PB84-129857

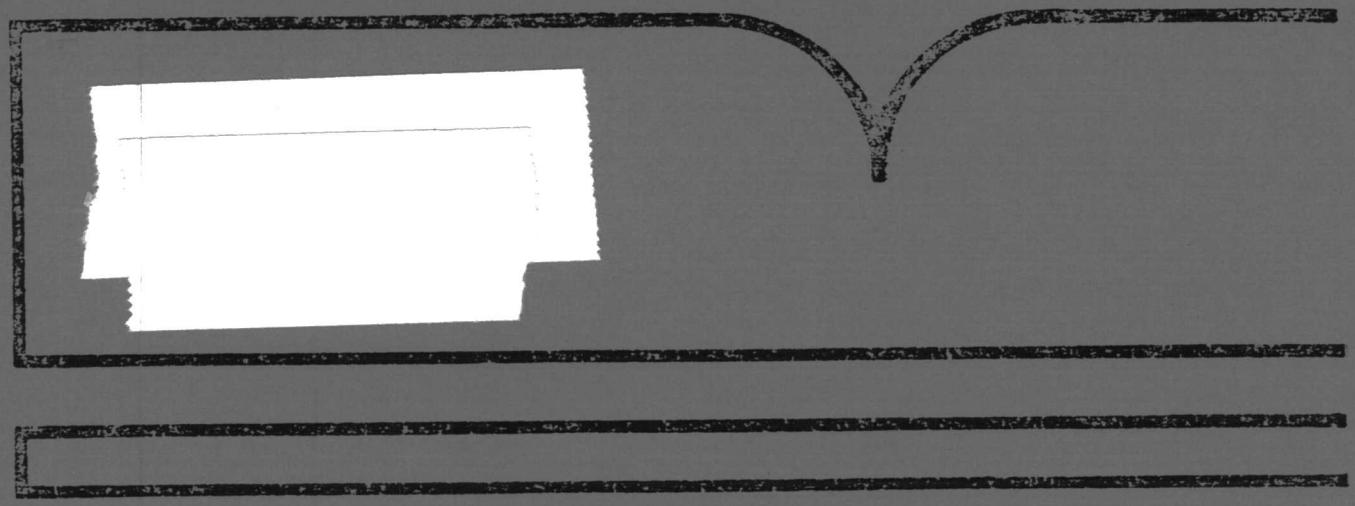
Southwest Florida Shelf Circulation Model
Volume 2. Model Documentation

New England Coastal Engineers, Inc., Bangor, ME

Prepared for

Minerals Management Service, Metairie, LA

Jul 82



PB84-129057

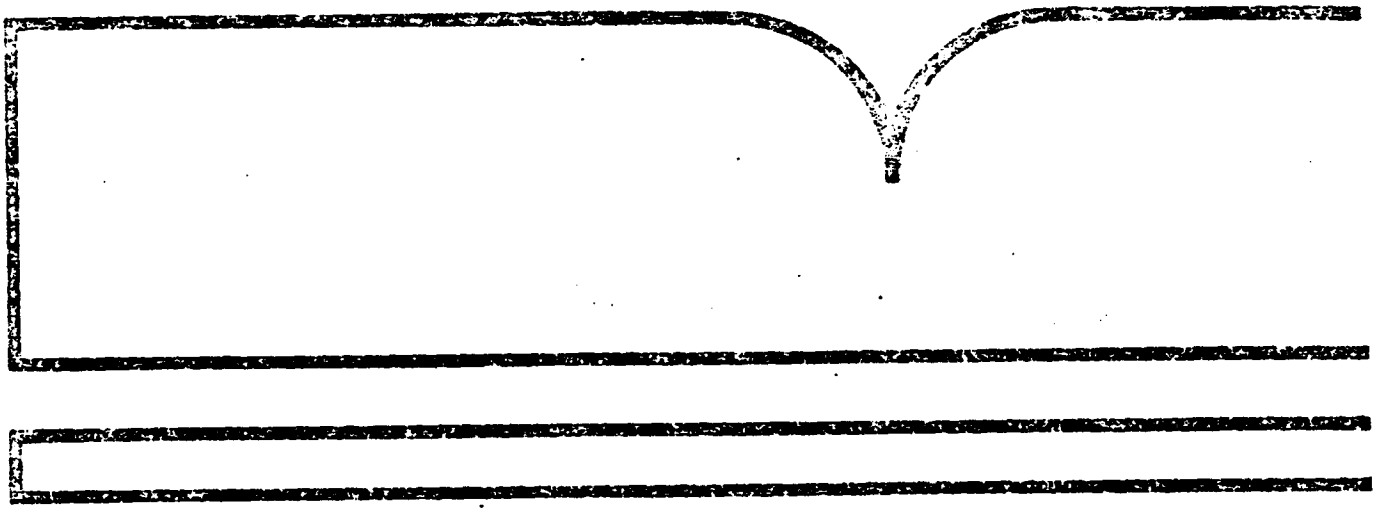
Southwest Florida Shelf Circulation Model
Volume 2. Model Documentation

New England Coastal Engineers, Inc., Bangor, ME

Prepared for

Minerals Management Service, Metairie, LA

Jul 82



U.S. Department of Commerce
National Technical Information Service
NTIS

FINAL REPORT
SOUTHWEST FLORIDA SHELF
CIRCULATION MODEL
VOLUME II

Prepared for:

United States Minerals Management Service
Contract No. AA851-CTO-72
July 1982

Prepared by:

NECE, Inc
Penobscot Plaza
Bangor, ME 04401

DISCLAIMER

This report has been reviewed by the Minerals Management Service and approved for publication. Approval does not signify that the contents necessarily reflect the views and policies of the Bureau, nor does mention of trade names or commercial products constitute endorsement or recommendation for use.

TABLE OF CONTENTS

LIST OF FIGURES

APPENDIX H

User's Manual

Appendix H

WEST FLORIDA SHELF CIRCULATION MODEL USER'S MANUAL

Introduction

A total of eight programs are described in this manual. All are written in FORTRAN IV and have been compiled and executed on an IBM VM 370 Operating system using both the G and H (with optimization) compiler. The circulation model is composed of five main programs: VISCOUS, DENSITY, CIRC, PRTVEL, and PLOTVEL. These programs are collectively known as GAL. Two programs, WIND and SPAT, have been provided to calculate the wind field for GAL. Figure H.1 summarizes the relationship between the seven programs. The eighth program, DENSTAT, is a useful utility program.

The CPU time needed to execute varies from one program to another. In general all the programs in GAL except CIRC require less than five seconds of CPU time on the IBM 4341 at the University of Maine. CIRC takes approximately two seconds of CPU time per hour of modeled time. WIND and SPAT generally execute in less than five seconds. DENSTAT can take a considerable amount of time, depending on the amount of data to be read from tape.

The notation used in this manual is consistent with that in Chapter 3 and Appendix E. The reader should be familiar with this notation before proceeding.

Each program has the following function:

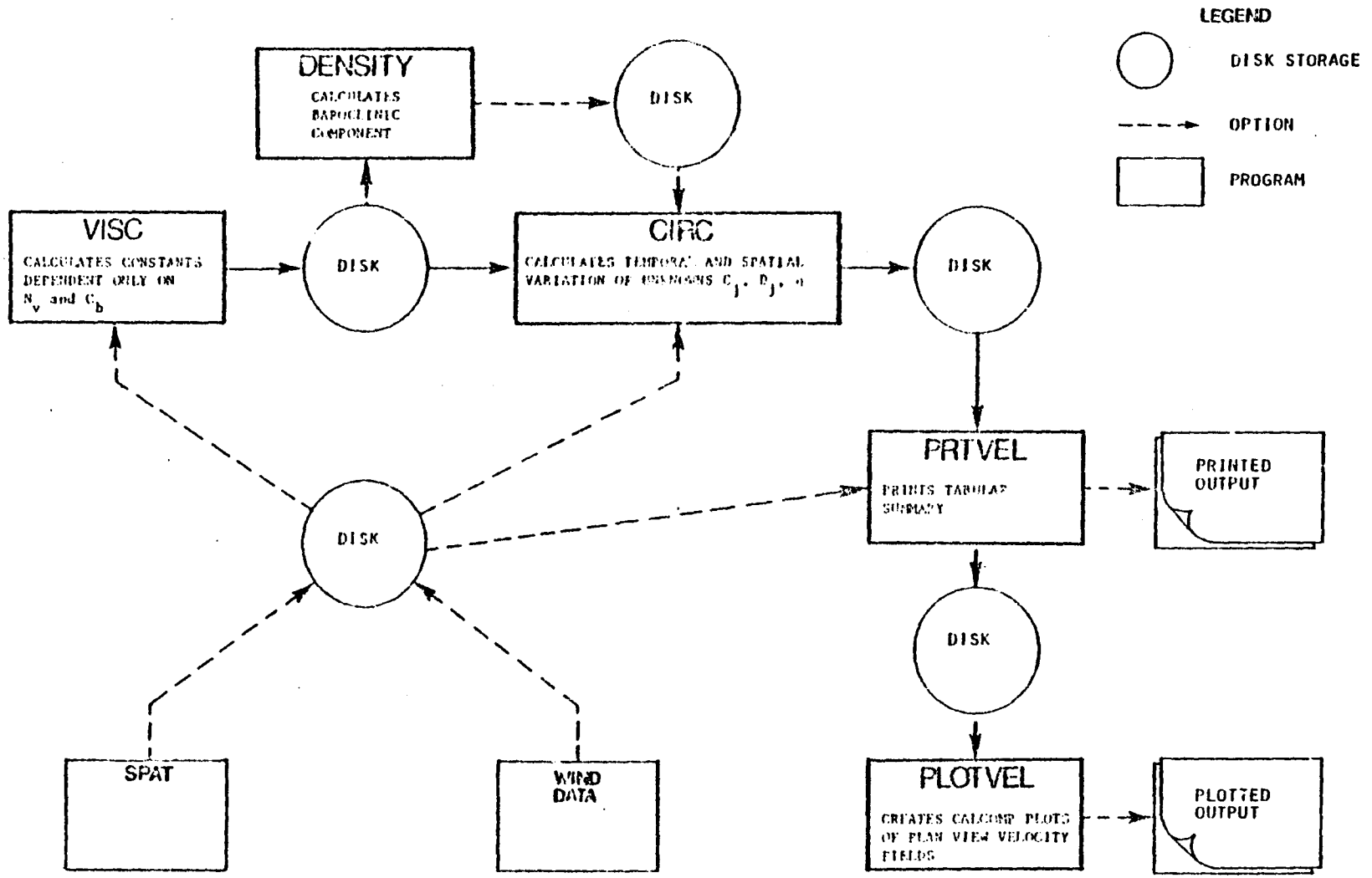


Figure H.1: Flow chart of GAL.

1. VISCOUS calculates parameters dependent on the: vertical eddy viscosity (N_v), the bottom friction coefficient (c_b), and the grid configuration. If these do not change there is no need to rerun VISCOUS. The program writes one disk file which is read by DENSITY, CIRC, and PRTVEL.

2. DENSITY calculates all parameters associated with the density-driven component of flow. It is an optional program that need be run only if hydrographic data indicate density-driven flows to be important. The user must specify either the density field or the salinity and temperature field. These values are specified for each horizontal grid element as well as for vertical layers. The number of layers is specified by the user and density is assumed constant over a layer. The program reads the disk file generated by VISC and creates a disk file which is read by CIRC and PRTVEL.

3. CIRC is the main computational program which calculates the unknowns c(j,l,m), d(j,l,m), and eta(l,m). The program reads disk files created by VISCOUS, DENSITY (if specified) and wind data. The program writes a disk file containing the temporal change of various parameters specified by the user such as c(j,l,m), d(j,l,m), ub(l,m), etc.

4. PRTVEL is the main ouput program. It reads disk files created by VISCOUS, DENSITY (if specified), wind data, and CIRC, and it writes a disk file read by PLOTVEL. PRTVEL creates summary tables of input parameters such as water depth, bottom friction coefficients, surface density, N_v, etc. PRTVEL also prints results such as vertical velocity profiles at specified locations, plan velocity views at specified depths, surface levels, and mass fluxes.

5. PLOTVEL creates a series of CALCOMP plots. Each plot consists of a plan view of current vectors at a specified depth. The program reads from a disk file created by PRTVEL.

6. WIND calculates and writes a disk file containing the wind friction velocity squared at each grid element. The program specifies a spatially constant wind field but allows an initial temporal change using a ramp function. This program is useful for sensitivity studies and production runs.

7. SPAT generates a spatially and temporally varying wind field for the model grid system by linearly interpolating concurrent wind measurements from specified observation points. The program is useful for making hindcasts of real time currents.

8. DENSTAT calculates first order statistics for the density field. Density data is read from a tape in NODC format. The program generates a summary table listing the statistics for one-half degree squares at four levels in the water column. The user specifies the months which are to be searched and summarized. Unlike the seven other programs listed above, DENSTAT is a utility program which is not intended to interface directly to GAL or the wind field programs.

All programs make use of the FORTRAN command NAMELIST for most data input. Users unfamiliar with this command are urged to consult a FORTRAN Reference before proceeding.

Most of the programs read and write to disk files. This requires that the user include in his Job Control Language (JCL) the appropriate commands to access and link the disk. On the IBM VM 370 OS the commands are:

```
CP LINK username device1 device2 WR password
ACCESS device2 mode
```

where the lowercase denotes items which are installation dependent. The user should consult his local computer support staff to determine the appropriate

items.

The following sections give a more detailed description on usage of the eight programs.

H.1 VISCOUS

Inputs

First Record

The first record should contain the title. This can be any sequence of up to 80 alphanumeric characters.

Namelist &MISC

JP	the number of cosine terms used in the trial functions; j'. Default: JP=3.
KP	the number of linear segments +1 used for N_v; k'. Default: KP=2.
LP	the number of elements in the x direction; l'. Default: LP=12.
MP	the number of elements in the y direction; m'. Default: MP=24.
NS	the number of trapezoids used in the numerical integration. Default: NS=20.
CONNV	logical variable which indicates the status of N_v where: CONNV=.TRUE. indicates N_v is to be taken as constants in the vertical. In this case KP should be set equal to 2. CONNV=.FALSE. indicates N_v is not constant in the vertical

and that KP will be greater than two. Default: CONNV=.FALSE.

NUCHG — the number of times during the model run that N_v will change. Note that if NUCHG.GT.1 more than one Namelist &EDYVIS should appear in the input file. Also, if EDDYT=.TRUE. (see below), N_v will automatically be updated each time the wind changes so NUCHG will be ignored. Default: NUCHG=1

Namelist &DEPTH

H — the still water depths in meters. If tides at the ocean boundaries are to be specified then H should be referenced to MLW (mean low water). Must be LP*MP values specified (Note the use of FORTRAN convention which uses a * to signify multiplication. This notation will be used consistently in this text). Entering sequence should be first row of grid (i.e. +x direction), followed by second row, etc. Default: H(1,m)=0.

Namelist &FRICT

CB — the friction coefficient in m s^{-1}. Must be LP*MP values specified. Entering sequence is same as for H. Default: CB=0.00025.

Namelist &EDDYL

CK — the layer depth of the eddy viscosity (meters). Must be KP*LP*MP values. the entering sequence is:

```
CK=k(1,1,1),k(2,1,1),. . .k(KP,1,1),k(1,2,1)
    . . . . .k(KP,2,1), . .k(KP,LP, MP),&END
```

where k(k,l,m) is the depth of N_v(k,l,m).

NODIM — logical variable indicating whether the units of CK are to be input as meters or as a fraction of the total depth. NODIM=.TRUE. indicates units of CK are dimensionless; in other words CK will assume values 0.LE.CK.LE.1.0. NODIM=.FALSE. indicates units of CK are in meters; in other words CK will assume values 0.LE.CK(k,l,m).LE.H(l,m). Default: NODIM=.FALSE.

Namelist &EDYVIS

Note that if NUCHG has been set .GT.1 in &MISC then one &EDYVIS statement should appear NUCHG times.

EDDYT — logical variable indicating whether N_v is to be varied in time and space. If EDDYT=.TRUE. then N_v will be calculated each time the wind changes and for each element according to the formula by Townsend (1976):

$$N_v = H(1,m) * WA2(1,m,t) / REYNO$$

where WA2 is the surface friction velocity. WA2 is specified by a wind model (i.e. WIND or SPAT). Caution: N_v will be updated whenever the wind changes so this option may take a great deal of computer time if the wind changes frequently. Default: EDDYT=.FALSE.

EDDYS — Logical variable indicating whether N_v is to be varied in space using the Townsend expression. TRUE indicates the Townsend expression will be used to calculate N_v. FALSE indicates values of N_v will be specified by user (see EDDYV below). Default: EDDYS=.TRUE.

REYNO — the value of the flow Reynolds Number to be used to calculate N_v in the Townsend expression. Value of REYNO normally varies between 12 to 32. Default: REYNO=16.

WA2 — the value in m s^{-1} of the surface friction velocity to be used in the Townsend expression. Normally WA2 is based on a wind condition which represents an average for the period of the simulation. WA2 can be found by first assuming an average wind speed for the simulation period and then converting wind speed to WA2 using an expression such as Wu(1980). Default: = 0.005.

EDDYV — the eddy viscosity in $m^2 s^{-1}$ at each layer for each grid. Must be specified if EDDYS=.FALSE. and EDDYT=.FALSE. Must be KP*LP*MP values specified. Entering sequence is:

EDDYV=N_v(1,1,1),N_v(2,1,1), . . N_v(KP,1,1),
N_v(1,2,1), ...
N_v(KP,2,1), . . . N_v(KP,LP, MP),&END

where: $N_v(k,l,m)$ is the numerical value of N_v for grid location l,m and layer k.

ETIME — the time (in seconds) up until which EDDYV, or REYNO and WA2 will apply. If NUCHG=1 than ETIME should be set large. Default: ETIME=1×10^{32}

Dimensioning

```
 COMMON/AA/CK(KP*LP*MP),DELSIG(xx),SS(JP*LP*MP),H(LP*MP),FMASS(LP*MP),
&    B(JP*LP*MP),E(JP*LP*MP),D1(JP*LP*MP),ALPHA((KP-1)*LP*MP),
&    BETA((KP-1)*LP*MP),NB(LP*MP),Q(JP*LP*MP),
&    A(JP*LP*MP),CB(LP*MP),EDDYV(KP*LP*MP),JP,KP,LP,MP,LPMP,LM,KPM1,
&    JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
&    ETIME,NUCHG,UA2(LP*MP),VA2(LP*MP),VERSIN(14),TITLE(20),
&    CONSTE,EDDYT,EDDYS,CONNV,LAND(LP*MP),FXY,PHI,THETA,F(LP*MP)
```

where xx=JP*LP*MP when CONNV=.TRUE. and xx=JP*JP*LP*MP when CONNV=.FALSE.

