

UFL/COEL/MP-88/1

**DOCUMENTATION OF THE
WAVE-INDUCED CIRCULATION MODEL**

By

Harley S. Winer

Version 1.0

August, 1988

DOCUMENTATION OF THE
WAVE-INDUCED
CIRCULATION MODEL

by

HARLEY S. WINER

Version 1.0
August 1988

TABLE OF CONTENTS

1	INTRODUCTION AND OVERVIEW OF THE PROGRAM	1
1.1	Common Statements	1
1.2	Brief Description of the Subroutines	4
2	THE WAVE MODEL IN GREATER DETAIL	10
2.1	The WAVRAD Subroutine	10
2.2	The AMPCO Subroutine	11
2.2.1	The J=2,N DO LOOP	12
2.2.2	The J=1 and J=N Boundaries	13
2.3	The RAD Subroutine	14
3	THE CIRCULATION MODEL IN GREATER DETAIL	16
3.1	The XCOEF Subroutine	17
3.2	The YCOEF Subroutine	19
3.3	The CIRC Subroutine	20
3.4	The FLOOD subroutine	21
4	VARIATIONS OF THE PROGRAM	23
4.1	The [WINER.BARR] Directory	23
4.2	The [WINER.GROIN] Directory	27
4.3	The [WINER.CLOSED] Directory	29
4.4	The [WINER.PLOT] Directory	30
APPENDICES		
A	VARIABLES USED IN THE COMPUTER CODE	32

CHAPTER 1 INTRODUCTION AND OVERVIEW OF THE PROGRAM

This report is to document the programs that I have developed for my dissertation. There are actually several separate programs in separate directories. They are named PRO, GROIN, and BREAK. Each of these programs is written for a Cartesian grid. The PRO program has no internal barriers. GROIN is an adaptation of PRO so as to include the internal boundary conditions that would exist at a shore perpendicular barrier. Similarly, BREAK is an adaptation of PRO for the case of a shore parallel breakwater. In both cases the barrier is represented as infinitesimally thin and impermeable. The two adaptations GROIN and BREAK will be described after the mechanics of PRO are explained. Also an adaptation for closed basins is included.

The main program PRO 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 linking command is

```
$LINK PRO/OPT <cr>
```

Each of the subroutines listed in PRO.OPT will be described separately. For reference the file PRO.OPT is printed below.

```
PRO,INPUT,TRIDA,XCOEF,XCOJ1,XCOJN,YCOEF,YCVAR,EXCOEF  
UP,CHECK,PRINT,WAVRAD,WAVNUM,AMPCO,DISP,FLOOD,SHEAR2  
CTRIDA,RAD,WDIS,CIRC,AMPOUT,WRITE
```

1.1 Common Statements

Each of the subroutines has as the second line

```
INCLUDE 'DIM.FOR'
```

where DIM.FOR is a file that has all of the dimension statements that are used as common statements. The reason for having all of the common statements in a separate file is to facilitate changing the size of the domain. This will be useful if the program is to be run on a larger computer for a larger domain. At present the dimension of the domain is 100 by 100. This is because the VAX 750 has limited storage.

The DIM.FOR file is printed below.

```

COMPLEX AA,BB,CC,DD,VVV,AMP,AMPO
COMMON/A/U(100,100),UNEW(100,100)
COMMON/B/ETA(100,100),ETANEW(100,100),ETADT(100,100)
COMMON/C/V(100,100),VNEW(100,100)
COMMON/D/H(100,100),TD(100,100),TD2(100,100)
COMMON/E/A(200),B(200),C(200),D(200),VV(200)
COMMON/H/WAVEHGT(100,100)
COMMON/I/TBX(100,100),TBY(100,100)
COMMON/J/SXX(100,100),SXY(100,100),SYY(100,100)
COMMON/K/G,DT,DX,DY,FRIC,T,RO,IWET(100)
COMMON/L/AA(100),BB(100),CC(100),DD(100),VVV(100)
COMMON/LL/AMP(100,100),AMPO(100)
COMMON/M/RK(100,100),RKO(100),CG(100,100),SIG(100,100)
COMMON/M/KBREAK(100)
COMMON/O/EX(100,100),EY,EMAX
COMMON/P/THETA(100),THETAH

```

Appendix A lists each of these variables along with many other variables in alphabetical order with a brief description.

The wave model part of the program solves for the complex amplitude at each of the grid points. The complex amplitude array is named AMP. The complex amplitude on the first grid row is named AMPO. The AMPO array is used so as to enable a smooth start-up. AMP on the first row is given by a time start function times AMPO. This is explained on page 59-60 of my dissertation. The arrays AA, BB, CC, and DD are the column vectors for the tridiagonal matrix equation obtained from the parabolic wave equation. These are complex. VVV is the solution vector for the tridiagonal matrix equation obtained from the parabolic wave equation. AA, BB, CC, DD, and VVV which are complex are distinct from the column arrays A, B, C, D, and VV which are the column vectors and solution vector for the non-complex tridiagonal matrix equation obtained in the circulation model.

The circulation model solves for the mean water surface elevation and the x and y components of the depth integrated mean current. U and V are the values of the x and y components of the current and ETA is the mean surface elevation measured upwards from the still water line. UNEW, VNEW and ETANEW are the new values of the current and surface elevation which are determined by the CIRC subroutine. ETADT is the time rate of change of ETA. This is used at the off shore boundary for one of the non-linear advective acceleration terms. This will be explained later. A, B, C, and D are the column vectors for the tridiagonal matrix equation obtained from the governing circulation equations. VV is the solution vector. It should be noted that A, B, C, D, and VV have a dimension that is twice that of the variables U, V, and ETA. This is because the tridiagonal matrix equation, which is solving for a staggered grid system, solves for both U and ETA in the x direction and for V and ETA in the y direction.

H is the depth measured downward from the still water line and TD is the total depth equal to H plus ETA. TD2 is also the total depth and is equal to TD when ever TD is positive. If TD is zero or less than zero, TD2 is set to an arbitrary value of 1 millimeter of water. The TD2 array is used in the wave model so as to avoid the necessity of internal boundary conditions.

WAVEHGT is the wave height which is actually twice the absolute value of the complex amplitude AMP. The WAVEHGT array is used in the SHEAR2 subroutine to determine the bottom shear stresses which are designated by the array names TBX and TBY. TBX is named for Tau Bottom in the X direction and TBY is for Tau Bottom in the Y direction. Two separate arrays are used since each is positioned at a different grid edge.

SXX, SXY, and SYX are the radiation stress components which are computed in the RAD subroutine. RK is the wave number, CG is the group velocity, SIG is the relative frequency, and RKO is an averaged wave number which is obtained by averaging RK across a grid row. RK, CG, SIG, and RKO are calculated in the subroutine WAVNUM. THETA is the wave angle at each grid row assuming a plane beach without currents as calculated by

Snell's law. THETAH is the wave angle calculated by Snell's law for the position half way between grid centers (i.e. at the grid edge) at the location of the breakwater. THETAH is used only in the BREAK program.

G, DT, DX, DY, FRIC, T, and RO are the gravitational acceleration constant, delta t, delta x, delta y, the friction factor, the wave period and ρ the water density respectively. IWET is an array which indicates the position of the shoreline for each of the grid columns. This array is used to allow the shoreline to move with raising or lowering of the water level. This mechanism will be discussed further in the section (3.4 explaining the FLOOD subroutine.

KBREAK is an index which is used in the WDIS subroutine to indicate whether or not dissipation due to breaking should be applied. This is discussed further in the description of the subroutine WDIS below.

EX, and EY are eddy viscosity coefficients which are calculated in the subroutine EX-COEF. EMAX is the maximum value that these coefficients may be. EMAX is used so as to place an upper limit on the eddy viscosity coefficients and thus avoid numerical stability problems that may arise from too large a coefficient. This is important since the lateral mixing terms are represented explicitly in the model and thus place a constraint on the size of either the delta t increment or the lateral mixing coefficient.

1.2 Brief Description of the Subroutines

INPUT(M,N,WAVAMP,THETAO,TIME,ILAST) is called by the main program. 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 size of the computational domain M and N the grid spacing DX and DY, the time step increment DT, the start-up time constant TIME, the incident wave height HO at the offshore grid row, the wave period T, the wave angle at the offshore grid row THETAO, the friction coefficient FRIC, the maximum for the lateral mixing coefficient EMAX, and the maximum number of iterations ILAST. For the special programs GROIN and BREAK the input data file also contains the position of the structure.

