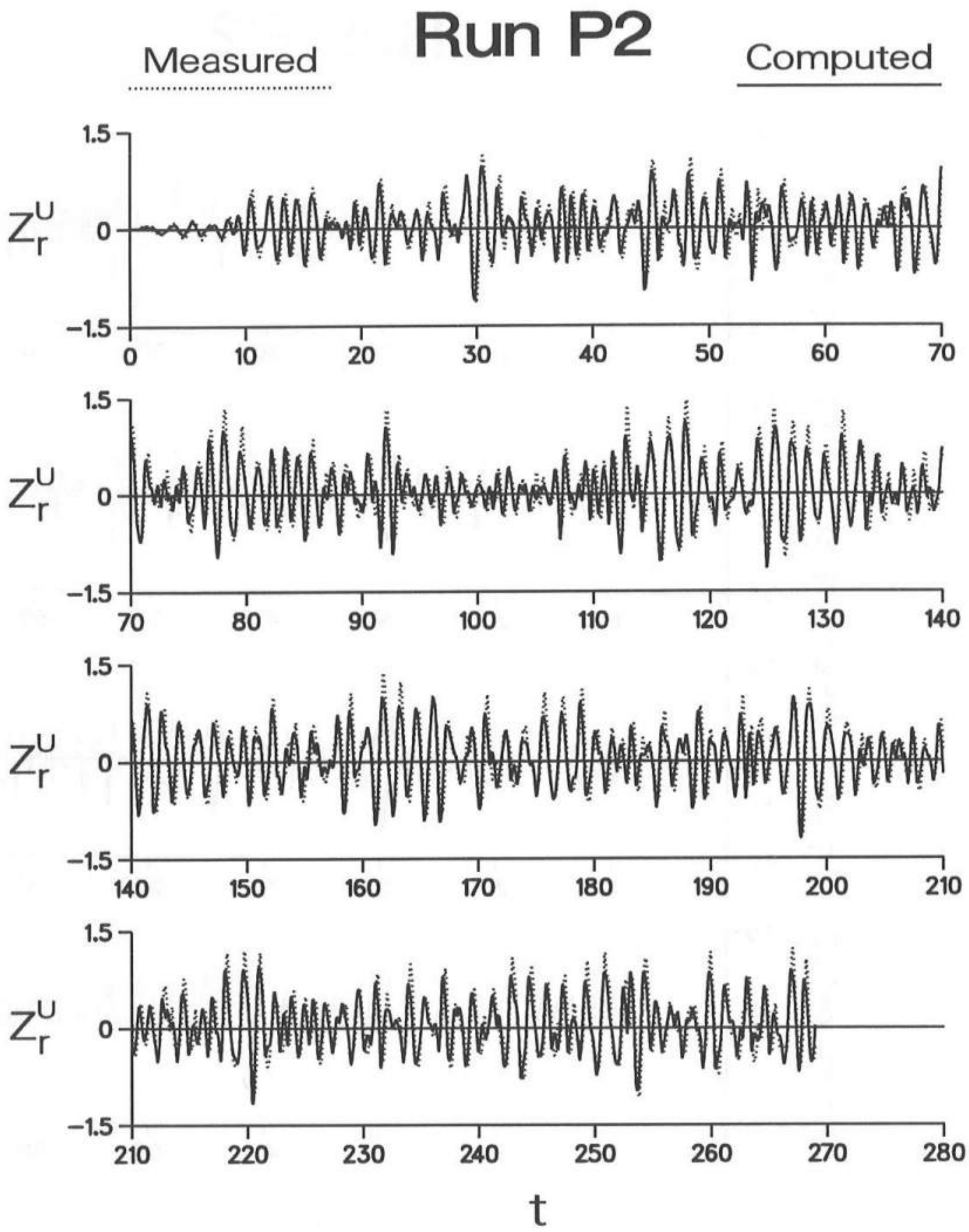

Run P2
1.5
0
-1.5
η at x=0.31
η at x=0.61
η at x=0.92
η at x=1.22
0
10
20
30
40
50
60
70
t

Figure C-3



Run P2
Measured
Computed
1.5
0
−1.5
Z_r^U
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
t

Figure C-4

Run P2

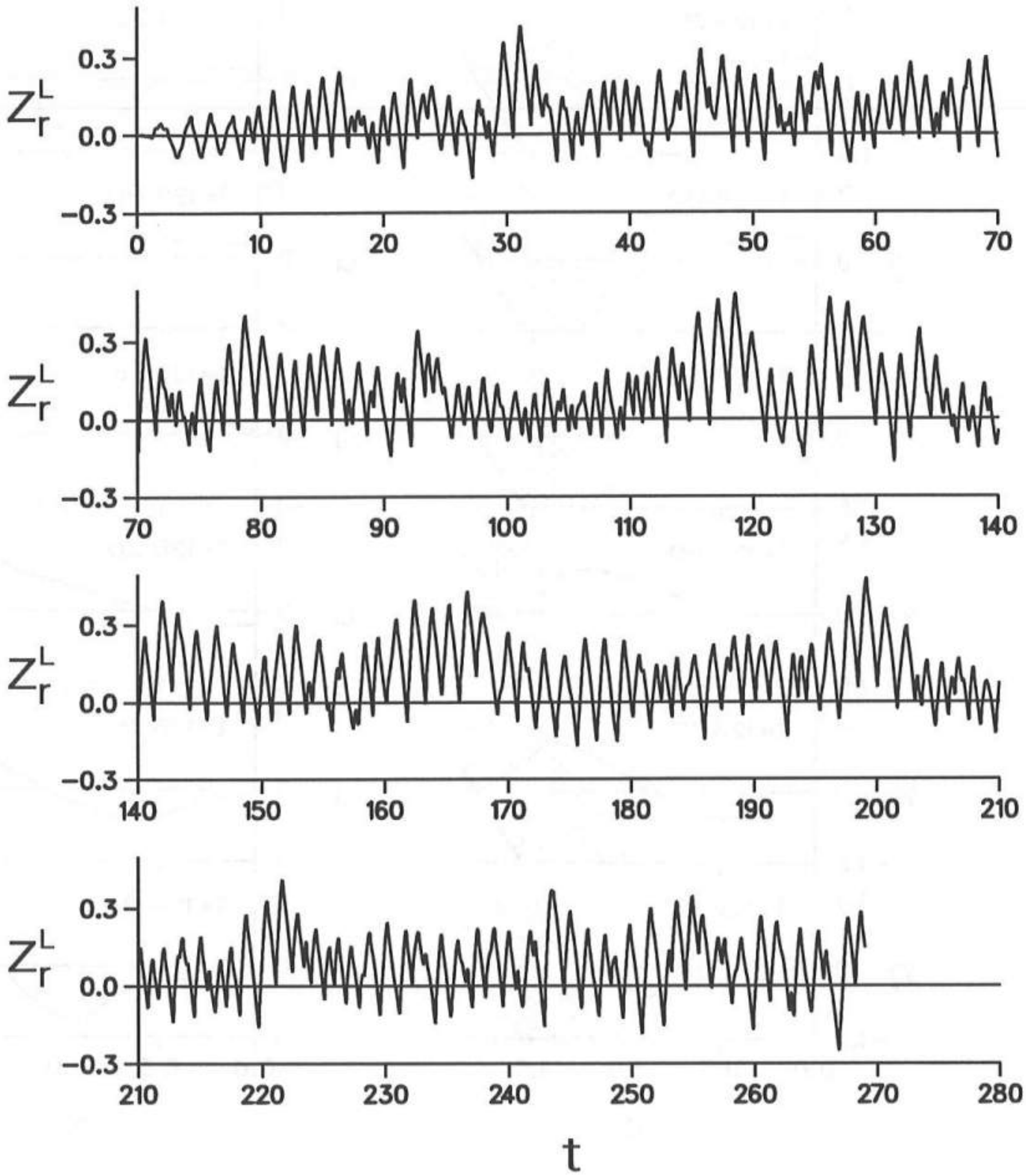


Figure C-5

Run P2

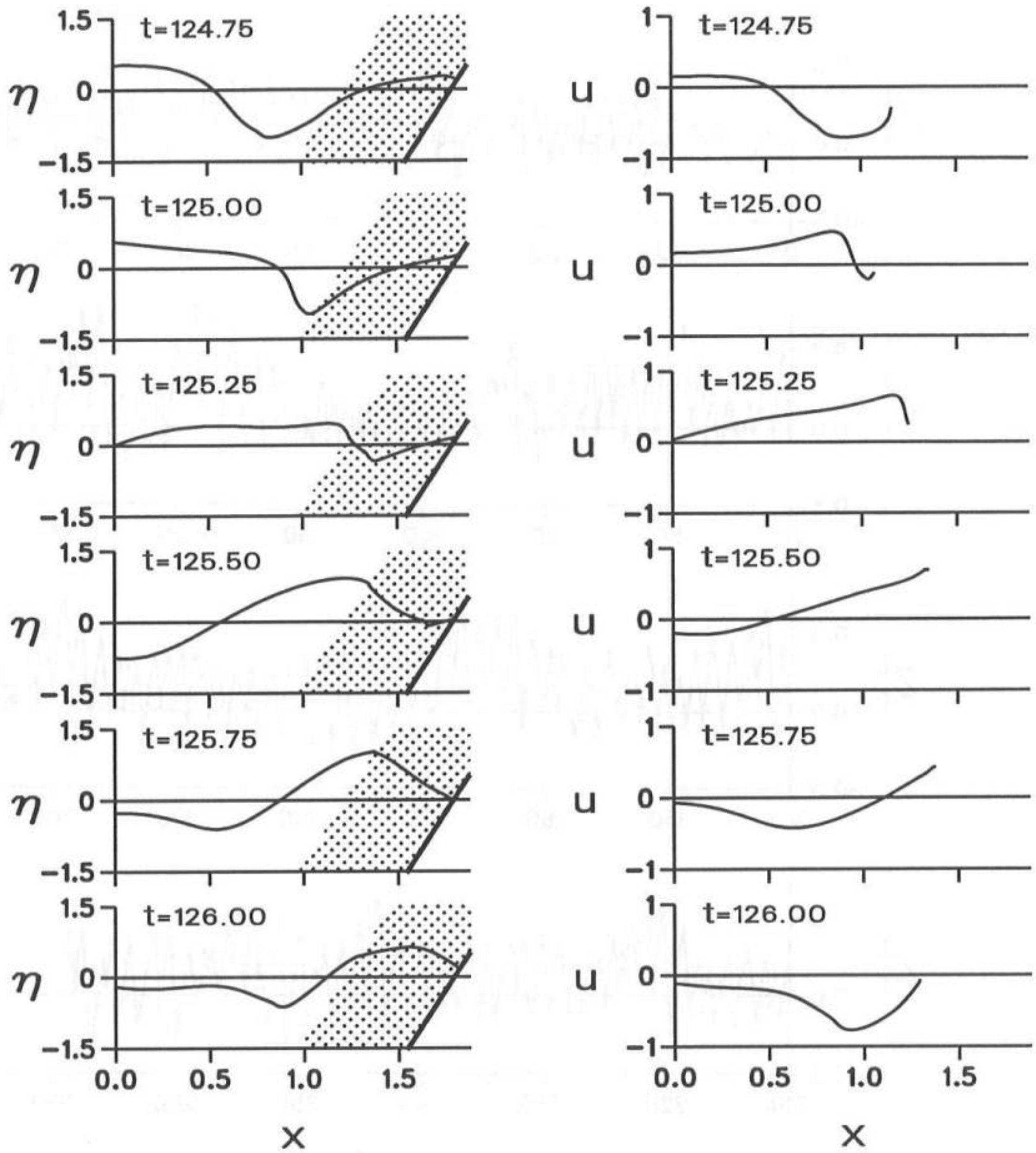


Figure C-6

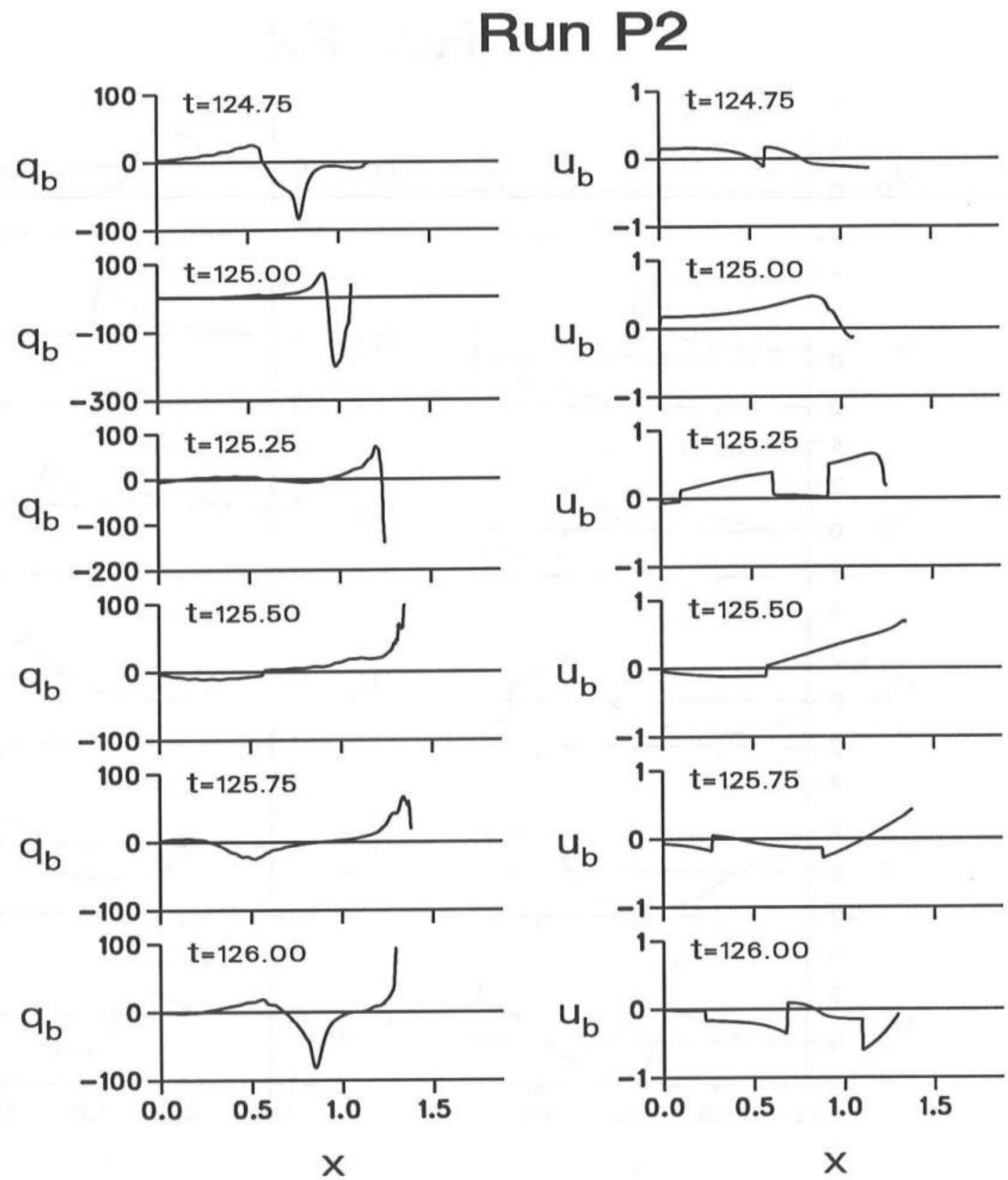

Run P2
t=124.75
t=125.00
t=125.25
t=125.50
t=125.75
t=126.00
q_b
u_b
100
0
-100
-300
-200
1
-1
0.0
0.5
1.0
1.5
x