Disk Files

The program writes a disk file on device #10 in an unformated, variable spanned format, and reads from disk file #4. The file definitions on the IBM VM 370 are:

```
FIDEF 4  DISK WIND DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
FIDEF 10 DISK VISC DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
```

Were 'mode' is as defined in the ACCESS command described in the INTRODUCTION.

Errors

ERROR 1 — the nondimensional value of CK(k,l,m) exceeds unity or is negative. Most probable cause: CK has been specified greater than the local water depth or has been specified

as negative. Check Namelist EDDYL.

ERROR 2 — the value of CK(1,1,m) is not equal to zero or CK(KP,1,m) is not equal to H(1,m) (or unity if NODIM=.TRUE.). Check Namelist EDDYL

ERROR 3 — KP has been specified greater than 2 and CONNV=.TRUE. This is inconsistent and the user should change either KP or CONNV so that they are consistent.

ERROR 4 — KP has been specified as 2 but CONNV=.FALSE. This is inconsistent and the user should change either KP or CONNV so that they are consistent.

ERROR 5 — KP has been specified as 1. KP must be two or greater.

ERROR 6 — N_v has been specified as zero which is physically unrealistic. Check Namelist EDYVIS.

H.2 DENSITY

This program implements horizontal density-driven currents as described in Appendix E.3. Notation shown below is taken from that appendix.

Inputs

First Record

The first record should contain the title. This can be any sequence of up to 80 alphanumeric characters.

Namelist &MISC

NUCHG — the number of times during the model run the density is expected to change. Used in conjunction with DTIME (see below). Default: NUCHG=1.

CALDEN — logical variable specifying whether density data will be directly specified (CALDEN=.FALSE.) or salinity and temperature data will be specified (CALDEN=.TRUE.), hence requiring calculation of density within DENSITY. Default: CALDEN= .FALSE.

Namelist &LAYER

IPP — the number of density layers in the vertical, i'. Not necessarily the same as KP used in Program VISCOUS. Default: IPP=2.

HP — the depth in meters referenced to SWL (still water level) of the bottom of each layer. Must be IPP values. Not necessarily the same as CK used in Program VISCOUS. Default: HP(1)=10, HP(2)=300.

DTIME — the time in seconds up until which the density data applies. Default: $=1 \times 10^{32}$.

Namelist &SALIN This namelist is optional and need be specified only if CALDEN=.TRUE.

SAL — the salinity in ppt (parts per thousand) for each layer of each element. Specified in horizontal layers or:

```
SAL=sal(1,1,1),sal(1,2,1),. . . .
sal(1,LP,1),sal(1,LP,2),. . . . sal(1,LP,MP),sal(2,1,1),. . .
sal(2,LP,1),sal(2,LP,2),. . . . . .
sal(IPP,LP,MP),&END
```

where for example sal(1,LP,2) would be the salinity in ppt at the upper level, i.e. HP(1).GT.z.LE.HP(2) at grid l=LP, m=2.

Namelist &TEMP

This namelist is optional and need be specified only if CALDEN=.TRUE.

T — the temperature in C^{o} for each layer and each element. Same entering sequence as SAL.

Namelist &RHO

This namelist is optional and need be specified only if CALDEN=.FALSE.
RHOWP the density in g cm^{-3}. Same entering sequence as SAL.

Dimensioning

```
 COMMON/AA/*DELHP(IPP),PHI(IPP),S(IPP),HP(IPP),H(LP*MP),T(IPP*LP*MP),
&          SAL(IPP*LP*MP),RHOWP(IPP*LP*MP),RHOW(IPP*LP*MP),
&          ROWX(IPP*LP*MP),ROWY(IPP*LP*MP),A(IPP*LP*MP),C(IPP*LP*MP),
&          CALDEN,LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
&          KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
&          ASTER,ETIME,LAND(LP*MP),BOTT(IPP*LP*MP)
```

Disk Files

The program reads from the disk file created by VISCOUS on file #10. The program writes disk file #11. The appropriate file definitions for the IBM VM 370 are:

```
FIDEF 10 DISK VISC DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
FIDEF 11 DISK DENSE DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
```

Errors

No execution time errors are written by the program.

H.3 CIRC

Inputs

First Record

The first record should contain the title. This can be any sequence of up to 80 alphanumeric characters.

Namelist &MISC

EDDYH	value of the lateral eddy viscosity in $m^2 s^{-1}$. Default: EDDYH=10^5.
DENSE	logical variable indicating whether the density driven term will be included in the computations. If TRUE then Program DENSITY should have been previously executed. Default: DENSE=.FALSE.

Namelist &TIME

DTT	the time step in seconds, delta t. Selection of DTT is critical for model stability. Consult Chapter 3 for the criteria which must be meet by DTT. Default: DTT=450.
TLAST	the last time step in seconds. Default: TLAST=360000.
ISTPR	the time step in seconds at which printed output (in CIRC and PRTVEL) and plots (in PLOTVEL) will begin. Default: ISTPR=90000.
IDTPR	the time increment in seconds between printed output (in CIRC and PRTVEL) and plots (in PLOTVEL). Default: IDTPR=45000.

Namelist &BCS

IAMP(l,m) indicates the type of boundary condition which is to be assumed at grid l,m. IAMP can assume the following values:

0 indicates a land element.

1 indicates the surface slope will be assumed zero in a direction normal to the boundary.

2 indicates an open ocean element where the tidal amplitude and period must be specified (see description of 'AMP' and PERIOD below).

3 not applicable in version 7-A.

4 indicates internal water element.

5 indicates c(j,l,m) and d(j,l,m) will be specified at the element. This condition is used to model the shear invoked by a current along the western boundary. Note that the c's and d's can be specified directly by the user or calculated internally by the program to provide a sinusoidal forcing at the boundary analogous to the eddy-wave field suggested by Niiler (1976). To excercise this latter option the characteristics of the eddy wave must be specified via WPEROD, WLNGTH, and WAMP (see below).

6 indicates the mass fluxes, UB and VB, will be specified at the element. This condition is used to model river inflow and outflow.

Note that IAMP is defaulted to '0' for all land elements, '1' for all non-land boundary elements, and '4' for all non-land, non-boundary elements.

PERIOD the period of the tidal oscillation in seconds. Default: PERIOD=44,500.

AMP — the amplitude of the tidal oscillation in decimeters at the western boundary. AMP is a two dimensional array of the form AMP(1,m) where 1 and m specify the grid element location. AMP should be specified as an integer and again note that the units are in decimeters. Default: AMP(1,m)=0.

UB,VB — Sources/sinks such as rivers can be modeled by specifying the mass fluxes, UB and/or VB. Units of UB and VB are $m^2 s^{-1}$. Note that IAMP must = 6. Entering sequence is same as for IAMP. Default: UB=VB=0.0

RLID — logical variable which specifies whether a rigid lid assumption is to be made. In other words, whether the surface elevation, eta, is to be set to zero throughout the grid. Default: RLID=.FALSE.

CBC,DBC — the value of the unknown coefficients, c(j,1,m) and d(j,1,m) respectively, in $m\ s^{-1}$ at the boundary. Note that IAMP(1,m) must = 5 and that j.LE.JP. Default: CBC=DBC=0.0

WPEROD — the period (seconds) of a sinusoid oscillation imposed at the western boundary. Can be used to model a Niiler eddy wave. Note that IAMP(1,m) must equal 5. If a temporally constant lateral shear is to be applied on the western boundary, WPEROD should be set to zero. Default: WPEROD=0.

WLNGTH — the length (meters) of the sinusoidal oscillation. Used in conjunction with WPEROD and WAMP. Default: WLNGTH=0.

WAMP — the wave amplitude ($m\ s^{-1}$) of the sinusoidal oscillation. Used in conjunction with WLNGTH and WPEROD. Default: WAMP=0.

Again note that if IAMP is set to 5 on the western boundary the user must either specify: (1) DBC and/or CBC in which case WLNGTH, WAMP and WPEROD should all be set to zero or (2) WAMP, WLNGTH, and WPEROD. The second option is used to simulate the Niiler eddy wave.

Namelist &INOUT

HOTSTR — logical variable indicating whether a "hot start" is to be made. If TRUE than CIRC will initialize the unknowns c(j,l,m), d(j,l,m) and eta(l,m) by accessing existing values on disk file #12 which should have be established by a prior run of CIRC. If HOTSTR=FALSE then these variables will initially be set to zero. Default: HOTSTR=.FALSE.

HOTIME — the time in seconds at which the "hot start" begins. Must be specified if HOTSTR=.TRUE. and must be .LE. the last time step executed by the prior run of the model.

PRTFLO — logical variable which indicates whether a printed summary of the flows in the x and y direction are desired in Program PRTVEL. Default: PRTFLO=.FALSE.

Dimensioning

```
 COMMON/EDDE/DELSIG(xx),SS(JP*LP*MP),H(LP*MP),FMASS(LP*MP),
&    B(JP*LP*MP),E(JP*LP*MP),D1(JP*LP*MP),CONNV,LAND(LP*MP)
 COMMON/UNKNOW/C(JP*LP*MP),D(JP*LP*MP),ETA(LP*MP),
&    UB(LP+1,MP+1),VB(LP+1,MP+1)
 COMMON/CON/JP,KP,LP,MP,KPM1,EDDYH,C2,C3,C4,C5,VERSIN(14),
&    AMP(LP,MP),IAMP(LP,MP),TIT1(20),JPLPMP,LPMP,JPLP,PRTFLO
&    MPM1,LPM1,CBC(JP,LP+1,MP+1),DBC(JP,LP+1,MP+1),HOTIME,HOTSTR
 COMMON/WINDY/UA2(LP*MP),VA2(LP*MP),HO(LP*MP),
&    FX(LP*MP),FY(LP*MP),TAUX(JP*LP*MP),TAUY(JP*LP*MP),DPDXJ(JP*LP*MP),
     DPDYJ(JP*LP*MP)
 COMMON/DEN/ROWX(JP*LP*MP),ROWY(JP*LP*MP),RHOW(LP*MP),IPP,DENSE,DTIME
```

where: xx=JP*LP*MP when CONNV=.TRUE. and xx=JP*JP*LP*MP when CONNV=.FALSE.

Disk Files

The program writes one disk file on device #12. It is an unformated, variable spanned file containing various parameters intended for Program PRTVEL. CIRC reads at least two files - file #4 which is the wind data and file #10 which are the parameters written by VISCOUS. If the variable DENSITY=.TRUE. in the CIRC input then CIRC will also access disk file #11, written by Program

DENSITY. The file definitions needed to run CIRC on the IBM 370 are:

```
FIDEF  4 DISK WIND  DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
FIDEF 10 DISK VISC  DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
FIDEF 11 DISK DENSE DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
FIDEF 12 DISK CIRC  DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
```

Errors

No error statements are generated by CIRC in this version.

H.4 PROGRAM PRTVEL

Inputs

Namelist &OUTOPS

PRTWND — logical variable which specifies whether a printed table of the x & y components of the wind friction velocity squared is to be created. Caution: a great deal of output can result with this option since a table will be generated each time the wind changes. Default: PRTWND=.FALSE.

PRTNV — logical variable which specifies whether a table of the N_v and associated parameters will be printed each time N_v changes. Default: PRTNV=.FALSE.

PRTDEN — logical variable which specifies whether a table of the surface density will be printed. Meaningful only if variable DENSE in Program CIRC has been set to TRUE. Table will be created each time the density changes Default: PRTDEN=.FALSE..

PRTETA — logical variable which specifies whether a table of the surface elevation at each element will be printed. Default: PRTETA=.FALSE.

NOGRDS — the number of elements where a table of the velocity profile is desired. If NOGRDS.GT.0, a print out will begin at ISTPR seconds and repeat every IDTPR seconds thereafter. Default: NOGRDS=0

LOCATE(I) — the coordinates of the grid elements for which a table of the velocity profile is desired. The number of LOCATE must be equal to NOGRDS. This variable is of the form: LOCATE(I)=(l,m) where l and m are the grid coordinates in the x and y direction, respectively.

LAYERS — the number of layers for which an instantaneous plan view of the velocities for the entire grid will be printed. Default: LAYERS=0.

LEVELS — the number of layers for which an instantaneous plan view of the velocities are to be plotted. Default: LEVELS=0

If NOGRDS .GT. 0 the namelist &OUTOPS should be immediately followed by NOGRDS records, each containing up to 16 constants in F5.0 format. Each record should be of the following form:

```
z(1,1)  z(1,2)  z(1,3) . . . . . z(1,b')
z(2,1)  z(2,2)  z(2,3) . . . . . z(2,b')
   .
   .
z(a',1) z(a',2) z(a',3). . . . . z(a',b')
```

where: z(a,b) is the depth in meters at which the velocity will be calculated for the grid element specified by LOCATE(a). z should be specified in F5.0 format. restrictions: a' = NOGRDS, b' .LT. 17 , and z(a,b) .LT. z(a,b+1)

If LAYERS .GT. 0, variables described in the previous paragraph must be followed immediately by 1 record containing up to 16 variables in F5.0 format. The record should be in the following form:

z(1) z(2) z(3) . . . . . .z(i)

where i= the no. of plan views which will be printed (must be .LT. 17) and z is the percentage depth (e.g. z=50 indicates that velocities at each element will be calculated at 50% of the local water depth), and expressed in F5.0 format. Print-out will occur starting at ISTPR seconds and repeating every IDTPR seconds thereafter. Note ISTPR and IDTPR are specified by the user in Program CIRC.

If LEVELS.GT.0, variables described in the previous paragraphs must be followed immediately by 1 record containing up to 16 variables in F5.0 format. The record should be in the following form:

z(1) z(2) . . . . . .z(I)

where: I= no. of plan views to be plotted (must be .LT. 17) and z is the percentage depth of the plan view expressed in F5.0 format. Plotting will start at ISTPR and repeat every IDTPR seconds thereafter.

Dimensioning

```
      COMMON/KNOW/H(LP*MP),CK(KP*LP*MP),ALPHA((KP-1)*LP*MP),
     &        BETA((KP-1)*LP*MP),NB(LP*MP),A(JP*LP*MP),CB(LP*MP),
     &        EDDYV(KP*LP*MP),CONNV,LAND(LP*MP)
      COMMON/UNKNOW/C(JP,LP+1,MP+1),D(JP,LP+1,MP+1),ETA(LP*MP),
     &        UB(LP*MP),VB(LP*MP),SPEED(LP*MP),DIREC(LP*MP)
      COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
     &        EDDYT,REYNO,WA2,AMP(LP*MP),IAMP(LP*MP),TIT1(20),TIT2(20),
     &        TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
      COMMON/WINDY/UA2(LP*MP),VA2(LP*MP)
      COMMON/DEN/ROWX(JP*LP*MP),ROWY(JP*LP*MP),RHOW(LP*MP),IPF,DENSE
```

Disk Files

One disk file is generated by PRTVEL on unit #13. This file contains

information for program PLOTVEL. PRTVEL reads at least three disk files: file #4 which is the wind data; file #10 written by Program VISCOUS; and file #12 written by Program CIRC. If the variable DENSITY=.TRUE. in Program CIRC than PRTVEL also accesses file #11, written by Program DENSITY. The file definitions on the IBM 370 are:

```
FIDEF  4 DISK WIND    DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
FIDEF 10 DISK VISC    DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
FIDEF 11 DISK DENSE   DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
FIDEF 12 DISK CIRC    DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
FIDEF 13 DISK PRTVEL  DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
```

Errors

No error statements can be generated by PRTVEL in this version.

H.5 PROGRAM PLOTVEL

Inputs

First Record

The first record should contain the title. This can be any sequence of up to 80 alphanumeric characters.

Namelist &MISC

FACT	a reduction factor. Plots will be generated with a grid size of FACT*GRDSZ. Default: FACT=.6
GRDSZ	the size of each grid element to be plotted in inches. Default: GRDSZ=0.5.

LP	the number of grid elements in the x direction. The axis is aligned with the width of CALCOMP paper. Default: LP=12.
MP	the number of grid elements in the y direction. The y axis is aligned with the length of the CALCOMP paper. Default: MP=24.
DL	the grid element size in meters. Default: DL=30000.
ISKIP	specifies how many of the plots are to be initially skipped. If first plot is to be drawn then ISKIP should be set to zero. Default: ISKIP=0.
INC	specifies the number of plots to be skipped between drawings. Default: INC=0.
VECTOR	logical variable. TRUE indicates velocity vectors will be drawn as vectors whose length will be proportional to the speed. FALSE indicates an arrow of constant length will be drawn with feathers on the tail. The speed is found by multiplying the number of feathers by SPDSCL, e.g. if 3 1/2 feathers appear then the speed at that grid is 3 1/2 times SPDSCL. Default: VECTOR=.FALSE.
SCALE	logical variable which indicates whether automatic scaling is to be used when VECTOR=.TRUE. If SCALE=.TRUE. each plot will be automatically scaled so that the largest velocity vector will be equal to GRDSZ. Note that the scale will not necessarily be constant for each plot. If SCALE=.FALSE. scaling will be based on the value of SPDMAX. Default: SCALE=.FALSE.
SPDMAX	only used if SCALE=.FALSE. SPDSCL is the length of one feather on the velocity arrow if VECTOR=FALSE. If VECTOR=TRUE, SPDMAX should be set to approximately the maximum velocity to be expected. Default: SPDMAX=0.1 m s^{-1}
ANGT	the angle from the x-axis to true north in degrees ccw from the x-axis.
NOPLTS	the number of plan views to be drawn. Default: NOPLTS=1.
PRT	print switch. TRUE indicates tables of the speed and direction arrays will be created. Default: PRT=.FALSE.
DIST	the distance in inches between two consecutive plan view

plots. Default: DIST=4.0.

Dimensioning

DIMENSION SPEED(LP*MP),DIREC(LP*MP)

Disk Files The program reads one disk file from file #13. It is an unformated, variable spanned file containing various parameters written by program PRTVEL. The file definitions for the IBM 370 are:

FIDEF 13 DISK PRTVEL DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM

Errors

No error statements are generated by PLOTVEL in this version.

H.6 PROGRAM WIND

The program calculates and writes a spatially constant wind field to a disk file. Wu's (1980) relationship is used to transform the user specified wind speed to wind friction velocity squared needed by GAL and written on the disk file.

Inputs

First Record

The first record should contain the title. This can be any sequence of up to

80 alphanumeric characters.

Namelist &MISC

Figure H.2 shows a sample ramp function used to vary the wind in time.

ITO — the time in seconds at which the wind reaches its full magnitude, WTO. In Figure H.2 ITO=2*IDTT. Default: ITO=72000.

WTHETA — the angle of the wind referenced to the positive x axis. The wind vector is described in terms of a Cartesian vector with the angle measured as positive in a counter-clockwise direction and the origin of the vector at the axis origin. Default: WTHETA=0.

