NOAA Technical Report NOS CS 5

MECCA2 PROGRAM DOCUMENTATION

Silver Spring, Maryland
January 2000

noaa National Oceanic and Atmospheric Administration

U.S. DEPARTMENT OF COMMERCE
National Ocean Service
Coast Survey Development Laboratory

Office of Coast Survey
National Ocean Service
National Oceanic and Atmospheric Administration
U.S. Department of Commerce

The Office of Coast Survey (CS) is the Nation's only official chartmaker. As the oldest United States scientific organization, dating from 1807, this office has a long history. Today it promotes safe navigation by managing the National Oceanic and Atmospheric Administration's (NOAA) nautical chart and oceanographic data collection and information programs.

There are four components of CS:

The Coast Survey Development Laboratory develops new and efficient techniques to accomplish Coast Survey missions and to produce new and improved products and services for the maritime community and other coastal users.

The Marine Chart Division collects marine navigational data to construct and maintain nautical charts, Coast Pilots, and related marine products for the United States.

The Hydrographic Surveys Division directs programs for ship and shore-based hydrographic survey units and conducts general hydrographic survey operations.

The Navigation Services Division is the focal point for Coast Survey customer service activities, concentrating predominantly on charting issues, fast-response hydrographic surveys and Coast Pilot updates.

NOAA Technical Report NOS CS 5

MECCA2 PROGRAM DOCUMENTATION

Kurt Hess

January 2000

noaa National Oceanic and Atmospheric Administration

U.S. DEPARTMENT
OF COMMERCE
William Daly, Secretary

National Oceanic and
Atmospheric Administration
D. James Baker, Under Secretary

National Ocean Service
Nancy Foster
Assistant Administrator

Office of Coast Survey
Captain David MacFarland

Coast Survey Development
Laboratory
Bruce Parker

NOTICE

TABLE OF CONTENTS

LIST OF FIGURES iv

1. INTRODUCTION 1

2. MODEL ATTRIBUTES 3
   Numerical Code 3
   Model Variables 3
   Model Grid 4
   Cell Attribute Codes 5
   Atmospheric Heat Flux 7
   Input and Output Files 7

3. IMPROVEMENTS 9
   Corrections to Previous Version 9
   Input Data Files 9
   Ocean and River Boundaries 12
   Vertical and Horizontal Diffusivity 13
   Shallow Water Wind Stress Reduction 14
   Internal-Mode Velocity Boundary Conditions 14
   Density Function and Gradients 16
   Output for Graphing 16
   Non-linear Horizontal Advection 16

4. ACKNOWLEDGMENTS 18

5. REFERENCES 19

APPENDIX A. ERRATA SHEET 21

APPENDIX B. SAMPLE GEOGRAPHY FILE 23

APPENDIX C. SAMPLE CONTROL FILE 27

APPENDIX D. CODE LISTING 29

LIST OF FIGURES

Figure 2.1. Plan View and Isometric View of grid cells showing placement of variables. 4

Figure 2.2. Chesapeake Bay model grid showing water cells. 6

Figure 3.1. Wind vectors over the Chesapeake Bay grid 11

Figure 3.2. Instantaneous surface currents around Rattray Island, Australia.. 17

1. INTRODUCTION

This report describes the latest version of the MECCA (Model for Estuarine and Coastal Circulation Assessment) code that was originally published by Hess (1989) and hereafter referred to as M1. The new version of the code was developed in response to demands for a hindcast model for Chesapeake Bay (Bosley, 1996; Bosley and Hess, 1998), and to a lesser extent, to provide a version which corrected some of the errors in the first release. The original code is still operable, provided a few corrections to the code are made as described in the errata sheet (Appendix A).

MECCA was originally developed to meet several shortcomings not found in existing models: (1) extensive documentation including step-by-step development of the applicable equations and their conversion into finite difference form; (2) creation of a standardized form for the input that requires little explanation, can be adapted to most regional applications, and minimizes recoding; (3) provision of a model with internal switches to selectively eliminate various terms in the equations for sensitivity purposes; and (4) the inclusion of imbedded, one-dimensional flow to better represent river and channel flows. This philosophy has not changed.

With the new version comes additional philosophic goals: (1) to provide a basic, streamlined (less code) version which requires the user to do more outside coding and (2) reading data from similar, external files to provide time series values for boundary conditions. For example, the new version reads all data files two records at a time; fewer values are stored at any time, but with linear interpolation the user needs to add more points in time to create a smooth curve - there is no cubic interpolation (which requires four time points). Other improvements include variable array sizes, corrections to the original code, and several other changes (see Section 3).

The model has been requested, distributed, and used numerous times. The author has completed several applications including the estimation of Chesapeake Bay's natural period (Hess, 1988a), sediment transport (Hess, 1988b), and crab larval drift (Johnson and Hess, 1990). It has been used for the Gulf of Maine (Brooks, 1992; Brooks and Churchill, 1992; Brooks, 1994) and in Australian coastal waters (Galloway et al., 1996 - but see Section 4; King et al., 1998). It has been applied to coastal flows in France (Smaoui, 1996; Berthet, 1996).

This report is intended to describe the new features with a short review of the previous version (see M1 for details). The report covers a brief overview of the model in Section 2, a description of new features in Section 3, and an overview of applications to Chesapeake Bay and Rattray Island, Australia, in Section 4. Appendix A contains an errata sheet for the original version of MECCA. Appendix B has a new sample Geography File and Appendix C has a new sample Control File. Appendix D contains a listing of the MECCA2 code.

2. MODEL ATTRIBUTES

The following is a brief discussion of MECCA attributes as discussed in M1 which serves as a refresher in preparation to the description of the modifications in Section 3.

Numerical Code

The MECCA code solves the hydrodynamic equations of momentum, mass, salinity, and temperature conservation. It is three-dimensional in space, uses a vertical sigma coordinate, has a time-varying free surface, and incorporates non-linear horizontal momentum advection. It includes a three-dimension time variable horizontal diffusion based on Smagorinsky (see Tag et al., 1979) and includes vertical turbulent diffusion based on a mixing length and Richardson number-dependent reduction (Munk and Anderson, 1948). For the horizontal momentum equations, the external gravity wave mode is split apart from the internal mode.

Variables are placed on an Arakawa C-grid with square cells in the horizontal and at uniform intervals along a sigma-stretched vertical coordinate. External-mode momentum is solved with an alternating-direction, semi-implicit method in the horizontal. The salinities, temperatures, and internal-mode velocities are solved with a semi-implicit method in the vertical.

The sigma vertical coordinate is defined here as

$$q = \frac{z - \eta}{d - \eta} \tag{2.1}$$

where η is water level departure from the reference surface (z = 0) and d is the depth relative to the reference surface. In recent years, some modelers have encountered certain problems with sigma coordinate systems (Haney, 1991). These problems arose from accurately representing the horizontal pressure gradient due to density, and can be overcome by using uniformly-spaced sigma levels and by subtracting the spatially-averaged density before computing the horizontal gradient. MECCA has both these features.

The model is coded in Fortran with a modest amount of vectorization. Constants are read in from a Control File, and basin attributes are stored in a Geography File. Output is saved at the end of the run to provide a restart capability.

Model Variables

Two-dimensional variables include mean sea level depth (D), water level (SE, SEP, SEPP), vertically-integrated velocities (UH, UHP, VH, VHP), bottom stress (TBX, TBY), wind (WX, WY), surface stress (TSX, TSY), vertically-integrated horizontal turbulent viscosity (AH), and velocity departure functions (THETA1, THETA2, THETA3, THETSU, THETSV), cell status (IFIELD), time-integrated variables (SOLD, UHOLD, VHOLD), channel width (BX, BY), flow indices (MFLUX, NFLUX), edge parameter (FEDGE), relative cell area (AREA), vertically-averaged horizontal diffusivity (AH and AHC), and imbedded channel widths (BX, BY).

Three-dimensional variables include internal mode velocities (U, V, W), vertical viscosity (AV), vertical diffusivity (DV), salinity (S), temperature (T), Richardson Number (R), and horizontal viscosity (AH3). All units are metric unless otherwise stated.

Model Grid

Variables in the numerical grid are indexed by M in the x direction, N in the y direction and L in the -q direction. Placement of variables in grid cells are shown in Figure 2.1.

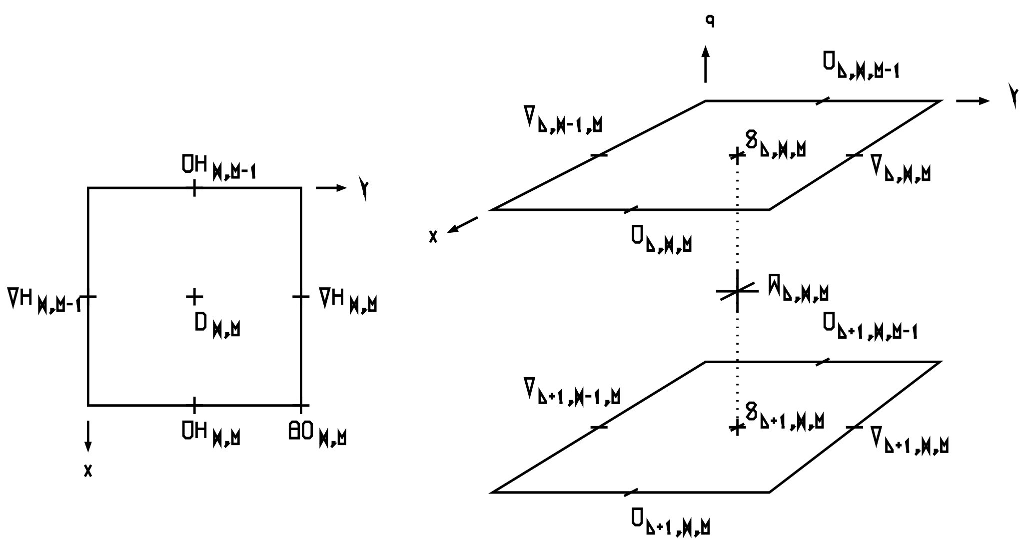
The positions of cell boundaries in the grid's horizontal plane determined by

$$x = N\Delta \text{ and } y = M\Delta, \tag{2.2}$$

where Δ is the grid size in meters. Position in the vertical is determined by

$$z = (1-L)\delta, \tag{2.3}$$

where $\delta = 1/(LBOT - 1)$ and LBOT is the number of the level that correspond to the bottom.



(a) Plan View (b) Isometric View

Figure 2.1. Plan View (a) and Isometric View (b) of grid cells showing placement of variables. The Plan View shows the two-dimensional variables. At D there are also SE, AH, FEDGE, AREA, PHI, WX, WY, IFIELD, FX, FY, TSX, TSY, and SOLD. At UH there are also U, UHOLD, UE, MFLUX, TBX, BX, GSTARX, THETA1, THETSU. At VH there are also V, VHOLD, VE, NFLUX, TBY, BY, GSTARY, THETA2, and THETSV. At AC (which is AHC) there is also THETA3. The Isometric View (b) shows the location of three-dimensional variables. At U there is also GRX and at V there is also GRY. At S there is also T and AH3. At W there is also AV and DV.

The grid is oriented to the surface of the earth by the following variables in the Geography File (Appendix B): BSNANG, BSNLAT, BSNLON, NCOR, and MCOR. The lower corner (i.e., closest to the origin) of cell (n = NCOR, m = MCOR) is at latitude BSNLAT and longitude BSNLON. The lower corner corresponds to location x = (NCOR - 1)Δ and y = (MCOR - 1)Δ, where Δ is the grid size. The y axis is oriented at an angle BSNANG clockwise from due east. Suppose μ_x is the Mercator transform from degrees longitude to X and μ_y is the Mercator transform from degrees latitude to Y. Then the Mercator coordinates of any cell's lower corner are

$$X = \mu_x\{BSNLON\} - (M-MCOR)\Delta' \sin\{BSNANG\} + (N-NCOR)\Delta' \cos\{BSNANG\}$$
$$Y = \mu_y\{BSNLAT\} - (M-MCOR)\Delta' \cos\{BSNANG\} - (N-NCOR)\Delta' \sin\{BSNANG\} \tag{2.4}$$

where

$$\Delta' = \mu_y\{BSNLAT + \frac{\Delta}{2}\} - \mu_y\{BSNLAT - \frac{\Delta}{2}\} \tag{2.5}$$

and Δ^o is the grid size converted to degrees of latitude

$$\Delta^o = \frac{\Delta}{1852 \times 60} \tag{2.6}$$

The grid for Chesapeake Bay is shown in Figure 4.2 (Bosley and Hess, 1998; Bosley, 1996). Simulations are being made on a grid with bathymetry that was previously developed at the U.S. Naval Academy (Hoff, 1990). The grid cell size (Δ) is 5,606 m and the model was run in barotropic mode.

Cell Attribute Codes

Each cell in the Geography File (Appendix B) is tagged with a two-digit number that defines certain attributes. The entire status of the cell is stored in the two-dimensional array IFIELD, where

$$IFIELD = 10(I) + J \tag{2.7}$$

For example, a cell may represent either land or water, and if water it may be an ocean or river boundary cell. It may also be either a full or half (triangular) cell, although boundary cells must be full. This is the cell's geographic status, and is coded in the single-digit integer I. For example,

I = 1 denotes a triangle
2 denotes a full water cell
5 denotes an ocean boundary cell
6 denotes a river boundary cell

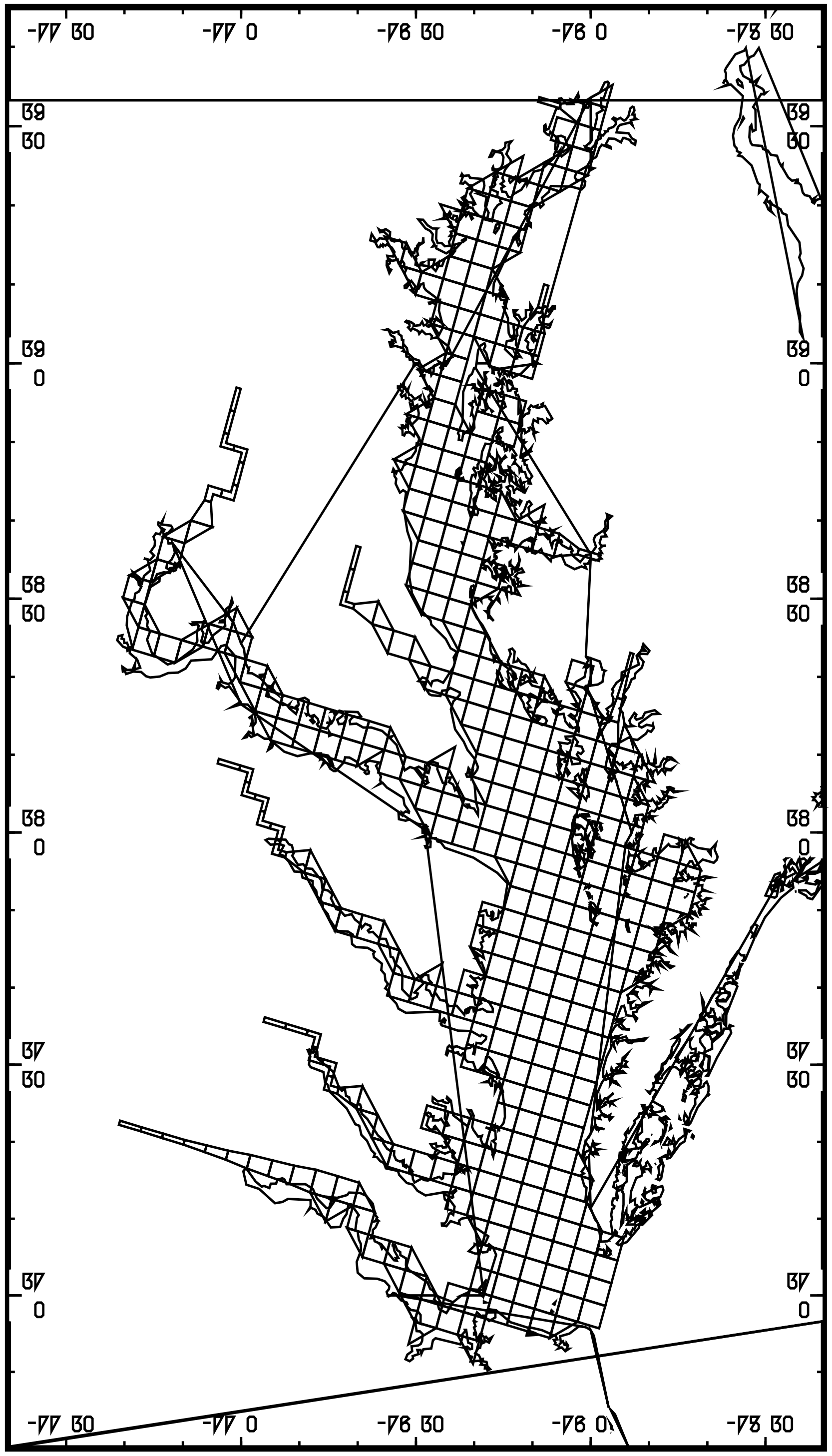


Figure 2.2. Chesapeake Bay model grid showing water cells. Cells are 5606 m on a side. Upper reaches of rivers are modeled as narrow channels.

A second cell feature is the potential existence of a physical barrier that prevents flow in either or both directions, and is coded in the single-digit integer J. For example,

J = 0 denotes no barriers
1 denotes a barrier in x-direction
2 denotes a barrier in y-direction
3 denotes barriers in both the x- and y-directions

The program uses IFIELD primarily to skip over land cells and to enforce zero flux at barriers.

NOTE: because MECCA sometimes makes calculations at cells near the edge of the grid, it is desirable to have extra rows and columns of non-water cells at the outer boundaries of the grid.

Atmospheric Heat Flux

The atmospheric heat flux formulation in M1 was tested with data for Tampa Bay (Hess, 1994) and found to be satisfactory. There are only a few minor changes to the code. Bottom heat flux has been removed since it had little effect on computed water temperatures.

Input and Output Files

MECCA is tailored for a specific application by reading data from a set of input files. Time step, printing, and other data, as well as the names of all other input files, are read in from the Control File. The grid, depths, orientation, and cell size are read from the Geography File. Environmental, or driving, data are read from additional files (see Section 3). If the run is restarted from a previous run, an Initialization File is read.

MECCA output is put into another set of files. These are the Screen File, which lists current timestep, the Print File, which contains top views and side views of various variables, and the Graphing File, which contains date and time series values at selected locations. A Save File is created which can be used for an Initialization File.

3. IMPROVEMENTS

Corrections to Previous Version

The previous version of the code (M1) is still very usable, provided a few corrections, especially one to the non-linear calculations in the internal-mode module, are made. A table showing the suggested changes appears in Appendix A.

Input Data Files

The boundary condition data are now read from separate input files, not from the Control File. In addition, to reduce array storage requirements, MECCA now stores only two records of data at a time; values needed in the program are based on a linear interpolation between the two values read in.

The following are, in the order they are read in the .CON file, the seven input data file types that can be read in MECCA:

Ocean Water Levels or Flowrates
Winds or Stresses and (Optional) Pressures Gradients
River Flowrates
Ocean Salinities
Ocean Temperatures
River Temperatures
Additional Meteorological Data

In the .CON file, for each of the above seven types, MECCA reads (1) a text file description (not used in computation), (2) the number of signals (NSIGS) in the file to be read, and (3) the file name. If NSIGS is 0, the file name is not read and therefore no data are read (all array values for that variable are zero).

For all files (except the wind/stress file), each record has a fixed format. A typical record contains the four-digit year, the day in year, and a number (=NSIGS) of values. The form is

YYYY DDD.DDDD V1 V2 V2 V3 V5

where YYYY is the 4-digit year, DDD.DDDD is a day-in-year date, and V1, V2, etc. are a set of values corresponding to that time given. Spaces or commas should separate all numbers. For example, a typical Additional Meteorological Data file is

1994	3.5000	12.04	0.50	0.32	1013.80
1994	3.6250	13.25	0.56	0.44	1014.10
1994	3.7500	12.72	0.63	0.30	1014.30
1994	3.8750	12.03	0.61	0.22	1014.20
1994	4.0000	10.86	0.59	0.29	1014.20

which contains the year, day, air temperature, relative humidity, cloud cover, and air pressure.

The time values are converted to a year-based time, YT, using the universal time coordinate (UT) and the number of days in the year, D. Note that noon on January 1 corresponds to UT = 1.5.

$$YT = (YEAR - 1900) + \frac{UT - 1}{D} \tag{3.1}$$

Wind data, unlike the other data types which are in ASCII, are stored in a binary file. These data require two unformatted records per time which contain:

YEAR, UTC, ITYPE1, ITYPE2, ITYPE3, NX, MX
FX, FY, (DPADX, DPADY)

Here YEAR and UTC are the date stamp; ITYPE1, ITYPE2, and ITYPE3 are indices (see below); and NX and MX are array sizes. The arrays FX(NX, MX) and FY(NX, MX) contain either wind speeds or wind stresses (per unit water density). DPADX and DPADY are atmospheric pressure gradients (mb/km). The indices are as follows: for FX and FY containing winds, ITYPE1 = 1; for FX and FY containing stresses, ITYPE1 = 2. For no atmospheric pressure gradient values to be read, ITYPE2 = 0; for values to be read, ITYPE2 = 1. For ITYPE3 = 0, model coordinate directions are used for winds, stresses, and pressure gradients. For ITYPE3 = 1, earth coordinate directions (east, north) are used.

For a wind with eastward and northward components WE and WN respectively, the components can be converted to model directions by

$$WX = -WN\cos\{BSNANG\} - WE\sin\{BSNANG\} \tag{3.2a}$$

$$WY = WE\cos\{BSNANG\} - WN\sin\{BSNANG\} \tag{3.2b}$$

The components can be converted back to earth coordinates by

$$WE = -WX\sin\{BSNANG\} + WY\cos\{BSNANG\} \tag{3.3a}$$

$$WN = -WX\cos\{BSNANG\} - WY\sin\{BSNANG\} \tag{3.3b}$$

A sample wind field for Chesapeake Bay is shown in Figure 3.1.

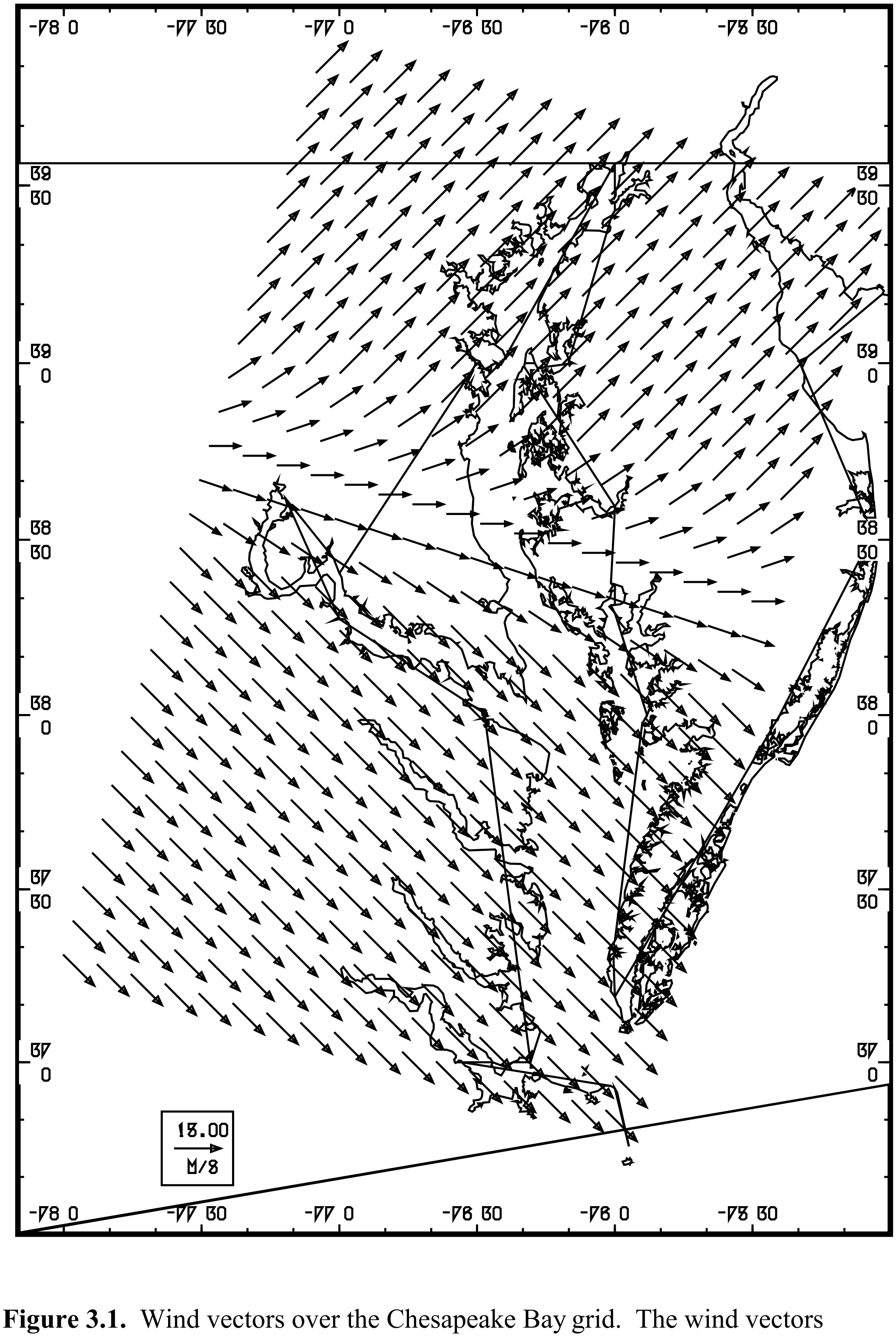


Figure 3.1. Wind vectors over the Chesapeake Bay grid. The wind vectors generated by applying the spatial interpolation scheme to a northeastward wind of 14 m/s at Thomas Point and a southeastward wind of 14 m/s at CBBT. Vectors at every other cell are plotted.

Ocean and River Boundaries

Ocean and river boundaries are defined in a new way. For an ocean boundary, as specified in the Geography File (Appendix B), the indices are

MB1, MB2, NB1, NB2, ITPO, JTPO, ISET1, ISET2

For an ocean boundary, the cells located at MB1 ≤ M ≤ MB2 and NB1 ≤ N ≤ NB2. The direction of outflow is set by ITPO, where

ITPO = 1 denotes flow in +x direction
-1 denotes flow in -x direction
2 denotes flow in +y direction
-2 denotes flow in -y direction

The type is set in JTPO, where

JTPO = 1 denotes a water level
2 denotes a flowrate per unit width.
3 denotes an Orlanski radiation condition
4 denotes a Riemann invariant condition

The Orlanski (1976) radiation condition is

$$\frac{\partial \eta}{\partial t} + c\frac{\partial \eta}{\partial x_n} = 0 \qquad (3.4)$$

and the Riemann invariant condition (Wurtele et al., 1971) is

$$\eta + \frac{H}{c}(u^2 + v^2)^{1/2} = 0 \qquad (3.5)$$

where c is the shallow water wave speed, H is total water depth, and x_n is the outward normal direction.

ISET1 and ISET2 are ocean signal numbers; water level at any cell is linearly interpolated in space between the present (time-interpolated) value in signal ISET1 at (NB1, MB1) and the value in signal ISET2 at (NB2, MB2).

For a river boundary, the indices are

MR1, MR2, NR1, NR2, ITPR, JTPR, ISETR

The direction of inflow, ITPR, is set analogously to ITPO, except that for example +1 means inflow in the +x direction (not outflow). The type is set in JTPR, where

JTPR = 1 denotes simple flowrate
2 denotes water falls

ISETR is river signal number, so that the first river so defined used signal ISETR in the river flowrate and temperature files. Also, there is a revised water falls condition. Unlike before (see M1, p. II-12 to II-15), the boundary water level increment and temperature are introduced directly into the river cell, not into the adjacent cell.

Ocean boundaries may have one or more of the following variables assigned as a boundary condition: water level, salinity, and temperature. River boundaries may have one or more of the following variables assigned as a boundary condition: flowrate, salinity, and temperature.

Vertical and Horizontal Diffusivity

Vertical mixing is handled in a more complex way and allows for negative Richardson Number (R) values. Negative values occur when the density is unstably stratified, and the new scheme causes large vertical diffusivities. The approach is based on that of Munk and Anderson (1948). As before, viscosity and diffusivity are related to a potential, A_z as follows:

$$A_z = \left(\kappa H q^{C_0}(1 + q)\right)^2 \left[\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2 \right]^{1/2} \tag{3.6}$$

$$A_v = A_{vo} + f_A A_z \tag{3.7}$$

$$D_v = D_{vo} + f_D A_z \tag{3.8}$$

where κ is von Karman's constant, A_{vo} and D_{vo} are small (molecular scale) values, and the functions f_A and f_D adjust for stratification.

Negative values of R, which imply denser water lying above lighter water, now create augmented mixing. For vertical momentum viscosity

$$\begin{aligned} f_A &= C_1(1 + C_2 R)^{C_3} \quad for \quad R \geq 0 \\ &= C_1(1 + C_4 R^2) \quad for \quad R < 0 \end{aligned} \tag{3.9}$$

And for vertical diffusivity,

$$\begin{aligned} f_D &= C_5(1 + C_6 R)^{C_7} \quad for \quad R \geq 0 \\ &= C_5(1 + C_8 R^2) \quad for \quad R < 0 \end{aligned} \tag{3.10}$$

Also, the values of R are bounded:

$$R_{\min} < R \leq R_{\max} \tag{3.11}$$

The 13 mixing variables (A_{vo}, D_{vo}, R_{min}, R_{max}, C_0 - C_8) are coded as (AV0, DV0, RIMIN, RIMAX, CR0, CRICH(1) - CRICH(8)) and have the nominal values (0.00001, 0.000005, -0.05, 1000., 1.0, 0.4, 10.0, 0.5, 20000., 0.1, 3.33, 1.5, 20000.). There is now no subroutine MIXUP, which (in M1) created a neutral density distribution.

Horizontal diffusivity has been improved by making it three dimensional. Horizontal diffusivity is the product of the input parameter DHDA and the viscosity, AH3.

Shallow Water Wind Stress Reduction

Wind stress is reduced in shallow water to avoid high velocities. In the code, wind stress is now multiplied by the factor f_τ, where

$$\begin{aligned} f_\tau &= 1 && d_2 \leq H \\ &= \frac{H - d_1}{d_2 - d_1} && d_1 < H \leq d_2 \\ &= 0 && H < d_1 \end{aligned} \tag{3.12}$$

and H is the total water depth. The values (d_1, d_2) are coded as (DTAU1, DTAU2) and have values (0.10, 1.0).

Internal-Mode Velocity Boundary Conditions

There is a new upper boundary condition on internal-mode velocity. At the upper surface, a one-sided approximation may be used by setting ITOPV=3. In that case, horizontal velocity is computed by setting the upper surface's vertical diffusion and advection to zero and reducing volume-related terms by half (as is done for the upper salinity and temperature boundary conditions: see M1, p I-77). In the new version of MECCA, the bottom stress per unit density is defined as the product of a friction factor and a representative velocity by the generalized form

$$\frac{\tau_b}{\rho} = \Phi[U + \gamma u_{LBOT} + (1 - \gamma) u_{LBOT-1}] \tag{3.13}$$

where Φ is the friction factor, U is the external-mode velocity, and u is the internal-mode velocity. For the general case,

$$\Phi = C_{DWB1} + C_{DWB2}|U + u_{LBOT}| \tag{3.14}$$

and $\gamma = 1$.

A new feature is the specification of a bottom logarithmic boundary layer with velocity

$$u(z) = \frac{1}{\kappa}\left(\frac{\tau_b}{\rho}\right)^{1/2} \ln\left(\frac{z}{z_o}\right) \tag{3.15}$$

The roughness height z_o is defined in the bottom stress subroutine as 0.003 m. At mid-depth in the bottom layer (which is at a distance of $H\delta/2$ above the bottom), we set the boundary layer velocity to be equal to the mean of the lowest two sigma level velocities. Squaring each side and rearranging gives

$$\frac{\tau_b}{\rho} = \left(\frac{\kappa}{\ln\{\frac{H\delta}{2z_o}\}}\right)^2 \left(U + \frac{1}{2}u_{LBOT} + \frac{1}{2}u_{LBOT-1}\right)^2 \tag{3.16}$$

Therefore, Φ is determined to be

$$\Phi = \left(\frac{\kappa}{\ln\{\frac{H\delta}{2z_o}\}}\right)^2 \left|U + \frac{1}{2}u_{LBOT} + \frac{1}{2}u_{LBOT-1}\right| \tag{3.17}$$

and $\gamma = 0.5$. The log-layer bottom boundary condition is activated by setting IBOTV=3.

Another bottom condition, which was included in the older version of MECCA, is one of no slip, or

$$U + u_{LBOT} = 0. \tag{3.18}$$

In this case, the bottom stress is set equal to the product of the viscosity at the mid-level of the lowest layer and the velocity gradient

$$\begin{aligned} \frac{\tau_b}{\rho} &= \frac{A_v}{H\delta}\Big((U + u_{LBOT-1}) - (U + u_{LBOT})\Big) \\ &= \frac{A_v}{H\delta}(U + u_{LBOT-1}) \end{aligned} \tag{3.19}$$

In the new version, the above expression is recast in the form of (3.13) by setting

$$\Phi = \frac{A_v}{H\delta} \tag{3.20}$$

and $\gamma = 0$. The no slip bottom boundary condition is activated by setting IBOTV=0.

Density Function and Gradients

Although there is a function for water density, there is no longer a separate function for difference in density. As before,

$$\rho = 10^3(1 + FRHO\{S,T\}) \tag{3.21}$$

The density difference is now computed directly by the difference in values of FRHO. This allows other density formulations to be used.

When either salinity or temperature are not being updated (c.f., KONCEN), they are set to their respective constant default values SAL0 (30 ppt) and TMP0 (10 C). If either salinity or temperature are being updated and a boundary file is missing, boundary values are set to the respective default values. This allows the full density equation (3.21) to be used with representative salinity or temperature values.

Vertical and horizontal density gradients can now be excluded selectively. For ICOUPL=0, both the vertical and the horizontal gradients are set to zero. For ICOUPL=1, only vertical gradients are included. For ICOUPL=2, both vertical and horizontal gradients are included.

Output for Graphing

The output file containing time series of model values at successive times has a new format. The first value in each record is the year, the second is the Julian day, and subsequent values are the selected modeled values. This makes them identical to the input data files.

Other attributes of the graphical file are selected by the parameters IGPH, NSTGPH, and IGPHOP (see Appendix C, Sample Control file). IGPH is the number of variables (as denoted by location L, N, M and ITYP) to be saved (up to 20). NSTGPH is the interval, in internal-mode time steps, that data are to be saved. IGHPOP (when set to 0) causes the graphical file header information not to be printed.

Non-Linear Horizontal Advection

The new code contains the corrected versions of the non-linear horizontal momentum advection terms in the internal mode calculations. For the terms affected, see Appendix A. These terms are approximated by upstream differencing as follows:

$$\frac{\partial uu}{\partial x} \Rightarrow \frac{1}{\Delta}[(1-\frac{u_m}{|u_m|})(u_{m+1}^2 - u_m^2) + (1+\frac{u_m}{|u_m|})(u_m^2 - u_{m-1}^2)] \tag{3.22}$$

The original code was corrected in about 1990 although the older version apparently did not affect the Gulf of Maine simulations (Brooks, 1992; Brooks and Churchill, 1992; Brooks, 1994). However, MECCA was used for a model inter-comparison study (Galloway et al., 1996). They simulated tidal flow around Rattray Island on Australia's east coast in an attempt to reproduce lee eddies that have been observed. Unfortunately, the authors were not successful in producing eddying flow with MECCA because they did not obtain a corrected version of the model. Although the authors were given a updated version of the code and plots showing the lee eddies, this information arrived too late to in their paper. Nevertheless, MECCA did produce eddies (Figure 3.2) that are virtually identical to those produced by the other three-dimensional models.

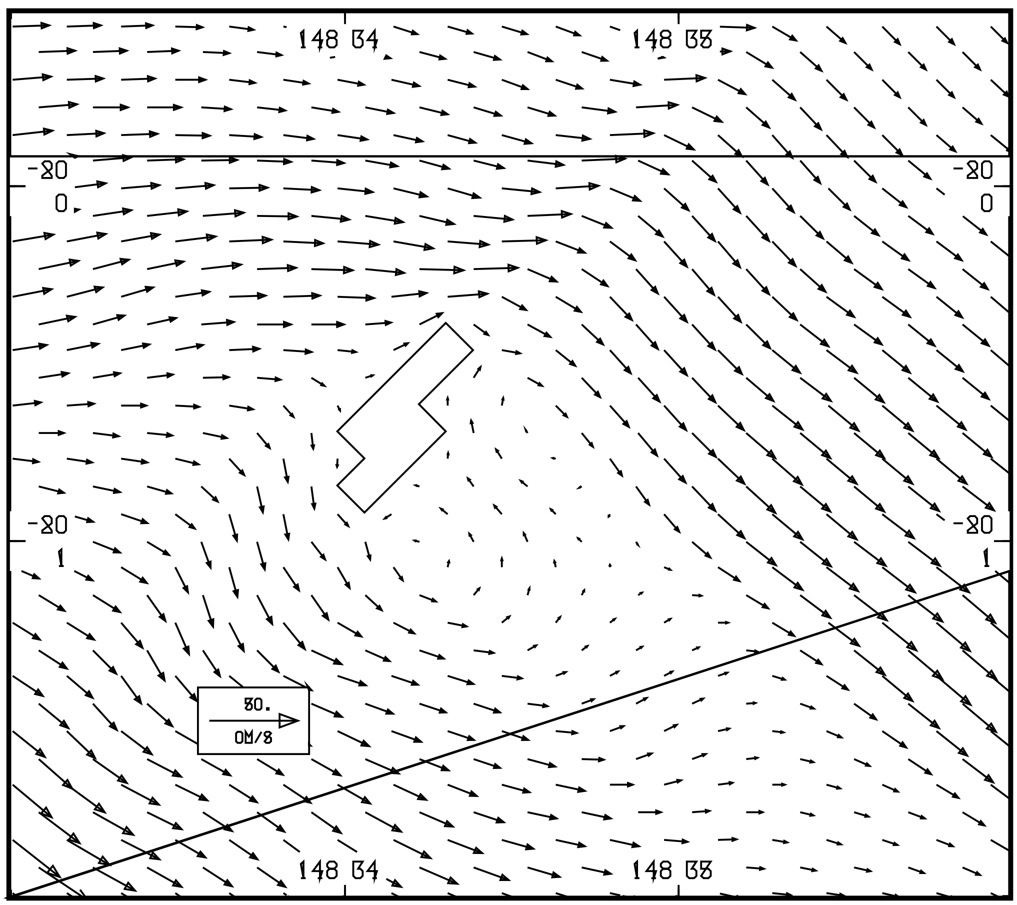


Figure 3.2. Instantaneous surface currents around Rattray Island, Australia.

4. ACKNOWLEDGMENTS

The author is indebted to Dr. Thor Aarup, who has been instrumental in modernizing portions of the code to make it more flexible. He has also assisted greatly in disseminating the code to other interested parties. Dr. Hassan Smaoui diligently checked the code and provided key corrections in the velocity calculations. He has also extended the model's capabilities by introducing a turbulence closure scheme for the diffusivity. Dr. Kathryn Bosley has also greatly spurred development of the new version by creating the specific needs of the Chesapeake Area Forecasting Experiment, in testing the model in a variety of scenarios, and in critically assessing the model's performance and assumptions.

5. REFERENCES

Berthet, C., 1996: Flow and three-dimensional coastal transport: numerical applications. Ph.D. thesis, Universite de Joseph Fourier, Grenoble, France (in French).

Bosley, K. T., and K. Hess, 1998: Development of a Experimental Nowcast/forecast System for Chesapeake Bay Water Levels. **Proceedings, 4th International Conference on Estuarine and Coastal Modeling**, Alexandria, VA, October 20-22, 1996. 413 - 426.

Bosley, K. T, 1996: Research aimed at prediction of water levels in the Chesapeake Bay. **Proceedings, Conference on Coastal Oceanic and Atmospheric Prediction**. AMS, Atlanta, 268 - 271.

Brooks, D. A., 1994: A model study of the buoyancy-driven circulation in the Gulf of Maine, **Journal of Physical Oceanography** 24 (11), 2387-2412.

Brooks, D. A., 1992: Tides and tidal power in Passamaquoddy Bay: a numerical simulation. **Continental Shelf Research** 12, 675-716.

Brooks, D. A., and L. U. Churchill, 1992: Experiments with a terrain-following hydrodynamic model for Cobscook Bay in the Gulf of Maine. In: **Estuarine and Coastal Modeling** (Spaulding et al., Eds), Am. Soc. Civil Eng., 786 pp.

Galloway, D., E. Wolanski, and B. King, 1996: Modeling eddy formation in coastal waters: a comparison between modeling capabilities. **Proceedings, 3rd International Conference on Estuarine and Coastal Modeling,** 13 - 25.

Haney, R. L., 1991: On the pressure gradient force over steep topography in sigma coordinate ocean models. **Journal of Physical Oceanography**, 21, 610 - 619.

Hess, K. W., 1994: Tampa Bay Oceanography Project: Development and Application of the Numerical Circulation Model. U.S. Department of Commerce, **NOAA Technical Report** NOS OES 005. 90 pp.

___, 1989: MECCA Program Documentation. U.S. Department of Commerce, **NOAA Technical Report** NESDIS 46. 258 pp.

___, 1988a: Linearized numerical model of constituent tides in Chesapeake Bay.(unpublished manuscript)

___, 1988b: Lagrangian drift model of suspended sediment transport in Chesapeake Bay, **Proceedings, Understanding the Estuary: Advances in Chesapeake Bay Research**, 352-368.

Hoff, M., 1990: A Chesapeake Bay Circulation Model. US Naval Academy, Annapolis, 26 pp.

Johnson, D. F., and K. W. Hess, 1990: Numerical simulations of blue crab larval dispersal and recruitment, **Bulletin of Marine Science**, 46(1), 195-213.

Johnson, D. F., and K. W. Hess , 1990: Numerical simulations of blue crab larval dispersal and recruitment, **Bulletin of Marine Science**, 46(1), 195-213.

King, B., S. Spagnol, E. Wolanski, and T. Done, 1997: Modeling the Mighty Burdekin River in Flood. **Proceedings, 4th International Conference on Estuarine and Coastal Modeling**, Alexandria, VA, October 20-22, 1996. 103-115.

Munk, W. H., and E. R. Anderson, 1948: Notes on the theory of the thermocline. **Journal of Marine Research**, 7, 276 - 295.

Smaoui, H., 1996: Three-dimensional numerical modeling of hydrodynamics and sediment transport in the eastern part of the English Channel and the southern part of the North Sea. Ph.D. thesis, University of Lille, France (in French).

Tag, P. M., F. W. Murray, and L. R. Koenig, 1979: A comparison of several forms of eddy viscosity parameterization in a two-dimensional cloud model. **Journal of Applied Meteorology**, 18, 1429 - 1441.

Wu, J., 1980: Wind-stress coefficients over sea surface under neutral conditions - a revisit. **Journal of Physical Oceanography**, 10, 727 - 740.

Wurtele, M. G., J. Paegle, and A. Sieleki, 1971: The use of open boundary conditions with the storm surge equations. **Monthly Weather Review**, 99, 6, 537 - 544.

APPENDIX A. ERRATA SHEET

This section contains some of the errors in the MECCA NESDIS report (Hess, 1989) in both the text (Part I) and the code (Part III).