All of the above are separated by commas in the first line of the input file, so that the read statement is in free format. A sample input file is IN.PL in the [WINER.MODEL] 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. H 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 set-up is produced. After reading the H depth array the subroutine determines the values of IWET(J) for all values of J=1,N. This is simply done by determining for each grid column the first grid from the shoreline to have a positive depth. Values for IMIN and IMAX the minimum and maximum of the IWET array are also determined. The values of IWET(J), IMIN, and IMAX determine the limits of the actual computation in the circulation model. The program updates the values of IWET, IMIN, and IMAX with each iteration by calls to the FLOOD subroutine from the main program. The subroutine initializes the values of U, V, and ETA to be zero. The final task of the INPUT subroutine is to determine the values of AMPO. This is done as follows. First a call is made to the subroutine DISP to determine the wavenumber on the offshore grid row. Then the real part of AMPO is given by $HO \cos(Y RK1 \cos(\text{THETAO}))$ and the imaginary part by $HO \sin(Y RK1 \cos(\text{THETAO}))$, where RK1 is the wave number along the first grid row assuming no currents, and Y is the distance along the y coordinate of the grid center from the reference $y=0$ position, and HO and THETAO are the wave height and wave angle at the offshore grid row.

TRIDA(IF,L) is a subroutine that is called by the CIRC subroutine 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 $2 \cdot IWET(J)$ or $2 \cdot N + 1$.

XCOEF(M,J) is called by the CIRC subroutine 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 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.

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

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

YCOEF(I,N) is called by the CIRC subroutine 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 $J=N$.

YCVAR(I,N) is an adaption of the YCOEF subroutine 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 statement 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).

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

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

CHECK(KFLAG,M,N) is called from the main program prior to the call to UP. It checks for convergence by comparing the new values (UNEW, VNEW, ETANEW) with the old values (U, V, ETA). Convergence is arbitrarily set to be when the percentage difference between the two is less than an arbitrarily small epsilon EPS for each grid and for each unknown U, V, and ETA. The KFLAG term is a flag that the main 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, V, and ETA. This subroutine was used primarily in the development stage.

WAVRAD(M,N,THETAO,WAVHT) is the wave model subroutine. It is called from the main program. It in turn makes calls to the subroutines WAVNUM, DISP, AMPCO, CTRIDA, and RAD. The purpose of this subroutine (and those that it calls) is to determine the radiation stress terms. This subroutine is the same as WAVE shown in the flow chart of the model on page 11 of the dissertation. WAVHT is the start-up function $C(t)$ times WAVAMP which is one half the incident wave height. WAVHT times AMPO(J) gives the value of AMP(1,J) to start the wave model marching.

WAVNUM(M,N,THETAO) is a subroutine that calculates the wavenumber RK, the averaged wave number across a grid row RKO, the group velocity CG, and the relative frequency SIG. A Newton method is used to solve for RK using equation (3.76) on page 44 of the dissertation. RKO is obtained by a simple averaging while CG and SIG are directly obtained from the appropriate formulas. The WAVNUM subroutine also solves for the THETA array using Snell's law. To do this calls are made to DISP to determine the wavenumber in the absence of 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.

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.

AMPCO(I,N) determines the coefficients of the tridiagonal matrix equation which re-

sults from the finite differencing of the parabolic wave equation. The governing equation used is equation (4.26) on page 52 of the dissertation. It is finite differenced using a Crank-Nicholson scheme and is written as equation (4.27) on page 53 of the dissertation. AMPCO is called sequentially for $I=1, IMAX-1$ from the WAVRAD subroutine.

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

RAD(M,N) calculates the radiation stress components SXX, SXY, and SYY from the gradients in the complex amplitude AMP using equation (2.57) on page 29 of the dissertation.

WDIS(I,J,TTTDD,CCGG,AATEMP,WAVDIS) is called from the AMPCO subroutine. It computes the value of the dissipation coefficient in the governing wave equation using equation (3.97) on page 47 of the dissertation. TTTDD is the total depth and CCGG is the group velocity. These values are passed in the argument rather than through common statements because of the need to compute the dissipation at the grid edge in the BREAK program. 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 use of equation (4.29) on page 54 of the dissertation.

CIRC(M,N) is the circulation model. It is called directly from the main program. It makes calls to XCOJ1, XCOEF, XCOJN, YCOEF, YCVAR, and TRIDA.

AMPOUT(M,N) writes out the absolute value of the complex amplitude to a data file named AMP.LST and writes out the value of the instantaneous water surface elevation due to the wave field to another data file named ETA.LST.

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 a lot of 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. The procedure is done in such a way as

to insure continuity of water mass in the domain. 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 taken from the one grid and 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 deficient amount is taken from the neighboring grid to produce a zero water column and the grid is removed from the computation domain.

SHEAR2(M,N) calculates the TBX and TBY arrays using the formulation given by equations (2.15) and (2.16) on page 20 of the dissertation.

WRITE(M,N) creates a file OUT.LST and writes out the ETA, U and V arrays. These values are written in free format. First ETA is written for $J=1,N$ for each grid row from $I=1$ through $I=IMAX$. Then U is written the same and then V is written for $J=1,N+1$ for each grid row. The values of the IWET array are also written so that the shoreline and shore values can be found.

CHAPTER 2 THE WAVE MODEL IN GREATER DETAIL

2.1 The WAVRAD Subroutine

The wave model part of the program is the WAVRAD subroutine. It makes calls to the following subroutines: WAVNUM, AMPCO, CTRIDA, and RAD. The WAVNUM subroutine in turn makes calls to DISP and the AMPCO subroutine makes calls to WDIS.

Within the WAVRAD subroutine the first do loop is to set the KBREAK array equal to zero. This array is used by the WDIS subroutine to determine whether dissipation due to wave breaking should be included. If KBREAK(J) is equal to zero than the wave dissipation is set to zero. If KBREAK(J) is one than dissipation is a positive value. The value of KBREAK is determined by the WDIS subroutine. The value remains at zero until the wave height (twice the absolute value of the complex amplitude) becomes equal to or greater than .78 time the total water depth, and is than set to one. Once set to one, the value of KBREAK(J) remains one until the wave height becomes less than .4 times the water depth. The rational 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.

Within the same do loop setting KBREAK equal to zero the value of AMP on the first grid row is determined by the expression $AMP(1,J)=WVAMP*AMPO(J)$, where WVAMP is passed from the main program. Some clarification is needed concerning the use of several variables such as WAVAMP and WVAMP and the array WAVEHGT. WAVAMP is equal to one half of the input incident wave height. It is set by INPUT and remains unchanged for the entire program. WVAMP is determined in the main and is the time function $C(t)$ of equation (4.45) on page 60 times WAVAMP. The time function $C(t)$ has the name UU in the main. WAVEHGT is the array of the wave height determined in the RAD subroutine

and is merely twice the absolute value of the complex amplitude AMP.

The WAVRAD subroutine then calls WAVNUM to calculate the RK, RKO, CG, THETA, and SIG arrays. After the call to WAVNUM there is the following do loop

```

DO 20 I=1,IMAX-1
CALL AMPCO(I,N)
CALL CTRIDA(1,N)
DO 20 J=1,N
AMP(I+1,J)=VVV(J)
20

```

(2.1)

This do loop is the meat of the wave model. It is the marching grid row by grid row of the parabolic equation. The AMPCO subroutine determines the column vectors AA, BB, CC, and DD of the tridiagonal matrix equation. The AA, BB, CC, and DD vectors are determined very straight forward. Grouping terms, equation (4.27) on page 53 of the dissertation is written as

$$AA(J) A_{j-1}^{i+1} + BB(J) A_j^{i+1} + CC(J) A_{j+1}^{i+1} = DD(J) \quad (2.2)$$

This equation then is the definition of the AA, BB, CC, and DD column vectors. CTRIDA then solves the tridiagonal matrix equation producing the solution vector VVV. And then the solution vector is assigned to the AMP values on the next or I+1 grid row until values of the AMP array are determined for every grid in the IMAX by N domain.

The last line in WAVRAD before the return to the main program is a call to RAD to calculate the radiation stress arrays according to equation (2.57) on page 29 of the dissertation.