WTO — the full magnitude of the wind in m s^{-1}. In the sample shown below wind reaches full magnitude after 2*IDTT seconds. Default: WTO=10.

IDTT — the time increment between changes in the wind magnitude. Default: IDTT=7200.

MAXT — the last time step in seconds. Default: MAXT=72000.

LP — the number of grid elements in the x direction, l'. Default: LP=12.

MP — the number of grid elements in the y direction, m'. Default: MP=24.

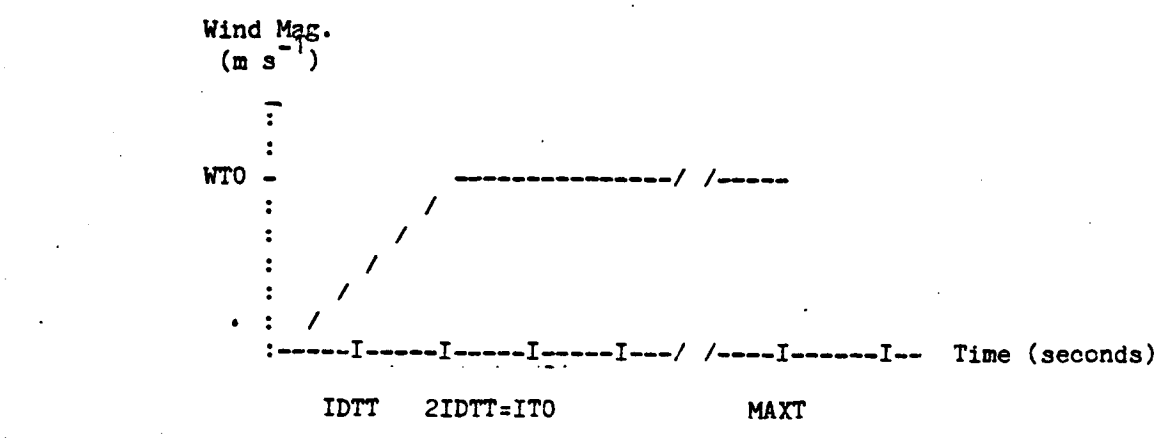


Figure H.2: Sample Wind Function

Dimensioning

DIMENSION UA2(LP*MP),VA2(LP*MP),TIT(20)

Disk Files

The program writes one disk file on device #4. It is an unformated, variable spanned file containing: the time in seconds, and the friction velocity squared in the x and y-direction. The file definition for the IBM 370 is:

FIDEF DISK WIND DISK mode(LRECL 796 BLOCK 800 RECFM VS PERM

Errors

One error can occur if the variable ITO is specified greater than MAXT, in which case an appropriate message is printed and execution immediately terminates.

H.7 SPAT

This program calculates the wind for each element of a grid system with LP by MP elements. The wind can change in time and space. A time series of concurrent measured winds are specified by the user at N stations. The program multiplies these measured winds by amplification factors (SYSW(I), I=1,N), thus giving the user a simple method to correct bias in the measurements. Amplification factors must be specified by the user. The program spatially interpolates between the concurrent wind measurements to

derive the wind field at each grid element. Measured winds are assumed to be at 1 hour intervals.

The interpolation scheme is a simple linear method which calculates the wind at a given grid point, (1,m), by taking the weighted contribution of each station. The weighting for a given station is based on the normalized distance from the station to (1,m). Stated mathematically, the wind component at grid (1,m) for time T is calculated as:

$$U(1,m) = W_1 U_1 + W_2 U_2 + \ldots . W_N U_N$$

where: U(1,m) is the wind component at grid (1,m) for time T, W_n is the weighting factor at station n for time T and U_n is the wind component at station n for time T. Note that U_n is the corrected component, that is $U_n = U'_n SYSW_n$ where U'_n is the observed wind component for time T at station n and $SYSW_n$ is the amplification factor for station n.

The weighting factor, W_n, is given by the expression:

$$W_n = W_1 R_1 / R_n$$

where R_n is the distance between (1,m) and station n. The weighting factor at station 1 is given by:

$$W_1 = 1/(1 + R_1/R_2 + R_1/R_3 + \ldots . R_1/R_N)$$

For illustration consider the following example. Let N=2, $SYSW_1$=1.1, $SYSW_2$=1.2, and assume the grid (1,m) lies a distance 3/8 R from station 1 and 5/8 R from station 2. It follows from the above equations that:

$$W_1 = 1.0/(1 + 3/5) = 5/8$$

$W_2 = 3/5 \times 5/8 = 3/8$

$U(1,m) = 5/8 \times 1.1 \times U_1 + 3/8 \times 1.2 \times U_2$

The program writes a disk file containing wind friction velocity squared at each grid point. The file is compatible with GAL. Wu's (1980) relationship is used to convert wind speed to friction velocity squared.

Inputs

LP	the number of grid elements in the x-direction.
MP	the number of grid elements in the y-direction.
N	the number of stations at which measurements will be specified. Restriction: N.LT.17.
IX(n)	the x grid coordinate of station n.
IY(n)	the y grid coordinate of station n.
SYSW(n)	the amplification factor at station n.

The namelist should be followed immediately by a record containing a title consisting of no more than 80 alpha characters describing the input wind data. The title record is immediately followed by records containing the observed winds at N stations organized in increasing time as follows:

$$U'_{1,T}, V'_{1,T}, U'_{2,T}, V'_{2,T}, \ldots V'_{N,T}$$
$$U'_{1,T+1}, \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots V'_{N,T+1}$$
$$\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$$

where all variables are as previously defined. The units of T are hours and the units of wind speed are in m s^{-1}. The program will continue reading and processing observed wind records until it runs out of data, at which time it will automatically stop. Input format for each record is 16I5.

Dimensioning

```
COMMON/ONE/LP,MP,LPMP,IX(N),IY(N),N,SYSW(N),INT
COMMON/TWO/IUM(N),IVM(N),UM(N),VM(N),W(N,LP*MP)
DIMENSION U(LP*MP),V(LP*MP)
DIMENSION WT(N)
DIMENSION R(N),WT(N)
DIMENSION U(LP*MP),V(LP*MP),USTAR(LP*MP),VSTAR(LP*MP)
```

Errors

The program checks to make sure that the friction velocity squared is less than 0.01 m^2 s^{-2} which corresponds to a wind speed of about 60 m s^{-1}. If the wind is greater than this than the program prints the grid element and time step and terminates execution.

Disk Files

The program writes one variable spanned disk file on unit #4. Each block contains the x and y components of the wind friction velocity squared for LP*MP elements. The file definition on the IBM 370 is:

```
FIDEF 4 DISK WIND DISK mode(LRECL 795 BLOCK 800 RECFM VS PERM
```

H.8 DENSTAT

This is a utility program which reads density data from the standard NODC format for STD data. DENSTAT's primary use is in developing input for DENSITY. The user must specify the starting date and ending date in which he

is interested. The program will then calculate the first order statistics summarizing sigma-t at four water levels for one-half degree squares on the West Florida Shelf.

Input

Namelist &MISC

ISTIME — the starting time consisting of 4 digits. The first two represent the month and the last two, the day.

IETIME — the termination time consisting of 4 digits (same format as ISTIME).

Dimensioning

no dimension changes should be needed when program is applied for what it was designed.

Errors

The program checks to see that ISTIME-IETIME.GT.1 month. If not an error statement is written and execution is terminated.

Disk Files

The reads data from an NODC formated tape. The JCL needed to run the program on the IBM 370 are:

ERASE DLOG DATA G /MOUNT TAPE US0439 ON TAP1 PASS NECE RING OUT FI 13 TAP1 (LRECL 80 BLOCK 3200 RECFM FB ASCII PERM FI 23 DISK DLOG DATA G (LRECL 80 BLOCK 80 RECFM F PERM TAPE REW TAPE FSF 1

H.9 EXAMPLE

Sample input/output (I/O) are given for each of the eight programs. The I/O sequence for WIND, VISCOUS, DENSITY, CIRC and PRTVEL are for production Run 21-5 i.e. summer currents including wind, density and Loop Current forcing. I/O for DENSTAT are for summer conditions and output is summarized in Table 6.1.1. The sample run for SPAT generates the wind field for February-March 1978 (FSU data set).

Program WIND

Sample Input

```
WIND=4.0 M/S AT 120 DEGREES, RAMP TO 20 HRS
&MISC ITO=7200C,WTHETA=120.,WTO=4.0,IDTT=7200,MAXT=72000,LP=12,MP=24,&END
```

Sample Output

```
TIME=,      2.000          SPEED=     0.4000
TIME=,      4.000          SPEED=     0.8000
TIME=,      6.000          SPEED=      1.200
TIME=,      8.000          SPEED=      1.600
TIME=,      10.00          SPEED=      2.000
TIME=,      12.00          SPEED=      2.400
TIME=,      14.00          SPEED=      2.800
TIME=,      16.00          SPEED=      3.200
TIME=,      18.00          SPEED=      3.600
TIME=,      20.00          SPEED=      4.000
TIME=,    0.1000E 33       SPEED=      4.000
```

Program VISCOUS

Sample Input

```
 WFSCM - CB=.00025, BETA PLANE,KP=4,CK=0.,.2,.4,1.,NVM=.03H/200,NVL=.003H/200
  &MISC  DL=30000,JP=3,KP=4,LP=12,MP=24,NS=20,CONNV=.FALSE.,NUCHG=1,&END
  &CORIO  FXY=.TRUE.,PHI=23.5,THETA=65.,&END
  &DEPTH H= 200,200,150,140,90,60,6*0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,60,50,30,25,15,10,0,
  200,200,150,140,90,7*0,
  &END
  &FRICT   CB=288*0.00025,&END
  &EDDYL NODIM=.TRUE.,
  CK= 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
       0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
  0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
       0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
  0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
       0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
  0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
       0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
  0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
       0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
  0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
       0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
  0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
```

```
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
 0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
          0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,0,.2,.4,1.,
    &END
&EDDYVIS  EDDYS=.FALSE.,EDDYT=.FALSE.,ETIME=1.E32,REYNO=12.,WA2=.005,
 EDDYV= .03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
```

```
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
```

```
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015 ,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
```

```
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
.03,.03,.003,.003,.03,.03,.003,.003,
.023,.023,.0023,.0023,.023,.023,.0023,.0023,.015,.015,.0015,.0015,
.007,.007,.0007,.0007,.007,.007,.0007,.0007,
.003,.003,.0003,.0003,.003,.003,.0003,.0003,.0015,.0015,.00015,.00015,
.0015,.0015,.00015,.00015,.0015,.0015,.00015,.00015,
&END
```

GAL:VERSION 7 - JULY 4, 1982 - REV. A - PROGRAM VISCOUS

WFSCM - CB=.00025, BETA PLANE,KP=4,CK=0.,.2,.4,1.,NVM=.03H/200,NVL=.003H/200

JP=3 KP=4 LP=12 MP=24 DL= 0.300E 05

LATITUDE OF GRID ORIGIN=24. DEG

BETA-PLANE APPROXIMATION USED - ANGLE BETWEEN X-AXIS AND TRUE NORTH= 65.

STILL WATER DEPTHS IN METERS FOLLOW:

	X DIRECTION									
-Y-	1	2	3	4	5	6	7	8	9	10
24	200.	200	150.	140.	90.0	0.0	0.0	0.0	0.0	0.0
23	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
22	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
21	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
20	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
19	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
18	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
17	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
16	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
15	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
14	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
13	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
12	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
11	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
10	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
9	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
8	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
7	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
6	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0

5	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
4	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
3	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
2	200.	200.	150.	140.	90.0	60.0	50.0	30.0	25.0	15.0
1	200.	200.	150.	140.	90.0	60.0	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

-Y-	11	12	13	14	X DIRECTION 15	16	17	18	19	20
24	0.0	0.0								
23	10.0	0.0								
22	10.0	0.0								
21	10.0	0.0								
20	10.0	0.0								
19	10.0	0.0								
18	10.0	0.0								
17	10.0	0.0								
16	10.0	0.0								
15	10.0	0.0								
14	10.0	0.0								
13	10.0	0.0								
12	10.0	0.0								
11	10.0	0.0								
10	10.0	0.0								
9	10.0	0.0								
8	10.0	0.0								
7	10.0	0.0								
6	10.0	0.0								
5	10.0	0.0								
4	10.0	0.0								
3	10.0	0.0								
2	10.0	0.0								
1	0.0	0.0								

THE LAYER DEPTHS FOR THE EDDY VISCOSITY FOLLOW (NONDIMENSIONAL):

					X DIRECTION					
-Y-	1	2	3	4	5	6	7	8	9	10
24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
24	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
24	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
23	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
23	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
22	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
22	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
21	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
21	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
20	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
20	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
19	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
19	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
18	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
18	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
17	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
17	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
16	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400

16	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
15	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
14	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
13	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
13	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
12	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
11	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
11	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
10	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
10	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
9	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
8	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
7	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200

6	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
5	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
4	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
3	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
2	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
1	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400	0.400
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
---	--	--	--	--	--	--	--	--	--	--
24	0.0	0.0								
24	0.200	0.200								
24	0.400	0.400								
24	1.00	1.00								

23	0.0	0.0
23	0.200	0.200
23	0.400	0.400
23	1.00	1.00
22	0.0	0.0
22	0.200	0.200
22	0.400	0.400
22	1.00	1.00
21	0.0	0.0
21	0.200	0.200
21	0.400	0.400
21	1.00	1.00
20	0.0	0.0
20	0.200	0.200
20	0.400	0.400
20	1.00	1.00
19	0.0	0.0
19	0.200	0.200
19	0.400	0.400
19	1.00	1.00
18	0.0	0.0
18	0.200	0.200
18	0.400	0.400
18	1.00	1.00
17	0.0	0.0
17	0.200	0.200
17	0.400	0.400
17	1.00	1.00
16	0.0	0.0
16	0.200	0.200
16	0.400	0.400
16	1.00	1.00
15	0.0	0.0
15	0.200	0.200
15	0.400	0.400
15	1.00	1.00
14	0.0	0.0
14	0.200	0.200
14	0.400	0.400

14	1.00	1.00
13	0.0	0.0
13	0.200	0.200
13	0.400	0.400
13	1.00	1.00
12	0.0	0.0
12	0.200	0.200
12	0.400	0.400
12	1.00	1.00
11	0.0	0.0
11	0.200	0.200
11	0.400	0.400
11	1.00	1.00
10	0.0	0.0
10	0.200	0.200
10	0.400	0.400
10	1.00	1.00
9	0.0	0.0
9	0.200	0.200
9	0.400	0.400
9	1.00	1.00
8	0.0	0.0
8	0.200	0.200
8	0.400	0.400
8	1.00	1.00
7	0.0	0.0
7	0.200	0.200
7	0.400	0.400
7	1.00	1.00
6	0.0	0.0
6	0.200	0.200
6	0.400	0.400
6	1.00	1.00
5	0.0	0.0
5	0.200	0.200
5	0.400	0.400
5	1.00	1.00
4	0.0	0.0
4	0.200	0.200

4	0.400	0.400
4	1.00	1.00
3	0.0	0.0
3	0.200	0.200
3	0.400	0.400
3	1.00	1.00
2	0.0	0.0
2	0.200	0.200
2	0.400	0.400
2	1.00	1.00
1	0.0	0.0
1	0.200	0.200
1	0.400	0.400
1	1.00	1.00

FRICTION COEFFICIENTS (M/SEC):

					X DIRECTION					
-Y-	1	2	3	4	5	6	7	8	9	10
24	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
23	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
22	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
21	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
20	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
19	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
18	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
17	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
16	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
15	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
14	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
13	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
12	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
11	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03

10	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
9	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
8	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
7	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
6	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
5	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
4	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
3	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
2	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03
1	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03	0.250E-03

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.250E-03	0.250E-03								
23	0.250E-03	0.250E-03								
22	0.250E-03	0.250E-03								
21	0.250E-03	0.250E-03								
20	0.250E-03	0.250E-03								
19	0.250E-03	0.250E-03								
18	0.250E-03	0.250E-03								
17	0.250E-03	0.250E-03								
16	0.250E-03	0.250E-03								
15	0.250E-03	0.250E-03								
14	0.250E-03	0.250E-03								
13	0.250E-03	0.250E-03								
12	0.250E-03	0.250E-03								
11	0.250E-03	0.250E-03								
10	0.250E-03	0.250E-03								
9	0.250E-03	0.250E-03								
8	0.250E-03	0.250E-03								
7	0.250E-03	0.250E-03								
6	0.250E-03	0.250E-03								
5	0.250E-03	0.250E-03								
4	0.250E-03	0.250E-03								

3	0.250E-03	0.250E-03
2	0.250E-03	0.250E-03
1	0.250E-03	0.250E-03

* *

* *

/ TIME = 6****** HOURS /

NV (SQ MTS/SEC):

	X DIRECTION									
-Y-	1	2	3	4	5	6	7	8	9	10
24	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
24	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
24	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
24	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
23	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
23	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
23	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
23	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
22	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
22	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
22	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
22	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
21	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
21	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
21	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03

21	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
20	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
20	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
20	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
20	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
19	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
19	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
19	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
19	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
18	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
18	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
18	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
18	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
17	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
17	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
17	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
17	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
16	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
16	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
16	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
16	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
15	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
15	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
15	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
15	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
14	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
14	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
14	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
14	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
13	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
13	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
13	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
13	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
12	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02

12	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
12	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
12	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
11	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
11	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
11	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
11	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
10	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
10	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
10	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
10	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
9	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
9	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
9	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
9	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
8	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
8	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
8	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
8	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
7	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
7	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
7	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
7	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
6	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
6	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
6	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
6	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
5	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
5	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
5	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
5	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
4	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
4	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
4	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
4	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
3	0.300 E-0E-03	0.300E-03	0.150E-03							
2	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
2	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
2	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03

2	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
1	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
1	0.300E-01	0.300E-01	0.230E-01	0.230E-01	0.150E-01	0.700E-02	0.700E-02	0.300E-02	0.300E-02	0.150E-02
1	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03
1	0.300E-02	0.300E-02	0.230E-02	0.230E-02	0.150E-02	0.700E-03	0.700E-03	0.300E-03	0.300E-03	0.150E-03

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.150E-02	0.150E-02								
24	0.150E-02	0.150E-02								
24	0.150E-03	0.150E-03								
24	0.150E-03	0.150E-03								
23	0.150E-02	0.150E-02								
23	0.150E-02	0.150E-02								
23	0.150E-03	0.150E-03								
23	0.150E-03	0.150E-03								
22	0.150E-02	0.150E-02								
22	0.150E-02	0.150E-02								
22	0.150E-03	0.150E-03								
22	0.150E-03	0.150E-03								
21	0.150E-02	0.150E-02								
21	0.150E-02	0.150E-02								
21	0.150E-03	0.150E-03								
21	0.150E-03	0.150E-03								
20	0.150E-02	0.150E-02								
20	0.150E-02	0.150E-02								
20	0.150E-03	0.150E-03								
20	0.150E-03	0.150E-03								
19	0.150E-02	0.150E-02								