Text Errors

No.	Page	Comments
1.	I-20	In Eq. 3.34, there should be no negative sign.
2.	I-24	In Eq. 4.16, Θ_{sv} should be $H\Theta_{sv}$
3.	I-33	Add the definition for $F_{n,m}$ (Eq. 4.8)

$$F_{n,m} = (\Phi_{n,m+1} + \Phi_{n,m})/(2H_{n,m}) + \beta_c THETSU_{n,m}$$

where $\Phi_{n,m}$ is evaluated at the cell center by Eq. 4.7 and

$$THETSU_{n,m} = C_{drgws} \mid UE_{n,m} \mid /(\Delta L\ BX_{n,m}) \sum_{k=1}^{LBOT-1} [CI_k (1 + U_{n,m,k} / \underline{UE}_{n,m}) \mid (1 + U_{n,m,k} / \underline{UE}_{n,m}) \mid]$$

And $\underline{UE}_{n,m} = \mathrm{sign}(UE_{n,m}) \max(\mid UE_{n,m} \mid, 0.001)$

and $CI_k = 1/(LBOT - 1)$

Add $C_{4A} = 2\Delta T/\Delta L^2$

Then in Eqs. 5.28 and 5.29, C_4 should be replaced by C_{4A}.

No.	Page	Comments
4.	I-42	Add the definition for $F_{n,m}$

$$F_{n,m} = (\Phi_{n+1,m} + \Phi_{n,m})/(2H_{n,m}) + \beta_c THETSV_{n,m}$$

where

$$THETSV_{n,m} = C_{drgws} \mid VE_{n,m} \mid /(\Delta L\ BY_{n,m}) \sum_{k=1}^{LBOT-1} [CI_k (1 + V_{n,m,k} / \underline{VE}_{n,m}) \mid (1 + V_{n,m,k} / \underline{VE}_{n,m}) \mid]$$

And $\underline{VE}_{n,m} = \mathrm{sign}(VE_{n,m}) \max(\mid VE_{n,m} \mid, 0.001)$

5. I-43 In Eqs. 6.20 and 6.21, C_4 should be replaced by C_{4A}.

6. I-54 In Eq. 8.33, the term $BHI_{n,m}$ (in first line) should be multiplied by 2.

Therefore, Eq. 8.34 should be revised to read: $C_3 = \Delta T/\Delta L^2$ (also applies to p. I-64, but text reads correctly)

7. I-55 In Eq. 8.45, the term multiplying β_a (first line) should be multiplied by 4. Therefore, add a new variable: $C_{7A} = \Delta T/(4\Delta L)$

8. I-56 In first line of Eq. 8.49, replace C_7 by C_{7A}.

9. I-63 In Eq. 9.3, Θ_{sv} should be $H\Theta_{sv}$

10. I-65 In first line of Eq. 9.26, replace C_7 by C_{7A}.

Program Errors

No.	Page	Comments
1.	III-70	Line 325, add CX9=2.*DT/DL**2. Lines 393, 394: change CX4 to CX9.
	III-70	Line 357, change 10. to 0.1/RHOW
	III-71	Line 413, the variable should be AHDUYY
	III-72	Line 477, change 10. to 0.1/RHOW
	III-73	Lines 513 and 515: change CX4 to CX9.
2.		Non-linear terms in Subroutine UPVP
	III-80	After Line 312, add B8=DTT/(4.*DL)
	III-81	Lines 388 and 389, change B7 to B8. Line 410, change UPM(L,M) to UPM(L,N)
	III-83	Lines 509 and 510, change =HI.. to =B8*HI.. Line 540, change VPMM to VPM.

APPENDIX B. SAMPLE GEOGRAPHY FILE

Sample MECCA2 Geography File.

```
  CHESAPEAKE BAY FOR MECCA2.0        TITLE
  2   0                              VERSION, ITEST
----- GRID PARAMETERS -------------------------------------------
GRID DIMENSIONS
   35  56   5606.00                  NMAX,MMAX,DL
 GRID ORIENTATION
   18  3  39.58  -76.14  16.0  38.   NCOR,MCOR,BSNLAT,BSNLON,BSNANG,REF
 OCEAN BOUNDARIES
    1                                NUMOBC 49 52 34 34 4 1 1
    49 52 34 34  2  1  1 1           MB1,MB2,NB1,NB2,ITPO,JTPO,ISET1,ISET2
 RIVER BOUNDARIES
    9                                NUMRIV
    3  3  18 18  2  2  1             MR1,MR2,NR1,NR2,ITRP,JTRP,ISETR
    11 11 20 20  1  2  2
    21 21 25 25 -2  2  3
    25 25 28 28  1  2  4
    24 24 15 15  1  2  5
    19 19  8  8  1  2  6
    34 34 12 12  2  2  7
    44 44 17 17  2  2  8
    50 50 12 12  2  2  9
 ----- CELL STATUS FIELD ----------------------------------------
 (100I2)                             DATA FORMAT
  35   5                             # COLUMNS/LINE, KOCNBC
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 012 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0602020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 02120 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 012102013 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 012222013 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0222020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 011202020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0202013 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0111210202012 060 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 01120202020 020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 02020202220 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01122112020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01120 02020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01020122220 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 011202020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 060 0 0 0 0 0 0 010202020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 020 0 0 0 0 0 0 011202022201212 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 01112 0 0 0 0 0 0 0202020202020202060 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 020 0 0 0 0 0 0 011202020 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 01013 0 0 0 0 0 0 0 02020202013 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 020 0 0 0 0 0 060 0 020202013 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 01013 0 0 0 0 0 020 0 011202012 0 0 0 0 0 060 0 0 0 0 0 0 0
 0 0 0 0 0 020 0 0 0 0 0 0 0112012 0 0112012 0 0 020 020 0 0 0 0 0 0 0
 0 0 0 0 01013 0 0 0 0 0 0 0 0 0112012 02020222012111220 0 0 0 0 0 0 0
 0 0 0 0 020 0 0 01012 0 0 0 0 0 0 011202020202020202012 0 0 0 0 0 0
 0 0 0 0 01112 0102220 0 0 0 0 0 0 0 01120202020 020202013 0 0 0 0 0
 0 0 0 0 0 0202013 0112012 0 0 0 0 0 0 0112020202022202012 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0112020202020212 012 0202020202020202020 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 01120202020132022201211202020 02020 02013 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0112020202020202020 02022202020 0 0
 0 0 0 0 0 0 0 0 0 0 06012 0 0 0 0 0 0 0112020202020202020202020 0 0
 0 0 0 0 0 0 0 0 0 0 0 01112 0 0 0 0 0 0 0 0 0 0112020202020202013 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 01112 0 0 0 0 0 0 0 0 0 02020202020202020 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0112012 0 0 0 0 0 0 020202020202020 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01112 0 0 0 0 0 020202020202013 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 011202012 0 0 011202020202020 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0111212 021202020202020 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 020202020202020202020 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 020202020202020 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 022202020202020 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0602012 0 0 0 0 0 0 0 020202020 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01112 0 0 0 0 0 0 020202020 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0112012 0 020202220202012 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01112 0 0112020202020 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0112020202020202020202012 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 020202020202020250 0
 0 0 0 0 0 0 0 0 0 0 0602020202020202020222012 0 0 0 0112020202050 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0111320 0 0 0 0 020202020250 0
```

```
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 011202012 0 0202020202050 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0112012202211202013 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 011202020 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 010 010 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
---- DEPTH FIELD ---------------------------------------------------
 (100I2)                                   DATA FORMAT
 35  0.01  0.2996                          # COLUMNS/LINE, HMSL, CON2M
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 015 5 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 718 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 41617 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 61519 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 61015 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7101215 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0101510 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 016161015181O 012 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9 7172012 015 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01520271720 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01338102030 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01555 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01955 01415 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01545152520 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 010404015 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 8 0 0 0 0 0 0 0 0 8401510 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 8 0 0 0 0 0 0 0 012327517151518 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 8 8 0 0 0 0 0 0 0 0307517322021211515 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 9 0 0 0 0 0 0 0 032752025 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 9 9 0 0 0 0 0 0 0 0 060351817 6 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 010 0 0 0 0 0 0 9 0 033602511 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 01312 0 0 0 0 0 0 9 0 010307030 0 0 0 0 0 0 8 0 0 0 0 0 0 0
 0 0 0 0 0 013 0 0 0 0 0 0 0 91515 0 0457510 0 0 0 3 0 9 0 0 0 0 0 0 0
 0 0 0 0 01111 0 0 0 0 0 0 0 0 0202325 03570 5 8 5 9 914 0 0 0 0 0 0 0
 0 0 0 0 012 0 0 0 7 8 0 0 0 0 0 0 04035337060 92015101511 0 0 0 0 0 0
 0 0 0 0 0 712 0 81818 0 0 0 0 0 0 0 02438803825 0 924 7 5 0 0 0 0 0 0
 0 0 0 0 0 0121318 02021 6 0 0 0 0 0 0 0 030428535 8 3322510 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 01622221616223032 021 0195020 9 3121316 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0152020202219353526161942502311 01130 0 7 4 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 03535404145502615 03710 610 4 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 7 7 0 0 0 0 0 0 0202835354550251040301014 9 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 8 8 0 0 0 0 0 0 0 0 0 0 020555020 850 73012 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 610 0 0 0 0 0 0 0 0 0 0 045452815501230 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0121216 0 0 0 0 0 0 01040252825203021 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01620 0 0 0 0 0 0 738323835321816 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 021212124 0 0 0 9354050383115 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0262618 010303344404220 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0303030303138385025 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 82233375228 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0131022375622 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0101010 0 0 0 0 0 0 015355035 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 01010 0 0 0 0 0 0 020355035 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0131515 0 01018111540454016 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 6 0 0 8162326354012 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0192028303015283227282 4 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 015 8163034252025 0
 0 0 0 0 0 0 0 0 0 0 0 0 7 7 7 7 7 7 8 9 9101313 0 0 0 0 4 92229402724 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0121213 0 0 0 0 0101920253025 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0101013 4 0 0151728293260 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0152035171517101017 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4111413 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9 0 9 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
----- CHANNEL WIDTH DATA ------------------------------------------
  41                                       NUMBXY
  3  18 1.00 0.20 0.20                     M,N,BX,BY,AREA  (SUSQUEHANNA RI.)
  11 20 0.20 1.00 0.20                     M,N,BX,BY,AREA  (CHESTER RI.)
  21 25 0.20 1.00 0.20                                     (CHOPTANK)
  21 24 0.25 1.00 0.25
  19  8 1.00 0.20 0.20                                     (POTOMAC)
  20  8 1.00 0.20 0.20
  21  8 0.20 0.20 0.20
  21  9 0.20 0.20 0.20
  22  8 1.00 0.22 0.22
  23  8 0.22 0.22 0.22
  23  9 0.22 0.22 0.22
  24  8 1.00 0.25 0.25
  24 15 1.00 0.20 0.20                                     (PATUXENT)
  25 15 1.00 0.20 0.20
  26 15 0.20 0.20 0.20
```

```
 26 16 0.20 1.00 0.20
 25 28 1.00 0.15 0.15                              (NANTICOKE)
 26 28 1.00 0.15 0.15
 27 28 1.00 0.15 0.15
 34 12 0.20 1.00 0.20                              (RAPPAHANNOCK)
 34 13 0.20 0.20 0.20
 35 13 0.20 0.20 0.20
 35 14 0.20 0.20 0.20
 36 14 0.22 0.22 0.22
 36 15 0.22 0.22 0.22
 37 15 0.24 0.24 0.24
 37 16 0.25 1.00 0.25
 44 17 0.20 1.00 0.20                              (YORK)
 44 18 0.20 1.00 0.20
 44 19 0.22 0.22 0.22
 45 19 0.22 0.22 0.22
 45 20 0.24 0.24 0.24
 46 20 0.25 0.25 0.25
 50 12 0.18 1.00 0.18                              (JAMES)
 50 13 0.19 1.00 0.19
 50 14 0.20 1.00 0.20
 50 15 0.21 1.00 0.21
 50 16 0.22 1.00 0.22
 50 17 0.23 1.00 0.23
 50 18 0.24 1.00 0.24
 50 19 0.25 1.00 0.25
----- END OF FILE ----------------------------------------------
```

APPENDIX C. SAMPLE CONTROL FILE

MECCA2 Control File. Asterisks in rightmost column denote lines that differ from the original version.

```
MECCA VERSION 2.0                          Public Release Version
fullbay20.geo                              NAME OF GEOGRAPHY FILE
 0   1   1                                 FILE VERSION,ITEST,KTEST    *
 ----- MODEL CONFIGURATION PARAMETERS ---------------------------------
 RUN PARAMETERS
 240.0  24.00    0.0    24.                HRMAX,HROUT,HROUT0,HRSAVE   *
 TIMESTEP(external), SPLIT, NUMBER OF LAYERS
  225.00   4    9                          DTE,ISPLIT,LAYRS
 TURBULENCE VARIABLES
  1.0  1.0     .01  1.0  -.05  1.E+3       AH00,AH0,CAH,DHAH,RIMIN,RIMAX*
 .003 .000010  .40 10.00  0.5  2.E+4       AV00,AV0,CRICH(1 - 4)        *
 .001 .000005  .10  3.33  1.5  2.E+4       DV00,DV0,CRICH(5 - 8)        *
   1   1   1.0                             IHVISC,IVISC,CR0             *
 DRAG COEFFICIENTS
 .0007  .0000 .000 .0008 .000065           CDWB1,CDWB2,CDRWS,CDRG1,CDRG2
 HEATING CONSTANTS
  0.10  6.0                                ALB,D10PCT                   *
 SWITCHES
  1   1   1   1                            ICOR,IBETAA,IBETAP,IBETAH
  1   0   0   0                            IEXTRN,INTER,KONCEN,ICOUPL
  3   2   0   0                            ITOPV,IBOTV,IHEAT,ICPOS      *
  ----- PRINT PARAMETERS-----------------------------------------------
 PLAN VIEW VARIABLES                       (s=surface, b=bottom)
  1 0 0 0 0  0 0 0 0  0 0 0 0     SE,UE,VE,Us,Vs,Ss,Sb,Ts,Tb,AH,AV,Wx,Wy*
 PAGE FORMATS
  2   1                                    KPRNT1(DIGIT,CHAR),2(NEW PG)
 CELLS WITH PRINT AT ALL LEVELS            M*1000+N
  2
  035026 047022                            NPRMN
 CELLS IN LONGITUDINAL SECTION             M*1000+N
  1                                        ISLICE:   JSLICE,MSNS
 10 006017 019017 020018 023018 026021 028021 030023 031023 035027 038027
 CELLS FOR LATER GRAPHING
  13   4    0                              IGPH,NSTGPH,IGPHOP            *
  1  50  34   1         ocean boundary             L,M,N,ITYP
  1  52  32   1         cbbt
  1  32  18   1         Colonial Beach
  1  11  13   1         baltimore
  1  33  21   1         lewisetta
  1  48  32   1         kiptopeake
  1  48  24   1         Glouster Pt
  1  28  20   1         Solomons
  1  15  16   1         Annapolis
  1  52  32   7         x-stress # 1 (CBBT)
  1  11  13   7         x-stress # 2 (Thomas Point)
  1  52  32   8
  1  11  13   8
 ----- TIME-VARIABLE BOUNDARY INPUTS----------------------------------
 STARTING DATE
 1994  1  1  00  00                   IYEAR,MONTH,IDAY,HOUR,MIN
=OCEAN WATER LEVELS
 1                                                                     *
wl.dat                                                                 *
=WINDS                                                                 *
 1                                                                     *
wind.dat.binary                                                        *
=RIVER FLOWRATES                                                       *
 9                                                                     *
rv.dat                                                                 *
=OCEANIC SALINITIES                                                    *
10                                                                     *
ocean_sal.dat                                                          *
=OCEANIC TEMPERATURES                                                  *
10                                                                     *
ocean_tmp.dat                                                          *
=RIVER TEMPERATURES                                                    *
 1                                                                     *
river_tmp.dat                                                          *
=ADDITIONAL MET DATA                                                   *
4                                                                      *
meto.dat                                                               *
=INITIAL CONDITIONS                                                    *
```

```
 0                                                                          *
-------- OUTPUT FILE NAMES---------------------------------------------*
FPRINT   :m20.prn                                                       *
FGRAPH   :m20.gph                                                       *
FMED     :MECMED.DAT                                                    *
=====END OF FILE ==========================================================
```