Figure C-7

Run P2

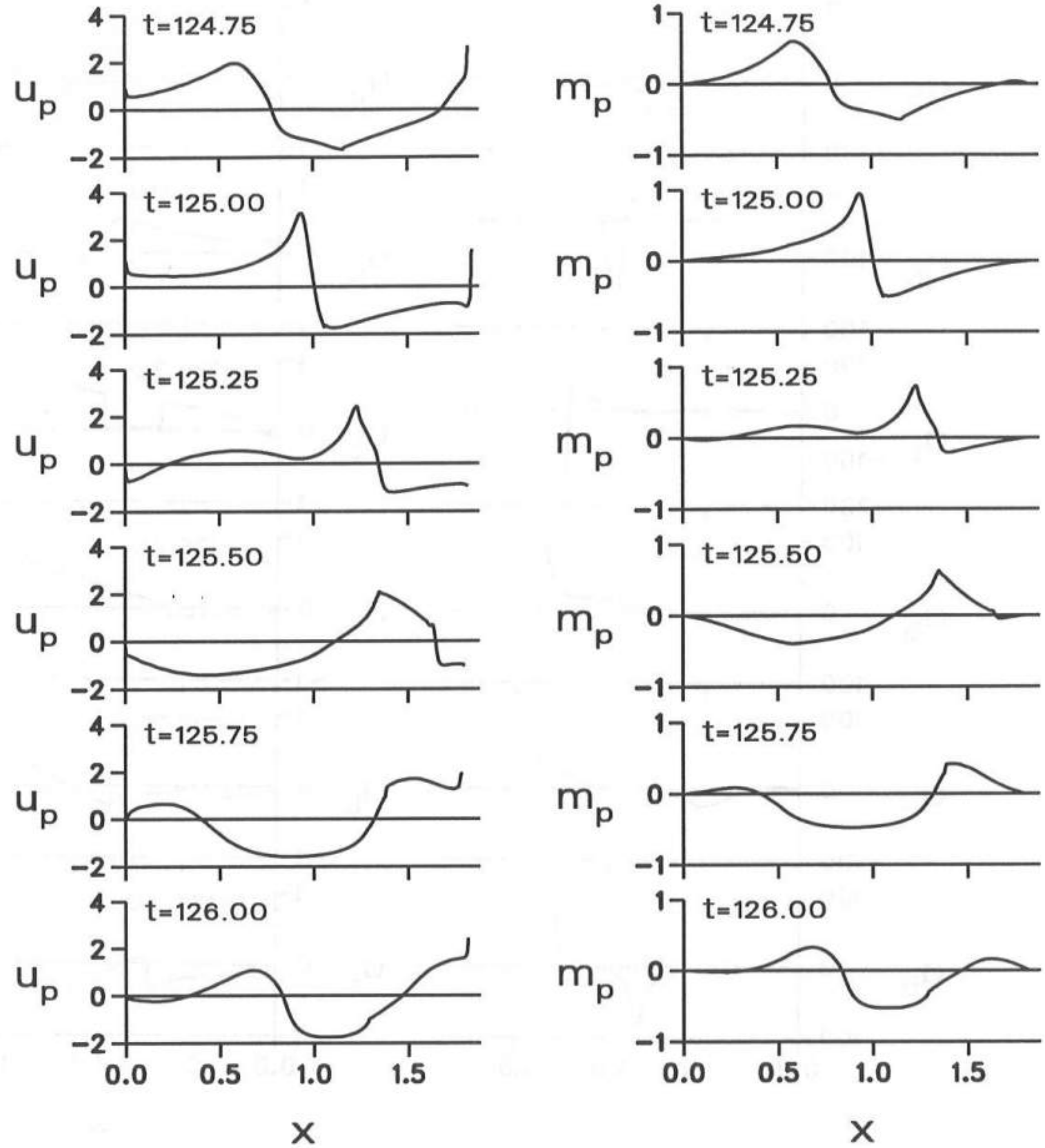


Figure C-8

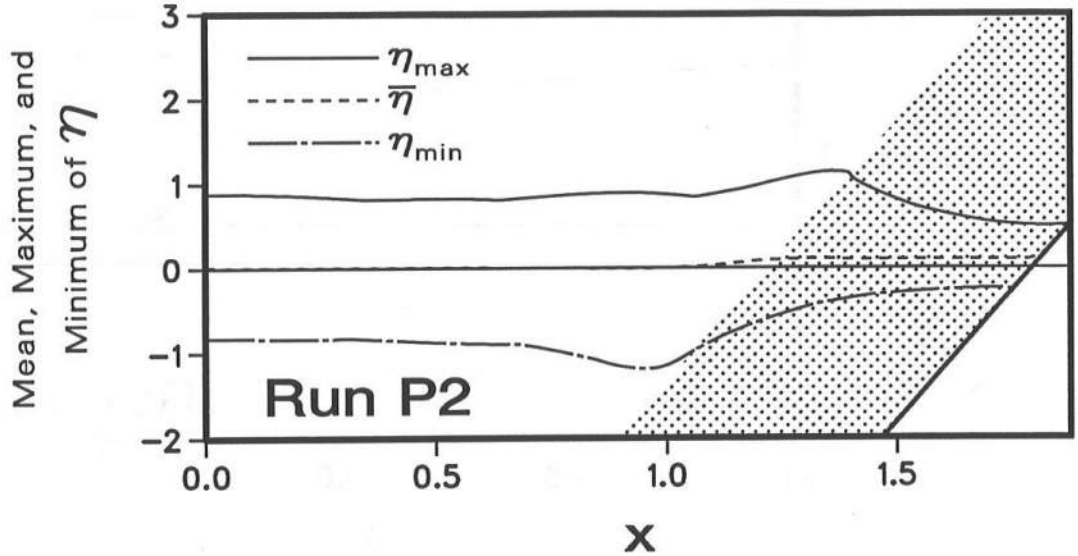


Figure C-9

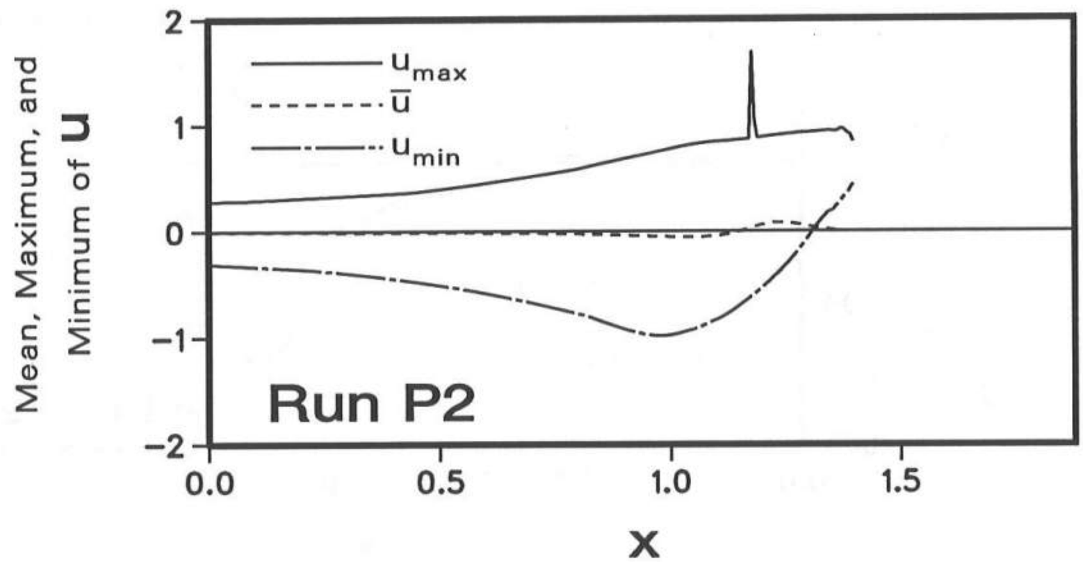


Figure C-10

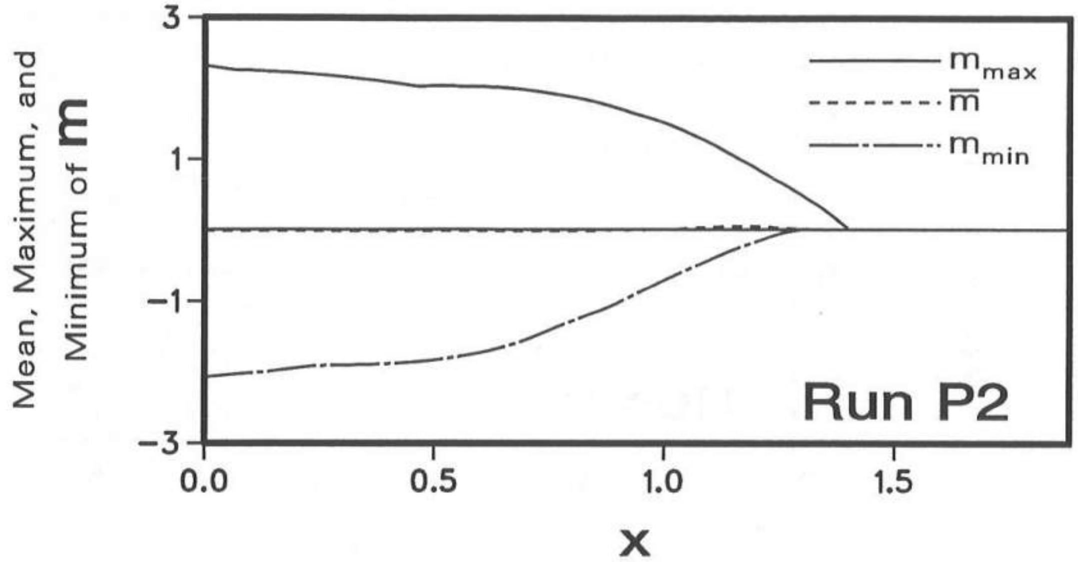


Figure C-11

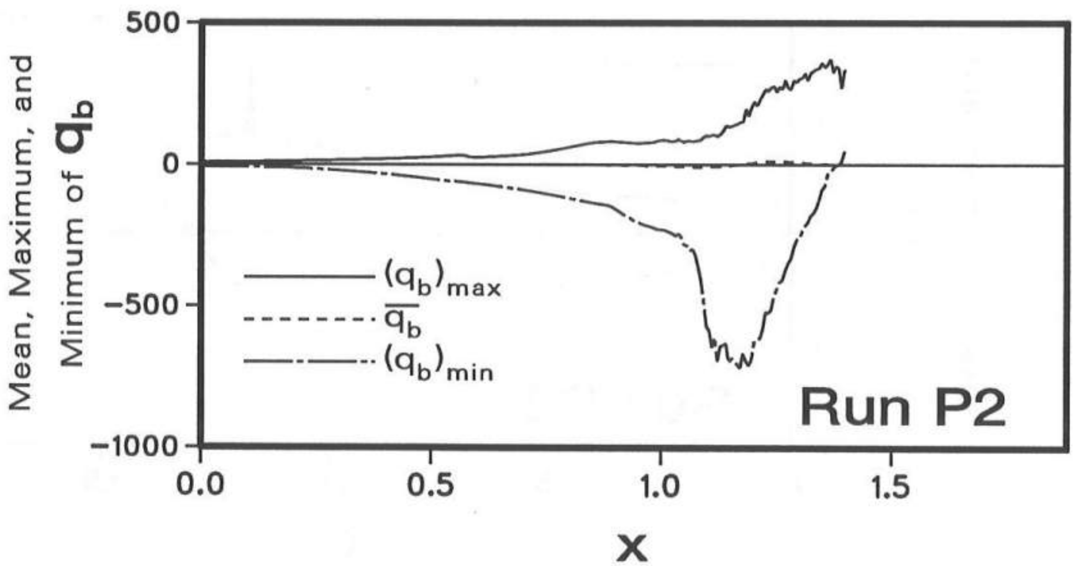


Figure C-12

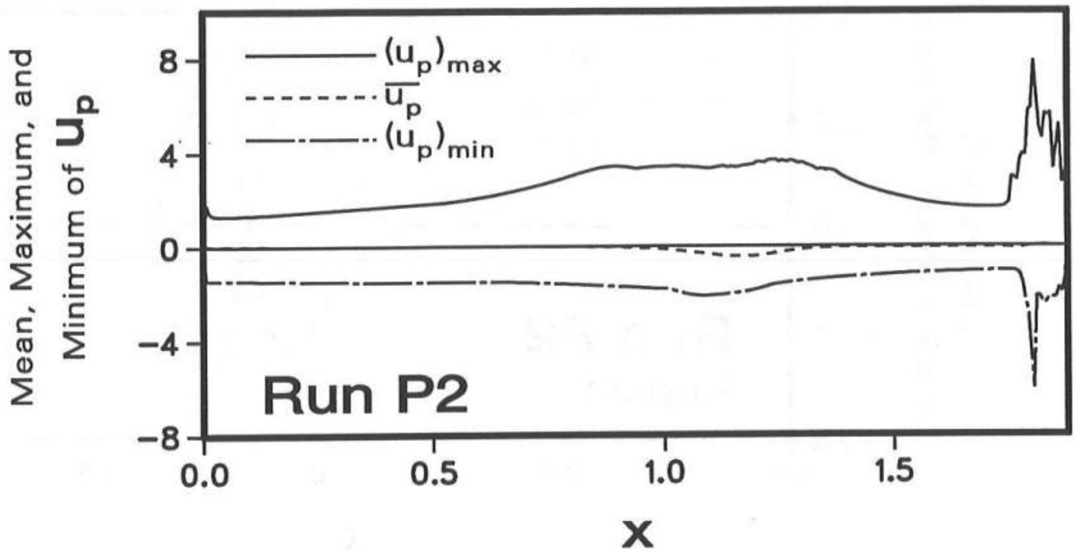


Figure C-13

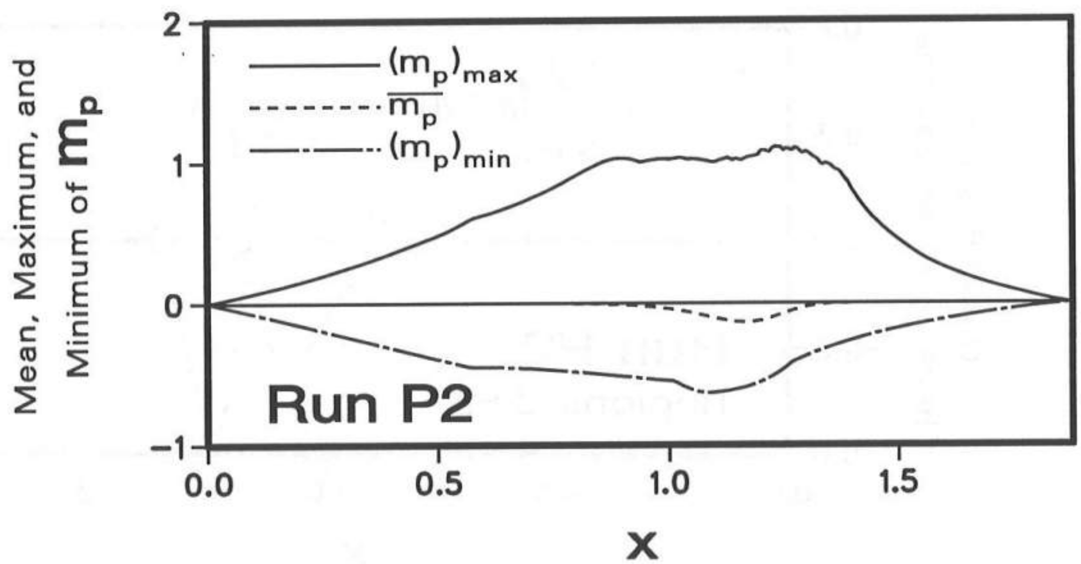


Figure C-14

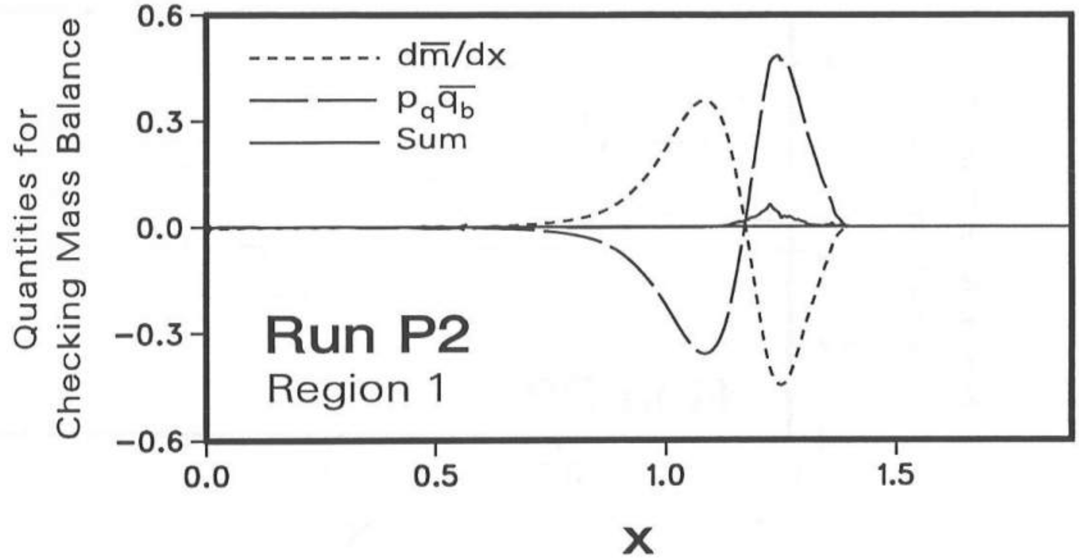


Figure C-15

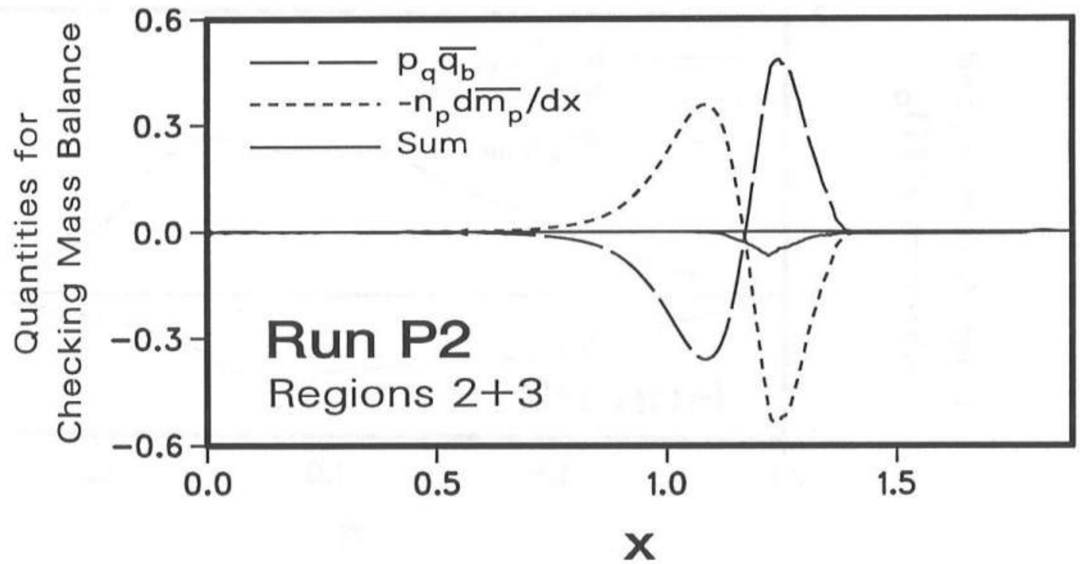


Figure C-16

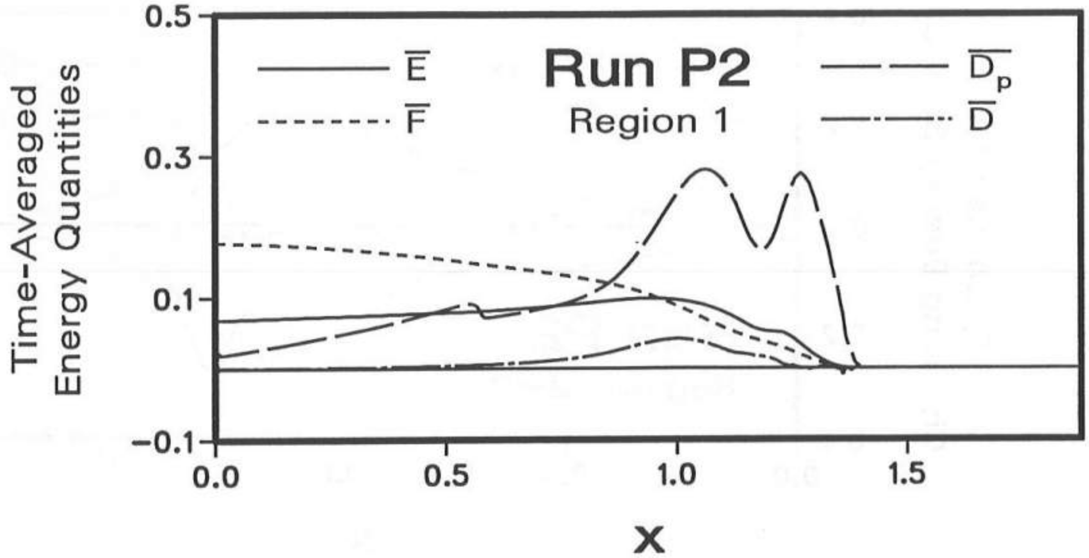


Figure C-17

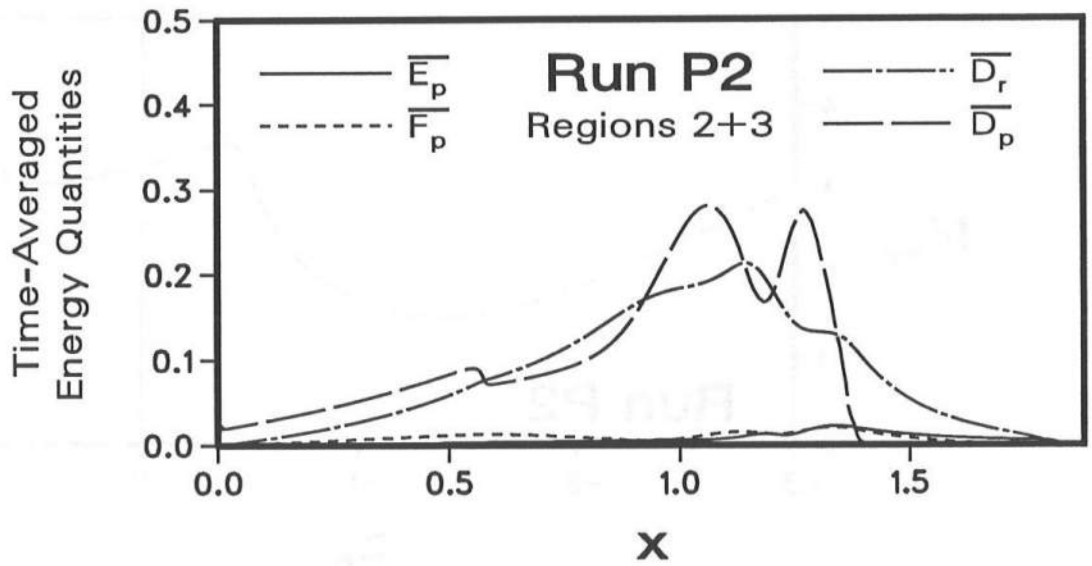


Figure C-18

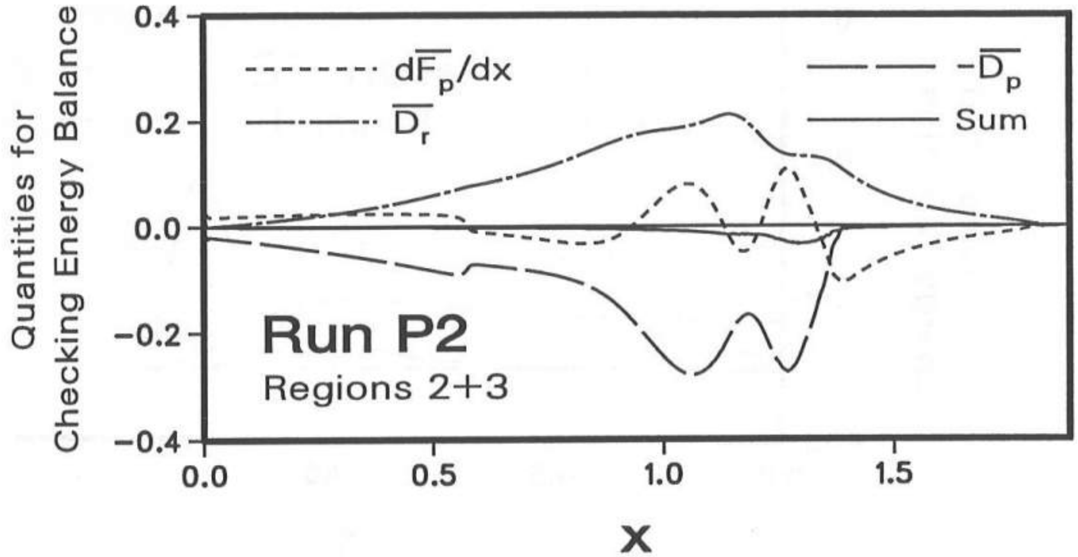


Figure C-19

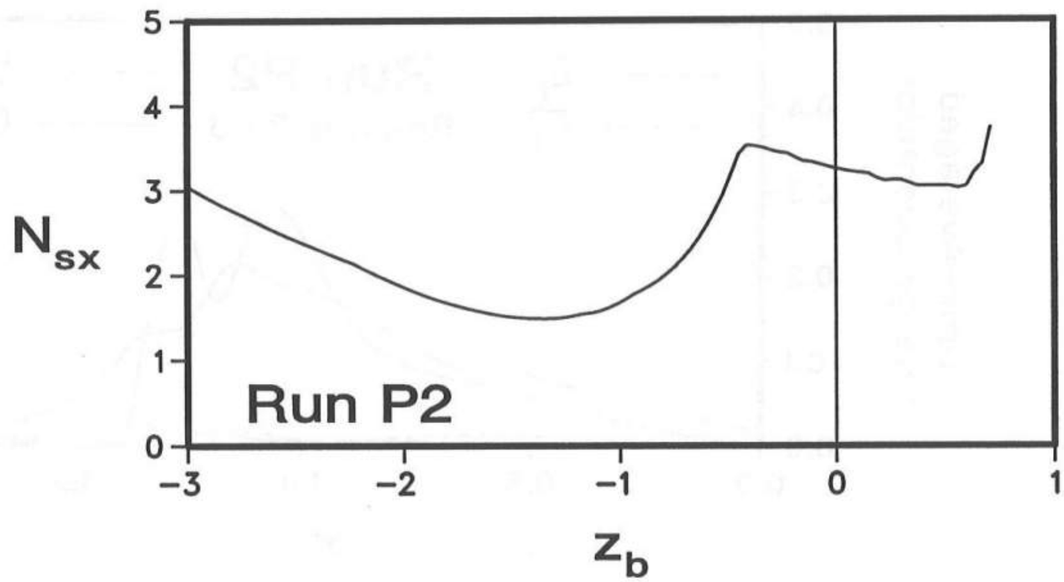


Figure C-20

Run P2

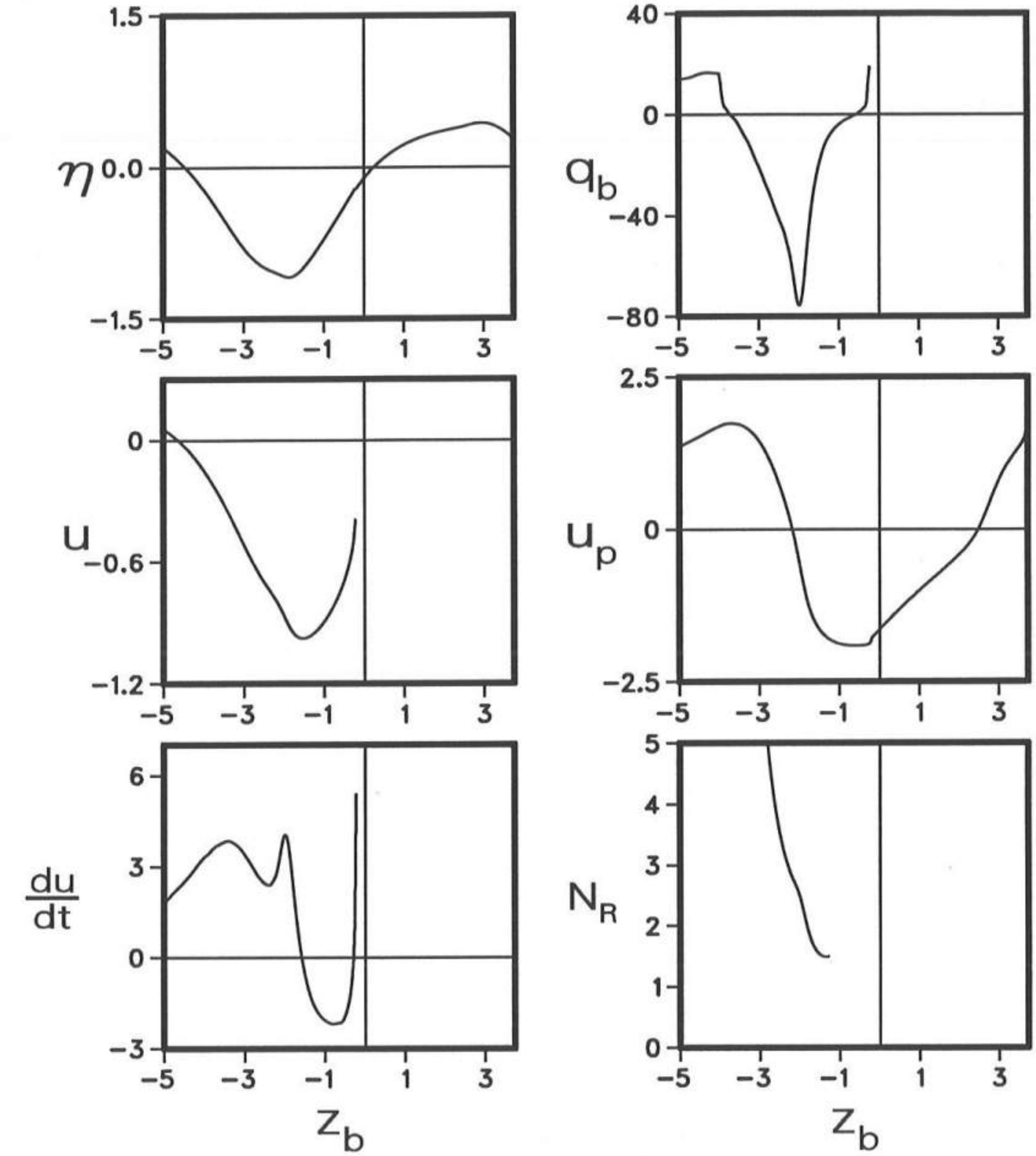


Figure C-21

Appendix D

Example: Run P3

This appendix shows the results of the PBREAK simulation for Run P3 of Kobayashi, Cox, and Wurjanto (1991). The structure and experiment procedure for Run P3 were the same as those for Run P1 described in Appendix B.

• Input Data •

Only the input parameters and values that are different from Run P1 are listed below.

- **Slope roughness.** For Run P3, the friction factor `FWP` is taken to be 0.1 on the basis of Kobayashi and Greenwald (1988).
- **Input wave train.** The duration of the input wave train for Run P3 was 365 seconds. The train is plotted in Figure D-1 and stored in the file `append.iwt` on the accompanying disk.
  - The name of the file containing the input wave train is `FINP2=tkcwp3` for this run.
  - `HREFP`=0.0457m is the significant wave height of the input wave train, whereas `TP`= 1.738s [reported as 1.74s in Kobayashi, Cox, and Wurjanto (1991)] is the mean period.
  - The normalized computation duration `TMAX` = 365s/1.738s = 210.011508.
- The *diagnostic* part of Run P3 consists only of a single computation with `ICOMP=1` where the following instrumental computation parameters are specified in the primary input data file.

Sequence	Computation Units	INITS	MULTIF	DELTAS	DELTAW	EPSI1	EPSI2
1	1-211	200	1	.001000	.001000	1.00	1.00

- `MSAVA1`=195, `MSAVA2`=200, `NTIMES`=4. The spatial variations of the hydrodynamic quantities η, u, q_b, u_b, u_p, and m_p are stored at the normalized time t = 194.25, 194.50, 194.75, ..., 200.00. In all, there are 24 instantaneous variations stored for each of the six quantities.

The primary input data file for the *actual* part of PBREAK computation for Run P3 is presented in Table D-1 and stored as the file `append.inp` on the accompanying disk. The file ODOC containing a concise documentation produced by PBREAK for Run P3 is presented in Table D-2 and stored on the accompanying disk under the name `append.doc`.

It is noted that for this run, we could have used the option ICOMP=4 as explained in Figure 4. To show that the computed results with ICOMP=1,3 and ICOMP=4 are identical, we present the file ODOC produced by the computation with ICOMP=4 in Table D-3. This computation with ICOMP=4 used the primary input data file presented in Table D-1.

• FIGURES •

Each figure presented in this appendix (Run P3) has a counterpart in Appendix B (Run P1). Since the outputs from PBREAK for Runs P1 and P3 are organized in the same way, and assuming the familiarity with the example of Run P1 in Appendix B, explanations on the names of variables and output files are not duplicated in this appendix. The figures presented in this appendix are listed on Table D-4.

Table D-1: Primary input data file for PBREAK computation of Run P3.

```
        4                   --> NLINES
-------------------------------------------------------------------------
              PBREAK        *** Kobayashi, Cox, and Wurjanto 1991
                            *** Run P3
-------------------------------------------------------------------------
1                           --> ISYST
1                           --> IBOT
      195      200      4   --> MSAVA1,MSAVA2,NTIMES
        5                   --> NNODB
        1       51    101   --> NODB(1)-(2)-(3)
      151      201          --> NODB(4)-(5)
tkcwp3                      --> FINP2
   210.011508               --> TMAX(normalized)
------ Block of instrumental computation parameters begins
      200                   --> INITS
        1                   --> MULTIF
      .001000     .001000   --> DELTAS,DELTAW(normalized)
     1.000000    1.000000   --> EPSI1,EPSI2
------ Block ends
        0                   --> MSTAT
       50                   --> NRATE
        4                   --> NDELR
    21.000000               --> DELRP(1)(millimeters)
    27.500000               -->        (2)
    15.000000               -->        (3)
     5.000000               -->        (4)
      .045700    1.738000   --> HREFP(meters),TP(seconds)
      .100000               --> FWP
      .400000               --> DSEAP(meters)
      .333333               --> TSLOPS
        1                   --> NUSEG
     2.400000     .333333   --> WUSEG(1)(meters),TUSEG(1)
        2                   --> NLSEG
      .566049     .000000   --> WLSEG(1)(meters),TLSEG(1)
     1.835000     .333333   -->        (2)                (2)
------ Block of permeable underlayer data begins
0                           --> ISALT
    20.000000               --> TEMPER(celcius)
      .021000               --> DPP(meters)
      .480000               --> PORO
  1140.000000    2.700000   --> ALPA0,BETA0
------ Block ends
------ Block of armor stability data begins
     2.700000    1.191754   --> SG,TANPHI
      .900000     .600000   --> C2,C3
      .500000     .300000   --> CD,CL
     1.500000               --> CM
     1.000000    -.800000   --> AMAX,AMIN
------ Block ends
```

Table D-2: File ODOC produced by PBREAK computation of Run P3.

```
---------------------------------------------------------------------
          PBREAK               *** Kobayashi, Cox, and Wurjanto 1991
                               *** Run P3
---------------------------------------------------------------------

WAVE CONDITION

Incident wave at seaward boundary given as input
Number of data points      NDATA =   5841