19	0.150E-02	0.150E-02
19	0.150E-03	0.150E-03
19	0.150E-03	0.150E-03
18	0.150E-02	0.150E-02
18	0.150E-02	0.150E-02
18	0.150E-03	0.150E-03
18	0.150E-03	0.150E-03
17	0.150E-02	0.150E-02
17	0.150E-02	0.150E-02
17	0.150E-03	0.150E-03
17	0.150E-03	0.150E-03
16	0.150E-02	0.150E-02
16	0.150E-02	0.150E-02
16	0.150E-03	0.150E-03
16	0.150E-03	0.150E-03
15	0.150E-02	0.150E-02
15	0.150E-02	0.150E-02
15	0.150E-03	0.150E-03
15	0.150E-03	0.150E-03
14	0.150E-02	0.150E-02
14	0.150E-02	0.150E-02
14	0.150E-03	0.150E-03
14	0.150E-03	0.150E-03
13	0.150E-02	0.150E-02
13	0.150E-02	0.150E-02
13	0.150E-03	0.150E-03
13	0.150E-03	0.150E-03
12	0.150E-02	0.150E-02
12	0.150E-02	0.150E-02
12	0.150E-03	0.150E-03
12	0.150E-03	0.150E-03
11	0.150E-02	0.150E-02
11	0.150E-02	0.150E-02
11	0.150E-03	0.150E-03
11	0.150E-03	0.150E-03
10	0.150E-02	0.150E-02
10	0.150E-02	0.150E-02
10	0.150E-03	0.150E-03
10	0.150E-03	0.150E-03

9	0.150E-02	0.150E-02
9	0.150E-02	0.150E-02
9	0.150E-03	0.150E-03
9	0.150E-03	0.150E-03
8	0.150E-02	0.150E-02
8	0.150E-02	0.150E-02
8	0.150E-03	0.150E-03
8	0.150E-03	0.150E-03
7	0.150E-02	0.150E-02
7	0.150E-02	0.150E-02
7	0.150E-03	0.150E-03
7	0.150E-03	0.150E-03
6	0.150E-02	0.150E-02
6	0.150E-02	0.150E-02
6	0.150E-03	0.150E-03
6	0.150E-03	0.150E-03
5	0.150E-02	0.150E-02
5	0.150E-02	0.150E-02
5	0.150E-03	0.150E-03
5	0.150E-03	0.150E-03
4	0.150E-02	0.150E-02
4	0.150E-02	0.150E-02
4	0.150E-03	0.150E-03
4	0.150E-03	0.150E-03
3	0.150E-02	0.150E-02
3	0.150E-02	0.150E-02
3	0.150E-03	0.150E-03
3	0.150E-03	0.150E-03
2	0.150E-02	0.150E-02
2	0.150E-02	0.150E-02
2	0.150E-03	0.150E-03
2	0.150E-03	0.150E-03
1	0.150E-02	0.150E-02
1	0.150E-02	0.150E-02
1	0.150E-03	0.150E-03
1	0.150E-03	0.150E-03

A(JLM):

	X DIRECTION									
-Y-	1	2	3	4	5	6	7	8	9	10
24	1.48	1.48	1.48	1.47	1.47	0.0	0.0	0.0	0.0	0.0
24	4.45	4.45	4.45	4.43	4.43	0.0	0.0	0.0	0.0	0.0
24	7.43	7.43	7.43	7.40	7.40	0.0	0.0	0.0	0.0	0.0
23	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
23	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
23	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
22	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
22	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
22	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
21	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
21	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
21	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
20	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
20	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
20	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
19	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
19	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
19	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
18	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
18	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
18	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
17	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
17	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
17	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
16	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
16	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
16	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
15	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
15	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
15	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
14	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51

14	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
14	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
13	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
13	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
13	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
12	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
12	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
12	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
11	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
11	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
11	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
10	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
10	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
10	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
9	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
9	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
9	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
8	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
8	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
8	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
7	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
7	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
7	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
6	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
6	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
6	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
5	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
5	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
5	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
4	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
4	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
4	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
3	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
3	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53

3	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
2	1.48	1.48	1.48	1.47	1.47	1.50	1.49	1.51	1.50	1.51
2	4.45	4.45	4.45	4.43	4.43	4.51	4.47	4.53	4.50	4.53
2	7.43	7.43	7.43	7.40	7.40	7.52	7.46	7.56	7.51	7.56
1	1.48	1.48	1.48	1.47	1.47	1.50	0.0	0.0	0.0	0.0
1	4.45	4.45	4.45	4.43	4.43	4.51	0.0	0.0	0.0	0.0
1	7.43	7.43	7.43	7.40	7.40	7.52	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.0	0.0								
24	0.0	0.0								
24	0.0	0.0								
23	1.48	0.0								
23	4.45	0.0								
23	7.43	0.0								
22	1.48	0.0								
22	4.45	0.0								
22	7.43	0.0								
21	1.48	0.0								
21	4.45	0.0								
21	7.43	0.0								
20	1.48	0.0								
20	4.45	0.0								
20	7.43	0.0								
19	1.48	0.0								
19	4.45	0.0								
19	7.43	0.0								
18	1.48	0.0								
18	4.45	0.0								
18	7.43	0.0								
17	1.48	0.0								
17	4.45	0.0								
17	7.43	0.0								

16	1.48	0.0
16	4.45	0.0
16	7.43	0.0
15	1.48	0.0
15	4.45	0.0
15	7.43	0.0
14	1.48	0.0
14	4.45	0.0
14	7.43	0.0
13	1.48	0.0
13	4.45	0.0
13	7.43	0.0
12	1.48	0.0
12	4.45	0.0
12	7.43	0.0
11	1.48	0.0
11	4.45	0.0
11	7.43	0.0
10	1.48	0.0
10	4.45	0.0
10	7.43	0.0
9	1.48	0.0
9	4.45	0.0
9	7.43	0.0
8	1.48	0.0
8	4.45	0.0
8	7.43	0.0
7	1.48	0.0
7	4.45	0.0
7	7.43	0.0
6	1.48	0.0
6	4.45	0.0
6	7.43	0.0
5	1.48	0.0
5	4.45	0.0
5	7.43	0.0
4	1.48	0.0
4	4.45	0.0
4	7.43	0.0

3	1.48	0.0
3	4.45	0.0
3	7.43	0.0
2	1.48	0.0
2	4.45	0.0
2	7.43	0.0
1	0.0	0.0
1	0.0	0.0
1	0.0	0.0

F(L,M) - CORIOLIS PARAMETER

	X DIRECTION									
-Y-	1	2	3	4	5	6	7	8	9	10
24	0.708E-04	0.710E-04	0.713E-04	0.716E-04	0.719E-04	0.721E-04	0.724E-04	0.726E-04	0.729E-04	0.731E-04
23	0.702E-04	0.705E-04	0.708E-04	0.710E-04	0.713E-04	0.716E-04	0.719E-04	0.720E-04	0.723E-04	0.725E-04
22	0.697E-04	0.700E-04	0.702E-04	0.705E-04	0.707E-04	0.710E-04	0.713E-04	0.715E-04	0.718E-04	0.720E-04
21	0.692E-04	0.694E-04	0.697E-04	0.700E-04	0.702E-04	0.704E-04	0.707E-04	0.710E-04	0.712E-04	0.714E-04
20	0.686E-04	0.689E-04	0.691E-04	0.694E-04	0.697E-04	0.700E-04	0.702E-04	0.704E-04	0.706E-04	0.709E-04
19	0.681E-04	0.683E-04	0.686E-04	0.688E-04	0.691E-04	0.694E-04	0.696E-04	0.699E-04	0.701E-04	0.704E-04
18	0.675E-04	0.678E-04	0.680E-04	0.683E-04	0.686E-04	0.688E-04	0.691E-04	0.693E-04	0.695E-04	0.698E-04
17	0.670E-04	0.672E-04	0.675E-04	0.678E-04	0.680E-04	0.683E-04	0.685E-04	0.688E-04	0.690E-04	0.693E-04
16	0.664E-04	0.667E-04	0.669E-04	0.672E-04	0.675E-04	0.677E-04	0.680E-04	0.682E-04	0.679E-04	0.687E-04
15	0.659E-04	0.661E-04	0.664E-04	0.667E-04	0.669E-04	0.672E-04	0.674E-04	0.677E-04	0.674E-04	0.676E-04
14	0.653E-04	0.656E-04	0.658E-04	0.661E-04	0.664E-04	0.666E-04	0.669E-04	0.666E-04	0.669E-04	0.672E-04
13	0.647E-04	0.650E-04	0.653E-04	0.655E-04	0.658E-04	0.660E-04	0.663E-04	0.661E-04	0.664E-04	0.667E-04
12	0.642E-04	0.644E-04	0.647E-04	0.650E-04	0.653E-04	0.655E-04	0.653E-04	0.656E-04	0.659E-04	0.663E-04
11	0.636E-04	0.639E-04	0.641E-04	0.644E-04	0.647E-04	0.649E-04	0.648E-04	0.651E-04	0.655E-04	0.658E-04
10	0.631E-04	0.633E-04	0.636E-04	0.638E-04	0.641E-04	0.640E-04	0.643E-04	0.646E-04	0.650E-04	0.654E-04
9	0.625E-04	0.628E-04	0.630E-04	0.633E-04	0.635E-04	0.635E-04	0.638E-04	0.642E-04	0.646E-04	0.612E-04
8	0.619E-04	0.622E-04	0.625E-04	0.627E-04	0.627E-04	0.630E-04	0.634E-04	0.638E-04	0.609E-04	0.610E-04
7	0.614E-04	0.616E-04	0.619E-04	0.622E-04	0.622E-04	0.626E-04	0.630E-04	0.604E-04	0.606E-04	0.603E-04
6	0.608E-04	0.611E-04	0.614E-04	0.614E-04	0.617E-04	0.621E-04	0.601E-04	0.603E-04	0.605E-04	0.607E-04
5	0.603E-04	0.605E-04	0.608E-04	0.609E-04	0.613E-04	0.597E-04	0.599E-04	0.601E-04	0.604E-04	0.606E-04
4	0.597E-04	0.600E-04	0.605E-04	0.605E-04	0.593E-04	0.595E-04	0.598E-04	0.600E-04	0.603E-04	0.605E-04
3	0.591E-04	0.594E-04	0.597E-04	0.589E-04	0.592E-04	0.594E-04	0.597E-04	0.598E-04	0.602E-04	0.604E-04
2	0.586E-04	0.588E-04	0.585E-04	0.588E-04	0.591E-04	0.593E-04	0.596E-04	0.599E-04	0.601E-04	0.604E-04
1	0.580E-04	0.583E-04	0.585E-04	0.588E-04	0.591E-04	0.593E-04	0.596E-04	0.598E-04	0.601E-04	0.604E-04

CONTINUED FROM PREVIOUS PAGE

					X DIRECTION					
-Y-	11	12	13	14	15	16	17	18	19	20
24	0.733E-04	0.736E-04								
23	0.728E-04	0.730E-04								
22	0.722E-04	0.717E-04								
21	0.717E-04	0.712E-04								
20	0.704E-04	0.707E-04								
19	0.699E-04	0.702E-04								
18	0.694E-04	0.697E-04								
17	0.689E-04	0.692E-04								
16	0.684E-04	0.688E-04								
15	0.680E-04	0.683E-04								
14	0.675E-04	0.679E-04								
13	0.671E-04	0.674E-04								
12	0.666E-04	0.670E-04								
11	0.662E-04	0.619E-04								
10	0.615E-04	0.617E-04								
9	0.614E-04	0.616E-04								
8	0.612E-04	0.614E-04								
7	0.611E-04	0.613E-04								
6	0.609E-04	0.612E-04								
5	0.608E-04	0.611E-04								
4	0.607E-04	0.610E-04								
3	0.607E-04	0.609E-04								
2	0.606E-04	0.609E-04								
1	0.606E-04	0.609E-04								

EOF:

Program DENSITY

Sample Input

```
WFSCM IFP=2,4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,12*1.025
 &MISC NUCHG=1,CALDEN=F,&END
 &LAYER IFP=2,HP=10.,300.,DTIME=1E30,&END
 &RHO RHOWP= 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235.1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231.1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,1.021,
 288*1.025,
 &END
```

Sample Output

* *

CAL:VERSION 7 - JUNE 16 1982 - REV. A - DENSITY

WFSCM IPP=2,4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,12*1.025,1.02

VISCOSITY TITLE: WFSCM - CB=.00025, BETA PLANE,KP=4,CK=0.,.2,.4,1.,NVM=.03H/200,NVL=.003H/200

JP=3 LP=12 MP=24

STILL WATER DEPTHS IN METERS FOLLOW:

					X DIRECTION					
-Y-	1	2	3	4	5	6	7	8	9	10
24	200.0000	200.0000	150.0000	140.0000	90.0000	0.0	0.0	0.0	0.0	0.0
23	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
22	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
21	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
20	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
19	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
18	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
17	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
16	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
15	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
14	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
13	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
12	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
11	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
10	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
9	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
8	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
7	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
6	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
5	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000

4	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
3	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
2	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	50.0000	30.0000	25.0000	15.0000
1	200.0000	200.0000	150.0000	140.0000	90.0000	60.0000	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.0	0.0								
23	10.0000	0.0								
22	10.0000	0.0								
21	10.0000	0.0								
20	10.0000	0.0								
19	10.0000	0.0								
18	10.0000	0.0								
17	10.0000	0.0								
16	10.0000	0.0								
15	10.0000	0.0								
14	10.0000	0.0								
13	10.0000	0.0								
12	10.0000	0.0								
11	10.0000	0.0								
10	10.0000	0.0								
9	10.0000	0.0								
8	10.0000	0.0								
7	10.0000	0.0								
6	10.0000	0.0								
5	10.0000	0.0								
4	10.0000	0.0								
3	10.0000	0.0								
2	10.0000	0.0								
1	0.0	0.0								

/ TIME 1****** HOURS /

THE LAYER DEPTHS FOLLOW(METERS):
10.0 300.

A(JLM):

X DIRECTION

-Y-	1	2	3	4	5	6	7	8	9	10
24	1.4824	1.4824	1.4806	1.4746	1.4733	0.0	0.0	0.0	0.0	0.0
24	4.4513	4.4513	4.4461	4.4291	4.4254	0.0	0.0	0.0	0.0	0.0
24	7.4343	7.4343	7.4265	7.4012	7.3958	0.0	0.0	0.0	0.0	0.0
23	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
23	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
23	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
22	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
22	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
22	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
21	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
21	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
21	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
20	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
20	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
20	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
19	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
19	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
19	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
18	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
18	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
18	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
17	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
17	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
17	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
16	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
16	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333

16	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
15	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
15	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
15	7.4343	7.4343	7.4265	7.4012	7.3958	7.5160	7.4583	7.5603	7.5080	7.5603
14	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
14	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
14	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
13	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
13	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
13	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
12	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
12	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
12	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
11	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
11	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
11	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
10	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
10	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
10	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
9	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
9	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
9	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
8	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
8	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
8	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
7	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
7	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
7	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
6	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
6	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
6	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
5	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
5	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
5	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603

4	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
4	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
4	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
3	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
3	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
3	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
2	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	1.4879	1.5105	1.4991	1.5105
2	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	4.4672	4.5333	4.4996	4.5333
2	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	7.4583	7.5603	7.5080	7.5603
1	1.4824	1.4824	1.4806	1.4746	1.4733	1.5010	0.0	0.0	0.0	0.0
1	4.4513	4.4513	4.4461	4.4291	4.4254	4.5056	0.0	0.0	0.0	0.0
1	7.4343	7.4343	7.4265	7.4012	7.3958	7.5166	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.0	0.0								
24	0.0	0.0								
24	0.0	0.0								
23	1.4824	0.0								
23	4.4513	0.0								
23	7.4343	0.0								
22	1.4824	0.0								
22	4.4513	0.0								
22	7.4343	0.0								
21	1.4824	0.0								
21	4.4513	0.0								
21	7.4343	0.0								
20	1.4824	0.0								
20	4.4513	0.0								
20	7.4343	0.0								
19	1.4824	0.0								
19	4.4513	0.0								
19	7.4343	0.0								
18	1.4824	0.0								

Sample DENSE Output

18	4.4513	0.0
18	7.4343	0.0
17	1.4824	0.0
17	4.4513	0.0
17	7.4343	0.0
16	1.4824	0.0
16	4.4513	0.0
16	7.4343	0.0
15	1.4824	0.0
15	4.4513	0.0
15	7.4343	0.0
14	1.4824	0.0
14	4.4513	0.0
14	7.4343	0.0
13	1.4824	0.0
13	4.4513	0.0
13	7.4343	0.0
12	1.4824	0.0
12	4.4513	0.0
12	7.4343	0.0
11	1.4824	0.0
11	4.4513	0.0
11	7.4343	0.0
10	1.4824	0.0
10	4.4513	0.0
10	7.4343	0.0
9	1.4824	0.0
9	4.4513	0.0
9	7.4343	0.0
8	1.4824	0.0
8	4.4513	0.0
8	7.4343	0.0
7	1.4824	0.0
7	4.4513	0.0
7	7.4343	0.0
6	1.4824	0.0
6	4.4513	0.0
6	7.4343	0.0
5	1.4824	0.0

5	4.4513	0.0
5	7.4343	0.0
4	1.4824	0.0
4	4.4513	0.0
4	7.4343	0.0
3	1.4824	0.0
3	4.4513	0.0
3	7.4343	0.0
2	1.4824	0.0
2	4.4513	0.0
2	7.4343	0.0
1	0.0	0.0
1	0.0	0.0
1	0.0	0.0

THE DENSITY PROFILES FOLLOW(G/CUBIC CM):

X DIRECTION

-Y-	1	2	3	4	5	6	7	8	9	10
24	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
24	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
23	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
23	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
22	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
22	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
21	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
21	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
20	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
20	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
19	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
19	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
18	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
18	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
17	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
17	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
16	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
16	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
15	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220

15	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
14	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
14	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
13	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
13	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
12	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
12	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
11	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
11	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
10	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
10	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
9	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
9	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
8	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
8	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
7	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
7	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
6	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
6	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
5	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
5	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
4	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
4	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
3	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
3	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
2	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
2	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250
1	1.0235	1.0235	1.0235	1.0235	1.0234	1.0233	1.0232	1.0231	1.0230	1.0220
1	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250	1.0250

