

UFL/COEL-92/019

DOCUMENTATION OF NEARSHORE INTERACTIVE
WAVE-CURRENT-TOPOGRAPHY CHANGE MODELS

by

Gang Miao

June 22, 1992

DOCUMENTATION OF NEARSHORE
INTERACTIVE
WAVE-CURRENT-TOPOGRAPHY
CHANGE MODELS

Gang Miao
Coastal and Oceanographic Engineering Department
336 Weil Hall, University of Florida
Gainesville, FL 32611, USA

June 22, 1992

Contents

1	OVERVIEW	1
2	OUTLINE OF THE MODEL	2
2.1	Wave Model	2
2.2	Circulation Model	3
2.3	Sediment Transport Model	4
3	APPLICATION OF THE 3-D BEACH EVOLUTION MODEL	8
3.1	MODEL PRO	9
3.1.1	Input Data	9
3.1.2	The Wave Model Subroutines	10
3.1.3	The Circulation Model Subroutines	12
3.1.4	The Sediment Transport Model Subroutines	14
3.1.5	Output Data	15
3.2	MODEL GROIN	16
3.2.1	Input Data	16
3.2.2	The Wave Model Subroutines	18

3.2.3	The Circulation Model Subroutines	23
3.2.4	The Sediment Transport Model Subroutines	25
3.2.5	Output Data	26
3.3	MODEL BARR	27
3.3.1	Input Data	28
3.3.2	The Wave Model Subroutines	29
3.3.3	The Circulation Model Subroutines	32
3.3.4	The Sediment Transport Model Subroutines	35
3.3.5	Output Data	36
3.4	MODEL SBINLET	37
3.4.1	Input Data	37
3.4.2	Output Data	39
4	CONCLUSIONS	40

Chapter 1

OVERVIEW

This report is to document the programs my colleague and I developed for computing time-dependent nearshore hydrographic changes including beach profile responses. The time scale of the model is suitable for storm events to seasonal changes, currently up to one year period. The model is very stable and is capable of handling complicated topographies including inlets and irregularly-shaped structures such as curved jetties and breakwaters.

The purpose of three-dimensional models is to predict the change of bottom topography from the spatial distribution of the sediment transport rates, which are evaluated from the nearshore wave and current fields computed point by point in small areas defined by a horizontal grid placed over the region of interest. Models of 3-D beach topography change require much fewer idealizations than do the line models.

Chapter 2

OUTLINE OF THE MODEL

The model consists of three submodels for calculating (1) waves, (2) nearshore currents, and (3) sediment transport and beach change. Nearshore waves, through radiation stresses, provide the driving force for the currents, and the resultant currents will conversely affect the wave field. The wave-current interaction may be treated by iteration, by alternate use of the two submodels for waves and for currents. Moreover, the change in bottom topography will produce changes in the nearshore waves and currents in the area. In order to incorporate this flow-topography interaction, additional iteration of the computations of waves, currents and topography is needed with a rather short time interval. At the first step, the initial beach topography and the geometry of the structures for the study area are given as input data. Next, the wave model determines the spatial distributions of radiation stresses and near-bottom orbital velocities. Then the circulation model computes the mean water surface level and the depth averaged mean currents using the radiation stresses from the wave model as the forcing terms and includes bottom friction, advective acceleration and the lateral diffusion terms. Finally, the sediment transport rates are computed at the local points from the wave-current conditions calculated in advance, and then the three dimensional bottom topography change is computed by solving the equation of sediment mass conservation. The wave and current fields are updated hourly to incorporate topography changes.

2.1 Wave Model

Kirby ([3], 1984) derived a mild-slope wave equation for a wave-current coexisting field, which is applicable to the computation of wave deformation due to combined effects of wave shoaling, refraction, diffraction, and breaking. The governing

equation is written as

$$\frac{D^2\tilde{\phi}}{Dt^2} + (\nabla_h \cdot \vec{U}) \frac{D\tilde{\phi}}{Dt} - \nabla_h(CC_g\nabla_h\tilde{\phi}) + (\sigma^2 - k^2CC_g)\tilde{\phi} = -\kappa \frac{D\tilde{\phi}}{Dt} \quad (2.1)$$

where t is the time, ∇_h is the horizontal gradient operator, \vec{U} is the depth averaged horizontal velocity vector, C is the phase velocity, C_g is the group velocity, σ is the angular frequency, and $\tilde{\phi}$ is the wave part of the velocity potential at the mean water level. The last term in Eq. (2.1) is the energy dissipation term, where κ is the energy dissipation coefficient. This term has been added in order to deal with the wave decay and recovery after breaking. Eventually coefficient κ will be related to the energy dissipation due to wave breaking following the work of Dally, Dean and Dalrymple ([1], 1984). The proper form of the dispersion relationship is:

$$\begin{aligned} \omega &= \sigma + \vec{k} \cdot \vec{U} \\ \sigma^2 &= gk \tanh kh \end{aligned}$$

Two stable and efficient computer algorithms were developed. One was based upon parabolic approximation of the Eq. (2.1) using a Crank-Nicholson finite difference scheme (Winer [9], 1988) with the equation given as:

$$\begin{aligned} (C_g \cos \theta + U)A_x + \frac{\sigma}{2} \left[\frac{C_g \cos \theta + U}{\sigma} \right]_x A + VA_y + \frac{\sigma}{2} \left(\frac{V}{\sigma} \right)_y A \\ - \frac{i}{2} k C_g (1 - \cos^2 \theta) A - \frac{i}{2} \left[CC_g \left(\frac{A}{\sigma} \right)_y \right]_y + \frac{\kappa}{2} A = 0 \end{aligned}$$

Here, the unknown is the complex amplitude A . The other was based upon elliptic wave equation using the Gragg's method (Lee and Wang, 1992):

$$\begin{aligned} -i\omega[2U \cdot \nabla\tilde{\phi} + (\nabla \cdot U)\tilde{\phi}] + (U \cdot \nabla)(U \cdot \nabla\tilde{\phi}) + (\nabla \cdot U)(U \cdot \nabla\tilde{\phi}) \\ - \nabla \cdot (CC_g\nabla\tilde{\phi}) + (\sigma^2 - \omega^2 - k^2CC_g - i\sigma\kappa)\tilde{\phi} = 0 \end{aligned}$$

The first one is suitable for general application whereas the second one is suitable for large wave angle and strong reflective boundaries.

2.2 Circulation Model

The governing equations for the circulation model are the depth integrated time averaged horizontal equations of momentum

$$\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} + V \frac{\partial U}{\partial y} + g \frac{\partial \bar{\eta}}{\partial x} + \frac{1}{\rho D} \tau_{bx} - \frac{1}{\rho D} \tau_{sx} + \frac{1}{\rho D} \left(\frac{\partial S_{xx}}{\partial x} + \frac{\partial S_{xy}}{\partial y} \right) + \frac{1}{\rho} \frac{\partial \tau_l}{\partial y} = 0$$

$$\frac{\partial V}{\partial t} + U \frac{\partial V}{\partial x} + V \frac{\partial V}{\partial y} + g \frac{\partial \bar{\eta}}{\partial y} + \frac{1}{\rho D} \tau_{by} - \frac{1}{\rho D} \tau_{sy} + \frac{1}{\rho D} \left(\frac{\partial S_{xy}}{\partial x} + \frac{\partial S_{yy}}{\partial y} \right) + \frac{1}{\rho} \frac{\partial \tau_l}{\partial x} = 0$$

and the continuity equation

$$\frac{\partial \bar{\eta}}{\partial t} + \frac{\partial}{\partial x}(UD) + \frac{\partial}{\partial y}(VD) = 0$$

where t is the time, (x, y) are the cartesian coordinates in a horizontal plane and (U, V) are the corresponding velocity components of the nearshore current, $D = h + \bar{\eta}$, h is the still water depth, and $\bar{\eta}$ is the elevation of the mean water level due to wave setup/setdown, τ_l is the lateral shear stress due to turbulent mixing, (τ_{bx}, τ_{by}) are the bottom shear stresses, (τ_{sx}, τ_{sy}) are the surface shear stresses, S_{xx}, S_{xy} and S_{yy} are the radiation stress components which arise from the excess momentum flux due to waves. The radiation stress terms are forcing terms, whereas the bottom friction terms and the lateral mixing terms represent flow impedances. These equations are obtained by integrating the local x and y momentum equations and the continuity equation over the depth of the water column and then time averaging the results (Ebersole and Dalrymple [2], 1979). The governing equations in the circulation model are solved by a matrix analysis using the alternating direction implicit (ADI) scheme. In order to treat the wave-current interaction, alternate computations of waves and of currents are necessary.

2.3 Sediment Transport Model

The sediment transport model is of energetic type following that developed by Watanabe et. al. ([8], 1986). The rate of sediment transport is treated as the summation of two energetic mechanisms one due to the mean current and the other due to the wave induced turbulence. In this model, the spatial distribution of sediment transport rates is estimated from local wave and current conditions computed with the wave and the nearshore current submodels. The change in local bottom elevation is calculated by solving the conservation equation for sediment mass, which allows us to obtain the time evolution of beach topography.

The change in local bottom elevation, z_b , or water depth, h , can readily be computed from the spatial distribution of the sediment transport rates by solving the following equation for the conservation of sediment volume

$$\frac{\partial h}{\partial t} = \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} \quad (2.2)$$

where t is the time, x and y are the horizontal coordinates, and q_x and q_y are the components of the sediment transport rate per unit width in the x and y directions.

$$\vec{q} = \vec{q}_w + \vec{q}_c = (q_x, q_y)$$

Since sediment transport is strongly nonlinear, the effects of coexistent waves and current are very complicated. It may seem unreasonable to treat transport rates under combined wave and current action as a linear summation of individual transport rates due separately to waves and to currents. However, at the present stage of knowledge, formulation of a complicated sediment transport rate prediction methodology for a wave-current coexistent system is not likely to succeed. It will be more practical to develop separate transport rate expressions for currents and for waves that are compatible with each other and consistent with past studies.

In order to easily utilize past work and to simplify the resulting formulas, the sediment transport rate has been divided into the transport due to mean currents and that due to the direct action of waves.

The sediment transport by a steady mean current under combined wave and current action is

$$\vec{q}_c = \frac{A_c(\tau_m - \tau_{cr})\vec{U}}{\rho g} \quad (2.3)$$

where A_c is a nondimensional coefficient (of the order 0.1 to 1.0), values of which should be empirically determined. τ_m is the maximum value of the bottom shear stress in a wave-current coexistent field. τ_{cr} is the critical shear stress for the onset of sediment movement, ρ is the density of water, and g is the gravitational acceleration. If $\tau_m \leq \tau_{cr}$, \vec{q}_c is zero. Eq. (2.3) is based on the power model concept, i.e., that the volume of sediments set in motion per unit area is proportional to the combined shear stress of waves and currents if the critical value is exceeded, and this volume is transported with the mean flow velocity (U, V). An important example is the longshore drift rate.