Norm. maximum surface elev.      =      1.094400
Norm. minimum surface elev.      =     -0.797800
Reference wave period            =      1.738000 sec.
Reference wave height            =      0.045700 meters
Depth at seaward boundary        =      0.400000 meters
Norm. depth at seaw. boundary    =      8.753
Normalized wave length           =      7.839
Ursell number                    =      7.021
Surf similarity parameter        =      3.386
"Sigma"                          =     25.464

PROPERTIES OF THE UPPER SLOPE

Friction factor              =       0.100000
Norm. friction factor        =       1.273199
Number of segments           =       1
-----------------------------------------------------------
  SEGMENT    WUSEG(I)    TUSEG(I)    XUSEG(I)    ZUSEG(I)
        I      meters                  meters      meters
-----------------------------------------------------------
        1    2.400000    0.333333    0.000000   -0.400000
        2                            2.400000    0.399999
-----------------------------------------------------------

GEOMETRY OF THE LOWER SLOPE

Number of Segments           =       2
-----------------------------------------------------------
  SEGMENT    WLSEG(I)    TLSEG(I)    XLSEG(I)    ZLSEG(I)
        I      meters                  meters      meters
-----------------------------------------------------------
        1    0.566049    0.000000    0.000000   -0.400000
        2    1.835000    0.333333    0.566049   -0.400000
        3                            2.401049    0.211666
-----------------------------------------------------------

PROPERTIES OF THE PERMEABLE UNDERLAYER and
PARAMETERS RELATED TO PERMEABILITY

Fresh water at temperature             TEMPER =        20.000 degrees C
Kinematic viscosity of water x 1.D+06
                        = 1.D+06 x VISCO =          1.004000 m.m/sec
Representative diameter of permeable
                underlayer materials  DPP =          0.021 meters
Porosity of the permeable underlayer  PORO =         0.480
```

Table D-2 continued.

```
Dimensionless constants                ALPA0 =          1140.000
                                       BETA0 =             2.700
Coefficient expressing the laminar flow resistance
                          ALPAP [alpha prime] =            1.584
Coefficient expressing the turbulent flow resistance
                           BETAP [beta prime] =          604.539
Dimensionless parameters that arise in consequence
    of normalizing the governing equations:
                                       PARPU =           0.785463D-01
                          PARPQ = PORO*PARPU =           0.377022D-01
                                       PARMU =           0.103787D+00

INPUT PARAMETERS FOR ARMOR STABILITY

Specific gravity              SG =    2.700
Armor friction Factor     TANPHI =    1.192
Area coefficient              C2 =    0.900
Volume coefficient            C3 =    0.600
Drag coefficient              CD =    0.500
Lift coefficient              CL =    0.300
Inertia coefficient           CM =    1.500
Upper and lower bounds of du/dt
        (normalized by gravity)
                            AMAX =    1.000
                            AMIN =   -0.800

COMPUTATION PARAMETERS

Total number of spatial nodes      JE =      400
No. of nodes with upper slope below SWL
                                INITS =      200
No. of nodes with lower slope below SWL
                                INITW =      295
Minimum allowable value of INVT
                              = MINVT =     1250
  where INVT is the number of time steps
  in one reference wave period
Computed time series are stored at rate NRATE
  per reference wave period with NRATE =      50
Normalized delta x     X =        0.515596D-02
Computation duration  TMAX =     210.011508 ref. wave periods
Statistical calculations are performed excluding
     the first MSTAT computation units
     with                       MSTAT =        0
Instantaneous spatial variations are stored NTIMES times
     at equal intervals per reference wave period from
     computation unit MSAVA1 to MSAVA2, inclusive,
     with                      NTIMES =        4
                               MSAVA1 =      195
                               MSAVA2 =      200
Other parameters are given in COMPUTATION LOG below
Positioning:
. Node      1         is at x =        0.000000
. Node      200 (INITS) is at x =      1.026035
. Node      295 (INITW) is at x =      1.515851
. Node      400 (JE)    is at x =      2.057227
```

Table D-2 continued.

COMPUTATION LOG

MULTIF = integer multiplication factor
MULTIF*MINVT = minimum value of INVT used in a sequence
INVT = number of time steps in one reference wave period
MINVT = minimum allowable value of INVT
EPSI1, EPSI2 = damping coefficients
OMEGA = numerical stability indicator
NDIV = time level in a computation unit when DELTAS or DELTAW changes value
DELTAS = normalized water depth defining the computational upper waterline
DELTAW = normalized water depth defining the computational lower waterline

Sequence Number	Computation Units	MULTIF	EPSI1	EPSI2
1	1 to 211	1	1.00	1.00

Comp. Unit	INVT	OMEGA	NDIV	DELTAS	DELTAW
1	1250	1.0	1250	0.001000	0.001000
2	1250	1.0	1250	0.001000	0.001000
3	1250	1.0	1250	0.001000	0.001000
4	1250	1.0	1250	0.001000	0.001000
5	1250	1.0	1250	0.001000	0.001000
6	1250	1.0	1250	0.001000	0.001000
7	1250	1.0	1250	0.001000	0.001000
8	1250	1.0	1250	0.001000	0.001000
9	1250	1.0	1250	0.001000	0.001000
10	1250	1.0	1250	0.001000	0.001000
11	1250	1.0	1250	0.001000	0.001000
12	1250	1.0	1250	0.001000	0.001000
13	1250	1.0	1250	0.001000	0.001000
14	1250	1.0	1250	0.001000	0.001000
15	1250	1.0	1250	0.001000	0.001000
16	1250	1.0	1250	0.001000	0.001000
17	1850	1.5	1850	0.001000	0.001000
18	1250	1.0	1250	0.001000	0.001000
19	1250	1.0	1250	0.001000	0.001000
20	1250	1.0	1250	0.001000	0.001000
21	1250	1.0	1250	0.001000	0.001000
22	1250	1.0	1250	0.001000	0.001000
23	1250	1.0	1250	0.001000	0.001000
24	1250	1.0	1250	0.001000	0.001000
25	1250	1.0	1250	0.001000	0.001000
26	1250	1.0	1250	0.001000	0.001000
27	1250	1.0	1250	0.001000	0.001000
28	1250	1.0	1250	0.001000	0.001000

Table D-2 continued.