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	1.0210	1.0210								
24	1.0250	1.0250								
23	1.0210	1.0210								
23	1.0250	1.0250								
22	1.0210	1.0210								
22	1.0250	1.0250								
21	1.0210	1.0210								
21	1.0250	1.0250								
20	1.0210	1.0210								
20	1.0250	1.0250								
19	1.0210	1.0210								
19	1.0250	1.0250								
18	1.0210	1.0210								
18	1.0250	1.0250								
17	1.0210	1.0210								
17	1.0250	1.0250								
16	1.0210	1.0210								
16	1.0250	1.0250								
15	1.0210	1.0210								
15	1.0250	1.0250								
14	1.0210	1.0210								
14	1.0250	1.0250								
13	1.0210	1.0210								
13	1.0250	1.0250								
12	1.0210	1.0210								
12	1.0250	1.0250								
11	1.0210	1.0210								
11	1.0250	1.0250								
10	1.0210	1.0210								
10	1.0250	1.0250								
9	1.0210	1.0210								
9	1.0250	1.0250								
8	1.0210	1.0210								
8	1.0250	1.0250								
7	1.0210	1.0210								
7	1.0250	1.0250								

6	1.0210	1.0210
6	1.0250	1.0250
5	1.0210	1.0210
5	1.0250	1.0250
4	1.0210	1.0210
4	1.0250	1.0250
3	1.0210	1.0210
3	1.0250	1.0250
2	1.0210	1.0210
2	1.0250	1.0250
1	1.0210	1.0210
1	1.0250	1.0250

Program CIRC

Sample Input

```
 WFSCM - CASE 21-5 - SUMMER HORIZ - CIRC 7.A
  &MISC    EDDYH=100000.,DENSE=.TRUE.,&END
  &TIME    TLAST=362000,DTT=450.,ISTPR=360000,
           IDTPR=36000,&END
  &DCS WLNGTH=0,WPERIOD=0,
  IAMP(1,1)=5,IAMP(1,2)=5,IAMP(1,3)=5,IAMP(1,4)=5,IAMP(1,5)=5,IAMP(1,6)=5,
  IAMP(1,7)=5,IAMP(1,8)=5,IAMP(1,9)=5,IAMP(1,10)=5,IAMP(1,11)=5,
  IAMP(1,12)=5,IAMP(1,13)=5,IAMP(1,14)=5,IAMP(1,15)=5,IAMP(1,16)=5,
  IAMP(1,17)=5,IAMP(1,18)=5,IAMP(1,19)=5,IAMP(1,20)=2,IAMP(1,21)=2,
  IAMP(1,22)=2,IAMP(1,23)=2,IAMP(1,24)=2,
  DBC(1,1,1)=-.5,DBC(1,1,2)=-.472,DBC(1,1,3)=-.444,DBC(1,1,4)=-.417,
  DBC(1,1,5)=-.389,DBC(1,1,6)=-.361,DBC(1,1,7)=-.333,DBC(1,1,8)=-.306,
  DBC(1,1,9)=-.278,DBC(1,1,10)=-.25,DBC(1,1,11)=-.222,DBC(1,1,12)=-.194,
  DBC(1,1,13)=-.167,DBC(1,1,14)=-.139,DBC(1,1,15)=-.111,DBC(1,1,16)=-.083,
  DBC(1,1,17)=-.056,DBC(1,1,18)=-.028,DBC(1,1,19)=0.,
  DBC(2,1,1)=-.4,DBC(2,1,2)=-.378,DBC(2,1,3)=-.356,DBC(2,1,4)=-.335,
  DBC(2,1,5)=-.313,DBC(2,1,6)=-.291,DBC(2,1,7)=-.269,DBC(2,1,8)=-.248,
  DBC(2,1,9)=-.225,DBC(2,1,10)=-.204,DBC(2,1,11)=-.182,DBC(2,1,12)=-.160,
  DBC(2,1,13)=-.139,DBC(2,1,14)=-.117,DBC(2,1,15)=-.095,DBC(2,1,16)=-.073,
  DBC(2,1,17)=-.052,DBC(2,1,18)=-.030,DBC(2,1,19)=0.,
  DBC(3,1,1)=-.235,DBC(3,1,2)=-.222,DBC(3,1,3)=-.209,DBC(3,1,4)=-.196,
  DBC(3,1,5)=-.183,DBC(3,1,6)=-.170,DBC(3,1,7)=-.157,DBC(3,1,8)=-.144,
  DBC(3,1,9)=-.131,DBC(3,1,10)=-.118,DBC(3,1,11)=-.104,DBC(3,1,12)=-.091,
  DBC(3,1,13)=-.078,DBC(3,1,14)=-.065,DBC(3,1,15)=-.052,DBC(3,1,16)=-.039,
  DBC(3,1,17)=-.026,DBC(3,1,18)=-.013,DBC(3,1,19)=0.,
  &END
  &INOUT HOTSTR=F,HOTIME=352800,PRTFLO=.FALSE.,&END
```

Sample Output

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

                                GAL - VER. 7.A - JULY 1982  PROGRAM CIRC
                                -------------------------------------

 ---------------------
/ TIME=100.00 HOURS  /
---------------------

                                        SUMMARY OUTPUT FOR GRID( 6,12)

                                      U**2=-0.101E-04     V**2= 0.175E-04 SQ M/SQ S
UNDETERMINED PARAMETERS

                    1           2           3
  C(J)    0.507E-01   0.204E-01   0.236E-02
  D(J)   -0.147E 00   0.334E-02  -0.278E-02

MASS FLUXES-SQ MTS/SEC:
  QX=-0.393E 00       QY=-0.217E 01

SURFACE ELEVATION(M)=-0.711E-01
```

Sample CIRC Output

Program PRTVEL

Sample Input