APPENDIX D. CODE LISTING

Code listing for program MECCA2. Common blocks appear at the end.

```
        PROGRAM MECCA2
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Model for Estuarine and Coastal Circulation Assessment II
C
C        PURPOSE - TO RUN A THREE-DIMENSIONAL (X-, Y-, Q-DIRECTION)
C                  HYDRO MODEL WITH NON-LINEAR ADVECTIVE TERMS AND
C                  VARIABLE DENSITY.  USES NUMERICAL SCHEME OF ABBOTT
C                  FOR THE EXTERNAL (2-D) MODE, AND ANOTHER SCHEME FOR
C                  THE INTERNAL (BAROCLINIC) MODE.  INCLUDES THOR
C                  AARUP'S VARIABLE ARRAY FEATURES.
C
C        LOGICAL UNITS USED
C                                                               Generic
C             Unit Name   No.    Used for                        Name
C             ---------   --     -----------------------       --------
C   INPUT:    LUCON        2     READ CONTROL FILE               FCON
C                                TO READ GEOGRAPHY FILE          FGEO
C                                READ INITIAL COND'S FILES       FINIT
C             LUKB         5     KEYBOARD                        -
C             LUTID       21     TIDE BC                         -
C             LUWND       22     GRIDDED WIND FILES              -
C             LURIV       23     RIVER FLOWRATES                 -
C             LUSAL       24     OCEAN SALINITY BC               -
C             LUOCT       25     OCEAN TEMPERATURE BC            -
C             LURVT       26     RIVER TEMPERATURES              -
C             LUMET       27     AIR PRESS,TMP,HUMIDITY          -
C
C   OUTPUT:   ISCR         6     CRT SCREEN                      -
C             IO          10     LINE PRINTER                    FPRINT
C             LUGRF       11     DATA AT GRAPH POINTS            FGRAPH
C             LUMED       12     INTERMEDIATE,FINAL VALUES       FMED
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C                     FLOW CHART FOR MAIN PROGRAM
C
C       MAIN
C         :
C         :------YTIME8
C         :------NYEAR
C         :
C         :------READZ-------ZEROS
C         :                  RDCON1
C         :                  RDCON2------JULIAN
C         :                  RDCON3      IRR
C         :                              RDWIND
C         :                  RDGEO-------CHECKS
C         :                  RDICS
C         :
C         :------INITS-------SETUP-------FLAGS
C         :                  SETSTP------BSTATE------RR
C         :                  THETAS
C         :                  WVERT
C         :
C         :------OUTPUT------PRNCON
C         :                  PRNBCG
C         :                  MEDSAV
C         :                  PGRAPH
C         :                  PRINTA------PRINTH-------DATES
C         :                              PRINTX-------PRNCHR
C   <BEGIN LOOP>                         PRINTI-------GRADP------FRHO
C         :                              PSLICE
C         :
C         :------FORCES------ATMOS-------RDWIND
C         :                  BSTRES
C         :                  ALGRAD
C         :
C         :------EXMODE------THETAS
C         :                  BNDRY1------RR
C         :                  HORVIS
C         :                  UHVH
C         :
C         :------INTRNL------BNDRY2
C         :                  VERVIS
C         :                  UPVP--------GRADP-------FRHO
C         :                  WVERT       GETCJ
C         :
C         :------CONCZ-------BNDRY3------BSTATE------RR
C         :                  BNDRY4      RR
C         :                  HEAT1
C         :                  GFLUX-------HEATZ
C         :
C         :------ANALYS
C         :
C         :------OUTPUT------PRNCON
C         :                  PRNBCG
C         :                  MEDSAV
C         :                  PGRAPH
C         :                  PRINTA------PRINTH-------DATES
C         :                              PRINTX-------PRNCHR
C    <END LOOP>                          PRINTI-------GRADP------FRHO
C         :                              PSLICE
C       (END)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C                                 VARIABLES
C
C                  AG = GRAVITATIONAL ACCELERATION (M/S**2)
C              AH(,) = HORIZONTAL EDDY VISCOSITY (M**2/S) TIMES DEPTH
C                       AT GRID CENTER
C             AHC(,) = HORIZONTAL EDDY VISCOSITY (M**2/S) TIMES DEPTH
C                       AT GRID CORNER
C            AH3(,,) = 3-D HORIZONTAL EDDY VISCOSITY (M**2/S) TIMES DEPTH
C                       AT GRID CENTER
C              AHMAX = MAXIMUM ALLOWABLE VALUE FOR AH(,)
C               AH00 = INITIAL VALUE OF HORIZONTAL EDDY VISCOSITY
C            AREA(,) = FRACTION OF AREA OF GRID THAT IS WATER
C             AV(,,) = VERTICAL EDDY VISCOSITY (M**2/S)
C               AV00 = INITIAL VALUE OF VERTICAL EDDY VISCOSITY
C             BSNANG = ANGLE BASIN GRID MUST BE ROTATED CLOCKWISE TO BE
C                       CORRECTLY ORIENTED ON EARTH'S SURFACE
C      BSNLAT,BSNLON = NORTH LATITUDE, WEST LONGITUDE OF BASIN GRID POI
C        BX(,),BY(,) = WIDTH FACTOR IN X-, Y-DIRECTION
C          CAW1,CAW2 = AIR-WATER INTERFACIAL DRAG COEFFICIENTS
C CRICH(1)-CRICH(8) = PARAMETERS IN RICHARDSON NUMBER REDUCTION OF
C                       DIFFUSION
C          CWB1,CWB2 = WATER-BOTTOM INTERFACIAL DRAG COEFFICIENTS
C               D(,) = MEAN SEA LEVEL DEPTH (M)
C             DENRAT = RATIO OF AIR DENSITY TO WATER DENSITY
C          DFDM,DFDN = CHANGE IN CORIOLIS PARAMETER PER UNIT CHANGE
C                       IN GRID INDEX M, N (TIMES .25*DTE)
C               DHAH = RATIO OF DH(N,M) TO AH(N,M)
C              DHMAX = MAXIMUM ALLOWABLE VALUE FOR DH(,)
C                 DL = GRID LENGTH (M)
C               DMAX = MAX. DEPTH IN COMPUTATIONAL REGION
C        DPADX,DPADY = ATMOSPHERIC PRESSURE GRADIENT (MB/KM)
C                 DQ = DIMENSIONLESS VERTICAL GRID INTERVAL
C                DTE = TIMESTEP (S) FOR EXTERNAL MODE
C                DTI = TIMESTEP (S) FOR INTERNAL MODE
C             DTEMAX = MAX. TIMESTEP (S) FOR EXTERNAL MODE
C             DTIMAX = MAX. TIMESTEP (S) FOR INTERNAL MODE.  UPDATED
C                       EACH TIME NEW TURBULENT VISCOSITY IS COMPUTED.
C             DV(,,) = VERTICAL MASS EXCHANGE COEFFICIENT
C               DV00 = INITIAL VALUE OF EDDY DIFFUSIVITY
C                  E = SMALL AMOUNT ADDED TO PREVENT ZERO DIVIDE
C        FA( ),FB( ) = RECURSION ARRAYS USED FOR THE IMPLICIT SCHEME
C           FEDGE(,) = EDGE FUNCTION TO RAMP UP FORCES AT OCEAN BOUNDAR
C        GA( ),GB( ) = ARRAYS USED FOR THE IMPLICIT SCHEME
C             HRCONC = TIME (HRS) WHICH MUST ELAPSE BEFORE S, T
C                       ARE UPDATED
C                 HR = HOURS ELAPSED IN THIS SEGMENT
C                HR0 = ELAPSED TIME (HRS) FROM PREVIOUS RUNS
C                HR1 = TOTAL ELAPSED TIME (HRS) = CUM HOURS
C             IBETAA = INDEX FOR INCLUSION OF NON-LINEAR ADVECTION
C                       TERMS: (0 = NO, 1=YES)
C             IBETAH = INDEX FOR NON-LINEAR FINITE-DEPTH TERMS (0=NO,
C                        1=YES)
C             IBETAP = INDEX TO INCLUDE HORIZONTAL PRESSURE GRADIENT
C                       DUE TO DENSITY STRUCTURE (0=NO, 1=YES)
C              IBOTV = INDEX FOR BOTTOM BC FOR THE INTERNAL-MODE VEL.
C                         0 = U IS ZERO AT BOTTOM
C                         1 = TAU=1ST-ORDER DU/DZ (DEFAULT WHEN INTER=0)
C                         2 = TAU=2ND-ORDER DU/DZ
C                         3 : AvDU/DZ = TAU   log-layer, mid-level
C             ICOUPL = INDEX TO INCLUDE DENSITY VARIATIONS
C                         0 = NO COUPLING
C                         1 = VERTICAL VARIATIONS ONLY (AV,DV)
C                         2 = VERT AND HORIZ (GRAD, GRX, GSTARX)
C              ICPOS = INDEX TO INSURE ZERO OR POSITIVE CONCENTRATIONS:
C                         0 = NO, OTHERWISE = YES
C    ICOL(,),IROW(,) = ARRAYS OF COLUMN AND ROW FLAG NUMBERS
C               ICOR = INDEX FOR CORIOLIS ACCEL (0 OR LESS=NONE,
C                       1=CONSTANT F,2 OR MORE=BETA PLANE APPX.)
C             IEXTRN = INDEX FOR DOING THE EXTERNAL MODE CALCS (0=NO,
C                       OTHERWISE YES)
C          IFIELD(,) = INDEX FOR COMPUTATIONAL GRID=IJ
C                        I = STATUS:0=LAND,1=HALF FULL,2=FULL,3=WATER
C                            LEVEL B.C.,4=RIVER FLOW B.C.
C                        J = BARRIER INDEX:0=X & Y FLOW,1=NO X FLOW,
C                            2=NO Y FLOW, 3=NO X, NO Y FLOW.
C               IGPH = NUMBER OF QUANTITIES TO BE GRAPHED
C              IHEAT = INDEX FOR ATMOSPHERIC HEATING:
C                        0=NO HEAT FLUX
C                        1=HEAT FLUX BUT WITH DAY OF YEAR CONSTANT (=NDAY0
C                        2=NORMAL HEATING
C             IHVISC = INTERVAL FOR UPDATING HORIZONTAL TURBULENT
C                       VISCOSITY (DTE TIMESTEPS)
C               ILFT = INDEX FOR LEFT-MOST BOUNDARY CONDITION:
C                        0 = WATER LEVEL (TIDE OR RADIATION CONDITION)
C                        1 = FLOWRATE (RIVER)
C                        2 = ZERO FLOWRATE (SOLID WALL)
C              INTER = INDEX FOR COMPUTING INTERNAL MODE VARIABLES
C                       (0=NO, OTHERWISE = YES)
C                 IO = PRINT OUTPUT CHANNEL NUMBER
C               IRGT = INDEX FOR RIGHT-MOST BOUNDARY CONDITION:
C                        0 = WATER LEVEL (TIDE OR RADIATION CONDITION)
C                        1 = FLOWRATE (RIVER)
C                        2 = ZERO FLOWRATE (SOLID WALL)
C               ISCR = CRT SCREEN OUTPUT CHANNEL NUMBER
C             ISPLIT = NO. OF EXTERNAL TIMESTEPS PER INTERNAL TIMESTEP
C              ITOPV = INDEX FOR TOP B.C. FOR INTERNAL MODE VELOCITY
C                       (1;TAU=1ST-ORDER DU/DZ, 2:TAU=2ND-ORDER DU/DZ)
C             ITPO() = DIRECTION OF OUTFLOW: 1=X, -1=-X, 2=Y, -2=-Y
C             JTPO() = TYPE OF OCEAN BOUNDARY CONDITION
C                        1 : WATER LEVEL SPECIFICATION
C                        2 : RADIATION OUTFLOW
C                        3 : RADIATION CONDITION, ORLANSKI
C                        4 : RADIATION CONDITION, RIEMANN
C             ITPR() = RIVER DIRECTION TYPE: 1=FLOW IN X, 2=FLOW IN Y
C                       (+=FLOW IN POSITIVE DIRECTION, -=FLOW IN NEG. DIR)
C             ITYP() = TYPE OF QUANTITY TO BE GRAPHED AT LGPH, MGPH,
C                          NGPH:
C                      1 : WATER LEVEL
C                      2 : UE
C                      3 : VE
C                      4 : UE + U
C                      5 : VE + V
C                      6 : W
C                      7 : WIND STRESS IN X DIRECTION
C                      8 : WIND STRESS IN Y DIRECTION
C                      9 : SALINITY
C                      10: TEMPERATURE (CELSIUS)
C                      11: AH3/H
C                      12: THETA1
C                      13: THETA2
C                      14: THETA3
C                      15: AV
C                      16: DV
C                      17: RI
C                      18: WIND IN X DIRECTION
C                      19: WIND IN Y DIRECTION
C                      20: UHOLD/H
```

```
C                       21: VHOLD/H
C               IVISC = INTERVAL FOR UPDATING VERTICAL TURBULENT
C                       VISCOSITY (DTI TIMESTEPS)
C             JPRNT() = INDICES FOR PRINTING OUT THE EXTERNAL FIELDS
C              JTPR() = RIVERINE CONDITION: 1=FLUME, 2=WATER FALLS
C              KONCEN = INDEX FOR UPDATING THE CONCENTRATIONS
C                        0=NO UPDATE, USE S=SAL0, T=TMP0
C                        1=UPDATE ONLY SALINITY, T=TMP0
C                        2=UPDATE ONLY TEMP., S=SAL0
C                        3=UPDATE BOTH SALINITY AND TEMP.
C            KPRNT1,2 = PRINT INDICES:  KPRNT1 : 1 PRINTS NUMBERS,
C                       2 PRINTS CHARACTERS IN PRINTX.  KPRNT2 : 2 SKIPS
C                       TO TOP OF PAGE IN PRINTX, PRINTI
C                   L = INDEX FOR VERTICAL LEVELS (1=TOP,LBOT=BOTTOM)
C               LAYRS = NUMBER OF LAYERS (LEVELS=LAYRS+1)
C                LBOT = NUMBER OF LEVELS, OR LEVEL OF BOTTOM (LBOT=LAYRS+1
C                   M = HORIZONTAL GRID NUMBER IN X DIRECTION
C                  MA = FIRST COMPUTATIONAL GRID IN COLUMN
C                 MAM = MA - 1, AND NUMBER OF LOWER B.C. GRID
C                   N = HORIZONTAL GRID NUMBER IN Y DIRECTION
C                  NA = FIRST COMPUTATIONAL GRID IN ROW
C                 NAM = NA - 1, AND NUMBER OF LOWER B.C. GRID IN ROW
C                  NB = LAST COMPUTATIONAL GRID IN ROW
C                NEGS = NUMBER OF EVENTS WHERE COMPUTED TOTAL WATER
C                       DEPTH IN CELLS IS NEGATIVE. WATER DEPTH RESET
C                       TO 0.10 M.
C                NSTE = INDEX FOR THE EXTERNAL TIMESTEP
C               NSTET = TOTAL NUMBER OF EXTERNAL TIMESTEPS
C               NSTIT = TOTAL NUMBER OF INTERNAL TIMESTEPS
C              NSTIMX = MAXIMUM NUMBER OF INTERNAL TIMESTEPS
C              NSTEMX = MAXIMUM NUMBER OF EXTERNAL TIMESTEPS PER NSTI
C               OMEGA = EARTH'S ROTATION RATE (RADIANS/S)
C              PHI(,) = EFFECTIVE BOTTOM DRAG COEFFICIENT
C                  PI = 3.1415927
C                   Q = DIMENSIONLESS VERTICAL VARIABLE (0=TOP, -1=BOT)
C             RATE( ) = RIVER FLOW RATE (M**3/S)
C                RHOW = REFERENCE WATER DENSITY
C              RI(,,) = RICHARDSON NUMBER
C              RICHNO = RICHARDSON NUMBER
C               S(,,) = SALINITY (PPT)
C                SAL0 = DEFAULT SALINITY
CSE(,),SEP(,),SEPP(,) = INITIAL, UPDATED VALUES OF SEA LEVEL ABOVE MSL
C             SOLD(,) = WATER LEVEL AT START OF INTERNAL TIMESTEP
C               T(,,) = WATER TEMPERATURE (C)
C       TBX(,),TBY(,) = INTERFACIAL WATER-BOTTOM STRESS (M**2/S**2) IN
C                       X-, Y-DIRECTION
C       TSX(,),TSY(,) = WIND STRESS PER UNIT WATER DENSITY (M**2/S**2)
C                       IN X-, Y-DIRECTION
C           THETA1(,) = (INTEGRAL OF (1 + U(,,)/UE(,))**2 OVER q)*BX*UE
C           THETA2(,) = (INTEGRAL OF (1 + V(,,)/VE(,))**2 OVER q)*BY*VE
C           THETA3(,) = (INTEGRAL OF (1 + U/UE)*(1+V/VE) OVER q)*UE*VE
C           THETSU(,) = INTEGRAL OF (1+U/UE)*ABS(1+U/UE) OVER q
C           THETSV(,) = INTEGRAL OF (1+V/VE)*ABS(1+V/VE) OVER q
C                TMP0 = DEFAULT WATER TEMPERATURE
C                  UT = UNIVERSAL TIME COORD (JULIAN TIME)
C                 UT0 = JULIAN TIME OF START OF VERY FIRST SEGMENT
C                 UT1 = JULIAN TIME OF START OF THIS SEGMENT
C               U(,,) = INTERNAL MODE HORIZONTAL VELOCITY
C                       (M/S) IN X-DIRECTION
C         UE(,),VE(,) = EXTERNAL MODE VELOCITY (EXCLUDES VARIABLE WIDTH
C                       FACTOR)
C        UH(,),UHP(,) = ORIGINAL, UPDATED HORIZONTAL FLOWRATE (M**2/S)
C                       IN X-DIRECTION
C   UHOLD(,),VHOLD(,) = INTEGRATED FLOWRATE OVER INTERNAL-MODE TIMESTEP (M
C               V(,,) = INTERNAL MODE HORIZONTAL VELOCITY
C                       (M/S) IN Y-DIRECTION
C        VH(,),VHP(,) = ORIGINAL, UPDATED HORIZONTAL FLOWRATE (M**2/S)
C                       IN Y-DIRECTION
C              VONKAR = VON KARMAN'S CONSTANT
C                 W10 = WIND SPEED AT 10 METERS
C               W(,,) = VERTICAL VELOCITY IN Q-COORDINATE (S**-1) TIMES
C                       THE LOCAL TOTAL DEPTH (M/S)
C         WX(,),WY(,) = WIND SPEEDS IN X-, Y-DIRECTION
C                YEAR = PRESENT YEAR AT EACH TIMESTEP
C               YEAR0 = YEAR AT START OF FIRST SEGMENT
C               YEAR1 = YEAR AT START OF THIS SEGMENT
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                       q(l)
C  PLAN      +---------+ Y(n)        ELEVATION     +-----Ul,n,m--+
C  VIEW      |         |             VIEW          |             |
C            |    Dn,m Vn,m                        |     Wl,n,m  |
C            |         |                           |             |
C            +----Un,m-+AHCn,m                     +---- Ul+1,n,m+
C            X(m)
C  AT D: SE,AH,AH3,S,T,AV,AREA,FEDGE,              AT U: V,AH3,S,T
C     PHI,TSX,TSY,WX,WY,W,SOLD,RI,IFIELD           AT W: RI,AV,DV
C     FX,FY,SEP,SEPP
C  AT U: UH,TBX,BX,GSTARX,GRX,UHOLD,
C     UE,THETA1,THETSU,MFLUX,UHP
C  AT V: VH,TBY,BY,GSTARY,GRY,VHOLD,
C     VE,THETA2,THETSV,NFLUX,VHP
C  AT AHC: THETA3
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      INCLUDE 'COMM20.FOR'
C         SET INTERNAL CONSTANTS
      DATA AG/9.81/,OMEGA/7.29E-5/,VONKAR/.40/,E/1.E-10/,PI/3.1415927/,
     1 SOLAR/1353./,PA/101400./,SB/5.67E-8/,EPSLON/.622/,RHOW/1000./,
     2 CWATER/4186./,RHOA/1.2/,CPAIR/1004./,ALV/2.6E+06/,CDRGWS/0./,
     3 RAD/.0174532927/,NEGS/0/,SAL0,TMP0/30.,10./,NDAYMO/
     4 31,28,31,30,31,30,31,31,30,31,30,31/,HRCON1,HRCON2/0.,48./,
     5 DTAU1,DTAU2/0.10,1.00/,ITEST/0/,VERS/1.00/,UT00/0.0/
C         LOGICAL UNIT NUMBERS
      DATA LUCON,LUKB,ISCR/2,5,6/,IO,LUGRF,LUMED/10,11,12/,LUTID,
     1 LUWND,LURIV,LUSAL,LUOCT,LURVT,LUMET/21,22,23,24,25,26,27/
C
C                 INITIATION PHASE
C
C         A. READ FILES
      CALL READZ
C
C         B. SET RUN PARAMETERS
      UT=UT1
      CALL YTIME8(YT,UT,YEAR1)
      CALL INITS
      NSTIMX=HRMAX*3600./DTI+0.5
      IHR=MAX1(1.,3600./DTI)
C
C         C. PRINT INITIAL VALUES
      CALL OUTPUT
C
C          ITERATION PHASE
C
      DO 300 NSTI=1,NSTIMX
      IPRNT1=0
      IF(NSTI.LE.IHR)IPRNT1=1
      IF(MOD(NSTI,2*IHR).EQ.0.AND.NSTI.LT.24*IHR)IPRNT1=1
      IF(MOD(NSTI,12*IHR).EQ.0)IPRNT1=1
      NSTIT=NSTIT+1                 !           NSTI+NST0
      DO 150 NSTE=1,ISPLIT
      NSTET=NSTET+1
c        time at end of steps NSTE
      HR=(FLOAT(NSTI-1)*DTI+FLOAT(NSTE)*DTE)/3600.
      HR1=HR+HR0
      CUMDAY=HR1/24.
      UT=UT1+HR/24.-UT00
      CALL NYEAR(UT,YEAR,UT00)
      CALL YTIME8(YT,UT,YEAR)
      IF(NSTI.EQ.1.AND.NSTE.EQ.1)WRITE(ISCR,110)UT
 110  FORMAT(/,X,'BEGIN COMPUTATIONS: UT=',F10.4)
C
C        D. GET TOP AND BOTTOM STRESSES
      CALL FORCES
C
C        E. CALCULATE THE EXTERNAL-MODE FLOWRATES AND WATER LEVELS
      IF(IEXTRN.NE.0)CALL EXMODE
 150  CONTINUE
C
C        F. GET VERTICAL AND HORIZONTAL INTERNAL MODE VELOCITIES
      IF(INTER.NE.0)CALL INTRNL
C
C        G. UPDATE CONCENTRATIONS
      IF(KONCEN.NE.0)CALL CONCZ
C
C        H. ANALYZE AND USE RESULTS
      CALL ANALYS
C
C        I. PRINT OUT VALUES
      IF(MOD(NSTI,IHR).EQ.0)WRITE(ISCR,200)HR,HR1,CUMDAY
 200  FORMAT(5X,'HOURS COMPLETED =',F8.1,'   TOTAL HOURS =',F8.1,
     1 5X,'CUM.DAY =',F7.2)

      CALL OUTPUT
C
C        LOOK FOR ABNORMAL STOP
      IF(ISTOP.EQ.1)GOTO 320
 300  CONTINUE

 320  STOP
      END
C
c-----------------------------------------------------------------------
      SUBROUTINE NYEAR(UT,YEAR,UT00)
c        increase year if Julian days exceed number in year
      IYEAR=YEAR
      IDAYS=365
      IF(MOD(IYEAR,4).EQ.0.AND.(.NOT.(MOD(IYEAR,100).EQ.0.AND.
     1 MOD(IYEAR,400).NE.0)))IDAYS=366
      IF(IFIX(UT).LE.IDAYS)RETURN
      UT00=UT00+UT-1.
      UT=UT-IDAYS
      YEAR=YEAR+1.
      RETURN
      END
C
C=======================================================================
C     MECCA FILE : INITS
C-----------------------------------------------------------------------
C
      SUBROUTINE INITS
C        JULY 1985   K. W. HESS   MEAD   VAX 11/750
C        PURPOSE - TO SET UP INITIAL GRID ARRAYS
C        VARIABLES -
C                 ICS = INDEX FOR READ INITIAL CONDITIONS:0=NO, 1=YES
      INCLUDE 'COMM20.FOR'
C        PRINT OUT INITIAL VALUES
      WRITE(ISCR,100)HRMAX,HROUT,HRCONC
 100  FORMAT(5X,'INITS: HRMAX=',F6.1,' HROUT=',F7.2,' HRCONC=',F7.1)
C        INITIALIZE THE ARRAYS
      CALL SETUP
C        INITIALIZE DIAGNOSTIC VARIABLES
      IF(KONCEN.GT.0)CALL SETSTP
      IF(ICS.EQ.1)CALL THETAS
      IF(ICS.EQ.1)CALL WVERT
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE FLAGS
C        MAY 1984   K. W. HESS   TDL   IBM
C        PURPOSE - TO PROCESS THE FIELD ARRAY AND SET UP THE ROW AND
C                  COLUMN IDENTIFIERS.
C        VARIABLES -
C           ICOL(K,L) = L-TH SET OF FLAG NUMBERS FOR A GRID COLUMN
C                       K=1 : COLUMN NUMBER, N
C                       K=2 : FIRST COMPUTATIONAL ROW, MA
C                       K=3 : LAST COMPUTATIONAL ROW, MB
C                       K=4 : LOWER END BOUNDARY CONDITION:
C                                  0=WATER LEVEL
C                                  1=FLOWRATE
C                                  2=ZERO FLOW
C                       K=5 : UPPER END BOUNDARY CONDITION (SEE K=4)
C         IFIELD(N,M) = FIELD SPECIFICATION ARRAY = IJ =10*I+J, WHERE
C                       I=0 : LAND GRID
C                         1 : HALF LAND, HALF WATER GRID
C                     2,5,6 : WATER GRID(S)
C         JFIELD(N,M) = FIELD SPECIFICATION ARRAY = IJ =10*I+J, WHERE
C                       I=0 : LAND GRID
C                         1 : HALF LAND, HALF WATER GRID
C                         2 : NON-BOUNDARY WATER GRID
C                         3 : BOUNDARY GRID FOR WATER LEVEL
C                         4 : BOUNDARY GRID FOR FLOWRATE
C                         5 : BOUNDARY GRID FOR WATER FALLS
C                    KOCNBC : OCEANIC BOUNDARY CONDITION GRID (=5)
C                    KRIVBC : RIVERINE BOUNDARY CONDITION GRID=KOCNBC+1
C                       J=0 : X-, Y-FLOW
C                         1 : NO X-FLOW
C                         2 : NO Y-FLOW
C                         3 : NO X-, Y-FLOW
      INCLUDE 'COMM20.FOR'
      DIMENSION NUM(NSIZE),JFIELD(NSIZE,MSIZE)
      KRIVBC=KOCNBC+1
      KFALLS=KOCNBC-1
      DO 100 N=1,NMAX
      NUM(N)=N
      DO 100 M=1,MMAX
 100  JFIELD(N,M)=IFIELD(N,M)
C        RESET JFIELD AT BOUNDARIES: 3=WATER LEVEL  4=FLOWRATE  5=WATER FA
      IF(NUMOBC.LE.0)GOTO 120
      DO 110 I=1,NUMOBC
      DO 110 M=MB1(I),MB2(I)
```

```
      DO 110 N=NB1(I),NB2(I)
      JFIELD(N,M)=30+MOD(JFIELD(N,M),10)
      IF(JTPO(I).EQ.2)JFIELD(N,M)=40+MOD(JFIELD(N,M),10)
  110 CONTINUE
C        RIVERS
  120 IF(NUMRIV.LE.0)GOTO 140
      DO 130 I=1,NUMRIV
      DO 130 M=MR1(I),MR2(I)
      DO 130 N=NR1(I),NR2(I)
      IF(JTPR(I).EQ.1)JFIELD(N,M)=40+MOD(JFIELD(N,M),10)
      IF(JTPR(I).EQ.2)JFIELD(N,M)=20+MOD(JFIELD(N,M),10)
  130 CONTINUE

C        FIND THE ICOL PARAMETERS
  140 DO 150 J=1,NM2SIZ
      DO 150 I=1,5
      ICOL(I,J)=0
  150 IROW(I,J)=0
      ICOUNT=0
      NCOL=0
      DO 190 N=1,NMAX
      ISTART=0
      DO 190 M=1,MMAX
      IC=JFIELD(N,M)/10
      IF(IC.EQ.0.OR.IC.GE.3)GOTO 190
C        INSERT GRID LINE SPECS IF THIS GRID STARTS A LINE
      IF(ISTART.EQ.1)GOTO 170
      ISTART=1
      NCOL=NCOL+1
C        DO A FIX-UP IF NCOL IS TOO LARGE FOR ARRAY
      IF(NCOL.LE.NM2SIZ)GOTO 160
      NCOL=NM2SIZ
      ICOUNT=ICOUNT+1
  160 ICOL(1,NCOL)=N
      ICOL(2,NCOL)=M
C        DETERMINE LOWER END BOUNDARY CONDITION
      ICOL(4,NCOL)=2
      IF(M.EQ.1)GOTO 170
      I=JFIELD(N,M-1)/10
      J=JFIELD(N,M-1)-10*I
      IF(J.EQ.1.OR.J.EQ.3)GOTO 170
      IF(I.EQ.3)ICOL(4,NCOL)=0
      IF(I.EQ.4)ICOL(4,NCOL)=1
      IF(I.EQ.5)ICOL(4,NCOL)=2
C        CHECK FOR UPPER END BOUNDARY CONDITIONS
  170 ICOL(3,NCOL)=M
      ICOL(5,NCOL)=2
      IF(M.EQ.MMAX)GOTO 180
      J=JFIELD(N,M)-10*(JFIELD(N,M)/10)
      IF(J.EQ.1.OR.J.EQ.3)GOTO 180
      IP=JFIELD(N,M+1)/10
      IF(IP.GT.0.AND.IP.LT.3)GOTO 190
      IF(IP.EQ.3)ICOL(5,NCOL)=0
      IF(IP.EQ.4)ICOL(5,NCOL)=1
      IF(IP.EQ.5)ICOL(5,NCOL)=2
  180 ISTART=0
  190 CONTINUE
C        FIND THE IROW PARAMETERS
  200 JCOUNT=0
      NROW=0
      DO 240 M=1,MMAX
      ISTART=0
      DO 240 N=1,NMAX
      IC=JFIELD(N,M)/10
      IF(IC.EQ.0.OR.IC.GE.3)GOTO 240
C        BEGIN TO INSERT LINE SPECS
      IF(ISTART.GT.0)GOTO 220
      ISTART=1
      NROW=NROW+1
C        DO A FIX-UP IF NECESSARY
      IF(NROW.LE.NM2SIZ)GOTO 210
      NROW=NM2SIZ
      JCOUNT=JCOUNT+1
  210 IROW(1,NROW)=M
      IROW(2,NROW)=N
C        DETERMINE LOWER END BOUNDARY CONDITION
      IROW(4,NROW)=2
      IF(N.EQ.1)GOTO 220
      I=JFIELD(N-1,M)/10
      J=JFIELD(N-1,M)-10*I
      IF(J.EQ.2.OR.J.EQ.3)GOTO 220
      IF(I.EQ.3)IROW(4,NROW)=0
      IF(I.GE.4)IROW(4,NROW)=1
      IF(I.GE.5)IROW(4,NROW)=2
C        CHECK FOR UPPER END BOUNDARY
  220 IROW(3,NROW)=N
      IROW(5,NROW)=2
      IF(N.EQ.NMAX)GOTO 230
      J=JFIELD(N,M)-10*(JFIELD(N,M)/10)
      IF(J.EQ.2.OR.J.EQ.3)GOTO 230
      IP=JFIELD(N+1,M)/10
      IF(IP.GT.0.AND.IP.LT.3)GOTO 240
      IF(IP.GE.3)IROW(5,NROW)=0
      IF(IP.GE.4)IROW(5,NROW)=1
      IF(IP.GE.5)IROW(5,NROW)=2
  230 ISTART=0
  240 CONTINUE
C        CHECK THE COUNTS
  280 IF(ICOUNT.GT.0)WRITE(IO,290)NM2SIZ,ICOUNT
  290 FORMAT(5X,'*** ERROR IN FLAGS: NUMBER OF ROW STRINGS EXCEEDS ',I4,
     1 ' BY ',I4)
      IF(JCOUNT.GT.0)WRITE(IO,295)NM2SIZ,JCOUNT
  295 FORMAT(5X,'*** ERROR IN FLAGS: NUMBER OF COLUMN STRINGS EXCEEDS',
     1  I4,' BY ',I4)
C        GET FIRST, LAST COLUMN IN EACH ROW.
      DO 310 M=1,MMAX
      NA=NMAX+1
      NB=1
      DO 300 N=1,NMAX
      IF(IFIELD(N,M)/10.EQ.0)GOTO 300
      NA=MIN0(N,NA)
      NB=MAX0(N,NB)
  300 CONTINUE
      NAB(M)=1000*NA+NB
  310 CONTINUE
C
C        CREATE NEW IFIELD BY INSERTING RIGHT SIDE AND
C                   BOTTOM SIDE BARRIERS IN WATER CELLS
      DO 320 M=1,MMAX
      DO 320 N=1,NMAX
      IF(IFIELD(N,M)/10.EQ.0)GOTO 320
c     IF(IFIELD(N,M)/10.EQ.KOCNBC)GOTO 320
c     IF(IFIELD(N,M)/10.EQ.KRIVBC)GOTO 320
      IX=0
      IF(M.EQ.MMAX.OR.IFIELD(N,MIN0(M+1,MMAX))/10.EQ.0)IX=1
      IF(MOD(IFIELD(N,M),10).EQ.1.OR.MOD(IFIELD(N,M),10).EQ.3)IX=1
      IY=0
      IF(N.EQ.NMAX.OR.IFIELD(MIN0(N+1,NMAX),M)/10.EQ.0)IY=2
      IF(MOD(IFIELD(N,M),10).EQ.2.OR.MOD(IFIELD(N,M),10).EQ.3)IY=2
      IFIELD(N,M)=10*(IFIELD(N,M)/10)+IX+IY
  320 CONTINUE
C        OCEAN BOUNDARY CELLS
      IF(NUMOBC.LE.0)GOTO 400
      DO 390 I=1,NUMOBC
      IX=0
      IY=0
      IF(IABS(ITPO(I)).EQ.1)THEN
      IY=2
      IF(ITPO(I).GT.0.AND.IFIELD(N,M+1).LT.9)IX=1
      ENDIF
      IF(IABS(ITPO(I)).EQ.2)THEN
      IX=1
      IF(ITPO(I).GT.0.AND.IFIELD(N+1,M).LT.9)IY=2
      ENDIF
      DO 390 M=MB1(I),MB2(I)
      DO 390 N=NB1(I),NB2(I)
  390 IFIELD(N,M)=10*KOCNBC+IX+IY
C        RIVERS
  400 IF(NUMRIV.LE.0)GOTO 430
      DO 420 I=1,NUMRIV
      IF(JTPR(I).NE.2)GOTO 420
      IX=0
      IY=0
      IF(IABS(ITPR(I)).EQ.1)IY=2
      IF(IABS(ITPR(I)).EQ.2)IX=1
      DO 410 M=MR1(I),MR2(I)
      DO 410 N=NR1(I),NR2(I)
  410 IFIELD(N,M)=10*KRIVBC+IX+IY
  420 CONTINUE
  430 CONTINUE
C            MFLUX(,) = INDEX FOR MASS FLUX IN X-DIRECTION (0=NO, 1=YES)
C            NFLUX(,) = INDEX FOR MASS FLUX IN Y-DIRECTION (0=NO, 1=YES)
C        LOOP DOWN THE LINES
      DO 450 M=1,MMAX
      MP=MIN0(MMAX,M+1)
      DO 450 N=1,NMAX
      NP=MIN0(NMAX,N+1)
      I=MOD(IFIELD(N,M),10)
      IX=1
      IF(IFIELD(N,M).LT.10.OR.IFIELD(N,MP).LT.10)IX=0
      IF(I.EQ.1.OR.I.EQ.3.OR.M.EQ.MMAX)IX=0
      MFLUX(N,M)=IX
      IY=1
      IF(IFIELD(N,M).LT.10.OR.IFIELD(NP,M).LT.10)IY=0
      IF(I.EQ.2.OR.I.EQ.3.OR.N.EQ.NMAX)IY=0
      NFLUX(N,M)=IY
  450 CONTINUE
      RETURN
      END
C
C-------------------------------------------------------------
C
      SUBROUTINE SETUP

C        MAY 1984   (REVISED APRIL 88)  K. W. HESS
C        PURPOSE - TO INITIALIZE THE PARAMETERS AFTER ALL INPUT
C                  FILES HAVE BEEN READ IN.
C        VARIABLES -
C               DPADX = GRADIENT OF ATMOS. PRESSURE IN X-DIRECTION
C                       (UNITS = MB/KM = (100 N/M**2)/(1000 M)
C                       SO DPADX*(.1/RHOW) HAS MKS UNITS
C                 ICS = INDEX FOR READ IN OF INITIAL CONDITIONS
C                       (0=NO, 1=YES)

      INCLUDE 'COMM20.FOR'
C        CHECK BOTTOM DRAG
      IF(INTER.EQ.0)IBOTV=1
C        SET CORIOLIS PARAMETERS
      COR=0.0
      IF(ICOR.NE.0)COR=2.*OMEGA*SIN(RAD*BSNLAT)
      FCOR0=DTE*COR
      DFDM=0.0
      DFDN=0.0
      IF(IABS(ICOR).GE.2)THEN
      CC=-DTE*2.*OMEGA*COS(BSNLAT*RAD)*DL*RAD/60./1852.
      DFDM=CC*COS(BSNANG*RAD)
      DFDN=CC*SIN(BSNANG*RAD)
      END IF
C        SET DENSITY COUPLING
      ICOUPL=MIN0(MAX0(0,ICOUPL),2)
C        COMPUTE VERTICAL GRID PARAMETERS
      DQ=1./FLOAT(LAYRS)
      HALFDQ=.5*DQ
      TWODQ=2.*DQ
      LBOT=LAYRS+1
C        GET IFIELD
      CALL FLAGS    ! newest version
C        INITIALIZE VISCOSITIES AND GET MAX, MIN CELL DEPTHS
      DMAX=0.0
      DMIN=1.E+10
      DO 130 M=1,MMAX
      DO 130 N=1,NMAX
      IF(IFIELD(N,M).LT.10)GOTO 130
      DMAX=AMAX1(DMAX,D(N,M))
      DMIN=AMIN1(DMIN,D(N,M))
      IF(ICS.EQ.1)GOTO 130
      AH(N,M)=D(N,M)*AH00
      AHC(N,M)=D(N,M)*AH00
      DO 120 L=1,LBOT
      AH3(L,N,M)=AH(N,M)
      S(L,N,M)=SAL0
      T(L,N,M)=TMP0
      IF(L.LT.LBOT)DV(L,N,M)=DV00
  120 IF(L.LT.LBOT)AV(L,N,M)=AV00
  130 CONTINUE

C        GET VERTICAL INTEGRATION COEFFICIENT FOR LEVELS (NOT LAYERS)
  148 DO 160 L=1,LBOT
      CI(L)=1.0/FLOAT(LBOT-1)
      IF(L.EQ.1.OR.L.EQ.LBOT)CI(L)=0.5/FLOAT(LBOT-1)
  160 CONTINUE
C        EDGE FUNCTION
      DO 250 M=1,MMAX
      DO 250 N=1,NMAX
      FEDGE(N,M)=1.0
      II=IFIELD(N,M)/10
      IF(II.EQ.KOCNBC)FEDGE(N,M)=0.0
      IF(M.EQ.1.OR.M.EQ.MMAX)GOTO 250
      IF(N.EQ.1.OR.N.EQ.NMAX)GOTO 250
      IF(II.EQ.KOCNBC)GOTO 250
      IF(IFIELD(N,M+1)/10.EQ.KOCNBC.OR.IFIELD(N,M-1)/10.EQ.KOCNBC.
     1 OR.IFIELD(N+1,M)/10.EQ.KOCNBC.OR.IFIELD(N-1,M)/10.EQ.KOCNBC)
```

```
     2 FEDGE(N,M)=0.5
 250  CONTINUE
      RETURN
      END
C=======================================================================
C     MECCA FILE : MPRINT
C-----------------------------------------------------------------------
C
      SUBROUTINE OUTPUT
C        APRIL 1988    HESS   MEAD   VAX
C        PURPOSE - TO PRINT OUT VARIABLES AT START, OTHER SELECTED TIMES
C                  AND END
      INCLUDE 'COMM20.FOR'
C
C        INITIAL PRINTOUTS