167	29	1250	1.0	1250	0.001000	0.001000
168	30	1250	1.0	1250	0.001000	0.001000
169	31	1250	1.0	1250	0.001000	0.001000
170	32	1250	1.0	1250	0.001000	0.001000
171	33	1250	1.0	1250	0.001000	0.001000
172	34	1250	1.0	1250	0.001000	0.001000
173	35	1250	1.0	1250	0.001000	0.001000
174	36	1250	1.0	1250	0.001000	0.001000
175	37	1250	1.0	1250	0.001000	0.001000
176	38	1250	1.0	1250	0.001000	0.001000
177	39	1250	1.0	1250	0.001000	0.001000
178	40	1850	1.5	1850	0.001000	0.001000
179	41	1250	1.0	1250	0.001000	0.001000
180	42	1250	1.0	1250	0.001000	0.001000
181	43	1250	1.0	1250	0.001000	0.001000
182	44	1250	1.0	1250	0.001000	0.001000
183	45	1250	1.0	1250	0.001000	0.001000
184	46	1250	1.0	1250	0.001000	0.001000
185	47	1250	1.0	1250	0.001000	0.001000
186	48	1250	1.0	1250	0.001000	0.001000
187	49	1250	1.0	1250	0.001000	0.001000
188	50	1250	1.0	1250	0.001000	0.001000
189	51	1850	1.5	1850	0.001000	0.001000
190	52	1250	1.0	1250	0.001000	0.001000
191	53	1250	1.0	1250	0.001000	0.001000
192	54	1250	1.0	1250	0.001000	0.001000
193	55	1250	1.0	1250	0.001000	0.001000
194	56	1250	1.0	1250	0.001000	0.001000
195	57	1250	1.0	1250	0.001000	0.001000
196	58	1250	1.0	1250	0.001000	0.001000
197	59	1250	1.0	1250	0.001000	0.001000
198	60	1250	1.0	1250	0.001000	0.001000
199	61	1850	1.5	1850	0.001000	0.001000
200	62	1250	1.0	1250	0.001000	0.001000
201	63	1250	1.0	1250	0.001000	0.001000
202	64	1250	1.0	1250	0.001000	0.001000
203	65	1250	1.0	1250	0.001000	0.001000
204	66	1250	1.0	1250	0.001000	0.001000
205	67	1250	1.0	1250	0.001000	0.001000
206	68	1250	1.0	1250	0.001000	0.001000
207	69	1250	1.0	1250	0.001000	0.001000
208	70	1250	1.0	1250	0.001000	0.001000
209	71	1250	1.0	1250	0.001000	0.001000
210	72	1850	1.5	1850	0.001000	0.001000
211	73	1250	1.0	1250	0.001000	0.001000
212	74	1250	1.0	1250	0.001000	0.001000
213	75	1250	1.0	1250	0.001000	0.001000
214	76	1250	1.0	1250	0.001000	0.001000
215	77	1250	1.0	1250	0.001000	0.001000
216	78	1250	1.0	1250	0.001000	0.001000
217	79	1250	1.0	1250	0.001000	0.001000
218	80	1850	1.5	1850	0.001000	0.001000
219	81	1250	1.0	1250	0.001000	0.001000
220	82	1250	1.0	1250	0.001000	0.001000
221	83	1250	1.0	1250	0.001000	0.001000
222	84	1250	1.0	1250	0.001000	0.001000

223	85	1250	1.0	1250	0.001000	0.001000
223	85	1250	1.0	1250	0.001000	0.001000
224	86	1250	1.0	1250	0.001000	0.001000
225	87	1250	1.0	1250	0.001000	0.001000
226	88	2750	2.2	2750	0.001000	0.001000
227	89	1250	1.0	1250	0.001000	0.001000
228	90	1250	1.0	1250	0.001000	0.001000
229	91	1250	1.0	1250	0.001000	0.001000
230	92	1850	1.5	1850	0.001000	0.001000
231	93	1250	1.0	1250	0.001000	0.001000
232	94	1250	1.0	1250	0.001000	0.001000
233	95	1250	1.0	1250	0.001000	0.001000
234	96	1250	1.0	1250	0.001000	0.001000
235	97	1250	1.0	1250	0.001000	0.001000
236	98	1850	1.5	1850	0.001000	0.001000
237	99	1250	1.0	1250	0.001000	0.001000
238	100	1250	1.0	1250	0.001000	0.001000
239	101	1850	1.5	1850	0.001000	0.001000
240	102	1250	1.0	1250	0.001000	0.001000
241	103	1250	1.0	1250	0.001000	0.001000
242	104	1250	1.0	1250	0.001000	0.001000
243	105	1250	1.0	1250	0.001000	0.001000
244	106	1250	1.0	1250	0.001000	0.001000
245	107	1250	1.0	1250	0.001000	0.001000
246	108	1250	1.0	1250	0.001000	0.001000
247	109	1850	1.5	889	0.001000	0.001000
248				1850	0.001000	0.000943
249	110	1250	1.0	1250	0.001000	0.000943
250	111	1250	1.0	1250	0.001000	0.000943
251	112	1250	1.0	1250	0.001000	0.000943
252	113	1250	1.0	1250	0.001000	0.000943
253	114	1250	1.0	1250	0.001000	0.000943
254	115	1250	1.0	1250	0.001000	0.000943
255	116	1850	1.5	1850	0.001000	0.000943
256	117	1250	1.0	1250	0.001000	0.000943
257	118	1250	1.0	1250	0.001000	0.000943
258	119	1850	1.5	1850	0.001000	0.000943
259	120	1250	1.0	1250	0.001000	0.000943
260	121	1250	1.0	1250	0.001000	0.000943
261	122	1250	1.0	1250	0.001000	0.000943
262	123	1250	1.0	210	0.001000	0.000943
263				1250	0.001000	0.000895
264	124	1850	1.5	1850	0.001000	0.000895
265	125	1250	1.0	1250	0.001000	0.000895
266	126	1250	1.0	1250	0.001000	0.000895
267	127	1250	1.0	1250	0.001000	0.000895
268	128	1250	1.0	1250	0.001000	0.000895
269	129	1250	1.0	1250	0.001000	0.000895
270	130	1250	1.0	1250	0.001000	0.000895
271	131	1250	1.0	1250	0.001000	0.000895
272	132	1250	1.0	1250	0.001000	0.000895
273	133	1250	1.0	1250	0.001000	0.000895
274	134	1250	1.0	1250	0.001000	0.000895
275	135	1250	1.0	1250	0.001000	0.000895
276	136	1250	1.0	1250	0.001000	0.000895
277	137	1250	1.0	1250	0.001000	0.000895

Table D-2 continued.

278	138	1250	1.0	1250	0.001000	0.000895
279	139	1250	1.0	1250	0.001000	0.000895
280	140	1250	1.0	1250	0.001000	0.000895
281	141	1250	1.0	1250	0.001000	0.000895
282	142	1250	1.0	1250	0.001000	0.000895
283	143	1250	1.0	1250	0.001000	0.000895
284	144	1250	1.0	1250	0.001000	0.000895
285	145	1250	1.0	1250	0.001000	0.000895
286	146	1250	1.0	1250	0.001000	0.000895
287	147	1250	1.0	1250	0.001000	0.000895
288	148	1250	1.0	1250	0.001000	0.000895
289	149	1250	1.0	1250	0.001000	0.000895
290	150	1250	1.0	468	0.001000	0.000895
291				1250	0.001000	0.000825
292	151	1250	1.0	1250	0.001000	0.000825
293	152	1250	1.0	1250	0.001000	0.000825
294	153	1250	1.0	1250	0.001000	0.000825
295	154	1250	1.0	1250	0.001000	0.000825
296	155	1250	1.0	1250	0.001000	0.000825
297	156	1850	1.5	1850	0.001000	0.000825
298	157	1250	1.0	1250	0.001000	0.000825
299	158	1250	1.0	1250	0.001000	0.000825
300	159	1250	1.0	1250	0.001000	0.000825
301	160	1850	1.5	1850	0.001000	0.000825
302	161	1250	1.0	1250	0.001000	0.000825
303	162	1250	1.0	1250	0.001000	0.000825
304	163	1250	1.0	1250	0.001000	0.000825
305	164	1250	1.0	1250	0.001000	0.000825
306	165	1250	1.0	1250	0.001000	0.000825
307	166	1250	1.0	1250	0.001000	0.000825
308	167	1250	1.0	1250	0.001000	0.000825
309	168	1250	1.0	1250	0.001000	0.000825
310	169	1250	1.0	1250	0.001000	0.000825
311	170	1250	1.0	1250	0.001000	0.000825
312	171	1250	1.0	1250	0.001000	0.000825
313	172	1250	1.0	1250	0.001000	0.000825
314	173	1250	1.0	1250	0.001000	0.000825
315	174	1250	1.0	1250	0.001000	0.000825
316	175	1250	1.0	1250	0.001000	0.000825
317	176	1250	1.0	1250	0.001000	0.000825
318	177	1250	1.0	1250	0.001000	0.000825
319	178	1250	1.0	1250	0.001000	0.000825
320	179	1250	1.0	1250	0.001000	0.000825
321	180	1250	1.0	1250	0.001000	0.000825
322	181	1250	1.0	1250	0.001000	0.000825
323	182	1250	1.0	1250	0.001000	0.000825
324	183	1250	1.0	1250	0.001000	0.000825
325	184	1250	1.0	1250	0.001000	0.000825
326	185	1850	1.5	1850	0.001000	0.000825
327	186	1250	1.0	1250	0.001000	0.000825
328	187	1250	1.0	1250	0.001000	0.000825
329	188	1250	1.0	1250	0.001000	0.000825
330	189	1250	1.0	1250	0.001000	0.000825
331	190	1250	1.0	1250	0.001000	0.000825
332	191	1250	1.0	1250	0.001000	0.000825
333	192	1250	1.0	1250	0.001000	0.000825

```
       193    1250     1.0    1250  0.001000  0.000825
       194    1250     1.0    1250  0.001000  0.000825
       195    1250     1.0    1250  0.001000  0.000825
       196    1250     1.0    1250  0.001000  0.000825
       197    9200     7.4    9200  0.001000  0.000825
       198    1250     1.0    1250  0.001000  0.000825
       199    1250     1.0    1250  0.001000  0.000825
       200    1250     1.0    1250  0.001000  0.000825
       201    1250     1.0    1250  0.001000  0.000825
       202    1250     1.0    1250  0.001000  0.000825
       203    1250     1.0    1250  0.001000  0.000825
       204    1250     1.0    1250  0.001000  0.000825
       205    1250     1.0    1250  0.001000  0.000825
       206    1250     1.0    1250  0.001000  0.000825
       207    1250     1.0    1250  0.001000  0.000825
       208    1250     1.0    1250  0.001000  0.000825
       209    1250     1.0    1250  0.001000  0.000825
       210    1250     1.0    1250  0.001000  0.000825
       211    1250     1.0      14  0.001000  0.000825
-----------------------------------------------------

STATISTICAL CALCULATIONS

Statistical calculations begins at     t=TSTAT1 =       0.000000
Statistical calculations ends at       t=TSTAT2 =     210.011200
Duration of statistical calculations      TSPAN =     210.011200
Computational waterlines at t=TSTAT1:
                             . Upper slope: IST1 =    200
                             . Lower slope: IWT1 =    295
Computational waterlines at t=TSTAT2:
                             . Upper slope: IST2 =    189
                             . Lower slope: IWT2 =    299

ELEVATIONS AT SEAWARD BOUNDARY

Time-averaged elevation due to incident waves =       0.000000
Elevations due to reflected waves:
                               . Time-averaged =      0.012634
                               . Maximum       =      0.259082
                               . Minimum       =     -0.249155
```

```
RUN-UP, RUN-DOWN, SET-UP

Smallest nodes reached by computational waterlines:
                         . Upper slope: ISMIN =      174
                         . Lower slope: IWMIN =      289
Largest nodes reached by computational waterlines:
                         . Upper slope: ISMAX =      241
                         . Lower slope: IWMAX =      315
--------------------------------------------------------------
 Slope    i   Deltar(i)    Setup(i)    Runup(i)   Rundown(i)
                [mm]
--------------------------------------------------------------
 Upper    1    21.000       0.051       1.856       -1.357
          2    27.500       0.045       1.852       -1.371
          3    15.000       0.056       1.851       -1.326
          4     5.000       0.069       1.814       -1.194
--------------------------------------------------------------
 Lower    1    21.000       0.108       0.854       -0.292
          2    27.500       0.107       0.860       -0.298
          3    15.000       0.109       0.852       -0.289
          4     5.000       0.110       0.859       -0.290
--------------------------------------------------------------

ARMOR STABILITY

Largest node number for which armor stability
     computation is performed    JSTABM =        232
Critical stability number Nsc      SNSC =          1.431
  occurring at normalized time    TSNSC =        199.115200
             at node number       JSNSC =        165
                               x at JSNSC =          0.845577
                              zb at JSNSC =         -1.575492
At the time of critical stability:
 . Armor stability computation is
   performed up to node           JSATM =        177
 . Upper waterline node           ISATM =        203
 . Lower waterline node           IWATM =        311
```

Table D-3: File ODOC produced by PBREAK computation of Run P3 with ICOMP=4.

```
---------------------------------------------------------------------
          PBREAK              *** Kobayashi, Cox, and Wurjanto 1991
                              *** Run P3
---------------------------------------------------------------------

WAVE CONDITION

Incident wave at seaward boundary given as input
Number of data points     NDATA =     5841
Norm. maximum surface elev.     =        1.094400
Norm. minimum surface elev.     =       -0.797800
Reference wave period           =        1.738000 sec.
Reference wave height           =        0.045700 meters
Depth at seaward boundary       =        0.400000 meters
Norm. depth at seaw. boundary   =        8.753
Normalized wave length          =        7.839
Ursell number                   =        7.021
Surf similarity parameter       =        3.386
"Sigma"                         =       25.464

PROPERTIES OF THE UPPER SLOPE

Friction factor               =       0.100000
Norm. friction factor         =       1.273199
Number of segments            =       1
-------------------------------------------------------
 SEGMENT    WUSEG(I)    TUSEG(I)    XUSEG(I)    ZUSEG(I)
     I       meters                  meters      meters
-------------------------------------------------------
     1     2.400000    0.333333    0.000000   -0.400000
     2                             2.400000    0.399999
-------------------------------------------------------

GEOMETRY OF THE LOWER SLOPE

Number of Segments            =       2
-------------------------------------------------------
 SEGMENT    WLSEG(I)    TLSEG(I)    XLSEG(I)    ZLSEG(I)
     I       meters                  meters      meters
-------------------------------------------------------
     1     0.566049    0.000000    0.000000   -0.400000
     2     1.835000    0.333333    0.566049   -0.400000
     3                             2.401049    0.211666
-------------------------------------------------------

PROPERTIES OF THE PERMEABLE UNDERLAYER and
PARAMETERS RELATED TO PERMEABILITY

Fresh water at temperature          TEMPER =          20.000 degrees C
Kinematic viscosity of water x 1.D+06
                        = 1.D+06 x VISCO =           1.004000 m.m/sec
Representative diameter of permeable
               underlayer materials    DPP =           0.021 meters
Porosity of the permeable underlayer  PORO =           0.480
```

Table D-3 continued.

```
Dimensionless constants                    ALPA0 =          1140.000
                                           BETA0 =             2.700
Coefficient expressing the laminar flow resistance
                       ALPAP [alpha prime] =                   1.584
Coefficient expressing the turbulent flow resistance
                        BETAP [beta prime] =                 604.539
Dimensionless parameters that arise in consequence
    of normalizing the governing equations:
                                           PARPU =         0.785463D-01
                        PARPQ = PORO*PARPU =               0.377022D-01
                                           PARMU =         0.103787D+00

INPUT PARAMETERS FOR ARMOR STABILITY

Specific gravity           SG =      2.700
Armor friction Factor  TANPHI =      1.192
Area coefficient           C2 =      0.900
Volume coefficient         C3 =      0.600
Drag coefficient           CD =      0.500
Lift coefficient           CL =      0.300
Inertia coefficient        CM =      1.500
Upper and lower bounds of du/dt
        (normalized by gravity)
                         AMAX =      1.000
                         AMIN =     -0.800

COMPUTATION PARAMETERS

Total number of spatial nodes        JE =      400
No. of nodes with upper slope below SWL
                                  INITS =      200
No. of nodes with lower slope below SWL
                                  INITW =      295
Minimum allowable value of INVT
                                = MINVT =     1250
  where INVT is the number of time steps
  in one reference wave period
Computed time series are stored at rate NRATE
  per reference wave period with NRATE =        50
Normalized delta x          X =           0.515596D-02
Computation duration  TMAX =            210.011508 ref. wave periods
Statistical calculations are performed excluding
     the first MSTAT computation units
     with                         MSTAT =        0
Instantaneous spatial variations are stored NTIMES times
     at equal intervals per reference wave period from
     computation unit MSAVA1 to MSAVA2, inclusive,
     with                        NTIMES =        4
                                 MSAVA1 =      195
                                 MSAVA2 =      200
Integer multiplication factor    MULTIF =        1
Minimum value of INVT used = MULTIF*MINVT
Damping coefficients              EPSI1 =         1.000000
                                  EPSI2 =         1.000000
Other parameters are given in COMPUTATION LOG below
```

```
Positioning:
. Node      1          is at x =        0.000000
. Node    200 (INITS)  is at x =        1.026035
. Node    295 (INITW)  is at x =        1.515851
. Node    400 (JE)     is at x =        2.057227

COMPUTATION LOG
```

Comp. Unit	# of Tries	INVT	OMEGA	NDIV	DELTAS	DELTAW
1	1	1250	1.0	1250	0.001000	0.001000
2	1	1250	1.0	1250	0.001000	0.001000
3	1	1250	1.0	1250	0.001000	0.001000
4	1	1250	1.0	1250	0.001000	0.001000
5	1	1250	1.0	1250	0.001000	0.001000
6	1	1250	1.0	1250	0.001000	0.001000
7	1	1250	1.0	1250	0.001000	0.001000
8	1	1250	1.0	1250	0.001000	0.001000
9	1	1250	1.0	1250	0.001000	0.001000
10	1	1250	1.0	1250	0.001000	0.001000
11	1	1250	1.0	1250	0.001000	0.001000
12	1	1250	1.0	1250	0.001000	0.001000
13	1	1250	1.0	1250	0.001000	0.001000
14	1	1250	1.0	1250	0.001000	0.001000
15	1	1250	1.0	1250	0.001000	0.001000
16	1	1250	1.0	1250	0.001000	0.001000
17	2	1850	1.5	1850	0.001000	0.001000
18	1	1250	1.0	1250	0.001000	0.001000
19	1	1250	1.0	1250	0.001000	0.001000
20	1	1250	1.0	1250	0.001000	0.001000
21	1	1250	1.0	1250	0.001000	0.001000
22	1	1250	1.0	1250	0.001000	0.001000
23	1	1250	1.0	1250	0.001000	0.001000
24	1	1250	1.0	1250	0.001000	0.001000
25	1	1250	1.0	1250	0.001000	0.001000
26	1	1250	1.0	1250	0.001000	0.001000
27	1	1250	1.0	1250	0.001000	0.001000
28	1	1250	1.0	1250	0.001000	0.001000
29	1	1250	1.0	1250	0.001000	0.001000
30	1	1250	1.0	1250	0.001000	0.001000
31	1	1250	1.0	1250	0.001000	0.001000
32	1	1250	1.0	1250	0.001000	0.001000
33	1	1250	1.0	1250	0.001000	0.001000
34	1	1250	1.0	1250	0.001000	0.001000
35	1	1250	1.0	1250	0.001000	0.001000
36	1	1250	1.0	1250	0.001000	0.001000
37	1	1250	1.0	1250	0.001000	0.001000
38	1	1250	1.0	1250	0.001000	0.001000
39	1	1250	1.0	1250	0.001000	0.001000
40	2	1850	1.5	1850	0.001000	0.001000
41	1	1250	1.0	1250	0.001000	0.001000
42	1	1250	1.0	1250	0.001000	0.001000
43	1	1250	1.0	1250	0.001000	0.001000
44	1	1250	1.0	1250	0.001000	0.001000

Table D-3 continued.