The transport due to waves is

$$\Phi = B_w(\Psi_m - \Psi_c)\Psi_m^{1/2} \quad (2.4)$$

where $\Phi = (1 - \lambda_v)q_w/w_0d$ is a dimensionless net transport rate, d , w_0 and λ_v are the diameter, settling velocity, and void ratio of the sediment, respectively, B_w is a nondimensional coefficient (of order 2 to 10), Ψ_m is the Shields parameter and Ψ_c is the critical value of Ψ_m for the onset of general movement of sediment. Since q_w is the absolute value of the net transport rate, a method is needed to determine the direction of the net transport. Criteria for predicting whether a beach will erode or accrete through cross-shore sand transport processes have been suggested by a number of authors. The two most commonly used factors include one parameter for characterizing the incident wave condition and another parameter involving some property of the sediment (grain size or fall speed) and/or beach slope. In development of SBEACH (Larson, Kraus and Byrnes, [4], 1989), the deepwater wave steepness H_0/L_0 and the parameter H_0/w_0T , called the dimensionless fall speed, were found to give a good distinction between profiles exhibiting mainly bar

and berm formation. The criterion for distinguishing beach erosion and accretion can be classified as follows,

$$\frac{H_0}{L_0} \leq C \left(\frac{H_0}{w_0 T} \right)^3, \quad \text{erosion}$$

$$\frac{H_0}{L_0} > C \left(\frac{H_0}{w_0 T} \right)^3, \quad \text{accretion}$$

in which $C = 0.00070$ is an empirical coefficient.

The maximum value of the bottom shear stress, τ_m , for a wave and current coexistent system will be evaluated by the friction law expressed in the following form:

$$\vec{\tau} = \rho C_f |\vec{u}_t| \vec{u}_t$$

where \vec{u}_t is comprised of both the mean current and the wave orbital velocities. The total velocity vector \vec{u}_t is expressed as

$$\vec{u}_t = (U + \hat{u} \cos \theta) \vec{i} + (V + \hat{u} \sin \theta) \vec{j}$$

so that its magnitude is given by

$$|\vec{u}_t| = \sqrt{U^2 + V^2 + \hat{u}^2 + 2U\hat{u} \cos \theta + 2V\hat{u} \sin \theta}$$

The wave orbital velocity \hat{u} is expressed as

$$\hat{u} = \hat{u}_m \cos \sigma t$$

where \hat{u}_m is the maximum wave orbital velocity at the bottom which is found to be

$$\hat{u}_m = \frac{\pi H}{T \sinh kh}$$

The absolute value of bottom shear stress can be expressed as

$$|\vec{\tau}| = \rho C_f |\vec{u}_t|^2$$

and

$$\tau_m = \max(|\vec{\tau}|)$$

The proposed transport rate formulas are based on the power model concept, in which it is assumed that the sediment transport rates depend on the bottom friction in excess of the critical shear stress, as well as on the flow velocity. Hence, it is necessary to determine the value of the critical shear stress, τ_{cr} , for the onset of sediment movement under the combined action of waves and currents. The critical condition for sediment movement under waves is given in terms of a critical value of the Shields parameter, Ψ_c

$$\Psi_c = \frac{\tau_{cr}}{(\rho_s - \rho)gd}$$

According to Watanabe (1980), the critical Shields parameter for the onset of general movement is approximately 0.11 for fine sand and 0.06 for coarse sand.

In the formulas for the sediment transport rates, the effect of bottom slope was not taken into account. Hence, even if a jagged bottom profile were to grow in the course of the calculation, such that the local slope exceeded the repose angle of the sediment, the unphysical slope could not be reduced. Although the wave-current field varies with the beach transformation, the change of the flow field alone cannot be expected to completely suppress the creation of a jagged profile. In reality, if the local slope becomes steep, sediment grains tend to move downward owing to the force of gravity. This effect may vary depending not only on the local slope but also on the degree of sediment transport calculated by neglecting the slope. In order to incorporate this effect, Eq. (2.5) is used instead of Eq. (2.2) to calculate the change of bottom elevation,

$$\frac{\partial h}{\partial t} = \frac{\partial}{\partial x} \left(q_x + \varepsilon_{sx} |q_x| \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \left(q_y + \varepsilon_{sy} |q_y| \frac{\partial h}{\partial y} \right) \quad (2.5)$$

where ε_{sx} , ε_{sy} are positive constants, the value of which will empirically be determined. Equation (2.5) is solved numerically with a finite difference method. A staggered mesh scheme is employed, in which the water depth, h , and the sediment transport rates, (q_x, q_y) , are defined at staggered points.

Chapter 3

APPLICATION OF THE 3-D BEACH EVOLUTION MODEL

In the preceding chapter, three submodels for simulating waves, currents, and beach change were described. These comprise a total numerical model for predicting three-dimensional beach evolution. When we apply this model for prediction, we must determine or specify various conditions for the computation, such as wave conditions, initial topography, time interval, and grid spacing, and values of certain coefficients associated with the wave breaking, current, and sediment transport calculations.

There are actually several separate programs in separate directories. They are named PRO.FOR, GROIN1.FOR, BARR.FOR, and JETTY.FOR. Each of these programs is written for a Cartesian grid. The PRO.FOR program has no internal barriers. GROIN1.FOR is an adaptation of PRO.FOR so as to include the internal boundary conditions that would exist at a shore perpendicular barrier. Similarly, BARR.FOR is an adaptation of PRO.FOR for the case of a shore parallel breakwater. In both cases, the barrier is represented as infinitesimally thin and impermeable. JETTY.FOR is specially modified to include the curved north jetty and straight south jetty of Sebastian Inlet of Florida as internal boundary conditions.

3.1 MODEL PRO

The model PRO is written for general use with open lateral boundaries and no internal barriers due to the presence of structures. The main program PRO.FOR and all the subroutines are in subdirectory [MIAO.PROFILE]. The main program PRO.FOR calls many subroutines which in turn call other subroutines. All of the subroutines used are listed in the file PRO.OPT which is also used for the linking command.

- The compiling command is `$@COMPILE <CR>`, or `$FOR PRO <CR>`, `$FOR INPUT <CR>`, etc.
- The linking command is `$LINK PRO/OPT <CR>`.
- The running command is `$RUN PRO <CR>`.

3.1.1 Input Data

Subroutine INPUT(M, N, WAVAMP, THETAO, TIME, ILAST, IDAY, IPRN, AC, AW) is called by the main program PRO.FOR. It is called only once at the beginning of the run. This subroutine prompts for the name of the input file, reads the file to set the following input parameters:

- M and N, the size of the computational domain.
- DX and DY, the grid spacing increment.
- DT, the time step increment.
- TIME, the start-up time constant.
- HO, the incident wave height at the offshore grid row.
- T, the wave period.
- THETAO, the wave angle at the offshore grid row.
- FRIC, the friction coefficient.
- EMAX, the maximum for the lateral mixing coefficient.
- ILAST, the maximum number of iterations.
- MIN/10, running duration in minutes divided by 10.

- IPRN, print out interval.
- AC and AW, sediment transport coefficient for current and wave.
- ICP, number of grids inside breaking point for plunging point position.

All of the above input parameters are separated by commas in the first two lines of the input file, so that the read statement in subroutine INPUT(M, N, WAVAMP, THETAO, TIME, ILAST, IDAY, IPRN, AC, AW) is in free format. A sample input file is PLANE.IN in the [MIAO.PROFILE] directory. The subroutine then prompts for the name of the depth file (the sample depth file is PLANE.DEP), opens it and reads the M by N array of the depth H(I,J). H(I,J) is measured positive downward from the still water line. Thus land above the still water line will be represented by a negative value. It is important that M be large enough to allow for flooding of the shoreline as wave set-up is produced.

3.1.2 The Wave Model Subroutines

The wave model part of the program is subroutine WAVRAD(M, N, THETAO, WAVHT) and those subroutines it calls. It is called from the main program PRO.FOR. It makes calls to the following subroutines: WAVNUM, AMPCO, CTRIDA, and RAD. The WAVNUM subroutine in turn makes calls to subroutine DISP, and the AMPCO subroutine makes calls to subroutine WDIS. The purpose of subroutine WAVRAD(M, N, THETAO, WAVHT) (and those it calls) is to determine the radiation stress terms. In do loop 10, the value of AMP(1,J) on the first grid row is determined by the expression $AMP(1,J) = WAVHT * AMPCO(J)$, where WAVHT is passed from the main program. The WAVRAD subroutine then calls subroutine WAVNUM to calculate the RK(I,J), RKO(I), CG(I,J), THETA(I), and SIG(I,J) arrays. Do loop 20 is the main part of the wave model. It is the marching solution of the parabolic wave equation grid row by grid row. The AMPCO subroutine determines the column vectors AA, BB, CC, and DD of the tridiagonal matrix equation. Subroutine CTRIDA then solves the tridiagonal matrix equation producing the solution vector VVV. And then the solution vector is assigned to the AMP(I,J) values on the next or I+1 grid row until values of the AMP(I,J) array are determined for every grid in the IMAX by N domain. The last line in subroutine WAVRAD before returning to the main program is a call to subroutine RAD to calculate the radiation stress arrays according to equation (2.57) on page 29 of Harley Winer's dissertation ([9], 1988).

Subroutine WAVNUM(M, N, THETAO) is a subroutine that calculates the wave number RK(I,J), the averaged wave number across a grid row RKO(I), the group velocity CG(I,J), and the relative frequency SIG(I,J). The Newton-Raphson

method is used to solve for $RK(I,J)$ using equation (3.1).

$$\omega = \sigma + k \cos \theta U \quad (3.1)$$

$RKO(I)$ is obtained by a simple averaging while $CG(I,J)$ and $SIG(I,J)$ are directly obtained from the appropriate formulas. The $WAVNUM$ subroutine also solves for the $THETA(I)$ array using Snell's law. To do this calls are made to $DISP$ subroutine to determine the wavenumber in the absence for any currents. The assumption is implicit in the model that at the lateral boundaries there are no significant on-offshore currents and that the topography has no y variations. Thus Snell's law for refraction applies at the lateral boundaries.

Subroutine $DISP(HH, RRKK)$ solves for the wavenumber in the absence of currents. HH is the depth of the water column and $RRKK$ is the wavenumber that is returned.