2.2 The AMPCO Subroutine

As explained above the AMPCO subroutine determines the AA, BB, CC, and DD column vectors of the tridiagonal matrix equation resulting from the finite differencing of the governing parabolic 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. The do loop for J=2,N will be examined first.

2.2.1 The J=2,N DO LOOP

The first eight lines of the do loop are to obtain spatially centered velocities. This is necessary since the velocities U and V are at the grid edges while all other variables are at grid centers. The eight values determined are VIJ, VIP1J, VIJP1, VIP1JP1, VIJM1, VIP1JM1, UIJ, and UIP1J. The logic of these code names is the velocity and the position. For example, VIP1JM1 is the V velocity at the I plus 1 and J minus 1 grid (V_{j-1}^{i+1}), while UIP1J is the U velocity at the I plus 1, J (U_j^{i+1}) grid center. These values are the average of the two values at the opposite grid edges. On the next two lines CGPU is $[C_g \cos \theta + U]_j^i$ and CGPIU is $[C_g \cos \theta + U]_j^{i+1}$.

The next several lines compute some intermediary terms which are used for short hand purposes (not necessarily for computational efficiency). These terms are listed below

$$\begin{aligned}
 A2 &= \left(\frac{V}{(C_g \cos \theta + U)} \right)_j^{i+1} \\
 B2 &= \frac{\left(\frac{C_g \cos \theta + U}{\sigma} \right)_j^{i+1} - \left(\frac{C_g \cos \theta + U}{\sigma} \right)_j^i}{\left(\frac{C_g \cos \theta + U}{\sigma} \right)_j^{i+1} + \left(\frac{C_g \cos \theta + U}{\sigma} \right)_j^i} \\
 C2 &= \left(\frac{\sigma}{C_g \cos \theta + U} \right)_j^{i+1} \frac{\left(\frac{V}{\sigma} \right)_{j+1}^{i+1} - \left(\frac{V}{\sigma} \right)_{j-1}^{i+1}}{4 DY} \\
 D2 &= \left(\frac{\sigma}{C_g \cos \theta + U} \right)_j^i \frac{\left(\frac{V}{\sigma} \right)_{j+1}^i - \left(\frac{V}{\sigma} \right)_{j-1}^i}{4 DY} \\
 PAA &= \frac{\left(\frac{\sigma C_g}{k} \right)_{j+1}^{i+1} + \left(\frac{\sigma C_g}{k} \right)_j^{i+1}}{2 DY^2 (C_g \cos \theta + U)_j^{i+1}} \\
 PBB &= \frac{\left(\frac{\sigma C_g}{k} \right)_j^{i+1} + \left(\frac{\sigma C_g}{k} \right)_{j-1}^{i+1}}{2 DY^2 (C_g \cos \theta + U)_j^{i+1}} \\
 PCC &= \frac{\left(\frac{\sigma C_g}{k} \right)_{j+1}^i + \left(\frac{\sigma C_g}{k} \right)_j^i}{2 DY^2 (C_g \cos \theta + U)_j^i} \\
 PDD &= \frac{\left(\frac{\sigma C_g}{k} \right)_j^i + \left(\frac{\sigma C_g}{k} \right)_{j-1}^i}{2 DY^2 (C_g \cos \theta + U)_j^i} \\
 CSQ &= \left(\frac{k C_g (1 - \cos^2 \theta)}{2 (C_g \cos \theta + U)} \right)_j^i
 \end{aligned}$$

$$\begin{aligned}
 CSQP1 &= \left(\frac{k C_g (1 - \cos^2 \theta)}{2 (C_g \cos \theta + U)} \right)_j^{i+1} \\
 CS &= (\bar{k}^i - k_j^i) \cos \theta^i \\
 CSP1 &= (\bar{k}^{i+1} - k_j^{i+1}) \cos \theta^{i+1}
 \end{aligned}$$

The PAA, PBB, PCC, and PDD terms are used for the double y derivative as expressed by equation (4.28) on page 54 of the dissertation.

The next three lines are

$$\begin{aligned}
 AMPIJ &= AMP(I,J) \\
 TTDD &= TD(I,J) \\
 CCG &= CG(I,J)
 \end{aligned} \tag{2.3}$$

These three lines are used to set values to be passed in the argument of the call to the WDIS subroutine. These values are passed in the argument rather than through common statements because for some applications such as a shore parallel breakwater the WDIS subroutine is called for the location at the grid edge where the depth is not an array. Also the second call to WDIS within the J=2,N do loop is for a temporary value of the wave amplitude that is not retained as an array.

The first call to WDIS returns the wave dissipation value W1J. This is the w_j^i term. Next an intermediate value of A_j^{i+1} is obtained using the explicit formulation of equation (4.29) on page 54 of the dissertation. This intermediate value of A_j^{i+1} is designated as ATEMP in the program. Using this intermediate ATEMP value a second call to WDIS is made returning the value of W2J, the w_j^{i+1} term.

Finally the AA(J), BB(J), CC(J), and DD(J) terms are determined using all of the short hand terms listed above.

2.2.2 The J=1 and J=N Boundaries

The J=1 and J=N values of the AA, BB, CC, and DD arrays are determined in a manner very similar to the determination in the above do loop except that certain boundary conditions are incorporated. First a value of ANG is determined by solving equation (4.47) on page 60 of the dissertation for m . Equation (4.47) is finite differenced on the I grid row

with known values of AMP. Thus

$$m = -\frac{i(A_2^i - A_1^i)}{DY \frac{(A_2^i + A_1^i)}{2}} \quad (2.4)$$

or in code $ANG = -AII * (AMP(I,2) - AMP(I,1)) / (DY * (AMP(I,2) + AMP(I,1)) / 2)$. There is of course a safety valve to avoid division of zero so that if $A_2^i + A_1^i$ is zero ANG is set to zero. The same procedure is used for $J=N$ using A_N^i and A_{N-1}^i to determine a value for ANG.

It is not really necessary to calculate a value for m since it would be easier to use $m = k_o \sin \theta_o = \text{constant}$ throughout. There is no advantage to computing m at each boundary for each grid row. However that is how it currently is and perhaps if other boundary conditions such as longshore gradients in the water surface due to tidal circulation are added in the future, the present method may be needed.

Once m is determined then the coefficient BVAL is determined

$$\frac{1 - im \frac{\Delta Y}{2}}{1 + im \frac{\Delta Y}{2}} \quad (2.5)$$

This term is used to eliminate A_0^i using equation (4.35) on page 55 of the dissertation. For $J=N$ A_{N+1}^i is eliminated using equation (4.36) on page 55 of the dissertation.

Other boundary conditions in the $J=1$ and $J=N$ computation are that there are no gradients of the current, depth, wave number, and group velocity terms across the lateral boundaries. This means that values of SIG, CG, RK, and V for grids outside the computational domain (but adjacent to the boundary) are eliminated in terms of the value of the term at the grid inside the domain (and adjacent to the boundary).

2.3 The RAD Subroutine

In the RAD subroutine the first task is to compute the WAVEHGT array which will be used in the SHEAR2 subroutine. Then the radiation stress terms are computed using equations (2.57) and (2.59) on page 29 of the dissertation. 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 off shore 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. Finally 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 following are some of the terms that are calculated for intermediary use

$$\begin{aligned}
 TWOKH &= 2kh \\
 DADX &= \frac{\partial B}{\partial X} = \frac{\partial A}{\partial x} + i\bar{k} \cos \theta A \\
 DACONDX &= \frac{\partial B^*}{\partial x} = \frac{\partial A^*}{\partial x} - i\bar{k} \cos \theta A^* \\
 DADY &= \frac{\partial B}{\partial y} = \frac{\partial A}{\partial y} \\
 DACONDY &= \frac{\partial B^*}{\partial y} = \frac{\partial A^*}{\partial y} \\
 AC &= |B| = AA^*
 \end{aligned}$$

Using these values the SXX, SYY, and SXY terms are then computed using equation (2.57) on page 29 of the dissertation. The code is straight-forward and should not be difficult to decipher using the terms above.

The do loop which calculates the radiation stress terms on the grid adjacent to the shore line uses a three point x derivative. The three points are the zero amplitude at the shore edge of the IWET(J) grid and the two amplitude values at the IWET(J) and IWET(J)-1 grid centers. Using a Taylor series to expand the terms about the IWET(J) grid center it can be shown that to second order the x derivative of $\frac{\partial A}{\partial x}$ is written as

$$\frac{(-3A_J^{IWET(J)} - A_J^{IWET(J)-1})}{3\Delta X} \quad (2.6)$$

with a similar differencing for the complex conjugate.