166	45	1	1250	1.0	1250	0.001000	0.001000
167	46	1	1250	1.0	1250	0.001000	0.001000
168	47	1	1250	1.0	1250	0.001000	0.001000
169	48	1	1250	1.0	1250	0.001000	0.001000
170	49	1	1250	1.0	1250	0.001000	0.001000
171	50	1	1250	1.0	1250	0.001000	0.001000
172	51	2	1850	1.5	1850	0.001000	0.001000
173	52	1	1250	1.0	1250	0.001000	0.001000
174	53	1	1250	1.0	1250	0.001000	0.001000
175	54	1	1250	1.0	1250	0.001000	0.001000
176	55	1	1250	1.0	1250	0.001000	0.001000
177	56	1	1250	1.0	1250	0.001000	0.001000
178	57	1	1250	1.0	1250	0.001000	0.001000
179	58	1	1250	1.0	1250	0.001000	0.001000
180	59	1	1250	1.0	1250	0.001000	0.001000
181	60	1	1250	1.0	1250	0.001000	0.001000
182	61	2	1850	1.5	1850	0.001000	0.001000
183	62	1	1250	1.0	1250	0.001000	0.001000
184	63	1	1250	1.0	1250	0.001000	0.001000
185	64	1	1250	1.0	1250	0.001000	0.001000
186	65	1	1250	1.0	1250	0.001000	0.001000
187	66	1	1250	1.0	1250	0.001000	0.001000
188	67	1	1250	1.0	1250	0.001000	0.001000
189	68	1	1250	1.0	1250	0.001000	0.001000
190	69	1	1250	1.0	1250	0.001000	0.001000
191	70	1	1250	1.0	1250	0.001000	0.001000
192	71	1	1250	1.0	1250	0.001000	0.001000
193	72	2	1850	1.5	1850	0.001000	0.001000
194	73	1	1250	1.0	1250	0.001000	0.001000
195	74	1	1250	1.0	1250	0.001000	0.001000
196	75	1	1250	1.0	1250	0.001000	0.001000
197	76	1	1250	1.0	1250	0.001000	0.001000
198	77	1	1250	1.0	1250	0.001000	0.001000
199	78	1	1250	1.0	1250	0.001000	0.001000
200	79	1	1250	1.0	1250	0.001000	0.001000
201	80	2	1850	1.5	1850	0.001000	0.001000
202	81	1	1250	1.0	1250	0.001000	0.001000
203	82	1	1250	1.0	1250	0.001000	0.001000
204	83	1	1250	1.0	1250	0.001000	0.001000
205	84	1	1250	1.0	1250	0.001000	0.001000
206	85	1	1250	1.0	1250	0.001000	0.001000
207	86	1	1250	1.0	1250	0.001000	0.001000
208	87	1	1250	1.0	1250	0.001000	0.001000
209	88	3	2750	2.2	2750	0.001000	0.001000
210	89	1	1250	1.0	1250	0.001000	0.001000
211	90	1	1250	1.0	1250	0.001000	0.001000
212	91	1	1250	1.0	1250	0.001000	0.001000
213	92	2	1850	1.5	1850	0.001000	0.001000
214	93	1	1250	1.0	1250	0.001000	0.001000
215	94	1	1250	1.0	1250	0.001000	0.001000
216	95	1	1250	1.0	1250	0.001000	0.001000
217	96	1	1250	1.0	1250	0.001000	0.001000
218	97	1	1250	1.0	1250	0.001000	0.001000
219	98	2	1850	1.5	1850	0.001000	0.001000
220	99	1	1250	1.0	1250	0.001000	0.001000
221	100	1	1250	1.0	1250	0.001000	0.001000

Table D-3 continued.

222	101	2	1850	1.5	1850	0.001000	0.001000
223	102	1	1250	1.0	1250	0.001000	0.001000
224	103	1	1250	1.0	1250	0.001000	0.001000
225	104	1	1250	1.0	1250	0.001000	0.001000
226	105	1	1250	1.0	1250	0.001000	0.001000
227	106	1	1250	1.0	1250	0.001000	0.001000
228	107	1	1250	1.0	1250	0.001000	0.001000
229	108	1	1250	1.0	1250	0.001000	0.001000
230	109	2	1850	1.5	889	0.001000	0.001000
231					1850	0.001000	0.000943
232	110	1	1250	1.0	1250	0.001000	0.000943
233	111	1	1250	1.0	1250	0.001000	0.000943
234	112	1	1250	1.0	1250	0.001000	0.000943
235	113	1	1250	1.0	1250	0.001000	0.000943
236	114	1	1250	1.0	1250	0.001000	0.000943
237	115	1	1250	1.0	1250	0.001000	0.000943
238	116	2	1850	1.5	1850	0.001000	0.000943
239	117	1	1250	1.0	1250	0.001000	0.000943
240	118	1	1250	1.0	1250	0.001000	0.000943
241	119	2	1850	1.5	1850	0.001000	0.000943
242	120	1	1250	1.0	1250	0.001000	0.000943
243	121	1	1250	1.0	1250	0.001000	0.000943
244	122	1	1250	1.0	1250	0.001000	0.000943
245	123	1	1250	1.0	210	0.001000	0.000943
246					1250	0.001000	0.000895
247	124	2	1850	1.5	1850	0.001000	0.000895
248	125	1	1250	1.0	1250	0.001000	0.000895
249	126	1	1250	1.0	1250	0.001000	0.000895
250	127	1	1250	1.0	1250	0.001000	0.000895
251	128	1	1250	1.0	1250	0.001000	0.000895
252	129	1	1250	1.0	1250	0.001000	0.000895
253	130	1	1250	1.0	1250	0.001000	0.000895
254	131	1	1250	1.0	1250	0.001000	0.000895
255	132	1	1250	1.0	1250	0.001000	0.000895
256	133	1	1250	1.0	1250	0.001000	0.000895
257	134	1	1250	1.0	1250	0.001000	0.000895
258	135	1	1250	1.0	1250	0.001000	0.000895
259	136	1	1250	1.0	1250	0.001000	0.000895
260	137	1	1250	1.0	1250	0.001000	0.000895
261	138	1	1250	1.0	1250	0.001000	0.000895
262	139	1	1250	1.0	1250	0.001000	0.000895
263	140	1	1250	1.0	1250	0.001000	0.000895
264	141	1	1250	1.0	1250	0.001000	0.000895
265	142	1	1250	1.0	1250	0.001000	0.000895
266	143	1	1250	1.0	1250	0.001000	0.000895
267	144	1	1250	1.0	1250	0.001000	0.000895
268	145	1	1250	1.0	1250	0.001000	0.000895
269	146	1	1250	1.0	1250	0.001000	0.000895
270	147	1	1250	1.0	1250	0.001000	0.000895
271	148	1	1250	1.0	1250	0.001000	0.000895
272	149	1	1250	1.0	1250	0.001000	0.000895
273	150	1	1250	1.0	468	0.001000	0.000895
274					1250	0.001000	0.000825
275	151	1	1250	1.0	1250	0.001000	0.000825
276	152	1	1250	1.0	1250	0.001000	0.000825
277	153	1	1250	1.0	1250	0.001000	0.000825

Table D-3 continued.

278	154	1	1250	1.0	1250	0.001000	0.000825
279	155	1	1250	1.0	1250	0.001000	0.000825
280	156	2	1850	1.5	1850	0.001000	0.000825
281	157	1	1250	1.0	1250	0.001000	0.000825
282	158	1	1250	1.0	1250	0.001000	0.000825
283	159	1	1250	1.0	1250	0.001000	0.000825
284	160	2	1850	1.5	1850	0.001000	0.000825
285	161	1	1250	1.0	1250	0.001000	0.000825
286	162	1	1250	1.0	1250	0.001000	0.000825
287	163	1	1250	1.0	1250	0.001000	0.000825
288	164	1	1250	1.0	1250	0.001000	0.000825
289	165	1	1250	1.0	1250	0.001000	0.000825
290	166	1	1250	1.0	1250	0.001000	0.000825
291	167	1	1250	1.0	1250	0.001000	0.000825
292	168	1	1250	1.0	1250	0.001000	0.000825
293	169	1	1250	1.0	1250	0.001000	0.000825
294	170	1	1250	1.0	1250	0.001000	0.000825
295	171	1	1250	1.0	1250	0.001000	0.000825
296	172	1	1250	1.0	1250	0.001000	0.000825
297	173	1	1250	1.0	1250	0.001000	0.000825
298	174	1	1250	1.0	1250	0.001000	0.000825
299	175	1	1250	1.0	1250	0.001000	0.000825
300	176	1	1250	1.0	1250	0.001000	0.000825
301	177	1	1250	1.0	1250	0.001000	0.000825
302	178	1	1250	1.0	1250	0.001000	0.000825
303	179	1	1250	1.0	1250	0.001000	0.000825
304	180	1	1250	1.0	1250	0.001000	0.000825
305	181	1	1250	1.0	1250	0.001000	0.000825
306	182	1	1250	1.0	1250	0.001000	0.000825
307	183	1	1250	1.0	1250	0.001000	0.000825
308	184	1	1250	1.0	1250	0.001000	0.000825
309	185	2	1850	1.5	1850	0.001000	0.000825
310	186	1	1250	1.0	1250	0.001000	0.000825
311	187	1	1250	1.0	1250	0.001000	0.000825
312	188	1	1250	1.0	1250	0.001000	0.000825
313	189	1	1250	1.0	1250	0.001000	0.000825
314	190	1	1250	1.0	1250	0.001000	0.000825
315	191	1	1250	1.0	1250	0.001000	0.000825
316	192	1	1250	1.0	1250	0.001000	0.000825
317	193	1	1250	1.0	1250	0.001000	0.000825
318	194	1	1250	1.0	1250	0.001000	0.000825
319	195	1	1250	1.0	1250	0.001000	0.000825
320	196	1	1250	1.0	1250	0.001000	0.000825
321	197	6	9200	7.4	9200	0.001000	0.000825
322	198	1	1250	1.0	1250	0.001000	0.000825
323	199	1	1250	1.0	1250	0.001000	0.000825
324	200	1	1250	1.0	1250	0.001000	0.000825
325	201	1	1250	1.0	1250	0.001000	0.000825
326	202	1	1250	1.0	1250	0.001000	0.000825
327	203	1	1250	1.0	1250	0.001000	0.000825
328	204	1	1250	1.0	1250	0.001000	0.000825
329	205	1	1250	1.0	1250	0.001000	0.000825
330	206	1	1250	1.0	1250	0.001000	0.000825
331	207	1	1250	1.0	1250	0.001000	0.000825
332	208	1	1250	1.0	1250	0.001000	0.000825
333	209	1	1250	1.0	1250	0.001000	0.000825

Table D-3 continued.

```
      210       1    1250     1.0    1250  0.001000  0.000825
      211       1    1250     1.0      14  0.001000  0.000825
----------------------------------------------------------------

STATISTICAL CALCULATIONS

Statistical calculations begins at   t=TSTAT1 =      1.000000
Statistical calculations ends at     t=TSTAT2 =    210.011200
Duration of statistical calculations    TSPAN =    209.011200
Computational waterlines at t=TSTAT1:
                        . Upper slope: IST1 =     200
                        . Lower slope: IWT1 =     295
Computational waterlines at t=TSTAT2:
                        . Upper slope: IST2 =     189
                        . Lower slope: IWT2 =     299

ELEVATIONS AT SEAWARD BOUNDARY

Time-averaged elevation due to incident waves =      0.000000
Elevations due to reflected waves:
                            . Time-averaged =        0.012634
                            . Maximum       =        0.259082
                            . Minimum       =       -0.249155

RUN-UP, RUN-DOWN, SET-UP

Smallest nodes reached by computational waterlines:
                       . Upper slope: ISMIN =     174
                       . Lower slope: IWMIN =     289
Largest nodes reached by computational waterlines:
                       . Upper slope: ISMAX =     241
                       . Lower slope: IWMAX =     315
---------------------------------------------------------------
 Slope   i   Deltar(i)   Setup(i)   Runup(i)   Rundown(i)
               [mm]
---------------------------------------------------------------
 Upper   1    21.000      0.051      1.856      -1.357
         2    27.500      0.045      1.852      -1.371
         3    15.000      0.056      1.851      -1.326
         4     5.000      0.069      1.814      -1.194
---------------------------------------------------------------
 Lower   1    21.000      0.108      0.854      -0.292
         2    27.500      0.107      0.860      -0.298
         3    15.000      0.109      0.852      -0.289
         4     5.000      0.110      0.859      -0.290
---------------------------------------------------------------
```

Table D-3 continued.

```

ARMOR STABILITY

Largest node number for which armor stability
     computation is performed   JSTABM =       232
Critical stability number Nsc      SNSC =        1.431
  occurring at normalized time    TSNSC =      199.115200
                at node number    JSNSC =      165
                       x at JSNSC =        0.845577
                      zb at JSNSC =       -1.575492
At the time of critical stability:
 . Armor stability computation is
   performed up to node           JSATM =      177
 . Upper waterline node           ISATM =      203
 . Lower waterline node           IWATM =      311
```

Table D-4: List of figures for Run P3 presented in Appendix D.

FIGURE	DESCRIPTION	COUNTERPART IN APPENDIX B (RUN P1)
D-1	Incident wave train η_i at the seaward boundary	Figure B-2
D-2	Reflected wave train η_r at the seaward boundary, computed and measured	Figure B-3
D-3	Time series of surface elevation at four different locations: $x = 0.26, 0.52, 0.77, 1.03$	Figure B-4
D-4	Time series of run-up height on the *upper* slope Z_r^U, computed and measured	Figure B-5
D-5	Time series of run-up height on the *lower* slope Z_r^L	Figure B-6
D-6	Instantaneous spatial variations of η and u	Figure B-7
D-7	Instantaneous spatial variations of q_b and u_b	Figure B-8
D-8	Instantaneous spatial variations of u_p and m_p	Figure B-9
D-9	Mean, maximum, and minimum values of η	Figure B-10
D-10	Mean, maximum, and minimum values of u	Figure B-11
D-11	Mean, maximum, and minimum values of m	Figure B-12
D-12	Mean, maximum, and minimum values of q_b	Figure B-13
D-13	Mean, maximum, and minimum values of u_p	Figure B-14
D-14	Mean, maximum, and minimum values of m_p	Figure B-15
D-15	Mass balance for Region 1	Figure B-16
D-16	Mass balance for Regions 2+3	Figure B-17
D-17	Time-averaged energy quantities for Region 1	Figure B-18
D-18	Time-averaged energy quantities for Regions 2+3	Figure B-19
D-19	Energy balance for Regions 2+3	Figure B-20
D-20	Spatial variation of the local stability number N_{sx}	Figure B-21
D-21	Spatial variations of various quantities at the time of the critical armor stability	Figure B-22