Subroutine $AMPCO(I, N)$ determines the coefficients of the tridiagonal matrix equation which results from the finite differencing of the parabolic wave equation. The governing equation used is equation (4.26) on page 52 of Harley Winer's dissertation ([9], 1988). It is finite differenced using a Crank-Nicholson scheme and is written as equation (4.27) on page 53 of the dissertation. The $AMPCO$ subroutine determines the column vectors $AA, BB, CC,$ and DD of the tridiagonal matrix equation. The values of these vectors are determined for each grid 1 through N . Special consideration is given for the J values of 1 and N so as to incorporate boundary conditions. Thus the subroutine has one set of computations for $J=1$, then a do loop for $J=2, N$ and then another computation for $J=N$. Subroutine $AMPCO$ is called sequentially for $I=1, IMAX-1$ from the $WAVRAD$ subroutine.

Subroutine $CTRIDA(IF, L)$ solves the tridiagonal matrix equation using a double sweep algorithm. Subroutine $CTRIDA(IF, L)$ is essentially the same as Subroutine $TRIDA(IF, L)$ except that it is for complex variables.

Subroutine $RAD(M, N)$ calculates the radiation stress components $SXX(I,J), SXY(I,J),$ and $SYI(I,J)$ from the gradients in the complex amplitude $AMP(I,J)$ using equation (2.57) on page 29 of Harley Winer's dissertation ([9], 1988). There is a do loop $J=2, N-1$ for the first $I=1$ grid row which uses a forward difference derivative in x because of the offshore boundary. Then there is the main do loop $I=2, IMAX-1$ with an internal do loop $J=2, N-1$ using central difference derivatives. Then there is a do loop for $J=2, N-1$ for each of the $IWET(J)$ grids using a three point derivative incorporating the zero wave height condition at the shore line. The final do loop of the RAD subroutine defines the radiation stress terms for the $J=1$ grid as being equal to those at the $J=2$ grid, and similarly the $J=N$ terms as being equal to those at the $J=N-1$ grid. This is an extension of the no gradients in the y direction boundary condition. This is done since the central difference y derivative can not be used at the $J=1$ and $J=N$ grids.

Subroutine WDIS(I, J, TTTDD, CCGG, AATEMP, WAVDIS) is called from the AMPCO subroutine. It computes the value of the energy dissipation coefficient in the governing wave equation using equation (3.97) on page 47 of Harley Winer's dissertation ([9], 1988). TTTDD is the total depth and CCGG is the group velocity. AATEMP is complex and is the value of the complex amplitude on the I row or the value of the complex amplitude calculated explicitly on the I+1 grid by equation (4.29) on page 54 of the dissertation. The values of KBREAK(J) and BRK(I,J) are determined by the WDIS subroutine. The values remain at zero until the wave height (twice the absolute value of the complex amplitude) becomes equal to or greater than 0.78 time the total water depth, and is then set to one. Once set to one, the values of KBREAK(J) and BRK(I,J) remain one until the wave height becomes less than 0.4 times the water depth. The rationale for this is found in section (3.3) starting on page 45 of the dissertation or the reader can go to the original work of Dally cited in the dissertation.

3.1.3 The Circulation Model Subroutines

The governing equations for the circulation model are the depth integrated time averaged horizontal equations of momentum and continuity by equation (2.1), (2.2), and (2.3) on page 16 of Harley Winer's dissertation ([9], 1988). When all of the assumptions concerning friction and mixing are included, the alternating direction implicit (ADI) scheme is used to finite difference and solve the governing equations.

Subroutine CIRC(M, N) is the circulation model. It is called directly from the main program PRO.FOR. It makes calls to the following subroutines: XCOJ1, XCOEF, XCOJN, YCOEF, YCVAR, and TRIDA. The basic subroutines in the circulation model are XCOEF and YCOEF which determine the coefficient vectors of the tridiagonal matrix. Subroutine CIRC sequentially calls the X-series subroutines. After each call to a X-series subroutine, a call is made to TRIDA to solve the tridiagonal matrix equation and then the solution vector VV is assigned to the appropriate variables. After the X-series of calls, the CIRC subroutine sequentially calls the Y-series of subroutines which make calls to TRIDA to solve the tridiagonal matrix equation and the solution vector VV is assigned to the appropriate variables.

Subroutine XCOEF(M, J) is called by subroutine CIRC(M, N) to determine the A, B, C, and D column vectors for each grid column in the x direction part of the alternating direction implicit method solution of the governing circulation equations. It is called sequentially for J=2, N-1. It determines all of the coefficients from I=1 to I=IWET(J). Special adaptations of XCOEF subroutine are made for any case where the J grid column is adjacent to either the domain boundary or

an internal boundary imposed by the presence of a barrier. The A, B, and C arrays are the coefficients of the tridiagonal matrix equation, this is the logic for the naming of the XCOEF subroutine.

Subroutine XCOJ1(M, J) is a special case of subroutine XCOEF(M, J) for J=1. This subroutine incorporates the lateral boundary conditions.

Subroutine XCOJN(M, J) is a special case of subroutine XCOEF(M, J) for J=N. Again, this subroutine incorporates the lateral boundary conditions.

Subroutine YCOEF(I, N) is called by subroutine CIRC(M, N) to determine the A, B, C, and D column vectors for each grid row in the y direction part of the alternating direction implicit method solution of the governing equations. This subroutine is called for values of I less than IMIN. It determines the coefficients for the I grid row from J=1 to J=N.

Subroutine YCVAR(I, N) is an adaption of subroutine YCOEF(I, N) for the case of I equal to or greater than IMIN. The adaptation is necessary so as to include the conditions that are imposed by adjacent dry grids. (Dry is used to refer to grids where the depth is such that it is above the water level. Similarly the term wet is used to refer to grids that have water.) This subroutine is also called from the CIRC subroutine. This subroutine is full of if statements to determine the relationship of the grid to the shoreline. If the neighboring grid in the y direction is dry then the V velocity between the two is set to zero. If the I+1 grid is dry then the edge of the grid is considered the shore and three point derivatives are taken using a zero value at the shoreline. (This is done for x derivatives of V and SXY.)

TRIDA(IF, L) is a subroutine that is called by subroutine CIRC(M, N) to solve the tridiagonal matrix equation using a double sweep method. IF and L are the first and last indices of the range of the column vectors A, B, C, and D which are passed in common statements. IF is usually 1 and L is usually 2*IWET(J) or 2*N+1.

EXCOEF(M, N) is the subroutine that determines the lateral mixing coefficients. It is called from the main program.

Subroutine UP(M, N) is called from the main program. It does the simple task of updating the values of U(I,J), V(I,J), and ETA(I,J) setting these values equal to those of UNEW(I,J), VNEW(I,J), and ETANEW(I,J). This subroutine also computes the ETADT terms for the first two grid rows.

Subroutine CHECK(KFLAG, M, N) is called from the main program prior to the call to subroutine UP. It checks for convergence by comparing the new values

(UNEW(I,J), VNEW(I,J), ETANEW(I,J)) with the old values (U(I,J), V(I,J), ETA(I,J)). Convergence is arbitrarily set to be when the percentage difference between the two is less than an arbitrarily small epsilon EPS for each grid. The KFLAG term is a flag that the main program PRO uses to decide whether to continue for another iteration or to exit the program.

PRINT(M, N) is a subroutine that prints the values of U(I,J), V(I,J), and ETA(I,J). This subroutine was used primarily in the development stage.

Subroutine FLOOD(M, N) determines whether the boundary of the computation domain at the shoreline should be increased or reduced due to changes in the water level. This is done with IF statements to determine whether a grid has gone dry or if the elevation in a neighboring grid is sufficient to wet the shoreward grid. The flooding procedure is done only in the x direction along grid columns. When it is determined that the water surface elevation in one grid is sufficient to flood the next grid an arbitrarily small amount of water is placed in the flooded grid. This is done so as to avoid a zero water depth in the computational domain (the wave routine will react to a zero water depth). If the depth in a grid becomes negative, then the velocity, wave height and wave set-up is set to zero, and the grid is removed from the computation domain.

Subroutine SHEAR(M, N) calculates the time averaged bottom shear stress array TBX(I,J) and TBY(I,J) using the formulation given by equations (2.15) and (2.16) on page 20 of Harley Winer's dissertation ([9], 1988) and modified by Miao.

3.1.4 The Sediment Transport Model Subroutines

The sediment transport part of the program is the SEDTR subroutine. Subroutine SEDTR(M, N, ITERA, AC, AW, WAVAMP, THETAO) is called from the main program. The purpose of this subroutine is to estimate the spatial distribution of sediment transport rates from local wave and current conditions computed with the wave and the near current submodels, and to calculate the change in local bottom elevation by solving the conservation equation for sediment mass.

Do loops 10 and 12 set the initial value of bottom orbital velocity amplitude, maximum bottom shear stress, and sediment transport rate to zero. Do loops 14 and 16 estimate the position of breaking point IB(J) and plunging point IP(J), and a decay coefficient LAMBDA(J). Do loop 18 is to calculate the maximum bottom shear stress TAOMU(I,J) and TAOMV(I,J) at the J=1 boundary. Do loops 20 and 30 are to calculate the maximum bottom shear stress TAOMU(I,J) and TAOMV(I,J) in x and y direction respectively from column J=2 to N and row I=1 to IWET(J) using equations in section (2.3). Do loop 32 is to calculate

the maximum bottom shear stress $TAOMU(I,J)$ and $TAOMV(I,J)$ at the $J=N+1$ boundary. Similarly, do loops 34, 36, 38 and 40 are written to calculate the sediment transport rates $QX(I,J)$ and $QY(I,J)$ in the x and y direction respectively using equations in section (2.3). Do loops 42, 44 and 45 perform a smoothing process to take into account the effect of bottom slope using Equation (2.5) in section (2.3). Finally, do loops 55 and 60 compute the change in water depth $H(I,J)$ by solving equation (2.2) for the conservation of sediment volume.

3.1.5 Output Data

Subroutine $WRITE(M, N, ITER)$ creates a file $OUT.LST$ and writes out the $ETA(I,J)$, $U(I,J)$ and $V(I,J)$ arrays. These values are written in free format. First $ETA(I,J)$ is written for $J=1, N$ for each grid row from $I=1$ through $I=M$. Then $U(I,J)$ is written the same way and then $V(I,J)$ is written for $J=1, N+1$ for each grid row. The values of the $IWET(J)$ array are also written so that the shoreline position can be found. Subroutine $WRITE(M, N, ITER)$ also creates files $OUTPC.LST$, $BRK.LST$, $VT.LST$, $HNEW.LST$, and $DH.LST$. The grid index (I, J) , the water depth array $H(I,J)$, and the water depth difference $HO(I,J)-H(I,J)$ are written in file $OUTPC.LST$. The wave breaking index $BRK(I,J)$ is written in file $BRK.LST$ with 1 indicating breaking waves and 0 indicating non-breaking waves. The total velocity array $VT(I,J)$ is written in file $VT.LST$. The time varying water depth array $H(I,J)$ is written in file $HNEW.LST$ in time interval $IPRN$ for $J=1, N$ for each grid row from $I=1$ through $I=M$. Similarly, the depth difference $HO(I,J)-H(I,J)$ is written in file $DH.LST$.