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. Technically this is not a correct application of the boundary condition. Perhaps more correct would be to use the $\frac{\partial A}{\partial y} = imA$ boundary condition. However I don't think that this introduces any significant error.

CHAPTER 3
THE CIRCULATION MODEL IN GREATER DETAIL

The governing equations for the circulation model are the depth integrated time averaged horizontal equations of momentum and continuity given by equations (2.1), (2.2), and (2.3) on page 16 of the dissertation. When all of the assumptions concerning friction and mixing are included, and the numerical ADI scheme described in section 4.1 is used these equations are finite differenced as follows. In the x direction the momentum equation is

$$\begin{aligned} & \frac{UNEW_{i,j} - U_{i,j}}{\Delta T} + g \frac{ETANEW_{i,j}^* - ETANEW_{i-1,j}^*}{\Delta X} \\ & + U_{i,j} \frac{U_{i+1,j} - U_{i-1,j}}{2\Delta X} + \frac{1}{4}(V_{i,j} + V_{i-1,j} + V_{i,j+1} + V_{i-1,j+1}) \frac{U_{i,j+1} - U_{i,j-1}}{2\Delta Y} \\ & + \frac{1}{\rho \bar{D}} \left(\frac{SXX_{i,j} - SXX_{i-1,j}}{\Delta X} + .5 \frac{SXY_{i,j+1} - SXY_{i,j-1} + SXY_{i-1,j+1} - SXY_{i-1,j-1}}{2\Delta Y} \right) \\ & + \frac{1}{\rho \bar{D}} TBX_{i,j} UNEW_{i,j} - EY \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{\Delta Y^2} = 0 \quad (3.1) \end{aligned}$$

and the continuity equation is

$$\frac{ETANEW_{i,j}^* - ETA_{i,j}}{\Delta T} + \frac{(\bar{D}UNEW)_{i+1,j} - (\bar{D}UNEW)_{i,j}}{\Delta X} = - \frac{(\bar{D}V)_{i,j+1} - (\bar{D}V)_{i,j}}{\Delta Y} \quad (3.2)$$

where \bar{D} is the depth at the grid edge obtained by averaging the values at the two appropriate grid centers.

For the y direction the momentum equation is differenced as

$$\begin{aligned} & \frac{VNEW_{i,j} - V_{i,j}}{\Delta T} + g \frac{ETANEW_{i,j} - ETANEW_{i,j-1}}{\Delta Y} \\ & + \frac{1}{4}(U_{i,j} + U_{i-1,j} + U_{i,j+1} + U_{i-1,j+1}) \frac{V_{i+1,j} - V_{i-1,j}}{2\Delta X} + V_{i,j} \frac{V_{i,j+1} - V_{i,j-1}}{2\Delta Y} \\ & + \frac{1}{\rho \bar{D}} \left(.5 \frac{SXY_{i+1,j} - SXY_{i-1,j} + SXY_{i+1,j-1} - SXY_{i-1,j-1}}{2\Delta X} + \frac{SYY_{i,j} - SYY_{i,j-1}}{\Delta Y} \right) \\ & + \frac{1}{\rho \bar{D}} TBY_{i,j} VNEW_{i,j} + EX_{i,j} \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{\Delta X^2} = 0 \quad (3.3) \end{aligned}$$

and the continuity equation as

$$\frac{ETANEW_{i,j} - ETANEW_{i,j}^*}{\Delta T} + \frac{(\bar{D}VNEW)_{i,j+1} - (\bar{D}VNEW)_{i,j}}{\Delta Y} = \frac{(\bar{D}V)_{i,j+1} - (\bar{D}V)_{i,j}}{\Delta Y} \quad (3.4)$$

where $\bar{E}X_{i,j}$ is the averaged EX term. Note that the $\frac{(\bar{D}V)_{i,j+1} - (\bar{D}V)_{i,j}}{\Delta Y}$ term is subtracted from the right hand side of (3.2) and added to the right hand side of (3.4). Thus if the two equations (3.2) and (3.4) are added, a fully implicit two dimensional continuity equation results.

The basic subroutines in the circulation model are XCOEF and YCOEF which determine the coefficient vectors of the tridiagonal matrix.

3.1 The XCOEF Subroutine

If equations (3.1) and (3.2) are written as

$$A_{2i-2}ETANEW^*(I-1,J) + B_{2i-2}UNEW(I,J) + C_{2i-2}ETANEW^*(I,J) = D_{2i-2} \quad (3.5)$$

and

$$A_{2i-1}UNEW(I,J) + B_{2i-1}ETANEW^*(I,J) + C_{2i-1}UNEW(I+1,J) = D_{2i-1} \quad (3.6)$$

then

$$A(2^*I-2) = -g * RX$$

$$B(2^*I-2) = 1 + \Delta T * TBX(I, J) / (\rho \bar{D})$$

$$C(2^*I-2) = g * RX$$

$$\begin{aligned} D(2^*I-2) = & U(I, J) - \Delta T * U_{i,j} \frac{U_{i+1,j} - U_{i-1,j}}{2\Delta X} \\ & - \frac{\Delta T}{4} (V_{i,j} + V_{i-1,j} + V_{i,j+1} + V_{i-1,j+1}) \frac{U_{i,j+1} - U_{i,j-1}}{2\Delta Y} \\ & - \frac{\Delta T}{\rho \bar{D}} \left(\frac{SXX_{i,j} - SXX_{i-1,j}}{\Delta X} + .5 \frac{SXY_{i,j+1} - SXY_{i,j-1} + SXY_{i-1,j+1} - SXY_{i-1,j-1}}{2\Delta Y} \right) \\ & + \Delta T * EY \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{\Delta Y^2} \end{aligned}$$

$$A(2^*I-1) = -RX * \bar{D}_{i,j}$$

$$B(2^*I-1) = 1.0$$

$$\begin{aligned}
 C(2*I-1) &= RX * \bar{D}_{i+1,j} \\
 D(2*I-1) &= ETA_{i,j} - RY * (V_{i,j+1} \bar{D}_{i,j+1} - V_{i,j} \bar{D}_{i,j})
 \end{aligned}
 \tag{3.7}$$

where $RX = \frac{\Delta T}{\Delta X}$ and $RY = \frac{\Delta T}{\Delta Y}$.

The above are the basis for all of the subroutines that have an X as the first letter in the name. Variations of the above are found near the boundaries or next to internal boundaries due to the presence of structures. Some examples of this follow.

In the XCOEF subroutine the offshore boundary condition of $\bar{\eta} = 0$ is given in the code as

$$\begin{aligned}
 A(1) &= 0.0 \\
 B(1) &= 1.0 \\
 C(1) &= 0.0 \\
 D(1) &= 0.0
 \end{aligned}$$

When (3.7) is written for $I=2$ there is the problem that $U_{i-1,j}$ is not defined since it is outside of the computational domain which goes offshore only as far as the center of the first grid row. The $U_{i-1,j}$ term, which is used in the advective acceleration terms, is avoided by using the continuity equation to solve for $\frac{\partial U}{\partial x}$.

$$\frac{\partial U}{\partial x} = -\frac{1}{D} \left[\frac{\partial \bar{\eta}}{\partial t} + U \frac{\partial D}{\partial x} + \frac{\partial V * D}{\partial y} \right]
 \tag{3.8}$$

The right hand side of the above equation is thus used for the x derivative of U in the computation of the D(2) term.

The shore boundary condition of no flow into the shore is represented as $U_{IWET(J)+1,J} = 0$. In code this is

$$\begin{aligned}
 A(2*IWET(J)) &= 0.0 \\
 B(2*IWET(J)) &= 1.0 \\
 C(2*IWET(J)) &= 0.0 \\
 D(2*IWET(J)) &= 0.0
 \end{aligned}$$

Once the XCOEF subroutine is understood the other X-series subroutines are easy to decipher. Each one is merely an adaptation to make sure that y derivatives and depth averages are not taken across a barrier or outside of the computational domain. XCOJ1 is written for the special case of $J=1$. XCOJN is for the case of $J=N$. In the [WINER.GROIN]

director there are also the special subroutines XCJGR and XCJGR1 which are written for the case of a shore perpendicular barrier between the $J=JGR$ and $J=JGR+1$ grid columns. These two subroutines are for the two grid columns adjacent to the barrier. In the [WINER.CLOSED] directory there is a subroutine XCOJ33 which is in the GPRO program which simulates the lab experiment of Gourlay as reported in the dissertation. This subroutine is written since the $J=33$ grid column is next to a boundary for the first 30 grids. Also in the [WINER.CLOSED] directory there is the XCO2 subroutine written for $J=34, N$ since computations are only for the $I=31$ through $I=IWET(J)$ grids.