```
 &OUTOPS NOGRDS=3,LOCATE(1)=(6,19),LOCATE(2)=(9,19),LOCATE(3)=(6,12),
  LAYERS=2,LEVELS=2,PRTETA=T,PRTWND=F,PRTHV=F,PRTDEN=F,&END
   0.  10.  25.  30.  39.  50.
   0.   5.  12.  17.
   0.   5.  12.  60. 100. 140.
   0.   .5   1.
   0.   .5   1.
```

Sample Output

GAL:VERSION 7 - JULY 4, 1982 - REV. A - PROGRAM PRTVEL

WFSCM - CASE 21-5 - SUMMER HORIZ - CIRC 7.A

VISCOUS TITLE: WFSCM - CB=.00025, BETA PLANE,KP=4,CK=0.,.2,.4,1.,NVH=.03H/200,NVL=.003H/200

DENSITY TITLE: WFSCM IPP=2,4*1.0235,1.0234,1.0233,1.0232,1.0231,1.023,1.022,1.021,12*1.025,1.02

WIND TITLE:WIND=4.0 M/S AT 120 DEGREES, RAMP TO 20 HRS

JP=3 KP=4 LP=12 MP=24 EDDYH= 0.100E 06

DT=450.SECS DL=30000.0 METERS LATITUDE= 24.DEG N

1STILL WATER DEPTHS (METERS)

	X DIRECTION									
-Y-	1	2	3	4	5	6	7	8	9	10
24	200.0	200.0	150.0	140.0	90.00	0.0	0.0	0.0	0.0	0.0
23	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
22	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
21	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
20	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
19	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
18	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
17	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
16	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
15	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
14	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
13	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00

12	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
11	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
10	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
9	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
8	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
7	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
6	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
5	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
4	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
3	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
2	200.0	200.0	150.0	140.0	90.00	60.00	50.00	30.00	25.00	15.00
1	200.0	200.0	150.0	140.0	90.00	60.00	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.0	0.0								
23	10.00	0.0								
22	10.00	0.0								
21	10.00	0.0								
20	10.00	0.0								
19	10.00	0.0								
18	10.00	0.0								
17	10.00	0.0								
16	10.00	0.0								
15	10.00	0.0								
14	10.00	0.0								
13	10.00	0.0								
12	10.00	0.0								
11	10.00	0.0								
10	10.00	0.0								
9	10.00	0.0								
8	10.00	0.0								
7	10.00	0.0								
6	10.00	0.0								

5	10.00	0.0
4	10.00	0.0
3	10.00	0.0
2	10.00	0.0
1	0.0	0.0

1FAICTION COEFFICIENTS (M/SEC)

X DIRECTION

-Y-	1	2	3	4	5	6	7	8	9	10
24	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
23	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
22	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
21	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
20	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
19	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
18	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
17	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
16	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
15	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
14	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
13	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
12	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
11	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
10	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
9	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
8	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
7	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
6	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
5	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
4	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
3	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
2	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03
1	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03	0.2500E-03

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.2500E-03	0.2500E-03								
23	0.2500E-03	0.2500E-03								
22	0.2500E-03	0.2500E-03								
21	0.2500E-03	0.2500E-03								
20	0.2500E-03	0.2500E-03								
19	0.2500E-03	0.2500E-03								
18	0.2500E-03	0.2500E-03								
17	0.2500E-03	0.2500E-03								
16	0.2500E-03	0.2500E-03								
15	0.2500E-03	0.2500E-03								
14	0.2500E-03	0.2500E-03								
13	0.2500E-03	0.2500E-03								
12	0.2500E-03	0.2500E-03								
11	0.2500E-03	0.2500E-03								
10	0.2500E-03	0.2500E-03								
9	0.2500E-03	0.2500E-03								
8	0.2500E-03	0.2500E-03								
7	0.2500E-03	0.2500E-03								
6	0.2500E-03	0.2500E-03								
5	0.2500E-03	0.2500E-03								
4	0.2500E-03	0.2500E-03								
3	0.2500E-03	0.2500E-03								
2	0.2500E-03	0.2500E-03								
1	0.2500E-03	0.2500E-03								

ILAYER DEPTHS FOR THE EDDY VISCOSITY (NONDIMENSIONAL):

X DIRECTION

-Y-	1	2	3	4	5	6	7	8	9	10
24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

24	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
24	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
24	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
23	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
23	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
22	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
22	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
21	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
21	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
20	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
20	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
19	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
19	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
18	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
18	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
17	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
17	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
16	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
16	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
15	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
15	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
14	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
14	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
13	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
13	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
12	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
12	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
11	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
11	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
10	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
10	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
9	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
9	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
8	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
8	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
7	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
7	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000

6	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
6	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
5	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
5	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
4	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
4	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
3	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
2	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
1	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
---	--	--	--	--	--	--	--	--	--	--
24	0.0	0.0								
24	0.2000	0.2000								
24	0.4000	0.4000								
24	1.000	1.000								

23	0.0	0.0
23	0.2000	0.2000
23	0.4000	0.4000
23	1.000	1.000
22	0.0	0.0
22	0.2000	0.2000
22	0.4000	0.4000
22	1.000	1.000
21	0.0	0.0
21	0.2000	0.2000
21	0.4000	0.4000
21	1.000	1.000
20	0.0	0.0
20	0.2000	0.2000
20	0.4000	0.4000
20	1.000	1.000
19	0.0	0.0
19	0.2000	0.2000
19	0.4000	0.4000
19	1.000	1.000
18	0.0	0.0
18	0.2000	0.2000
18	0.4000	0.4000
18	1.000	1.000
17	0.0	0.0
17	0.2000	0.2000
17	0.4000	0.4000
17	1.000	1.000
16	0.0	0.0
16	0.2000	0.2000
16	0.4000	0.4000
16	1.000	1.000
15	0.0	0.0
15	0.2000	0.2000
15	0.4000	0.4000
15	1.000	1.000
14	0.0	0.0
14	0.2000	0.2000
14	0.4000	0.4000

14	1.000	1.000
13	0.0	0.0
13	0.2000	0.2000
13	0.4000	0.4000
13	1.000	1.000
12	0.0	0.0
12	0.2000	0.2000
12	0.4000	0.4000
12	1.000	1.000
11	0.0	0.0
11	0.2000	0.2000
11	0.4000	0.4000
11	1.000	1.000
10	0.0	0.0
10	0.2000	0.2000
10	0.4000	0.4000
10	1.000	1.000
9	0.0	0.0
9	0.2000	0.2000
9	0.4000	0.4000
9	1.000	1.000
8	0.0	0.0
8	0.2000	0.2000
8	0.4000	0.4000
8	1.000	1.000
7	0.0	0.0
7	0.2000	0.2000
7	0.4000	0.4000
7	1.000	1.000
6	0.0	0.0
6	0.2000	0.2000
6	0.4000	0.4000
6	1.000	1.000
5	0.0	0.0
5	0.2000	0.2000
5	0.4000	0.4000
5	1.000	1.000
4	0.0	0.0
4	0.2000	0.2000

4	0.4000	0.4000
4	1.000	1.000
3	0.0	0.0
3	0.2000	0.2000
3	0.4000	0.4000
3	1.000	1.000
2	0.0	0.0
2	0.2000	0.2000
2	0.4000	0.4000
2	1.000	1.000
1	0.0	0.0
1	0.2000	0.2000
1	0.4000	0.4000
1	1.000	1.000

1INITIAL AMPLITUDES(MTS)

X DIRECTION

-Y-	1	2	3	4	5	6	7	8	9	10
24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	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
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.0	0.0								
23	0.0	0.0								
22	0.0	0.0								
21	0.0	0.0								
20	0.0	0.0								
19	0.0	0.0								
18	0.0	0.0								
17	0.0	0.0								
16	0.0	0.0								
15	0.0	0.0								
14	0.0	0.0								
13	0.0	0.0								
12	0.0	0.0								
11	0.0	0.0								
10	0.0	0.0								
9	0.0	0.0								
8	0.0	0.0								
7	0.0	0.0								
6	0.0	0.0								
5	0.0	0.0								

4	0.0	0.0
3	0.0	0.0
2	0.0	0.0
1	0.0	0.0

PERIOD OF THE TIDAL OSCILLATION= 0.454E 05SECS

!ELEMENT TYPES

X DIRECTION

-Y-	1	2	3	4	5	6	7	8	9	10
24	2.000	1.000	1.000	1.000	1.000	0.0	0.0	0.0	0.0	0.0
23	2.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
22	2.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
21	2.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
20	2.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
19	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
18	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
17	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
16	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
15	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
14	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
13	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
12	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
11	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
10	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
9	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
8	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
7	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
6	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
5	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
4	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
3	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
2	5.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000
1	5.000	1.000	1.000	1.000	1.000	1.000	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.0	0.0								
23	4.000	0.0								
22	4.000	0.0								
21	4.000	0.0								
20	4.000	0.0								
19	4.000	0.0								
18	4.000	0.0								
17	4.000	0.0								
16	4.000	0.0								
15	4.000	0.0								
14	4.000	0.0								
13	4.000	0.0								
12	4.000	0.0								
11	4.000	0.0								
10	4.000	0.0								
9	4.000	0.0								
8	4.000	0.0								
7	4.000	0.0								
6	4.000	0.0								
5	4.000	0.0								
4	4.000	0.0								
3	4.000	0.0								
2	4.000	0.0								
1	0.0	0.0								

/ TIME =100.00 HOURS /

WATER ELEVATION(MTS) ABOVE STILL WATER

X DIRECTION

-Y-	1	2	3	4	5	6	7	8	9	10
24	0.0	-0.8471E-02	-0.1998E-01	-0.2960E-01	-0.3814E-01	0.0	0.0	0.0	0.0	0.0
23	0.0	-0.8471E-02	-0.1998E-01	-0.2960E-01	-0.3814E-01	-0.4403E-01	-0.4521E-01	-0.3138E-01	-0.2836E-01	-0.1322E-01
22	0.0	-0.7172E-02	-0.1985E-01	-0.3091E-01	-0.4202E-01	-0.6216E-01	-0.6278E-01	-0.6480E-01	-0.5635E-01	-0.4981E-01
21	0.0	-0.5071E-02	-0.1830E-01	-0.3003E-01	-0.4141E-01	-0.5952E-01	-0.6116E-01	-0.6756E-01	-0.5975E-01	-0.5475E-01
20	0.0	-0.1732E-02	-0.1587E-01	-0.2851E-01	-0.4004E-01	-0.5761E-01	-0.5957E-01	-0.6796E-01	-0.6070E-01	-0.5646E-01
19	0.0	0.4147E-02	-0.1387E-01	-0.2781E-01	-0.3943E-01	-0.5655E-01	-0.5873E-01	-0.6767E-01	-0.6078E-01	-0.5701E-01
18	0.0	-0.4430E-0232E-01	-0.6577E-01	-0.6697E-01	-0.7327E-01	-0.6606E-01	-0.6237E-01			
13	0.0	-0.1663E-01	-0.3122E-01	-0.4600E-01	-0.5559E-01	-0.6860E-01	-0.5927E-01	-0.7515E-01	-0.6764E-01	-0.6385E-01
12	0.0	-0.1855E-01	-0.3412E-01	-0.4917E-01	-0.5854E-01	-0.7113E-01	-0.7122E-01	-0.7683E-01	-0.6907E-01	-0.6528E-01
11	0.0	-0.2045E-01	-0.3684E-01	-0.5203E-01	-0.6107E-01	-0.7324E-01	-0.7274E-01	-0.7819E-01	-0.7027E-01	-0.6657E-01
10	0.0	-0.2224E-01	-0.3932E-01	-0.5455E-01	-0.6315E-01	-0.7485E-01	-0.7376E-01	-0.7920E-01	-0.7124E-01	-0.6782E-01
9	0.0	-0.2391E-01	-0.4155E-01	-0.5670E-01	-0.6475E-01	-0.7593E-01	-0.7424E-01	-0.7983E-01	-0.7198E-01	-0.6924E-01
8	0.0	-0.2542E-01	-0.4351E-01	-0.5849E-01	-0.6583E-01	-0.7644E-01	-0.7411E-01	-0.8007E-01	-0.7263E-01	-0.7071E-01
7	0.0	-0.2686E-01	-0.4521E-01	-0.5988E-01	-0.6641E-01	-0.7632E-01	-0.7330E-01	-0.7990E-01	-0.7313E-01	-0.7245E-01
6	0.0	-0.2825E-01	-0.4667E-01	-0.6091E-01	-0.6648E-01	-0.7549E-01	-0.7171E-01	-0.7926E-01	-0.7352E-01	-0.7472E-01
5	0.0	-0.2945E-01	-0.4781E-01	-0.6160E-01	-0.6605E-01	-0.7380E-01	-0.6932E-01	-0.7812E-01	-0.7392E-01	-0.7795E-01
4	0.0	-0.3040E-01	-0.4862E-01	-0.6201E-01	-0.6517E-01	-0.7110E-01	-0.6598E-01	-0.7660E-01	-0.7456E-01	-0.8282E-01
3	0.0	-0.3086E-01	-0.4904E-01	-0.6215E-01	-0.6417E-01	-0.6703E-01	-0.6148E-01	-0.7526E-01	-0.7590E-01	-0.9049E-01
2	0.0	-0.3042E-01	-0.4890E-01	-0.6236E-01	-0.6373E-01	-0.6069E-01	-0.7292E-01	-0.1021	-0.9990E-01	-0.1288
1	0.0	-0.3042E-01	-0.4890E-01	-0.6236E-01	-0.6373E-01	-0.6069E-01	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.0	0.0								

23	-0.9693E-02	0.0
22	-0.4182E-01	0.0
21	-0.4706E-01	0.0
20	-0.4852E-01	0.0
19	-0.4902E-01	0.0
18	-0.4965E-01	0.0
17	-0.5066E-01	0.0
16	-0.5199E-01	0.0
15	-0.5349E-01	0.0
14	-0.5503E-01	0.0
13	-0.5652E-01	0.0
12	-0.5794E-01	0.0
11	-0.5935E-01	0.0
10	-0.6090E-01	0.0
9	-0.6244E-01	0.0
8	-0.6413E-01	0.0
7	-0.6628E-01	0.0
6	-0.6939E-01	0.0
5	-0.7434E-01	0.0
4	-0.8285E-01	0.0
3	-0.9833E-01	0.0
2	-0.1446	0.0
1	0.0	0.0

VELOCITY PROFILE - GRID(6,19)

Z-MTS	U(Z)-M/S	V(Z)-M/S	SPEED-M/S	DIREC-DEG
0.0	0.184E-01	-0.121E-01	0.221E-01	-33.4
10.0	0.225E-01	-0.332E-01	0.401E-01	-55.9
25.0	0.109E-01	-0.508E-01	0.520E-01	-77.9
30.0	0.992E-02	-0.536E-01	0.545E-01	-79.5
39.0	0.134E-01	-0.537E-01	0.553E-01	-76.0
50.	0.158E-01	-0.404E-01	0.434E-01	-68.7

UNDETERMINED PARAMETERS:

	1	2	3
C(J)	0.809E-01	0.162E-01	0.800E-02
D(J)	-0.156E 00	-0.395E-02	-0.218E-02

MASS FLUXES-SQ MTS/SEC:
QX= 0.0 QY= 0.0

SURFACE HEIGHT - MTS:-0.565E-01

VELOCITY PROFILE - GRID(9,19)

Z-MTS	U(Z)-M/S	V(Z)-M/S	SPEED-M/S	DIREC-DEG
0.0	0.140E-01	0.437E-01	0.459E-01	72.2
5.0	0.251E-01	0.130E-01	0.283E-01	27.4
12.0	0.916E-02	-0.753E-02	0.119E-01	-39.4
17.0	-0.139E-01	0.172E-03	0.139E-01	179.3

UNDETERMINED PARAMETERS:

	1	2	3
C(J)	0.724E-01	0.342E-01	-0.832E-02
D(J)	-0.915E-01	-0.196E-01	0.896E-02

MASS FLUXES-SQ MTS/SEC:
QX= 0.0 QY= 0.0

SURFACE HEIGHT - MTS:-0.608E-01

VELOCITY PROFILE - GRID(6,12)

Z-MTS	U(Z)-M/S	V(Z)-M/S	SPEED-M/S	DIREC-DEG
0.0	-0.132E-01	0.383E-02	0.138E-01	163.8
5.0	-0.770E-02	-0.819E-02	0.112E-01	-133.2

12.0	-0.531E-02	-0.231E-01	0.237E-01 -102.9
60.0	0.125E-03	-0.118E-01	0.118E-01 -89.4
100.0	DEPTH SPECIFIED IS BELOW THE STILL WATER DEPTH FOR THE GRID(60.0)		
140.0	DEPTH SPECIFIED IS BELOW THE STILL WATER DEPTH FOR THE GRID(60.0)		

UNDETERMINED PARAMETERS:

	1	2	3
C(J)	0.507E-01	0.204E-01	0.236E-02
D(J)	-0.147E 00	0.334E-02	-0.278E-02

MASS FLUXES-SQ MTS/SEC:
QX= 0.0 QY= 0.0

SURFACE HEIGHT - MTS:-0.711E-01

CURRENT SPEEDS AT LEVEL 0.0 % IN MTS/SEC

X DIRECTION

-Y-	1	2	3	4	5	6	7	8	9	10
24	0.3880E-01	0.3880E-01	0.4018E-01	0.3552E-01	0.1667E-01	0.0	0.0	0.0	0.0	0.0
23	0.3704E-01	0.3704E-01	0.4099E-01	0.4179E-01	0.2328E-01	0.6971E-01	0.5933E-01	0.1203	0.9299E-01	0.1373
22	0.3324E-01	0.3324E-01	0.4184E-01	0.4950E-01	0.4492E-01	0.3474E-01	0.2328E-01	0.8548E-01	0.7050E-01	0.1245
21	0.2687E-01	0.2687E-01	0.4133E-01	0.5533E-01	0.5804E-01	0.2798E-01	0.2461E-01	0.6683E-01	0.5073E-01	0.1179
20	0.1336E-01	0.1336E-01	0.3954E-01	0.5910E-01	0.6901E-01	0.2528E-01	0.3139E-01	0.5564E-01	0.5143E-01	0.1128
19	0.3967E-02	0.1094E-01	0.3804E-01	0.6139E-01	0.6801E-01	0.2206E-01	0.3388E-01	0.4792E-01	0.4592E-01	0.1090
18	0.4574E-01	0.4973E-02	0.3810E-01	0.6170E-01	0.6799E-01	0.1836E-01	0.3335E-01	0.4331E-01	0.4213E-01	0.1067
17	0.2127E-01	0.3225E-01	0.4618E-01	0.6268E-01	0.6556E-01	0.1536E-01	0.3109E-01	0.4150E-01	0.4042E-01	0.1058
16	0.8418E-01	0.6499E-01	0.5919E-01	0.6472E-01	0.6453E-01	0.1352E-01	0.2701E-01	0.4304E-01	0.4056E-01	0.1061
15	0.1505	0.1002	0.7453E-01	0.6755E-01	0.6220E-01	0.1252E-01	0.2395E-01	0.4535E-01	0.4203E-01	0.1070
14	0.2174	0.1361	0.9082E-01	0.7094E-01	0.5971E-01	0.1211E-01	0.1981E-01	0.4805E-01	0.4412E-01	0.1083
13	0.2843	0.1722	0.1075	0.7480E-01	0.5719E-01	0.1244E-01	0.1564E-01	0.5207E-01	0.4632E-01	0.1095
12	0.3494	0.2080	0.1244	0.7911E-01	0.5470E-01	0.1375E-01	0.1168E-01	0.5605E-01	0.4830E-01	0.1104
11	0.4164	0.2443	0.1416	0.8391E-01	0.5235E-01	0.1610E-01	0.8209E-02	0.5765E-01	0.4968E-01	0.1107
10	0.4844	0.2809	0.1591	0.8920E-01	0.5022E-01	0.1931E-01	0.5934E-02	0.5961E-01	0.5074E-01	0.1104

9	0.5507	0.3173	0.1767	0.9493E-01	0.4839E-01	0.2322E-01	0.6194E-02	0.6153E-01	0.5090E-01	0.1092
8	0.6189	0.3541	0.1945	0.1011	0.4693E-01	0.2785E-01	0.8983E-02	0.6297E-01	0.5045E-01	0.1066
7	0.6843	0.3904	0.2124	0.1075	0.4598E-01	0.3350E-01	0.1329E-01	0.6455E-01	0.4942E-01	0.1023
6	0.7515	0.4268	0.2303	0.1143	0.4593E-01	0.4079E-01	0.1903E-01	0.6602E-01	0.4849E-01	0.9643E-01
5	0.8185	0.4627	0.2478	0.1212	0.4712E-01	0.5064E-01	0.2656E-01	0.7077E-01	0.4856E-01	0.8955E-01
4	0.8848	0.4972	0.2641	0.1278	0.5055E-01	0.6412E-01	0.3683E-01	0.7690E-01	0.4982E-01	0.8337E-01
3	0.9480	0.5275	0.2775	0.1337	0.5640E-01	0.8190E-01	0.5118E-01	0.8463E-01	0.5379E-01	0.8159E-01
2	1.011	0.5478	0.2852	0.1370	0.6203E-01	0.1013	0.2243E-01	0.3420E-01	0.3555E-01	0.3759E-01
1	1.018	0.5338	0.2826	0.1374	0.6144E-01	0.8655E-01	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.0	0.0								
23	0.9527E-01	0.0								
22	0.9596E-01	0.0								
21	0.9553E-01	0.0								
20	0.9425E-01	0.0								
19	0.9281E-01	0.0								
18	0.9179E-01	0.0								
17	0.9131E-01	0.0								
16	0.9131E-01	0.0								
15	0.9162E-01	0.0								
14	0.9206E-01	0.0								
13	0.9246E-01	0.0								
12	0.9268E-01	0.0								
11	0.9262E-01	0.0								
10	0.9243E-01	0.0								
9	0.916[illegible]E-01	0.0								
8	0.90[illegible]E-01	0.0								
7	0.8714E-01	0.0								
6	0.8264E-01	0.0								
5	0.7603E-01	0.0								
4	0.6714E-01	0.0								
3	0.5810E-01	0.0								

2	0.2667E-01	0.0
1	0.0	0.0

CURRENT DIRECTIONS AT LEVEL 0.0 % IN DEG FROM THE X AXIS

	X DIRECTION									
-Y-	1	2	3	4	5	6	7	8	9	10
24	-72.68	-72.68	-65.57	-58.41	25.57	0.0	0.0	0.0	0.0	0.0
23	-71.82	-71.82	-66.08	-63.56	-49.63	119.2	124.9	124.8	123.5	122.3
22	-69.24	-69.24	-62.42	-61.79	-50.32	64.98	70.06	102.5	102.4	106.3
21	-64.58	-64.58	-59.09	-62.46	-55.49	19.99	-1.398	85.44	85.27	97.58
20	-49.52	-49.52	-57.16	-65.47	-62.12	-9.908	-30.20	76.42	75.40	93.74
19	0.0	68.62	-59.55	-70.87	-69.30	-33.41	-46.43	74.96	72.21	92.79
18	89.67	-86.96	-80.80	-80.81	-77.30	-56.64	-58.40	78.86	73.39	93.27
17	-125.8	-112.7	-100.3	-91.07	-84.96	-81.79	-67.73	84.90	76.72	94.14
16	-111.6	-118.5	-113.0	-100.1	-91.56	-107.4	-74.65	90.11	80.24	94.92
15	-110.3	-121.4	-121.0	-107.8	-97.08	-131.7	-79.12	93.20	82.73	95.42
14	-110.0	-123.2	-126.2	-114.4	-101.8	-154.7	-81.03	94.35	83.96	95.59
13	-109.9	-124.3	-129.9	-120.2	-106.2	-176.6	-79.98	94.12	84.15	95.51
12	-110.0	-125.1	-132.7	-125.4	-110.7	163.8	-74.79	93.07	83.59	95.24
11	-110.0	-125.7	-134.9	-130.2	-115.6	148.2	-62.32	91.58	82.51	94.85
10	-110.0	-126.1	-136.8	-134.8	-121.2	136.6	-35.07	89.78	80.99	94.44
9	-110.1	-126.5	-138.4	-139.0	-127.6	128.0	3.648	87.62	78.91	94.03
8	-110.1	-126.9	-139.9	-143.1	-135.0	121.3	29.00	84.94	75.82	92.97
7	-110.2	-127.2	-141.2	-146.9	-143.5	115.8	41.94	81.79	71.05	91.31
6	-110.2	-127.5	-142.3	-150.5	-153.4	111.4	50.46	78.08	64.86	88.93
5	-110.2	-127.6	-143.3	-153.8	-164.4	108.6	59.08	75.34	59.16	85.95
4	-110.3	-127.6	-143.9	-156.6	-175.3	107.7	70.62	76.27	59.35	84.02
3	-109.7	-127.4	-144.2	-158.6	175.9	109.4	86.69	85.37	76.45	89.25
2	-109.1	-127.1	-144.2	-159.6	171.3	112.7	180.0	180.0	180.0	180.0
1	90.00	-126.3	-144.9	-160.2	176.2	116.6	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.0	0.0								
23	118.6	0.0								
22	105.9	0.0								
21	99.27	0.0								
20	96.69	0.0								
19	96.03	0.0								
18	96.19	0.0								
17	96.57	0.0								
16	96.92	0.0								
15	97.14	0.0								
14	97.22	0.0								
13	97.19	0.0								
12	97.08	0.0								
11	96.96	0.0								
10	97.09	0.0								
9	96.90	0.0								
8	96.44	0.0								
7	95.72	0.0								
6	94.63	0.0								
5	93.10	0.0								
4	91.54	0.0								
3	93.77	0.0								
2	180.0	0.0								
1	0.0	0.0								

CURRENT SPEEDS AT LEVEL 50. % IN MTS/SEC

X DIRECTION

-Y-	1	2	3	4	5	6	7	8	9	10
24	0.4911E-01	0.4911E-01	0.5130E-01	0.4861E-01	0.2670E-01	0.0	0.0	0.0	0.0	0.0
23	0.4746E-01	0.4746E-01	0.5183E-01	0.5399E-01	0.4555E-01	0.1892E-01	0.2380E-01	0.3955E-01	0.2944E-01	0.3789E-01
22	0.4414E-01	0.4414E-01	0.5146E-01	0.5806E-01	0.5652E-01	0.2761E-01	0.2459E-01	0.1625E-01	0.1625E-01	0.3053E-01

21	0.3874E-01	0.3874E-01	0.5094E-01	0.6225E-01	0.6533E-01	0.4129E-01	0.3677E-01	0.3443E-02	0.2800E-02	0.3035E-01
20	0.3063E-01	0.3063E-01	0.5128E-01	0.6698E-01	0.7225E-01	0.4952E-01	0.4651E-01	0.1230E-01	0.5723E-02	0.3034E-01
19	0.1900E-03	0.1964E-01	0.5387E-01	0.7203E-01	0.7753E-01	0.5453E-01	0.5261E-01	0.1718E-01	0.9887E-02	0.2991E-01
18	0.5388E-01	0.1870E-01	0.5408E-01	0.7496E-01	0.8072E-01	0.5731E-01	0.5583E-01	0.1923E-01	0.1150E-01	0.2704E-01
17	0.6085E-01	0.2720E-01	0.5611E-01	0.7711E-01	0.8239E-01	0.5850E-01	0.5689E-01	0.1976E-01	0.1168E-01	0.2571E-01
16	0.6813E-01	0.3596E-01	0.5984E-01	0.7916E-01	0.8306E-01	0.5850E-01	0.5636E-01	0.1929E-01	0.1092E-01	0.2536E-01
15	0.7537E-01	0.4439E-01	0.6453E-01	0.8125E-01	0.8318E-01	0.5757E-01	0.5471E-01	0.1802E-01	0.9508E-02	0.2585E-01
14	0.8281E-01	0.5271E-01	0.6971E-01	0.8342E-01	0.8295E-01	0.5601E-01	0.5236E-01	0.1613E-01	0.7715E-02	0.2684E-01
13	0.9046E-01	0.6101E-01	0.7508E-01	0.8559E-01	0.8253E-01	0.5409E-01	0.4969E-01	0.1386E-01	0.5826E-02	0.2793E-01
12	0.9834E-01	0.6922E-01	0.8042E-01	0.8765E-01	0.8195E-01	0.5193E-01	0.4694E-01	0.1144E-01	0.4162E-02	0.2881E-01
11	0.1062	0.7729E-01	0.8549E-01	0.8945E-01	0.8116E-01	0.4966E-01	0.4427E-01	0.9099E-02	0.3228E-02	0.2923E-01
10	0.1147	0.8503E-01	0.9016E-01	0.9086E-01	0.8005E-01	0.4724E-01	0.4171E-01	0.7088E-02	0.3683E-02	0.2895E-01
9	0.1220	0.9245E-01	0.9438E-01	0.9178E-01	0.7850E-01	0.4458E-01	0.3927E-01	0.5893E-02	0.5547E-02	0.2801E-01
8	0.1299	0.9938E-01	0.9801E-01	0.9207E-01	0.7638E-01	0.4151E-01	0.3684E-01	0.6325E-02	0.8722E-02	0.2610E-01
7	0.1374	0.1059	0.1010	0.9163E-01	0.7350E-01	0.3780E-01	0.3427E-01	0.8642E-02	0.1329E-01	0.2338E-01
6	0.1445	0.1120	0.1032	0.9037E-01	0.6965E-01	0.3307E-01	0.3130E-01	0.1212E-01	0.1926E-01	0.2106E-01
5	0.1512	0.1173	0.1045	0.8832E-01	0.6477E-01	0.2689E-01	0.2733E-01	0.1564E-01	0.2600E-01	0.2208E-01
4	0.1572	0.1215	0.1051	0.8574E-01	0.5910E-01	0.1876E-01	0.2153E-01	0.1760E-01	0.3103E-01	0.2729E-01
3	0.1621	0.1241	0.1052	0.8340E-01	0.5350E-01	0.9649E-02	0.1337E-01	0.1520E-01	0.2777E-01	0.2940E-01
2	0.1654	0.1249	0.1056	0.8222E-01	0.4970E-01	0.1076E-01	0.3807E-02	0.2395E-02	0.9597E-02	0.3585E-02
1	0.1157	0.1251	0.1061	0.8362E-01	0.5341E-01	0.1315E-01	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

X DIRECTION

-Y-	11	12	13	14	15	16	17	18	19	20
24	0.0	0.0								
23	0.2359E-01	0.0								
22	0.2809E-01	0.0								
21	0.3127E-01	0.0								
20	0.3212E-01	0.0								
19	0.3156E-01	0.0								
18	0.3068E-01	0.0								
17	0.3006E-01	0.0								
16	0.2987E-01	0.0								
15	0.3004E-01	0.0								

14	0.3040E-01	0.0
13	0.3078E-01	0.0
12	0.3101E-01	0.0
11	0.3098E-01	0.0
10	0.3075E-01	0.0
9	0.3004E-01	0.0
8	0.2856E-01	0.0
7	0.2604E-01	0.0
6	0.2229E-01	0.0
5	0.1777E-01	0.0
4	0.1587E-01	0.0
3	0.1921E-01	0.0
2	0.2888E-02	0.0
1	0.0	0.0

CURRENT DIRECTIONS AT LEVEL 50.% IN DEG FROM THE X AXIS

	X DIRECTION									
-Y-	1	2	3	4	5	6	7	8	9	10
24	-90.59	-90.59	-85.39	-84.35	-80.95	0.0	0.0	0.0	0.0	0.0
23	-90.61	-90.61	-85.44	-84.91	-84.71	-132.8	-144.9	176.9	-174.0	158.8
22	-90.23	-90.23	-82.41	-81.27	-76.71	-67.53	-94.79	172.6	176.1	120.9
21	-90.62	-90.62	-79.82	-79.72	-75.94	-66.39	-78.02	-89.95	172.8	92.71
20	-92.01	-92.01	-77.93	-80.28	-78.71	-72.31	-78.09	-55.06	-35.10	78.20
19	0.0	-89.45	-75.92	-82.08	-82.82	-79.52	-82.50	-64.29	-47.25	71.96
18	76.09	-46.15	-78.05	-85.64	-87.48	-86.73	-87.87	-76.77	-60.95	70.72
17	66.85	-28.41	-77.11	-87.84	-91.28	-92.91	-92.86	-88.92	-73.45	72.31
16	61.25	-24.35	-78.74	-88.37	-93.65	-97.46	-96.85	-99.36	-83.82	74.93
15	57.23	-23.21	-72.05	-87.76	-94.69	-100.3	-99.63	-107.7	-91.92	77.31
14	54.04	-22.70	-69.37	-86.48	-94.73	-101.6	-101.2	-113.9	-97.37	79.64
13	51.38	-22.28	-66.82	-84.87	-94.13	-101.9	-101.8	-119.0	-99.03	78.89
12	49.21	-21.86	-64.49	-83.17	-93.19	-101.3	-101.6	-119.8	-93.35	73.19
11	47.33	-21.32	-62.35	-81.48	-92.09	-100.3	-100.7	-118.3	-74.40	76.68
10	46.05	-20.61	-60.37	-79.83	-90.92	-99.02	-99.35	-110.5	-50.48	74.34
9	44.59	-20.01	-58.53	-78.23	-89.75	-97.51	-97.45	-92.30	-40.21	70.73
8	43.77	-19.23	-56.75	-76.64	-88.56	-95.80	-94.93	-66.04	-38.77	64.17

Sample PRTVEL Output

7	42.99	-18.40	-54.99	-75.04	-87.41	-93.96	-91.71	-46.13	-41.28	52.67
6	42.34	-17.41	-53.19	-73.39	-86.32	-92.17	-87.69	-36.11	-44.58	32.52
5	41.90	-16.28	-51.37	-71.75	-85.47	-91.24	-83.76	-31.23	-47.71	4.308
4	41.73	-15.01	-49.58	-70.17	-85.13	-94.73	-81.11	-29.29	-51.08	-19.81
3	41.97	-13.75	-48.04	-68.83	-85.96	-120.8	-85.25	-34.66	-58.32	-34.89
2	42.66	-13.28	-47.26	-68.08	-87.43	161.5	180.0	180.0	180.0	0.0
1	90.00	-13.63	-47.50	-68.46	-87.66	-140.9	0.0	0.0	0.0	0.0

CONTINUED FROM PREVIOUS PAGE

-Y-	11	12	13	14	X DIRECTION 15	16	17	18	19	20
24	0.0	0.0								
23	141.7	0.0								
22	107.7	0.0								
21	90.55	0.0								
20	83.10	0.0								
19	80.16	0.0								
18	79.54	0.0								
17	79.98	0.0								
16	80.74	0.0								
15	81.38	0.0								
14	81.72	0.0								
13	81.71	0.0								
12	81.35	0.0								
11	80.69	0.0								
10	80.04	0.0								
9	78.65	0.0								
8	76.18	0.0								
7	71.82	0.0								
6	63.36	0.0								
5	44.81	0.0								
4	7.433	0.0								
3	-29.06	0.0								
2	0.0	0.0								
1	0.0	0.0								

DISK FILE CONTAINING CURRENT SPEED ANDDIRECTION FOR LEVEL 0.0 % WAS CREATED

DISK FILE CONTAINING CURRENT SPEED ANDDIRECTION FOR LEVEL 50. % WAS CREATED

* * * * * A TOTAL OF 2 PLOTS WERE CREATED

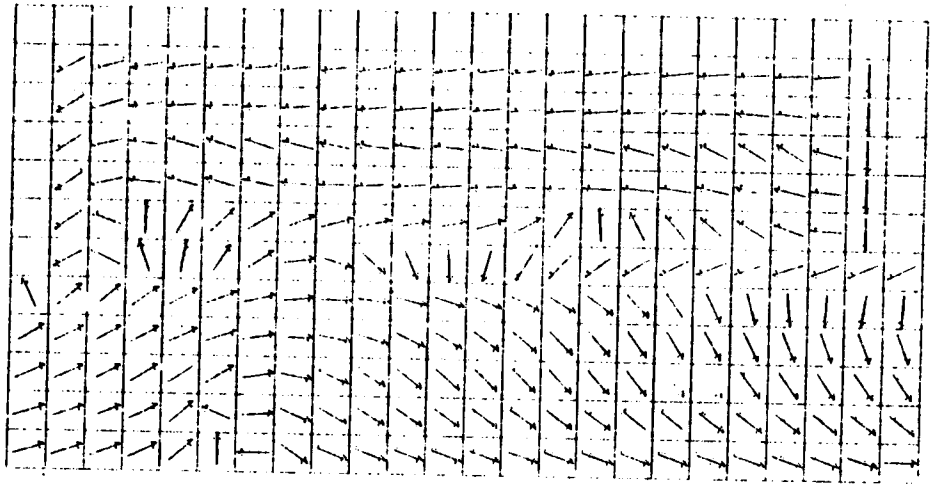
Sample Input