      IF(NSTI.EQ.0)CALL PRNCON(1)
      IF(NSTI.EQ.0)CALL PRNBCG(1)
C        SAVE INTERMEDIATE RESULTS EACH HRSAVE HOURS
      IF(NSTI.GT.0.AND.MOD(NSTI,IHR*IFIX(HRSAVE)).EQ.0)CALL MEDSAV
C        STORE VALUES FOR GRAPHING
      CALL PGRAPH
C        PRINT OUT SELECTED FIELDS
      CALL PRINTA
C         END OF RUN
      IF(ISTOP.EQ.1)GOTO 120
      IF(NSTI.LT.NSTIMX)GOTO 700
C        WRITE HOURS TO CONSOLE
 120  NRR=NN+IFIX(HR0)
      CALL PRNCON(0)
      WRITE(IO,525)
 525  FORMAT(/,1X,'VII. RUN COMPLETION',55('-'),/)
      WRITE(IO,520)NEGS
 520  FORMAT(5X,'NO. OF TIMES A WATER TOTAL DEPTH WENT NEGATIVE=',I4)
      WRITE(IO,530)FPRINT,FMED
      WRITE(ISCR,530)FPRINT,FMED
 530  FORMAT(5X,'PRINT OUTPUT IS IN FILE : ',A40,/
     1       5X,'VELOCITY DATA IS IN FILE: ',A40)
      IF(IGPH.GT.1)WRITE(ISCR,540)FGRAPH
      IF(IGPH.GT.1)WRITE(IO  ,540)FGRAPH
 540  FORMAT(5X,'GRAPHING OUTPUT IS IN   : ',A40)
 620  WRITE(ISCR,630)
 630  FORMAT(1X,'RUN COMPLETE')
      IF(IO.NE.ISCR)WRITE(IO,640)
 640  FORMAT(1X,'RUN COMPLETE',//,1X,74('-'))
      WRITE(IO,  *)'ISTOP=',ISTOP
      WRITE(ISCR,*)'ISTOP=',ISTOP
 700  CONTINUE
      RETURN
      END
C
C--------------------------------------------------------------------
C
      SUBROUTINE MEDSAV
C       PURPOSE - TO SAVE RESULTS TO AN INTERMEDIATE FILE
C
      INCLUDE 'COMM20.FOR'
C        NOTIFY PRINT FILE
      WRITE(IO,110)NSTI,UT
 110  FORMAT(/,1X,'MEDSAV:  NSTI=',I5,' UT=',f10.4)
C        OPEN FILES
      CALL FUOPEN(LUMED,FMED)
C        WRITE TO DATA FILE
      K=10+KONCEN
      WRITE(LUMED)NMAX,MMAX,LBOT,NSTET,UT,YEAR,K
      WRITE(LUMED)SE,UE,VE,SOLD,UHOLD,VHOLD,AH,AV,PHI,TBX,TBY,
     1 U,V,W,THETA1,THETA2,THETA3
      WRITE(LUMED)AH3,WX,WY,TSX,TSY
      IF(KONCEN.GT.0)WRITE(LUMED)S,T,DV,RI,NSTINF
C        CLOSE FILES
      CLOSE (LUMED)
 120  RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE PRINTA
C        1986   MEAD   K.HESS   VAX
C        PURPOSE - TO CALL THE VARIOUS OCCASIONAL PRINTING SUBROUTINES
C        VARIABLES -
C                 HROUT0 = HOUR OF FIRST P/O
      INCLUDE 'COMM20.FOR'
      NOUT0=AMAX1(1.,HROUT0*3600.01/DTI)
      NOUT1=AMAX1(1.,HROUT *3600.01/DTI)
      IF(HR1.LT.HROUT0)GOTO 120
c     IF(.NOT.(NSTI.EQ.NSTIMX.OR.MOD(NSTIT-NOUT0,NOUT1).EQ.0.OR.
      IF(.NOT.(NSTI.EQ.NSTIMX.OR.MOD(NSTIT,NOUT1).EQ.0.OR.
     1 NSTI.EQ.0.OR.ISTOP.EQ.1))GOTO 110
      write(6,*)'PRINTA: NOUT0,NOUT1=',nout0,nout1
      IF(NSTI.EQ.0)WRITE(IO,100)
 100  FORMAT(//,1X,'IV. INITIAL FIELDS',55('-'))
C        PRINT (B) TOP VIEW OF EXTERNAL MODE
      CALL PRINTX
C        PRINT (C) ALONG A SLICE
      CALL PSLICE
C        PRINT (D) AT ALL LEVELS AT ONE CELL
      CALL PRINTI
 110  CONTINUE
 120  CONTINUE
      IF(NSTI.EQ.0)WRITE(IO,130)
 130  FORMAT(/,1X,'V. RUN-TIME OUTPUT',55('-'))
      RETURN
      END
C
C------------------------------------------------------------------
C
      SUBROUTINE PRINTI
C        MAY 1984   K. W. HESS
C        PURPOSE - TO PRINT OUT THE INTERNAL-MODE VARIABLES.
      INCLUDE 'COMM20.FOR'
      DIMENSION RJ(7),FPX(LSIZE),FPY(LSIZE),WT(LSIZE),TRAD(LSIZE)
C        CHECK FOR PRINT
      IF(NPRMN.LE.0)GOTO 450
      WRITE(IO,100)UT
 100  FORMAT(/,1X,'D. INTERNAL MODE VARIABLES AT ALL LEVELS IN A CELL',
     1 ' AT UT=',F9.4)
      WFACT=10.**NINT(ALOG10(DL/DMAX/DQ))
C        GET ROW OR COLUMN INDICES
      DO 440 I=1,NPRMN
      M=IPRMN(I)/1000
      N=IPRMN(I)-1000*M
      MP=MIN0(M+1,MMAX)
      NP=MIN0(N+1,NMAX)
C        CHECK FOR CELL INSIDE WATER FIELD
      IF(N.GE.1.AND.M.GE.1.AND.N.LE.NMAX.AND.M.LE.MMAX)GOTO 120
      WRITE(IO,110)N,M
 110  FORMAT(3X,'ERROR: CELL N=',I2,', M=',I2,' IS OUTSIDE GRIDMESH')
      GOTO 440
 120  IF(IFIELD(N,M).GE.10)GOTO 140
      WRITE(IO,130)N,M
 130  FORMAT(3X,'ERROR: CELL N=',I3,', M=',I3,' IS NOT WATER')
      GOTO 440
 140  CONTINUE
      WRITE(IO,150)N,M,UT,NSTI,WFACT
 150  FORMAT(/,X,'N=',I3,'  M=',I3,' UT=',F10.4,' NSTI=',I6,
     1 7x,'f=',E10.4,/,1X,
     1 '  L',6x,'UTm',7X,'UTp',7x,'VTm',7X,'VTp',7X,'Wxf',7X,
     2 ' S ',7x,' T ')
      IF(M.LE.0.OR.N.LE.0)GOTO 440
C        VELOCITIES
      DHDT1=(SE(N,M)-SOLD(N,M))/DTI
      DO 160 L=1,LBOT
      RJ(1)=UE(N,M-1)+U(L,N,M-1)
      RJ(2)=UE(N,M)+U(L,N,M)
      RJ(3)=VE(N-1,M)+V(L,N-1,M)
      RJ(4)=VE(N,M)+V(L,N,M)
      Q=FLOAT(1-L)*DQ
      WT(L)=W(L,N,M)+(1.+Q)*DHDT1
      RJ(5)=W(L,N,M)*WFACT
      RJ(6)=S(L,N,M)
      RJ(7)=T(L,N,M)
      WRITE(IO,155)L,(RJ(K),K=1,7)
 155  FORMAT(2X,I2,7F10.3,F10.4)
 160  CONTINUE
C        INTERNAL HEATING
      FS=0.0
      FQ=0.0
      IF(NSTI.GT.ISPLIT.AND.IHEAT.GT.0.AND.KONCEN.GE.2)
     1  CALL HEATZ(N,M,TRAD,FTSURF)
      WRITE(IO,170)
 170  FORMAT(1X,'  L',4x,'10+4 Av ',2X,'10+4 Dv ',5X,'Ri',7X,
     1 'Trad',6X,'AH3',6x,'Rho',5x,'10+6 Tau')
C        DIFFUSIVITY, DENSITY, STRESS
      DZ=2.*DQ*(D(N,M)+SE(N,M))
      DO 195 L=1,LAYRS
      RJ(1)=AV(L,N,M)*10000.
      RJ(2)=DV(L,N,M)*10000.
      RJ(3)=RI(L,N,M)
      TRAD(L)=0.
      TW=T(1,N,M)
      RJ(4)=TRAD(L)
      RJ(5)=AH3(L,N,M)/(D(N,M)+SE(N,M))
      RJ(6)=RHOW+1000.*FRHO(S(L,N,M),T(L,N,M))
      RJ(7)=1.E+6*AV(L,N,M)*(U(L,N,M)-U(L+1,N,M)+U(L,N,M-1)
     1 -U(L+1,N,M-1)+V(L,N,M)-V(L+1,N,M)+V(L,N-1,M)-V(L+1,N-1,M))/DZ
      WRITE(IO,180)L,(RJ(K),K=1,7)
c180  FORMAT(2X,I2,5F10.3,3x,F10.4)
 180  FORMAT(2X,I2,7F10.3)
 195  CONTINUE

C        TWO-DIMENSIONAL VARIABLES
      MP=MIN0(M+1,MMAX)
      NP=MIN0(N+1,NMAX)
      H1=0.5*(D(N,M)+SE(N,M)+D(N,MP)+SE(N,MP)+E)
      H2=0.5*(D(N,M)+SE(N,M)+D(NP,M)+SE(NP,M)+E)
C        PRINT OTHER VARIABLES, LIKE DEPTH AND BOTTOM STRESSES
      FAH=AH(N,M)/(D(N,M)+SE(N,M))
      UO=UHOLD(N,M)/H1
      VO=VHOLD(N,M)/H2
      UBOT=U(LBOT,N,M)
      VBOT=V(LBOT,N,M)
      LMID=MAX0(1,LBOT/2)
      PHIX=.5*(PHI(N,M)+PHI(N,M+1))
      PHIY=.5*(PHI(N,M)+PHI(N+1,M))
      WRITE(IO,250)SE(N,M),UE(N,M),VE(N,M),D(N,M),TBX(N,M),TBY(N,M),
     1 QI,QA,QB,QS,QE,
     2 GSTARX(N,M),GSTARY(N,M),AREA(N,M),BX(N,M),BY(N,M),
     3 PHI(N,M),UO,VO,SOLD(N,M),FAH,
     4 THETA1(N,M),THETA2(N,M),THETA3(N,M),
     5 THETSU(N,M),THETSV(N,M),FEDGE(N,M),
     6 UBOT,VBOT,LMID,W(LMID,N,M),TSX(N,M),TSY(N,M),
     7 DPADX,DPADY,SE(N,MP),SE(NP,M),AHC(N,M),PHIX,PHIY
 250  FORMAT(3X,' SE=',F8.5,' UE,VE=',2F8.5,
     1 ' D=',F7.2,' TBX,Y=',2E11.4,/,3X,' QI=',E9.3,
     2 ' QA=',E9.3,' QB=',E9.3,' QS=',E9.3,' QE=',E9.3,
     3 /,3X,' GSTARX,Y=',2E11.4,' AREA=',F5.3,
     4 ' BX=',F5.3,' BY=',F5.3,/,3X,' PHI=',F7.5,
     5 ' UHOLD/D=',F7.3,' VHOLD/D=',F7.3,' SOLD=',F6.3,' AH/D=',F9.3,
     6 /,3X,' THETA1,2,3=',3F7.3,' THETSU,V=',2E10.3,' EDG=',F3.1,/,
     7 3X,' U,VBOT=',2E11.4,' W(',I2,')=',E10.4,' TSX,Y=',2E9.2,
     8 /,3X,' DPADX,Y=',2E9.2,' SEmp,SEnp=',F7.4,F8.4,' AHC=',F9.3,
     9 /,3x,' PHIX,Y=',2F10.7)
 440  CONTINUE
 450  RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE PRINTX
C        MAY 1984   K. W. HESS   MEAD   VAX11/750
C        PURPOSE - TO PRINT OUT THE TOP VIEWS OF EXTERNAL VARIABLES
C        VARIABLES -
C                   I = VARIABLE INDEX
C                       1 = WATER LEVEL
C                       2 = U MEAN VELOCITY
C                       3 = V MEAN VELOCITY
C                       4 = TOTAL SURFACE X VELOCITY
C                       5 = TOTAL SURFACE Y VELOCITY
C                       6 = SURFACE SALINITY
C                       7 = BOTTOM SALINITY
C                       8 = SURFACE TEMPERATURES  (C)
C                       9 = BOTTOM TEMPERATURES  (C)
C                      10 = HORIZONTAL VISCOSITY
C                      11 = VERTICAL VISCOSITY
C                      12 = WX
C                      13 = WY
C                 JPRNT() = PRINT INDEX: IF=1 , PRINT
      INCLUDE 'COMM20.FOR'
      DIMENSION NUM(NSIZE),NO(NSIZE),JFIELD(NSIZE)
      CHARACTER*1 ANUM(3*NSIZE)
      CHARACTER*43 QTITLE(13)
      DATA QTITLE/
     1 'WATER LEVELS (CM)                          ',
     2 'X-DIR VERTICALLY-AVERAGED VELOCITIES (CM/S)',
     2 'Y-DIR VERTICALLY-AVERAGED VELOCITIES (CM/S)',
     3 'X-DIR TOTAL SURFACE VELOCITIES (CM/S)      ',
     3 'Y-DIR TOTAL SURFACE VELOCITIES (CM/S)      ',
     3 'SURFACE SALINITIES (PPT)                   ',
     4 'BOTTOM SALINITIES (PPT)                    ',
     5 'SURFACE TEMPERATURES (DEG. C)              ',
     5 'BOTTOM TEMPERATURES (DEG. C)               ',
     5 'HORIZONTAL VISCOSITY (M**2/S)              ',
     5 'VERTICAL VISCOSITY x 100 (M**2/S )         ',
```

```
     5 'WX (M/S)                                       ',
     5 'WY (M/S)                                       '/
      DIMENSION LEV(13),ISEL(13),FAC(13)
      DATA LEV/1,1,1,1,1,1,1,1,1,1,1,1,1/
      DATA FAC/100.,100.,100.,100.,100.,1.,1.,1.,1.,1.,100.,1.,1./
      DATA ISEL/1,2,3,4,5,9,9,10,10,11,15,18,19/
      LEV(7)=LBOT-1
      LEV(9)=LBOT-1
      LEV(11)=LBOT/2
      N3MAX=3*NMAX
      DO 100 N=1,NMAX
 100  NO(N)=N
C        LOOP THRU THE VARIABLES
      ICALL=0
      DO 250 I=1,13
      IF(JPRNT(I).NE.1)GOTO 250
C        PRINT PAGE FORM
      IF(KPRNT2.NE.2)WRITE(IO,110)
 110  FORMAT(1X)
      IF(KPRNT2.EQ.2)WRITE(IO,120)
 120  FORMAT(1H1)
C        WRITE DESCRIPTOR LINE
      ICALL=ICALL+1
      IF(ICALL.EQ.1)WRITE(IO,125)
 125  FORMAT(/,1X,'B. TOP VIEW OF SELECTED VARIABLE FIELDS')
c     WRITE(IO,130)QTITLE(I),UT,NSTI,HR,HR1,CUMDAY
c130  FORMAT(2X,A43,/,1X,'UT=',F10.4,' NSTI=',I5,'  HR=',F8.2,' CUM.HR='
      RELDAY=HR/24.
      WRITE(IO,130)QTITLE(I),YEAR,UT,RELDAY,CUMDAY,NSTI
 130  FORMAT(2X,A43,/,1X,'YR=',F5.0,' UT=',F10.4,
     1 ' REL.DAY=',F8.2,' CUM.DAY=',F8.2,' NSTI=',I5)
      NPL=25          ! number per line
      KMAX=1+(NMAX-1)/NPL
      DO 220 K=1,KMAX
      N1=1+NPL*(K-1)
      N2=MIN0(NMAX,N1+NPL-1)
      WRITE(IO,134)(NO(N),N=N1,N2)
 134  FORMAT(4X,40I3,3(/5X,40I3))
C       LOOP THRU ARRAY AND GET VALUE
      DO 180 M=1,MMAX
      DO 136 N=1,NMAX
 136  NUM(N)=0
      DO 140 N=N1,N2
      JFIELD(N)=IFIELD(N,M)
 140  NUM(N)=NINT(FAC(I)*SELECT(ISEL(I),LEV(I),N,M))
C        OPTION 1: PRINT OUT VECTOR AS STRAIGHT NUMBERS
      IF(KPRNT1.EQ.2)GOTO 160
      WRITE(IO,150)M,(NUM(N),N=N1,N2)
 150  FORMAT(1X,I3,40I5,3(/4X,40I5))
      GOTO 180
C        OPTION 2: PRINT OUT VECTORS AS CHARACTERS, WITH "." FOR LAND
 160  CALL PRNCHR(3,NUM,JFIELD,ANUM,NMAX)
      N1A=1+3*(N1-1)
      N2A=MIN0(3*NMAX,N1A-1+3*NPL)
      WRITE(IO,170)M,(ANUM(N),N=N1A,N2A)
 170  FORMAT(1X,I3,140A1,3(/5X,140A1))
 180  CONTINUE
 220  CONTINUE
 250  CONTINUE
      RETURN
      END
C
C-------------------------------------------------------------------
C
      SUBROUTINE PRNCHR(MAXDIG,NUM,JFIELD,ANUM,KMAX)
C        OCTOBER 1984   K. W. HESS   MEAD   VAX11/750
C        PURPOSE - TO CONVERT NUMERIC VALUES TO CHARACTER VALUES.
C                  IT'S ASSUMED THAT THE OUTPUT NUMBER IS AT MOST
C                  3 (=MAXDIG) DIGITS LONG.
      INCLUDE 'COMM20.FOR'
      DIMENSION NUM(NPMSIZ),IMAX(4),JFIELD(NPMSIZ)
      CHARACTER*1 ANUM(3*NPMSIZ)
      CHARACTER*1 DIGIT(10),CHAR(5)
      DATA DIGIT/'0','1','2','3','4','5','6','7','8','9'/
      DATA CHAR/' ','0','-','*','.'/
      DATA IMAX/9,99,999,9999/
C        LOOP THRU THE NUMBERS
      DO 200 N=1,KMAX
      NO=NUM(N)
      IS=1
      IF(NO.LT.0)IS=-1
      NO=NO*IS
      IF(JFIELD(N).GT.9)GOTO 110
C        LAND GRID SQUARES
      DO 100 I=1,MAXDIG
      II=I+MAXDIG*(N-1)
      ANUM(II)=CHAR(1)
      IF(I.EQ.MAXDIG)ANUM(II)=CHAR(5)
 100  CONTINUE
      GOTO 200
C        NUMBERS TOO LARGE: PRINT ****
 110  IF(NO.LE.IMAX(MAXDIG).AND.IS*NO.GT.
     1  -IMAX(MAXDIG)/10)GOTO 130
      DO 120 I=1,MAXDIG
      II=I+MAXDIG*(N-1)
 120  ANUM(II)=CHAR(4)
      GOTO 200
C        WATER GRID SQUARES: FIRST INSERT BLANKS
 130  DO 140 I=1,MAXDIG
      II=I+MAXDIG*(N-1)
 140  ANUM(II)=CHAR(1)
C        INSPECT GRIDS FROM LEFT TO RIGHT
      DO 150 J=1,MAXDIG
      I=MAXDIG+1-J
      II=I+MAXDIG*(N-1)
      IDIG=(NO-10**(MAXDIG+1-I)*(NO/(10**(MAXDIG+1-I))))
     1  /10**(MAXDIG-I)
      ANUM(II)=DIGIT(IDIG+1)
C           LEADING BLANKS
      IF(NO.GT.10**J-1.OR.J.EQ.MAXDIG)GOTO 150
      ANUM(II-1)=CHAR(1)
C           SIGN CHARACTER
      IF(IS.LT.1)ANUM(II-1)=CHAR(3)
      GOTO 200
 150  CONTINUE
 200  CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE PRNCON(INDEX0)

C         PURPOSE - TO PRINT MOST OF THE RUN PARAMETERS IN A COMPACT FOR
C         VARIABLES -
C                 INDEX0 = PARAMETER TO SPECIFY PRINTOUT OF SECONDARY
C                          VARIABLES LIKE DEPTHS (0=NO, 1=YES).  USUALLY
C                          INDEX0=1 AT START OF RUN, 0 LATER.
      INCLUDE 'COMM20.FOR'
      COMMON/A1/DTIMAX
      DIMENSION FRICH(10),FRR(10),FRT(10),NUM(101)
      DIMENSION JFIELD(40),ANUM(120)
      CHARACTER*2 TITLY(3)
      CHARACTER*2 CHNUM(2)
      CHARACTER*1 ANUM
      DATA CHNUM/'I.','VI'/,TITLY/'D.','  ','  '/
      JN=1
      IF(INDEX0.EQ.0)JN=2
      WRITE(IO,70)VERS
 70   FORMAT(1X,'MECCA2 (MODEL FOR ESTUARINE AND COASTAL CIRCULATION',
     1 ' ASSESSMENT), VERSION ',F5.2)
      WRITE(IO,65)CHNUM(JN)
 65   FORMAT(1X,A2,' MODEL CONSTANTS',60(1H-))
C        INPUT FILE NAMES
      IF(INDEX0.EQ.0)GOTO 85
      WRITE(IO,80)FGEO
 80   FORMAT(/,1X,'A. INPUT GEOGRAPHY FILE:',A40)
 85   CONTINUE
C        PRINT OUT INTERNAL (SET IN THE CODE) PARAMETERS
      WRITE(IO,90)AG,ALV,CPAIR,CWATER,
     1 E,EPSLON,IO,ISCR,OMEGA,PA,PI,RHOA,RHOW,SB,SOLAR,VONKAR
 90   FORMAT(/,1X,'B. INTERNAL PARAMETER VALUES:',/,' AG=',F6.3,' ALV=',
     1 E10.4,' CPAIR=',F6.1,' CWATER=',F6.1,' E=',E10.4,/,
     3 ' EPSLON=',F5.3,' IO=',I2,' ISCR=',I2,' OMEGA=',E9.3,' PA=',
     4 F8.1,' PI=',F9.7,' RHOA=',F5.3,/,' RHOW=',F6.1,' SB=',E9.3,
     5 ' SOLAR=',F6.1,' VONKAR=',F4.2)
C        INPUT DATA VALUES
      WRITE(IO,100)
     1 AH00,AH0,ALB,AV00,AV0,BSNANG,BSNLAT,BSNLON,CAH,
     2 CDRGWS,CDWB1,CDWB2,CDR1,CDR2,CLOUD,CR0,(CRICH(I),I=1,8),
     3 DL,DTE,DV0,DV00,D10PCT,DHAH,HR0,HRCONC,HRMAX,HROUT,
     4 HROUT0,IBETAA,IBETAH,IBETAP,IBOTV,ICOR,ICOUPL,ICPOS,IEXTRN,IGPH,
     6 IHEAT,IHVISC,INTER,ITOPV,IVISC,(JPRNT(I),I=1,13),KOCNBC,
     7 KONCEN,LAYRS,MCOR,MMAX,NCOR,NMAX,NUMOBC,NUMRIV,NUMBXY
 100  FORMAT(/,1X,'C. INPUT VALUES:',/,1X,'C.1 CONFIGURATION DATA',/,
     1 ' AH00=',F5.1,' AH0=',F5.1,' ALB=',F4.2,
     2 ' AV00=',F7.5,' AV0=',F7.5,' BSNANG=',F7.3,/,' BSNLAT=',F7.3,
     3 ' BSNLON',F8.3,' CAH=',F5.3,' CDRGWS=',F6.4,' CDWB1=',F6.4,
     4 ' CDWB2=',F6.4,/,' CDR1=',F8.6,' CDR2=',F8.6,' CLOUD=',F4.2,
     4 ' CN=',F4.2,
     5 ' CRICH1=',E10.4,/,' CRICH2=',E10.4,' CRICH3=',E10.4,' CRICH4=',
     6  E10.4,' CRICH5=',E10.4,/,' CRICH6=',E10.4,' CRICH7=',E10.4,
     7 ' CRICH8=',E10.4,' DL=',F8.1,' DTE=',F8.2,/,' DV00=',F8.6,
     7 ' DV0=',F8.6,' D10PCT=',F4.1,' DHAH=',F6.2,
     8 ' HR0=',F6.1,' HRCONC=',F6.1,/,' HRMAX=',F6.1,' HROUT=',F6.2,
     9 ' HROUT0=',F6.1,' IBETAA=',I1,' IBETAH=',I1,' IBETAP=',I1,
     1 ' IBOTV=',I1,/,' ICOR=',I1,' ICOUPL=',I1,' ICPOS=',I1,
     2 ' IEXTRN=',I1,' IGPH=',I2,' IHEAT=',I2,' IHVISC=',I2,
     3 ' INTER=',I2,' ITOPV=',I1,/,' IVISC=',I1,
     2 ' JPRNT=',13I2,' KOCNBC=',I2,
     3 ' KONCEN=',I1,' LAYRS=',I2,' MCOR=',I3,/,' MMAX=',I3,' NCOR='
     4 ,I3,' NMAX=',I3,' NUMOBC=',I2,
     7 ' NUMRIV=',I2,' NUMBXY=',I4)
C        FORMAT VARIABLES
      IF(INDEX0.EQ.0)GOTO 245
      WRITE(IO,180)(JPRNT(I),I=1,13),KPRNT1,KPRNT2,NPRMN
 180  FORMAT(/,1X,'C.2 PRINT VARIABLES',/,1X,'JPRNT=',13I2,
     1 '  KPRNT1,2=',2I5,/,1X,'P/O AT SELECTED CELLS: NPRMN=',I3)
      IF(NPRMN.GT.0)WRITE(IO,190)(IPRMN(J),J=1,NPRMN)
 190  FORMAT(1X,' IPRMN=',20I7)
      IF(ISLICE.EQ.0)GOTO 220
      WRITE(IO,195)ISLICE
 195  FORMAT(1X,'SLICE PRINT SECTIONS: ISLICE=',I2)
      DO 200 I=1,ISLICE
      JS=JSLICE(I)
      DO 197 J=1,JS
 197  NUM(J)=1000*MSLICE(J,I)+NSLICE(J,I)
 200  WRITE(IO,210)JS,(NUM(J),J=1,JS)
 210  FORMAT(1X,' JSLICE=',I2,' M,NSLICE=',5I7,/,10I7,/,10I7)
      WRITE(IO,212)IGPH,NSTGPH,IGPHOP
 212  FORMAT(1X,'VARIABLES SAVED FOR GRAPHING: IGPH=',I2,
     1 ' NSTGPH=',I3,' IGPHOP=',i1)
      I=1
      IF(IGPH.GT.0)WRITE(IO,216)(I,LGPH(I),MGPH(I),NGPH(I),
     1 ITYP(I),PTITLE(ITYP(I)),I=1,IGPH)
 216  FORMAT(11X,I2,':     L=',I2,'  M=',I3,'  N=',I3,'  ITYP=',I2,
     1 '    (',A10,')')
 220  CONTINUE
 245  CONTINUE
C
C        WRITE THE SECONDARY PARAMETERS
      AHMAX=DL**2/(8.*DTE)
C       CHECK MAXIMUM TIME STEP FOR EXTERNAL, INTERNAL MODES
      DTEMAX=DL/SQRT(DMAX*AG)
      DHMAX=DHAH*AHMAX/FLOAT(ISPLIT)
      WRITE(IO,102)COR,DENRAT,DFDM,DFDN,DMAX,DMIN,DQ,
     1 DTI,IHR,KRIVBC,NBCEL,NCELL,NSTET,NSTIT,NSTIMX,RAD
 102  FORMAT(/,1X,'D. PARAMETERS COMPUTED FROM INPUT DATA:',/,
     1 ' COR=',F8.6,' DENRAT=',F7.5,' DFDM=',E8.2,
     2 ' DFDN=',E8.2,/,' DMAX=',F8.2,' DMIN=',F8.2,' DQ=',F5.3,
     3 ' DTI=',F7.1,' IHR=',I3,' KRIVBC=',I1,/,
     4 ' NBCEL=',I3,' NCELL=',I6,
     5 ' NSTET=',I9,' NSTIT=',I9,' NSTIMX=',I5,' RAD=',F10.8)
      SUM1=0.0
      DO 115 L=1,LBOT
 115  SUM1=SUM1+CI(L)
      WRITE(IO,120)(CI(L),L=1,2   ),SUM1
 120  FORMAT(1X,'CI(1)=',F6.4,' CI(2)=',F6.4,' SUM OF LBOT CIs= ',F4.2)
C
      IF(INDEX0.EQ.0)GOTO 300
      DTEMAX=DL/SQRT(2.*DMAX*AG)
      DTIMAX=(DMIN*DQ)**2/(2.*AV00)
      F=3600./DTI
      WRITE(IO,160)DTEMAX,DTIMAX,F
 160  FORMAT(/,1X,'E. CHECKS',/,
     2  2X,'TIMESTEPS: DTEMAX= DL/SQRT(DMAX*AG)=    ',F11.2,/,
     3  13X,'DTIMAX=(DMIN*DQ)**2/(8*AV00)=',F10.2,/
     4  13X,'IHR (STEPS PER HOUR)=3600/DTI=',F12.6)
      WRITE(IO,166)KONCEN,ICOUPL,IBETAP
 166  FORMAT(2X,'DENSITY: KONCEN=',I1,' ICOUPL=',I1,' IBETAP=',I1)
 170  CONTINUE
 300  CONTINUE
 700  CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE PRNBCG(INDEX)

C         PURPOSE - TO PRINT MOST OF THE RUN PARAMETERS IN A COMPACT FOR
C         VARIABLES -
```

```
C                  INDEX = PARAMETER TO SPECIFY PRINTOUT OF SECONDARY
C                          VARIABLES LIKE DEPTHS (0=NO, 1=YES).  USUALLY
C                          INDEX=1 AT START OF RUN, 0 LATER.
      INCLUDE 'COMM20.FOR'
      DIMENSION TITLX(3),DAVG(NPMSIZ),NUMM(NPMSIZ),
     1 NUMN(NPMSIZ),NMBRM(MSIZE),NMBRN(NSIZE),ICELL(5),JFIELD(NSIZE)
      CHARACTER*10 TITLX
      CHARACTER*2 TITLY(3)
      CHARACTER*1 ANUM(3*NSIZE)
      DATA TITLY/'D.',' ',' '/
      DATA TITLX/'AREA*100. ','BX*100.   ','BY*100.   '/
      DATA ICELL/5*0/
C        PRINT INPUT DATA
      IF(INDEX.NE.1)RETURN
      WRITE(IO,100)
  100 FORMAT(//,1X,'II. TIME-VARIABLE INPUT',55(1H-),/1X,
     1 'A. REFERENCE DATE ')
C        ENVIRONMENTAL DATA
  110 WRITE(IO,120)IYEAR,MONTH,IDAY,IHOUR,IMIN,UT0
  120 FORMAT(1X,'STARTING DATE: IYEAR=',I4,' MONTH=',I2,' IDAY=',I2,
     1 ' IHOUR=',I4,' IMIN=',I3,'  UTC0=',F10.4)
C        PRINT WATER LEVEL DATA (TIDES)
      WRITE(IO,150)NSIGT
  150 FORMAT(/1X,'B. OCEAN WATER LEVEL DATA: NSIGT=',I2)
      IF(NSIGT.GT.0)THEN
      DO I=1,2
      WRITE(IO,180)I,YTID(I),DTID(I),(TDLEV(I,N),N=1,NSIGT)
  180 FORMAT(5X,'RECORD=',I2,F8.1,F10.4,20F10.2)
      ENDDO
      ENDIF
C        PRINT WINDS AND AIR TEMPERATURE
      WRITE(IO,170)NSIGW
  170 FORMAT(/1X,'C. WIND DATA: NSIGW=',I2)
C        PRINT RIVER FLOW
      WRITE(IO,190)NSIGR
  190 FORMAT(/1X,'D. RIVER FLOW RATE DATA: NSIGR=',I2)
      IF(NSIGR.GT.0)THEN
      DO I=1,2
      WRITE(IO,180)I,YRIV(I),DRIV(I),(QRIV(I,N),N=1,NSIGR)
      ENDDO
      ENDIF
C        PRINT OCEAN BOUNDARY SALINITY
      WRITE(IO,210)NSIGS
  210 FORMAT(/,1X,'E. OCEANIC BOUNDARY SALINITY: NSIGS=',I2)
      IF(NSIGS.GT.0)THEN
      DO I=1,2
      WRITE(IO,180)I,YSAL(I),DSAL(I),(SALOCN(I,N),N=1,NSIGS)
      ENDDO
      ENDIF
C        PRINT OCEAN BOUNDARY TEMPERATURE
      WRITE(IO,230)NSIGTO
  230 FORMAT(/,1X,'F. OCEANIC BOUNDARY TEMPERATURE: NSIGTO=',I2)
      IF(NSIGTO.GT.0)THEN
      DO I=1,2
      WRITE(IO,180)I,YOTP(I),DOTP(I),(TMPOCN(I,N),N=1,NSIGTO)
      ENDDO
      ENDIF
C        PRINT RIVER BOUNDARY TEMPERATURE
      WRITE(IO,250)NSIGRT
  250 FORMAT(/,1X,'G. RIVER BOUNDARY TEMPERATURE: NSIGRT=',I2)
      IF(NSIGRT.GT.0)THEN
      DO I=1,2
      WRITE(IO,180)I,YRVT(I),DRVT(I),(TRIV(I,N),N=1,NSIGRT)
      ENDDO
      ENDIF
C        PRINT ADDITIONAL MET DATA
      WRITE(IO,260)NSIGM
  260 FORMAT(/,1X,'H. ADDITIONAL MET DATA: NSIGM=',I2)
      IF(NSIGM.GT.0)THEN
      DO I=1,2
      WRITE(IO,180)I,YMET(I),DMET(I),(VMET(I,N),N=1,NSIGM)
      ENDDO
      ENDIF

c         BOUNDARY LOCATIONS
      IF(NUMOBC+NUMRIV.GT.0)WRITE(IO,410)
  410 FORMAT(/,1X,'BOUNDARY SPECIFICATIONS IN GEO FILE')
      IF(NUMOBC.GT.0)THEN
      DO 420 N=1,NUMOBC
  420 WRITE(IO,430)N,MB1(N),MB2(N),NB1(N),NB2(N),ITPO(N),JTPO(N),
     1 ISET1(N),ISET2(N)
  430 FORMAT(1X,'OCEAN BND. ',I2,'  MB1,2=',2I4,'  NB1,2=',2I4,
     1 ' ITPO=',I2,' JTPO=',I2,' ISET1,2=',2i4)
      ENDIF
      IF(NUMRIV.GT.0)THEN
      DO 440 N=1,NUMRIV
  440 WRITE(IO,450)N,MR1(N),MR2(N),NR1(N),NR2(N),ITPR(N),JTPR(N),
     1 ISETR(N)
  450 FORMAT(1X,'RIVER BND. ',I2,'  MR1,2=',2I4,'  NR1,2=',2I4,
     1 ' ITPR=',I2,' JTPR=',I2,' ISETR=',I2)
      ENDIF
C        CATEGORIES OF IFIELD VARIABLES
  480 DO 490 N=1,NMAX
  490 NMBRN(N)=N
      DO 492 M=1,MMAX
  492 NMBRM(M) = M
      DO 495 M=1,MMAX
      DO 495 N=1,NMAX
      IF(IFIELD(N,M).LT.10)ICELL(1)=ICELL(1)+1
      IF(IFIELD(N,M)/10.EQ.1)ICELL(2)=ICELL(2)+1
      IF(IFIELD(N,M)/10.EQ.2)ICELL(3)=ICELL(3)+1
      IF(IFIELD(N,M)/10.EQ.KOCNBC)ICELL(4)=ICELL(4)+1
      IF(IFIELD(N,M)/10.EQ.KOCNBC+1)ICELL(5)=ICELL(5)+1
  495 CONTINUE
      ISUM=0
      DO 496 I=1,5
  496 ISUM=ISUM+ICELL(I)
      JSUM=NMAX*MMAX
C        PRINT IFIELD
      KP1=KOCNBC+1
      WRITE(IO,500)KOCNBC,KP1,BSNANG,BSNLAT,BSNLON,DL,MCOR,NCOR
  500 FORMAT(//,1X,'III. GEOGRAPHY DATA',60(1H-),//,
     1 1X,'A. IFIELD CELL CODES:',/,4X,
     2' LAND=0  TRIANGULAR=10  WATER=20  OCEAN BND=',I1,'0  RIVER BND=',
     3 I1,'0',/,1X,'    WATER & NO X-FLOW=21',
     4 4X,' WATER & NO Y-FLOW=22    WATER & NO X- OR Y-FLOW=23',/,
     5 1X,'    BSNANG=',F8.3,' BSNLAT=',F8.3,' BSNLON=',F8.3,' DL=',
     6 F9.2,' MCOR=',I2,' NCOR=',I2)
      WRITE(IO,510)(ICELL(J),J=1,5),ISUM,JSUM
  510 FORMAT(/,1X,'CELL COUNTS: LAND=',I6,/,2X,'WATER:TRIANGULAR=',I6,/,
     1 6X,'WATER:SQUARE=',I6,/,9X,'OCEAN BND=',I6,/,9X,'RIVER BND=',I6,
     2 /,10X,'CELL SUM=',I6,'  VS. NMAX*MMAX=',I6)
C
C        PRINT DEPTHS
  530 NPERL=24
      KMAX=1+(NMAX-1)/NPERL
      WRITE(IO,531)
  531 FORMAT(/,1X,'C. DEPTHS (M) AT MSL')
      DO 563 K=1,KMAX
      N1=1+(K-1)*NPERL
      N2=MIN0(NMAX,N1+NPERL-1)
      WRITE(IO,532)(NMBRN(N),N=N1,N2)
  532 FORMAT(2X,' M N=',I2,30I3)
      DO 550 M=1,MMAX
      DO 540 N=N1,N2
      NUMN(N)=-1
  540 IF(IFIELD(N,M).GT.0)NUMN(N)=D(N,M)
      WRITE(IO,560)M,(NUMN(N),N=N1,N2)
  560 FORMAT(1X,I3,2X,30I3)
  550 CONTINUE
  563 CONTINUE

C        COMPUTE AVERAGE DEPTHS
      DO 570 M=1,MMAX
      NUMN(M)=M
      NN=0
      SUM=0.
      DO 565 N=1,NMAX
      IF(D(N,M).LT.E)GOTO 565
      NN=NN+1
      SUM=SUM+D(N,M)
  565 CONTINUE
  570 DAVG(M)=SUM/(E+FLOAT(NN))
      WRITE(IO,580)
  580 FORMAT(/,1X,'MEAN OF NON-ZERO DEPTHS')
      NPL=5
      KMAX=1+(MMAX-1)/NPL
      DO 600 K=1,KMAX
      M1=1+(K-1)*NPL
      M2=MIN0(MMAX,M1+NPL-1)
  595 WRITE(IO,591)(NUMN(M),DAVG(M),M=M1,M2)
c591 FORMAT(1X,4('  ','M=',I3,' D=',F8.2,'  '))
  591 FORMAT(1X,'M,D=',5(I3,F8.2,2x))
  600 CONTINUE

C        VARIABLE WIDTHS
      IF(NUMBXY.LE.0)GOTO 660
      N3MAX=3*NMAX
      DO 650 J=1,3
      WRITE(IO,610)TITLY(J),TITLX(J)
  610 FORMAT(/,1X,A2,' VARIABLE-WIDTH PARAMETERS: ',A10)
      NPL=25          ! number per line
      KMAX=1+(NMAX-1)/NPL
      DO 640 K=1,KMAX
      N1=1+NPL*(K-1)
      N2=MIN0(NMAX,N1+NPL-1)
      WRITE(IO,612)(NMBRN(N),N=N1,N2)
  612 FORMAT(4x,40I3)
      DO 630 M=1,MMAX
      DO 620 N=N1,N2
      IF(J.EQ.1)F1=AREA(N,M)
      IF(J.EQ.2)F1=BX(N,M)
      IF(J.EQ.3)F1=BY(N,M)
      JFIELD(N)=IFIELD(N,M)
  620 NUMN(N)=F1*100.
      CALL PRNCHR(3,NUMN,JFIELD,ANUM,NMAX)
      N1A=1+3*(N1-1)
      N2A=MIN0(3*NMAX,N1A-1+3*NPL)
      WRITE(IO,635)M,(ANUM(N),N=N1A,N2A)
  635 FORMAT(1X,I3,120A1,3(/5X,120A1))
  630 CONTINUE          ! end mmax
  640 CONTINUE          ! end k
  650 CONTINUE          ! end j
  660 CONTINUE
C        PRINT FLAGS
      DO 670 N=1,NSIZE
  670 NUMN(N)=N
C        WRITE OUT THE TAGS
      NX=MAX0(1,NCOL,NROW)
      WRITE(IO,675)
  675 FORMAT(/,1X,'E. ROW/COLUMN FLAGS:',/,6X,'B.C. FOR IL/IR: 0=WATER',
     1 ' LEVEL,  1=FLOWRATE,  2=ZERO FLOW',/,12X,2('ICOL',14X,'IROW'
c    2 ,18X),/,2X,2(' I',(4X,'N MA MB IL IR',4X,' M NA NB IL IR',3X)))
     2 ,18X),/,2X,2(' I',(4X,'N  MA  MB L R',4X,' M  NA  NB L R',3X)))
      NXM=1+(NX-1)/2
      DO 680 N=1,NXM
      M=N+NXM
  680 WRITE(IO,690)N,(ICOL(K,N),K=1,5),(IROW(K,N),K=1,5),M,
     1 (ICOL(K,M),K=1,5),(IROW(K,M),K=1,5)
  690 FORMAT(1X,2(I3,':',3I4,2I2,2X,3I4,2I2,2X))
C        PRINT OUT NEW IFIELD (WITH BARRIERS)
      NPL=30          ! number per line
      KMAX=1+(NMAX-1)/NPL
      DO 730 K=1,KMAX
      N1=1+NPL*(K-1)
      N2=MIN0(NMAX,N1+NPL-1)
      WRITE(IO,692)(NUMN(I),I=N1,N2)
  692 FORMAT(/,1X,'F. MODIFIED IFIELD',/,1X,'  M  NA  NB  ',40I2,
     1         3(/10X,40I2))
      DO 700 M=1,MMAX
      NA=NAB(M)/1000
      NB=MOD(NAB(M),1000)
  700 WRITE(IO,710)M,NA,NB,(IFIELD(I,M),I=N1,N2)
  710 FORMAT(1X,I3,2I4,2X,40I2,3(/14X,40I2))
  730 CONTINUE

      DO 780 K=1,KMAX
      N1=1+NPL*(K-1)
      N2=MIN0(NMAX,N1+NPL-1)
      WRITE(IO,750)(NUMN(I),I=N1,N2)
  750 FORMAT(/,1X,'MFLUX',/,6X,40I2)
      DO 760 M=1,MMAX
  760 WRITE(IO,770)M,(MFLUX(I,M),I=N1,N2)
  770 FORMAT(1X,I3,2X,40I2)
  780 CONTINUE

      DO 820 K=1,KMAX
      N1=1+NPL*(K-1)
      N2=MIN0(NMAX,N1+NPL-1)
      WRITE(IO,800)(NUMN(I),I=N1,N2)
  800 FORMAT(/,1X,'NFLUX',/,6X,40I2)
      DO 810 M=1,MMAX
  810 WRITE(IO,770)M,(NFLUX(I,M),I=N1,N2)
  820 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE PSLICE
C        JANUARY 1985   K. W. HESS   MEAD   VAX 11/750
C        PURPOSE - PRINT ALL CONCENTRATIONS OVER DEPTH AT ALL GRID CELLS
```

```
C                 ALONG A SLICE OF THE DOMAIN.  UP TO
C                 5 SLICES ALLOWED, EACH WITH UP TO 10 POINTS.
C        VARIABLES -
C              ISLICE = NO. OF SLICES
C               ITYPE = TYPE OF VARIABLE
C           JSLICE(I) = NO. OF POINTS IN I-TH SLICE
C         MSLICE(I,J) = J-TH VALUE OF THE M-GRID POINT ON THE I-TH SLICE
      INCLUDE 'COMM20.FOR'
      DIMENSION NO(NM2SIZ),MO(NM2SIZ),NDAT(LSIZE,NM2SIZ),II(NM2SIZ),
     1    PSE(NM2SIZ),DP(NM2SIZ),F(LSIZE,NM2SIZ),IITYPE(8)
      DATA IITYPE/4,5,6, 9,10, 22,15,16/ ! UT,VT,W, S,T, AH3,AV,DV
      CHARACTER*5 ANUM(NM2SIZ),BNUM(NM2SIZ)
      DO 100 N=1,NM2SIZ
      II(N)=N
      ANUM(N)='====='
  100 BNUM(N)='-----'
C         LOOP THRU DATA
      IF(ISLICE.LE.0)RETURN
C        LOOP THRU THE SLICES
      DO 300 IS=1,ISLICE
C        CONSTRUCT THE SLICE
      K=0
      DO 130 J=1,JSLICE(IS)-1
      NA=NSLICE(J,IS)
      MA=MSLICE(J,IS)
      NB=NSLICE(J+1,IS)
      MB=MSLICE(J+1,IS)
C        GET DIRECTION (INCREASING OR DECREASING)
      MDIR=ISIGN(1,MB-MA)
      NDIR=ISIGN(1,NB-NA)
      IF(NB.EQ.NA)NDIR=0
      IF(MB.EQ.MA)MDIR=0
C        LINE:  save m, n, and depth
      NN=NA-NDIR
      MM=MA-MDIR
  110 MM=MM+MDIR
      NN=NN+NDIR
      IF(K.GT.1.AND.(MM.EQ.MO(K).AND.NN.EQ.NO(K)))GOTO 110
      K=K+1
      DP(K)=D(NN,MM)
      PSE(K)=SE(NN,MM)
      MO(K)=MM
      NO(K)=NN
      IF(MM.EQ.MB.AND.NN.EQ.NB)GOTO 130
      GOTO 110
  130 CONTINUE
      KMAX=K
C        LOOP THRU THE VARIABLES
      DO 270 IT=1,8
      LEVS=LBOT
      IF(IITYPE(IT).EQ.6)LEVS=LAYRS
      IF(IITYPE(IT).GE.15.AND.IITYPE(IT).LE.17)LEVS=LAYRS
      FMAX=0.
      DO 140 L=1,LEVS
      DO 140 K=1,KMAX
      F(L,K)=SELECT(IITYPE(IT),L,NO(K),MO(K))
  140 FMAX=AMAX1(ABS(F(L,K)),FMAX)
c        scale
      I1=0
      IF(FMAX.GT.0)I1=INT(ALOG10(FMAX)+10.)-13
      SCL=10.**I1
      DO 145 L=1,LEVS
      DO 145 K=1,KMAX
  145 NDAT(L,K)=NINT(F(L,K)/SCL)
c        loop thru sets
      KPERL=14
      KSET=(KMAX-1)/KPERL+1
      DO KS=1,KSET
C        WRITE OUT THE HEADER LINE
      IF(KS.EQ.1)WRITE(IO,150)PTITLE(IITYPE(IT)),SCL,UT,FMAX
  150 FORMAT(/,1X,'SECTION : ',A10,' DIVIDED BY ',1PE9.3,'   UT=',
     1 0PF10.4,'  FMAX=',E10.4)
      K1=1+KPERL*(KS-1)
      K2=MIN0(KMAX,K1+KPERL-1)
      WRITE(IO,152)(II(K),K=K1,K2 )
  152 FORMAT(4X,'K=',14I5)
      WRITE(IO,160)(NO(K),K=K1,K2 )
  160 FORMAT(4X,'N=',14I5)
      WRITE(IO,170)(MO(K),K=K1,K2)
  170 FORMAT(4X,'M=',14I5)
c        depths
      WRITE(IO,210)(DP(K),K=K1,K2)
  210 FORMAT(2X,'  D=',14F5.1)
      WRITE(IO,220)(PSE(K),K=K1,K2)
  220 FORMAT(2X,' SE=',14F5.2)
      WRITE(IO,240)(BNUM(K),K=K1,K2)
  240 FORMAT(6X,14A5)
C        RUN DOWN THE COLUMN
      DO 250 L=1,LEVS
  250 WRITE(IO,260)L,(NDAT(L,K),K=K1,K2)
  260 FORMAT(1X,'L=',I2,1X,14I5)
      WRITE(IO,265)ANUM(1),(ANUM(K),K=K1,K2)
  265 FORMAT(X,15A5)
      ENDDO         ! end set
  270 CONTINUE      ! end type
  300 CONTINUE      ! end slice
      RETURN
      END
C
C------------------------------------------------------------
C
      SUBROUTINE PGRAPH
C        OCTOBER 1984   K. HESS   MEAD   VAX 11/750
C        PURPOSE - TO SAVE VARIABLES FOR LATER GRAPHING
C        VARIABLES -
C              IO1,IO2 = CHANNEL NUMBERS FOR OUTPUT (20,21).
C                        (SET IN SUB. INITS)
      INCLUDE 'COMM20.FOR'
      DIMENSION XGR(NDGPH),NUM(NDGPH)
      DATA PTITLE/
     1 'SE        ','UE        ','VE        ','Utotal    ','Vtotal    ',
     2 'W (m/s)   ','TSX*1.E+7 ','TSY*1.E+7 ','S (PPT)   ','T (C)     ',
     3 'AH/H      ','THETA1    ','THETA2    ','THETA3    ','AV        ',
     4 'DV        ','RI        ','WX        ','WY        ','UHOLD/H   ',
     5 'VHOLD/H   ','AH3/H     ','          ','          ','          '/
C        CHECK FOR NUMBER OF GRAPHING VARIABLES
      IF(IGPH.LE.0)RETURN
C        CHECK FOR WRITE:
      IF(.NOT.(NSTI.EQ.0.OR.NSTI.EQ.NSTIMX.OR.NSTI.EQ.NSTGPH*(NSTI/
     1 NSTGPH)))GOTO 210
C        STORE PRESENT VALUES IN ARRAY "XGR"
      DO 110 I=1,IGPH
      IF(I.GT.NDGPH)GOTO 110