3.2 The YCOEF Subroutine

If equations (3.3) and (3.4) are written as

$$A_{2j-1}ETANEW(I,J-1) + B_{2j-1}VNEW(I,J) + C_{2j-1}ETANEW(I,J) = D_{2j-1} \quad (3.9)$$

and

$$A_{2j}VNEW(I,J) + B_{2j}ETANEW(I,J) + C_{2j}VNEW(I,J+1) = D_{2j} \quad (3.10)$$

then

$$A(2*J-1) = -g * RY$$

$$B(2*J-1) = 1 + \Delta T * TBY(I, J) / (\rho \bar{D})$$

$$C(2*J-1) = g * RY$$

$$D(2*J-1) = V(I, J) - \Delta T * V_{i,j} \frac{V_{i,j+1} - V_{i,j-1}}{2\Delta Y} - \frac{\Delta T}{4} (U_{i,j} + U_{i+1,j} + U_{i,j-1} + U_{i+1,j-1}) \frac{V_{i+1,j} - V_{i-1,j}}{2\Delta X} - \frac{\Delta T}{\rho \bar{D}} \left(\frac{SYY_{i,j} - SYY_{i,j-1}}{\Delta Y} + .5 \frac{SXY_{i+1,j} - SXY_{i-1,j} + SXY_{i+1,j-1} - SXY_{i-1,j-1}}{2\Delta X} \right) + \Delta T * EX_{i,j} \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{\Delta X^2}$$

$$A(2*J) = -RY * \bar{D}_{i,j}$$

$$B(2*J) = 1.0$$

$$C(2*J) = RY * \bar{D}_{i,j+1}$$

$$D(2*J) = ETANEW_{i,j}^* + RY * (V_{i,j+1} \bar{D}_{i,j+1} - V_{i,j} \bar{D}_{i,j}) \quad (3.11)$$

The above is the essence of the YCOEF subroutine. For $J=1$ and for $J=N+1$ the $D(2*J-1)$ term is coded so as to incorporate the lateral boundary condition of no y gradients of the circulation terms at the boundary. This means that the y derivative terms are not included for $D(1)$ and $D(2*N+1)$.

As with the X-series of subroutines so with the Y-series of subroutines. Any subroutine with a Y as the first letter is a variation of YCOEF. The YCVAR subroutine, which is named for the variable shoreline position is the same as the YCOEF subroutine except that it has many IF statements to determine the position of the shoreline (computational domain boundary) in relation to the present grid. For each grid, if the boundary is adjacent to the grid in the x direction then the x derivative terms for the radiation stress SXY , the mixing and the advective acceleration are all written using a three point derivative that incorporates the zero longshore velocity ($V=0$) and zero wave height ($SXY=0$) at the shore grid edge.

In the [WINER.BARR] directory YCOJJJ and YCJJJ1 are written for the grid row on the offshore and onshore sides respectively of a barrier that is on the shore edge of the $I=JJJ$ grid row.

In the [WINER.CLOSED] directory which is written for a closed basin the $V=0$ lateral boundary condition is incorporated by having $D(1)=0$ and $D(2*N+1)=0$. YCOI1 is used for the $I=1$ grid row and incorporates the $U(1,J)=0$ offshore boundary condition (which is used for a closed lab basin). YCI31 is written for the GPRO program to incorporate the boundary of the basin from $J=34$ through $J=N$.

3.3 The CIRC Subroutine

The CIRC subroutine varies in each of the directories. The variation is but to include the calls to the necessary special subroutines when appropriate. 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 subroutines sequentially calls

the Y-series of subroutines making a call to TRIDA to solve the tridiagonal matrix equation and the solution vector VV is assigned to the appropriate variables.

It should be noted that the sequential calls to the X-series and then to the Y-series subroutines is done only because of the linear nature of the VAX computer. The calls to the XCOEF could be vectorized if a computer with parallel processors is used. Likewise the calls to the Y-series could be vectorized. However the X-series of calls must be completed before any calls to the Y-series is made since the Y-series uses ETANEW* which is determined in the x sweep.

3.4 The FLOOD subroutine

The FLOOD subroutine is called directly from the main program after the call to UP. This discussion of the FLOOD subroutine is included in the chapter on the circulation model since the FLOOD subroutine controls the shoreline boundary of the circulation model.

The FLOOD subroutine is quite simple in concept but a little cumbersome in its code. The subroutine goes grid column by grid column from J=1 through J=N to determine whether the shore boundary should be moved. First it determines whether the last grid in the grid column has gone dry. This is simply whether the total depth TD(IWET(J),J) is less than zero. If it is IWET(J), the number of grids in the J grid column, is reduced by one. Further the amount of the water deficiency in the IWET(J+1),J grid (which used to be the IWET(J),J grid) is taken from the IWET(J),J grid and added to the IWET(J+1),J grid so that the total depth in the IWET(J+1),J grid is zero. This is done so as to maintain conservation of water mass in the computational domain. The above requires corrections to both the ETA and TD values.

If the total depth in the IWET(J),J grid is not less than zero the subroutine checks to see if the water level in the IWET(J),J grid is greater than the elevation of the neighboring IWET(J+1),J grid. This is a simple IF statement

```
IF((ETA(IWET(J),J)+H(IWET(J),J)).GT.0.0)THEN
```

which determines this since ETA and H are measured positive in opposite directions. If

the above IF statement is true then IWET(J) is incremented by one and an arbitrary one millimeter (.001 meters) of water is taken from the IWET(J-1),J grid and placed in the newly wetted IWET(J),J grid. Again this is done for both the ETA and TD values.

At present the flooding mechanism is only in the x direction. This requires a lot of if statements. To allow for flooding in the y direction would require many more IF statements and it is doubtful whether this would be worth while since with a parabolic wave model wave energy is essentially propagated in only one direction. If other wave models are to be included in the model, it may be worth implementing a two directional flooding mechanism.

CHAPTER 4 VARIATIONS OF THE PROGRAM

All of the above described subroutines and the executable program can be found in the [WINER.MODEL] directory (or else have been put on tape in which case see Sidney or Suburna for their location). The program in the [WINER.MODEL] directory is for cases with open lateral boundaries and no internal boundaries due to the presence of structures. Variations of the main program in [WINER.MODEL] have been written for the following cases: for closed (reflective) lateral boundaries, for shore parallel breakwaters and for shore perpendicular breakwaters. The programs for closed lateral boundaries are in the [WINER.CLOSED] directory. Those for a shore parallel breakwater are in the [WINER.BARR] directory and those for a shore perpendicular breakwater are in [WINER.GROIN]. This chapter describes the adaptations made in each of these directories. The last section of this chapter also describes some of the plotting programs found in the [WINER.PLOT] directory.

4.1 The [WINER.BARR] Directory

In this directory 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 structure of the program is essentially the same as described in the preceding chapters. Many of the subroutines are exactly the same. Those that are changed are described below. The main program is named BARR.FOR and 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. The linking command is

```
$LINK BARR/OPT <cr>
```

In the BARR.OPT file there is listed the new subroutines ACOH1, ACOH2, WVNMHA, YCOJJJ, and YCJJJ1. WVRAD is a modification of WAVRAD, CIRBARR is a modification of CIRC and RADBARR is a modification of RAD. With the exception of slight changes to XCOEF and INPUT and the main program, the other files are unchanged.