```
 WFSCM - RUN 21-5 -  SUMMER PRODUCTION RUN
  &MISC   FACT=.6,GRDSZ=.5,LP=12,MP=24,DL=30000,
          ISKIP=0,INC=0,SCALE=.FALSE.,SPDSCL=.1,DIST=4.,
          ANGI=60.,NOPLTS=2,NWF1=13,PAT=F,VECTOR=F,&END
```

Sample PLOTVEL Input

WFSCM - NECE

Sample Output



Program SPAT

Sample Input

The input for SPAT generates a wind field for the FSU study, starting in 25 February 1978. Observations from five stations are used as the basis for extrapolation - the NDBO data buoy, Apalachicola, Tampa, Fort Myers, and Key West. Observations are provided at one hour increments. The wind field generated by the sample input is only five hours long, which is sufficient to illustrate the use of the program

```
 &GRID LIMIT=5, IP=24,MP=12,N=5,
      IX=-4,6,11,11,8,
      IY=13,23,18,8,1,
      SYSW=1.0,2.54,2.54,2.09,1.37,
      INT=3,&END
FLA SHELF WINDS FROM 25 FEB 78 0400 HRS GMT WRT GRID NORTH
 -147 -146  189   -7  112 -111   78 -260 -258 -237
 -145 -129  174    6  105  -96   55 -239 -252 -240
 -152 -122  199   -1  126  -86   57 -243 -277 -238
 -165 -118  192  -10  116  -76   60 -231 -263 -220
 -180 -120  185   -9  122  -78   50 -219 -287 -225
```

Sample Output

```
FLA SHELF WINDS FROM 25 FEB 78 0400 HRS GMT WRT GRID NORTH
TIME =    1.
TIME =    2.
TIME =    3.
TIME =    4.
TIME =    5.


READING TERMINATED DUE TO EOF OR ERROR
 AFTER      6 RECORDS.
```

Program DENSTAT

Sample Input

```
&MISC ISTIME=0401,IETIME=0930,&END
```

Sample Output

SIGMA-T STATISTICS FOR DEPTHS 0 -10 M 1 APR TO 30 SEP

MEAN	22.55	22.01	21.69						30°
STD.DEV.	1.95	1.55	1.47						
MINIMUM	20.86	20.40	19.56						
MAXIMUM	24.24	24.10	23.16						
NO. OBS.	4	11	10						
MEAN	22.21	22.11	22.31	21.36					
STD. DEV.	0.96	1.55	1.04	1.20					
MINIMUM	20.77	19.52	21.17	18.80					
MAXIMUM	23.18	24.04	23.55	23.00					
NO. OBS.	12	21	12	19					
MEAN	22.42	24.69	22.71	22.16					
STD. DEV.	0.86	1.62	0.88	0.79					
MINIMUM	21.62	21.35	21.71	20.96					
MAXIMUM	23.57	26.21	24.14	23.64					
NO. OBS.	20	60	21	23					
MEAN	22.60	22.85	22.60	21.89	21.84				
STD. DEV.	1.67	0.68	0.70	0.86	0.83				
MINIMUM	19.30	21.70	21.25	20.35	21.19				
MAXIMUM	24.42	23.75	24.11	22.54	22.36				
NO. OBS.	16	29	29	21	10				
MEAN	22.34	23.23	23.32	23.28	22.53				
STD. DEV.	1.76	1.24	1.28	1.10	1.03				
MINIMUM	18.26	20.11	20.71	21.33	20.61				
MAXIMUM	24.42	25.98	26.02	26.05	24.57				28°
NO. OBS.	23	66	71	148	91				
MEAN	23.23	23.15	23.60	23.43	23.06				
STD. DEV.	0.82	1.17	1.13	1.20	0.99				
MINIMUM	22.18	21.67	22.06	20.92	20.52				
MAXIMUM	24.53	24.74	25.79	25.61	24.59				
NO. OBS.	14	20	26	80	43				
MEAN	22.93	23.35	23.00	23.24	23.32	22.44			
STD. DEV.	0.55	0.85	1.07	1.09	0.87	1.39			
MINIMUM	21.99	21.93	20.87	21.51	21.71	19.45			
MAXIMUM	23.80	24.66	24.36	24.42	24.97	24.52			
NO. OBS.	21	18	24	14	45	52			
MEAN	23.54	23.67	23.55	23.55	23.27	22.65	20.48		
STD. DEV.	0.57	0.49	0.44	0.59	0.97	1.23	2.47		
MINIMUM	22.50	22.03	21.00	22.00	21.51	18.63	15.53		
MAXIMUM	24.96	25.09	25.01	25.74	25.35	25.03	23.55		
NO. OBS.	26	181	296	57	60	90	9		
MEAN	23.20	23.40	22.97	23.17	23.54	22.91	23.07		
STD. DEV.	0.77	0.73	0.48	0.79	0.62	0.34	0.61		
MINIMUM	22.18	22.74	22.40	22.21	22.40	22.30	22.05		
MAXIMUM	24.71	24.55	24.17	24.65	24.23	24.01	24.00		26°
NO. OBS.	28	8	76	29	19	93	26		
MEAN	23.31	23.32	23.24	23.08	23.50	23.48	23.12	22.71	
STD. DEV.	0.57	0.54	0.75	0.64	1.01	0.60	0.92	0.48	
MINIMUM	22.55	22.49	22.12	22.55	21.77	22.37	22.14	21.92	
MAXIMUM	24.22	24.20	25.15	25.25	25.23	25.18	24.96	23.80	
NO. OBS.	18	32	42	134	98	173	28	186	
	80°				85°				

Sample DENSTAT Output

SIGMA-T STATISTICS FOR DEPTHS 30 - 100 M 1 APR TO 30 SEP

MEAN							
STD. DEV.							
MINIMUM							
MAXIMUM							30°
NO. OBS.							
MEAN	25.62						
STD. DEV.	0.36						
MINIMUM	25.15						
MAXIMUM	26.00						
NO. OBS.	4						
MEAN	25.46	25.11	25.40				
STD. DEV.	0.83	0.74	0.0				
MINIMUM	23.64	24.36	25.40				
MAXIMUM	26.52	26.43	25.40				
NO. OBS.	23	19	2				
MEAN	25.41	25.01	24.66				
STD. DEV.	0.74	0.74	0.63				
MINIMUM	23.19	22.96	23.28				
MAXIMUM	26.72	26.15	25.74				
NO. OBS.	56	40	25				
MEAN	25.31	25.22	25.20	24.97			
STD. DEV.	0.86	0.77	0.60	0.72			
MINIMUM	22.98	23.50	23.30	22.88			
MAXIMUM	26.49	26.60	26.23	26.27			28°
NO. OBS.	55	112	104	65			
MEAN	24.67	25.42	25.42	24.94			
STD. DEV.	0.78	0.59	0.69	0.81			
MINIMUM	22.97	23.74	24.03	23.00			
MAXIMUM	26.26	26.49	26.46	26.33			
NO. OBS.	59	41	41	87			
MEAN	24.72	25.21	25.30	25.46	25.91		
STD. DEV.	0.90	1.20	0.76	0.88	0.08		
MINIMUM	23.31	23.17	23.30	23.90	25.86		
MAXIMUM	26.43	26.81	26.18	26.53	25.97		
NO. OBS.	59	39	54	18	2		
MEAN	24.84	24.86	24.85	25.09	24.93		
STD. DEV.	0.64	0.52	0.54	0.59	0.60		
MINIMUM	23.67	23.26	23.59	23.34	23.73		
MAXIMUM	26.13	26.24	26.84	26.07	26.01		
NOL OBS.	100	976	1662	85	52		
MEAN	24.78	24.60	24.72	24.99	24.83	23.21	
STD. DEV.	0.75	0.90	0.80	0.77	0.81	0.07	
MINIMUM	23.41	23.25	22.94	23.48	23.26	23.07	
MAXIMUM	26.22	25.95	26.70	26.14	25.90	23.34	26°
NO. OBS.	96	26	268	72	16	38	
MEAN	25.20	24.56	24.82	24.56	24.63	23.95	
STD. DEV.	0.76	0.97	0.89	0.83	0.69	0.91	
MINIMUM	23.54	22.97	23.04	22.82	23.30	23.08	
MAXIMUM	26.34	26.62	26.68	26.23	25.84	25.18	
NO. OBS.	85	144	156	338	43	16	
	85°				83°		

Appendix I

Program Listings

```
00010 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
00020 C
00030 C
00040 C                          PROGRAM VISCOUS(GAL)
00050 C
00060 C                   VERSION 7 - JULY 1982 - REVISION A
00070 C
00080 C                          BY CORTIS COOPER, NECE
00090 C
00100 C     THIS PROGRAM CALCULATES ALL CONSTANTS WHICH CHANGE ONLY WHEN THE
00110 C     EDDY VISCOSITY CHANGES.  THE CONSTANTS ARE WRITTEN ON DISK FILE
00120 C     AND LATER ACCESSED BY PROGRAM CIRC WHICH IS THE PROGRAM THAT
00130 C     ACTUALLY CALCULATES VELOCITIES.  FOR INFORMATION ON HOW TO USE
00140 C     THE PROGRAM AS WELL AS CIRC, SEE THE GAL USER'S MANUAL.  THIS
00150 C     MANUAL ALSO CONTAINS A LIST OF REVISIONS.
00160 C 6.A, 10/81, SPLIT COMMON /AA/ INTO TWO BLOCKS.  CHANGED 1.0E70'S
00170 C      TO 1.0E20'S.  INITIALIZED ALL COMMON VARIABLES IN
00180 C      BLOCK DATA. PUT NAMELIST READ'S INTO MAIN PROGRAM.
00190 C      REMOVED &'S FROM ALTERNATE RETURN CALLS.  PUT 'KPM1' IN
00200 C      COMMON /AA/.
00210 C 6.E, 2/82, ADDED SPATIAL VARIATION IN CORIOLIS PARAMETER.
00220 C 7.A, 7/82, MODIFIED TO RUN ON IBM. CHANGED FCN ASIN TO ARSIN.
00230 C
00240       COMMON/AA/CF(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
00250      &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),O(864),
00260      &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
00270      &   JLMP,KLMP,G,NBP,REYNC,WA2,NS,IS,KT.DL,
00280      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
00290      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
00300 C
00310       REAL NB,NBP
00320       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY,NODIM
00330 C
00340       NAMELIST/MISC/DL,JP,KP,LP,MP,NS,NUCHG,CONNV
```

```
00350       NAMELIST/DEPTH/H
00360       NAMELIST/FRICT/CB
00370       NAMELIST/EDDYL/NODIM,CK
00380       NAMELIST/EDYVIS/EDDYS,EDDYT,WA2,REYNO,EDDYV,ETIME
00390       NAMELIST/CORIO/FXY,PHI,THETA
00393 C
00395       DATA NODIM/T/
00400 C
00410       READ(1,100) TITLE
00420 100   FORMAT(20A4)
00430       READ(1,MISC)
00440       READ(1,CORIO)
00450       LPMP=LP*MP
00460       KPLPMP=KP*LPMP
00470       JPLPMP=JP*LPMP
00480       JJLPMP=JP*JPLPMP
00490       KLMM1=LP*MP*(KP-1)
00500       KPM1=KP-1
00510       READ(1,DEPTH)
00520       CALL HEIGHT
00530       READ(1,FRICT)
00540       READ(1,EDDYL)
00550       CALL EDYLYR(NODIM,1000)
00560       CALL CORIOL
00570       CALL HEADER
00580       IF(EDDYT) READ(1,EDYVIS)
00590       IF(EDDYT) NUCHG=99
00600       DO 999 II=1,NUCHG
00610       IF(.NOT.EDDYT) READ(1,EDYVIS)
00620       IF(EDDYS.OR.EDDYT) CALL WINDNV(II)
00630       CALL CHEKEV(1000)
00640       IF(II.EQ.1) WRITE(10)TITLE,LP,MP,JP,KP,CONNV,EDDYS,EDDYT,REYNO,WA2
00650      &            ,DL,PHI,THETA,FXY
00660       IF(II.EQ.1) WRITE(10) (H(LM),LM=1,LPMP),(LAND(LM),LM=1,LPMP),
00670      & (CB(LM),LM=1,LPMP),(CK(KLM),KLM=1,KPLPMP),(F(LM),LM=1,LPMP)
00680 C  * * * MAIN SPATIAL LOOP
00690 C
00700       DO 500 M=1,MP
00710       LMP=(M-1)*LP
00720       DO 500 L=1,LP
00730       LM=L+LMP
00740       IF(LAND(LM)) GO TO 500
00750       JLMP=(LM-1)*JP
00760       KLMP=(LM-1)*KPM1
00770       KT=(LM-1)*KP
00780       IS=KLMP+1
00790       CALL BOTTN
00800       CALL AJLM
```