Run P3

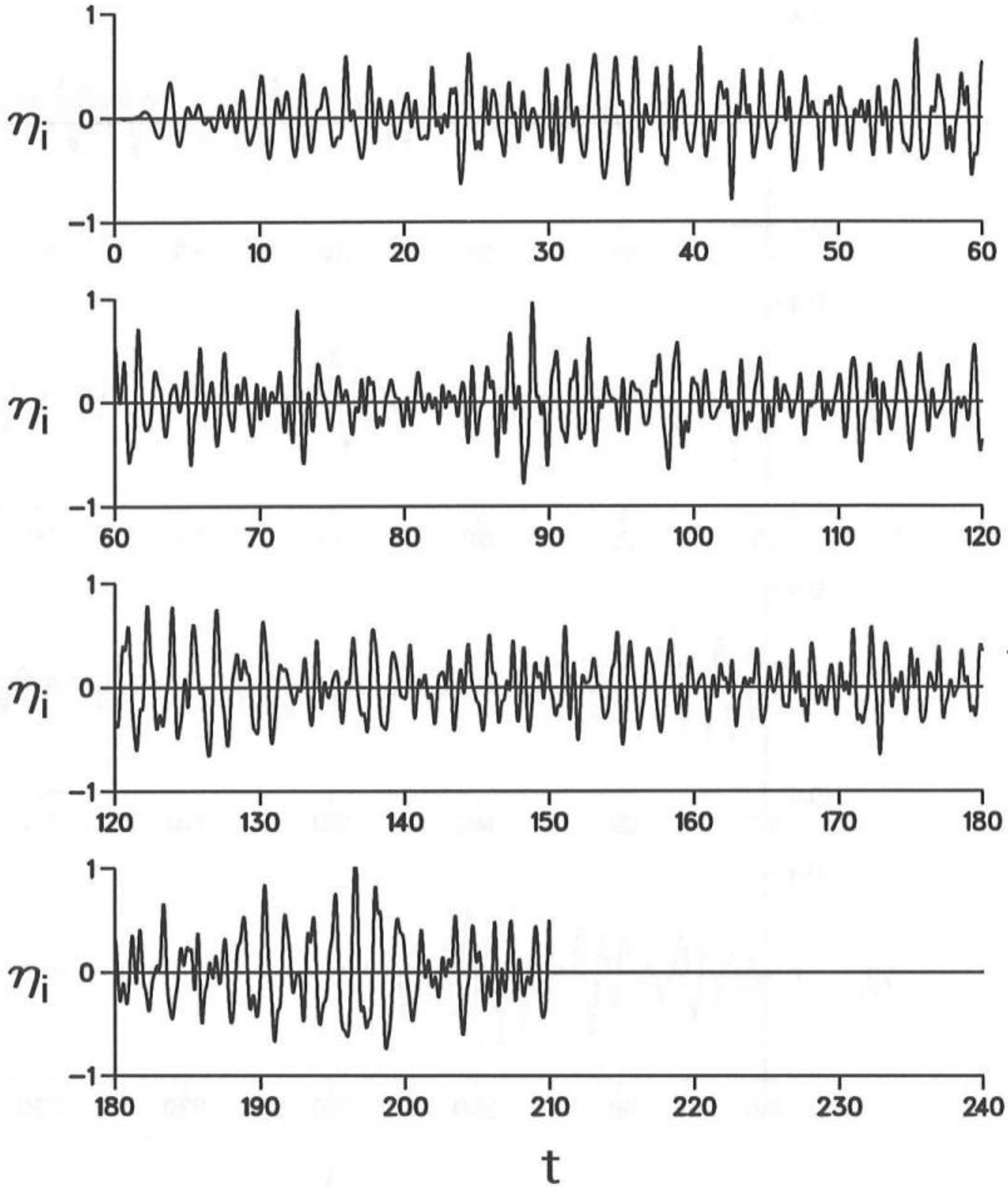


Figure D-1

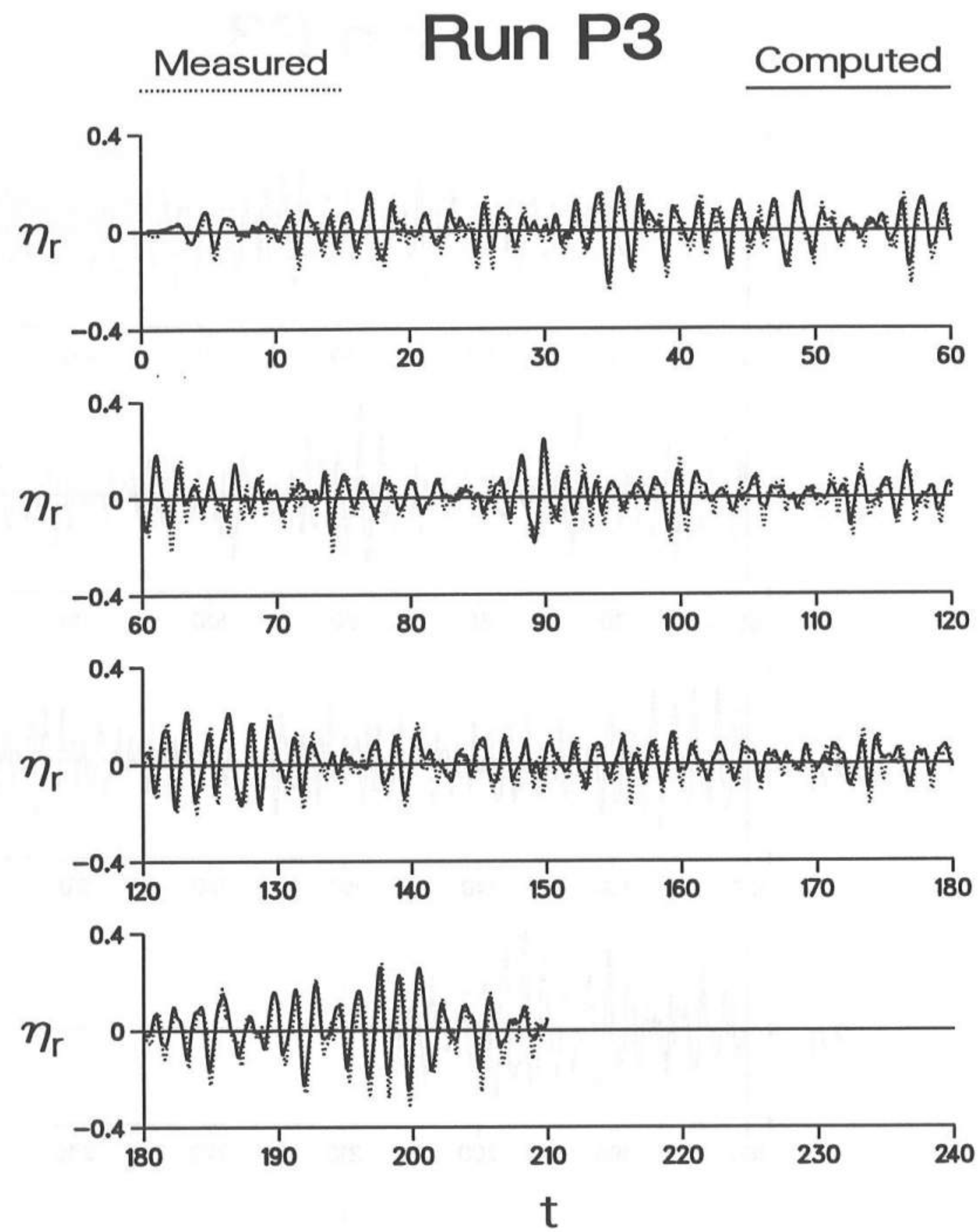

Run P3
Measured
Computed
ηr
0.4
0
−0.4
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
t

Figure D-2

Run P3

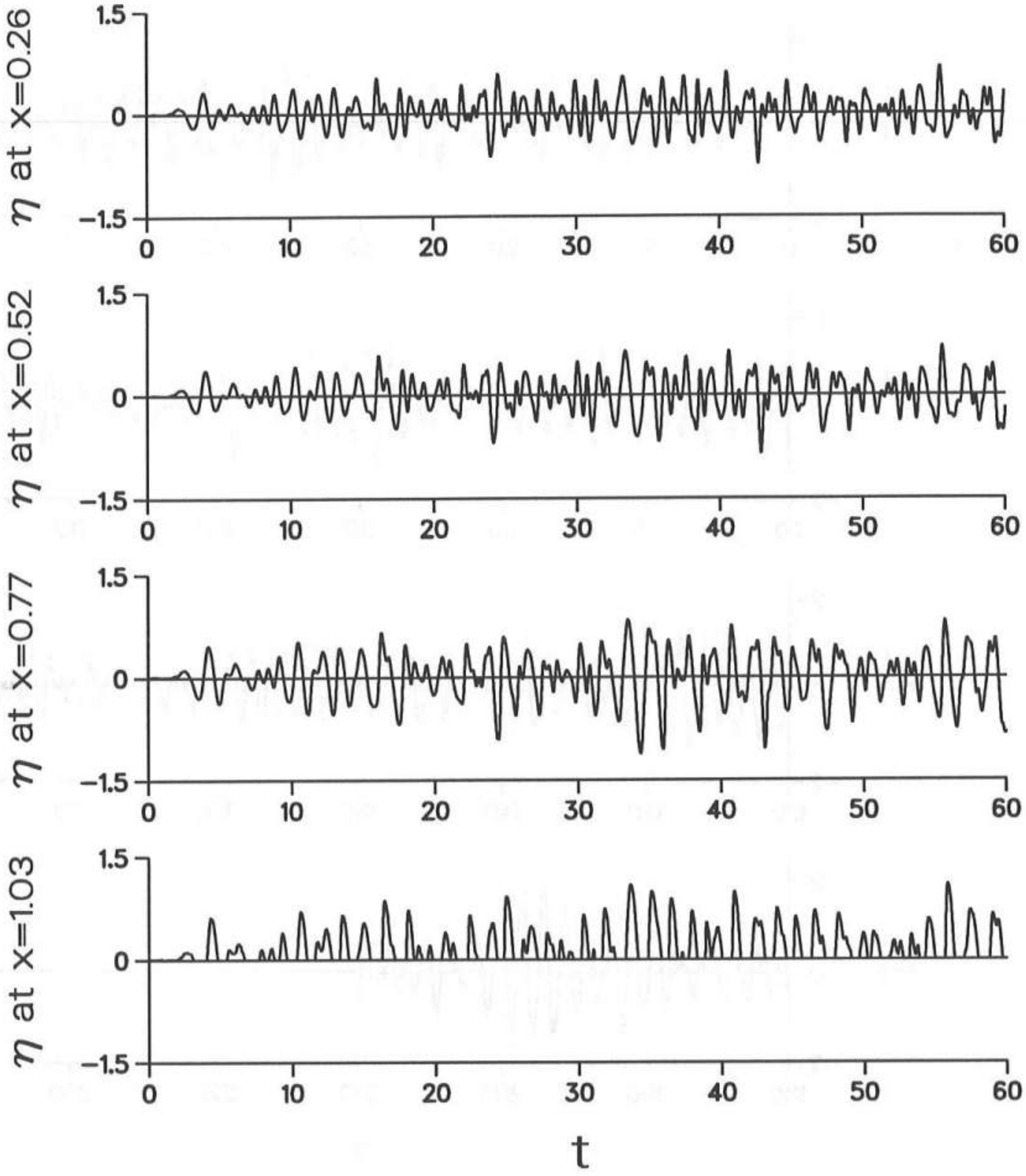


Figure D-3

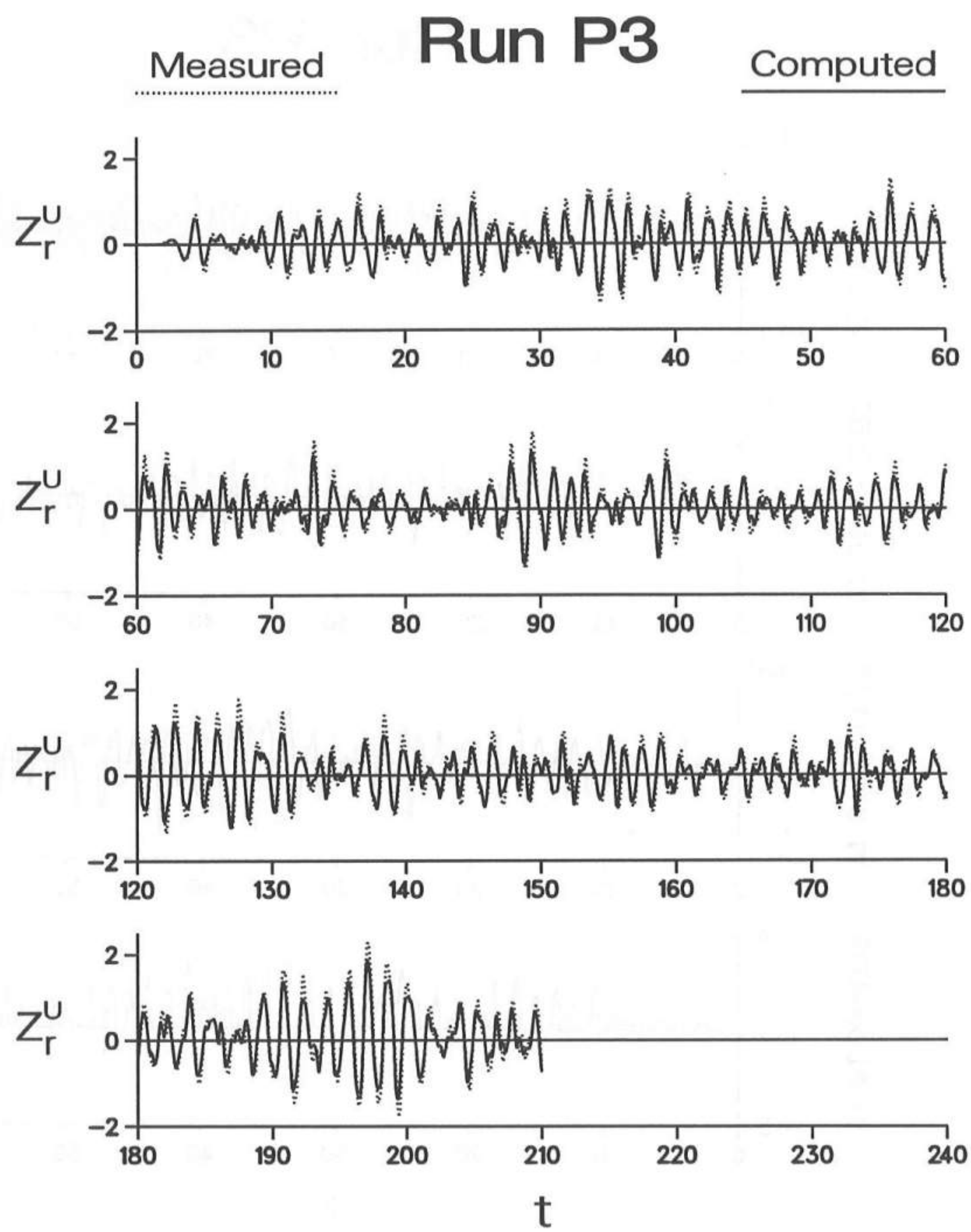

Run P3
Measured
Computed
Z_r^U
2
0
−2
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
t