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)
```

SIGHA(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 WVRAD and RADBARR subroutines. REFLEC and REFL are not used in the program. They remain in the common statement as a remnant of an unsuccessful attempt to add reflection from the breakwater to the model.

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. These 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 rest of the INPUT subroutine is unchanged from that described previously. Within the [WINER.BARR] directory there is a sample run available. The input file IN.BARR and depth file PL.DEP is a test case of normal incident waves with a breakwater 100 meters offshore (JJJ=10, DX=10m for a plane beach with 20 wet grids in the x direction) and 100 meters in length (NBB=5, DY=10).

Once NBB is passed back to the main 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. From the main these are passed in the WVRAD and the CIRBARR subroutines.

The WVRAD subroutine is an adaptation of the WAVRAD subroutine described above with the modification to include the barrier. This subroutine makes calls to WAVNUM which is unchanged and WVNMHA which determines the wave characteristics (k, C_g, σ, \bar{k} , and $\bar{\theta}$) at the position of the edge of the grid row where the barrier is located. The WVNMHA subroutine does this by taking the depth at the grid edge to be the average of the two adjacent grids. The WVNMHA subroutine has the same structure and logic as the WAVNUM subroutine. Next the WVRAD subroutine initializes the values of KBREAK(J) and AMP(1,J) the same as was done in the WAVRAD subroutine described previously. Next is a DO loop for I=1, JJJ-1 which calls AMPCO then CTRIDA and then assigns the solution vector to the AMP values on the next grid row. This gives the values of the AMP array up to and including the I=JJJ grid row.

Then a call is made to ACOH1 which has the same structure as AMPCO except that DX is set to one half DX. ACOH1 determines the coefficients of the tridiagonal matrix equation in terms of the known AMP 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, CGHA, SIGHA, and RKOHA values). Then a call is made to CTRIDA so that the solution vector is the complex amplitude at the grid edge half way between the I=JJJ and I=JJJ+1 grid centers. The solution vector VVV is then assigned to the AMPH 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 RADBARR 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 model to the I=JJJ+1 grid center.

Next the WVRAD subroutine calls the ACOH2 subroutine. The ACOH2 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 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=J+1 grid row. A call is then made to CTRIDA and the solution vector is assigned to the AMP values on the I=J+1 grid row. With the values of AMP known on the I=J+1 grid row the rest of the AMP array is determined by a do loop for I=J+1, IMAX-1 which calls AMPCO then CTRIDA and then assigns the solution vector to the AMP values on the next grid row.

The final execution in the WVRAD subroutine is a call to RADBARR which is the same as the RAD subroutine previously described except that there is the additional do loops for J=N1, N2 for both I=J and I=J+1. These do loops have the same structure as other computations in the RAD subroutines except that DADX and DACONDX are written as three point derivatives. For the I=J grid row the three point derivatives are written using the AMPSAV array and on the I=J+1 grid row the derivative are written using a zero value at the lee side of the breakwater.

The CIRBARR subroutine differs from the CIRC subroutine previously described only in that for the y sweep calls to YCOEF are made for J=2, J-1 and then a call is made to YCOJJ and then YCJJ1 which are adaptations of YCOEF for the I=J and I=J+1 grid rows respectively. These two subroutines are just like 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.

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 XCOEF. 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 $U(JJJ+1,J)=0$.

4.2 The [WINER.GROIN] Directory

This directory contains a modification of the program found in the [WINER.MODEL] directory. The modification is for the case of a shore perpendicular breakwater that is represented as an infinitesimally thin impermeable barrier located between the $J=JGR$ and $J=JGR+1$ grid columns. Most of the subroutines are the same as in the [WINER.MODEL] directory or with minor modifications. The main program is GROIN.FOR and the executable file is also GROIN. A list of all the subroutines is found in GROIN.OPT and the link command is

```
$LINK GROIN/OPT <cr>
```

The only new subroutines are AMPLL and AMPLLL which are modifications of AMPCO to include the internal boundary conditions at the shore perpendicular breakwater, and XCJGR and XCJGR1 which are modifications of the XCOEF subroutine to incorporate the boundary conditions adjacent to the breakwater.

The INPUT subroutine is modified to read the position and length of the groin (groin and breakwater and jetty are used interchangeably in this section) and then pass this info to the main 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 jetty. The input data file has two more variables in this directory than in the [WINER.MODEL] 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 array is determined by the definition

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

the LLL and JGR variables are then sent in the arguments of the CIRC and WAVRAD calls so that the internal boundary conditions can be met.

The WAVRAD subroutine has the values of LLL and JGR passed in the argument of the call statement. This subroutine has the same structure as the subroutine with the same name in the [WINER.MODEL] directory. The WAVNUM, AMPCO, CTRIDA, and WDIS subroutines are unchanged. The first do loop to call AMPCO in the WAVRAD subroutine is for $I=1, LLL-2$. This moves the solution for the AMP 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. AMPLL which is specifically for the $I=LLL-1$ call which 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 AMPLLL which is called sequentially for $I=LLL, IMAX-1$ to complete the solution for the AMP array.

The internal boundary condition is handled in the following manner. When the parabolic equation (equation 4.27 on page 53 of the dissertation) 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 the dissertation.

In the RAD subroutine the values of LLL and JGR are passed in the argument of the call. In RAD a special do loop is written for the two grid columns on each side of the groin to

insure that the y derivatives are not taken across the barrier. Thus for $I=LLL, IWET(JGR)$ on the $J=JGR$ grid column the y derivatives of AMP (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 on the other side of the barrier in terms of the value of AMP on the proper side of the barrier.

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 XCJGR and XCJGR1 subroutines are adaptations of XCOEF so as to avoid taking derivatives across the impermeable barrier. The YCOEF and YCVAR subroutines are 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 in the previous section.

4.3 The [WINER.CLOSED] Directory

In this directory there are two executable programs PRO and GPRO. This directory has adaptations of the program for laboratory basin studies where there lateral boundary conditions are zero velocity across the boundaries and also a similar condition on the offshore boundary. The discussion here will be brief since the reader by now should understand how adaptations are made to the program.

For the PRO program the changes are minimal. the AMPCO subroutine is changed to have a reflective condition at $J=1$ and $J=N$. This is easily accomplished by setting BVAL

equal to one. The YCOEF and YCVAR subroutines are similarly changed to incorporate the $V=0$ condition for $J=1$ and $J=N+1$. The CIRC subroutine also has a call to a new subroutine YCOI1 which is for the y sweep for the special case of $I=1$ which must be computed since the ETA values on the offshore grid row are not constrained to being zero.

The GPRO program is a special adaptation used to simulate the experiments of Gourlay as described in section 5.6 of the dissertation. By looking at figure 5.30 on page 98 of the dissertation it is seen that the computational domain is somewhat complicated. The CIRC subroutine thus has special calls to XCOJ33 for the special case of $J=33$, to XCO2 which is an adaptation of XCOEF for the cases of $J=34, N$ since the computation begins at $I=31$, and to XCOJN2 which differs from XCOJN in that the computations begin at $I=31$. Also there is a call to YCOI31 for the special case of $I=31$ which includes boundary conditions. For the XC series of subroutines the indices of the coefficients is shifted since the computations begin with $U(1, J)$ rather than $ETA(1, J)$ as in the other directories.

IN.PL is a data file and PL.DEP is a depth file for a case of a plane beach at angle to the wave maker in a rectangular basin. GOUR.IN and GOUR.DEP are the input and depth files for the Gourlay experiment.

4.4 The [WINER.PLOT] Directory

In this directory there are several files which use the NCAR and QTEK plotting routines. If the reader is familiar with these packages there should be no problem using the plotting programs in this directory. The NCAR programs are only as fortran files. This is because in the NCAR package the dimension statement must be the exact same as the dimension of the array that is sent in the argument of the call to the NCAR subroutine. This is remedied by having a parameter statement as the first statement of the NCAR fortran files. These files then must be edited to change the parameters M and N to the appropriate size, compiled and then linked. The linking command is simply