Subroutine $AMPOUT(M, N)$ writes out the wave height array $WAVEHGT(I,J)$ to a data file named $HGT.LST$ and writes out the value of the instantaneous water surface elevation due to the wave field to another data file named $ETA.LST$.

3.2 MODEL GROIN

The model GROIN is written for general use with open lateral boundaries and an internal barrier due to the presence of a shore perpendicular breakwater. The main program GROIN1.FOR and all the subroutines are in subdirectory [MIAO.PLGR]. This directory contains a modification of the program in the [MIAO.PROFILE] directory. The modification is for the case of a shore perpendicular breakwater that is represented as an infinitesimally thin barrier located between the $J=JGR$ and $J=JGR+1$ grid columns. Most of the subroutines are the same as in the [MIAO.PROFILE] directory or with minor modifications. The main program GROIN1.FOR calls many subroutines which in turn call other subroutines. The executable file is also GROIN1. All of the subroutines used are listed in the file GROIN1.OPT which is also used for the linking command.

- The compiling command is `$@COMPILE <CR>`, or `$FOR GROIN1 <CR>`, `$FOR INPUT <CR>`, etc.
- The linking command is `$LINK GROIN1/OPT <CR>`.
- The running command is `$RUN GROIN1 <CR>`.

Within the [MIAO.PLGR] directory there is a sample run available. The input file PLANE.IN and the depth file PLANE.DEP represent a test case of normal incident waves with a groin 50 meters in length located at $JGR=30$.

3.2.1 Input Data

Subroutine INPUT(M, N, WAVAMP, THETAO, TIME, ILAST, JGR, LLL, I HOUR, IPRN, AC, AW) is called by the main program GROIN1.FOR. The INPUT subroutine is modified to read the position and length of the groin and then pass this information to the main program through the argument of the call. Thus the argument of the INPUT subroutine has two additional variables JGR and LLL. JGR is the J position of the breakwater. It is located between the $J=JGR$ and the $J=JGR+1$ grid columns. JGR is read directly from the input data file. LLL is the grid row position of the end of the groin. The input data file has two more variables in this subdirectory than in the [MIAO.PROFILE] directory. The first is JGR and the second is the length of the groin GRLEN. The length of the groin should be in increments of DX. LLL is then determined after the IWET(J) array is determined by the definition:

$$LLL = IWET(JGR) - INT(GRLEN/DX) + 1$$

Subroutine INPUT is called only once at the beginning of the run. This subroutine prompts for the name of the input file, reads the file to set the following input parameters:

- M and N, the size of the computational domain.
- DX and DY, the grid spacing increment.
- DT, the time step increment.
- TIME, the start-up time constant.
- HO, the incident wave height at the offshore grid row.
- T, the wave period.
- THETAO, the wave angle at the offshore grid row.
- FRIC, the friction coefficient.
- EMAX, the maximum for the lateral mixing coefficient.
- ILAST, the maximum number of iterations.
- JGR, the J column position of the groin.
- GRLEN, the length of the groin.
- IHOURL, running duration in hours.
- IPRN, print out interval.
- AC and AW, sediment transport coefficient for current and wave.
- ICP, number of grids inside breaking point for plunging point position.

All of the above input parameters are separated by commas in the first two lines of the input file, so that the read statement in subroutine INPUT(M, N, WAVAMP, THETAO, TIME, ILAST, JGR, LLL, IHOURL, IPRN, AC, AW) is in free format. A sample input file is PLANE.IN in the [MIAO.PLGR] directory. The subroutine then prompts for the name of the depth file, opens it and reads the M by N array of the depth H(I,J). H(I,J) is measured positive downward from the still water line. Thus land above the still water line will be represented by a negative value. It is important that M be large enough to allow for flooding of the shoreline as wave set-up is produced.

3.2.2 The Wave Model Subroutines

The wave model part of the program is subroutine WAVRAD(M, N, THETAO, WAVHT, LLL, JGR) and those subroutines it calls. The WAVRAD subroutine has the values of LLL and JGR passed in the argument of the call statement. This subroutine is called from the main program GROIN1.FOR. It makes calls to the following subroutines: WAVNUM, AMPCO, AMPLL, AMPLLL, CTRIDA, and RAD. The WAVNUM subroutine in turn makes calls to subroutine DISP, and the AMPCO, AMPLL, AMPLLL subroutines make calls to subroutine WDIS. The purpose of subroutine WAVRAD(M, N, THETAO, WAVHT, LLL, JGR) (and those it calls) is to determine the radiation stress terms. In do loop 10, the value of AMP(1,J) on the first grid row is determined by the expression $AMP(1,J)=WAVHT*AMPCO(J)$, where WAVHT is passed from the main program. The WAVRAD subroutine then calls subroutine WAVNUM to calculate the RK(I,J), RKO(I), CG(I,J), THETA(I), and SIG(I,J) arrays. Do loops 20 to 50 is the main part of the wave model. It is the marching solution of the parabolic wave equation grid row by grid row. The AMPCO subroutine determines the column vectors AA, BB, CC, and DD of the tridiagonal matrix equation. Subroutine CTRIDA then solves the tridiagonal matrix equation producing the solution vector VVV. And then the solution vector is assigned to the AMP(I,J) values on the next or I+1 grid row until values of the AMP(I,J) array are determined for every grid in the IMAX by N domain. The first do loop to call Subroutine AMPCO in the WAVRAD Subroutine is for I=1, LLL-2. This moves the solution for the AMP(I,J) array up to the LLL-1 grid row which is the grid row just offshore of the end of the groin. Two adaptations of the AMPCO subroutine are then used to solve for the rest of the domain. Subroutine AMPLL is specifically written for the I=LLL-1 grid row. It determines the tridiagonal matrix coefficients for the I=LLL-1 grid row which does not have an internal boundary condition, and the I=LLL grid row which does have an internal boundary condition. The other adaptation of the AMPCO subroutine is subroutine AMPLLL which is called sequentially for I=LLL, IMAX-1 to complete the solution for the AMP(I,J) array. The last line in subroutine WAVRAD before returning to the main program is a call to subroutine RAD to calculate the radiation stress arrays according to equation (2.57) on page 29 of Harley Winer's dissertation([9], 1988).

WAVNUM(M, N, THETAO, LLL, JGR) is a subroutine that calculates the wave number RK(I,J), the averaged wave number across a grid row RKO(I), the group velocity CG(I,J), and the relative frequency SIG(I,J). The Newton-Raphson method is used to solve for RK(I,J) using equation:

$$\omega = \sigma + k \cos \theta U$$

RKO(I) is obtained by a simple averaging while CG(I,J) and SIG(I,J) are directly obtained from the appropriate formulas. The WAVNUM subroutine also solves

for the THETA(I) array using Snell's law. To do this calls are made to DISP subroutine to determine the wavenumber in the absence for any currents. The assumption is implicit in the model that at the lateral boundaries there are no significant on-offshore currents and that the topography has no y variations. Thus Snell's law for refraction applies at the lateral boundaries.

Subroutine DISP(HH, RKN) solves for the wavenumber in the absence of currents. HH is the depth of the water column and RRKK is the wavenumber that is returned.

Subroutine AMPCO(I, N) determines the coefficients of the tridiagonal matrix equation which results from the finite differencing of the parabolic wave equation. The governing equation used is equation (4.26) on page 52 of Harley Winer's dissertation ([9], 1988). It is finite differenced using a Crank-Nicholson scheme and is written as equation (4.27) on page 53 of the dissertation. The AMPCO subroutine determines the column vectors AA, BB, CC, and DD of the tridiagonal matrix equation. The values of these vectors are determined for each grid 1 through N. Special consideration is given for the J values of 1 and N so as to incorporate boundary conditions. Thus the subroutine has one set of computations for J=1, then a do loop for J=2, N and then another computation for J=N. Subroutine AMPCO is called sequentially for I=1, IMAX-1 from the WAVRAD subroutine.

Subroutine AMPLL(I, N, JGR) is an adaptation of the AMPCO subroutine. It is specifically written for the I=LLL-1 grid row. It determines the tridiagonal matrix coefficients for the I=LLL-1 grid row which does not have an internal boundary condition, and the I=LLL grid row which does have an internal boundary condition. The internal boundary condition is handled in the following manner. When the parabolic equation (equation 4.27 on page 53 of Harley Winer's dissertation [9], 1988) is written for J=JGR, there are terms for the J=JGR+1 grid on the other side of the barrier. The barrier is treated as perfectly reflective. Thus any quantity on the other side of the barrier is eliminated from the equation in terms of the quantity on the proper side of the barrier. Similarly for the J=JGR+1 grid, values at the J=JGR grid are eliminated in terms of the value at the J=JGR+1 grid. Essentially the barrier at the groin is treated the same as the reflective boundary condition as discussed in section 4.4 of Harley Winer's dissertation ([9], 1988).

Subroutine AMPLLL(I, N, JGR) is the other adaptation of the AMPCO subroutine. Similar to subroutine AMPLL(I, N, JGR), the internal boundary condition at the groin is treated as perfectly reflective.

Thus any quantity on the other side of the barrier is eliminated from the equation in terms of the quantity on the proper side of the barrier. Subroutine AMPLLL(I, N, JGR) is called sequentially for I=LLL, IMAX-1 to complete the solution for the AMP(I,J) array.

Subroutine CTRIDA(IF, L) solves the tridiagonal matrix equation using a double sweep algorithm. Subroutine CTRIDA(IF, L) is essentially the same as Subroutine TRIDA(IF, L) except that it is written for complex variables.

In the RAD subroutine the values of LLL and JGR are passed in the argument of the call. Subroutine RAD(M, N, LLL, JGR) calculates the radiation stress components SXX(I,J), SXY(I,J), and SYI(I,J) from the gradients in the complex amplitude AMP(I,J) using equation (2.57) on page 29 of Harley Winer's dissertation ([9], 1988). There is a do loop J=2, N-1 for the first I=1 grid row which uses a forward difference derivative in x because of the offshore boundary. Then there is the main do loop I=2, IMAX-1 with an internal do loop J=2, N-1 using central difference derivatives. Then there is a do loop for J=2, N-1 for each of the IWET(J) grids using a three point derivative incorporating the zero wave height condition at the shore line. The final do loop of the RAD subroutine defines the radiation stress terms for the J=1 grid as being equal to those at the J=2 grid, and similarly the J=N terms as being equal to those at the J=N-1 grid. This is an extension of the no gradients in the y direction boundary condition. This is done since the central difference y derivative can not be used at the J=1 and J=N grids. In the RAD subroutine a special do loop is written for the two grid columns on each side of the groin to insure that the y derivative are not taken across the barrier. Thus for I=LLL, IWET(JGR) on the J=JGR grid column the y derivatives of AMP(I,J) (and similarly for the complex conjugate) are written as

$$(AMP(I, J) - AMP(I, J - 1)) / (2 * DY)$$

and for the J=JGR+1 grid column they are written as

$$(AMP(I, J + 1) - AMP(I, J)) / (2 * DY)$$

These two expressions result from using the reflective condition to eliminate the value of AMP(I,J) on the other side of the barrier in terms of the value of AMP(I,J) on the proper side of the barrier.