Figure D-4

Run P3

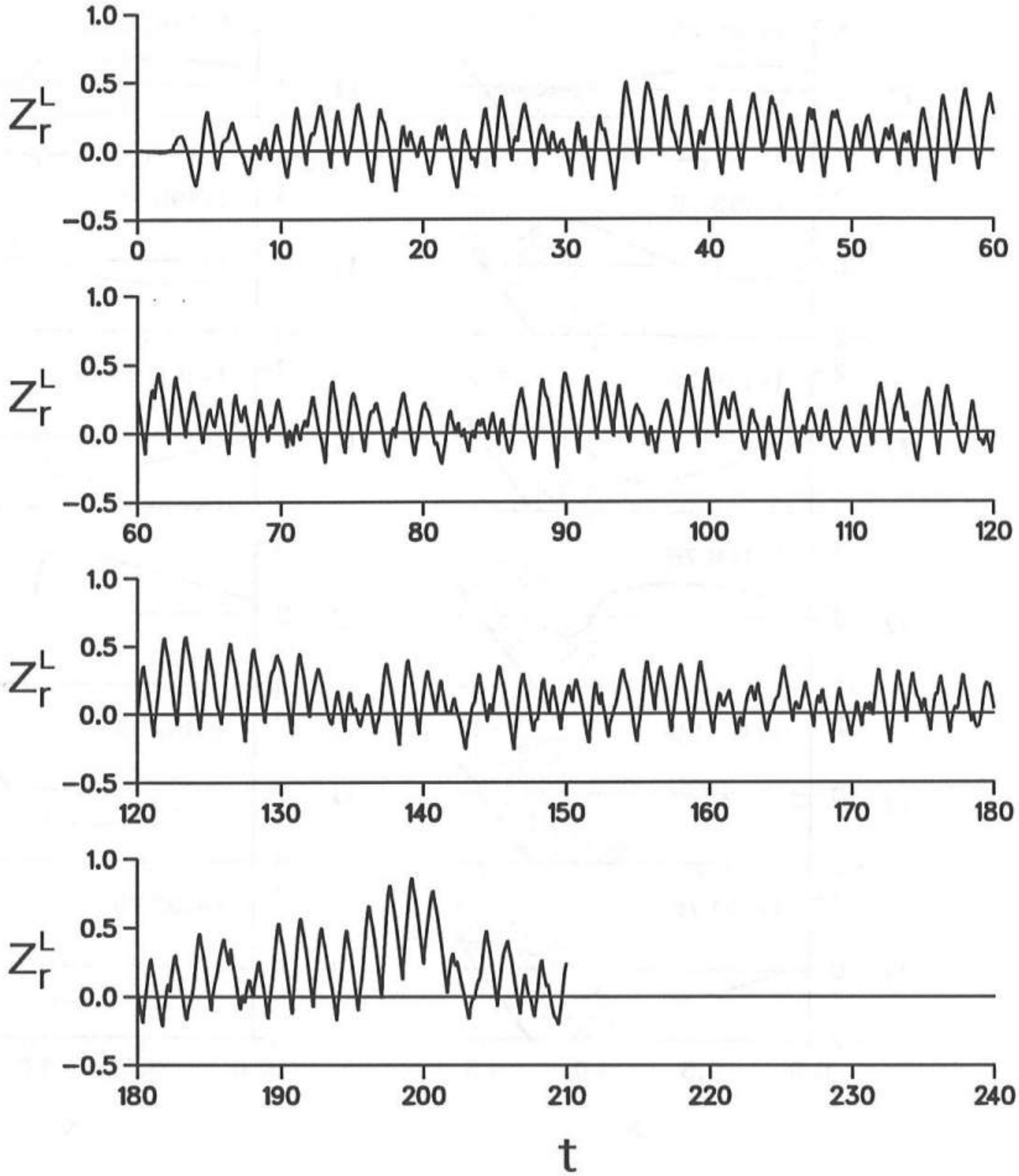


Figure D-5

Run P3

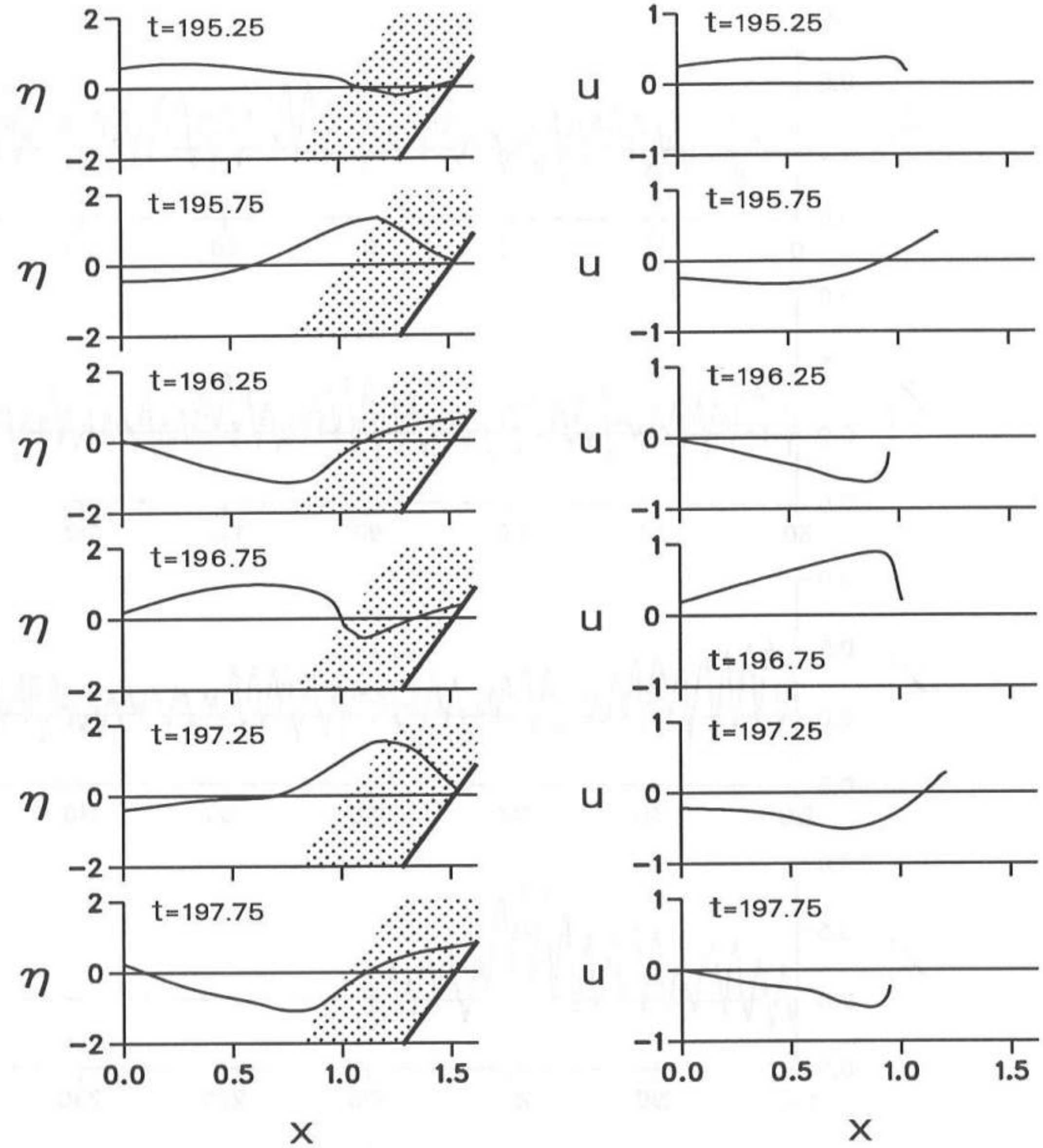


Figure D-6

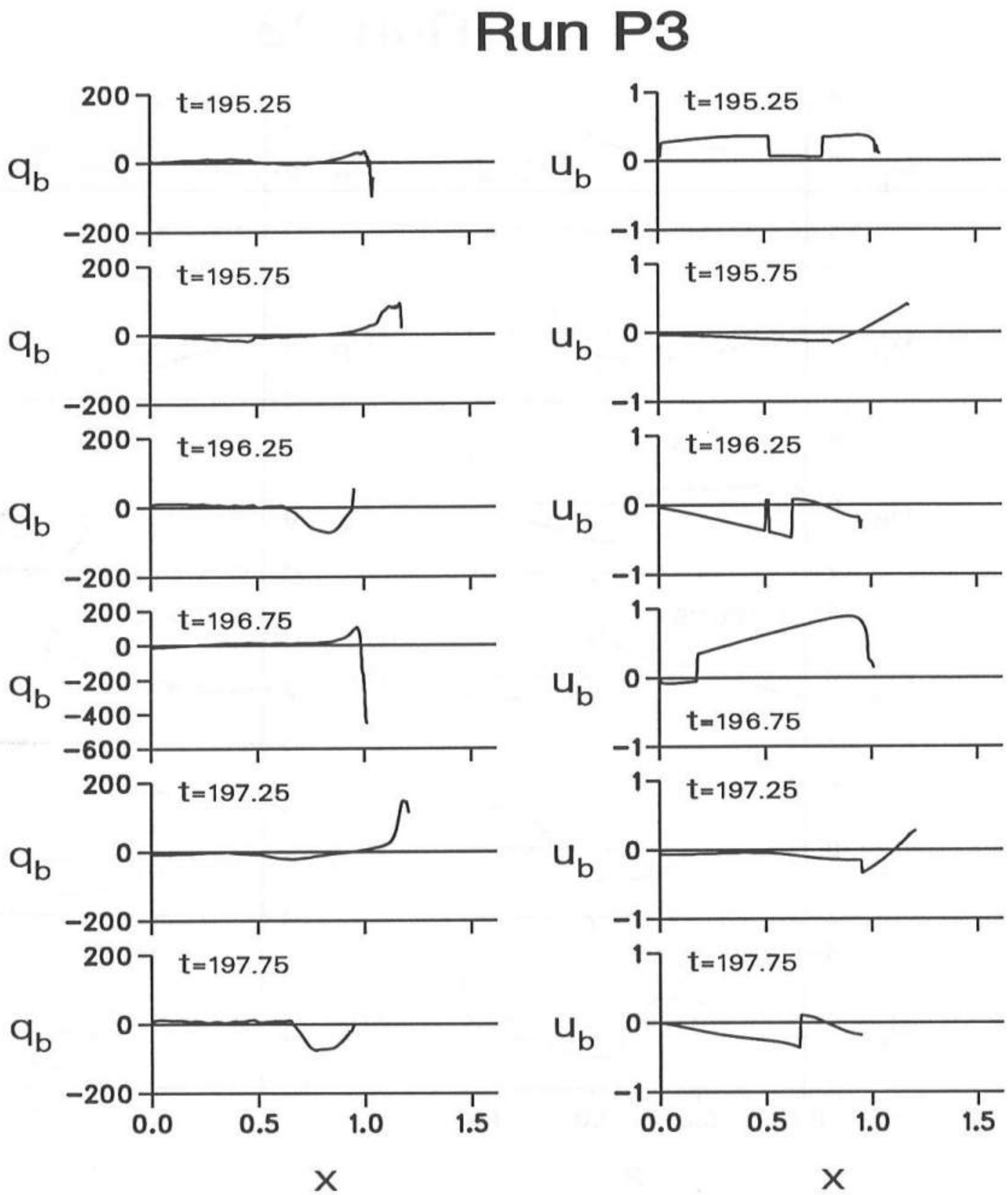

Run P3
t=195.25
t=195.75
t=196.25
t=196.75
t=197.25
t=197.75
q_b
u_b
200
0
-200
-400
-600
1
0
-1
0.0
0.5
1.0
1.5
x

Figure D-7

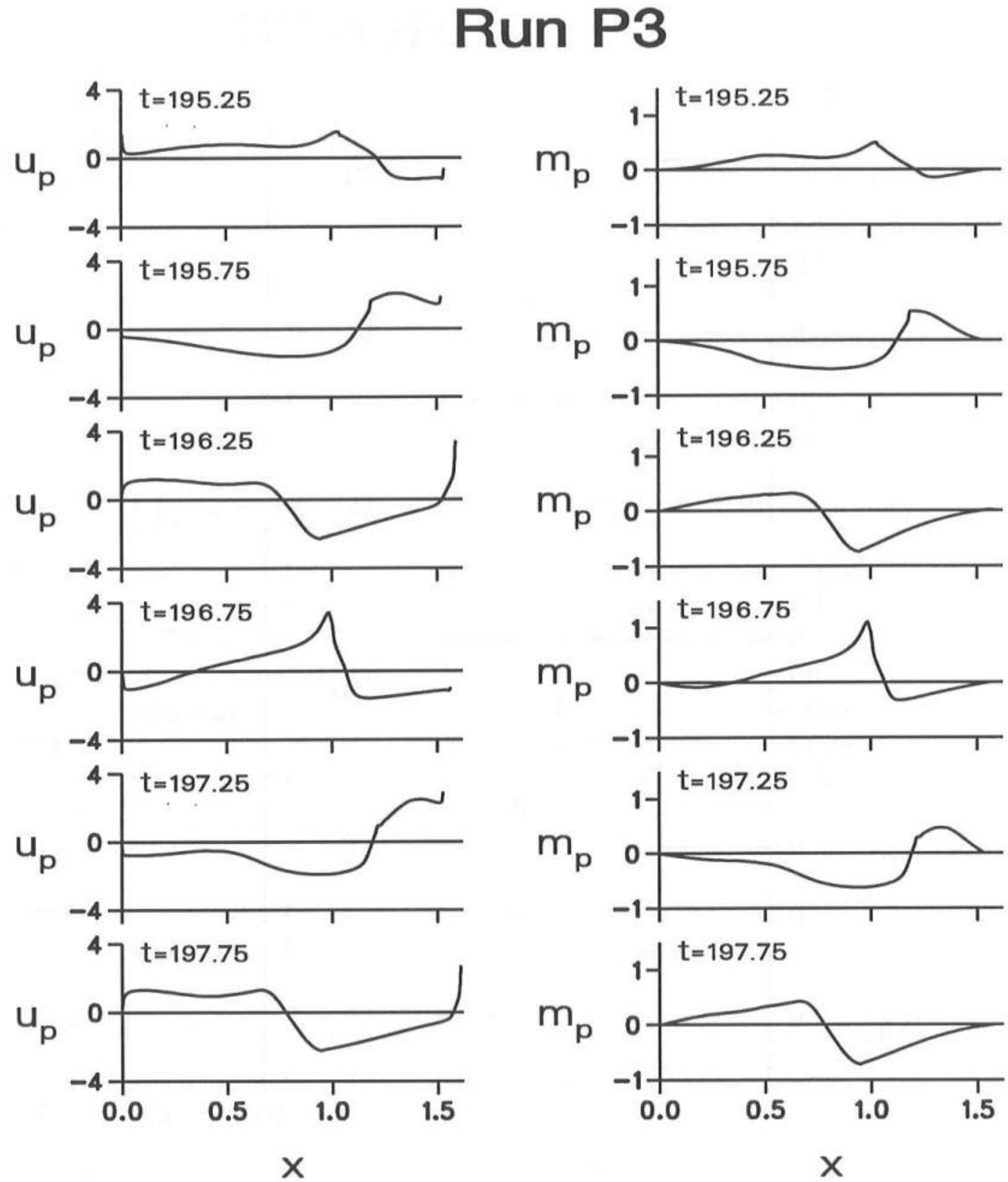

Run P3
t=195.25
t=195.75
t=196.25
t=196.75
t=197.25
t=197.75
u_p
m_p
4
0
-4
1
0
-1
0.0
0.5
1.0
1.5
x