```
$LINK filnam <cr>
```

if a

\$@[KIRBY]NCAR <cr>

command has already been executed. The user could put such a command in his LOGIN.COM file in the main directory so as to automatically execute it. This command makes the NCAR library a part of the system library as far as linking command are concerned.

The NCAR fortran files in the [WINER.PLOT] directory are ETA, SURFACE, VEL, STREAM, and QVEC. Each of these programs reads an output file in the same format as it was written by the WRITE subroutine. The ETA program reads only the ETA array and makes a call to the NCAR subroutine EZCNTR to produce a contour plot of the surface elevation. The ETA program can also be used to read the AMP.LST and ETA.LST files to get contours of the wave height and of the instantaneous water surface due to the wave field. The SURFACE file makes a call to SRFACE. This file was obtained from Rengi Phillip so I cannot describe the logic. DX and DY should be changed as needed along with the M and N in the parameter statement. Otherwise the program prompts for the data file (which must be entered with a single quote on each side) and the eye position of the viewer, etc. The VEL program reads the output file to get the ETA U and V arrays, averages the U and V from two opposite grid edges to get grid centered velocity arrays (called UC and VC) and calls the NCAR subroutine EZVEC to a plot of the velocity vectors. The QVEC program is a minor change to the VEL program. The QVEC program also reads a depth file to get a depth array and then creates the total depth array by adding the ETA array to the depth array. Then the UC and VC arrays are multiplied by the total depth array so as to create arrays of the x and y components of the total discharge. A call to EZVEC then gives a plot of the vectors of the total discharge.

APPENDIX A
VARIABLES USED IN THE COMPUTER CODE

A is a column vector of coefficients for the tridiagonal matrix equation resulting from the governing circulation equations. It is dimensioned (200).

A2 in the AMPCO subroutine is $\left(\frac{v}{(C_s \cos \theta + U)}\right)_j^{i+1}$

AA is a column vector of coefficients for the tridiagonal matrix equation resulting from the parabolic equation. It is dimensioned (100).

AB in the AMPOUT subroutine is the absolute value of the amplitude.

AII is an imaginary number, the square root of minus one

AMP is the complex amplitude. It is dimensioned (100,100).

AMPIJ is used in the AMPCO subroutine and is the value of $A_{i,j}$ which is passed in the argument of the WDIS call.

AMPO is the complex amplitude on the first grid row. It is dimensioned (100)

ANG is used in the AMPCO subroutines and is the m or longshore component of the wavenumber.

ATEMP is a complex variable used in the AMPCO series subroutines. It is actually the value of the complex amplitude as computed by an explicit finite differencing of the parabolic wave equation. The value of ATEMP is then passed to the WDIS subroutine in order to compute the amount of wave energy dissipation to be used. It is computed using equation (4.29) on page 54 of the dissertation.

ATMP is used in the AMPCO subroutines and is $im \frac{\Delta Y}{2}$

B is a column vector of coefficients for the tridiagonal matrix equation resulting from the governing circulation equations. It is dimensioned (200).

B2 in the AMPCO subroutine is $\frac{\left(\frac{C_g \cos \theta + U}{\sigma}\right)_j^{i+1} - \left(\frac{C_g \cos \theta + U}{\sigma}\right)_j^i}{\left(\frac{C_g \cos \theta + U}{\sigma}\right)_j^{i+1} + \left(\frac{C_g \cos \theta + U}{\sigma}\right)_j^i}$

BB is a column vector of coefficients for the tridiagonal matrix equation resulting from the parabolic equation. It is dimensioned (100).

BETA is a variable that is used both the TRIDA and the CTRIDA subroutines. It is complex in CTRIDA and not complex in TRIDA. It is not passed by any common statements and is used only in the subroutines for the double sweep solution. In CTRIDA it must be dimensioned at least as large as N. In TRIDA it must be at least 2N+1 or 2M+1, which ever is larger.

BVAL is used at the lateral boundaries in the AMPCO subroutines. It is the coefficient of equation (4.35) or (4.36) on page 55 of the dissertation.

C is a column vector of coefficients for the tridiagonal matrix equation resulting from the governing circulation equations. It is dimensioned (200).

C2 in the AMPCO subroutine is $\left(\frac{\sigma}{C_g \cos \theta + U}\right)_j^{i+1} \frac{\left(\frac{V}{\sigma}\right)_{j+1}^{i+1} - \left(\frac{V}{\sigma}\right)_{j-1}^{i+1}}{4DY}$

CC is a column vector of coefficients for the tridiagonal matrix equation resulting from the parabolic equation. It is dimensioned (100).

CG is the group velocity and is dimensioned (100,100). The values of the CG array are determined in the WAVNUM subroutine using the relationship $CG = n$ times the celerity, where

$$n = \frac{1}{.2} \left(1 + \frac{2RK TD2}{\sinh(2RK TD2)} \right) \quad (A.1)$$

CGPU in the AMPCO routine is $(C_g + U)_{i,j}$.

CGP1U in the AMPCO routine is $(C_g + U)_{i+1,j}$

CNN is a constant used in the EXCOEF subroutine it is the N of equation (2.24) on page 23 of the dissertation. It has arbitrarily been set to .08.

CS in the AMPCO subroutine is $(\bar{k}^i - k_j^i) \cos \theta^i$

CSP1 in the AMPCO subroutine is $(\bar{k}^{i+1} - k_j^{i+1}) \cos \theta^{i+1}$

CSQ in the AMPCO subroutine is $\left(\frac{k C_g (1 - \cos^2 \theta)}{2(C_g \cos \theta + U)}\right)_j^i$

CSQP1 in the AMPCO subroutine is $\left(\frac{k C_g (1 - \cos^2 \theta)}{2(C_g \cos \theta + U)}\right)^{i+1}_j$

CXP in the AMPOUT subroutine is $e^i \int k dx$

D is a column vector of the right hand side for the tridiagonal matrix equation resulting from the governing circulation equations. It is dimensioned (200).

D2 in the AMPCO subroutine is $\left(\frac{\sigma}{C_g \cos \theta + U}\right)^i_j \frac{\left(\frac{v}{\sigma}\right)^i_{j+1} - \left(\frac{v}{\sigma}\right)^i_{j-1}}{4DY}$

DACONDY in the RAD subroutine is the x derivative of the complex conjugate of $Ae^i \int k dx$

DACONDY in the RAD subroutine is the y derivative of the complex conjugate of $Ae^i \int k dx$

DADX in the RAD subroutine is the x derivative of $Ae^i \int k dx$

DADY in the RAD subroutine is the y derivative of $Ae^i \int k dx$

DD is a column vector of right hand side for the tridiagonal matrix equation resulting from the parabolic equation. It is dimensioned (100).

DT is the time step increment used in the circulation model.

DX is the grid spacing in the x direction.

DY is the grid spacing in the y direction.

EMAX is the maximum value that the eddy viscosity coefficients may be. EMAX is used so as to place a cap on the eddy viscosity coefficients and thus avoid numerical stability problems that may arise from too large a coefficient. This is important since the lateral mixing terms are represented explicitly in the model and thus place a constraint on the size of either the delta t increment or the lateral mixing coefficient.

EPS and EPSS are arbitrary ϵ used in the DISP, CHECK, and WAVNUM subroutines.

ETA is the mean surface elevation measured upwards from the still water line. It is dimensioned (100,100).

ETABAR in the AMPOUT subroutine is the value of the instantaneous water surface due to the waves obtained from equation (3.72) on page 42 of the dissertation.

ETANEW is the (100,100) array of the newly calculated mean water surface elevation.

ETADT is the time rate of change of ETA. It is dimensioned (100,100), but only values for the first two offshore grid rows are used. The values of ETADT are determined in the UP subroutine prior to reassigning the values of ETA. The values of ETADT are used in the XCOEF series of subroutines to determine the non-linear $\frac{\partial U}{\partial x}$ term. This is necessary since U(1,J) is not calculated and thus a centered x derivative at the U(2,J) position is not directly obtainable. Instead the continuity equation is solved for $\frac{\partial U}{\partial x}$.

$$\frac{\partial U}{\partial x} = -\frac{1}{TD} \left(ETADT + U \frac{\partial TD}{\partial x} + \frac{\partial TD V}{\partial y} \right) \quad (A.2)$$

EX is an eddy viscosity coefficient that is dimensioned (100,100). Its values are calculated in the subroutine EXCOEF according to a Longuet-Higgins formulation given by equation (2.24) on page 23 of the dissertation.

EY is an eddy viscosity coefficient which is equal to EMAX.

FILNAM is used in the INPUT subroutine. It is declared as LOGICAL*1 FILNAM(30) and is used to enter the name of the input data file and the depth data file.

FRIC is a constant friction coefficient that is an input variable.

G is the gravitational acceleration constant.