Subroutine WDIS(I, J, TTTDD, CCGG, AATEMP, WAVDIS) is called from the AMPCO subroutine. It computes the value of the energy dissipation coefficient in the governing wave equation using equation (3.97) on page 47 of Harley Winer's dissertation ([9], 1988). TTTDD is the total depth and CCGG is the group velocity. AATEMP is complex and is the value of the complex amplitude on the I row or the value of the complex amplitude calculated explicitly on the I+1 grid by equation (4.29) on page 54 of the dissertation. The values of KBREAK(J) and BRK(I,J) are determined by the WDIS subroutine. The values remain at zero until the wave height (twice the absolute value of the complex amplitude) becomes equal to or greater than 0.78 time the total water depth, and is then set to one. Once set to one, the values of KBREAK(J) and BRK(I,J) remain one until the wave height becomes less than 0.4 times the water depth. The rationale for this is found in

section (3.3) starting on page 45 of the dissertation or the reader can go to the original work of Dally cited in the dissertation.

3.2.3 The Circulation Model Subroutines

The governing equations for the circulation model are the depth integrated time averaged horizontal equations of momentum and continuity by equation (2.1), (2.2), and (2.3) on page 16 of Harley Winer's dissertation ([9], 1988). When all of the assumptions concerning friction and mixing are included, the alternating direction implicit (ADI) scheme is used to finite difference and solve the governing equations.

Subroutine CIRC(M, N, LLL, JGR) is the circulation model. It is called directly from the main program GROIN1.FOR. It makes calls to the following subroutines: XCOJ1, XCOEF, XCOJN, XCJGR, XCJGR1, YCOEF, YCVAR, and TRIDA. The CIRC subroutine is altered in that LLL and JGR are passed in the argument of the call, and that the two subroutines XCJGR and XCJGR1 are added to determine the coefficients for the x sweep for the special cases of $J=JGR$ and $J=JGR+1$ respectively. The basic subroutines in the circulation model are XCOEF and YCOEF which determine the coefficient vectors of the tridiagonal matrix. Subroutine CIRC sequentially calls the X-series subroutines. After each call to a X-series subroutine, a call is made to TRIDA to solve the tridiagonal matrix equation and then the solution vector VV is assigned to the appropriate variables. After the X-series of calls, the CIRC subroutine sequentially calls the Y-series of subroutines which make calls to TRIDA to solve the tridiagonal matrix equation and the solution vector VV is assigned to the appropriate variables.

Subroutine XCOEF(M, J) is called by subroutine CIRC(M, N, LLL, JGR) to determine the A, B, C, and D column vectors for each grid column in the x direction part of the alternating direction implicit method solution of the governing circulation equations. It is called sequentially for $J=2, N-1$. It determines all of the coefficients from $I=1$ to $I=IWET(J)$. Special adaptations of XCOEF subroutine are made for any case where the J grid column is adjacent to either the domain boundary or an internal boundary imposed by the presence of a barrier. The A, B, and C arrays are the coefficients of the tridiagonal matrix equation, this is the logic for the naming of the XCOEF subroutine.

Subroutine XCOJ1(M, J) is a special case of subroutine XCOEF(M, J) for $J=1$. This subroutine incorporates the lateral boundary conditions.

Subroutine XCOJN(M, J) is a special case of subroutine XCOEF(M, J) for $J=N$. Again, this subroutine incorporates the lateral boundary conditions.

Subroutine XCJGR(M, J, LLL) is a special case of subroutine XCOEF(M, J) for $J=JGR$. This subroutine determines the coefficients for the x sweep for $J=JGR$ so as to avoid taking derivatives across the impermeable barrier.

Subroutine XCJGR1(M, J, LLL) is a special case of subroutine XCOEF(M, J) for $J=JGR+1$. This subroutine determines the coefficients for the x sweep for $J=JGR+1$ so as to avoid taking derivatives across the impermeable barrier.

Subroutine YCOEF(I, N, LLL, JGR) is called by subroutine CIRC(M, N, LLL, JGR) to determine the A, B, C, and D column vectors for each grid row in the y direction part of the alternating direction implicit method solution of the governing equations. This subroutine is called for values of I less than IMIN. It determines the coefficients for the I grid row from $J=1$ to $J=N$. Subroutine YCOEF(I, J, LLL, JGR) is also altered in that LLL and JGR are passed in the argument. If I is equal to or greater than LLL the internal boundary condition $V(I, JGR+1)=0$ is imposed by setting the appropriate coefficients to zero in the same manner as was done for the XCOEF subroutine.

Subroutine YCVAR(I, N, LLL, JGR) is an adaption of subroutine YCOEF(I, N, LLL, JGR) for the case of I equal to or greater than IMIN. The adaptation is necessary so as to include the conditions that are imposed by adjacent dry grids. (Dry is used to refer to grids where the depth is such that it is above the water level. Similarly the term wet is used to refer to grids that have water.) This subroutine is also called from the CIRC subroutine. This subroutine is full of if statements to determine the relationship of the grid to the shoreline. If the neighboring grid in the y direction is dry then the V velocity between the two is set to zero. If the I+1 grid is dry then the edge of the grid is considered the shore and three point derivatives are taken using a zero value at the shoreline. (This is done for x derivatives of V and SXY). Subroutine YCVAR(I, J, LLL, JGR) is also altered in that LLL and JGR are passed in the argument. If I is equal to or greater than LLL the internal boundary condition $V(I, JGR+1)=0$ is imposed by setting the appropriate coefficients to zero in the same manner as was done for the XCOEF subroutine.

TRIDA(IF, L) is a subroutine that is called by subroutine CIRC(M, N, LLL, JGR) to solve the tridiagonal matrix equation using a double sweep method. IF and L are the first and last indices of the range of the column vectors A, B, C, and D which are passed in common statements. IF is usually 1 and L is usually $2*IWET(J)$ or $2*N+1$.

EXCOEF(M, N) is the subroutine that determines the lateral mixing coefficients. It is called from the main program.

Subroutine UP(M, N) is called from the main program. It does the simple

task of updating the values of $U(I,J)$, $V(I,J)$, and $ETA(I,J)$ setting these values equal to those of $UNEW(I,J)$, $VNEW(I,J)$, and $ETANEW(I,J)$. This subroutine also computes the ETADT terms for the first two grid rows.

Subroutine CHECK(KFLAG, M, N) is called from the main program prior to the call to subroutine UP. It checks for convergence by comparing the new values ($UNEW(I,J)$, $VNEW(I,J)$, $ETANEW(I,J)$) with the old values ($U(I,J)$, $V(I,J)$, $ETA(I,J)$). Convergence is arbitrarily set to be when the percentage difference between the two is less than an arbitrarily small epsilon EPS for each grid. The KFLAG term is a flag that the main program PRO uses to decide whether to continue for another iteration or to exit the program.

PRINT(M, N) is a subroutine that prints the values of $U(I,J)$, $V(I,J)$, and $ETA(I,J)$. This subroutine was used primarily in the development stage.

Subroutine FLOOD(M, N) determines whether the boundary of the computation domain at the shoreline should be increased or reduced due to changes in the water level. This is done with IF statements to determine whether a grid has gone dry or if the elevation in a neighboring grid is sufficient to wet the shoreward grid. The flooding procedure is done only in the x direction along grid columns. When it is determined that the water surface elevation in one grid is sufficient to flood the next grid an arbitrarily small amount of water is placed in the flooded grid. This is done so as to avoid a zero water depth in the computational domain (the wave routine will react to a zero water depth). If the depth in a grid becomes negative, then the velocity, wave height and wave set-up is set to zero, and the grid is removed from the computation domain.

Subroutine SHEAR(M, N) calculates the time averaged bottom shear stress array $TBX(I,J)$ and $TBY(I,J)$ using the formulation given by equations (2.15) and (2.16) on page 20 of Harley Winer's dissertation ([9], 1988) and modified by Miao.

3.2.4 The Sediment Transport Model Subroutines

The sediment transport part of the program is the SEDTR subroutine. Subroutine SEDTR(M, N, LLL, JGR, ITERA, AC, AW, WAVAMP, THETAO) is called from the main program. The purpose of this subroutine is to estimate the spatial distribution of sediment transport rates from local wave and current conditions computed with the wave and the near current submodels, and to calculate the change in local bottom elevation by solving the conservation equation for sediment mass.

Subroutine SEDTR(M, N, LLL, JGR, ITERA, AC, AW, WAVAMP, THETAO)

is also altered in that LLL and JGR are passed in the argument. Do loops 10 and 12 set the initial value of bottom orbital velocity amplitude, maximum bottom shear stress, and sediment transport rate to zero. Do loops 14 and 16 estimate the position of breaking point IB(J) and plunging point IP(J), and a decay coefficient LAMBDA(J). Do loop 18 is to calculate the maximum bottom shear stress TAOMU(I,J) and TAOMV(I,J) at the J=1 boundary. Do loop 20 and 30 are to calculate the maximum bottom shear stress TAOMU(I,J) and TAOMV(I,J) in x and y direction respectively from column J=2 to N and row I=1 to IWET(J) using equations in section (2.3). Do loop 32 is to calculate the maximum bottom shear stress TAOMU(I,J) and TAOMV(I,J) at the J=N+1 boundary. Similarly, do loops 34, 36, 38 and 40 are written to calculate the sediment transport rates QX(I,J) and QY(I,J) in the x and y direction respectively. Do loops 42, 44 and 45 perform a smoothing process to take into account the effect of bottom slope using Equation (2.5) in section (2.3). Do loop 46 is added to set the internal boundary condition for a groin. If I is equal to or greater than LLL the internal boundary condition $QYP(I, JGR+1)=0$ is imposed. Finally, do loops 52 and 62 compute the change in water depth H(I,J) by solving equation (2.2) for the conservation of sediment volume.

3.2.5 Output Data

Subroutine WRITE(M, N, ITER) creates a file OUT.LST and writes out the ETA(I,J), U(I,J) and V(I,J) arrays. These values are written in free format. First ETA(I,J) is written for J=1, N for each grid row from I=1 through I=M. Then U(I,J) is written the same way and then V(I,J) is written for J=1, N+1 for each grid row. The values of the IWET(J) array are also written so that the shoreline position can be found. Subroutine WRITE(M, N, ITER) also creates files HNEW.LST and SHEAR.LST. The time varying water depth array H(I,J) is written in file HNEW.LST in time interval IPRN for J=1, N for each grid row from I=1 through I=M. Similarly, the bottom shear stress is written in file SHEAR.LST.