      NUM(I)=I
      N=NGPH(I)
      M=MGPH(I)
      L=LGPH(I)
      XGR(I)=SELECT(ITYP(I),L,N,M)
  110 CONTINUE
C        LOOP THRU VARIABLES
      NUMPLN=IGPH
      KMAX=1+(IGPH-1)/NUMPLN
      DO 200 K=1,KMAX
      J1=1+NUMPLN*(K-1)
      J2=MIN0(J1+NUMPLN-1,IGPH)
C          WRITE HEADER: IGXX=FORMAT,  IV=INTVEL, NKOLS=NO. COLUMNS HERE
      IF(NSTI.GT.0.OR.IGPHOP.EQ.0)GOTO 125
      IGXX=3
      NKOLS=1+J2-J1
      WRITE(LUGRF,112)CTITLE,IGPH,NSTIMX,NSTGPH,DTI,YEAR
  112 FORMAT(8A10,/,1X,3I7,F9.3,2F9.3)
      DO 115 J=J1,J2
  115 WRITE(LUGRF,120)J,PTITLE(ITYP(J)),LGPH(J),MGPH(J),NGPH(J)
  120 FORMAT(2X,'COLUMN ',I3,' HAS ',A10,6H AT L=,I2,3H M=,I3,3H N=,I3)
      WRITE(LUGRF,124)(NUM(I),I=J1,J2)
  124 FORMAT(4X,'YEAR    UT',40I11)
C        PRINT THE VALUES (UP TO 40)
  125 WRITE(LUGRF,190)YEAR,UT,(XGR(J),J=J1,J2)
  190 FORMAT(1X,F6.0,F9.4,40(x,F10.4))
C        WRITE NUMBERS AT END
  200 IF(NST.EQ.NSTMAX.AND.IGPHOP.EQ.1)WRITE(LUGRF,124)(NUM(I),
     1 I=J1,J2)
  210 RETURN
      END
C
C------------------------------------------------------------
C
      FUNCTION SELECT(ITYPE,L,N,M)
C        SELECT VARIABLE BASED ON THE GRAPHING TABLE VALUES
c          IGPH ranges from 1 to 21
      INCLUDE 'COMM20.FOR'
      SELECT=0.0
      MP=MIN0(M+1,MMAX)
      NP=MIN0(N+1,NMAX)
      GOTO(100,110,120,130,140,150,160,170,180,190,200,210,
     1 220,230,240,250,260,270,280,290,300,310),ITYPE
      GOTO 400
  100 SELECT=SE(N,M)                !  1
      GOTO 400
  110 SELECT=UE(N,M)
      GOTO 400
  120 SELECT=VE(N,M)                !  3
      GOTO 400
  130 SELECT=UE(N,M)+U(L,N,M)
      GOTO 400
  140 SELECT=VE(N,M)+V(L,N,M)       !  5
      GOTO 400
  150 SELECT=W(L,N,M)
      GOTO 400
  160 SELECT=TSX(N,M)*1.E+7         !  7
      GOTO 400
  170 SELECT=TSY(N,M)*1.E+7
      GOTO 400
  180 SELECT=S(L,N,M)
      GOTO 400
  190 SELECT=T(L,N,M)               ! 10
      GOTO 400
  200 SELECT=AH(N,M)/(D(N,M)+SE(N,M)+E)
      GOTO 400
  210 SELECT=THETA1(N,M)            ! 12
      GOTO 400
  220 SELECT=THETA2(N,M)
      GOTO 400
  230 SELECT=THETA3(N,M)
      GOTO 400
  240 SELECT=AV(L,N,M)*1.E+4               ! 15
      GOTO 400
  250 SELECT=DV(L,N,M)*1.E+4
      GOTO 400
  260 SELECT=RI(L,N,M)
      GOTO 400
  270 SELECT=WX(N,M)                       ! 18
      GOTO 400
  280 SELECT=WY(N,M)
      GOTO 400
  290 SELECT=UHOLD(N,M)/(D(N,M)+SE(N,M)+D(N,MP)+SE(N,MP))   ! 20
      GOTO 400
  300 SELECT=VHOLD(N,M)/(D(N,M)+SE(N,M)+D(NP,M)+SE(NP,M))   ! 21
      GOTO 400
  310 SELECT=AH3(L,N,M)/(D(N,M)+SE(N,M)+E)                  ! 22
  400 CONTINUE
      RETURN
      END
C==================================================================
C      MECCA FILE : EXMODE.FOR
C------------------------------------------------------------------
C
      SUBROUTINE EXMODE
C        JANUARY 1986   K.W. HESS   MEAD   VAX 11/750
C        PURPOSE - TO COMPUTE EXTERNAL-MODE FLOWRATES AND WATER LEVELS
C                  FROM REVISED EQUATIONS FOR VARIABLE GRID WIDTH (TO
C                  MODEL NARROW RIVER WIDTHS).
      INCLUDE 'COMM20.FOR'
c        UPDATE THETA FUNCTIONS
      CALL THETAS
C        SET THE BOUNDARY CONDITIONS
      CALL BNDRY1
C        UPDATE HORIZONTAL VISCOSITY COEFFICIENTS
      CALL HORVIS
C        STORE PREVIOUS WATER LEVELS FOR THE SALINITY AND TEMP. CALCS.
      IF(NSTE.GT.1)GOTO 150
      DO 130 N=1,NMAX
      DO 130 M=1,MMAX
      UHOLD(N,M)=UH(N,M)
      VHOLD(N,M)=VH(N,M)
  130 SOLD(N,M)=SE(N,M)
  150 CONTINUE
C        COMPUTE VELOCITIES
      CALL UHVH
      RETURN
      END
C
C------------------------------------------------------------------
C
      SUBROUTINE BNDRY1
C        OCTOBER 1984   K. W. HESS   MEAD   VAX11/750
C        PURPOSE - TO SET THE EXTERNAL-MODE OPEN BOUNDARY WATER LEVEL OR
C                  FLOWRATE CONDITIONS.
C        VARIABLES -
C                VAL = TIDAL ELEVATION/FLOWRATE AT ENDS OF OCEAN BOUNDARY
C                 FT = TIME PAST PREVIOUS HI/LO (HR)
C                 IS = GRID INCREMENT FOR RADIATION TO SET VELOCITY AT
```

```
C                       OUTSIDE EQUAL TO VELOCITY IN INTERIOR.
C                ITPO = DIRECTION OF OUTFLOW: 1=X, -1=-X, 2=Y, -2=-Y
C                JTPO = TYPE OF OCEAN BOUNDARY CONDITION
C                        1 : WATER LEVEL SPECIFICATION
C                        2 : TRANSPORT OUTFLOW
C                        3 : ORLANSKI RADIATION OUTFLOW (REQUIRES WL)
C                        4 : REIMANN INVARIENT RADIATION OUTFLOW (REQUIRES
C                ITPR = DIRECTION OF RIVER INFLOW: 1=X, -1=-X, 2=Y, -2=-Y
C                JTPR = RIVERINE CONDITION: 1=FLUME, 2=WATER FALLS
C                   J = GRID INCREMENT FOR TIDES TO SET VELOCITY AT
C                       OUTSIDE EQUAL TO VELOCITY IN INTERIOR.
C              NUMOBC = NUMBER OF TIDAL BOUNDARIES
C                QRIV = RIVER FLOWRATE (M**3/S)
      INCLUDE 'COMM20.FOR'
      DIMENSION FINAL(NDTID2),QRATE(NDRIV2)
      C1=DTE/DL
      ITEST=IPRNT1
      IF(NSTI.GT.10.AND.NSTE.GT.1)ITEST=0
      IF(ITEST.GT.0)WRITE(ISCR,100)UT,NSTI,NSTE,NUMOBC,NUMRIV
 100  FORMAT(/,X,'BNDRY1 : UT=',F10.4,' NSTI=',I6,' NSTE=',I2,
     1 ' NUMOBC=',i2,' NUMRIV=',I2)
      RAMPT=AMIN1(1.0,CUMDAY)
      RAMPR=AMIN1(1.0,CUMDAY)
      IF(NUMOBC.LE.0)GOTO 220
C        LOOP THRU OCEANIC BOUNDARIES
      DO 200 IB=1,NUMOBC
C        FIND THE LATEST VALUES FROM INPUT DATA
      CALL RR(YT,ISCR,LUTID,IENDTD,DTID,YTID,NSIGT,NDTID2,TDLEV,FINAL)
      F1=ISIGN(1,ITPO(IB))
      IUH=-(ISIGN(1,ITPO(IB))+1)/2
      ISE=-ISIGN(1,ITPO(IB))
      IDIR=IABS(ITPO(IB))
      IF(ITEST.GT.0)WRITE(ISCR,110)IB,F1,ISE,IUH,IDIR,JTPO(IB),FINAL(IB)
 110  FORMAT(X,'IB=',I2,' F1,ISE,IUH,IDIR,JTPO=',F4.1,4I3,
     1 '  VAL=',E12.4)
C        RUN ACROSS BOUNDARY AND SET CONDITIONS
      KMAX=IABS(MB1(IB)-MB2(IB))+IABS(NB1(IB)-NB2(IB))
      K=0
      DO 200 N=NB1(IB),NB2(IB)
      DO 200 M=MB1(IB),MB2(IB)
      K=K+1
      EL=RAMPT*FINAL(ISET1(IB))
      IF(KMAX.GT.0)EL=RAMPT*(FLOAT(K-1)*FINAL(ISET2(IB))+
     1 FLOAT(KMAX+1-K)*FINAL(ISET1(IB)))/FLOAT(KMAX)
      GOTO(120,130,140,150),JTPO(IB)
c        1.SIMPLE WATER LEVEL
 120  SEPP(N,M)=EL
      SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
      GOTO 190
C        2.OUTWARD TRANSPORT
 130  IF(IABS(ITPO(IB)).EQ.1)THEN
      UHP(N,M+IUH)=F1*EL
      SEPP(N,M)=SE(N,M+ISE)
      ELSE
      VHP(N+IUH,M)=F1*EL
      SEPP(N,M)=SE(N+ISE,M)
      ENDIF
      SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
      GOTO 190
c        3.RADIATION CONDITION, ORLANSKI
 140  IF(IABS(ITPO(IB)).EQ.2)GOTO 145
      DELSE=SE(N,M)-SE(N,M+ISE)
      IF(D(N,M+2*ISE).GT.0.)DELSE=1.5*SE(N,M)-2.*SE(N,M+ISE)
     1 +.5*SE(N,M+2*ISE)
      SEPP(N,M)=SE(N,M)-C1*SQRT(E+AG*.5*(D(N,M)+D(N,M+ISE)
     1  +SE(N,M)+SE(N,M+ISE)))*DELSE
      SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
      GOTO 190
 145  DELSE=SE(N,M)-SE(N+ISE,M)
      IF(D(N+2*ISE,M).GT.0.)DELSE=1.5*SE(N,M)-2.*SE(N+ISE,M)
     1 +.5*SE(N+2*ISE,M)
      SEPP(N,M)=SE(N,M)-C1*SQRT(E+AG*.5*(D(N,M)+D(N+ISE,M)
     1  +SE(N,M)+SE(N+ISE,M)))*DELSE
      SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
      GOTO 190
c        4.RADIATION CONDITION, RIEMANN INVARIENT
 150  IF(IABS(ITPO(IB)).EQ.2)GOTO 155
      SEPP(N,M)=EL+F1*UH(N,M+IUH)/SQRT(E+AG*.5*(D(N,M)+D(N,M+ISE)
     1  +SE(N,M)+SE(N,M+ISE)))
      SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
      GOTO 190
C        Y-DIRECTION
 155  SEPP(N,M)=EL+F1*VH(N+IUH,M)/SQRT(E+AG*.5*(D(N,M)+D(N+ISE,M)
     1  +SE(N,M)+SE(N+ISE,M)))
      SEP(N,M)=.5*(SE(N,M)+SEPP(N,M))
 190  CONTINUE
      IF(ITEST.GT.0)WRITE(ISCR,210)N,M,EL,SEPP(N,M)
 210  FORMAT(3X,'N,M=',2I4,' EL=',F7.4,' SEPP=',F7.4)
 200  CONTINUE
C
C        RIVER FLOW BOUNDARIES
 220  IF(NUMRIV.EQ.0)RETURN
      IF(NSTE.EQ.1.AND.IPRNT1.GT.0)WRITE(ISCR,221)NSTI,NSTE,RAMPR
 221  FORMAT(1X,'BNDRY1: RIVER BOUNDARIES AT NSTI=',I6,' NSTE=',I6,
     1 ' RAMPR=',f4.2)
      CALL RR(YT,ISCR,LURIV,IENDRV,DRIV,YRIV,NSIGR,NDRIV2,QRIV,QRATE)
      DO N=1,NSIGR
      RATE(N)=QRATE(ISETR(N))*RAMPR
      IF(NSTE.EQ.1.AND.IPRNT1.GT.0)WRITE(ISCR,225)N,ITPR(N),JTPR(N),
     1 ISETR(N),RATE(N)
 225  FORMAT(1X,'NR=',I2,' ITPR,JTPR=',2i2,' ISETR=',I2,
     1 ' RATE x RAMPR (m**3/s)=',F12.3)
      ENDDO
      DO 350 NR=1,NSIGR
      IS=0
      IF(ITPR(NR).LT.0)IS=-1
      ISS=1+2*IS
      F1=FLOAT(ISS)/DL
      DO 350 M=MR1(NR),MR2(NR)
      DO 350 N=NR1(NR),NR2(NR)
      IF(JTPR(NR).EQ.1)GOTO 330
C        WATER FALLS CONDITION
      SE(N,M)=SE(N,M)+DTE*RATE(NR)/(AREA(N,M)*DL*DL)
      F2=DTE*RATE(NR)/(AREA(N,M)*DL*DL)
      IF(NSTE.EQ.1.AND.IPRNT1.GT.0)WRITE(ISCR,*)
     1 '  WATER FALLS AT N,M=',N,M,' DEL=',F2
      GOTO 350
 330  IF(IABS(ITPR(NR)).EQ.2)GOTO 340
C        INPUT FLOW, X-DIRECTION
      IBARR=MOD(IFIELD(N,M+IS),10)
      IF(IBARR.EQ.1.OR.IBARR.EQ.3)GOTO 350
      IF(NSTE.GT.1)UH(N,M+IS)=UHP(N,M+IS)
      UHP(N,M+IS)=F1*RATE(NR)/BX(N,M+IS)
      SEPP(N,M)=SE(N,M+ISS)
      SE(N,M)=SE(N,M+ISS)
      GOTO 350
C        INPUT FLOW, Y-DIRECTION
 340  IBARR=MOD(IFIELD(N+IS,M),10)
      IF(IBARR.EQ.2.OR.IBARR.EQ.3)GOTO 350
      IF(NSTE.GT.1)VH(N+IS,M)=VHP(N+IS,M)
      VHP(N+IS,M)=F1*RATE(NR)/BY(N+IS,M)
      SEPP(N,M)=SE(N+ISS,M)
      SE(N,M)=SE(N+ISS,M)
 350  CONTINUE
 360  RETURN
      END
C
C----------------------------------------------------------------------
C
      SUBROUTINE HORVIS
C        APRIL 1986   K. HESS   MEAD   VAX-11/750   (REV 9/87)
C        PURPOSE - TO UPDATE THE HORIZONTAL EDDY VISCOSITY
C        VARIABLES -
C                AH(,) = ARRAY TO STORE PRODUCT OF HORIZONTAL
C                        VISCOSITY AND DEPTH, AT GRID CENTER
C                        AH=AH0+CAH*DL*SQRTU2(U,X)**2+2(V,Y)**2+(U,Y+V,X
C                  AH0 = BACKGROUND VISCOSITY
C                  CAH = FACTOR FOR VISCOSITY
      INCLUDE 'COMM20.FOR'
      DIMENSION UL(LSIZE,NSIZE,MSIZE),VL(LSIZE,NSIZE,MSIZE)
      DATA ISMAG,ALFA/2, 1./
c        skip if necessary
      IF(IHVISC.LE.0.OR.MOD(NSTET,IHVISC).NE.0)GOTO 180
c        save 3-d velocity field
      DO M=1,MMAX
      DO N=1,NMAX
      DO L=1,LBOT
      UL(L,N,M)=UE(N,M)+ALFA*U(L,N,M)
      VL(L,N,M)=VE(N,M)+ALFA*V(L,N,M)
      ENDDO
      ENDDO
      ENDDO
C        LOOP OVER THE INTERIOR CELLS
      KOCN=10*(KOCNBC-1)
      DO 120 M=1,MMAX
      NA=NAB(M)/1000
      NB=MOD(NAB(M),1000)
      IF(NA.GT.NB)GOTO 120
      MM=MAX0(M-1,1)
      MP=MIN0(M+1,MMAX)
      DO 110 N=NA,NB
      IF(IFIELD(N,M).LT.10.OR.IFIELD(N,M).GE.KOCN)GOTO 110
      NM=MAX0(N-1,1)
      NP=MIN0(N+1,NMAX)
      H=D(N,M)+SE(N,M)
      AH1=0.
C        INTERIOR CELLS
      DO 100 L=1,LBOT
      IF(INTER.EQ.0)GOTO 90
      IF(ISMAG.EQ.1)THEN
      DUDY=.25*((UL(L,NP,M)+UL(L,N,M))*AMAX0(NFLUX(N,M),NFLUX(N,MP))
     1      -(UL(L,N,M)+UL(L,NM,M))*AMAX0(NFLUX(NM,M),NFLUX(NM,MP))
     2      +(UL(L,NP,MM)+UL(L,N,MM))*AMAX0(NFLUX(N,MM),NFLUX(N,M))
     3      -(UL(L,N,MM)+UL(L,NM,MM))*AMAX0(NFLUX(NM,MM),NFLUX(NM,M)))
      DVDX=.25*((VL(L,N,M)+VL(L,N,MP))*AMAX0(MFLUX(N,M),MFLUX(NP,M))
     1      -(VL(L,N,M)+VL(L,N,MM))*AMAX0(MFLUX(N,MM),MFLUX(NP,MM))
     2      +(VL(L,NM,MP)+VL(L,NM,M))*AMAX0(MFLUX(N,M),MFLUX(NM,M))
     3      -(VL(L,NM,M)+VL(L,NM,MM))*AMAX0(MFLUX(N,MM),MFLUX(NM,MM)))
      ELSE
      DUDY=.25*(ABS(UL(L,NP,M)  -UL(L,N,M)) +ABS(UL(L,N,M)  -UL(L,NM,M))
     1       +ABS(UL(L,NP,MM)-UL(L,N,MM))+ABS(UL(L,N,MM)-UL(L,NM,MM)))
      DVDX=.25*(ABS(VL(L,N,M)   -VL(L,N,MP))+ABS(VL(L,N,M)  -VL(L,N,MM))
     1       +ABS(VL(L,NM,MP)-VL(L,NM,M))+ABS(VL(L,NM,M)-VL(L,NM,MM)))
      ENDIF
      ARG=2.*(((UL(L,N,M)-UL(L,N,MM)))**2+
     1 ((VL(L,N,M)-VL(L,NM,M)))**2)+(DUDY+DVDX)**2
C        UPDATED VALUE
 90   AMM=AH0+CAH*DL*SQRT(ARG)
      IF(AH3(L,N,M).EQ.0.0)AH3(L,N,M)=AMM*H
      IF(AH3(L,N,M).GT.0.0)AH3(L,N,M)=SQRT(AH3(L,N,M)*AMM*H)
 100  AH1=AH1+CI(L)*AH3(L,N,M)
      AH(N,M)=AH1
 110  CONTINUE
 120  CONTINUE
C        BOUNDARY CELLS
      IF(NUMOBC.LE.0)GOTO 140
      DO 130 IB=1,NUMOBC
      IDIR=IABS(ITPO(IB))
      IS=-ISIGN(1,ITPO(IB))
      DO 130 M=MB1(IB),MB2(IB)
      DO 130 N=NB1(IB),NB2(IB)
      H=D(N,M)+SE(N,M)
c        x-direction
      IF(IDIR.EQ.1)THEN
      F1=H/(D(N,M+IS)+SE(N,M+IS))
      AH(N,M)=AH(N,M+IS)*F1
      DO L=1,LBOT
      AH3(L,N,M)=AH3(L,N,M+IS)*F1
      ENDDO
      ENDIF
c        y-direction
      IF(IDIR.EQ.2)THEN
      F2=H/(D(N+IS,M)+SE(N+IS,M))
      AH(N,M)=AH(N+IS,M)*F2
      DO L=1,LBOT
      AH3(L,N,M)=AH3(L,N+IS,M)*F2
      ENDDO
      ENDIF
 130  CONTINUE
C        RIVER BOUNDARIES
 140  IF(NUMRIV.LE.0)GOTO 160
      DO 150 IB=1,NUMRIV
      IDIR=IABS(ITPR(IB))
      IS=ISIGN(1,ITPR(IB))
      DO 150 M=MR1(IB),MR2(IB)
      DO 150 N=NR1(IB),NR2(IB)
      H=D(N,M)+SE(N,M)
c        x-direction
      IF(IDIR.EQ.1)THEN
      F1=H/(D(N,M+IS)+SE(N,M+IS))
      AH(N,M)=AH(N,M+IS)*F1
      DO L=1,LBOT
      AH3(L,N,M)=AH3(L,N,M+IS)*F1
      ENDDO
      ENDIF
c        y-direction
      IF(IDIR.EQ.2)THEN
      F2=H/(D(N+IS,M)+SE(N+IS,M))
      AH(N,M)=AH(N+IS,M)*F2
      DO L=1,LBOT
      AH3(L,N,M)=AH3(L,N+IS,M)*F2
      ENDDO
```

```
      ENDIF
  150 CONTINUE
C        CORNER VALUES
  160 DO 170 M=1,MMAX
      MP=MIN0(M+1,MMAX)
      DO 170 N=1,NMAX
      NP=MIN0(N+1,NMAX)
      IF(IFIELD(N,M).LT.10)GOTO 170
      AHC(N,M)=.25*(AH(N,M)+AH(N,MP)+AH(NP,M)+AH(NP,MP))
  170 CONTINUE
  180 RETURN
      END
C
C----------------------------------------------------------------------
C
      SUBROUTINE UHVH
C        JANUARY 1997   K.W. HESS
C        PURPOSE - TO COMPUTE EXTERNAL-MODE FLOWRATES AND WATER LEVELS
C                  FROM REVISED EQUATIONS FOR VARIABLE GRID WIDTH (TO
C                  MODEL NARROW RIVER WIDTHS).
C        VARIABLES -
C                  BX = GRID WIDTH FOR FLOW IN X DIRECTION
C                  BY = GRID WIDTH FOR FLOW IN Y DIRECTION
C              THETA1 = INTEGRAL OF (1+U/UB)**2 TIMES UE*BX
C              THETA2 = INTEGRAL OF (1+V/VB)**2 TIMES VE*BY
C              THETA3 = VERTICAL INTEGRAL OF UB*VB*(1+U/UB)*(1+V/VB)
      INCLUDE 'COMM20.FOR'
      BETAH=MAX0(0,MIN0(1,IBETAH))
      BETAA=MAX0(0,MIN0(1,IBETAA))
      CX1=.25*DTE/DL
      CX2=.5*AG*DTE/DL
      CX3=DTE/DL
      CX4=DTE/DL**2
      CX5=.5*DTE/DL
      CX6=.25*AG*DTE/DL
      CX7=1./CX1
      CX8=1./CX5
      CX9=2.*DTE/DL**2                !   additional constant
      CX10=0.1/RHOW
      GAMMA=1.0
      IF(IBOTV.EQ.0)GAMMA=.0
      IF(IBOTV.EQ.3)GAMMA=.5
C
C        X-DIRECTION SWEEP
C
      DO 220 NC=1,NCOL
      N=ICOL(1,NC)
      MA=ICOL(2,NC)
      MB=ICOL(3,NC)
      ILFT=ICOL(4,NC)
      IRGT=ICOL(5,NC)
      MAM=MA-1
      NM=MAX0(N-1,1)
      NP=MIN0(N+1,NMAX)
C        SWEEP DOWN THE COLUMN
      DO 200 M=MAM,MB
      MM=MAX0(M-1,1)
      MP=MIN0(M+1,MMAX)
      MPP=MIN0(M+2,MMAX)
      HS=D(N,M)+BETAH*SE(N,M)+E
      HP=D(N,MP)+BETAH*SE(N,MP)+E
      HM=D(N,MM)+BETAH*SE(N,MM)+E
      HPP=D(N,MPP)+BETAH*SE(N,MPP)+E
      HB=.5*(HS+HP)
      AREAP=AREA(N,M)
      HH=CX2*(HS+HP)
C        TOTAL STRESS FORMULATION
      FBAR=.5*(PHI(N,M)+PHI(N,MP))
      EDGE=AMIN1(FEDGE(N,M),FEDGE(N,MP))
      XDT=DTE*(EDGE*(.5*(TSX(N,M)+TSX(N,MP))-CX10*HB*DPADX-HB*
     1 GSTARX(N,M))-FBAR*(GAMMA*U(LBOT,N,M)+(1.-GAMMA)*U(LBOT-1,N,M)))
C        CHANNEL:    NO:BETAC=0     YES:BETAC=1
      BETAC=1-IFIX(BX(N,M))
      DTFNM=DTE*(FBAR/HB+BETAC*THETSU(N,M))
C        LOWER END BOUNDARY CONDITIONS
      IF(M.GT.MAM)GOTO 160
      AREAP=AREA(N,MP)
      DMP=AREAP*SE(N,MP)-CX1*(BX(N,MP)*UH(N,MP)-BX(N,M)*UH(N,M)
     1 +2.*(BY(N,MP)*VH(N,MP)-BY(NM,MP)*VH(NM,MP)))
C         WATER LEVEL CONDITION
      IF(ILFT.GT.0)GOTO 150
      DENOM=1./(1.+DTFNM)
      FA(M)=(UH(N,M)+XDT+HH*SEP(N,M))*DENOM
      FB(M)=HH*DENOM
      GOTO 200
C        FLOWRATE CONDITION
  150 FA(M)=UHP(N,M)
      FB(M)=0.0
      GOTO 200
C        INTERIOR COMPUTATIONAL GRIDS: MASS RECURSION
  160 AREAX=AREAP
      DENOM=1./(AREAX+CX1*BX(N,MM)*FB(MM))
      GA(M)=(DMP+CX1*BX(N,MM)*FA(MM))*DENOM
      GB(M)=CX1*BX(N,M)*DENOM
C        CHECK FOR UPPER BOUNDARY (M=MB)
      IF(M.EQ.MB)GOTO 180
C        START COMPUTING THE MOMENTUM RECURSION ARRAYS
      VB=(VH(N,M)+VH(N,MP)+VH(NM,M)+VH(NM,MP))/(FLOAT(NFLUX(N,M)+
     1  NFLUX(N,MP)+NFLUX(NM,M)+NFLUX(NM,MP))+E)
      FCOR=EDGE*(FCOR0+DFDM*FLOAT(M-MCOR)+DFDN*FLOAT(N-NCOR))
      AREAP=AREA(N,MP)
      DMP=AREAP*SE(N,MP)-CX1*(BX(N,MP)*UH(N,MP)-BX(N,M)*UH(N,M)
     1 +2.*(BY(N,MP)*VH(N,MP)-BY(NM,MP)*VH(NM,MP)))
      BXI=1.0/BX(N,M)
      BP=CX9*(1.+BX(N,MP)*BXI)*AH(N,MP)
      BM=CX9*(1.+BX(N,MM)*BXI)*AH(N,M)
C        NON-LINEAR TERMS
      AUV=BETAA*CX3*(THETA3(N,M)-THETA3(NM,M))
      CU=SIGN(1.0,UH(N,M))
      FN1=BETAA*CX5*(1.+CU)*BXI*THETA1(N,MM)
      FN2=BETAA*CX5*((1.+CU)-(1.-CU))*BXI*THETA1(N,M)
      FN3=BETAA*CX5*(1.-CU)*BXI*THETA1(N,MP)
C        LATERAL VISCOSITY TERM
      AHDUYY=CX4*(AHC(N,M)*(UE(NP,M)-UE(N,M)+VE(N,MP)-VE(N,M))-
     1  AHC(NM,M)*(UE(N,M)-UE(NM,M)+VE(NM,MP)-VE(NM,M)))*(1.-BETAC)
C        SET RECURSION ARRAY HERE
      F1=-BM/(HS+HM)-FN1
      F2=1.+DTFNM+(BP+BM)/(HS+HP)+FN2
      F3=-BP/(HP+HPP)+FN3
      F4=UH(N,M)+XDT+AHDUYY+FCOR*VB-AUV
      BXIP=F3/BX(N,MP)
      DENOM=1./(GB(M)*(HH+F1*FB(MM))+F2+BX(N,M)*BXIP)
      FA(M)=(F4-F1*FA(MM)+GA(M)*(F1*FB(MM)+HH)-DMP*CX7*BXIP)*DENOM
      FB(M)=(HH-AREAP*CX7*BXIP)*DENOM
      GOTO 200
C        UPPER END BOUNDARY CONDITIONS
  180 IF(IRGT.GT.0)GOTO 190
      DENOM=1./(1.+HH*GB(M)+DTFNM)
      FA(M)=(UH(N,M)+XDT+HH*GA(M))*DENOM
      FB(M)=HH*DENOM
      F1=UH(N,M)+XDT+HH*GA(M)
      GOTO 200
  190 FA(M)=UHP(N,M)
      FB(M)=0.0
  200 CONTINUE
C        LOOP BACK DOWN THE COLUMN, CALCULATING FLOWRATE & WATER LEVEL
      DO 210 MM=MA,MB
      M=MA+MB-MM
      UHP(N,M)=FA(M)-FB(M)*SEP(N,M + 1)
      SEP(N,M)=GA(M)-GB(M)*UHP(N,M)
C        MAKE SURE DEPTH IS POSITIVE
      IF(SEP(N,M)+D(N,M).GT.0.0)GOTO 210
      NEGS=NEGS+1
      SEP(N,M)=0.10-D(N,M)
  210 CONTINUE
      UHP(N,MAM)=FA(MAM)-FB(MAM)*SEP(N,MA)
  220 CONTINUE
C
C        Y-DIRECTION SWEEP
C
  280 DO 370 NR=1,NROW
      M=IROW(1,NR)
      NA=IROW(2,NR)
      NB=IROW(3,NR)
      ILFT=IROW(4,NR)
      IRGT=IROW(5,NR)
      NAM=NA-1
      MM=MAX0(M-1,1)
      MP=MIN0(M+1,MMAX)
C        SWEEP ACROSS ROWS
      DO 350 N=NAM,NB
      NM=MAX0(N-1,1)
      NP=MIN0(N+1,NMAX)
      NPP=MIN0(N+2,NMAX)
      HS=D(N,M)+BETAH*SE(N,M)+E
      HP=D(NP,M)+BETAH*SE(NP,M)+E
      HM=D(NM,M)+BETAH*SE(NM,M)+E
      HPP=D(NPP,M)+BETAH*SE(NPP,M)+E
      HB=.5*(HS+HP)
      AREAP=AREA(N,M)
      HH=CX6*(HS+HP)
C        EXTERNAL FORCES
      FBAR=.5*(PHI(N,M)+PHI(NP,M))
      EDGE=AMIN1(FEDGE(N,M),FEDGE(NP,M))
      YDT=DTE*(EDGE*(.5*(TSY(N,M)+TSY(NP,M))-CX10*HB*DPADY-HB*
     1 GSTARY(N,M))-FBAR*(GAMMA*V(LBOT,N,M)+(1.-GAMMA)*V(LBOT-1,N,M)))
C        CHANNEL:    NO:BETAC=0     YES:BETAC=1
      BETAC=1-IFIX(BY(N,M))
      DTFNM=DTE*(FBAR/HB+BETAC*THETSV(N,M))
C        CHECK FOR LOWER END BOUNDARY CONDITIONS
      IF(N.GT.NAM)GOTO 310
      AREAP=AREA(NP,M)
      DNP=AREAP*SEP(NP,M)-CX1*(BX(NP,M)*(UHP(NP,M)+UH(NP,M))
     1 -BX(NP,MM)*(UHP(NP,MM)+UH(NP,MM)))
C        WATER LEVEL CONDITION
      IF(ILFT.GT.0)GOTO 300
      DENOM=1./(1.+DTFNM)
      FA(N)=(VH(N,M)+YDT+HH*(SEPP(N,M)-SE(NP,M)+SE(N,M)))*DENOM
      FB(N)=HH*DENOM
      GOTO 350
C        FLOWRATE CONDITION
  300 FA(N)=VHP(N,M)
      FB(N)=0.0
      GOTO 350
C        INTERIOR COMPUTATIONAL GRIDS: MASS RECURSION
  310 AREAX=AREAP
      DENOM=1./(AREAX+CX5*BY(NM,M)*FB(NM))
      GA(N)=(DNP+CX5*BY(NM,M)*FA(NM))*DENOM
      GB(N)=CX5*BY(N,M)*DENOM
C        CHECK FOR UPPER END (N=NB)
      IF(N.EQ.NB) GOTO 330
C        START COMPUTING THE MOMENTUM RECURSION ARRAYS
      UB=(UH(N,M)+UH(NP,M)+UH(N,MM)+UH(NP,MM))/(FLOAT(MFLUX(N,M)+
     1  MFLUX(NP,M)+MFLUX(N,MM)+MFLUX(NP,MM))+E)
      FCOR=EDGE*(FCOR0+DFDM*FLOAT(M-MCOR)+DFDN*FLOAT(N-NCOR))
      AREAP=AREA(NP,M)
      DNP=AREAP*SEP(NP,M)-CX1*(BX(NP,M)*(UHP(NP,M)+UH(NP,M))
     1 -BX(NP,MM)*(UHP(NP,MM)+UH(NP,MM)))
      BYI=1.0/BY(N,M)
      BP=CX9*(1.+BY(NP,M)*BYI)*AH(NP,M)
      BM=CX9*(1.+BY(NM,M)*BYI)*AH(N,M)
C        NON-LINEAR TERMS
      AUV=BETAA*CX3*(THETA3(N,M)-THETA3(N,MM))
      CV=SIGN(1.0,VH(N,M))
      FN1=BETAA*CX5*(1.+CV)*BYI*THETA2(NM,M)
      FN2=BETAA*CX5*((1.+CV)-(1.-CV))*BYI*THETA2(N,M)
      FN3=BETAA*CX5*(1.-CV)*BYI*THETA2(NP,M)
C        LATERAL VISCOSITY TERM
      AHDVXX=CX4*(AHC(N,M)*(UE(NP,M)-UE(N,M)+VE(N,MP)-VE(N,M))-
     1  AHC(N,MM)*(UE(NP,MM)-UE(N,MM)+VE(N,M)-VE(N,MM)))*(1.-BETAC)
C        SET RECURSION ARRAY HERE
      F1=-BM/(HS+HM)-FN1
      F2=1.+DTFNM+(BP+BM)/(HS+HP)+FN2
      F3=-BP/(HP+HPP+E)+FN3
      F4=VH(N,M)+YDT+AHDVXX-FCOR*UB-AUV-HH*(SE(NP,M)-SE(N,M))
      BYIP=F3/BY(NP,M)
      DENOM=1./(GB(N)*(HH+F1*FB(NM))+F2+BYIP*BY(N,M))
      FA(N)=(F4-F1*FA(NM)+GA(N)*(HH+F1*FB(NM))-DNP*CX8*BYIP)*DENOM
      FB(N)=(HH-CX8*AREAP*BYIP)*DENOM
      GOTO 350
C        UPPER END BOUNDARY CONDITIONS
  330 IF(IRGT.GT.0)GOTO 340
      DENOM=1./(1.+HH*GB(N)+DTFNM)
      FA(N)=(VH(N,M)+YDT+HH*(GA(N)-SE(NP,M)+SE(N,M)))*DENOM
      FB(N)=HH*DENOM
      GOTO 350
  340 FA(N)=VHP(N,M)
      FB(N)=0.0
  350 CONTINUE
C        LOOP BACK DOWN THE ROW, CALCULATING FLOWRATE AND WATER LEVEL
      DO 360 NN=NA,NB
      N=NA+NB-NN
      VHP(N,M)=FA(N)-FB(N)*SEPP(N + 1,M)
      SEPP(N,M)=GA(N)-GB(N)*VHP(N,M)
C        MAKE SURE DEPTHS ARE POSITIVE
      IF(SEPP(N,M)+D(N,M).GT.0.0)GOTO 360
      NEGS=NEGS+1
      SEPP(N,M)=0.10-D(N,M)
  360 CONTINUE
      VHP(NAM,M)=FA(NAM)-FB(NAM)*SEPP(NA,M)
  370 CONTINUE
C
```

```
C        UPDATE AND SAVE VARIABLES
      F1=2.
      F2=1.
      IF(NSTE.EQ.ISPLIT)THEN
      F1=1.
      F2=.5/FLOAT(ISPLIT)
      END IF
      DO 390 N=1,NMAX
      DO 390 M=1,MMAX
      UH(N,M)=UHP(N,M)
      VH(N,M)=VHP(N,M)
      SE(N,M)=SEPP(N,M)
      UHOLD(N,M)=F2*(UHOLD(N,M)+F1*UH(N,M))
 390  VHOLD(N,M)=F2*(VHOLD(N,M)+F1*VH(N,M))
C        COMPUTE EXTERNAL-MODE VELOCITY
      DO 400 N=1,NMAX
      NP=MIN0(N+1,NMAX)
      DO 400 M=1,MMAX
      MP=MIN0(M+1,MMAX)
      UE(N,M)=UH(N,M)*(1.+FLOAT(MFLUX(N,M)))/
     1 ((D(N,M)+D(N,MP)+BETAH*(SE(N,M)+SE(N,MP))+E))
      VE(N,M)=VH(N,M)*(1.+FLOAT(NFLUX(N,M)))/
     1 ((D(N,M)+D(NP,M)+BETAH*(SE(N,M)+SE(NP,M))+E))
 400  CONTINUE
C
      RETURN
      END
C=====================================================================
C     FILE INTRNL.FOR
C---------------------------------------------------------------------
C
      SUBROUTINE INTRNL
C        MAY 1988   K. W. HESS   TDL
C        PURPOSE - TO COMPUTE THE INTERNAL MODE VARIABLES WITH VARIABLE
C                  WIDTH. THESE ARE THE VERTICAL VELOCITIES, EDDY VISCOS
C                  AND HORIZONTAL VELOCITY DEPARTURES FROM THE VERTICAL
C                  MEAN.
C        VARIABLES -
C                ISKIP = INDEX FOR SKIPPING THE UPDATE OF THE EDDY VISCO
C
      INCLUDE 'COMM20.FOR'
C        CHECK FOR SKIPPING ALL INTERNAL-MODE CALCS
      IF(INTER.EQ.0)GOTO 100
C        GET INTERNAL-MODE BOUNDARY CONDITIONS FOR NON-WATERFALLS
      CALL BNDRY2
C        UPDATE TURBULENT MOMENTUM TRANSFER COEFFICIENTS.
      IF(IVISC.GT.0.AND.(NSTI.LE.IHR.OR.MOD(NSTI,IVISC).EQ.0))
     1 CALL VERVIS
C         GET INTERNAL-MODE VELOCITIES
      CALL UPVP
C         GET VERTICAL VELOCITY
      CALL WVERT
C         GET UPDATED THETA'S
 100  CONTINUE
      RETURN
      END
C
C---------------------------------------------------------------------
C
      SUBROUTINE BNDRY2
C        OCTOBER 1984   K. W. HESS   MEAD   VAX11/750
C        PURPOSE - TO SET THE RIVER'S INTERNAL-MODE VELOCITY BOUNDARY
C                  CONDITIONS FOR NON-FALLS CONDITION.
C        VARIABLES -
C                QRIV = RIVER FLOWRATE (M**3/S)
C              JTPR() = RIVERINE CONDITION: 1=FLUME, 2=WATER FALLS
      INCLUDE 'COMM20.FOR'
C        RIVER FLOW BOUNDARIES
      IF(NUMRIV.LE.0)GOTO 130
C        LOOP THRU THE RIVERS
      DO 120 NR=1,NUMRIV
      IF(JTPR(NR).EQ.2)GOTO 120
C        FLUME CONDITIONS.  SET DIRECTION AND SENSE
      IDIR=IABS(ITPR(NR))
c     IS=0
c     IF(ITPR(NR).LT.0)IS=-1
c     F1=1.
c     IF(ITPR(NR).LE.0)F1=-1.
      F1=ISIGN(1,ITPR(NR))
      IS=(1-ISIGN(1,ITPR(NR)))/2
C        LOOP THRU CELLS AT BOUNDARY
      UTOP=0.
      DO 110 M=MR1(NR),MR2(NR)
      DO 110 N=NR1(NR),NR2(NR)
C        SET NEW VELOCITIES
      F2=F1*UTOP
      IBARR=0
      IF(IDIR.EQ.1)IBARR=MOD(IFIELD(N,M+IS),10)
      IF(IDIR.EQ.2)IBARR=MOD(IFIELD(N+IS,M),10)
      IF(IDIR.EQ.1.AND.(IBARR.EQ.1.OR.IBARR.EQ.3))F2=0.0
      IF(IDIR.EQ.2.AND.IBARR.GE.2)F2=0.0
      F3=PI*DQ
      DO 100 L=1,LBOT
      FVE=COS(F3*FLOAT(L-1))
      IF(IDIR.EQ.1)U(L,N,M+IS)=F2*FVE
      IF(IDIR.EQ.2)V(L,N+IS,M)=F2*FVE
 100  CONTINUE
 110  CONTINUE
 120  CONTINUE
 130  RETURN
      END
C
C-----------------------------------------------------------------
C
      FUNCTION FRHO(SAL,TMP)
C        JUNE 1996       K. W. HESS    CEOB    SGI
C        PURPOSE - TO GENERATE THE ADDED WATER DENSITY DUE TO SALINITY
C                  (S) AND TEMPERATURE (T) DEG. C.  THAT IS,
C                      RHO = RHOW + 1000*FRHO(S,T)  kg/m**3
C                  MAMAEV FORMULATION
      SS=AMAX1(SAL,0.0)
      TT=AMAX1(TMP,0.0)
      FRHO= 7.E-5+SS*(8.02E-4-2.0E-6*TT)-TT*(3.5E-6+4.69E-6*TT)
      RETURN
      END
C
C-----------------------------------------------------------------
C
      FUNCTION FRHO2(S1,T1)
C                  UNESCO FORMULATION
      DATA RHO0/1000./
      TR=AMAX1(0.0,T1)
      SR=AMAX1(0.0,S1)
      RHOR = 999.842594 + 6.793952E-2*TR
     $        - 9.095290E-3*TR**2 + 1.001685E-4*TR**3
     $        - 1.120083E-6*TR**4 + 6.536332E-9*TR**5
      RHOR = RHOR + (0.824493 - 4.0899E-3*TR
     $       + 7.6438E-5*TR**2 - 8.2467E-7*TR**3
     $       + 5.3875E-9*TR**4) * SR
     $       + (-5.72466E-3 + 1.0227E-4*TR
     $       - 1.6546E-6*TR**2) * SR**1.5
     $       + 4.8314E-4 * SR**2
      FRHO=(RHOR-RHO0)*1.E-3
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE GRADP(N,M,INTGRL)
C        PURPOSE -  TO COMPUTE HORIZONTAL GRADIENTS DUE TO DENSITY.
C                   NOTE: RHO = RHOW + FRHO(S,T)
C        VARIABLES -
C              INTGRL = INDEX TO UPDATE INTEGRATED PRESSURE:1=YES
      INCLUDE 'COMM20.FOR'
C        SET INITIAL GRADIENTS
      DO 100 L=1,LBOT
      GRX(L)=0.0
 100  GRY(L)=0.0
      IF(ICOUPL.LE.1.OR.IBETAP.EQ.0.OR.RAMPG.EQ.0.)RETURN
C        SET CONSTANTS
      MP=MIN0(M+1,MMAX)
      NP=MIN0(N+1,NMAX)
      CO1=RAMPG*AG/DL
      CO2=CO1*DQ
C        SET BATHYMETRY VALUES
      H1=(D(N,M)+SE(N,M))
      H2=(D(N,MP)+SE(N,MP))
      DELSX=SE(N,MP)-SE(N,M)
      DELHX=H2-H1
      H3=(D(NP,M)+SE(NP,M))
      DELSY=SE(NP,M)-SE(N,M)
      DELHY=H3-H1
C        INITIALIZE PRESSURE AND DENSITY TERMS
      SX=0.
      SXP=0.
      SY=0.
      SYP=0.
      PX=0.
      PY=0.
C        LOOP OVER DEPTHS
      IF(MFLUX(N,M).EQ.0.OR.H2.LE.0.0)GOTO 120
C        X-DIRECTION GRADIENTS
      DO 110 L=1,LBOT
      Q=FLOAT(1-L)*DQ
      SAVG=.5*(S(L,N,M)+S(L,N,MP))
      TAVG=.5*(T(L,N,M)+T(L,N,MP))
      GRX(L)=(DELSX+DELHX*Q)*CO1*FRHO(SAVG,TAVG)
      IF(L.EQ.1)GOTO 105
      SAVGMP=.5*(S(L,N,MP)+S(L-1,N,MP))
      TAVGMP=.5*(T(L,N,MP)+T(L-1,N,MP))
      SXP=SXP+CO2*FRHO(SAVGMP,TAVGMP)
      SAVGM=.5*(S(L,N,M)+S(L-1,N,M))
      TAVGM=.5*(T(L,N,M)+T(L-1,N,M))
      SX=SX+CO2*FRHO(SAVGM,TAVGM)
      GRX(L)=GRX(L)+(SXP*H2-SX*H1)
 105  PX=PX+CI(L)*GRX(L)
 110  CONTINUE
C        Y-DIRECTION
 120  IF(NFLUX(N,M).EQ.0.OR.H3.LE.0.0)GOTO 140
      DO 130 L=1,LBOT
      Q=FLOAT(1-L)*DQ
      SAVG=.5*(S(L,N,M)+S(L,NP,M))
      TAVG=.5*(T(L,N,M)+T(L,NP,M))
      GRY(L)=(DELSY+DELHY*Q)*CO1*FRHO(SAVG,TAVG)
c     IF(L.EQ.1)GOTO 130
      IF(L.EQ.1)GOTO 125
      SAVGNP=.5*(S(L,NP,M)+S(L-1,NP,M))
      TAVGNP=.5*(T(L,NP,M)+T(L-1,NP,M))
      SYP=SYP+CO2*FRHO(SAVGNP,TAVGNP)
      SAVGN=.5*(S(L,N,M)+S(L-1,N,M))
      TAVGN=.5*(T(L,N,M)+T(L-1,N,M))
      SY=SY+CO2*FRHO(SAVGN,TAVGN)
      GRY(L)=GRY(L)+(SYP*H3-SY*H1)
 125  PY=PY+CI(L)*GRY(L)
 130  CONTINUE
 140  CONTINUE
C        INTEGRATED PRESSURE
      IF(INTGRL.NE.1)GOTO 150
      GSTARX(N,M)=PX
      GSTARY(N,M)=PY
 150  RETURN
      END
C
C----------------------------------------------------------------------
C
      SUBROUTINE THETAS
C        JAN 1997   K.W.HESS
C        PURPOSE -  TO COMPUTE THE THETA FUNCTIONS.
C        VARIABLES -
C           THETA1(,) = BX*UE*INTEGRAL OVER DEPTH OF:(1 + U/UE)**2
C           THETA2(,) = BY*VE*INTEGRAL OVER DEPTH OF:(1 + V/VE)**2
C           THETA3(,) = UE*VE*INTEGRAL OVER DEPTH OF:(1+U/UE)*(1+V/VE)
C           THETSU(,) = INTEGRAL OVER DEPTH OF (1 + U/UE)*ABS(1 + U/UE)
C                       TIMES CDRGWS*H/BX
C           THETSV(,) = INTEGRAL OVER DEPTH OF (1 + V/VE)*ABS(1 + V/VE)
C                       TIMES CDRGWS*H/BY
      INCLUDE 'COMM20.FOR'
      MMAXM=MMAX-1
      NMAXM=NMAX-1
      DO 250 M=1,MMAXM
      DO 250 N=1,NMAXM
      IF(IFIELD(N,M).LT.10)GOTO 250
c        EXTERNAL-MODE ONLY
      IF(INTER.NE.0)GOTO 150
      TH1=1.0
      TH2=1.0
      TH3=.25*(UE(N,M)+UE(N+1,M))*(VE(N,M)+VE(N,M+1))
      TS1=1.0
      TS2=1.0
      GOTO 210
C        INTERNAL MODE. GET RECIPROCAL OF UE, VE
 150  UBI=1./SIGN(AMAX1(ABS(UE(N,M)),0.0001),UE(N,M))
      VBI=1./SIGN(AMAX1(ABS(VE(N,M)),0.0001),VE(N,M))
C        SUM OVER DEPTH
      TH1=0.0
      TH2=0.0
      TH3=0.0
      TS1=0.0
      TS2=0.0
      DO 200 L=1,LBOT
      IF(IBETAA.LE.0)GOTO 180
      TH1=TH1+CI(L)*(1.+U(L,N,M)*UBI)**2
```

```
      TH2=TH2+CI(L)*(1.+V(L,N,M)*VBI)**2
      TH3=TH3+CI(L)*.25*(UE(N,M)+U(L,N,M)+UE(N+1,M)+U(L,N+1,M))*
     1                  (VE(N,M)+V(L,N,M)+VE(N,M+1)+V(L,N,M+1))
  180 IF(BX(N,M).LT.1.0)TS1=TS1+CI(L)*(1.+U(L,N,M)*UBI)*ABS(1.+
     1  U(L,N,M)*UBI)
      IF(BY(N,M).LT.1.0)TS2=TS2+CI(L)*(1.+V(L,N,M)*VBI)*ABS(1.+
     1 V(L,N,M)*VBI)
  200 CONTINUE
C        CHECK MAGNITUDES OF NON-LINEAR TERM INTEGRALS
  210 IF(IBETAA.LE.0)GOTO 230
      THETA1(N,M)=.5*(THETA1(N,M)+AMIN1(10.0,TH1)*UE(N,M)*BX(N,M))