Figure D-8

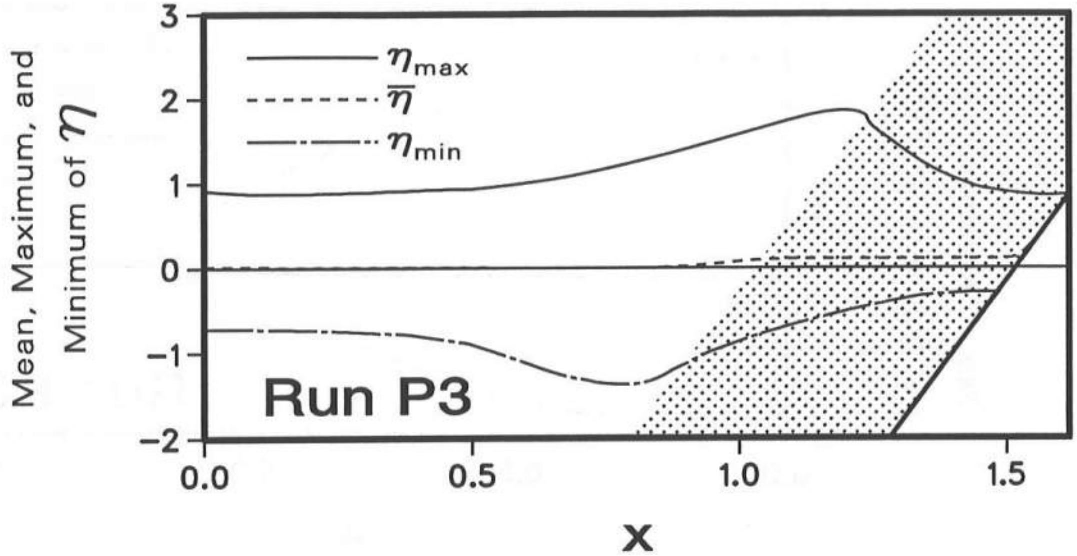


Figure D-9

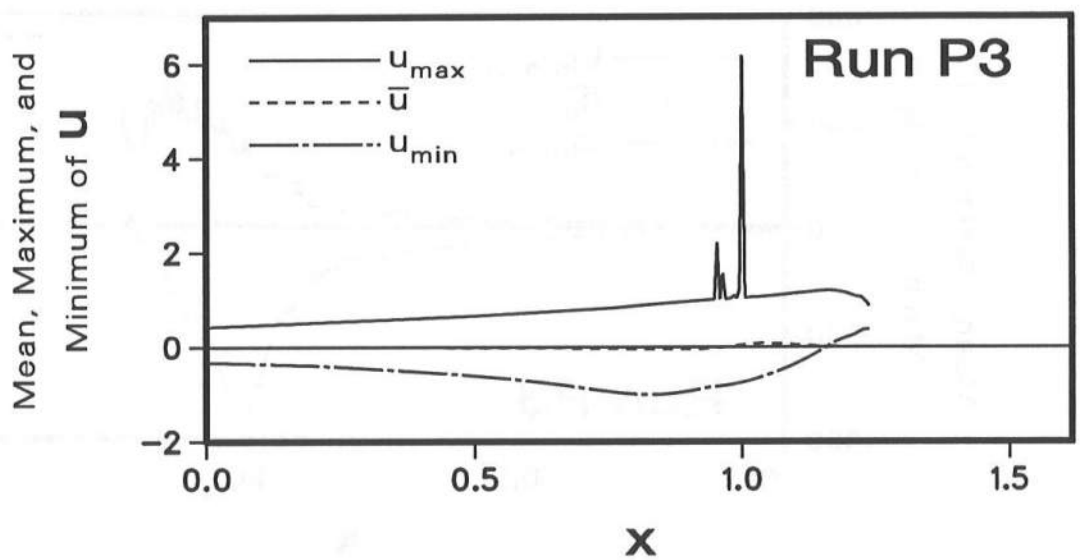


Figure D-10

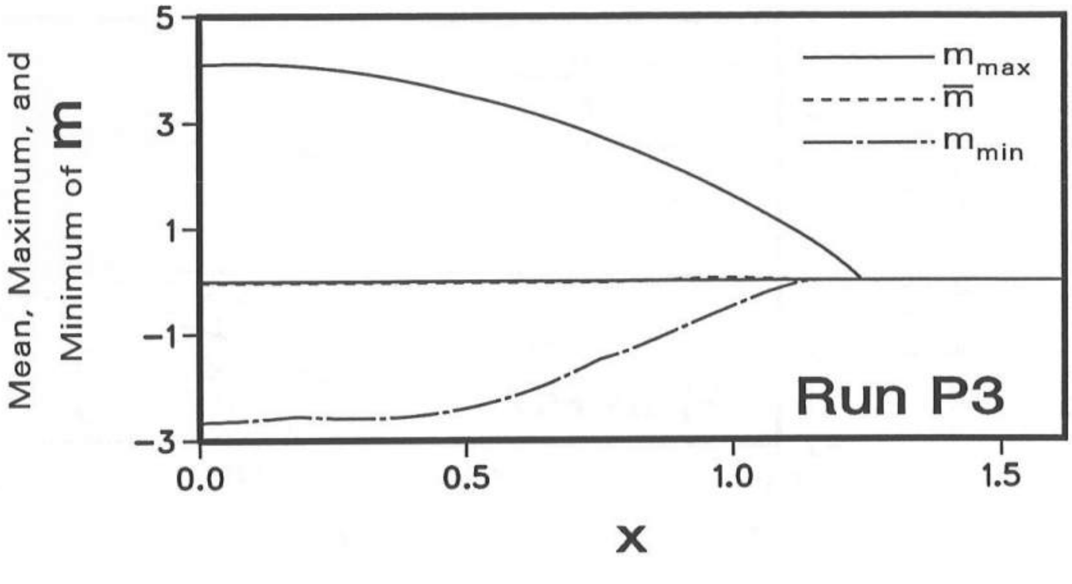


Figure D-11

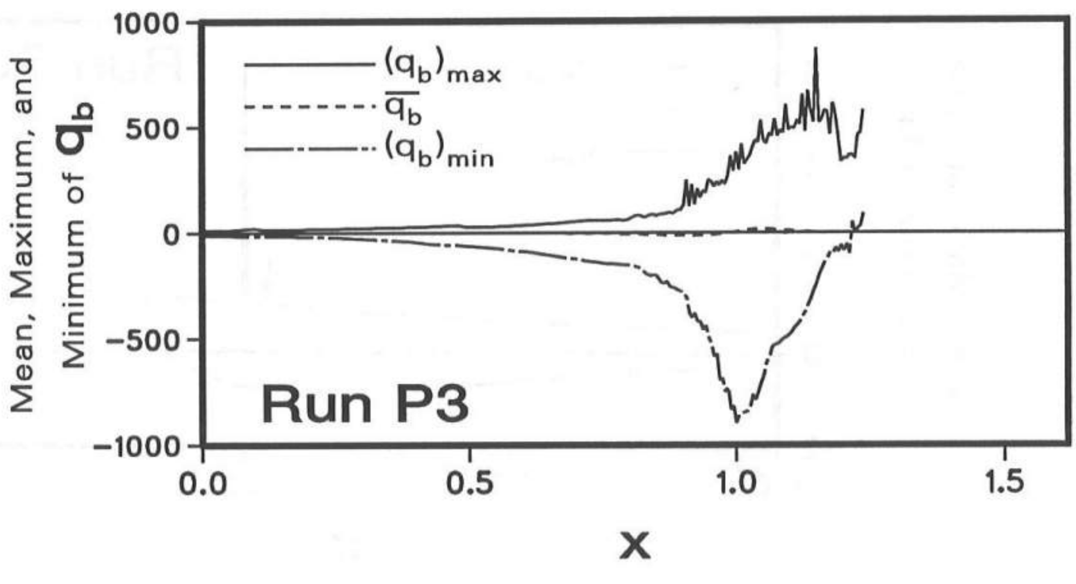


Figure D-12

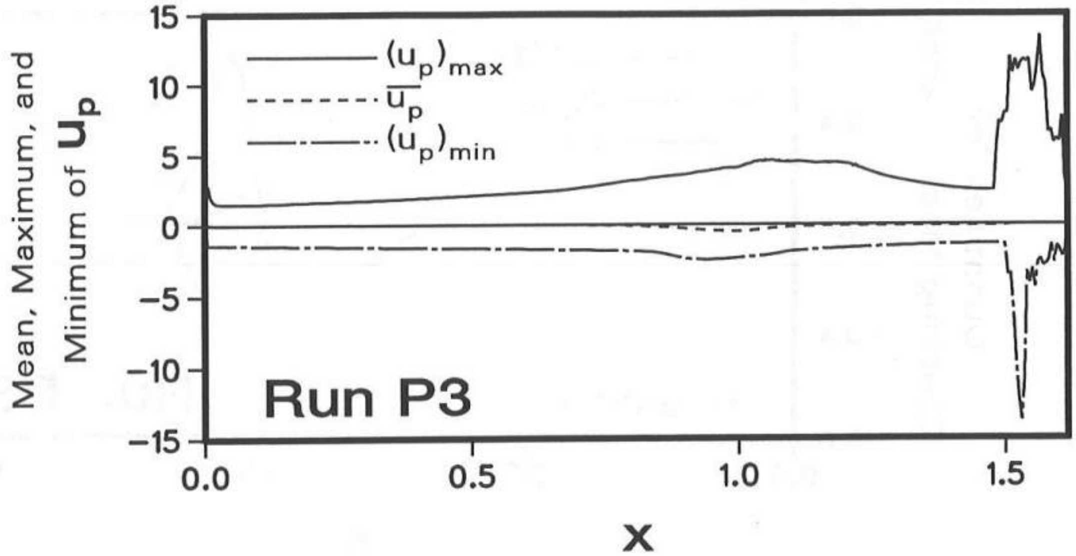


Figure D-13

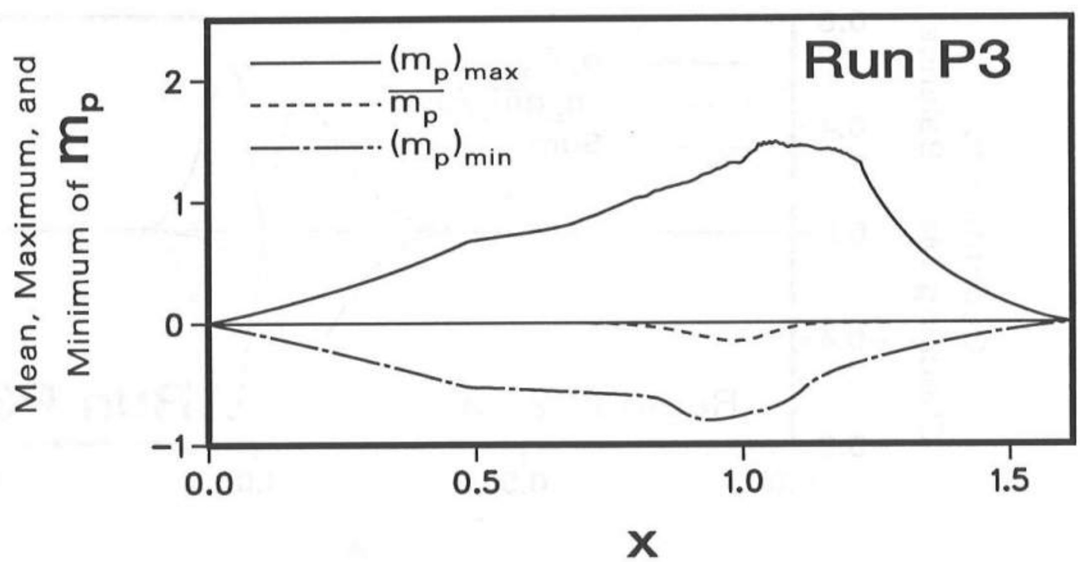


Figure D-14

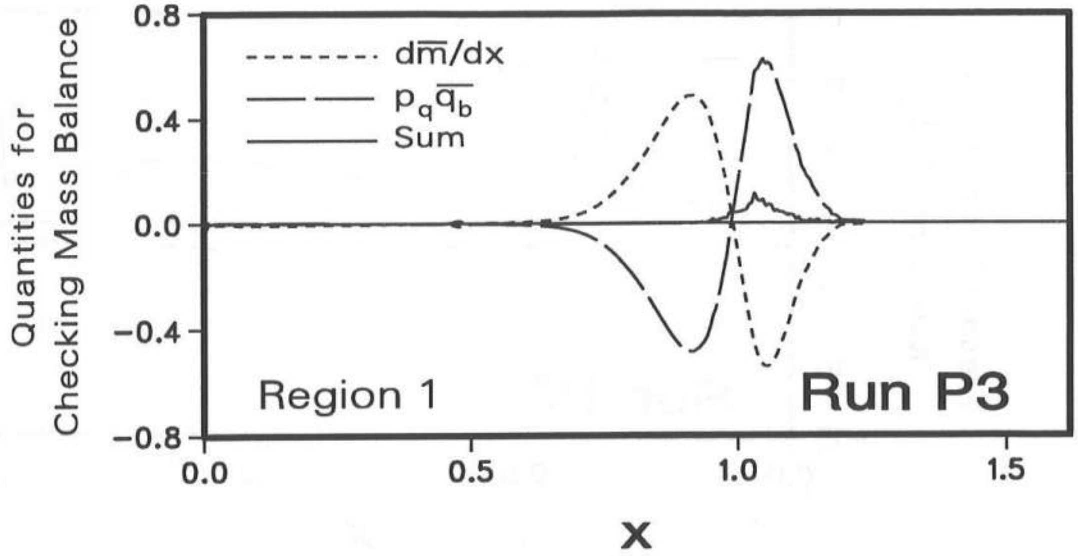


Figure D-15

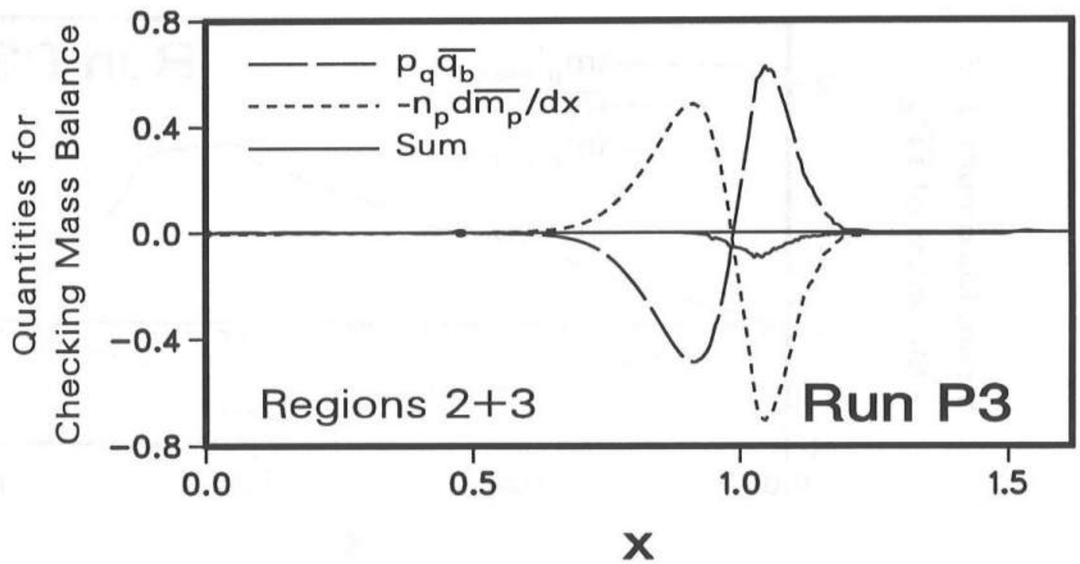


Figure D-16

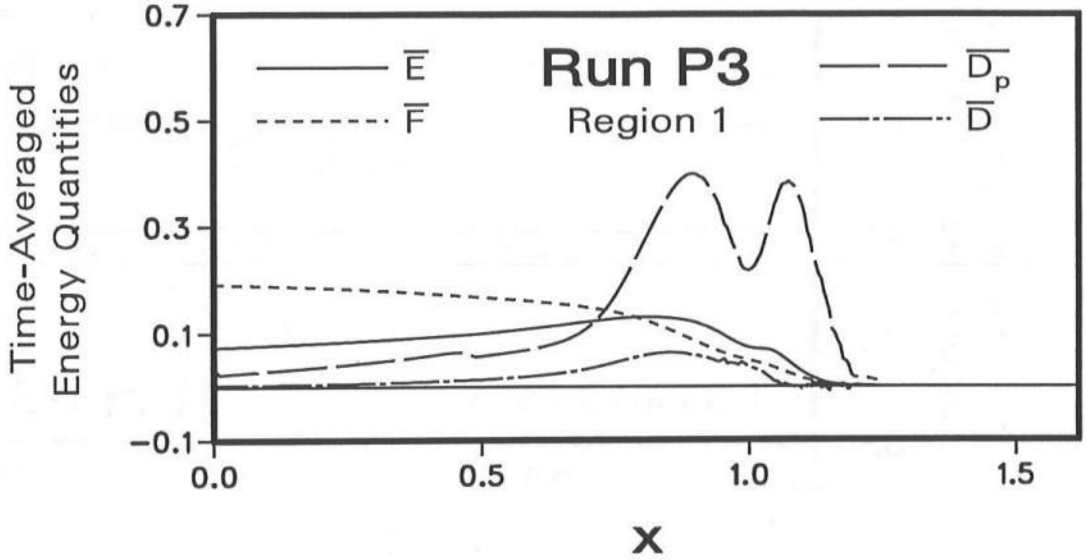


Figure D-17

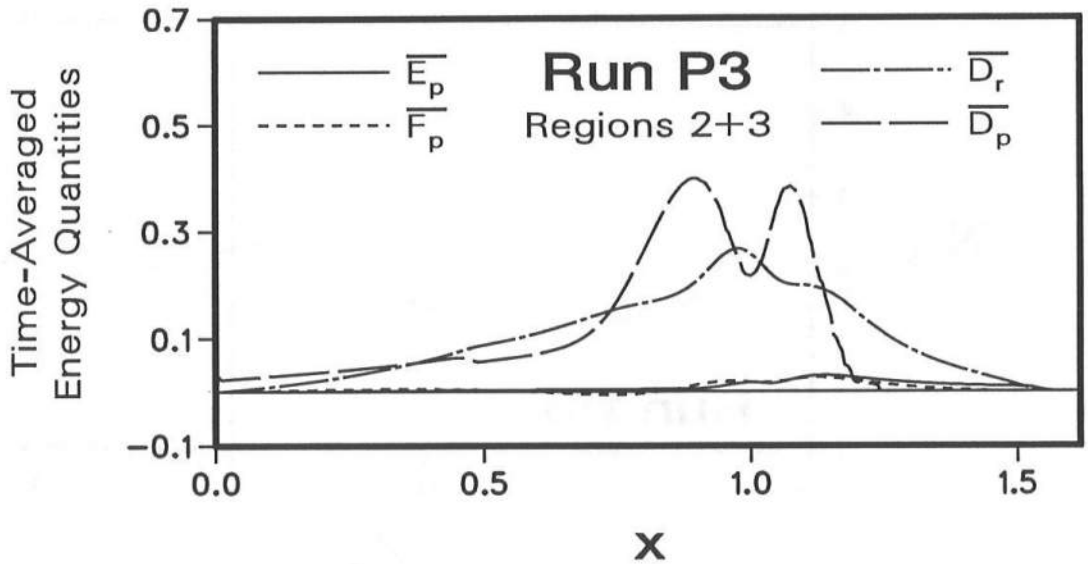


Figure D-18

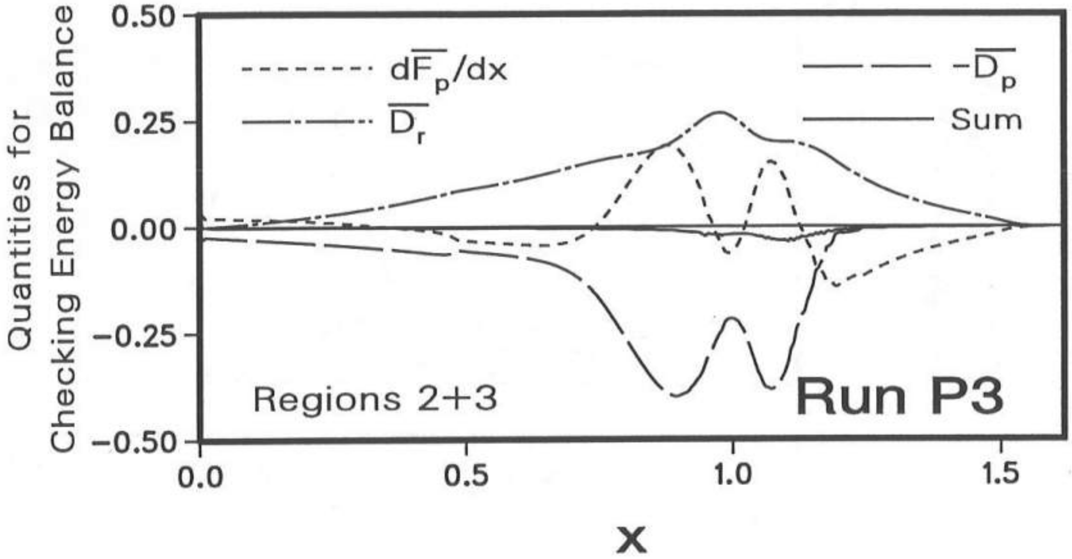


Figure D-19

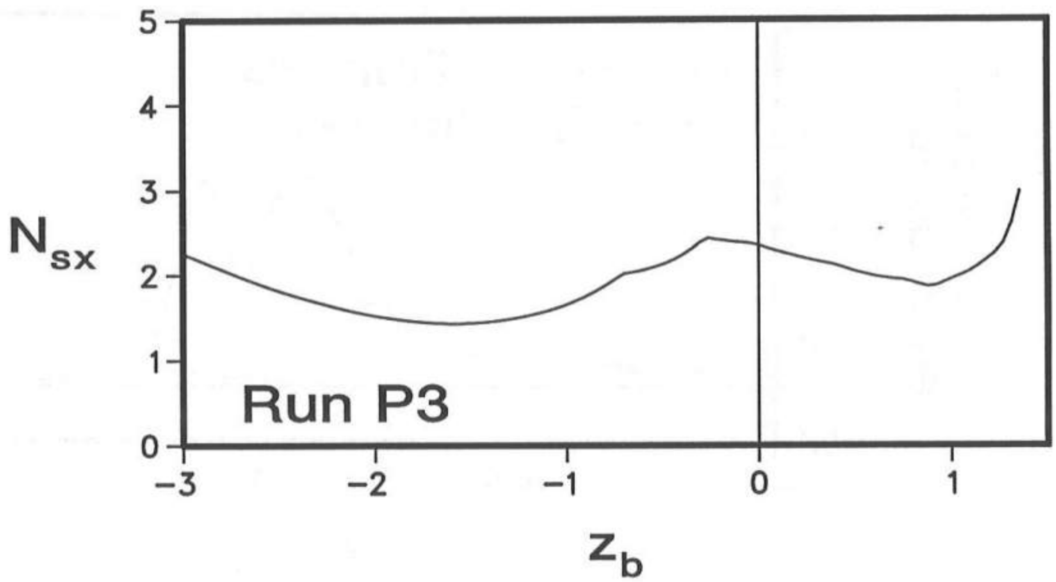


Figure D-20

Run P3

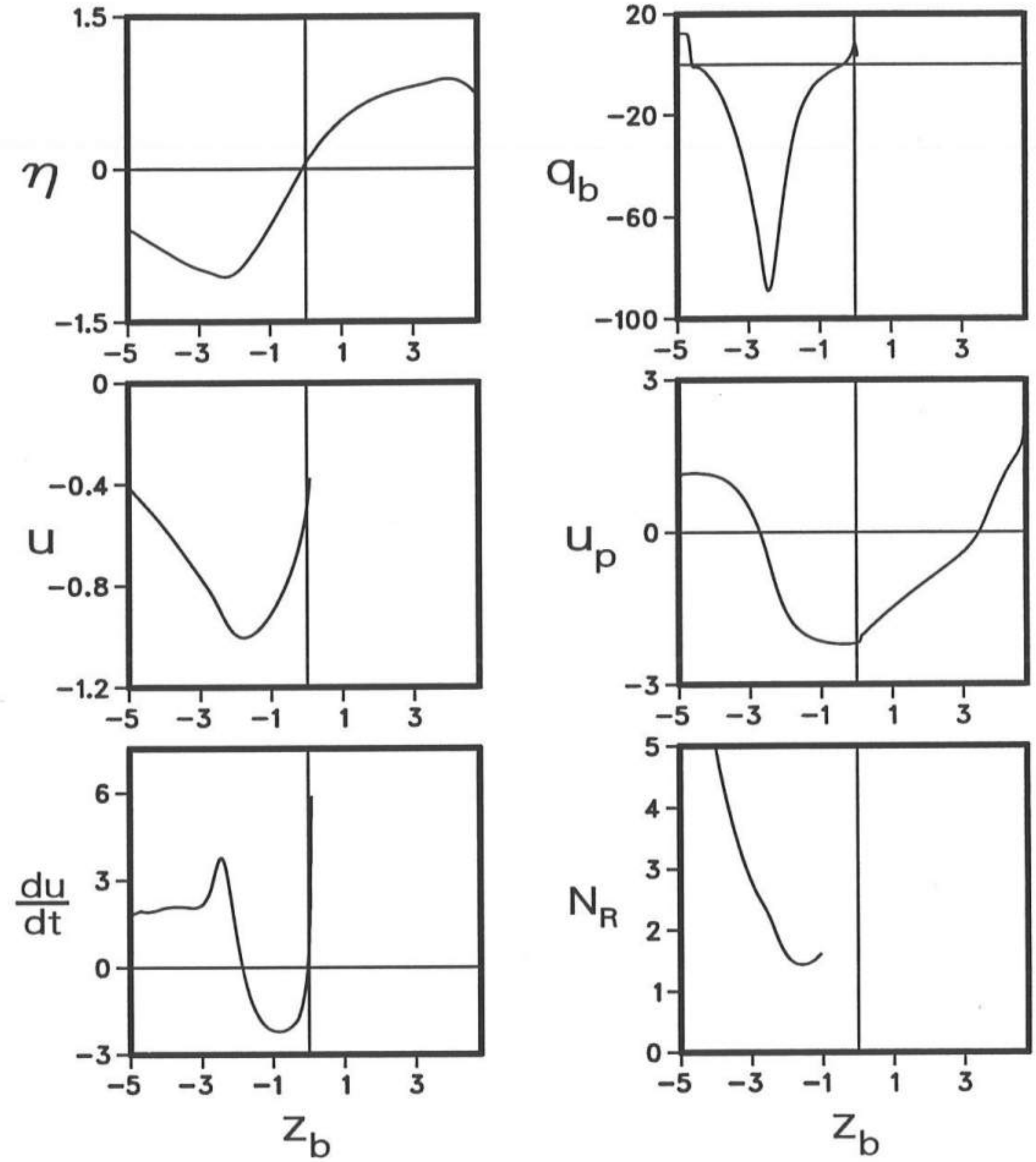


Figure D-21

APPENDIX E

CONTENTS OF THE ACCOMPANYING DISK

This report is accompanied by a 3.5 inch, high-density, IBM-PC-formatted floppy disk containing computer files as listed in Table E. The general rule in naming the computer files is as follows: [*.for] = FORTRAN program, [*.inp] = the primary input data file, [*.iwt] = the file containing the input wave train, and [*.doc] = the file ODOC.

Table E: Contents of the floppy disk accompanying this report.

DIRECTORY	NAME OF FILE ON DISK			REFER TO
	PROGRAM	INPUT FILES	FILE ODOC	
program	pbreak.for			Part III & Appendix A
appenb		appenb.inp appenb.iwt	appenb.doc	Appendix B
appenc		appenc.inp appenc.iwt	appenc.doc	Appendix C
append		append.inp append.iwt	append.doc	Appendix D

Appendix F

Computation Time

The following table presents approximate CPU time for each of the computed examples discussed in Appendices B, C, and D. All the computations made for this report were performed on Sun Microsystem SPARC2 machines.

Table F: Approximate CPU time for the computations discussed in this report.

Stage of Computation	Approximate CPU Time in Minutes on Sun Microsystem SPARC2 Machines		
	Run P1	Run P2	Run P3
Diagnostic (ICOMP=1,2) • Sequence 1 • Sequence 2 • Sequence 3	 40 75 -	 15 155 35	 205 - -
Actual (ICOMP=3)	145	270	285
Diagnostic+Actual in one sweep (ICOMP=4)	-	-	480