Subroutine AMPOUT(M, N) writes out the wave height array WAVEHGT(I,J) to a data file named HGT.LST and writes out the value of the instantaneous water surface elevation due to the wave field to another data file named ETA.LST.

3.3 MODEL BARR

The model BARR is written for general use with open lateral boundaries and a shore parallel breakwater. The main program BARR.FOR and all the subroutines are in subdirectory [MIAO.BAR]. In this subdirectory the program is modified to include the internal boundary conditions found with a shore parallel breakwater that is represented as an infinitesimally thin impermeable barrier. The main program is BARR.FOR which calls many subroutines which in turn call other subroutines. The executable file is BARR.EXE. The option file is BARR.OPT and has a listing of all the subroutines used in the BARR program. New subroutines ACOH1, ACOH2, WVNMHA, YCOJJJ, and YCJJJ1 are listed in the BARR.OPT file. The DIM.FOR file that is included in all the subroutines remains unchanged except for the addition of several arrays that are passed through the following common statements.

```
COMMON/Q/RKHA(100), RKOHA, CGHA(100), SIGHA(100)
COMMON/R/REFLEC(100,100), AMPSAV(100), AMPH(100), REFL(100)
```

SIGMA(100), CGHA(100), and RKHA(100) are the relative frequency, the group velocity, the wave number arrays at the position half way between the I=JJJ and I=JJJ+1 grid centers, which is where the breakwater is located. RKOHA is the averaged wave number at the shore side of the I=JJJ grid row. These values are computed in the WVNMHA subroutine. REFLEC, AMPSAV, AMPH, and REFL are declared complex. AMPH and AMPSAV are the complex amplitude at the grid edge position of the breakwater. These are used in the WAVRAD and RAD subroutines.

- The compiling command is `$@COMPILE <CR>`, or `$FOR BARR <CR>`, `$FOR INPUT <CR>`, etc.
- The linking command is `$LINK BARR/OPT <CR>`.
- The running command is `$RUN BARR <CR>`.

Within the [MIAO.BAR] directory there is a sample run available. The input file PLANE.IN and the depth file PLANE.DEP represent a test case of normal incident waves with a breakwater 60 meters offshore (JJJ=14, DX=10m for a plane beach with 20 wet grids in the x direction) and 40 meters in length (NBB=2, DY=10m).

3.3.1 Input Data

Subroutine INPUT(M, N, WAVAMP, THETAO, TIME, ILAST, JJJ, NBB, IDAY, IPRN, AC, AW) is called by the main program BARR.FOR. It is called only once at the beginning of the run. The INPUT subroutine is modified so as to read and then pass the position of the breakwater. The input data file has two additional values that are read by the INPUT subroutine. They are JJJ and NBB which are passed in the argument of the INPUT subroutine back to the main program. JJJ is the position of the breakwater in the x direction. The breakwater is on the shore side of the I=JJJ grid row, between the I=JJJ and I=JJJ+1 grid rows. NBB is the number of grids for one half the length of the breakwater. Thus the breakwater length is equal to $2*NBB * DY$. The INPUT subroutine prompts for the name of the input file, reads the file to set the following input parameters:

- M and N, the size of the computational domain.
- DX and DY, the grid spacing increment.
- DT, the time step increment.
- TIME, the start-up time constant.
- HO, the incident wave height at the offshore grid row.
- T, the wave period.
- THETAO, the wave angle at the offshore grid row.
- FRIC, the friction coefficient.
- EMAX, the maximum for the lateral mixing coefficient.
- ILAST, the maximum number of iterations.
- JJJ, the number of grids from offshore boundary for the position of the breakwater in the x direction.
- NBB, the number of grids for one half the length of the breakwater.
- I HOUR, running duration in hours.
- IPRN, print out interval.
- AC and AW, sediment transport coefficient for current and wave.
- ICP, number of grids inside breaking point for plunging point position.

All of the above input parameters are separated by commas in the first two lines of the input file, so that the read statement in subroutine INPUT(M, N, WAVAMP, THETAO, TIME, ILAST, JJJ, NBB, IDAY, IPRN, AC, AW) is in free format. A sample input file is PLANE.IN in the [MIAO.PROFILE] directory. Once NBB is passed back to the main program it is used to determine the fixed end positions of the breakwater N1 and N2. Thus the breakwater is on the shore side of the I=JJJ grid row and is from the J=N1 through the J=N2 grids. JJJ, N1 and N2 are then passed in the arguments of the calls to various subroutines. Subroutine INPUT then prompts for the name of the depth file, opens it and reads the M by N array of the depth H(I,J). H(I,J) is measured positive downward from the still water line. Thus land above the still water line will be represented by a negative value. It is important that M be large enough to allow for flooding of the shoreline as wave set-up is produced.

3.3.2 The Wave Model Subroutines

The wave model part of the program is subroutine WAVRAD(M, N, WAVHT, JJJ, N1, N2, THETAO) and those subroutines it calls. It is called from the main program BARR.FOR. It makes calls to the following subroutines: WAVNUM, WVMNHA, AMPCO, ACOH1, ACOH2, CTRIDA, and RAD. The WAVNUM subroutine in turn makes calls to subroutine DISP, and the AMPCO subroutine makes calls to subroutine WDIS. The purpose of subroutine WAVRAD(M, N, WAVHT, JJJ, N1, N2, THETAO) (and those it calls) is to determine the radiation stress terms. In do loop 10, the value of AMP(1,J) on the first grid row is determined by the expression $AMP(1,J)=WAVHT*AMPCO(J)$, where WAVHT is passed from the main program. The WAVRAD subroutine then calls subroutine WAVNUM to calculate the RK(I,J), RKO(I), CG(I,J), THETA(I), and SIG(I,J) arrays. It then calls subroutine WVMNHA which determines the wave characteristics ($k, C_g, \sigma, \bar{k}, \bar{\theta}$) at the position of the edge of the grid row where the barrier is located. Do loops 20 to 50 are the main part of the wave model. It is the marching solution of the parabolic wave equation grid row by grid row. Do loop 20 is for I=1, JJJ-1 which calls the AMPCO subroutine to determine the column vectors AA, BB, CC, and DD of the tridiagonal matrix equation. Subroutine CTRIDA then solves the tridiagonal matrix equation producing the solution vector VVV. This gives the values of the AMP(I,J) array up to and including the I=JJJ grid row. Then a call is made to subroutine ACOH1 which has the same structure as subroutine AMPCO except that DX is set to one half of DX. Subroutine ACOH1 determines the coefficients of the tridiagonal matrix equation in terms of the known AMP(I,J) values and the wave characteristics (k, \bar{k}, σ, C_g) on the I=JJJ grid row and the wave characteristics on the grid edge (the RKHA(J), CGHA(J), SIGHA(J), and RKOHA values). Then a call is made to subroutine CTRIDA so that the solution vector is the complex amplitude at the grid edge half way between the I=JJJ and

I=J+1 grid centers. The solution vector VVV is then assigned to the AMPH(J) array. Then for J=N1, N2 the value of AMPH(J) is assigned to AMPSAV(J) and then AMPH(J) is set to zero. This is done so that AMPSAV(J) can be used in the RAD subroutine to get a three point x derivative on the offshore side of the breakwater and so that AMPH(J) can be used in the ACOH2 subroutine to advance the parabolic wave model to the I=J+1 grid center. Next the WAVRAD subroutine calls the ACOH2 subroutine. The ACOH2(I, N) subroutine has the same structure as the AMPCO subroutine. The difference is that DX is again one half DX and the coefficients are in terms of the known AMPH(J) values and the wave characteristics on the grid edge (the RKHA, CGHA, SIGHA, and RKOHA values) and the wave characteristics (k, \bar{k}, σ, C_g) on the I=J+1 grid row. A call is then made to subroutine CTRIDA and the solution vector is assigned to the AMP(I,J) array on the I=J+1 grid row. With the values of AMP(I,J) array known on the I=J+1 grid row the rest of the AMP(I,J) array is determined by a do loop for I=J+1, IMAX-1 which calls subroutine AMPCO, then subroutine CTRIDA, and then the solution vector is assigned to the AMP(I,J) values on the next or I+1 grid row until values of the AMP(I,J) array are determined for every grid in the IMAX by N domain. The final execution in the WAVRAD subroutine is a call to subroutine RAD which is the same as the RAD subroutine previously described except that there are the additional do loops for J=N1, N2 for both I=J and I=J+1 grid rows.

Subroutine WAVNUM(M, N, THETAO, J, N1, N2) is a subroutine that calculates the wave number RK(I,J), the averaged wave number across a grid row RKO(I), the group velocity CG(I,J), and the relative frequency SIG(I,J). The Newton-Raphson method is used to solve for RK(I,J) using equation

$$\omega = \sigma + k \cos \theta U$$

RKO(I) is obtained by a simple averaging while CG(I,J) and SIG(I,J) are directly obtained from the appropriate formulas. The WAVNUM subroutine also solves for the THETA(I) array using Snell's law. To do this calls are made to DISP subroutine to determine the wavenumber in the absence for any currents. The assumption is implicit in the model that at the lateral boundaries there are no significant on-offshore currents and that the topography has no y variations. Thus Snell's law for refraction applies at the lateral boundaries.

Subroutine WVMHA(I, N, THETAO) determines the wave characteristics ($k, C_g, \sigma, \bar{k}, \bar{\theta}$) at the position of the edge of the grid row where the barrier is located. The WVMHA subroutine does this by taking the depth at the grid edge to be the average of the two adjacent grids. The WVMHA subroutine has the same structure and logic as the WAVNUM subroutine.

Subroutine DISP(HH, RRKK) solves for the wavenumber in the absence of currents. HH is the depth of the water column and RRKK is the wavenumber that

is returned.

Subroutine AMPCO(I, N) determines the coefficients of the tridiagonal matrix equation which results from the finite differencing of the parabolic wave equation. The governing equation used is equation (4.26) on page 52 of Harley Winer's dissertation ([9], 1988). It is finite differenced using a Crank-Nicholson scheme and is written as equation (4.27) on page 53 of the dissertation. The AMPCO subroutine determines the column vectors AA, BB, CC, and DD of the tridiagonal matrix equation. The values of these vectors are determined for each grid 1 through N. Special consideration is given for the J values of 1 and N so as to incorporate boundary conditions. Thus the subroutine has one set of computations for J=1, then a do loop for J=2, N and then another computation for J=N. Subroutine AMPCO is called sequentially for I=1, IMAX-1 from the WAVRAD subroutine.