      THETA2(N,M)=.5*(THETA2(N,M)+AMIN1(10.0,TH2)*VE(N,M)*BY(N,M))
      TH3=SIGN(AMIN1(ABS(TH3),10.0),TH3)
      IF(MOD(IFIELD(N,M),10).GT.0.OR.IFIELD(N+1,M).LT.10.OR.
     1 IFIELD(N,M+1).LT.10.OR.IFIELD(N+1,M+1).LT.10)TH3=0.0
      THETA3(N,M)=.5*(THETA3(N,M)+TH3)
C        SIDE FRICTION TERMS
  230 THETSU(N,M)=CDRGWS*TS1*ABS(UE(N,M))/(DL*BX(N,M))
      THETSV(N,M)=CDRGWS*TS2*ABS(VE(N,M))/(DL*BY(N,M))
  250 CONTINUE
      RETURN
      END
C
c---------------------------------------------------------------
C
      SUBROUTINE GETCJ(CJ,CI,LBOT,LSIZE,IBOTV)
C        FACTOR TO REDISTRIBUTE ANY NON-ZERO INTERNAL-MODEL VELOCITY,
C        MAINTAINING SAME BOTTOM VELOCITY IF IBOTV=0.
      DIMENSION CJ(LSIZE),CI(LSIZE)
      SUM=0.
      DO L=1,LBOT
      CJ(L)=1.
      IF(IBOTV.EQ.0)CJ(L)=MAX0(MIN0(LBOT-L,2),0)
      SUM=SUM+CI(L)*CJ(L)
      ENDDO
      DO L=1,LBOT
      CJ(L)=CJ(L)/SUM
      ENDDO
      RETURN
      END
C
C----------------------------------------------------------------------
C
      SUBROUTINE UPVP

C        MARCH 1996   K. W. HESS   CEOB
C        PURPOSE - TO COMPUTE THE INTERNAL MODE VARIABLES WITH VARIABLE
C                  WIDTH. THESE ARE THE VERTICAL VELOCITIES, EDDY VISCOS
C                  AND HORIZONTAL VELOCITY DEPARTURES FROM THE VERTICAL
C                  (SEE MECCA PROGRAM DOCUMENTATION, PART B).
C                  INCLUDES NON-LINEAR TERMS W/O IF STATEMENTS AND
C                  HAS 3-D HORIZONTAL VISCOSITY.
C        VARIABLES -
C               IBOTV = BOTTOM B.C. INDEX:
C                          0 : U=0
C                          1 : AvDU/DZ = TBX   first order DU/DZ
C                          2 : AvDU/DZ = TBX   second order DU/DZ
C                          3 : AvDU/DZ = TBX   log-layer, mid-level
C               ITOPV = ORDER OF TOP B.C. DERIVATIVE
C                          1 : FIRST ORDER DU/DZ
C                          2 : SECOND.
C                          3 : FIRST-ORDER TOTAL
C
      INCLUDE 'COMM20.FOR'
      DIMENSION UP(LSIZE),UPM(LSIZE,NSIZE),UPMM(LSIZE,NSIZE),VP(LSIZE),
     2          VPM(LSIZE,NSIZE),VPMM(LSIZE,NSIZE),FBC(LSIZE),CJ(LSIZE)
      CALL GETCJ(CJ,CI,LBOT,LSIZE,IBOTV)
C        NONLINEAR TERMS: INCLUDE IF IBETAA=1
      FNONL0=0.
      IF(IBETAA.NE.0)FNONL0=1.
C        NONLINEAR TERMS:FOR NONL, 1=NO, 2=YES
      BETAH=MAX0(0,MIN0(1,IBETAH))
      KOCN=10*KOCNBC
      KRIV=10*(KOCNBC+1)
      LAYRM=LAYRS-1
      DIFF=0.0
C
      B1=DTI/(8.*DQ)
      B2=DTI/(2.*DQ**2)
      B3=DTI/(2.*DL**2)
      B4=DTI/DL**2
      B5=DTI/(2.*DL)
      B6=DTI/DL
      B7=DTI/(16.*DL)
      B8=DTI/(4.*DL)
      B9=2./3.
      B10=4./3.
      FSPLIT=ISPLIT
C
      DO 90 L=1,LBOT
   90 FBC(L)=0.0
      IF(ITOPV.EQ.2)FBC(2)=0.33333
C
C        BEGIN LOOP THRU THE MESH
      DO 550 M=1,MMAX
C        STORE THE PRESENT VELOCITIES
      DO 120 N=1,NMAX
      DO 120 L=1,LBOT
      IF(M.GT.1)GOTO 100
      UPMM(L,N)=0.0
      VPMM(L,N)=0.0
      GOTO 110
  100 UPMM(L,N)=UPM(L,N)
      IF(M.EQ.1)UPMM(L,N)=2.*UPM(L,M)-U(L,N,M+1)
      VPMM(L,N)=VPM(L,N)
  110 UPM(L,N)=U(L,N,M)
      VPM(L,N)=V(L,N,M)
      IF(N.EQ.1)VPM(L,M)=2.*V(L,N,m)-V(L,N+1,M)
  120 CONTINUE
C        DEFINE THE COLUMN LIMITS
      NA=NAB(M)/1000
      NB=NAB(M)-1000*NA
      IF(NA.GT.NB)GOTO 550
      DO 545 N=NA,NB
      II=IFIELD(N,M)
      IF(II.LT.10)GOTO 540
      IBARR=MOD(II,10)
      IF(IBARR.EQ.3)GOTO 540
C        GET THE INTERNAL PRESSURE GRADIENT
      CALL GRADP(N,M,1)
      MM=MAX0(M-1,1)
      MP=MIN0(M+1,MMAX)
      NM=MAX0(N-1,1)
      NP=MIN0(N+1,NMAX)
C
C        X-DIRECTION HORIZONTAL VELOCITY DEPARTURE
C
      IF(IBARR.EQ.1.OR.M.EQ.MMAX)GOTO 330
      IF(IFIELD(N,M).EQ.KOCN.AND.IFIELD(N,MP).EQ.KOCN)GOTO 330
C        CHECK FOR INCLUSION OF NON-LINEAR TERMS:NLT(1=NO;2=YES)
      EDGE=AMIN1(FEDGE(N,M),FEDGE(N,MP))
      FNONL=FNONL0
      IF(EDGE.LT.0.55)FNONL=0.
C        CHECK FOR CHANNEL:NO=0, YES=1.
      BETAC=1-IFIX(BX(N,M))
C        DEPTH-INDEPENDENT VARIABLES
      HS=D(N,M)+BETAH*SE(N,M)
      HP=D(N,MP)+BETAH*SE(N,MP)
      HPP=D(N,MP+1)+BETAH*SE(N,MP+1)
      HB=.5*(HS+HP)
      HI=1./(HB+E)
      HISQ=B2*HI**2
C        TOTAL EXTERNAL RETARDING FORCES PLUS SIDE STRESS
      CFS=DTI*BETAC*CDRGWS/(DL*BX(N,M))
      XDT=DTI*((TBX(N,M)-.5*TSX(N,M)-.5*TSX(N,MP))*HI+EDGE*GSTARX(N,M)
     1 + THETSU(N,M)*UE(N,M))
      FCOR=EDGE*FSPLIT*(FCOR0+DFDM*FLOAT(M-MCOR)+DFDN*FLOAT(N-NCOR))
      DELH=.25*(SEPP(N,M)+SEPP(N,MP)-SOLD(N,M)-SOLD(N,MP))*HI
      RAP=B3*HI/BX(N,M)*(BX(N,M)+BX(N,MP))
      RAM=B3*HI/BX(N,M)*(BX(N,M)+BX(N,MM))
      RBP=B4*HI
      RBM=B4*HI
      FAVG=1./(E+FLOAT(NFLUX(N,M)+NFLUX(NM,M)+NFLUX(N,MP)+NFLUX(NM,MP)))
C        NON-LINEAR TERMS
      CU=SIGN(1.0,UH(N,M))
      IF(FNONL.EQ.0.)THEN
      ANA=0.
      ELSE
      F1=B7*HI/BX(N,M)
      HBXM=F1*BX(N,MM)*(HS+D(N,MM)+BETAH*SE(N,MM))
      HBX= F1*BX(N,M)*(HS+HP)
      HBXP=F1*BX(N,MP)*(HP+HPP)
      HCP=B8*HI*(HS+HP+D(NP,M)+D(NP,MP)+BETAH*(SE(NP,M)+SE(NP,MP)))
      HCM=B8*HI*(HS+HP+D(NM,M)+D(NM,MP)+BETAH*(SE(NM,M)+SE(NM,MP)))
      ANA=FNONL*(B5*HI/BX(N,M)*((1.-CU)*(THETA1(N,MP)*UH(N,MP)
     1 -THETA1(N,M)*UH(N,M))+(1.+CU)*(THETA1(N,M)*UH(N,M)
     2 -THETA1(N,MM)*UH(N,MM)))+B6*HI*(THETA3(N,M)-THETA3(NM,M)))
      END IF
C        APPLY TOP BOUNDARY CONDITIONS
      IF(ITOPV.EQ.1)THEN
      FA(1)=HB*DQ/(AV(1,N,M)+AV(1,N,MP))*(TSX(N,M)+TSX(N,MP))
      FB(1)=1.0
      ELSE IF(ITOPV.EQ.2)THEN
      FA(1)=B9*HB*DQ/(AV(1,N,M)+AV(1,N,MP))*(TSX(N,M)+TSX(N,MP))
      FB(1)=B10
      ELSE IF(ITOPV.EQ.3)THEN
      DP=HISQ*(AV(1,N,M)+AV(1,N,MP))
      GP=FNONL*B1*HI*(W(2,N,M)+W(2,N,MP)+W(1,N,M)+W(1,N,MP))
      DIFF=RAP*AH3(1,N,MP)*(U(1,N,MP)-U(1,N,M))
     1    -RAM*AH3(1,N,M)*(U(1,N,M)-UPMM(1,N))
     2 +(1.-BETAC)*(RBP*(U(1,NP,M)-U(1,N,M)+V(1,N,MP)-V(1,N,M))
     3  *.25*(AH3(1,N,M)+AH3(1,NP,M)+AH3(1,N,MP)+AH3(1,NP,MP))
     4         -RBM*(U(1,N,M)-UPM(1,NM)+V(1,NM,MP)-VPM(1,NM))
     5  *.25*(AH3(1,N,M)+AH3(1,NM,M)+AH3(1,N,MP)+AH3(1,NM,MP)) )
      UCEN=U(1,N,M)+UE(N,M)
      VB=(VPM(1,N)+VPM(1,NM)+V(1,N,MP)+V(1,NM,MP))*FAVG
      ANB=FNONL*((1.-CU)*(HBXP*(U(1,N,MP)+UE(N,MP))**2-HBX*(UCEN)**2)
     1           +(1.+CU)*(HBX*(UCEN)**2-HBXM*(UPMM(1,N)+UE(N,MM))**2)
     2 +HCP*(U(1,NP,M)+UE(NP,M)+UCEN)*(V(1,N,M)+VE(N,M)
     3 +V(1,N,MP)+VE(N,MP))-HCM*(UCEN+UPM(1,NM)+UE(NM,M))*(VPM(1,NM)
     4 +VE(NM,M)+V(1,NM,MP)+VE(NM,MP)) )
      XDT1=.5*DTI*(TSX(N,M)+TSX(N,MP))*HI/DQ
      DENOM=1./(1.+DELH+2.*(DP-GP)+CFS*ABS(UCEN))
      FA(1)=(UPM(1,N)*(1.-DELH)+(XDT-CFS*ABS(UCEN)*UE(N,M)
     1 -EDGE*DTI*GRX(1)+FCOR*VB+DIFF+ANA-ANB)+XDT1)*DENOM
      FB(1)=2.*(GP+DP)*DENOM
      ENDIF
C
C        LOOP THRU THE DEPTHS
  180 DO 220 L=2,LAYRS
      DP=HISQ*(AV(L,N,M)+AV(L,N,MP))
      DM=HISQ*(AV(L-1,N,M)+AV(L-1,N,MP))
      VB=(VPM(L,N)+VPM(L,NM)+V(L,N,MP)+V(L,NM,MP))*FAVG
C        MOMENTUM DIFFUSION TERMS
      DIFF=RAP*AH3(L,N,MP)*(U(L,N,MP)-U(L,N,M))
     1    -RAM*AH3(L,N,M)*(U(L,N,M)-UPMM(L,N))
     2 +(1.-BETAC)*(RBP*(U(L,NP,M)-U(L,N,M)+V(L,N,MP)-V(L,N,M))
     3  *.25*(AH3(L,N,M)+AH3(L,NP,M)+AH3(L,N,MP)+AH3(L,NP,MP))
     4         -RBM*(U(L,N,M)-UPM(L,NM)+V(L,NM,MP)-VPM(L,NM))
     5  *.25*(AH3(L,N,M)+AH3(L,NM,M)+AH3(L,N,MP)+AH3(L,NM,MP)) )
C        NON-LINEAR TERMS
      UCEN=U(L,N,M)+UE(N,M)
      ANB=FNONL*((1.-CU)*(HBXP*(U(L,N,MP)+UE(N,MP))**2-HBX*(UCEN)**2)
     1           +(1.+CU)*(HBX*(UCEN)**2-HBXM*(UPMM(L,N)+UE(N,MM))**2)
     2 +HCP*(U(L,NP,M)+UE(NP,M)+UCEN)*
     3      (V(L,N,M)+VE(N,M)+V(L,N,MP)+VE(N,MP))
     4 -HCM*(UCEN+UPM(L,NM)+UE(NM,M))*
     5 (VPM(L,NM)+VE(NM,M)+V(L,NM,MP)+VE(NM,MP)) )
      GM=FNONL*B1*HI*(W(L-1,N,M)+W(L-1,N,MP)+W(L,N,M)+W(L,N,MP))
      GP=FNONL*B1*HI*(W(L+1,N,M)+W(L+1,N,MP)+W(L,N,M)+W(L,N,MP))
C         CALCULATE THE RECURSIVE TERMS
      F1=GM-DM
      F2=1.+DELH+GM-GP+DM+DP+CFS*ABS(UCEN)
      F3=-GP-DP
      F4=UPM(L,N)*(1.-DELH)+XDT-CFS*ABS(UCEN)*UE(N,M)-EDGE*DTI*GRX(L)
     1 +FCOR*VB+DIFF+ANA-ANB
      DENOM=1./(F2+F1*FB(L-1))
      FA(L)=(F4-F1*FA(L-1))*DENOM
      FB(L)=(FBC(L)*F1-F3)*DENOM
  220 CONTINUE
C        APPLY BOTTOM BOUNDARY CONDITIONS
      IF(IBOTV.EQ.0)THEN
      UP(LBOT)=-UE(N,M)
      ELSE IF(IBOTV.EQ.1)THEN  !  FIRST-ORDER
      RR=DQ*HB*(PHI(N,M)+PHI(N,MP))/(AV(LAYRS,N,M)+AV(LAYRS,N,MP))
      UP(LBOT)=(FA(LAYRS)-RR*UE(N,M))/(1.+RR-FB(LAYRS))
      ELSE IF(IBOTV.EQ.2)THEN
      RR=DQ*HB*(PHI(N,M)+PHI(N,MP))/(AV(LAYRS,N,M)+AV(LAYRS,N,MP))
      UP(LBOT)=(2.*RR*UE(N,M)+FA(LAYRS-1)-(4.-FB(LAYRS
     1 -1.))*FA(LAYRS))/((4.-FB(LAYRS-1))*FB(LAYRS)-3.-2.*RR)
      ELSE IF(IBOTV.EQ.3)THEN  !  STRESS AT MID-LEVEL
      F1=.25*(PHI(N,M)+PHI(N,MP))
      F2=.50*(AV(LAYRS,N,M)+AV(LAYRS,N,MP))/(DQ*HB)
      UP(LBOT)=(FA(LAYRS)*(F2-F1)-2.*F1*UE(N,M))/(F1+F2
     1 -FB(LAYRS)*(F2-F1))
      ENDIF
C        FIND UPDATED VELOCITY, AND NET VELOCITY
  270 UNET=HALFDQ*UP(LBOT)
      DO 280 I=1,LAYRM
      UP(LBOT-I)=FA(LBOT-I)+FB(LBOT-I)*UP(LBOT-I+1)
  280 UNET=UNET+DQ*UP(LBOT-I)
C        SECOND ORDER
```

```
      GOTO(290,300,290),ITOPV
 290  UP(1)=FA(1)+FB(1)*UP(2)
      GOTO 310
 300  UP(1)=FB(1)*UP(2)-.33333*UP(3)+FA(1)
 310  UNET=UNET+HALFDQ*UP(1)
C        ENFORCE ZERO NET INTERNAL-MODE FLOW
      DO 320 L=1,LBOT
 320  U(L,N,M)=UP(L)-UNET*CJ(L)
C
C        Y-DIRECTION INTERNAL MODE VELOCITY
C
 330  IF(IBARR.GE.2)GOTO 540
      IF(N.EQ.NMAX)GOTO 540
      IF(IFIELD(N,M).EQ.KOCN.AND.IFIELD(NP,M).EQ.KOCN)GOTO 540
C         CHECK FOR NON-LINEAR TERMS
      EDGE=AMIN1(FEDGE(N,M),FEDGE(NP,M))
      FNONL=FNONL0
      IF(EDGE.LT.0.55)FNONL=0.
C        CHECK FOR CHANNEL:NO=0, YES=1.
      BETAC=1-IFIX(BY(N,M))
C        DEPTH VARIABLES
      HS=D(N,M)+BETAH*SE(N,M)
      HP=D(NP,M)+BETAH*SE(NP,M)
      HPP=HP
      IF(NP.LT.NMAX)HPP=D(NP+1,M)+BETAH*SE(NP+1,M)
      HB=.5*(HS+HP)
      HI=1./(HB+E)
      CFS=DTI*BETAC*CDRGWS/(DL*BY(N,M))
C        TOTAL EXTERNAL RETARDING FORCE
      YDT=DTI*((TBY(N,M)-.5*(TSY(N,M)+TSY(NP,M)))*HI+EDGE*GSTARY(N,M)
     1 + (THETSV(N,M))*VE(N,M))
      HISQ=B2*HI**2
      DELH=.25*(SEPP(N,M)+SEPP(NP,M)-SOLD(N,M)-SOLD(NP,M))*HI
      FAVG=1./(E+FLOAT(MFLUX(N,M))+FLOAT(MFLUX(N,MM))
     1 +FLOAT(MFLUX(NP,M))+FLOAT(MFLUX(NP,MM)))
      FCOR=EDGE*FSPLIT*(FCOR0+DFDM*FLOAT(M-MCOR)+DFDN*FLOAT(N-NCOR))
      RAP=B3*HI/BY(N,M)*(BY(N,M)+BY(NP,M))
      RAM=B3*HI/BY(N,M)*(BY(N,M)+BY(NM,M))
      RBP=B4*HI
      RBM=B4*HI
C        NON-LINEAR TERMS
      CV=SIGN(1.0,VH(N,M))
      IF(FNONL.EQ.0.)THEN
      ANA=0.
      ELSE
      F1=B7*HI/BY(N,M)
      HBYM=F1*BY(NM,M)*(HS+D(NM,M)+BETAH*SE(NM,M))
      HBY=F1*BY(N,M)*(HS+HP)
      HBYP=F1*BY(NP,M)*(HP+HPP)
      HCP=B8*HI*(HS+HP+D(N,MP)+D(NP,MP)+BETAH*(SE(N,MP)+SE(NP,MP)))
      HCM=B8*HI*(HS+HP+D(N,MM)+D(NP,MM)+BETAH*(SE(N,MM)+SE(NP,MM)))
      ANA=B5*HI/BY(N,M)*((1.-CV)*(THETA2(NP,M)*VH(NP,M)
     1 -THETA2(N,M)*VH(N,M))+(1.+CV)*(THETA2(N,M)*VH(N,M)
     2 -THETA2(NM,M)*VH(NM,M)))+B6*HI*(THETA3(N,M)-THETA3(N,MM))
      ENDIF
C        APPLY TOP BOUNDARY CONDITIONS
      IF(ITOPV.EQ.1)THEN
      FA(1)=HB*DQ/(AV(1,N,M)+AV(1,NP,M))*(TSY(N,M)+TSY(NP,M))
      FB(1)=1.0
      ELSE IF(ITOPV.EQ.2)THEN
      FA(1)=.66667*HB*DQ/(AV(1,N,M)+AV(1,NP,M))*(TSY(N,M)+TSY(NP,M))
      FB(1)=1.33333
      ELSE IF(ITOPV.EQ.3)THEN
      DP=HISQ*(AV(1,N,M)+AV(1,NP,M))
      GP=FNONL*B1*HI*(W(2,N,M)+W(2,NP,M)+W(1,N,M)+W(1,NP,M))
      DIFF=RAP*AH3(1,NP,M)*(V(1,NP,M)-V(1,N,M))
     1    -RAM*AH3(1,N,M)*(V(1,N,M)-VPM(1,NM))
     2 +(1.-BETAC)*(RBP*(U(1,NP,M)-UPM(1,N)+V(1,N,MP)-V(1,N,M))
     3  *.25*(AH3(1,N,M)+AH3(1,NP,M)+AH3(1,N,MP)+AH3(1,NP,MP))
     4 -RBM*(UPMM(1,NP)-UPMM(1,N)+V(1,N,M)-VPM(1,N))
     5  *.25*(AH3(1,N,M)+AH3(1,NP,M)+AH3(1,N,MM)+AH3(1,NP,MM)) )
      VCEN=V(1,N,M)+VE(N,M)
      UB=(UPM(1,N)+UPMM(1,N)+UPM(1,NP)+UPMM(1,NP))*FAVG
      ANB=FNONL*((1.-CV)*(HBYP*(V(1,NP,M)+VE(NP,M))**2-HBY*(VCEN)**2)
     1   +(1.+CV)*(HBY*(VCEN)**2-HBYM*(VPM(1,NM)+VE(NM,M))**2)
     2 +HCP*(U(1,N,M)+UE(N,M)+U(1,NP,M)+UE(NP,M))*(VCEN
     3 +V(1,N,MP)+VE(N,MP))-HCM*(UPMM(1,N)+UE(N,MM)+UPMM(1,NP)
     4 +UE(NP,MM))*(VCEN+VPMM(1,N)+VE(N,MM)))
      YDT1=.5*DTI*(TSY(N,M)+TSY(NP,M))*HI/DQ
      DENOM=1./(1.+DELH+2.*(DP-GP)+CFS*ABS(VCEN))
      FA(1)=(VPM(1,N)*(1.-DELH)+(YDT-CFS*ABS(VCEN)*VE(N,M)
     1 -EDGE*DTI*GRY(1)-FCOR*UB+DIFF+ANA-ANB)+YDT1)*DENOM
      FB(1)=2.*(GP+DP)*DENOM
      ENDIF
C
C        LOOP THRU THE LAYERS
 390  DO 430 L=2,LAYRS
      DP=HISQ*(AV(L,N,M)+AV(L,NP,M))
      DM=HISQ*(AV(L-1,N,M)+AV(L-1,NP,M))
      UB=(UPM(L,N)+UPMM(L,N)+UPM(L,NP)+UPMM(L,NP))*FAVG
C        HORIZONTAL DIFFUSION OF MOMENTUM
      DIFF=RAP*AH3(L,NP,M)*(V(L,NP,M)-V(L,N,M))
     1    -RAM*AH3(L,N,M)*(V(L,N,M)-VPM(L,NM))
     2 +(1.-BETAC)*(RBP*(U(L,NP,M)-UPM(L,N)+V(L,N,MP)-V(L,N,M))
     3  *.25*(AH3(L,N,M)+AH3(L,NP,M)+AH3(L,N,MP)+AH3(L,NP,MP))
     4 -RBM*(UPMM(L,NP)-UPMM(L,N)+V(L,N,M)-VPM(L,N))
     5  *.25*(AH3(L,N,M)+AH3(L,NP,M)+AH3(L,N,MM)+AH3(L,NP,MM)) )
C        NON-LINEAR TERMS
      VCEN=V(L,N,M)+VE(N,M)
      ANB=FNONL*((1.-CV)*(HBYP*(V(L,NP,M)+VE(NP,M))**2-HBY*(VCEN)**2)
     1          +(1.+CV)*(HBY*(VCEN)**2-HBYM*(VPM(L,NM)+VE(NM,M))**2)
     2 +HCP*(U(L,N,M)+UE(N,M)+U(L,NP,M)+UE(NP,M))*
     3      (VCEN+V(L,N,MP)+VE(N,MP))
     4 -HCM*(UPMM(L,N)+UE(N,MM)+UPMM(L,NP)+UE(NP,MM))*
     5      (VCEN+VPMM(L,N)+VE(N,MM)))
      GM=FNONL*B1*HI*(W(L-1,N,M)+W(L-1,NP,M)+W(L,N,M)+W(L,NP,M))
      GP=FNONL*B1*HI*(W(L+1,N,M)+W(L+1,NP,M)+W(L,N,M)+W(L,NP,M))
C        COMPUTE THE RECURSIVE ARRAYS
      F1=GM-DM
      FS=CFS*ABS(VCEN)
      F2=1.+DELH+GM-GP+DM+DP+FS
      F3=-GP-DP
      F4=VPM(L,N)*(1.-DELH)+YDT-FS*VE(N,M)-EDGE*DTI*GRY(L)-FCOR*UB+DIFF
     1 +ANA-ANB
      DENOM=1./(F2+F1*FB(L-1))
      FA(L)=(F4-F1*FA(L-1))*DENOM
      FB(L)=(FBC(L)*F1-F3)*DENOM
 430  CONTINUE
C         APPLY BOTTOM BOUNDARY CONDITIONS
      IF(IBOTV.EQ.0)THEN
      VP(LBOT)=-VE(N,M)
      ELSE IF(IBOTV.EQ.1)THEN  !       FIRST ORDER
      RR=DQ*HB*(PHI(N,M)+PHI(NP,M))/(AV(LAYRS,N,M)+AV(LAYRS,NP,M))
      VP(LBOT)=(FA(LAYRS)-RR*VE(N,M))/(1.+RR-FB(LAYRS))
      ELSE IF(IBOTV.EQ.2)THEN  !       SECOND ORDER
      RR=DQ*HB*(PHI(N,M)+PHI(NP,M))/(AV(LAYRS,N,M)+AV(LAYRS,NP,M))
      VP(LBOT)=(2.*RR*VE(N,M)+FA(LAYRS-1)-(4.-FB(LAYRS
     1 -1.))*FA(LAYRS))/((4.-FB(LAYRS-1))*FB(LAYRS)-3.-2.*RR)
      ELSE IF(IBOTV.EQ.3)THEN  !       STRESS AT MID-LEVEL
      F1=.25*(PHI(N,M)+PHI(NP,M))
      F2=.50*(AV(LAYRS,N,M)+AV(LAYRS,NP,M))/(DQ*HB)
      VP(LBOT)=(FA(LAYRS)*(F2-F1)-2.*F1*VE(N,M))/(F1+F2
     1 -FB(LAYRS)*(F2-F1))
      END IF
C        FIND UPDATED VELOCITY AND NET VELOCITY
 480  UNET=HALFDQ*VP(LBOT)
      DO 490 I=1,LAYRM
      VP(LBOT-I)=FA(LBOT-I)+FB(LBOT-I)*VP(LBOT-I+1)
 490  UNET=UNET+DQ*VP(LBOT-I)
      GOTO(500,510,500),ITOPV
 500  VP(1)=FA(1)+FB(1)*VP(2)
      GOTO 520
 510  VP(1)=FB(1)*VP(2)-.33333*VP(3)+FA(1)
 520  UNET=UNET+HALFDQ*VP(1)
C        ENFORCE ZERO NET FLO
      DO 530 L=1,LBOT
 530  V(L,N,M)=VP(L)-UNET*CJ(L)
C
 540  CONTINUE
 545  CONTINUE
 550  CONTINUE
C
 660  CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE VERVIS
C        SEPTEMBER 1987   MEAD   K.W.HESS   VAX 11/780
C        PURPOSE - TO UPDATE ALL VERTICAL EXCHANGE COEFFICIENTS.
C                  USE THE MONK AND ANDERSON FORMULATION.
C        VARIABLES -
C               INDEX = RUN UPDATE INDEX:0=RESTART CONDITION,1=NORMAL
C              RICHNO = RICHARDSON NUMBER = {D(RHO)/DZ}/{(DU/DZ**2)*RHO*G}
      INCLUDE 'COMM20.FOR'
      COMMON/A1/DTIMAX
      BETAH=MAX0(0,MIN0(1,IBETAH))
      IF(NSTI.EQ.1)DTIMAX=1.E+10
C        SET VERTICAL DENSITY CHANGE (GM/CC) PER METER
      DO 130 M=1,MMAX
      DO 120 N=1,NMAX
      IF(IFIELD(N,M).LE.0)GOTO 120
      AVMAX=0.0
      MM=MAX0(M-1,1)
      MP=MIN0(M+1,MMAX)
      NM=MAX0(N-1,1)
      NP=MIN0(N+1,NMAX)
      HS=AMAX1(1.0,D(N,M)+BETAH*SE(N,M))
      F1=-AG/(HS*DQ)
      F2=1./(HS*DQ)
C        ESTABLISH WEIGHTING FACTORS FOR VELOCITY
      FD=1./AMAX1(1.414,FLOAT(MFLUX(N,M)+MFLUX(N,MM)))
      FX1=FLOAT(MFLUX(N,M))*FD
      IF(M.EQ.1)FX1=1.0
      FX2=FLOAT(MFLUX(N,MM))*FD
      FD=1./AMAX1(1.414,FLOAT(NFLUX(N,M)+NFLUX(NM,M)))
      FY1=FLOAT(NFLUX(N,M))*FD
      IF(N.EQ.1)FY1=1.0
      FY2=FLOAT(NFLUX(NM,M))*FD
C         LOOP THRU LAYERS
      DO 100 L=1,LAYRS
      FAV=0.
      FDV=0.
      Q=ABS(FLOAT(1-L)*DQ-HALFDQ)
c        mixing length for momentum diffusivity
      Z1=VONKAR*HS*(1.-Q)*Q**CR0
C       GET VELOCITY GRADIENT
      DUM=U(L,N,MM)-U(L+1,N,MM)
      IF(M.EQ.1)DUM=0.0
      DVM=V(L,NM,M)-V(L+1,NM,M)
      IF(N.EQ.1)DVM=0.0
      DELU=FX1*(U(L,N,M)-U(L+1,N,M))+FX2*(DUM)
      DELV=FY1*(V(L,N,M)-V(L+1,N,M))+FY2*(DVM)
      DUDZ= SQRT(DELU**2+DELV**2)*F2+E
C        GET DENSITY GRADIENT PER UNIT DENSITY
      IF(ICOUPL.GT.0)THEN
      DELTA=FRHO(S(L,N,M),T(L,N,M))-FRHO(S(L+1,N,M),T(L+1,N,M))
      ELSE
      DELRHO=0.0000
      DELTA=DELRHO*HS*DQ
      END IF
C        NEW RICHARDSON NUMBER: ALLOWS FOR NEGATIVE RI
      RICHNO=RAMPG*F1*DELTA/(DUDZ**2)
      RICHNO=AMAX1(RIMIN,AMIN1(RIMAX,RICHNO))
      RI(L,N,M)=RICHNO
      IF(RICHNO.GT.0.0)THEN
      FAV=CRICH(1)*(1.+CRICH(2)*RICHNO)**(-CRICH(3))
      FDV=CRICH(5)*(1.+CRICH(6)*RICHNO)**(-CRICH(7))
      ELSE
      FAV=CRICH(1)*(1.+CRICH(4)*RICHNO**2)
      FDV=CRICH(5)*(1.+CRICH(8)*RICHNO**2)
      ENDIF
 90   VISNEW=AV0+FAV*DUDZ*Z1**2
      DIFNEW=DV0+FDV*DUDZ*Z1**2
C        UPDATE
      IF(AV(L,N,M).EQ.0.0)THEN
      AV(L,N,M)=VISNEW
      ELSE
      AV(L,N,M)=SQRT(AV(L,N,M)*VISNEW)
      END IF
      IF(DV(L,N,M).EQ.0.0)THEN
      DV(L,N,M)=DIFNEW
      ELSE
      DV(L,N,M)=SQRT(DV(L,N,M)*DIFNEW)
      END IF
 100  IF(L.GT.1)AVMAX=AMAX1(AVMAX,AV(L,N,M))
C        UPDATE SCALE EXPLICIT INTERNAL-MODE TIMESTEP BASED ON DIFFUSION
      IF(HR.GT.HRCON2.AND.AVMAX.GT.0.0)DTIMAX=AMIN1(DTIMAX,
     1    .25*(DQ*HS)**2/AVMAX)
 120  CONTINUE
 130  CONTINUE
C        LOOP THRU OCEANIC BOUNDARIES
      IF(NUMOBC.LE.0)RETURN
      DO 200 IB=1,NUMOBC
      DO 200 N=NB1(IB),NB2(IB)
      DO 200 M=MB1(IB),MB2(IB)
      MP=M
      IF(IABS(ITPO(IB)).EQ.1)MP=M-ISIGN(1,ITPO(IB))
      NP=N
      IF(IABS(ITPO(IB)).EQ.2)NP=N-ISIGN(1,ITPO(IB))
      DO L=1,LAYRS
```

```
      AV(L,N,M)=AV(L,NP,MP)
      DV(L,N,M)=DV(L,NP,MP)
      ENDDO
  200 CONTINUE
      IF(IPRNT1.GT.0)WRITE(ISCR,*)'UT=',UT,' DTIMAX=',DTIMAX
      IF(IPRNT1.GT.0)WRITE(ISCR,*)'AVMAX=',AVMAX
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE WVERT
C        MARCH 1986   K.W. HESS   MEAD   VAX 11/750
C        PURPOSE - TO CALCULATE W, THE PRODUCT OF H AND DQ/DTI
      INCLUDE 'COMM20.FOR'
C        SET CONSTANTS
      F1=.25*DQ/DL
      KOCN=10*KOCNBC
C        LOOP THRU THE GRID MESH
      DO 140 M=1,MMAX
      NA=NAB(M)/1000
      NB=NAB(M)-1000*NA
      MM=MAX0(M-1,1)
      MP=MIN0(M+1,MMAX)
      IF(NA.GT.NB)GOTO 140
      DO 130 N=NA,NB
      IF(IFIELD(N,M)/10.LT.1.OR.IFIELD(N,M)/10.EQ.KOCNBC)GOTO 130
C         WATER GRID HERE
      HC=D(N,M)+SE(N,M)
      F2=F1/AREA(N,M)
      HMP=F2*BX(N,M)*(HC+D(N,MP)+SE(N,MP))
      HMM=F2*BX(N,MM)*(HC+D(N,MM)+SE(N,MM))
      NP=MIN0(N+1,NMAX)
      NM=MAX0(N-1,1)
      HNP=F2*BY(N,M)*(HC+D(NP,M)+SE(NP,M))
      HNM=F2*BY(NM,M)*(HC+D(NM,M)+SE(NM,M))
C        WC IS THE PRODUCT OF THE DEPTH AND THE DIMENSIONLESS VERT. VEL.
      WC(LBOT)=0.0
      DO 100 L=LAYRS,1,-1
  100 WC(L)=WC(L+1)-(HMP*(U(L,N,M)+U(L+1,N,M))-HMM*(U(L,N,MM)
     1 +U(L+1,N,MM))+HNP*(V(L,N,M)+V(L+1,N,M))-HNM*(V(L,NM,M)
     2 +V(L+1,NM,M)))
C        GET THE ADJUSTED DIMENSIONLESS VERTICAL VELOCITY
      DO 110 L=1,LBOT
  110 W(L,N,M)=WC(L)-WC(1)*FLOAT(LBOT-L)*DQ
  130 CONTINUE
  140 CONTINUE
      RETURN
      END
c=======================================================================
c     MECCA:  FORCES
C-----------------------------------------------------------------------
C
      SUBROUTINE FORCES

C        APRIL 1988   K. W. HESS   MEAD   VAX11/750
C        PURPOSE - TO SET THE INTERFACIAL STRESS TERMS:
C                  BOTTOM STRESS AND WIND STRESS AND AIR TEMPERATURE
      INCLUDE 'COMM20.FOR'
C        GET WIND STRESS AND AIR TEMPERATURE
      IF(NSTE.EQ.1)CALL ATMOS
C        UPDATE BOTTOM STRESS
      CALL BSTRES
C        UPDATE EXTERNAL PRESSURE GRADIENTS
      CALL ALGRAD
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE ATMOS
C        APRIL 1996   K. W. HESS   CEOB   SGI/IRIS
C        PURPOSE - TO SET THE WIND STRESS AND AIR TEMPERATURE
C        VARIABLES -
      INCLUDE 'COMM20.FOR'
      COMMON/METOX/TMET8(2),ITYPE1
      REAL*8 TMET8,FA8,FB8
      RAMPW=AMIN1(1.0,CUMDAY)
c        initialize
      IF(NSIGW.GT.0.AND.IENDWN.EQ.0)GOTO 100
      DPADX=0.0
      DPADY=0.0
      DO N=1,NMAX
      DO M=1,MMAX
      TSX(N,M)=0
      TSY(N,M)=0
      ENDDO
      ENDDO
      IF(NSIGW.EQ.0)RETURN
      IF(IENDWN.EQ.1)THEN
      WRITE(6,*)'     NO MORE GRIDDED WIND DATA'
      RETURN
      ENDIF
c        check for time of available data
  100 CONTINUE
      IF(IPRNT1.EQ.1)WRITE(6,105)TMET8(1),YT,TMET8(2)
  105 FORMAT(5X,'TMET8(1)=',F12.7,' YT=',F12.7,' TMET8(2)=',F12.7)
  110 IF(YT.GT.TMET8(2))THEN
c        save
      TMET8(1)=TMET8(2)
      DO N=1,NMAX
      DO M=1,MMAX
      FX(1,N,M)=FX(2,N,M)
      FY(1,N,M)=FY(2,N,M)
      ENDDO
      ENDDO
c        set default values
      DO N=1,NMAX
      DO M=1,MMAX
      FX(2,N,M)=0
      FY(2,N,M)=0
      ENDDO
      ENDDO
c        read next array
      I=2
      CALL RDWIND(I,IEND)
      IF(IEND.EQ.1)RETURN
      IF(IPRNT1.EQ.1)WRITE(6,105)TMET8(1),YT,TMET8(2)
      GOTO 110
      ENDIF
c        get interpolated stress
      FB8=(YT-TMET8(1))/(TMET8(2)-TMET8(1))
      F2=FB8
      FA8=1.-FB8
      F1=FA8
      IF(IPRNT1.EQ.1)WRITE(6,*)'    F1,2, ITYPE1=',F1,F2,ITYPE1
      DO N=1,NMAX
      DO M=1,MMAX
      TSX(N,M)=RAMPW*(F1*FX(1,N,M)+F2*FX(2,N,M))
      TSY(N,M)=RAMPW*(F1*FY(1,N,M)+F2*FY(2,N,M))
      IF(ITYPE1.EQ.1)THEN
      W1=TSX(N,M)
      W2=TSY(N,M)
      W10=SQRT(W1**2+W2**2)
      CDRGAW=CDR1+CDR2*W10
      TSX(N,M)=DENRAT*CDRGAW*W10*W1
      TSY(N,M)=DENRAT*CDRGAW*W10*W2
      WX(N,M)=W1
      WY(N,M)=W2
      ENDIF
      ENDDO
      ENDDO
c        adjust atmospheric pressure gradient
      DPADX=RAMPW*DPADX
      DPADY=RAMPW*DPADY
c        adjust for shallow water
      IF(DTAU2.LT.0.0)GOTO 130
      DO 120 M=1,MMAX
      DO 120 N=1,NMAX
      IF(MFLUX(N,M).EQ.1)THEN
      DEFF=AMIN1(D(N,M)+SE(N,M),D(N,M+1)+SE(N,M+1))
      FACTOR=AMAX1(0.0, AMIN1(1.,(DEFF-DTAU1)/(DTAU2-DTAU1)))
      TSX(N,M)=TSX(N,M)*FACTOR
      ENDIF
      IF(NFLUX(N,M).EQ.1)THEN
      DEFF=AMIN1(D(N,M)+SE(N,M),D(N+1,M)+SE(N+1,M))
      FACTOR=AMAX1(0.0, AMIN1(1.,(DEFF-DTAU1)/(DTAU2-DTAU1)))
      TSY(N,M)=TSY(N,M)*FACTOR
      ENDIF
  120 CONTINUE
  130 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE BSTRES
C        SEPTEMBER 1996   K. W. HESS
C        PURPOSE - TO UPDATE THE BOTTOM STRESS.  TBX IS AT LOCATION
C                  OF UH, TBY IS AT VH.
C        VARIABLES -
C             IBOTV = BOTTOM CONDITION INDEX
C                     0 = NON-SLIP
C                     1 = SLIP, FIRST ORDER
C                     2 = SLIP, SECOND ORDER
C                     3 = LOG LAYER
      INCLUDE 'COMM20.FOR'
      DIMENSION UTO(NSIZE,MSIZE),VTO(NSIZE,MSIZE)
      DATA Z0/0.003/
C        SELECT WEIGHTING FACTOR FOR OLD VS. NEW STRESS
      FNEW=1.
      FOLD=1.-FNEW
C        COMPUTE THE WEIGHTING FACTORS FOR NO-SLIP AND LOG-LAYER STRESS
      GAMMA=1.
      IF(IBOTV.EQ.0)GAMMA=.0
      IF(IBOTV.EQ.3)GAMMA=.5
C        COMPUTE THE TOTAL BOTTOM VELOCITY
      DO M=1,MMAX
      DO N=1,NMAX
      UTO(N,M)=UE(N,M)+GAMMA*U(LBOT,N,M)+(1.-GAMMA)*U(LBOT-1,N,M)
      VTO(N,M)=VE(N,M)+GAMMA*V(LBOT,N,M)+(1.-GAMMA)*V(LBOT-1,N,M)
      ENDDO
      ENDDO
C        COMPUTE THE EFFECTIVE BOTTOM DRAG COEFFICIENT, PHI
      DO 110 M=1,MMAX
      DO 110 N=1,NMAX
      IF(IFIELD(N,M).LT.10)GOTO 110
      MM=MAX0(M-1,1)
      NM=MAX0(N-1,1)
C        DRAG COEFFICIENT AT GRID CENTER BASED ON LOCAL VELOCITY
      UB=(UTO(N,M)+UTO(N,MM))/(FLOAT(MFLUX(N,M)+MFLUX(N,MM))+E)
      VB=(VTO(N,M)+VTO(NM,M))/(FLOAT(NFLUX(N,M)+NFLUX(NM,M))+E)
c        zero bottom velocity
      IF(IBOTV.EQ.0)THEN
      PHI(N,M)=AV(LAYRS,N,M)/((D(N,M)+E)*DQ)
c        log layer
      ELSE IF(IBOTV.EQ.3)THEN
      PHI(N,M)=(0.4/ALOG(.5*D(N,M)*DQ/Z0))**2*SQRT(UB**2+VB**2)
      ELSE
      PHI(N,M)=CDWB1+CDWB2*SQRT(UB**2+VB**2)
      ENDIF
  110 CONTINUE
C         STRESS
      DO 140 M=1,MMAX
      MP=MIN0(M+1,MMAX)
      DO 140 N=1,NMAX
      IF(IFIELD(N,M).LT.10)GOTO 140
      NP=MIN0(N+1,NMAX)
      TNEWX=.5*(PHI(N,M)+PHI(N,MP))*UTO(N,M)
      TNEWY=.5*(PHI(N,M)+PHI(NP,M))*VTO(N,M)
C        TIME AVERAGE
      TBX(N,M)=FOLD*TBX(N,M)+FNEW*TNEWX
      TBY(N,M)=FOLD*TBY(N,M)+FNEW*TNEWY
  140 CONTINUE
  150 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE ALGRAD
C        APRIL 1988   HESS   MEAD   VAX
C        PURPOSE - TO SET ALL HORIZONTAL DENSITY GRADIENTS
      INCLUDE 'COMM20.FOR'
C        SET THE INTERNAL PRESSURE RAMP FACTOR
      RAMPG=AMAX1(0.,AMIN1(1.,(HR1-HRCON1)/(HRCON2-HRCON1)))
C        SET INITIAL DENSITY GRADIENTS
      IF(KONCEN.EQ.0)GOTO 330
      DO 320 M=1,MMAX
      DO 320 N=1,NMAX
      IF(IFIELD(N,M).LT.10)GOTO 320
      IBARR=MOD(IFIELD(N,M),10)
      IF(IBARR.EQ.3)GOTO 320
      CALL GRADP(N,M,1)
  320 CONTINUE
  330 CONTINUE
      RETURN
      END
C=======================================================================
C     MECCA FILE : MREAD
C-----------------------------------------------------------------------
C
```

```
      SUBROUTINE READZ

C        APRIL 1988   K. W. HESS   MEAD   VAX 11/750
C        PURPOSE - TO READ IN RUN-TIME FILE NAMES IN AN INTERACTIVE
C                   MODE
      INCLUDE 'COMM20.FOR'
C        ZERO OUT INITIAL ARRAYS
      CALL ZEROS
C        OPEN FILE AND READ THE CONTROL FILE
      LUCON=LUKB
      RAD=PI/180.
      DENRAT=RHOA/RHOW
      CALL RDCON1(JTEST,IVER)
      CALL RDCON2(JTEST,IVER)
      CALL RDCON3(JTEST,IVER)
C        OPEN OUTPUT PRINT FILE
      CALL FFOPEN(IO,FPRINT)
C        OPEN GRAPHING FILES
      IF(IGPH.GT.0)CALL FFOPEN(LUGRF,FGRAPH)
C        OPEN GEOGRAPHY FILE
      CALL FFOPEN(LUCON,FGEO)
      CALL RDGEO(LUCON)
      CLOSE (LUCON)
C        GET INITIAL CONDITIONS
      NSTI=0
      NSTE=0
      NSTIT=0
      NSTET=0
      HR0=0.0
      UT1=UT0
      YEAR1=YEAR
      IF(ICS.EQ.1)THEN
      CALL RDICS
      HR0=FLOAT(NSTET)*DTE/3600.
      CUMDAY=HR0/24.
      NSTIT=NSTET/ISPLIT
      ENDIF
      RETURN
      END
C
C---------------------------------------------------------------------
C
      SUBROUTINE RDCON1(JTEST,IVER)
C        OCTOBER 1984   K. HESS   MEAD    VAX 11/70 (REV 8-89)
C        PURPOSE - TO READ IN THE CONTROL FILE
      INCLUDE 'COMM20.FOR'
      DIMENSION MXNX(NPMSIZ)
C        THIS IS FOR CON FILE VERSIONS 4 - 5   (8-89)
      JVER1=4
      JVER2=5
      IUNIT=LUCON
C        READ DATA FILE HERE
      READ(IUNIT,100)CTITLE
 100  FORMAT(8A10)
      WRITE(ISCR,101)CTITLE
 101  FORMAT(5X,'CON FILE TITLE: ',8A10)
      READ(IUNIT,103)FGEO
 103  FORMAT(A40)
      READ(IUNIT,*)IVER,JTEST,KTEST
      WRITE(ISCR,*)IVER,JTEST,KTEST
      WRITE(ISCR,105)FGEO
 105  FORMAT(5X,'GEOGRAPHY FILE: ',A40)
C
C        READ MODEL CONFIGURATION DATA
      IF(JTEST.EQ.1)WRITE(ISCR,108)
 108  FORMAT(1X,'MODEL CONFIGURATION PARAMETERS')
      READ(IUNIT,1002)
 1001 FORMAT(1X)
 1002 FORMAT(/,1X)
 1100 READ(IUNIT,*,ERR=1108)HRMAX,HROUT,HROUT0,HRSAVE
      IF(JTEST.EQ.1)PRINT*,HRMAX,HROUT,HROUT0,HRSAVE
      GOTO 1109
 1108  WRITE(ISCR,1107)
 1107  FORMAT(5X,'*** (RDCON) ERROR READING HRMAX, ETC. ***')
 1109  CONTINUE
      READ(IUNIT,1001)
      READ(IUNIT,*,ERR=1118)DTE,ISPLIT,LAYRS
      DTI=ISPLIT*DTE
      IF(JTEST.EQ.1)PRINT*,DTE,ISPLIT,LAYRS
      GOTO 1119
 1118  WRITE(ISCR,1117)
 1117  FORMAT(5X,'*** (RDCON) ERROR READING DTE, ISPLIT, ETC. ***')
 1119 CONTINUE
      NDZM=LSIZE-1
c     IF(LAYRS.GE.3.AND.LAYRS.LE.NDZM)GOTO 1122
c     WRITE(ISCR,1120)LAYRS,NDZM
c1120  FORMAT(1X,'*** ERROR: LAYRS=',I3,' IS NOT BETWEEN 3 AND ',I3)
c     STOP
 1122 CONTINUE
C        TURBULENCE VARIABLES
      DHAH=1.0
      READ(IUNIT,1001)
      READ(IUNIT,*,ERR=1217)AH00,AH0,CAH,DHAH,RIMIN,RIMAX
      IF(JTEST.EQ.1)PRINT*,AH00,AH0,CAH,DHAH,RIMIN,RIMAX
      GOTO 1220
 1217 WRITE(ISCR,1218)
 1218 FORMAT(5X,'*** (RDCON) ERROR READING AHO, CAH  ***')
 1220 READ(IUNIT,*,ERR=1222)AV00,AV0,(CRICH(I),I=1,4)
      IF(JTEST.EQ.1)PRINT*,AV00,AV0,(CRICH(I),I=1,4)
      GOTO 1224
 1222 WRITE(ISCR,1223)
 1223 FORMAT(5X,'*** (RDCON) ERROR READING AV0, CRICH1, ETC ***')
 1224 READ(IUNIT,*,ERR=1226)DV00,DV0,(CRICH(I),I=5,8)
      IF(JTEST.EQ.1)PRINT*,DV00,DV0,(CRICH(I),I=5,8)
      GOTO 1230
 1226 WRITE(ISCR,1227)
 1227 FORMAT(5X,'*** (RDCON) ERROR READING DV0 CRICH3, ETC  ***')
C        UPDATE INTERVALS
 1230 READ(IUNIT,*,ERR=1255)IHVISC,IVISC,CR0
      IF(JTEST.EQ.1)PRINT*,IHVISC,IVISC,CR0
      GOTO 128
 1255 WRITE(ISCR,1256)
 1256 FORMAT(5X,'*** (RDCON) ERROR READING IHVISC, IVISC. ***')
C        DRAG COEFFICIENTS
 128  READ(IUNIT,1001)
      READ(IUNIT,*,ERR=1231)CDWB1,CDWB2,CDRGWS,CDR1,CDR2
      IF(JTEST.EQ.1)PRINT*,CDWB1,CDWB2,CDRGWS,CDR1,CDR2
      GOTO 1234
 1231 WRITE(ISCR,1232)
 1232 FORMAT(5X,'*** (RDCON) ERROR READING CDWB1, CDWB2,CDRGWS  ***')
C        HEATING CONSTANTS
 1234 READ(IUNIT,1001)
      READ(IUNIT,*,ERR=1240)ALB,D10PCT
      IF(JTEST.EQ.1)PRINT*,ALB,D10PCT
      GOTO 1245
 1240 WRITE(ISCR,1244)
 1244 FORMAT(5X,'*** (RDCON) ERROR READING CLOUD, RH ***')
C        SWITCHES
 1245 READ(IUNIT,1001)
      READ(IUNIT,*,ERR=1260)ICOR,IBETAA,IBETAP,IBETAH
      IF(JTEST.EQ.1)PRINT*,ICOR,IBETAA,IBETAP,IBETAH
      GOTO 1270
 1260 WRITE(ISCR,1262)
 1262 FORMAT(5X,'*** (RDCON) ERROR READING ICOR, ETC. ***')
 1270 READ(IUNIT,*,ERR=1272)IEXTRN,INTER,KONCEN,ICOUPL
      IF(JTEST.EQ.1)PRINT*,IEXTRN,INTER,KONCEN,ICOUPL
      GOTO 1280
 1272 WRITE(ISCR,1274)
 1274 FORMAT(5X,'*** (RDCON) ERROR READING IEXTRN, ETC. ***')
 1280 READ(IUNIT,*,ERR=1285)ITOPV,IBOTV,IHEAT,ICPOS