GAMMA is a variable that is used both the TRIDA and the CTRIDA subroutines. It is complex in CTRIDA and not complex in TRIDA. It is not passed by any common statements and is used only in the subroutines for the double sweep solution. In CTRIDA it must be dimensioned at least as large as N. In TRIDA it must be at least 2N+1 or 2M+1, whichever ever is larger.

H is the water depth measured positive down from the still water line. It is dimensioned (100,100)

IMAX is the largest value in the IWET array. The computational domain of the wave model is IMAX by N. Also the output is in the form of IMAX by N arrays.

IMIN is the smallest value in the IWET array.

ITER is a count of the number of interactions.

ITERST is a constant that is used in the main to determine the smoothly increasing

values of the wave height during the start-up of the model. It is the same as C1 in equation (4.45) on page 60 of the dissertation except that it is expressed in units of iterations.

IWET, which is dimensioned (100), is the number of grids (for each grid column) from the offshore boundary to the shoreline. The values of the IWET array are changed by the FLOOD subroutine with the rise or fall of the mean water level.

KBREAK is an index which is used in the WDIS subroutine to indicate whether or not dissipation due to breaking should be applied. If KBREAK(J) is zero then the WDIS subroutine returns a zero dissipation coefficient. If KBREAK(J) is one, a positive dissipation coefficient is returned. The WDIS subroutine controls the value of KBREAK(J). If the wave height becomes greater than .78 times the depth KBREAK(J) is set to one and remains so until the wave height drops below a value of .4 times the local depth. The rationale for this is in Dally's dissertation cited in my dissertation.

KFLAG is an indicator that is controlled by the CHECK subroutine to indicate convergence. KFLAG is set to zero at the beginning of the CHECK subroutine and then any non-convergence will set KFLAG to 2 and return to the main.

M is the number of grids in the x direction. It should be noted that M must be sufficiently large to allow for flooding of the shoreline. The actual computations will take place over a number of grids somewhat less than M.

N is the number of grids in the y direction.

OMEGA is the wave frequency = $2*PI/T$ where PI is $\pi=3.14159$

PAA in the AMPCO subroutine is used as part of the differencing of the $(CC_G \frac{A}{\sigma})_y$ term. It is
$$\frac{\left(\frac{\sigma C_g}{k}\right)_{j+1}^{i+1} + \left(\frac{\sigma C_g}{k}\right)_j^{i+1}}{2 DY^2 (C_g \cos \theta + U)_{j+1}^{i+1}}$$

PBB in the AMPCO subroutine is used as part of the differencing of the $(CC_G \frac{A}{\sigma})_y$ term. It is
$$\frac{\left(\frac{\sigma C_g}{k}\right)_j^{i+1} + \left(\frac{\sigma C_g}{k}\right)_{j-1}^{i+1}}{2 DY^2 (C_g \cos \theta + U)_j^{i+1}}$$

PCC in the AMPCO subroutine is used as part of the differencing of the $(CC_G \frac{A}{\sigma})_y$ term. It is
$$\frac{\left(\frac{\sigma C_g}{k}\right)_{j+1}^i + \left(\frac{\sigma C_g}{k}\right)_j^i}{2 DY^2 (C_g \cos \theta + U)_j^i}$$

PDD in the AMPCO subroutine is used as part of the differencing of the $(CC_G \frac{\Delta}{\sigma})_y$ term. It is $\frac{\left(\frac{\sigma C_g}{k}\right)_j + \left(\frac{\sigma C_g}{k}\right)_{j-1}}{2 DY^2 (C_g \cos \theta + U)_j^2}$

RK is the wavenumber array. It is dimensioned (100,100) and is determined in the WAVNUM subroutine using a Newton method solution to equation (3.76) on page 44 of the dissertation.

$$\omega = \sigma + k \cos \theta U \quad (\text{A.3})$$

where σ is the relative frequency given by

$$\sigma^2 = gk \tanh kh \quad (\text{A.4})$$

RKO is the average of the wave number across a grid row. It is dimensioned (100).

RKN is the wave number along the first grid row. It is used in the INPUT subroutine.

RO is the fluid density. It has arbitrarily been set to 1000 kilograms per cubic meter. The value of the fluid density is not important since in the places it is used it is cancelled out by use in both the denominator and the numerator.

RX is used in all of the coefficient subroutines that are called from CIRC. It is =DT/DX

RY is used in all of the coefficient subroutines that are called from CIRC. It is =DT/DY

SIG is the relative wave frequency and is dimensioned (100,100). It is determined in the WAVNUM subroutine using a Newton method solution to the dispersion relationship in the absence of currents.

SUM in the AMPOUT subroutine is $\int \bar{k} dx$

SXX is the S_{xx} radiation stress term. It is dimensioned (100,100) and is determined in the RAD subroutine.

SXY is the S_{xy} radiation stress term. It is dimensioned (100,100) and is determined in the RAD subroutine.

SYY is the S_{yy} radiation stress term. It is dimensioned (100,100) and is determined in the RAD subroutine.

T is the wave period which is an input variable.

TBX is the bottom shear stress in the x direction. It has a dimension of (100,100). It is named for Tau Bottom in the X direction.

TBY is the bottom shear stress in the y direction. It has a dimension of (100,100). It is named for Tau Bottom in the Y direction.

TD is the total depth of the water column. It is dimensioned (1000,100) and is equal to H plus ETA. The values of the TD array are updated each iteration in the UP subroutine.

TD2 is also the total depth of the water column and is dimensioned (100,100). It is equal to TD except for the grids where TD is equal to or less than zero (TD should never be less than zero, but just in case by error it is, the .LE. if statement is used). If TD is zero or negative TD2 is set to an arbitrarily small value of 1 millimeter. The TD2 array is updated in the WAVNUM subroutine and is used in all of the wave model subroutines.

THETA is the wave angle at each grid row assuming a plane beach without currents as calculated by Snell's law. It is dimensioned (100).

THETAH is the wave angle calculated by Snell's law for the position half way between grid centers (i.e. at the grid edge) at the location of the breakwater. THETAH is used only in the BREAK program and in the GPRO program.

THETAO is the input value of the wave angle at the offshore grid row location.

TIME is the time in minutes for the start-up of the model. It is actually the C1 constant of equation (4.45) on page 60 of the dissertation.

U is the (100,100) array of x component of the depth integrated mean current.

UIJ is a grid centered $U_{i,j}$ used in the AMPCO subroutine.

UIP1J is a grid centered $U_{i+1,j}$ used in the AMPCO subroutine.

UNEW is the (100,100) array of the newly calculated values of the x component of the depth integrated mean current.

UU is used in the main program and is actually the $C(t)$ of equation (4.45) on page 60 of the dissertation. It is used to gradually increase the wave height so as to avoid a shock start- up.

V is the (100,100) array of y component of the depth integrated mean current.

VIJ is a grid centered $V_{i,j}$ used in the AMPCO subroutine.

VIJM1 is a grid centered $V_{i,j-1}$ used in the AMPCO subroutine.

VIJP1 is a grid centered $V_{i,j+1}$ used in the AMPCO subroutine.

VIP1J is a grid centered $V_{i+1,j}$ used in the AMPCO subroutine.

VIP1JM1 is a grid centered $V_{i+1,j-1}$ used in the AMPCO subroutine.

VIP1JP1 is a grid centered $V_{i+1,j+1}$ used in the AMPCO subroutine.

VNEW is the (100,100) array of the newly calculated values of the y component of the depth integrated mean current.

VV is the solution vector for the circulation model. It is determined by the TRIDA subroutine and in the CIRC subroutine (which called TRIDA) the VV vector is assigned to the circulation values of UNEW or VNEW and ETANEW. It should be noted that VV has a dimension that is twice that of the variables U, V, and ETA. This is because the tridiagonal matrix equation, which is solving for a staggered grid system, solves for both U and ETA in the x direction and for V and ETA in the y direction.

WAVAMP is the amplitude of the incident wave on the offshore grid row. It is equal to one half of WAVGHT which is the input wave height. WAVAMP is set in INPUT and passed through the call statement to the main program.

WAVGHT is the input wave height.

WAVEHGT, dimensioned (100,100), is the wave height which is actually twice the absolute value of the complex amplitude AMP. The WAVEHGT array is used in the SHEAR2 subroutine to determine the bottom shear stresses which are designated by the array names TBX and TBY.

WVAMP is a variable that it is passed in the argument of the WAVRAD subroutine. It is some value from 0.0-1.0 times WAVAMP. The value 0.0-1.0 is actually $C(t)$ as determined by equation (4.45) on page 60 of the dissertation. WVAMP times AMPO(J) is used to initialize AMP on the first grid row to start the row marching of the parabolic wave scheme.