Subroutine ACOH1(I, N) has the same structure as subroutine AMPCO except that DX is set to one half of DX. Subroutine ACOH1 determines the coefficients of the tridiagonal matrix equation in terms of the known AMP(I,J) values and the wave characteristics $(k, \bar{k}, \sigma, C_g)$ on the I=JJJ grid row and the wave characteristics on the grid edge (the RKHA(J), CGHA(J), SIGHA(J), and RKOHA values).

The ACOH2(I, N) subroutine has the same structure as the AMPCO subroutine. The difference is that DX is again one half DX and the coefficients are in terms of the known AMPH(J) values and the wave characteristics on the grid edge (the RKHA, CGHA, SIGHA, and RKOHA values) and the wave characteristics $(k, \bar{k}, \sigma, C_g)$ on the I=JJJ+1 grid row.

Subroutine CTRIDA(IF, L) solves the tridiagonal matrix equation using a double sweep algorithm. Subroutine CTRIDA(IF, L) is essentially the same as Subroutine TRIDA(IF, L) except that it is for complex variables.

Subroutine RAD(M, N, JJJ, N1, N2) calculates the radiation stress components SXX(I,J), SXY(I,J), and SYI(I,J) from the gradients in the complex amplitude AMP(I,J) using equation (2.57) on page 29 of Harley Winer's dissertation ([9], 1988). There is a do loop J=2, N-1 for the first I=1 grid row which uses a forward difference derivative in x because of the offshore boundary. Then there is the main do loop I=2, IMAX-1 with an internal do loop J=2, N-1 using central difference derivatives. Then there is a do loop for J=2, N-1 for each of the IWET(J) grids using a three point derivative incorporating the zero wave height condition at the shore line. The do loop 30 of the RAD subroutine defines the radiation stress terms for the J=1 grid as being equal to those at the J=2 grid, and similarly the J=N terms as being equal to those at the J=N-1 grid. This is an extension of the no gradients in the y direction boundary condition. This is done since the central difference y derivative can not be used at the J=1 and J=N grids. There are additional do loops 320 and 420 for J=N1, N2 for both I=JJJ and I=JJJ+1

grid rows. These do loops have the same structure as other computations in the RAD subroutine except that DADX and DACONDX are written as three point derivatives. For the I=JJJ grid row the three point derivatives are written using the AMPSAV(J) array and on the I=JJJ+1 grid row the derivatives are written using a zero value at the lee side of the breakwater.

Subroutine WDIS(I, J, TTTDD, CCGG, AATEMP, WAVDIS) is called from the AMPCO subroutine. It computes the value of the energy dissipation coefficient in the governing wave equation using equation (3.97) on page 47 of Harley Winer's dissertation ([9], 1988). TTTDD is the total depth and CCGG is the group velocity. AATEMP is complex and is the value of the complex amplitude on the I row or the value of the complex amplitude calculated explicitly on the I+1 grid by equation (4.29) on page 54 of the dissertation. The values of KBREAK(J) and BRK(I,J) are determined by the WDIS subroutine. The values remain at zero until the wave height (twice the absolute value of the complex amplitude) becomes equal to or greater than 0.78 time the total water depth, and is then set to one. Once set to one, the values of KBREAK(J) and BRK(I,J) remain one until the wave height becomes less than 0.4 times the water depth. The rationale for this is found in section (3.3) starting on page 45 of the dissertation or the reader can go to the original work of Dally cited in the dissertation.

3.3.3 The Circulation Model Subroutines

The governing equations for the circulation model are the depth integrated time averaged horizontal equations of momentum and continuity by equation (2.1), (2.2), and (2.3) on page 16 of Harley Winer's dissertation ([9], 1988). When all of the assumptions concerning friction and mixing are included, the alternating direction implicit (ADI) scheme is used to finite difference and solve the governing equations.

Subroutine CIRC(M, N, JJJ, N1, N2) is the circulation model. It is called directly from the main program BARR.FOR. It makes calls to the following subroutines: XCOJ1, XCOEF, XCOJN, YCOEF, YCOJJJ, YCJJJ1, YCVAR, and TRIDA. The basic subroutines in the circulation model are XCOEF and YCOEF which determine the coefficient vectors of the tridiagonal matrix. Subroutine CIRC sequentially calls the X-series subroutines. After each call to a X-series subroutine, a call is made to TRIDA to solve the tridiagonal matrix equation and then the solution vector VV is assigned to the appropriate variables. After the X-series of calls, the CIRC subroutine sequentially calls the Y-series of subroutines which make calls to TRIDA to solve the tridiagonal matrix equation and the solution vector VV is assigned to the appropriate variables. The CIRC(M, N, JJJ, N1, N2) subroutine differs from the CIRC subroutine previously described only in that for

the y sweep calls to YCOEF are made for J=2, JJJ-1 and then a call is made to YCOJJJ and then YCJJJ1 which are adaptations of YCOEF for the I=JJJ and I=JJJ+1 grid rows respectively. The only other change in the circulation model is a modification to the XCOEF subroutine. JJJ, N1, and N2 are passed in the argument of the call to subroutine XCOEF(M, J, JJJ, N1, N2).

Subroutine XCOEF(M, J, JJJ, N1, N2) is called by subroutine CIRC(M, N, JJJ, N1, N2) to determine the A, B, C, and D column vectors for each grid column in the x direction part of the alternating direction implicit method solution of the governing circulation equations. It is called sequentially for J=2, N-1. It determines all of the coefficients from I=1 to I=IWET(J). Special adaptations of XCOEF subroutine are made for any case where the J grid column is adjacent to either the domain boundary or an internal boundary imposed by the presence of a barrier. The A, B, and C arrays are the coefficients of the tridiagonal matrix equation, this is the logic for the naming of the XCOEF subroutine. JJJ, N1, and N2 are passed in the argument of the call to subroutine XCOEF(M, J, JJJ, N1, N2). The subroutine is the same as previously described except for two if statements at the end to determine if J is equal to or greater than N1 and if it is also equal to or less than N2. If it is, then A(2*JJJ), C(2*JJJ), and D(2*JJJ) are set to zero and B(2*JJJ) is set to one. These are the tridiagonal matrix coefficients for the equation $U(JJJ+1, J) = 0$.

Subroutine XCOJ1(M, J) is a special case of subroutine XCOEF(M, J) for J=1. This subroutine incorporates the lateral boundary conditions.

Subroutine XCOJN(M, J) is a special case of subroutine XCOEF(M, J) for J=N. Again, this subroutine incorporates the lateral boundary conditions.

Subroutine YCOEF(I, N) is called by subroutine CIRC(M, N, JJJ, N1, N2) to determine the A, B, C, and D column vectors for each grid row in the y direction part of the alternating direction implicit method solution of the governing equations. This subroutine is called for values of I less than IMIN. It determines the coefficients for the I grid row from J=1 to J=N.

Subroutines YCOJJJ(I, N, N1, N2) and YCJJJ1(I, N, N1, N2) are adaptations of subroutine YCOEF(I, N) for the I=JJJ grid row. These two subroutines are just like subroutine YCOEF except that for values of J=N1, N2 changes are made so that x derivatives ($\frac{\partial V}{\partial x}$, $\frac{\partial S_{xy}}{\partial x}$, $\frac{\partial^2 V}{\partial x^2}$) are not taken across the boundary. For the derivatives involving V the assumption is made that the barrier has a free-slip condition on the V velocity which enables the use of the $\frac{\partial V}{\partial x} = 0$ condition at the barrier to eliminate the value on the other side of the barrier from the derivative. For the derivatives of S_{xy} a reflective $\frac{\partial S_{xy}}{\partial x} = 0$ condition is used on the offshore side of the barrier and on the lee side the zero wave condition ($S_{xy} = 0$) at the barrier is used in a three point derivative.

Subroutine YCVAR(I, N) is an adaption of subroutine YCOEF(I, N) for the case of I equal to or greater than IMIN. The adaptation is necessary so as to include the conditions that are imposed by adjacent dry grids. (Dry is used to refer to grids where the depth is such that it is above the water level. Similarly the term wet is used to refer to grids that have water.) This subroutine is also called from the CIRC subroutine. This subroutine is full of if statements to determine the relationship of the grid to the shoreline. If the neighboring grid in the y direction is dry then the V velocity between the two is set to zero. If the I+1 grid is dry then the edge of the grid is considered the shore and three point derivatives are taken using a zero value at the shoreline. (This is done for x derivatives of V and SXY.)

TRIDA(IF, L) is a subroutine that is called by subroutine CIRC(M, N, JJJ, N1, N2) to solve the tridiagonal matrix equation using a double sweep method. IF and L are the first and last indices of the range of the column vectors A, B, C, and D which are passed in common statements. IF is usually 1 and L is usually 2*IWET(J) or 2*N+1.

EXCOEF(M, N) is the subroutine that determines the lateral mixing coefficients. It is called from the main program.

Subroutine UP(M, N) is called from the main program. It does the simple task of updating the values of U(I,J), V(I,J), and ETA(I,J) setting these values equal to those of UNEW(I,J), VNEW(I,J), and ETANEW(I,J). This subroutine also computes the ETADT terms for the first two grid rows.

Subroutine CHECK(KFLAG, M, N) is called from the main program prior to the call to subroutine UP. It checks for convergence by comparing the new values (UNEW(I,J), VNEW(I,J), ETANEW(I,J)) with the old values (U(I,J), V(I,J), ETA(I,J)). Convergence is arbitrarily set to be when the percentage difference between the two is less than an arbitrarily small epsilon EPS for each grid. The KFLAG term is a flag that the main program PRO uses to decide whether to continue for another iteration or to exit the program.

PRINT(M, N) is a subroutine that prints the values of U(I,J), V(I,J), and ETA(I,J). This subroutine was used primarily in the development stage.

Subroutine FLOOD(M, N) determines whether the boundary of the computation domain at the shoreline should be increased or reduced due to changes in the water level. This is done with IF statements to determine whether a grid has gone dry or if the elevation in a neighboring grid is sufficient to wet the shoreward grid. The flooding procedure is done only in the x direction along grid columns. When it is determined that the water surface elevation in one grid is sufficient to flood the next grid an arbitrarily small amount of water is placed in the flooded

grid. This is done so as to avoid a zero water depth in the computational domain (the wave routine will react to a zero water depth). If the depth in a grid becomes negative, then the velocity, wave height and wave set-up is set to zero, and the grid is removed from the computation domain.

Subroutine SHEAR(M, N) calculates the time averaged bottom shear stress array TBX(I,J) and TBY(I,J) using the formulation given by equations (2.15) and (2.16) on page 20 of Harley Winer's dissertation ([9], 1988) and modified by Miao.