      IF(JTEST.EQ.1)PRINT*,ITOPV,IBOTV,IHEAT,ICPOS
      GOTO 1290
 1285 WRITE(ISCR,1287)
 1287 FORMAT(5X,'*** (RDCON) ERROR READING IHEAT, ETC. ***')
 1290 CONTINUE
C        READ PRINT PARAMETERS
 136  CONTINUE
      IF(JTEST.EQ.1)WRITE(ISCR,135)
 135  FORMAT(1X,'MODEL PRINT PARAMETERS')
      READ(IUNIT,1002)
C        PLAN VIEW VARIABLES
      READ(IUNIT,*,ERR=137)(JPRNT(I),I=1,13)
      IF(JTEST.EQ.1)PRINT*,(JPRNT(I),I=1,13)
      GOTO 1390
 137  WRITE(ISCR,138)
 138  FORMAT(5X,'*** (RDCON) ERROR READING JPRNT, ETC. ***')
C         PAGE FORMATS
 1390 READ(IUNIT,1001)
      READ(IUNIT,*,ERR=1392)KPRNT1,KPRNT2
      IF(JTEST.EQ.1)PRINT*,KPRNT1,KPRNT2
      GOTO 140
 1392  WRITE(ISCR,1394)
 1394  FORMAT(5X,'*** (RDCON) ERROR READING IEXTRN, ETC. ***')
C        PRINT AT ALL LEVELS
 140  READ(IUNIT,1001)
      READ(IUNIT,*,ERR=141)NPRMN
      IF(JTEST.EQ.1)PRINT*,NPRMN
      GOTO 144
 141  WRITE(ISCR,142)
 142  FORMAT(5X,'*** (RDCON) ERROR READING NPRMN, ETC. ***')
 144  IF(NPRMN.LE.NDPRN)GOTO 146
      WRITE(ISCR,145)NPRMN,NDPRNN
 145  FORMAT(1X,'*** ERROR: NPRMN=',I2,' GREATER THAN NDPRN=',I2)
      STOP
 146  CONTINUE
      IF(NPRMN.GT.0)READ(IUNIT,*)(IPRMN(J),J=1,NPRMN)
      IF(NPRMN.GT.0)PRINT*,(IPRMN(J),J=1,NPRMN)
C        CELLS IN LONGITUDINAL SECTION
      READ(IUNIT,1001)
      READ(IUNIT,*,ERR=1400)ISLICE
      IF(JTEST.EQ.1)PRINT*,ISLICE
      GOTO 1406
 1400 WRITE(ISCR,1402)
 1402 FORMAT(1X,'*** ERROR READING ISLICE ***')
 1410 IF(ISLICE.LE.NDSLC1)GOTO 1406
      WRITE(ISCR,1404)ISLICE,NDSLC1
 1404 FORMAT(1X,'*** ERROR: ISLICE=',I2,' GREATER THAN NDSLC1=',I2)
       STOP
 1406  CONTINUE
      IF(ISLICE.EQ.0)GOTO 180
      DO 170 I=1,ISLICE
      READ(IUNIT,*)JSLICE(I),(MXNX(J),J=1,JSLICE(I))
      IF(JTEST.EQ.1)PRINT*,JSLICE(I),(MXNX(J),J=1,JSLICE(I))
      IF(JSLICE(I).LE.NDSLC2)GOTO 155
      WRITE(ISCR,152)JSLICE(I),NDSLC2
 152  FORMAT(1X,'*** ERROR: JSLICE=',I2,' GREATER THAT NDSLC2=',I2)
      STOP
 155  NTOTAL=0
      DO 160 J=1,JSLICE(I)
      MSLICE(J,I)=MXNX(J)/1000
      NSLICE(J,I)=MXNX(J)-1000*(MXNX(J)/1000)
      IF(J.GT.1)THEN
      NDIF=IABS(NSLICE(J,I)-NSLICE(J-1,I))
      MDIF=IABS(MSLICE(J,I)-MSLICE(J-1,I))
      NTOTAL=NTOTAL+1
      IF(.NOT.((MDIF.EQ.0.AND.NDIF.GT.0).OR.
     1  (NDIF.EQ.0.AND.MDIF.GT.0).OR.
     2  (NDIF.EQ.MDIF)))THEN
      WRITE(ISCR,157 )I,MDIF,NDIF
 157  FORMAT(1X,'***ERROR: IN SLICE NO.=',I2,' MDIF=',
     1 I3,' AND NDIF=',I3)
      STOP
      END IF
      END IF
 160  CONTINUE
 166  CONTINUE
 170  CONTINUE
C        CELLS FOR LATER GRAPHING
 180  READ(IUNIT,1001)
      READ(IUNIT,*,ERR=1411)IGPH,NSTGPH,IGPHOP
      IF(JTEST.EQ.1)PRINT*,IGPH,NSTGPH,IGPHOP
      GOTO 1420
 1411 WRITE(ISCR,1412)
 1412 FORMAT(5X,'*** ERROR READING IGPH,NGPH ***')
 1420 IF(IGPH.LE.0)GOTO 191
      IF(IGPH.LE.NDGPH)GOTO 1430
      WRITE(ISCR,1424)IGPH,NDGPH
 1424 FORMAT(1X,'*** ERROR: IGPH=',I2,' GREATER THAN NDGPH=',I2)
      STOP
 1430 JGPH=MIN0(IGPH,36)
      DO 190 I=1,IGPH
      IF(I.GT.JGPH)READ(IUNIT,*)X
      IF(I.LE.JGPH)READ(IUNIT,*,ERR=185)LGPH(I),MGPH(I),NGPH(I),ITYP(I)
      IF(JTEST.EQ.1)PRINT*,LGPH(I),MGPH(I),NGPH(I),ITYP(I)
      GOTO 190
 185  WRITE(ISCR,186)
 186  FORMAT(5X,'*** (RDCON) ERROR READING LGPH, MGPH, ETC ***')
 190  CONTINUE
 191  CONTINUE
      IGPH=JGPH
      RETURN
      END
C
C---------------------------------------------------------------------
C
      SUBROUTINE RDCON2(JTEST,IVER)

C        JUNE 1994   K. HESS   CEOB     HP9000
C        PURPOSE - TO READ IN THE ENVIRONMENTAL DATA AS SEPARATE FILES
      INCLUDE 'COMM20.FOR'
      CHARACTER*40 FDATA
      IUNIT=LUCON
```

```
      IU=LUCON
C        READ RUN START DATE
      IF(JTEST.EQ.1)WRITE(ISCR,90)
 90   FORMAT(1X,'TIME VARIABLE INPUTS')
      READ(IUNIT,1002)
 1001 FORMAT(1X)
 1002 FORMAT(/,1X)
      READ(IUNIT,*,ERR=95 )IYEAR,MONTH,IDAY,IHOUR,IMIN
      IF(IYEAR.LT.100)IYEAR=IYEAR+1900
      YEAR=IYEAR
      YEAR0=YEAR
      YEAR1=YEAR
      IF(JTEST.EQ.1)PRINT*,'YEAR=',IYEAR,MONTH,IDAY,IHOUR,IMIN
      GOTO 98
 95   WRITE(ISCR,96 )
 96   FORMAT(5X,'*** (RDCON) ERROR READING IYEAR, ETC. ***')
      STOP
 98   CALL JULIAN(IYEAR,MONTH,IDAY,UT,NDAYMO)
      UT0=UT+(FLOAT(IHOUR)+FLOAT(IMIN)/60.)/24.
      IF(JTEST.EQ.1)PRINT*,'UT0=',UT0
C
C        READ TIDAL WATER LEVEL DATA
      WRITE(ISCR,*)' '
      WRITE(ISCR,*)'WATER LEVEL DATA'
      CALL IRR(IU,ISCR,LUTID,IENDTD,DTID,YTID,NSIGT,NDTID2,TDLEV)
C
C        READ WIND DATA:
      WRITE(ISCR,*)' '
      WRITE(ISCR,*)'WIND DATA'
      READ(IUNIT,1001)
      READ(IUNIT,*)NSIGW
      IENDWN=0
      IF(NSIGW.EQ.0)THEN
      WRITE(ISCR,*)' '
      WRITE(ISCR,*)'     NO INPUT DATA '
      IENDWN=1
      GOTO 300
      ENDIF
      READ(IUNIT,100)FDATA
 100  FORMAT(A40)
      WRITE(ISCR,130)FDATA,NSIGW,LUWND
 130  FORMAT(5X,'FILE NAME=',A40,/,5x,'NSIG=',I2,'  LUT=',I2)
      CALL FUOPEN(LUWND,FDATA)
      DO 220 I=1,2
 220  CALL RDWIND(I,IEND)
      IENDWN=IEND
 300  CONTINUE
C
C        READ IN RIVER FLOWRATE DATA
      WRITE(ISCR,*)' '
      WRITE(ISCR,*)'RIVER FLOWRATE DATA'
      CALL IRR(IU,ISCR,LURIV,IENDRV,DRIV,YRIV,NSIGR,NDRIV2,QRIV)

C        READ IN OCEANIC SALINITY CONCENTRATION
      WRITE(ISCR,*)' '
      WRITE(ISCR,*)'OCEAN SALINITY DATA'
      CALL IRR(IU,ISCR,LUSAL,IENDSO,DSAL,YSAL,NSIGS,NDOCN2,SALOCN)

C        READ IN OCEANIC TEMPERATURES
      WRITE(ISCR,*)' '
      WRITE(ISCR,*)'OCEAN TEMPERATURE DATA'
      CALL IRR(IU,ISCR,LUOCT,IENDTO,DOTP,YOTP,NSIGTO,NDOCN2,TMPOCN)
C
C        READ IN RIVER TEMPERATURES
      WRITE(ISCR,*)' '
      WRITE(ISCR,*)'RIVER TEMPERATURE DATA'
      CALL IRR(IU,ISCR,LURVT,IENDRT,DRVT,YRVT,NSIGRT,NDRIV2,TRIV)

C        READ IN ADDITIONAL MET DATA
      WRITE(ISCR,*)' '
      WRITE(ISCR,*)'ADDITIONAL MET DATA'
      CALL IRR(IU,ISCR,LUMET,IENDMT,DMET,YMET,NSIGM,NDMET2,VMET)
C
C        READ IN INITIAL CONDITIONS INDEX AND FILE
      READ(IUNIT,1001)
      READ(IUNIT,*)ICS
      WRITE(ISCR,510)ICS
 510  FORMAT(/,1X,'INITIALIZATION.  ICS=',I2)
      IF(ICS.EQ.0)GOTO 600
      READ(IUNIT,100)FINIT
      WRITE(ISCR,520)FINIT
 520  FORMAT(1X,' INITIALIZATION FILE NAME=',A40)
 600  CONTINUE
      RETURN
      END

C-----------------------------------------------------------------------
C
      SUBROUTINE RDCON3(JTEST,IVER)
C        OCTOBER 1984   K. HESS   MEAD    VAX 11/70 (REV 8-89)
C        PURPOSE - TO READ IN THE CONTROL FILE
      INCLUDE 'COMM20.FOR'
      DIMENSION MXNX(NPMSIZ)
      IUNIT=LUCON
c      READ OUTPUT FILE NAMES
      READ(IUNIT,1001)
 1001 FORMAT(1X)
      READ(IUNIT,440)FPRINT
      READ(IUNIT,440)FGRAPH
      READ(IUNIT,440)FMED
      WRITE(6,*)'OUTPUT FILE NAMES'
      WRITE( 6,440)FPRINT
      WRITE( 6,440)FGRAPH
      WRITE( 6,440)FMED
 440  FORMAT(10X,A40)
C        READ END OF DATA STATEMENT
      READ(IUNIT,1001)
      IF(JTEST.EQ.1)WRITE(ISCR,1303)
 1303 FORMAT(1X,'END OF CONTROL FILE READ')
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE JULIAN(IY,MO,IDAY,UT)
C         JAN 1995   HESS   CEOB   SGI 4D
c         purpose - to convert year, month, etc to a Julian date and
c                   number of hours from start of year
C          NOTE: UT IS NOS CONVENTION, SO NOON ON JANUARY 1
C                         IS UT=1.50
      DIMENSION NMON(12,2)
      DATA NMON/0,31,59,90,120,151,181,212,243,273,304,334,
     1          0,31,60,91,121,152,182,213,244,274,305,335/
C      CONVERT TO 4-DIGIT IF NOT
      IYEAR=IYR4(IY)
c        look for leap year
      L=1
      IF(MOD(IYEAR,4).EQ.0.AND.(.NOT.(MOD(IYEAR,100).EQ.0.AND.
     1 MOD(IYEAR,400).NE.0)))L=2
      UT=NMON(MO,L)+IDAY
      RETURN
      END
C-----------------------------------------------------------------------
      FUNCTION IYR4(IY)
      IYR4=IY
      IF(IY.LT.100.AND.IY.GT.50)IYR4=IY+1900
      IF(IY.LT.50)IYR4=IY+2000
      RETURN
      END
C-----------------------------------------------------------------------

      SUBROUTINE FFOPEN(IUNIT,FNAME)
C        PURPOSE - TO OPEN A FORMATTED FILE
      CHARACTER FNAME*(*)
 100  OPEN(UNIT=IUNIT,FILE=FNAME,FORM='FORMATTED',STATUS='UNKNOWN',
     1 IOSTAT=IER,ERR=110)
      GOTO 140
 110  WRITE(6,120)FNAME
 120  FORMAT(1X,'***ERROR IN FILE : ',A50,/,x,'***RUN TERMINATED')
      STOP
 140  RETURN
      END

C-----------------------------------------------------------------------

      SUBROUTINE FUOPEN(IUNIT,FNAME)
C        PURPOSE - TO OPEN A FORMATTED FILE
      CHARACTER FNAME*(*)
 100  OPEN(UNIT=IUNIT,FILE=FNAME,FORM='UNFORMATTED',STATUS='UNKNOWN',
     1 IOSTAT=IER,ERR=110)
      GOTO 140
 110  WRITE(6,120)FNAME
 120  FORMAT(1X,'***ERROR IN FILE : ',A50,/,x,'***RUN TERMINATED')
      STOP
 140  RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE RDGEO(IGEO)
C        OCTOBER 1984   K. W. HESS   ASIC/MEAD    VAX 11/750
C        PURPOSE - TO READ IN PARAMETERS, PLUS IFIELD, DEPTH, AND
C                  FLAG DATA.
C        VARIABLES -
C                 IAH = INDEX TO READ HORIZ. EDDY VISCOSITY: 0=NO
C               IGRID = INDEX TO READ STRETCHED GRIDS:0=NO, 1=READ XE,YE
C                       2=READ NSEGX, ETC.
C              NUMBXY = NUMBER OF CHANNEL GRIDS
C               NO,NR = NUMBER OF OCEAN, RIVER BOUNDARY CELLS AS
C                       DETERMINED BY IFIELD
C            NO1,NRV1 = NUMBER OF OCEAN, RIVER BOUNDARY CELLS AS
C                       DETERMINED BY NUMOBC, NUMRIV
C               XC(,) = DISTANCE FROM ORIGIN TO GRID CENTER
C               XL(,) = DISTANCE TO LOWER SIDE OF GRID M
      INCLUDE 'COMM20.FOR'
      DIMENSION NUM(NPMSIZ),XL(NPMSIZ),YL(NPMSIZ)
      CHARACTER*10 FORM,GTITLE(6)
      DATA NDX,NDY/MSIZE,NSIZE/
C        THIS IS VERSION 2  (8-89)
      JVER=2
C        READ IN MECCA FILE HEADER BLOCK
 200  READ(IGEO,202)(GTITLE(N),N=1,6)
 202  FORMAT(1X,6A10)
      READ(IGEO,*)J,ITEST
      IF(ITEST.EQ.1)WRITE(ISCR,2200)(GTITLE(N),N=1,6),J
 2200 FORMAT(/,5X,'GEOGRAPHY FILE',/,1X,'TITLE=',6A10,'   J=',I3)
      IF(J.NE.JVER)THEN
      WRITE(ISCR,204)J,JVER
 204  FORMAT(1X,'INPUT FILE IS VERSION=',I2,'  RDGEO IS VERSION=',I2,
     1 '   RUN IS STOPPED')
      STOP
      END IF
C
C        GRID PARAMETERS
      READ(IGEO,1002)
 1001 FORMAT(1X)
 1002 FORMAT(/,1X)
      READ(IGEO,*,ERR=222)NMAX,MMAX,DL
      IF(ITEST.EQ.1)WRITE(ISCR,2220)NMAX,MMAX,DL
 2220 FORMAT(1X,'NMAX,MMAX,DL=',2I5,E10.5)
      GOTO 226
 222  WRITE(ISCR,224)
 224  FORMAT(1X,'*** (RDGEO) ERROR READING NMAX,MMAX,DL ***')
 226  IF(MMAX.LE.NDX.AND.NMAX.LE.NDY)GOTO 228
      WRITE(ISCR,227)NMAX,NDY,MMAX,NDX
 227  FORMAT(1X,'*** ERROR: NMAX=',I3,' GT NDY=',I3,' OR MMAX=',I3,
     1 ' GT NDX=',I3)
      STOP
 228  CONTINUE
      READ(IGEO,1001)
      READ(IGEO,*,ERR=232)NCOR,MCOR,BSNLAT,BSNLON,BSNANG
      IF(ITEST.EQ.1)WRITE(ISCR,2260)NCOR,MCOR,BSNLAT,BSNLON,BSNANG
 2260 FORMAT(1X,'NCOR=',2I5,3F10.3)
      GOTO 234
 232  WRITE(ISCR,233)
 233  FORMAT(1X,'**** RDGEO. PROBLEM READING LINE NCOR,MCOR, ETC ***')
      RETURN

C        OCEAN BOUNDARY CONDITIONS
 234  READ(IGEO,1001)
      READ(IGEO,*)NUMOBC
      NUMBC=NUMOBC
      IF(ITEST.EQ.1)WRITE(ISCR,2340)NUMOBC
 2340 FORMAT(1X,'NUMOBC=',I3)
 237  IF(NUMOBC.LE.0)GOTO 242
      DO 240 I=1,NUMOBC
      READ(IGEO,*)MB1(I),MB2(I),NB1(I),NB2(I),ITPO(I),JTPO(I),ISET1(I),
     1 ISET2(I)
c     READ(IGEO,*)MB1(I),MB2(I),NB1(I),NB2(I),ITPO(I),ISET1(I),ISET2(I)
      IF(ITEST.EQ.1)WRITE(ISCR,2360)MB1(I),MB2(I),NB1(I),NB2(I),ISET1(I)
     1 ,ISET2(I)
 2360 FORMAT(1X,'MB1=',6I5)
      IF(.NOT.(MB1(I).LE.MB2(I).AND.NB1(I).LE.NB2(I)))THEN
      WRITE(ISCR,238)I,MB1(I),MB2(I),NB1(I),NB2(I)
 238  FORMAT(/,1X,'*** ERROR *** AT OCEAN BOUNDARY # ',I2,' MB1=',I3,
     1 ' IS GT MB2=',I3,/,'  OR NB1=',I3,' IS GT NB2=',I3)
      STOP
      END IF
      IF(.NOT.(MB1(I).EQ.MB2(I).OR.NB1(I).EQ.NB2(I)))THEN
      WRITE(ISCR,2238)I,MB1(I),MB2(I),NB1(I),NB2(I)
 2238 FORMAT(/,1X,'*** ERROR *** AT OCEAN BOUNDARY # ',I2,' MB1=',I3,
     1 ' IS .NE. MB2=',I3,/,'  OR NB1=',I3,' IS .NE. NB2=',I3)
```

```
      STOP
      END IF
  240 CONTINUE
C
C        READ IN RIVER BOUNDARIES
  242 READ(IGEO,1001)
      READ(IGEO,*)NUMRIV
      IF(ITEST.EQ.1)WRITE(ISCR,2450)NUMRIV
 2450 FORMAT(1X,'NUMRIV=',I5)
      IF(NUMRIV.LE.0)GOTO 248
      DO 246 I=1,NUMRIV
      READ(IGEO,*)MR1(I),MR2(I),NR1(I),NR2(I),ITPR(I),JTPR(I),ISETR(I)
      IF(ITEST.EQ.1)WRITE(ISCR,2460)MR1(I),MR2(I),NR1(I),NR2(I),ITPR(I)
     1 ,JTPR(I),ISETR(I)
 2460 FORMAT(1X,'MR1=',7I5)
      JTPR(I)=MAX0(1,MIN0(2,JTPR(I)))
      IF(.NOT.(MR1(I).LE.MR2(I).AND.NR1(I).LE.NR2(I)))THEN
       WRITE(ISCR,1238)I,MR1(I),MR2(I),NR1(I),NR2(I)
 1238  FORMAT(/,1X,'*** ERROR *** AT RIVER BOUNDARY # ',I2,' MB1=',I3,
     1 ' IS GT MB2=',I3,/,'  OR NB1=',I3,' IS GT NB2=',I3)
      STOP
      END IF
      IF(.NOT.(MR1(I).EQ.MR2(I).OR.NR1(I).EQ.NR2(I)))THEN
       WRITE(ISCR,1239)I,MR1(I),MR2(I),NR1(I),NR2(I)
 1239  FORMAT(/,1X,'*** ERROR *** AT RIVER BOUNDARY # ',I2,' MB1=',I3,
     1 ' IS .NE. MB2=',I3,/,'  OR NB1=',I3,' IS .NE. NB2=',I3)
      STOP
      END IF
  246 CONTINUE
C
C        READ THE CELL STATUS FIELD (IFIELD)
  248 READ(IGEO,1001)
      READ(IGEO,2490,ERR=249)FORM
 2490 FORMAT(1X,A10)
      IF(ITEST.EQ.1)WRITE(ISCR,2491)FORM
 2491 FORMAT(1X,'CELL STATUS: FORM=',A10)
      GOTO 2494
  249 WRITE(ISCR,2492)
 2492 FORMAT(1X,'*** ERROR READING FORM ***')
 2494 READ(IGEO,*)NPERL,KOCNBC
      IF(ITEST.EQ.1)WRITE(ISCR,2496)NPERL,KOCNBC
 2496 FORMAT(1X,'NPERL=',2I5)
      IF(KOCNBC.GE.4.AND.KOCNBC.LE.8)GOTO 2502
      WRITE(ISCR,2501)KOCNBC
 2501 FORMAT(1X,'*** ERROR: KOCNBC=',I3,' NOT BETWEEN 4 AND 8')
      STOP
 2502 IF(ITEST.EQ.1)WRITE(ISCR,*)' IFIELD'
      KMAX=1+(NMAX-1)/NPERL
      DO 250 K=1,KMAX
      N1=1+(K-1)*NPERL
      N2=MIN0(N1+NPERL-1,NMAX)
      DO 250 M=1,MMAX
      READ(IGEO,FORM)(IFIELD(N,M),N=N1,N2)
      IF(ITEST.EQ.1)WRITE(ISCR,2497)(IFIELD(N,M),N=N1,N2)
 2497 FORMAT(40I2)
  250 CONTINUE
C
C        READ THE DEPTHS
      READ(IGEO,1001)
      READ(IGEO,2490)FORM
      IF(ITEST.EQ.1)WRITE(ISCR,2642)FORM
 2642 FORMAT(1X,'DEPTHS: FORM=',A10)
      READ(IGEO,*,ERR=264)NCOL2,HMSL,CON2M
      IF(ITEST.EQ.1)WRITE(ISCR,2643)NCOL2,HMSL,CON2M
 2643 FORMAT(1X,'NCOL2=',I5,2F10.3)
      GOTO 266
  264 WRITE(ISCR,265)
  265 FORMAT(1X,'*** ERROR READING NCOL2,HMSL,CON2M ***')

  266 NSWEEP=1+(NMAX-1)/NCOL2
      DO 285 NN=1,NSWEEP
      N1=1+NCOL2*(NN-1)
      N2=MIN0(NMAX,N1+NCOL2-1)
      DO 280 M=1,MMAX
      READ(IGEO,FORM)(NUM(N),N=N1,N2)
      IF(ITEST.EQ.1)WRITE(ISCR,FORM)(NUM(N),N=N1,N2)
C        CONVERT DEPTHS TO METERS AND ADD HMSL
      DO 275 N=N1,N2
      D(N,M)=0.0
  275 IF(IFIELD(N,M).GT.0)D(N,M)=FLOAT(NUM(N))*CON2M+HMSL
  280 CONTINUE
  285 CONTINUE
C
C        SET GRID WIDTHS AND AREA
      READ(IGEO,1001)
      READ(IGEO,*,ERR=301)NUMBXY
      IF(ITEST.EQ.1)WRITE(ISCR,3301)NUMBXY
 3301 FORMAT(1X,'WIDTH DATA: NUMBXY=',I5)
      GOTO 305
  301 WRITE(ISCR,302)
  302 FORMAT(1X,'*** ERROR READING NUMBXY ***')
  305 CONTINUE
C        GRID X,Y COORDINATES
      DO 311 I=1,NPMSIZ
      XL(I)=I-1
  311 YL(I)=I-1
      DO 320 M=1,MMAX
      DO 320 N=1,NMAX
      BX(N,M)=1.0
      BY(N,M)=1.0
      AREA(N,M)=(XL(M+1)-XL(M))*(YL(N+1)-YL(N))
      IF(IFIELD(N,M)/10.EQ.1)AREA(N,M)=0.5*AREA(N,M)
  320 CONTINUE
C        READ THRU THE CHANNEL WIDTHS
      IF(NUMBXY.LE.0)GOTO 350
      DO 340 J=1,NUMBXY
      READ(IGEO,*)M,N,BX(N,M),BY(N,M),F1
      IF(ITEST.EQ.1)WRITE(ISCR,3111)J,M,N,BX(N,M),BY(N,M),F1
 3111 FORMAT(1X,'J=',I4,' M,N=',2I5,3F10.4)
      AREA(N,M)=AREA(N,M)*F1
  340 CONTINUE
  350 CONTINUE
C
      IF(ITEST.EQ.1)WRITE(ISCR,3391)
 3391 FORMAT(1X,'END OF GEOGRAPHY DATA FILE')
C        CHECK BOUNDARIES
  400 CALL CHECKS
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE CHECKS
C        OCTOBER 1986    HESS    MEAD    VAX 780
C        PURPOSE - TO CHECK FOR CONSISTANCY BETWEEN RIVER AND OCEAN
C                  BOUNDARIES AND THE IFIELD SPECIFICATION
C
      INCLUDE 'COMM20.FOR'
      DIMENSION IFLOWS(6)
      DATA NDBNDS/NM2SIZ/
C        CHECK FOR TIMESTEP SIZE
      PERHR=3600./DTI
      IF(ABS(3600.-IFIX(PERHR)*DTI).GT.0.001)THEN
      WRITE(ISCR,100)DTI,PERHR
  100 FORMAT(1X,'***ERROR: TIMESTEP (SEC) = ',F8.3,', BUT NUMBER PER',
     1 ' HOUR =',F10.4,' IS NOT AN INTEGER')
      STOP
      END IF
C        CHECK FOR NO DEPTH
      NCELL=0
      NBAD=0
      DO 280 M=1,MMAX
      DO 275 N=1,NMAX
      IF(IFIELD(N,M).LT.10)GOTO 275
      NCELL=NCELL+1
      IF(D(N,M).GT.0.0)GOTO 275
      NBAD=NBAD+1
      WRITE(ISCR,272)M,N,D(N,M)
  272 FORMAT(1X,'*** (CHECKS) BAD DEPTH VALUE AT M=',I3,'  N=',I3,
     1 ' D=',F8.2)
  275 CONTINUE
  280 CONTINUE
      IF(NBAD.GT.0)THEN
      WRITE(ISCR,285)
  285 FORMAT(/,1X,'*** BAD DEPTH DATA.  RUN ENDED ***')
      STOP
      END IF
C        CHECK OCEAN FLAGS WITH MESH CELLS
      NO1=0
      NBAD=0
      IF(NUMOBC.LE.0)GOTO 320
      DO 310 I=1,NUMOBC
      DO 310 M=MB1(I),MB2(I)
      DO 310 N=NB1(I),NB2(I)
      NO1=NO1+1
      IF(IFIELD(N,M)/10.NE.KOCNBC)THEN
      WRITE(ISCR,305)KOCNBC,N,M
  305 FORMAT(1X,'*** IFIELD/10 SHOULD BE ',I1,' AT N=',I3,' M=',I3)
      NBAD=NBAD+1
      END IF
  310 CONTINUE
C        CHECK RIVER FLAGS WITH MESH CELLS
  320 IF(NUMRIV.LE.0)GOTO 350
      DO 330 I=1,NUMRIV
      DO 330 M=MR1(I),MR2(I)
      DO 330 N=NR1(I),NR2(I)
      NO1=NO1+1
C        MAKE SURE IFIELD/10 EQUALS KOCNBC+1
      IF(IFIELD(N,M)/10.NE.KOCNBC+1)THEN
      WRITE(ISCR,321)N,M
  321 FORMAT(1X,'***  IFIELD SHOULD BE 10*(KOCNBC+1) AT N=',I3,' M=',I3)
      NBAD=NBAD+1
      END IF
  330 CONTINUE
  350 CONTINUE
C        CHECK FOR CELLS WITH WRONG INDEX
      IF(NBAD.GT.0)THEN
      WRITE(ISCR,353)NBAD
  353 FORMAT(1X,'***ERROR: NUMBER OF CELLS ON BOUND BAD=',I4)
      STOP
      END IF
C        CHECK FOR TOTAL NUMBER OF BOUNDARY CELLS
      IF(NO1.LE.NDBNDS)GOTO 356
      WRITE(ISCR,354)NO1,NDBNDS
  354 FORMAT(1X,'***ERROR: BOUNDARY CELLS=',I4,' GT NDBNDS=',I3)
      STOP
  356 CONTINUE
C        CHECK IFIELD SPECIFICATIONS AGAINST BOUNDARY FLAGS
      NBAD=0
      DO 450 M=1,MMAX
      DO 450 N=1,NMAX
      IF(IFIELD(N,M)/10.EQ.KOCNBC)GOTO 360
      IF(IFIELD(N,M)/10.EQ.KOCNBC+1)GOTO 400
      GOTO 450
C        OCEAN CHECK
  360 IF(NUMOBC.LE.0)GOTO 390
      DO 380 I=1,NUMOBC
      M1=MB1(I)
      M2=MB2(I)
      N1=NB1(I)
      N2=NB2(I)
      DO 380 L=M1,M2
      DO 380 K=N1,N2
  380 IF(L.EQ.M.AND.K.EQ.N) GOTO 450
  390 WRITE(ISCR,395)N,M
  395 FORMAT(/,1X,'*** WARNING. CELL N=',I3,', M=',I3,' IS NOT SET IN',
     1 ' ANY OCEANIC BOUNDARY FLAG')
      NBAD=NBAD+1
      GOTO 450
C        RIVER CHECK
  400 IF(NUMRIV.LE.0)GOTO 430
      DO 420 I=1,NUMRIV
      DO 420 K=NR1(I),NR2(I)
      DO 420 L=MR1(I),MR2(I)
  420 IF(K.EQ.N.AND.L.EQ.M)GOTO 450
  430 WRITE(ISCR,440)N,M
  440 FORMAT(/,1X,'*** WARNING. CELL AT N=',I3,', M=',I3,' IS NOT ',
     1 'SET IN ANY RIVER BOUNDARY FLAG')
      NBAD=NBAD+1
  450 CONTINUE
C        CHECK FOR TOTAL BAD CELLS
      IF(NBAD.GT.0)THEN
      WRITE(ISCR,455)NBAD
  455 FORMAT(1X,'*** ERROR: CELLS IN GRID THAT ARE NOT IN A FLAG=',I4)
      STOP
      END IF
C
C       CHECK TRIANGULAR CELLS FOR LAND/BARRIERS ON TWO ADJACENT SIDES
      NBAD=0
      DO 500 M=1,MMAX
      DO 500 N=1,NMAX
      IF(IFIELD(N,M)/10.NE.1)GOTO 500
C        X FLOW
      IXM=1
      IF(M.EQ.1)THEN
      IXM=0
      ELSE
      II=IFIELD(N,M-1)
      IF(II.LT.10.OR.MOD(II,10).EQ.1.OR.MOD(II,10).EQ.3)IXM=0
      END IF
      IXP=1
      II=IFIELD(N,M)
```

```
      IF(IFIELD(N,M+1).LT.10.OR.MOD(II,10).EQ.1.OR.MOD(II,10).EQ.3)IXP=0
C         Y FLOW
      IYM=1
      IF(N.EQ.1)THEN
      IYM=0
      ELSE
      II=IFIELD(N-1,M)
      IF(II.LT.10.OR.MOD(II,10).EQ.2.OR.MOD(II,10).EQ.3)IYM=0
      END IF
      IYP=1
      II=IFIELD(N,M)
      IF(IFIELD(N+1,M).LT.10.OR.MOD(II,10).EQ.2.OR.MOD(II,10).EQ.3)IYP=0
C        CHECK FOR ADJACENTS
      IFLOWS(1)=IXP
      IFLOWS(2)=IYP
      IFLOWS(3)=IXM
      IFLOWS(4)=IYM
      IFLOWS(5)=IFLOWS(1)
      IFLOWS(6)=IFLOWS(2)
      DO 460 I=1,4
      IF(IFLOWS(I)+IFLOWS(I+1).EQ.0)GOTO 480
  460 CONTINUE
      WRITE(ISCR,470)N,M,IXP,IYP,IXM,IYM
  470 FORMAT(/,1X,'*** WARNING. TRIANGULAR CELL AT N=',I3,', M=',I3,
     1 ' HAS INCORRECT SIDES',/,1X,'IXP,IYP,IXM,IYM=',4I4)
      NBAD=NBAD+1
  480 CONTINUE
C        CHECK FOR TOTAL TRIANGULAR CELLS NOT WELL DEFINED
      IF(NBAD.GT.0)THEN
      WRITE(ISCR,495)NBAD
  495 FORMAT(1X,'***ERROR: TRIANGULAR CELLS NOT WELL DEFINED=',I4)
      STOP
       END IF
  500 CONTINUE
c
c        CHECK FOR CORRESPONDANCE BETWEEN (1) BOUNDARIES AND (2) INPUT SIG
      WRITE(ISCR,*)' '
      WRITE(ISCR,*)'CHECK BOUNDARIES AND DATA'
      NBAD=0
c        ocean boundaries
      IF(NUMOBC.GT.0)THEN
      IF(NSIGTD.EQ.0)THEN
        WRITE(ISCR,*)'**NO WATER LEVEL SIGNALS AT OCEAN BOUNDARY**'
        NBAD=1
        ENDIF
      IF((KONCEN.EQ.1.OR.KONCEN.EQ.3).AND.NSIGS.EQ.0)THEN
        WRITE(ISCR,*)'**NO SALINITY SIGNALS AT OCEAN BOUNDARY**'
        NBAD=1
        ENDIF
      IF((KONCEN.EQ.2.OR.KONCEN.EQ.3).AND.NSIGTO.EQ.0)THEN
        WRITE(ISCR,*)'**NO TEMPERATURE SIGNALS AT OCEAN BOUNDARY**'
        NBAD=1
        ENDIF
      ENDIF
c        river boundaries
      IF(NUMRIV.GT.0)THEN
      IF(NSIGR.EQ.0)THEN
        WRITE(ISCR,*)'**NO FLOW RATE DATA AT RIVER BOUNDARY**'
        NBAD=1
        ENDIF
      IF((KONCEN.EQ.2.OR.KONCEN.EQ.3).AND.NSIGRT.EQ.0)THEN
        WRITE(ISCR,*)'**NO TEMPERATURE SIGNALS AT RIVER BOUNDARY**'
        NBAD=1
        ENDIF
      ENDIF
c        air-sea boundaries
      IF((KONCEN.EQ.2.OR.KONCEN.EQ.3).AND.NSIGW.EQ.0)THEN
        WRITE(ISCR,*)'**NO WIND DATA AT AIR-SEA BOUNDARY**'
        NBAD=1
        ENDIF
      IF((KONCEN.EQ.2.OR.KONCEN.EQ.3).AND.NSIGM.EQ.0)THEN
        WRITE(ISCR,*)'**NO ADDITIONAL MET DATA AT AIR-SEA BOUNDARY**'
        NBAD=1
        ENDIF
      IF(NBAD.EQ.1)STOP

      RETURN
      END
C
C------------------------------------------------------------------
C
      SUBROUTINE RDICS
C        FEBRUARY 1997     K. W. HESS     IRIS
C        PURPOSE - TO READ IN THE INITIAL CONDITIONS
      INCLUDE 'COMM20.FOR'
C        READ THE INDICES
      LUICS=LUCON
      WRITE(ISCR,100)LUICS
  100 FORMAT(X,'NOW READING THE ICS FILE ON UNIT=',I2)
      CALL FUOPEN(LUICS,FINIT)
      READ(LUICS)NIX,MIX,LBOT1,NSTET,UT01,YEAR01,K
      WRITE(ISCR,110)UT1,YEAR1,NSTET,K
  110 FORMAT(5X,'UT1=',F10.4,'  YEAR1=',F6.1,'  NSTET=',I6,'  K=',I2)
      READ(LUICS)SE,UE,VE,SOLD,UHOLD,VHOLD,AH,AV,PHI,TBX,TBY,
     1 U,V,W,THETA1,THETA2,THETA3
      IF(K.GE.10)READ(LUICS)AH3,WX,WY,TSX,TSY
      IF(MOD(K,10).GT.0)READ(LUICS)S,T,DV,RI,NSTINF
C        RESET SECONDARY VARIABLES
      YEAR1=YEAR01
      UT1=UT01
      DO 120 M=1,MMAX
      MP=MIN0(M+1,MMAX)
      DO 120 N=1,NMAX
      IF(IFIELD(N,M).LT.10)GOTO 120
      NP=MIN0(N+1,NMAX)
      UH(N,M)=UE(N,M)*.5*(D(N,M)+SE(N,M)+D(N,MP)+SE(N,MP)+E)*BX(N,M)
      VH(N,M)=VE(N,M)*.5*(D(N,M)+SE(N,M)+D(NP,M)+SE(NP,M)+E)*BY(N,M)
      AHC(N,M)=.25*(AH(N,M)+AH(N,MP)+AH(NP,M)+AH(NP,MP))
  120 CONTINUE
c        if no AH3
      IF(INTER.GT.0.AND.KONC.LT.10)THEN
      DO 140 M=1,MMAX
      DO 140 N=1,NMAX
      DO 140 L=1,LBOT
  140 AH3(L,N,M)=AH(N,M)
      ENDIF
      WRITE(ISCR,*)'COMPLETE RDICS'
      CLOSE (LUICS)
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE ZEROS
C        APRIL 1988   K. W. HESS
C        PURPOSE - INITIALIZE THE PARAMETERS TO ZERO BEFORE
C                       INPUT FILES HAVE BEEN READ IN.
C        VARIABLES -
      INCLUDE 'COMM20.FOR'
C        INITIALIZE VISCOSITIES AND DEPTHS
      LBOT=LSIZE
      DO 130 M=1,MSIZE
      DO 130 N=1,NSIZE
      AREA(N,M)=0.0
      TSX(N,M)=0.0
      TSY(N,M)=0.0
      WX(N,M)=0.0
      WY(N,M)=0.0
      TBX(N,M)=0.0
      TBY(N,M)=0.0
      UH(N,M)=0.0
      VH(N,M)=0.0
      SE(N,M)=0.0
      SEP(N,M)=0.0
      SEPP(N,M)=0.0
      UHP(N,M)=0.0
      VHP(N,M)=0.0
      SOLD(N,M)=0.0
      UHOLD(N,M)=0.0
      VHOLD(N,M)=0.0
      THETA1(N,M)=0.0
      THETA2(N,M)=0.0
      THETA3(N,M)=0.0
      UE(N,M)=0.0
      VE(N,M)=0.0
      AH(N,M)=00
      AHC(N,M)=00
      FEDGE(N,M)=1.0
      PHI(N,M)=0.0
      MFLUX(N,M)=0.0
      NFLUX(N,M)=0.0
C       LOOP OVER LEVELS
      DO 120 L=1,LSIZE
      U(L,N,M)=0.0
      V(L,N,M)=0.0
      W(L,N,M)=0.0
      AH3(L,N,m)=0.
      S(L,N,M)=0.0
      T(L,N,M)=0.0
      RI(L,N,M)=0.0
      DV(L,N,M)=00
  120 AV(L,N,M)=00
  130 CONTINUE

      DO 135 L=1,LSIZE
      DO 135 I=1,NM2SIZ
  135 NSTINF(L,I)=0
C        SET HEAT FLUXES
      QI=0.0
      QA=0.0
      QB=0.0
      QS=0.0
      QE=0.0
      RETURN
      END
c---------------------------------------------------------------------
      SUBROUTINE YTIME8(YT8,UT,YEAR)
c        create a date from year and Julian day
      REAL*8 YT8,UT8,DAYS
c        look for leap year
      IYEAR=YEAR
      DAYS=365.
      IF(MOD(IYEAR,4).EQ.0.AND.(.NOT.(MOD(IYEAR,100).EQ.0.AND.
     1 MOD(IYEAR,400).NE.0)))DAYS=366.
      UT8=UT
      YT8=FLOAT(IYEAR-1900)+(UT8-1.)/DAYS
      RETURN
      END
C
C---------------------------------------------------------------------
C
      SUBROUTINE RR(YT,ISCR,LUT,IEND,DD,YD,NSIG,NN,VALS,FINAL)
c        generic read for input data records
      DIMENSION DD(2),YD(2),VALS(2,NN),FINAL(NN)
      REAL*8 YT,YT1,YT2
      DO 90 N=1,NN
   90 FINAL(N)=0.0
      IF(NSIG.EQ.0.OR.IEND.EQ.1)RETURN
      CALL YTIME8(YT1,DD(1),YD(1))
      CALL YTIME8(YT2,DD(2),YD(2))
      IF(YT.LT.YT1)THEN
      WRITE(ISCR,95)LUT,YT,YT1
   95 FORMAT(1X,'** EARLIER THAN FIRST DATA TIME ON UNIT=',I2,/,
     1 'YT=',F12.8,'  YT1=',F12.8,' **')
      RETURN
      ENDIF
  100 IF(YT.GT.YT2)THEN
      DO 120 N=1,NSIG
  120 VALS(1,N)=VALS(2,N)
      DD(1)=DD(2)
      YD(1)=YD(2)
      YT1=YT2
      READ(LUT,*,END=130)YD(2),DD(2),(VALS(2,N),N=1,NSIG)
      CALL YTIME8(YT2,DD(2),YD(2))
      GOTO 100
      ENDIF
      GOTO 150
  130 IEND=1
      WRITE(ISCR,140)LUT
  140 FORMAT(1X,'** NO MORE DATA ON UNIT=',I2,' **')
      RETURN
c        interpolate
  150 F2=(YT-YT1)/(YT2-YT1)
      DO 160 N=1,NSIG
  160 FINAL(N)=(1.-F2)*VALS(1,N)+F2*VALS(2,N)
      RETURN
      END
C
c-------------------------------------------------------------------
C
      SUBROUTINE IRR(IUNIT,ISCR,LUT,IEND,DD,YD,NSIG,NN,VALS)
      DIMENSION DD(2),YD(2),VALS(2,NN)
      CHARACTER*40 FDATA
C        INITIAL READ OF DATA
      IEND=0
      READ(IUNIT,100)
  100 FORMAT(1X)
      READ(IUNIT,*)NSIG
      IF(NSIG.GT.0)GOTO 110
      WRITE(ISCR,*)'     NO INPUT DATA '
      GOTO 140
c        read file
```

```
 110  READ(IUNIT,120)FDATA
 120  FORMAT(A40)
      WRITE(ISCR,130)FDATA,NSIG,LUT
 130  FORMAT(5X,'FILE NAME=',A40,/,5x,'NSIG=',I2,'  LUT=',I2)
      CALL FFOPEN(LUT,FDATA)
      READ(LUT,*,END=140)YD(1),DD(1),(VALS(1,N),N=1,NSIG)
      READ(LUT,*,END=140)YD(2),DD(2),(VALS(2,N),N=1,NSIG)
      IEND=0
      GOTO 150
 140  IEND=1
 150  RETURN
      END
c
c-------------------------------------------------------
c
      SUBROUTINE RDWIND(I,IEND)
c        PURPOSE - TO READ THE METEOROLOGICAL FILES
C        VARIABLES -
C            ITYPE1 = INDEX FOR WIND OR STRESS.
C                     1=WIND, 2=STRESS
C            ITYPE2 = INDEX FOR HORIZONTAL ATMOSPHERIC PRESSURE GRADIENT
C                     0=NONE, 1=VALUES TO BE READ
C            ITYPE3 = INDEX FOR COORDINATES FOR WIND/STRESS AND PRESSURE
C                     0=IN MODEL COORDS
C                     1=IN EAST, NORTH
      INCLUDE 'COMM20.FOR'
      COMMON/METOX/TMET8(2),ITYPE1
      DIMENSION ATX(NSIZE,MSIZE),ATY(NSIZE,MSIZE)
      REAL*8 TMET8,UTM
c        set end-of-file index
      IEND=0
c        read data
      READ(LUWND,END=110)DATE1,DATE2,ITYPE1,ITYPE2,ITYPE3,NIX,MIX
      IF(DATE1.GT.370.)THEN
      YMET(I)=DATE1
      DMET(I)=DATE2
      ELSE
      YMET(I)=DATE2
      DMET(I)=DATE1
      ENDIF
c        read arrays
      DPADX=0.
      DPADY=0.
      IF(NIX.GT.1.AND.MIX.GT.1)THEN
      IF(ITYPE2.EQ.0)READ(LUWND)((ATX(N,M),N=1,NIX),M=1,MIX),
     1  ((ATY(N,M),N=1,NIX),M=1,MIX)
      IF(ITYPE2.EQ.1)READ(LUWND)((ATX(N,M),N=1,NIX),M=1,MIX),
     1  ((ATY(N,M),N=1,NIX),M=1,MIX),DPADX,DPADY
      ELSE
c        read single wind speed/stress
      IF(ITYPE2.EQ.0)READ(LUWND)AT1,AT2
      IF(ITYPE2.EQ.1)READ(LUWND)AT1,AT2,DPADX,DPADY
      DO M=1,MSIZE
      DO N=1,NSIZE
      ATX(N,M)=AT1
      ATY(N,M)=AT2
      ENDDO
      ENDDO
      ENDIF
C        rotate winds to model basin angle
      IF(ITYPE3.EQ.1)THEN
      ARG=RAD*BSNANG
      DO M=1,MSIZE
      DO N=1,NSIZE
      ATX1=-COS(ARG)*ATY(N,M)-SIN(ARG)*ATX(N,M)
      ATY1= COS(ARG)*ATX(N,M)-SIN(ARG)*ATY(N,M)
      ATX(N,M)=ATX1
      ATY(N,M)=ATY1
      ENDDO
      ENDDO
      DPADX1=-COS(ARG)*DPADY-SIN(ARG)*DPADX
      DPADY1= COS(ARG)*DPADX-SIN(ARG)*DPADY
      DPADX=DPADX1
      DPADY=DPADY1
      ENDIF
c        create date
      CALL YTIME8(UTM,DMET(I),YMET(I))
      TMET8(I)=UTM
      WRITE(ISCR,100)I,NSTI,ITYPE1,ITYPE2,ITYPE3,DMET(I),YMET(I),
     1 TMET8(I),NIX,MIX
 100  FORMAT(5X,'RDWIND: I,NSTI=',2I6,' ITYPE1,2,3=',3I6,/,
     1 5x,'DY,YR=',2F12.5,' TMET8(I)=',F12.7,/,5x,'NIX,MIX=',2I5)
      WRITE(ISCR,105)ATX(1,1),ATY(1,1),DPADX,DPADY
 105  FORMAT(5x,'ATX,ATY(1,1)=',2E12.4,'  DPADX,DPADY=',2E12.4)
c        save data
      DO N=1,NSIZE
      DO M=1,MSIZE
      FX(I,N,M)=ATX(N,M)
      FY(I,N,M)=ATY(N,M)
      ENDDO
      ENDDO
      IENDWN=IEND
      RETURN
c        end of data
 110  IEND=1
      WRITE(6,*)'END OF MET DATA REACHED'
      IENDWN=IEND
      RETURN
      END
C======================================================================
C     MECCA: ANALYS
C
C----------------------------------------------------------------------
C
      SUBROUTINE ANALYS

C        JULY 1988    K. HESS    MEAD    VAX780
C        PURPOSE - TO CALL THE ANALYSIS ROUTINES
      INCLUDE 'COMM20.FOR'
      CALL CHECK2
      RETURN
      END
C
C----------------------------------------------------------------------
C
      SUBROUTINE CHECK2

C        JULY 1988    K. HESS    MEAD    VAX780
C        PURPOSE - TO CALL THE ANALYSIS ROUTINES
      INCLUDE 'COMM20.FOR'
C          CHECK FOR LARGE WATER LEVEL VALUES
      ISTOP=0
      SEMAX=3.
      DO 110 N=1,NMAX
      DO 110 M=1,MMAX
      IF(ABS(SE(N,M)).GT.SEMAX)THEN
      WRITE(6,105)NSTI,N,M,SE(N,M),D(N,M)
 105  FORMAT(1X,'ANALYSIS: NSTI=',I6,' N,M=',2I4,' SE=',F6.2,' D=',F6.2)
      ISTOP=1
      ENDIF
 110  CONTINUE

C         CHECK FOR LARGE SALINITY VALUES
      IF(KONCEN.EQ.0)RETURN
      IF(IPRNT1.EQ.0)RETURN
      SALMIN= 1.E+10
      SALMAX=-1.E+10
      DO M=1,MMAX
      DO N=1,NMAX
      IF(IFIELD(N,M).GT.0)THEN
      DO L=1,LBOT
c        check for max
      IF(S(L,N,M).GT.SALMAX)THEN
       MSMAX=M
       NSMAX=N
       LSMAX=L
       SALMAX=S(L,N,M)
       ENDIF
      IF(S(L,N,M).LT.SALMIN)THEN
       MSMIN=M
       NSMIN=N
       LSMIN=L
       SALMIN=S(L,N,M)
       ENDIF
      ENDDO
      ENDIF
      ENDDO
      ENDDO
      WRITE(6,150)UT,NSTI,SALMAX,MSMAX,NSMAX,LSMAX
 150  FORMAT(1X,'AT UT=',F10.4,' NSTI=',I6,T30,' SALMAX=',F10.2,
     1 ' AT M,N,L=',3I4)
      WRITE(6,160)SALMIN,MSMIN,NSMIN,LSMIN
 160  FORMAT(T30,' SALMIN=',F10.2,' AT M,N,L=',3I4)

      RETURN
      END
C======================================================================
C     MECCA FILE : CONC.FOR  -  CONCENTRATION SUBROUTINES
C----------------------------------------------------------------------
C
      SUBROUTINE CONCZ
C        MARCH 1986   K. W. HESS     (LAST REVISED 23 JULY 87)
C        PURPOSE - SUPER-STREAMLINED, SQUARE-GRID VERSION.
C                  TO COMPUTE NEW DISTRIBUTION OF CONCENTRATE,
C                  WHICH INCLUDES VARIABLE WIDTH AND VARIABLE HORIZONTAL
C                  VISCOSITY. NEW FORMULATION OF UPPER BOUNDARY CONDITIO
C        INITIALIZE BOUNDARY CONCENTRATIONS
      INCLUDE 'COMM20.FOR'
      COMMON/BNDY4/ISALT,ITEMP,NB
c        CHECK FOR TIME
      IF(HR1.LT.HRCON1)RETURN
C        SET INDICES:  1=DO IT, 2=SKIP
      ISALT=2
      IF(KONCEN.EQ.1.OR.KONCEN.EQ.3)ISALT=1
      ITEMP=2
      IF(KONCEN.EQ.2.OR.KONCEN.EQ.3)ITEMP=1

C        GET OCEAN BOUNDARY CONDITIONS
      CALL BNDRY3
C        GET RIVER BOUNDARY CONDITIONS
      CALL BNDRY4
C        SOLVE GENERALIZED TRANSPORT EQUATION
      CALL GFLUX
      END
C
C----------------------------------------------------------------------
C
      SUBROUTINE BNDRY3
C        OCTOBER 1984   K. W. HESS   MEAD   VAX11/750
C        PURPOSE - TO SET THE OCEANIC SALINITY AND TEMPERATURE
C                  BOUNDARY CONDITIONS
C        VARIABLES -
C                IDIR = INFLOW DIRECTION PARAMETER:1=X DIR, 2=Y DIR
C         ISALT,ITEMP = SET INDICES:  1=DO IT, 2=SKIP
C              ISENSE = INFLOW PARAMETER:1=INFLOW IS IN POSITIVE DIR.
C                       2=INFLOW IS IN NEGATIVE DIRECTION
C                ITPO = TYPE OF OCEAN BOUNDARY CONDITION
C                       +/-1 : OUTFLOW IN +/-X DIRECTION
C                       +/-2 : OUTFLOW IN +/-Y DIRECTION
C                JTPO = TYPE OF OCEAN BOUNDARY CONDITION
C                        1 : WATER LEVEL SPECIFICATION
C                        2 : TRANSPORT OUTFLOW
C                        3 : ORLANSKI RADIATION OUTFLOW (REQUIRES WL)
C                        4 : REIMANN INVARIENT RADIATION OUTFLOW (REQUIRES
C                ITPR = DIRECTION OF RIVER BOUNDARY INFLOW TO ESTUARY
C                       +/-1 : INFLOW IN +/-X DIRECTION
C                       +/-2 : INFLOW IN +/-Y DIRECTION
C                JTPR = TYPE OF RIVER BOUNDARY CONDITION
C                       1 : CHANNEL
C                       2 : FALLS
C                QRIV = RIVER FLOWRATE (M**3/S)
      INCLUDE 'COMM20.FOR'
      COMMON/BNDY4/ISALT,ITEMP,NB
      DIMENSION SN(LSIZE),TN(LSIZE)
      CHARACTER*6 ANAMES(2)
      DATA ANAMES/'OUTFLO','INFLO '/,HRINF/6./,IP/0/
C        MAKE SURE THERE ARE BOUNDARIES TO SET
      IF(NUMOBC.LE.0)RETURN
C        SET OCEANIC CONDITIONS
      CALL BSTATE
      NHRINF=HRINF*IHR
      CIN1=DTI/DL
      IP1=IPRNT1
      IF(IP.EQ.0)IP1=0
C        LOOP THRU OCEANIC BOUNDARIES
      NB=0
      DO 360 IB=1,NUMOBC
      IF(IP1.EQ.1)WRITE(ISCR,*)'OCEAN SALINITY BND: IB=',IB,
     1 '  NSTI=',NSTI
      MF=MB1(IB)
      ML=MB2(IB)
      NF=NB1(IB)
      NL=NB2(IB)
C        SET DIRECTIONAL PARAMETER, IDIR: 1=X, 2=Y
      IDIR=IABS(ITPO(IB))
      ISENSE=(3-ISIGN(1,ITPO(IB)))/2
C        OUTFLOW SENSE:FOR +DIRECTION, ISENSE=1; FOR -DIR, ISENSE=2
      ND1=(IDIR/2)*(2*ISENSE-3)
      MD1=(1-IDIR/2)*(2*ISENSE-3)
C          LOOP THRU EACH GRID
      DO 295 N=NF,NL
      DO 295 M=MF,ML
```

```
      N1=N+ND1
      M1=M+MD1
      NB=NB+1
C        LOOP OVER DEPTHS
      DO 220 L=1,LBOT
      GOTO(100,110)IDIR
  100 UL=UE(N,M-2+ISENSE)+U(L,N,M-2+ISENSE)
      GOTO 120
  110 UL=VE(N-2+ISENSE,M)+V(L,N-2+ISENSE,M)
  120 CONTINUE
      IF(-UL*ISIGN(1,ISENSE-2).LE.0.0)GOTO 150
C        OUTFLOW CONDITIONS
  130 JFLOW=1
      NSTINF(L,NB)=NSTIT
C        FIRST-ORDER ADVECTION
      F0=ABS(CIN1*UL)
      F1=F0*(2-ISALT)
      F2=F0*(2-ITEMP)
      SN(L)=(S(L,N,M)*(1.-F1)+F1*S(L,N1,M1))
      TN(L)=(T(L,N,M)*(1.-F2)+F2*T(L,N1,M1))
      GOTO 200
C        INFLOW CONDITIONS
  150 JFLOW=2
      F0=FLOAT(ISPLIT)/FLOAT(MAX0(ISPLIT,NSTINF(L,NB)+NHRINF-NSTIT))
      F1=F0*(2-ISALT)
      F2=F0*(2-ITEMP)
      SN(L)=(S(L,N,M)*(1.-F1)+F1*SBND(L,NB))
      TN(L)=(T(L,N,M)*(1.-F2)+F2*TBND(L,NB))
  200 CONTINUE
  220 CONTINUE
C         CHECK FOR POSITIVE VALUES
      IF(ICPOS.EQ.0)GOTO 240
      DO L=1,LBOT
      SN(L)=AMAX1(SN(L),0.0)
      TN(L)=AMAX1(TN(L),0.0)
      ENDDO
  240 CONTINUE
      DO L=1,LBOT
      T(L,N,M)=TN(L)
      S(L,N,M)=SN(L)
      ENDDO
  295 CONTINUE
  360 CONTINUE
  370 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE BNDRY4
C        NOVEMBER 1986   HESS & PYTLOWANY   MEAD   VAX
C        PURPOSE - TO COMPUTE THE RIVERINE SALINITY AND TEMPERATURE
C                  BOUNDARY CONDITIONS
      INCLUDE 'COMM20.FOR'
      COMMON/BNDY4/ISALT,ITEMP,NB
      DIMENSION FSZ(LSIZE)
      DO L=1,LBOT
      FSZ(L)=2*(1.-FLOAT(L-1)/FLOAT(LBOT-1))
      ENDDO
      IP=0
      IP=IPRNT1
C        RIVER FLOW BOUNDARIES
      IF(NUMRIV.LE.0)GOTO 300
      NB=0
      DO 250 NR=1,NUMRIV
      MF=MR1(NR)
      ML=MR2(NR)
      NF=NR1(NR)
      NL=NR2(NR)
      DO 220 M=MF,ML
      DO 220 N=NF,NL
      NB=NB+1
      IF(JTPR(NR).EQ.2)GOTO 130
C        FLUME CONDITION
      DO 100 L=1,LBOT
      IF(ISALT.EQ.1)S(L,N,M)=SBND(L,NB+NBCELO)
  100 IF(ITEMP.EQ.1)T(L,N,M)=TBND(L,NB+NBCELO)
      GOTO 220
C        WATER FALLS CONDITION
  130 SAL=SBND(1,NB+NBCELO)
      TMP=TBND(1,NB+NBCELO)
      KMAX=1+ML-MF+NL-NF
      ND=N
      MD=M
      VOL0=(D(ND,MD)+SE(ND,MD))*DQ*AREA(ND,MD)*DL**2
      VOL1=DTI*RATE(NR)*DQ/FLOAT(KMAX)
      F3=VOL1/(VOL0+VOL1)
      IF(IP.EQ.1)WRITE(ISCR,140)NR,ND,MD
  140 FORMAT(3X,'RIVER: NR=',I2,' N,M=',2I4)
      IF(IP.EQ.1)WRITE(ISCR,*)'SAL,TMP=',SAL,TMP,'  F3=',F3
      DO 170 L=1,LBOT
      F2=F3*FSZ(L)
      F1=1.-F2
      IF(ISALT.EQ.1)S(L,ND,MD)=(S(L,ND,MD)*F1+SAL*F2)
      IF(ITEMP.EQ.1)T(L,ND,MD)=(T(L,ND,MD)*F1+TMP*F2)
      IF(IP.EQ.1)WRITE(ISCR,160)L,S(L,ND,MD),T(L,ND,MD)
  160 FORMAT(3X,'L=',I3,'  S=',F5.2,'  T=',F5.2)
  170 CONTINUE
  220 CONTINUE
  250 CONTINUE
  300 RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE BSTATE
C        FEBRUARY 1996   K. W. HESS   CEOB   SGI/IRIS
C        PURPOSE - TO SET THE RIVERINE AND OCEANIC SALINITY AND
C                  TEMPERATURE BOUNDARY STATES BY INTERPOLATION
C        VARIABLES -
C           SBND(L,NB) = INTERPOLATED STATE OF BOUNDARY SALINITY AT
C                        LEVEL L AND BOUNDARY GRID NB
C           TBND(L,NB) = INTERPOLATED STATE OF BOUNDARY TEMPERATURE
C                  NBC = TOTAL NUMBER OF OCEANIC AND RIVERINE
C                        BOUNDARY GRIDS (UP TO 100 ALLOWED)
      INCLUDE 'COMM20.FOR'
      COMMON/BNDY4/ISALT,ITEMP,NB
      DIMENSION SFINL(LSIZE),TFINL(LSIZE),TRFINL(NDRIV2)
C        SET STANDARD CONDITIONS
      IF(IPRNT1.EQ.1)WRITE(ISCR,100)UT,HR1,NSTI
  100 FORMAT(/,1X,'BSTATE : UT=',F10.4,' CUM HR=',F10.2,' NSTI=',I6)
C        set to default values
      DO L=1,LBOT
      SFINL(L)=SAL0
      TFINL(L)=TMP0
      DO N=1,NM2SIZ
      SBND(L,N)=SAL0
      TBND(L,N)=TMP0
      ENDDO
      ENDDO
      DO N=1,NDRIV2
      TRFINL(N)=TMP0
      ENDDO
c        initialize cell counts
      NBCEL=0
      NBCELO=0
c        check on ocean conditions
      IF(NUMOBC.LE.0)GOTO 300
C        SET OCEANIC SALINITY CONDITIONS
      IF(KONCEN.EQ.2.OR.NSIGS.EQ.0.OR.IENDSO.EQ.1)GOTO 150
c        read more salinity data
      CALL RR(YT,ISCR,LUSAL,IENDSO,DSAL,YSAL,NSIGS,LSIZE,SALOCN,SFINL)
c         OCEAN TEMP
  150 IF(KONCEN.EQ.1.OR.NSIGTO.EQ.0.OR.IENDTO.EQ.1)GOTO 165
c        read more ocean temperature data
      CALL RR(YT,ISCR,LUOCT,IENDTO,DOTP,YOTP,NSIGTO,LSIZE,TMPOCN,TFINL)
  165 IF(IPRNT1.EQ.0)GOTO 200
      DO L=1,LBOT
      WRITE(ISCR,170)L,SFINL(L),TFINL(L)
  170 FORMAT(3x,'L=',I2,'  SFINL=',F7.3,'  TFINL=',F7.3)
      ENDDO
C        LOOP THRU OCEANIC BOUNDARIES
  200 DO 250 IB=1,NUMOBC
C         LOOP THRU EACH GRID
      DO 250 N=NB1(IB),NB2(IB)
      DO 250 M=MB1(IB),MB2(IB)
      NBCEL=NBCEL+1
      NBCELO=NBCELO+1
C        SET VALUES OVER DEPTH
      DO 240 L=1,LBOT
      SBND(L,NBCEL)=SFINL(L)
  240 TBND(L,NBCEL)=TFINL(L)
  250 CONTINUE
C
C        RIVER BOUNDARIES
  300 IF(NUMRIV.LE.0)GOTO 400
c        read more river temperature data
      IF(KONCEN.EQ.1.OR.NSIGRT.EQ.0.OR.IENDRT.EQ.1)GOTO 310
      CALL RR(YT,ISCR,LURVT,IENDRT,DRVT,YRVT,NSIGRT,NDRIV2,TRIV,TRFINL)
C        LOOP THRU THE RIVERS
  310 DO 340 NR=1,NUMRIV
C         LOOP THRU THE GRIDS
      DO 340 M=MR1(NR),MR2(NR)
      DO 340 N=NR1(NR),NR2(NR)
      NBCEL=NBCEL+1
C        LOOP OVER DEPTH
      DO 320 L=1,LBOT
      SBND(L,NBCEL)=0.0
  320 TBND(L,NBCEL)=TRFINL(NR)
      IF(IPRNT1.EQ.1)WRITE(ISCR,330)NR,TRFINL(NR)
  330 FORMAT(3x,'NR=',I2,'  TRFINL=',F7.3)
  340 CONTINUE
  400 CONTINUE

      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE GFLUX
C        JUNE 1996   K. W. HESS
C        PURPOSE - TO COMPUTE NEW DISTRIBUTION OF CONCENTRATE,
C                  WHICH INCLUDES VARIABLE WIDTH AND VARIABLE HORIZONTAL
C                  VISCOSITY. NEW FORMULATION OF UPPER BOUNDARY
C                  CONDITION.
C        VARIABLES -
C                SN() = NEW (UPDATED) SALINITY
C               SC(,) = SALINITY AT ROW AT START OF UPDATE
C               SM(,) = SALINITY AT PREVIOUS ROW AT START OF UPDATE
C           FA(),FB() = RECURSION ARRAYS FOR SALINITY, TEMPERATURE
C                FTOP = FLUX OF C AT THE AIR-WATER INTERFACE

c    L-1  + U(L-1), W, S
c         |
c ------- | DV(L-1)    ----------      ULM, (DM,GM), FZZA(L-1)
c         |
c     L   + U(L), W, S                 FFCC(L)
c         |
c ------- | DV(L)      ----------      ULP, DP,GP,  FFZA(L),FFZD(L)
c         |
c    L+1  + U(L+1), W, S

      INCLUDE 'COMM20.FOR'
      COMMON/BNDY4/ISALT,ITEMP,NB
      DIMENSION SN(LSIZE),SC(LSIZE,NSIZE),SM(LSIZE,NSIZE),
     2 TN(LSIZE),TC(LSIZE,NSIZE),TM(LSIZE,NSIZE),TRAD(LSIZE),
     2 FFCC(LSIZE),FFXM(LSIZE),FFXP(LSIZE),FFYM(LSIZE),FFYP(LSIZE),
     3 FFZA(LSIZE),FFZD(LSIZE),DEN(LSIZE),PROD(LSIZE)
      DATA IP/0/

      C1=DTI/(4.*DQ)
      C2=DTI/DQ**2
      C3=DHAH*DTI/(2.*DL**2)
      C4=DTI/(2.*DL)
      C5=DTI/(16.*DL)

C        INITIALIZE SURFACE HEAT FLUX
      IF(KONCEN.GE.2)CALL HEAT1
C
C         SOLVE GENERALIZED TRANSPORT EQUATION
C
C        LOOP DOWN THE LINES
      DO 400 M=1,MMAX
C        STORE VALUES
      DO 130 N=1,NMAX
      DO 130 L=1,LBOT
      IF(M.GT.1)GOTO 110
      SM(L,N)=S(L,N,M)
      TM(L,N)=T(L,N,M)
      GOTO 120
  110 SM(L,N)=SC(L,N)
      TM(L,N)=TC(L,N)
  120 SC(L,N)=S(L,N,M)
      TC(L,N)=T(L,N,M)
  130 CONTINUE

C        RUN ACROSS ROW
      NA=NAB(M)/1000
      NB=MOD(NAB(M),1000)
      IF(NA.GT.NB)GOTO 400
      MM=MAX0(M-1,1)
      MP=MIN0(MMAX,M+1)
```

```
C        LOOP ACROSS ROWS
      DO 390 N=NA,NB
      IF(IFIELD(N,M).LT.10.OR.IFIELD(N,M)/10.EQ.KOCNBC)GOTO 390
      NM=MAX0(N-1,1)
      NP=MIN0(NMAX,N+1)
C        COMPUTE REPEATEDLY-USED TERMS
      H=D(N,M)+SE(N,M)+E
      ANM=AREA(N,M)
      F1=C1*ANM
      F2=C2*ANM/H
      HH=ANM*H
      SS=ANM*(SE(N,M)-SOLD(N,M))
C        HORIZONTAL X-DIRECTION FLUX TERMS
      UHO1=BX(N,M-1)*C4*UHOLD(N,M-1)
      H1=BX(N,M-1)*C5*(H+D(N,M-1)+SE(N,M-1))
      DH1=BX(N,M-1)*C3*FLOAT(MFLUX(N,MM))
      UHO2=BX(N,M)*C4*UHOLD(N,M)
      H2=BX(N,M)*C5*(H+D(N,M+1)+SE(N,M+1))
      DH2=BX(N,M)*C3*FLOAT(MFLUX(N,M))
C        HORIZONTAL Y-DIRECTION FLUX TERMS
      VHO1=BY(N-1,M)*C4*VHOLD(N-1,M)
      H3=BY(N-1,M)*C5*(H+D(N-1,M)+SE(N-1,M))
      DH3=BY(N-1,M)*C3*FLOAT(NFLUX(NM,M))
      VHO2=BY(N,M)*C4*VHOLD(N,M)
      H4=BY(N,M)*C5*(H+D(N+1,M)+SE(N+1,M))
      DH4=BY(N,M)*C3*FLOAT(NFLUX(N,M))
C
C        INITIAL VELOCITIES
      L=LBOT
      ULP1=U(L,N,M-1)+U(L-1,N,M-1)
      ULP2=U(L,N,M)+U(L-1,N,M)
      VLP1=V(L,N-1,M)+V(L-1,N-1,M)
      VLP2=V(L,N,M)+V(L-1,N,M)
C        LOOP THRU THE LAYERS, STARTING AT BOTTOM
      DO 150 L=LBOT,1,-1
      LM=MAX0(1,L-1)
C        VERTICAL ADVECTIVE AND DIFFUSIVE FLUX TERMS: units are m/s
      FFZA(L)=F1*(W(L,N,M)+W(L+1,N,M))      ! Gp
      FFZD(L)=F2*DV(L,N,M)                  ! Dp
C        GET NEW VELOCITIES
      LEM=L-1
      IF(L.EQ.1)LEM=L+1
      ULM1=U(L,N,M-1)+U(LEM,N,M-1)
      ULM2=U(L,N,M)+U(LEM,N,M)
      VLM1=V(L,N-1,M)+V(LEM,N-1,M)
      VLM2=V(L,N,M)+V(LEM,N,M)
C        GET HORIZONTAL GENERALIZED FLUX COEFFICIENTS : units are meters
      FFXM(L)=  (H1*(ULM1+ULP1)+UHO1)+DH1*(AH3(L,N,M)+AH3(L,N,M-1))
      FFCC(L)=  (H1*(ULM1+ULP1)+UHO1)-DH1*(AH3(L,N,M)+AH3(L,N,M-1))
     1         -(H2*(ULM2+ULP2)+UHO2)-DH2*(AH3(L,N,M)+AH3(L,N,M+1))
     2         +(H3*(VLM1+VLP1)+VHO1)-DH3*(AH3(L,N,M)+AH3(L,N-1,M))
     3         -(H4*(VLM2+VLP2)+VHO2)-DH4*(AH3(L,N,M)+AH3(L,N+1,M))
     4         +HH
      FFXP(L)=-(H2*(ULM2+ULP2)+UHO2)+DH2*(AH3(L,N,M)+AH3(L,N,M+1))
      FFYM(L)=  (H3*(VLM1+VLP1)+VHO1)+DH3*(AH3(L,N,M)+AH3(L,N-1,M))
      FFYP(L)=-(H4*(VLM2+VLP2)+VHO2)+DH4*(AH3(L,N,M)+AH3(L,N+1,M))
C        SAVE UPPER (LM) VELOCITIES IN LOWER (LP) VELOCITIES
      ULP1=ULM1
      ULP2=ULM2
      VLP1=VLM1
      VLP2=VLM2
  150 CONTINUE
      FFZA(LBOT)=0.0
      FFZD(LBOT)=0.0
c
c        LOOP UP COLUMN, COMPUTE REPEATED QUANTITIES
      DEN(LBOT)=1./(HH+SS+2.*(FFZA(LAYRS)+FFZD(LAYRS)))
      FB(LBOT)=2.*(FFZD(LAYRS)-FFZA(LAYRS))*DEN(LBOT)
      DO 160 L=LAYRS,2,-1
      LM=L-1
      DEN(L)=1./(HH+SS+FFZA(LM)-FFZA(L)+FFZD(LM)+FFZD(L)-FB(L+1)*
     1 (FFZA(L)+FFZD(L)))
      FB(L)=(FFZD(LM)-FFZA(LM))*DEN(L)
  160 CONTINUE
      DEN(1)=1./(HH+SS+2.*(FFZD(1)-FFZA(1)-FB(2)*(FFZA(1)+FFZD(1))))
C
C        SALINITY CALCULATIONS
      GOTO(200,250),ISALT
  200 CONTINUE
C        BOTTOM CONDITIONS
      L=LBOT
      CFLUX=               FFXM(L)*SM(L,N)
     1 +FFYM(L)*SC(L,NM)+FFCC(L)*SC(L,N)+FFYP(L)*SC(L,NP)
     2                    +FFXP(L)*S(L,N,MP)
      FA(L)=CFLUX*DEN(L)
C        CORE SALINITY
      DO 230 L=LAYRS,2,-1
      CFLUX=               FFXM(L)*SM(L,N)
     1 +FFYM(L)*SC(L,NM)+FFCC(L)*SC(L,N)+FFYP(L)*SC(L,NP)
     2                    +FFXP(L)*S(L,N,MP)
      FA(L)=(CFLUX+FA(L+1)*(FFZA(L)+FFZD(L)))*DEN(L)
  230 CONTINUE
C        TOP CONDITIONS
      L=1
      CFLUX=               FFXM(L)*SM(L,N)
     1 +FFYM(L)*SC(L,NM)+FFCC(L)*SC(L,N)+FFYP(L)*SC(L,NP)
     2                    +FFXP(L)*S(L,N,MP)
      S(1,N,M)=(CFLUX+2.*FA(2)*(FFZA(1)+FFZD(1)))*DEN(1)
c        INVERT HERE
      DO L=2,LBOT
      S(L,N,M)=FA(L)+FB(L)*S(L-1,N,M)
      ENDDO
C
C        TEMPERATURE
  250 GOTO(260,390),ITEMP
C        TOP AND BOTTOM UPWARD HEAT FLUX
  260 CALL HEATZ(N,M,TRAD,FTSURF)
C        BOTTOM CONDITIONS
      FTBOT=0.    ! heat lost into bottom sediment
      L=LBOT
      TFLUX=               FFXM(L)*TM(L,N)
     1 +FFYM(L)*TC(L,NM)+FFCC(L)*TC(L,N)+FFYP(L)*TC(L,NP)
     2                    +FFXP(L)*T(L,N,MP)
      FA(LBOT)=(TFLUX+HH*(TRAD(L)+FTBOT))*DEN(L)
C        CORE TEMPERATURE
      DO 300 L=LAYRS,2,-1
      TFLUX=               FFXM(L)*TM(L,N)
     1 +FFYM(L)*TC(L,NM)+FFCC(L)*TC(L,N)+FFYP(L)*TC(L,NP)
     2                    +FFXP(L)*T(L,N,MP)
      FA(L)=(TFLUX+HH*TRAD(L)+FA(L+1)*(FFZA(L)+FFZD(L)))*DEN(L)
  300 CONTINUE
C        TOP TEMPERATURES
      L=1
      TFLUX=               FFXM(L)*TM(L,N)
     1 +FFYM(L)*TC(L,NM)+FFCC(L)*TC(L,N)+FFYP(L)*TC(L,NP)
     2                    +FFXP(L)*T(L,N,MP)
      T(1,N,M)=(TFLUX+HH*(TRAD(1)+FTSURF)+2.*FA(2)*(FFZA(L)
     1 +FFZD(L)))/(HH+SS+2.*(FFZD(L)-FFZA(L)-FB(2)*(FFZA(L)+FFZD(L))))
c        INVERT HERE
      DO L=2,LBOT
      T(L,N,M)=FA(L)+FB(L)*T(L-1,N,M)
      ENDDO
  390 CONTINUE
  400 CONTINUE
c        make positive
      IF(ICPOS.EQ.0)RETURN
      DO 600 M=1,MMAX
      DO 600 N=1,NMAX
      DO 600 L=1,LBOT
      S(L,N,M)=AMAX1(S(L,N,M),0.0)
  600 T(L,N,M)=AMAX1(T(L,N,M),0.0)
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE HEAT1
C        APRIL 1985   K. W. HESS   MEAD   VAX 11/750
C        PURPOSE - TO COMPUTE OCEAN SURFACE HEAT FLUX CONSTANTS FOR
C                    THIS TIMESTEP FOR ALL CELLS
c        INPUTS -
C               IHEAT = HEAT FLUX INDEX:
C                          0=NO HEAT,
C                          1=HEAT FLUX BUT WITH DAY OF YEAR CONSTANT (=NDAY
C                          2=NORMAL HEATING
C               CLOUD = FRACTION OF SKY COVERED BY CLOUDS (0.0 - 1.0)
C               TAIRK = AIR TEMPERATURE (K) AT 10 M
C                 W10 = WIND SPEED AT 10 M (M/S)
C                  PA = ATMOSPHERIC PRESSURE (PASCALS)
C              RELHUM = RELATIVE HUMIDITY AT 10 M (0.0 - 1.0)
C                  TW = WATER TEMPERATURE AT NEAR-SURFACE (K)
C        VARIABLES -
C                COSZ = COSINE OF Z
C              CDRGAW = AIR-WATER INTERFACIAL DRAG COEFFICIENT
C                 DEC = SUN'S DECLINATION (ANGLE ABOVE ECLIPTIC)
C               EA,ES = SATURATION VAPOR PRESSURE (PASCALS)
C              HSOLAR = HOUR IN DAY ON A 24-HOUR CLOCK
C              NETDAY = DAY OF THE YEAR, COUNTED FROM JANUARY 1.
C                  T0 = AIR TEMPERATURE AT SEA SURFACE (K)
C        CONSTANTS -
C                 ALB = NET ALBEDO
C                 ALV = LATENT HEAT OF VAPORIZATION (2.6 X 10 6 J/KG)
C                RHOA = AIR DENSITY (KG/M3)
C                RHOW = SEA WATER DENSITY (KG/M3)
C               CPAIR = SPECIFIC HEAT CAPACITY OF DRY AIR(J/KG/K)
C              CWATER = SPECIFIC HEAT CAPACITY OF WATER (J/KG/K)
C                  SB = STEFAN-BOLTZMAN CONSTANT
      INCLUDE 'COMM20.FOR'
      DIMENSION VAL(4)
      DATA NDAY0/180/,IP/0/
C        CLOUDINESS FUNCTION
      FCC(X)=1.-X
      T2KEL(TEMP)=TEMP+273.*(.5-SIGN(.5,TEMP-200.))
C        GET DATA
      IF(NSIGM.EQ.0)THEN
      TAIRC=20.
      RELHUM=.7
      CLOUD=.5
      PA=1014.0*100.
      ELSE
      CALL RR(YT,ISCR,LUMET,IENDMT,DMET,YMET,NSIGM,4,VMET,VAL)
      TAIRC=VAL(1)
      RELHUM=VAL(2)
      CLOUD=VAL(3)
      PA=VAL(4)*100.
      ENDIF
      IF(IHEAT.EQ.1)NDAY=NDAY0
      IF(IHEAT.EQ.2)NDAY=UT
      IF(IPRNT1.EQ.1)WRITE(ISCR,100)TAIRC,RELHUM,CLOUD,PA,NDAY
  100 FORMAT(1X,'HEAT1: TAIRC=',F5.2,' RELHUM=',F5.3,' CLOUD=',F4.2,
     1 ' PA=',F10.2,' NDAY=',I3)
C        NEAR-SURFACE AIR TEMPERATURE
      TAIRK=TAIRC+273.
      T10=TAIRK
C        SOLAR ANGLES
      HSOLAR=24.*(UT-FLOAT(IFIX(UT)))-.5*DTI/3600.
      DEC=23.44*COS(FLOAT(172-NDAY)*PI/182.5)
      COSZ=SIN(BSNLAT*RAD)*SIN(DEC*RAD)+COS(BSNLAT*RAD)*COS(DEC*RAD)*
     1 COS((12.-HSOLAR)*PI/12.)
C        CONSTANTS
      CSOL1=FCC(CLOUD)*(1.-ALB)*SOLAR*(COSZ**2)
      CSOL2=0.10+1.085*COSZ
      CSOL3=(COSZ+2.7)/(10**5)
C        10-METER VAPOR PRESSURE
      ES=611.0*10.**(7.5*(T10-273.16)/(T10-35.86))
      EA=RELHUM*ES
      CEVP2=EA/(PA-(1.-EPSLON)*EA)
      CTOT=1./(RHOW*CWATER)
      CH1=SB*CTOT
C        ATMOSPHERIC LONG WAVE INWARD
      A=MIN0(1,IHEAT)
      QA=A*SB*TAIRK**4*(1.-0.261*EXP(-0.000777*(273.-TAIRK)**2))
      IF(IPRNT1.EQ.1)WRITE(ISCR,200)COSZ,CSOL1,CSOL2,CSOL3
  200 FORMAT(1X,'COSZ,CSOL1,2,3=',4E10.2)
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE HEATZ(N,M,TRAD,FTSURF)
C        APRIL 1985   K. W. HESS   MEAD   VAX 11/750
C        PURPOSE - TO COMPUTE OCEAN SURFACE HEAT FLUXES INTO A
C                    PARTICULAR CELL
C        VARIABLES -
C               CSENS = COEFFICIENT FOR SENSIBLE HEAT GAIN (SEE HEAT1)
C               EA,ES = SATURATION VAPOR PRESSURE (PASCALS)
C              EPSLON = RATIO OF MOLECULAR WEIGHTS OF VAPOR AND DRY AIR
C                       (O2 + H2 + N2)
C              FTSURF = SURFACE LAYER HEATING (DEG C)
C                  PA = ATMOSPHERIC PRESSURE (PASCALS)
C      QI,QA,QB,QE,QS = heat inputs (Watts/m**2/K**4)
C              RELHUM = RELATIVE HUMIDITY (0.0 < RELHUM < 1.0)
C                  SB = STEFAN-BOLTZMAN CONSTANT (Watt/meter**2/K**4)
C               TAIRK = AIR TEMPERATURE (K)
C                  T0 = AIR TEMPERATURE AT SEA SURFACE (K)
C                TRAD = CHANGE IN TEMPERATURE OVER ONE TIMESTEP (DEG C)
C                  TW = WATER TEMPERATURE AT NEAR-SURFACE (K)
      INCLUDE 'COMM20.FOR'
      DIMENSION TRAD(LSIZE)
      DATA IP/0/
      T2KEL(TEMP)=TEMP+273.*(.5-SIGN(.5,TEMP-200.))
      IP1=0
```

```
      IF(NPRMN.EQ.0)GOTO 100
      DO J=1,NPRMN
      IF(IP.EQ.1.AND.IPRNT1.EQ.1.AND.N+1000*M.EQ.IPRMN(J))IP1=1
      ENDDO
 100  CONTINUE
C        SET CONSTANT
      A=MIN0(1,IHEAT)
C        WATER SURFACE TEMP (C)
      TWS=T2KEL(.5*(T(1,N,M)+T(2,N,M)))
C        SET AIR TEMP JUST ABOVE SURFACE (K) EQUAL TO WATER TEMP
      FBETA=.9
      T0=FBETA*TWS+(1.-FBETA)*TAIRK
c        BULK TRANSFER COEFFICIENT
      W10=SQRT(WX(N,M)**2+WY(N,M)**2)
      CDRAGW=0.0025+SIGN(.0015,TWS-TAIRK)    ! HIGHER IF UNSTABLE (TW > TA
      CSENS=RHOA*CDRGAW*CPAIR*AMAX1(W10,0.2)
      CEVP1=ALV*RHOA*CDRGAW*EPSLON*AMAX1(W10,0.2)
c        print
      IF(IP1.EQ.1)WRITE(ISCR,105)UT,N,M
 105  FORMAT(/,1X,'HEATZ: UT=',F10.4,'  N,M=',2I4)
      IF(IP1.EQ.1)WRITE(ISCR,110)TAIRK,TWS,T0,W10
 110  FORMAT(1X,'TAIRK,TWS,T0 (K)=',3F8.2,'  W10=',F6.2)
C        SATURATION VAPOR PRESSURE AT AIR-WATER INTERFACE
      ES=611.0*10.**(7.5*(T0-273.16)/(T0-35.86))
      EA=RELHUM*ES
C        SOLAR SHORT-WAVE INWARD (Watt/m**2)
      QI=0.0
      IF(COSZ.GT.0.0)QI=A*AMAX1(0.0,CSOL1/(CSOL2+CSOL3*EA))
C        WATER BLACK BODY INWARD
      QB=A*(-.97)*SB*(TWS)**4
C        SENSIBLE HEAT GAIN
      QS=A*CSENS*(TAIRK-TWS)
C        EVAPORATIVE HEAT GAIN
      QE=A*CEVP1*(CEVP2-EA/(PA-(1.-EPSLON)*EA))
      QSUM=QA+QS+QE+QB
      IF(IP1.EQ.1)WRITE(ISCR,130)QB,QS,QA,QE,QSUM
 130  FORMAT(1x,'QB,QS,QA,QE,QSUM=',5E10.3)
C        TOTALS:  FQ, FS HAVE UNITS (DEG C)*M/S    CTOT=1./(RHOW*CWATER)
      FQ=CTOT*QSUM
C        SURFACE LAYER HEATING: UNITS=DEG C
      FTSURF=FQ*DTI/((D(N,M)+SE(N,M))*HALFDQ)
      IF(IP1.EQ.1)WRITE(ISCR,*)'FTSURF=',FTSURF
C        INTERNAL SOLAR HEATING:   TRAD units= deg C
      FS=CTOT*QI        !    deg C x m/s
      H=D(N,M)+SE(N,M)
      IF(IP1.EQ.1)WRITE(ISCR,*)'FS=',FS,' H=',H
      FD10=2.30/D10PCT
      DO 220 L=1,LBOT
      TRAD(L)=0.
      ZTOP=H*AMIN1(0.0,DQ*FLOAT(1-L)+HALFDQ)
      IF(ZTOP.LT.-2.*D10PCT)GOTO 220
      ZBOT=H*AMAX1(-1.,DQ*FLOAT(1-L)-HALFDQ)
      TRAD(L)=FS*DTI*(EXP(FD10*ZTOP)-EXP(FD10*ZBOT))/(ZTOP-ZBOT)
      IF(IP1.EQ.1)WRITE(ISCR,150)L,ZTOP,TRAD(L)
 150  FORMAT(1X,'L=',I2,'  ZTOP=',F6.2,'  TRAD=',E10.4)
220   CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE SETSTP

C        JUNE 1985   K. W. HESS   MEAD   VAX 11/750
C        PURPOSE - SET UP THE INITIAL FIELDS OF SALINITY AND
C                  TEMPERATURE BY INTERPOLATING FROM THE BOUNDARY
C                  CONDITIONS. SET INITIAL HORIZONTAL PRESSURE GRADIENTS
C        VARIABLES -
C                  ICS = INDEX FOR READING INITIAL CONDITIONS
C                        (0=N0, 1=YES)
C              NBCELL = TOTAL NUMBER OF BOUNDARY GRIDS (RIV + OCEAN)
      INCLUDE 'COMM20.FOR'
      DIMENSION NO(NM2SIZ),MO(NM2SIZ),R(NM2SIZ),SBNDP(LSIZE,NM2SIZ)
C        SET BOUNDARY STATE
      CALL BSTATE
      IF(ICS.EQ.1)GOTO 310
C        SET DEFAULT VALUES
      DO 100 L=1,LBOT
      DO 100 N=1,NM2SIZ
      SBNDP(L,N)=SBND(L,N)
      IF(N.GT.NBCELO.AND.KONCEN.EQ.2)SBNDP(L,N)=SAL0
      IF(N.GT.NBCELO.AND.MOD(KONCEN+1,2).EQ.0)SBNDP(L,N)=5.
 100  continue
C        SET OCEANIC BOUNDARY VALUES
      J=0
      IF(NUMOBC.LE.0)GOTO 130
      DO 120 I=1,NUMOBC
      DO 120 N=NB1(I),NB2(I)
      DO 120 M=MB1(I),MB2(I)
      J=J+1
      MO(J)=M
      NO(J)=N
      DO 120 L=1,LBOT
      S(L,N,M)=SBNDP(L,J)
      T(L,N,M)=TBND(L,J)
 120  CONTINUE
C        SET RIVER BOUNDARY VALUES
 130  IF(NUMRIV.LE.0)GOTO 200
      DO 140 I=1,NUMRIV
      DO 140 M=MR1(I),MR2(I)
      DO 140 N=NR1(I),NR2(I)
      J=J+1
      MO(J)=M
      NO(J)=N
      DO 140 L=1,LBOT
      S(L,N,M)=SBNDP(L,J)
      T(L,N,M)=TBND(L,J)
 140  CONTINUE
C        LOOP THRU THE INTERIOR COMPUTATIONAL GRIDS
 200  DO 300 M=1,MMAX
      DO 300 N=1,NMAX
      IF(IFIELD(N,M).LT.10.OR.IFIELD(N,M).GE.10*(KOCNBC))GOTO 300
      IF(D(N,M).LE.0.0)WRITE(ISCR,240)N,M
 240  FORMAT(1X,'*** SETUP: NO DEPTH AT N=',I3,',  M=',I3)
C        FIND RADAII TO BOUNDARY DATUM, AND SUM OF RADAII
      RSUM=0.0
      DO 250 I=1,NBCEL
      R(I)=1./(0.25+(FLOAT((N-NO(I))**2)+FLOAT((M-MO(I))**2)))
 250  RSUM=RSUM+R(I)
 255  RINV=1./RSUM
      DO 270 L=1,LBOT
      S(L,N,M)=0.0
      T(L,N,M)=0.0
      DO 260 I=1,NBCEL
      S(L,N,M)=S(L,N,M)+(R(I)*RINV)*SBNDP(L,I)
      T(L,N,M)=T(L,N,M)+(R(I)*RINV)*TBND(L,I)
 260  CONTINUE
 270  CONTINUE
 300  CONTINUE
 310  CONTINUE
      RETURN
      END
c=======================================================================
C          COMMS.FOR - COMMON BLOCKS FOR PROGRAM MECCA
       PARAMETER (NSIZE = 34, MSIZE= 55, LSIZE= 10, NMSIZE=NSIZE*MSIZE,
     1 LNMSIZ=LSIZE*NSIZE*MSIZE, NPMSIZ=NSIZE+MSIZE+2, NM2SIZ=2*(NSIZE+
     2 MSIZE), NDTID2=5, NDRIV2=10, NDOCN2=LSIZE, NDMET2=4)
C          MODEL CONSTANTS
       COMMON/CONS1/AG,OMEGA,PI,RAD,E,VONKAR,CRICH(8),RIMIN,RIMAX,CR0,
     1 IEXTRN,COR,NCOR,MCOR,DFDN,DFDM,BSNLAT,BSNANG,NONLIN,DL,CDRGWB,
     2 FCOR0,VERS,FEDGE(NSIZE,MSIZE),IHDSE,CDWB1,CDWB2,RAMPG,IGRADP,
     3 RAMPT,RAMPW,IBETAA,IBETAP,IBETAH,RHOBAR,INTER,SAL0,TMP0,CS0,CS1,
     4 CST,CT1,CT2,ISTOP,ICS,KTEST,NDAYMO(12),IHVISC,AH00,AH0,CAH,RHOA,
     5 RHOW,DHAH,DTE,DTI,ISPLIT,NSTI,NSTIT,NSTE,NSTET,NSTIMX,DTAU1,DTAU2
C          OUTPUT FORMAT VARIABLES
       PARAMETER (NDPRN=20, NDSLC1=5, NDSLC2=20, NDGPH=20)
       CHARACTER*10 CTITLE,PTITLE
       REAL*8 YT
       COMMON/TIME1/NST0,UT,UT0,UT1,HR,HR0,HR1,IHR,YT,IYEAR,MONTH,
     1 IDAY,IHOUR,IMIN,YEAR,YEAR0,YEAR1,CUMDAY,NMAX,MMAX,NCELL,DMAX,
     2 NEGS,ICHECK,ICOR,IBOTV,ITOPV,IVISC,NVISC,IPRNT1,JPRNT(15),
     3 KPRNT1,KPRNT2,HRMAX,HROUT,HROUT0,NPRMN,IPRMN(NDPRN),
     4 CTITLE(8),ISLICE,JSLICE(NDSLC1),NSLICE(NDSLC2,NDSLC1),MSLICE(
     5 NDSLC2,NDSLC1),ISTART,NGPMAX,IGPH,NSTGPH,IGPHOP,HRSAVE,
     6 LGPH(NDGPH),MGPH(NDGPH),NGPH(NDGPH),ITYP(NDGPH),PTITLE(25)
C          MODEL GRID AND GEOGRAPHY VARIABLES
       COMMON/GEOG1/BSNLON,CON2M,HMSL,NUMRIV,NCOL1,IFIELD(NSIZE,MSIZE),
     1 ICOL(5,NM2SIZ),IROW(5,NM2SIZ),NCOL,NROW,NAB(NM2SIZ),DMIN,KOCNBC,
     2 KRIVBC,IBARR,GRX(LSIZE),GRY(LSIZE),AREA(NSIZE,MSIZE),
     3 BX(NSIZE,MSIZE),BY(NSIZE,MSIZE),THETA1(NSIZE,MSIZE),
     4 THETA2(NSIZE,MSIZE),THETA3(NSIZE,MSIZE),NUMBXY,Z0
C          TIDE, MET, AND RIVER VARIABLES
       COMMON/TIDES1/NSIGT,IENDTD,YTID(2),DTID(2),TDLEV(2,NDTID2),
     1 IENDWN,WX(NSIZE,MSIZE),WY(NSIZE,MSIZE),DPADX,DPADY,
     2 FX(2,NSIZE,MSIZE),FY(2,NSIZE,MSIZE),DENRAT,CDR1,CDR2,TAIRK,TAIRC,
     3 TSX(NSIZE,MSIZE),TSY(NSIZE,MSIZE),IHEAT,NSIGW,DWND(2),YWND(2),
     4 PA,CLOUD,RELHUM,NSIGM,DMET(2),YMET(2),VMET(2,NDMET2),IENDMT,
     5 NSIGR,YRIV(2),DRIV(2),QRIV(2,NDRIV2),RATE(NDRIV2),ISETR(NDRIV2),
     6 IENDRV,MR1(NDRIV2),MR2(NDRIV2),NR1(NDRIV2),NR2(NDRIV2),IENDRT,
     7 ITPR(NDRIV2),JTPR(NDRIV2),NSIGRT,YRVT(2),DRVT(2),TRIV(2,NDRIV2)
C          OCEAN BOUNDARY CONDITIONS
       COMMON/FLAG1/NUMOBC,MB1(NDOCN2),MB2(NDOCN2),NB1(NDOCN2),NB2(
     1 NDOCN2),ITPO(NDOCN2),JTPO(NDOCN2),ISET1(NDOCN2),ISET2(NDOCN2),
     2 NSIGS,YSAL(2),DSAL(2),SALOCN(2,NDOCN2),IENDSO,NSIGTO,YOTP(2),
     3 DOTP(2),TMPOCN(2,NDOCN2),IENDTO,NBCEL,NBCELO,SBND(LSIZE,NM2SIZ),
     4 TBND(LSIZE,NM2SIZ),NSTINF(LSIZE,NM2SIZ),ITPO2(NDOCN2)
C          INTERNAL MODE (3-D) VARIABLES
       COMMON/TRID1/LAYRS,LBOT,DQ,HALFDQ,TWODQ,AH3(LSIZE,NSIZE,MSIZE),
     1 U(LSIZE,NSIZE,MSIZE),V(LSIZE,NSIZE,MSIZE),UE(NSIZE,MSIZE),
     2 VE(NSIZE,MSIZE),AV(LSIZE,NSIZE,MSIZE),ISIDE,AV0,AV00,WC(LSIZE),
     3 W(LSIZE,NSIZE,MSIZE),DV(LSIZE,NSIZE,MSIZE),DV0,DV00,CDRGWS,
     4 RI(LSIZE,NSIZE,MSIZE),AH(NSIZE,MSIZE),AHC(NSIZE,MSIZE),THETSU(
     5 NSIZE,MSIZE),THETSV(NSIZE,MSIZE)
C          EXTERNAL MODE VARIABLES
       COMMON/VELS1/UH(NSIZE,MSIZE),UHP(NSIZE,MSIZE),VH(NSIZE,MSIZE),
     1 VHP(NSIZE,MSIZE),D(NSIZE,MSIZE),SE(NSIZE,MSIZE),SEP(NSIZE,MSIZE),
     2 SEPP(NSIZE,MSIZE),FA(NPMSIZ),FB(NPMSIZ),GA(NPMSIZ),GB(NPMSIZ),
     3 ANB,ANC,SOLD(NSIZE,MSIZE),UHOLD(NSIZE,MSIZE),VHOLD(NSIZE,MSIZE),
     4 PHI(NSIZE,MSIZE),TBX(NSIZE,MSIZE),TBY(NSIZE,MSIZE),ANA
C           CONCENTRATION (SALINITY AND TEMPERATURE) VARIABLES
       COMMON/CONC1/KONCEN,ICPOS,ICOUPL,CI(LSIZE),RVERT,HRCONC,
     1 S(LSIZE,NSIZE,MSIZE),MFLUX(NSIZE,MSIZE),NFLUX(NSIZE,MSIZE),
     2 GSTARX(NSIZE,MSIZE),GSTARY(NSIZE,MSIZE),HRCON1,HRCON2,T(LSIZE,
     3 NSIZE,MSIZE),COSZ,CSOL1,CSOL2,CSOL3,SB,CSENS,CEVP1,CEVP2,D10PCT,
     4 EPSLON,CTOT,QSUM,QA,QI,QB,QS,QE,EA,ES,SOLAR,ALB,CWATER,CPAIR,ALV
C           FILE HANDELING
       CHARACTER*40 FCON,FGEO,FINIT,FPRINT,FGRAPH,FMED
       COMMON/FILE1/FCON,FGEO,FINIT,FPRINT,FGRAPH,FMED,ISCR,LUKB,IO,
     2 LUGRF,LUCON,LUMED,LUTID,LUWND,LURIV,LUSAL,LUOCT,LUMET,LURVT
```