3.3.4 The Sediment Transport Model Subroutines

The sediment transport part of the program is the SEDTR subroutine. Subroutine SEDTR(M, N, JJJ, N1, N2, ITERA, AC, AW, WAVAMP, THETAO) is called from the main program. The purpose of this subroutine is to estimate the spatial distribution of sediment transport rates from local wave and current conditions computed with the wave and the near current submodels, and to calculate the change in local bottom elevation by solving the conservation equation for sediment mass.

Do loops 10 and 12 set the initial value of bottom orbital velocity amplitude, maximum bottom shear stress, and sediment transport rate to zero. Do loops 14 and 16 estimate the position of breaking point IB(J) and plunging point IP(J), and a decay coefficient LAMBDA(J). Do loop 18 is to calculate the maximum bottom shear stress TAOMU(I,J) and TAOMV(I,J) at the J=1 boundary. Do loops 20 and 30 are to calculate the maximum bottom shear stress TAOMU(I,J) and TAOMV(I,J) in x and y direction respectively from column J=2 to N and row I=1 to IWET(J) using equations in section (2.3). Do loop 32 is to calculate the maximum bottom shear stress TAOMU(I,J) and TAOMV(I,J) at the J=N+1 boundary. Similarly, do loops 34, 36, 38 and 40 are written to calculate the sediment transport rates QX(I,J) and QY(I,J) in the x and y direction respectively using equations in section (2.3). Do loops 42, 44 and 45 perform a smoothing process to take into account the effect of bottom slope using Equation (2.5) in section (2.3). Do loop 46 is written to set the interior boundary condition QYP(I,J)=0 for the shore parallel breakwater. Finally, do loops 55 and 60 compute the change in water depth H(I,J) by solving equation (2.2) for the conservation of sediment volume.

3.3.5 Output Data

Subroutine WRITE(M, N, ITER) creates a file OUT.LST and writes out the ETA(I,J), U(I,J) and V(I,J) arrays. These values are written in free format. First ETA(I,J) is written for J=1, N for each grid row from I=1 through I=M. Then U(I,J) is written the same way and then V(I,J) is written for J=1, N+1 for each grid row. The values of the IWET(J) array are also written so that the shoreline position can be found. Subroutine WRITE(M, N, ITER) also creates files HNEW.LST and SHEAR.LST. The time varying water depth array H(I,J) is written in file HNEW.LST in time interval IPRN for J=1, N for each grid row from I=1 through I=M. Similarly, the bottom shear stress is written in file SHEAR.LST.

Subroutine AMPOUT(M, N) writes out the wave height array WAVEHGT(I,J) to a data file named HGT.LST and writes out the value of the instantaneous water surface elevation due to the wave field to another data file named ETA.LST.

3.4 MODEL SBINLET

The model SBINLET is written specifically for the area around Sebastian Inlet of Florida with open lateral boundaries and two internal barrier due to the presence of north and south jetties and an inlet boundary. The main program JETTY.FOR and all the subroutines are in subdirectory [MIAO.SBINLET]. This directory contains modifications of the program in the [MIAO.PROFILE], [MIAO.PLGR], and [MIAO.BAR] directories. The modifications are mainly for a curved north jetty and the presence of Sebastian inlet. Most of the subroutines are the same as in the [MIAO.PROFILE], [MIAO.PLGR], [MIAO.BAR] directories or with minor modifications. The main program JETTY.FOR calls many subroutines which in turn call other subroutines. The executable file is also JETTY. All of the subroutines used are listed in the file JETTY.OPT which is also used for the linking command.

- The compiling command is `$@COMPILE <CR>`, or `$FOR JETTY <CR>`, `$FOR INPUT <CR>`, etc.
- The linking command is `$LINK JETTY/OPT <CR>`.
- The running command is `$RUN JETTY <CR>`.

Within the [MIAO.SBINLET] directory there is a sample run available. The input data files are SBI.IN and SBI.DEP.

3.4.1 Input Data

Subroutine INPUT(M, N, WHGT, THETAO, TIME, ILAST, I HOUR, IPRN, AC, AW) is called by the main program JETTY.FOR. The INPUT subroutine is modified to read the position of the inlet (JINLTB and JINLTE). The north jetty is located between $J=JJTY1$ and the $J=JJTY1+1$ grid columns. The same is true for the south jetty. LLL1 and LLL2 are the grid row position of the end of the jetties. Subroutine INPUT is called only once at the beginning of the run. This subroutine prompts for the name of the input file, reads the file to set the following input parameters:

- M and N, the size of the computational domain.
- DX and DY, the grid spacing increment.
- DT, the time step increment.

- TIME, the start-up time constant.
- HO, the incident wave height at the offshore grid row.
- T, the wave period.
- THETAO, the wave angle at the offshore grid row.
- FRIC, the friction coefficient.
- EMAX, the maximum for the lateral mixing coefficient.
- ILAST, the maximum number of iterations.
- JINLTB, the starting J column position of the inlet.
- JINLTE, the ending J column position of the inlet.
- JJTY1, the J column position of the north jetty.
- JJTY2, the J column position of the south jetty.
- LLL1, the I grid row position of the end of the straight portion of the north jetty.
- LLL2, the I grid row position of the end of the south jetty.
- TIDERG, the tidal range.
- IHOURL, running duration in hours.
- IPRN, print out interval.
- AC and AW, sediment transport coefficient for current and wave.
- ICP, number of grids inside breaking point for plunging point position.

All of the above input parameters are separated by commas in the first two lines of the input file, so that the read statement in subroutine INPUT(M, N, WHGT, THETAO, TIME, ILAST, IHOURL, IPRN, AC, AW) is in free format. A sample input file is SBI.IN in the [MIAO.SBINLET] directory. The subroutine then prompts for the name of the depth file, opens it and reads the M by N array of the depth H(I,J). H(I,J) is measured positive downward from the still water line. Thus land above the still water line will be represented by a negative value. It is important that M be large enough to allow for flooding of the shoreline as wave set-up is produced.

3.4.2 Output Data

Subroutine WRITE(M, N, ITER) creates a file OUT.LST and writes out the ETA(I,J), U(I,J) and V(I,J) arrays. These values are written in free format. First ETA(I,J) is written for J=1, N for each grid row from I=1 through I=M. Then U(I,J) is written the same way and then V(I,J) is written for J=1, N+1 for each grid row. The values of the IWET(J) array are also written so that the shoreline position can be found. Subroutine WRITE(M, N, ITER) also creates files OUTPC.LST, BRK.LST, VT.LST, HNEW.LST, and DH.LST. The grid index (I, J), the water depth array H(I,J), and the water depth difference HO(I,J)-H(I,J) are written in file OUTPC.LST. The wave breaking index BRK(I,J) is written in file BRK.LST with 1 indicating breaking waves and 0 indicating non-breaking waves. The total velocity array VT(I,J) is written in file VT.LST. The time varying water depth array H(I,J) is written in file HNEW.LST in time interval IPRN for J=1, N for each grid row from I=1 through I=M. Similarly, the depth difference HO(I,J)-H(I,J) is written in file DH.LST.

Subroutine AMPOUT(M, N) writes out the wave height array WAVEHGT(I,J) to a data file named HGT.LST and writes out the value of the instantaneous water surface elevation due to the wave field to another data file named ETA.LST.

Chapter 4

CONCLUSIONS

A numerical model has been developed for computing time-dependent nearshore hydrographic changes including beach profile responses. The time scale of the model is suitable for storm events to seasonal changes, currently up to one year period. The model is very stable and is capable of handling complicated topographies including inlets and irregularly-shaped structures such as curved jetties and breakwaters.

The hydrodynamic model computes fully interacted current and wave fields based on coupled mild slope wave equation and depth-averaged circulation equations. The rate of sediment transport is treated as the summation of two energetic mechanisms one due to the mean current and the other due to the wave induced turbulence.

The model has been successful comparing with the evolution of beach profiles in the 2-D GWK tank tests as well as the CERC's large tank tests. The model is capable of describing the growth and movement of main breakpoint bars and corresponding berm processes with reasonable reliability. It is also tested against results from 3-D movable bed experiments of an inlet-beach system. Overall model performance is satisfactory. This model can provide estimates of beach erosion/accretion, shoreline change, gross and net longshore and/or cross-shore sediment transport, and the volume of sand transported out and/or into the inlet. It can also be used to judge the relative behavior and merits of various coastal structure constructions or beach nourishments. Because of its great power and generality, this model can provide a framework for developing shore protection problem and solution statements, for organizing the collection and analysis of data and, most importantly, for evaluating alternative designs and optimizing the selected design. Testing of this class of models must continue, with emphasis on field verification and model refinement.

Bibliography

- [1] Dally, W. R. , Dean, R. G. and Dalrymple, R. A. (1984). "A Model for Breaker Decay on Beaches" Proc. 19th Coastal Eng. Conf. , Houston, 82-98.
- [2] Ebersole, B. A., and Dalrymple, R. A. (1979). "A Numerical Model for Nearshore Circulation Including Convective Acceleration and Lateral Mixing" Ocean Engineering Report No. 21, Dept. of Civil Eng., University of Delaware, Newark, Delaware.
- [3] Kirby, J. T. (1984). "A Note on Linear Surface Wave-Current Interaction over Slowly Varying Topography" J. Geophys. Res., Vol. 89, No. C1, pp. 745-747.
- [4] Larson, M., Kraus, N. C. and Byrnes, M. R. (1989). "SBEACH: Numerical Model for Simulating Storm-Induced Beach Change" Technical Report CERC-89-9, Coastal Engineering Research Center, US Army Corps of Engineers.
- [5] Ohnaka, S. and Watanabe, A. (1990). "Modeling of Wave-Current Interaction and Beach Change" Proceedings of the 22nd Coastal Engineering Conference, ASCE, Vol. 3, pp. 2443-2456, July 2-6, 1990, Delft, The Netherlands.
- [6] Wang, H., Lin, L., Zhong, H., and Miao, G. (1991). "Sebastian Inlet Physical Model Studies, Part I — Fixed Bed Model" Coastal and Oceanographic Engineering Department, University of Florida. UFL/COEL-91/001.
- [7] Wang, H., Lin, L., and Miao, G. (1992). "Sebastian Inlet Physical Model Studies, Part II — Movable Bed Model" Coastal and Oceanographic Engineering Department, University of Florida. UFL/COEL-92/006.
- [8] Watanabe, A. and Maruyama, K. (1986). "Numerical Modeling of Nearshore Wave Field under Combined Refraction, Diffraction and Breaking" Coastal Engineering in Japan, Vol. 29, pp. 19-39.
- [9] Winer, H., S. (1988) "Numerical Modeling of Wave-Induced Currents Using a Parabolic Wave Equation" Coastal and Oceanographic Engineering Department, University of Florida. UFL/COEL-TR/080.