```
00810       CALL QJLM
00820       CALL COEFF
00830 500   CONTINUE
00840       CALL PRINTS
00850       IF(CONNV) JJLPMP=JPLPMP
00860       IF(II.EQ.NUCHG) ETIME=1.0E20
00870       WRITE(10) ETIME,(A(JLM),JLM=1,JPLPMP),(Q(JLM),JLM=1,JPLPMP),
00880      &          (B(JLM),JLM=1,JPLPMP),(DELSIG(JJLM),JJLM=1,JJLPMP),
00890      &          (D1(JLM),JLM=1,JPLPMP),(E(JLM),JLM=1,JPLPMP),
00900      &          (SS(JLM),JLM=1,JPLPMP),(FMASS(LM),LM=1,LPMP),
00910      &          (EDDYV(KLM),KLM=1,KPLPMP),(NB(LM),LM=1,LPMP),
00920      &          (ALPHA(KLM),KLM=1,KLMM1),(BETA(KLM),KLM=1,KLMM1)
00930 999   CONTINUE
00940 1000  STOP
00950       END
00960 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
00970 C
00980       BLOCK DATA
00990 C
01000       COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
01010      &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),Q(864),
01020      &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
01030      &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
01040      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
01050      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
01060       REAL NB,NBP
01070       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
01080 C
01090       DATA VERSIN/'GAL:','VERS','ION ','7 - ','JULY',' 4, ',
01100      &     '1982',' - R','EV. ','A - ','PROG','RAM ','VISC','OUS '/
01110       DATA TITLE/20*'    '/,ETIME/1.0E20/
01120       DATA CONSTE,EDDYT,EDDYS,CONNV,FXY,NUCHG/T,F,T,T,T,1/
01123       DATA PHI,THETA/23.5,65./
01130       DATA DELSIG/2592*0.0/
01140       DATA A,SS,B,E,D1,Q/864*0.0,864*0.0,864*0.0,864*0.0,864*0.0,
01150      &     864*1.0/
01160       DATA CK,EDDYV/1552*0.0,1552*0.0/
01170       DATA ALPHA,BETA/864*0.0,864*0.0/
01180       DATA H,NB,CB,FMASS,UA2,VA2,F,LAND/288*0.0,288*0.0,
01190      &     288*.00025,288*0.0,288*0.0,288*0.0,288*0.0,288*.FALSE./
01200       DATA JP,KP,LP,MP,LPMP,LM,KPM1,JLMP,KLMP,IS,KT,NS/
01203      &     3,2,12,24,288,0,1,1,1,1,1,20/
01210       DATA G,REYNO,WA2,DL,NBP/9.8,16.,.005,30000.,1./
01220       END
01230 C
01240 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01250 C
01260 C
```

```
01270       SUBROUTINE EDYLYR(NODIM,*)
01280 C
01290 C
01300 C  ROUTINE WHICH READS IN THE LAYER DEPTH OF THE NODES OF
01310 C  THE EDDY VISCOSITY FOR EACH GRID.
01320       COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
01330      &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),Q(864),
01340      &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
01350      &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
01360      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
01370      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
01380       REAL NB,NBP
01390       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY,NODIM
01400       LPMP=LP*MP
01410       DO 100 LM=1,LPMP
01420       IF(LAND(LM)) GO TO 100
01430       KT=(LM-1)*KP+1
01440       HT=H(LM)
01450       IF(NODIM)   HT=1.0
01460       KLMP=(LM-1)*KP
01470       DO 70 K=1,KP
01480       KLM=KLMP+K
01490       CK(KLM)=CK(KLM)/HT
01500       IF(CK(KLM).GT.1.0.OR.CK(KLM).LT.0.0) CALL ERROR(1,LM,K)
01510 70    CONTINUE
01520 C   CHECKS TO MAKE SURE CK AT SURFACE IS EQUAL TO ZERO AND CK AT BOTTOM
01530 C   IS EQUAL TO 1.
01540       KTKP=KT+KP-1
01550       IF(CK(KT).GT.0.01.AND.CK(KTKP).LT.0.99) CALL ERROR(2,LM,K)
01560 100   CONTINUE
01570       IF(KP.GT.2.AND.CONNV) CALL ERROR(3,LM,K)
01580       IF(KP.EQ.2.AND.(.NOT.CONNV)) CALL ERROR(4,LM,K)
01590       IF(KP.EQ.1) CALL ERROR(5,LM,K)
01600       RETURN
01610 200   RETURN1
01620       END
01630 C
01640 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01650 C
01660 C
01670       SUBROUTINE HEIGHT
01680 C
01690 C
01700 C  ROUTINE WHICH READS IN STILL WATER DEPTHS AND SETS LOGICAL VARIABLE
01710 C  TO TRUE IF ELEMENT IS LAND.
01720 C
01730       COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
01740      &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),Q(864),
```

```
01750      &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
01760      &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
01770      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
01780      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
01790       REAL NB,NBP
01800       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
01810       DO 100 M=1,MP
01820       LMP=(M-1)*LP
01830       DO 100 L=1,LP
01840       LM=L+LMP
01850       IF(H(LM).GT.0.01) GO TO 100
01860       LAND(LM)=.TRUE.
01870  100  CONTINUE
01880       RETURN
01890       END
01900 C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01910 C
01920       SUBROUTINE WINDNV(II)
01930 C
01940 C
01950 C  ROUTINE WHICH SETS CONSTANT EDDY VISCOSITY USING THE TURBULENT
01960 C  REYNOLDS NUMBER FROM TOWSEND.
01970 C
01980       COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
01990      &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),Q(864),
02000      &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
02010      &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
02020      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
02030      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
02040       REAL NB,NBP
02050       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
02060       DATA WTIME/0.0/
02070 C
02080       CONST=1./REYNO
02090       IF(EDDYT.AND.ETIME.GE.WTIME) READ(4) WTIME,
02100      &   (UA2(LM),LM=1,LPMP),(VA2(LM),LM=1,LPMP),
02110      &   (DUM,LM=1,LPMP),(DUM,LM=1,LPMP),
02120      &   (DUM,LM=1,LPMP)
02130       WTIME=WTIME*3600.
02140       IF(EDDYT) ETIME=WTIME
02150       DO 100 LM=1,LPMP
02160       IF(LAND(LM)) GO TO 100
02170       KLM=(LM-1)*2+1
02180       IF(EDDYT) WA2=(UA2(LM)*UA2(LM)+VA2(LM)*VA2(LM))**.25
02190       EDDYV(KLM)=CONST*WA2*H(LM)
02200       EDDYV(KLM+1)=EDDYV(KLM)
02210  100  CONTINUE
02220       IF(WTIME.GE.1.0E20) II=NUCHG
```

```
02230       RETURN
02240       END
02250 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02260 C
02270        SUBROUTINE CHEKEV(*)
02280 C
02290 C   ROUTINE WHICH CHECKS TO MAKE SURE EDDY VISC HAS NOT BEEN SPECIFIED A
02300 C    ZERO ON A NONLAND ELEMENT.
02310 C
02320       COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
02330      &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),Q(864),
02340      &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
02350      &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
02360      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
02370      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
02380       REAL NB,NBP
02390       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
02400       DO 200 LM=1,LPMP
02410       IF(LAND(LM)) GO TO 200
02420       KLMP=(LM-1)*KP
02430       DO 100 K=1,KP
02440       KLM=KLMP+K
02450       IF(EDDYV(KLM).LT.1.E-20) CALL ERROR(6,LM,K)
02460 100   CONTINUE
02470 200   CONTINUE
02480       RETURN
02490 500   RETURN1
02500       END
02510 C
02520 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02530 C
02540       SUBROUTINE BOTTN
02550 C
02560 C
02570 C     ROUTINE WHICH CALCULATES THE EDDY VISCOSITY AT THE BOTTOM AND THE
02580 C     SLOPE (ALPHA) AND INTERCEPT(BETA) OF THE EDDY VISCOSITY.
02590 C
02600       COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
02610      &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),Q(864),
02620      &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
02630      &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
02640      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
02650      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
02660       REAL NB,NBP
02670       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
02680       ALPHA(IS)=(EDDYV(KT+2)-EDDYV(KT+1))/(CK(KT+2)*H(LM))
02690       BETA(IS)=EDDYV(KT+1)
02700       NB(LM)=ALPHA(IS)*H(LM)+BETA(IS)
```

```
02710       IF(KP.LE.2) GO TO 30
02720       DO 20 K=2,KPM1
02730       KLM=KLMP+K
02740       KTK=KT+K
02750       ALPHA(KLM)=(EDDYV(KTK+1)-EDDYV(KTK))/(H(LM)*(CK(KTK+1)-CK(KTK)))
02760       BETA(KLM)=EDDYV(KTK+1)-ALPHA(KLM)*H(LM)*CK(KTK+1)
02770 20    CONTINUE
02780 C
02790 C  CALCULATE THE EDDY VISCOSITY AT THE BOTTOM, NBP.
02800 30    NBP=EDDYV(KT+KP)
02810 C  CHECKS TO SEE IF CONSTATNT EDDY VISCOSITY HAS BEEN SPECIFIED.
02820 C  IF SO THEN A LOGICAL VARIABLE IS SET TO TRUE.
02830       CONSTE=.FALSE.
02840       DIF=ABS((EDDYV(KT+1)-NB(LM))/NB(LM))
02850       IF(DIF.LT.0.01) CONSTE=.TRUE.
02860 C  CALCULATES THE PARAMTER G AND THEN FMASS.  USES ONE OF TWO
02870 C  FORMAULAS TO CALCULATE G DEPENDING ON WHETHER OR NOT EDDY
02880 C  VISCOSITY IS CONSTANT.
02890       G=H(LM)*H(LM)*0.5/BETA(IS)
02900       IF(CONSTE) GO TO 15
02910       G=(H(LM)*ALOG(NB(LM))-(NB(LM)*ALOG(NB(LM);-NB(LM)-
02920      &  BETA(IS)*ALOG(BETA(IS))+BETA(IS))/ALPHA(IS))/ALPHA(IS)
02930 15    FMASS(LM)=-H(LM)*H(LM)/(12.0*NB(LM))+G
02940       RETURN
02950       END
02960 C
02970 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02980 C
02990 C
03000       SUBROUTINE AJLM
03010 C
03020 C
03030 C
03040 C      ROUTINE WHICH CALCULATES THE COEFFICIENT A(JLM)
03050 C
03060 C
03070       COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
03080      &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),C(864),
03090      &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
03100      &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
03110      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
03120      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
03130       REAL NB,NBP
03140       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
03150       DO 10 J=1,JP
03160       JLM=J+JLMP
03170       C=CB(LM)*H(LM)/NBP
03180       AP=(2.0*FLOAT(J)-1.0)*3.14159/2.0
```

```
5     A0=C/AP
      A(JLM)=ATAN(A0)+(FLOAT(J)-1.0)*3.14159
      A1=A(JLM)-AP
      A2=ABS(A1)
      IF(A2 .LE. 0.01) GO TO 10
      AP=A(JLM)
      GO TO 5
10    CONTINUE
      RETURN
      END
C
C * * * * * * * * * *  ** * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE QJLM
C
C    ROUTINE WHICH CALCULATES THE COEFFICIENT Q(JLM).
C
      COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
     &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),Q(864),
     &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
     &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
     &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
     &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
C
      REAL NB,NBP
      LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
C
      DO 100 J=1,JP
      JLM=JLMP+J
      AJLM2=2.0*A(JLM)
      Q(JLM)=AJLM2/(AJLM2+SIN(AJLM2))
100   CONTINUE
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
      SUBROUTINE COEFF
C
C
C     ROUTINE WHICH CALCULATES THE COEFFICIENTS:  DELSIG, B, E,
C     D1, SS, AND A
C
C
      COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
     &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),Q(864),
     &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
```

```
03670      &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
03680      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
03690      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
03700       REAL NB,NBP
03710       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
03720 C
03730       KKLM=2*(LM-1)+1
03740       DO 400 J=1,JP
03750       JLM=J+JLMP
03760       ASQRD=A(JLM)*A(JLM)
03770       IF(CONNV) GO TO 80
03780       IJLMP=(JLM-1)*JP
03790       DO 70 I=1,JP
03800       IJLM=I+IJLMP
03810       ILM=I+JLMP
03820       THET=A(ILM)-A(JLM)
03830       THETP=A(ILM)+A(JLM)
03840       S1=0.0
03850       S2=0.0
03860       S3=0.0
03870 C
03880       DO 50 K=2,KP
03890       KTK=KT+K
03900       KLM=(K-1)+KLMP
03910       IF(I .EQ. J) GO TO 25
03920       RLAMA=(COS(THET*CK(KTK))-COS(THET*CK(KTK-1)))/THET+
03930      &   (COS(THETP*CK(KTK))-COS(THETP*CK(KTK-1)))/THETP
03940       SLAMA=(SIN(THET*CK(KTK))-SIN(THET*CK(KTK-1)))/THET+
03950      &   (SIN(THETP*CK(KTK))-SIN(THETP*CK(KTK-1)))/THETP
03960       XI=(COS(THET*CK(KTK))-COS(THET*CK(KTK-1)))/THET**2
03970      &   +(COS(THETP*CK(KTK))-COS(THETP*CK(KTK-1)))/THETP**2+
03980      &(CK(KTK)*SIN(THET*CK(KTK))-CK(KTK-1)*SIN(THET*CK(KTK-1)))
03990      &   /THET+(CK(KTK)*SIN(THETP*CK(KTK))-CK(KTK-1)*SIN(THETP*
04000      &   CK(KTK-1)))/THETP
04010       GO TO 35
04020 25    RLAMA=((COS(A(ILM)*CK(KTK)))**2-(COS(A(ILM)*CK(KTK-1)))**2)
04030      &  /A(ILM)
04040       SLAMA=CK(KTK)-CK(KTK-1)+(SIN(2.0*A(ILM)*CK(KTK))
04050      &   -SIN(2.0*A(ILM)*CK(KTK-1)))/(2.0*A(ILM))
04060       XI=(CK(KTK)**2-CK(KTK-1)**2)*0.5+(COS(2.0*A(ILM)*CK(KTK))-
04070      &   COS(2.0*A(ILM)*CK(KTK-1)))/(4.0*A(ILM)**2)+(CK(KTK)
04080      &   *SIN(2.0*A(ILM)*CK(KTK))-
04090      &   CK(KTK-1)*SIN(2.0*A(ILM)*CK(KTK-1)))/(2.0*A(ILM))
04100 35    S1=ALPHA(KLM)*XI+S1
04110       S2=SLAMA*BETA(KLM)/H(LM)+S2
04120       S3=ALPHA(KLM)*RLAMA+S3
04130 50    CONTINUE
04140 C
```

```
04150        DEL=A(ILM)*S3*0.5
04160        SIGMA=(S1+S2)*A(ILM)*A(ILM)*0.5
04170        DELSIG(IJLM)=(DEL-SIGMA)*O(JLM)
04180 C      IF(IJLM.GE.73.AND.IJLM.LT.93) WRITE(3,600) S1,S2,S3,DEL,SIGMA,
04190 C     &    DELSIG(IJLM),A(JLM),A(ILM)
04200 70     CONTINUE
04210        GO TO 85
04220 C
04230 80     CALL DS(JLM,KKLM,ASORD)
04240 85     S4=0.0
04250        S5=0.0
04260        S6=0.0
04270        S7=0.0
04280        DO 100 K=2,KP
04290        KTK=KT+K
04300        KLM=K-1+KLMP
04310        GAMA=2.0*CK(KTK)*COS(A(JLM)*CK(KTK))/A(JLM)**2+((A(JLM)*
04320       &   CK(KTK))**2-2.0)*SIN(A(JLM)*CK(KTK))/A(JLM)**3-2.0*
04330       &   CK(KTK-1)*COS(A(JLM)*CK(KTK-1))/
04340       &   A(JLM)**2-((A(JLM)*CK(KTK-1))**2-2.0)*SIN(A(JLM)*CK(KTK-1))/
04350       &   A(JLM)**3
04360        PHII=(COS(A(JLM)*CK(KTK))+A(JLM)*CK(KTK)*SIN(A(JLM)*
04370       & CK(KTK))-COS(A(JLM)*CK(KTK-1))-A(JLM)*CK(KTK-1)*SIN(A(JLM)*
04380       &  CK(KTK-1)))/(A(JLM)*A(JLM))
04390        SP =(SIN(CK(KTK)*A(JLM))-SIN(CK(KTK-1)*A(JLM)))/A(JLM)
04400        S4=ALPHA(KLM)*GAMA+S4
04410        S5=ALPHA(KLM)*PHII+S5
04420        S6=BETA(KLM)*SP+S6
04430        S7=BETA(KLM)*PHII+S7
04440 100    CONTINUE
04450 C
04460        PHII=(COS(A(JLM))+SIN(A(JLM))*A(JLM)-1.0)/ASORD
04470        PSI=(SIN(A(JLM))+3.0*COS(A(JLM))/A(JLM)-6.0*PHII/A(JLM))/A(JLM)
04480        GAMA=(2.0*A(JLM)*COS(A(JLM))+(ASORD-2.0)*SIN(A(JLM)))/(ASORD*
04490       &   A(JLM))
04500 C
04510        RNU=0.0
04520        RNUP=0.0
04530        GAMAF=0.0
04540        IF(CONSTE) GO TO 200
04550        Z1=0.0
04560        A1=0.0
04570        NS1=NS*J
04580        DELZ=H(LM)/FLOAT(NS1)*0.5
04590        DO 120 I=1,NS1
04600        Z2=FLOAT(I)/FLOAT(NS1)
04610        A2=SIN(A(JLM)*Z2)/(ALPHA(IS)*Z2*H(LM)+BETA(IS))
04620        GAMAF=GAMAF+A2+A1
```

```
04630        A1=A2
04640 120    CONTINUE
04650        GAMAF=GAMAF*DELZ/A(JLM)
04660 C
04670        IF(KP.LE.2) GO TO 200
04680        Z4=A(JLM)/H(LM)
04690        DO 160 K=3,KP
04700        KTK=KT+K
04710        KLM=(K-1)+KLMP
04720        A6=0.0
04730        DZ=(CK(KTK)-CK(KTK-1))*H(LM)/FLOAT(NS)
04740        Z3=CK(KTK-1)*H(LM)
04750        Z1=Z3
04760        B1=1.0/(ALPHA(IS)*Z1+BETA(IS))
04770        C11=Z4*Z1
04780        A1=(ALPHA(KLM)*Z1+BETA(KLM))*COS(C11)*B1*B1
04790        A3=COS(C11)*B1
04800        A5=0.5*DZ
04810 C
04820        DO 150 I=1,NS
04830        Z2=Z3+DZ*FLOAT(I)
04840        B2=1.0/(ALPHA(IS)*Z2+BETA(IS))
04850        C22=Z4*Z2
04860        A2=(ALPHA(KLM)*Z2+BETA(KLM))*COS(C22)*B2*B2
04870        A4=COS(C22)*B2
04880        A6=A6+(A3+A4)*A5
04890        RNUP=RNUP+(A2+A1)*A5
04900        Z1=Z2
04910        B1=B2
04920        C11=C22
04930        A1=A2
04940        A3=A4
04950 150    CONTINUE
04960        RNU=A6*ALPHA(KLM)+RNU
04970 160    CONTINUE
04980        RNUP=RNUP*ALPHA(IS)
04990 C
05000 200    IF(.NOT.CONNV) AP=(H(LM)*(9.0*S4-4.0*S5)+6.0*S7-2.0*S6)/NB(LM)+
05010       &               (RNUP-RNU)
05020        B(JLM)=SIN(A(JLM))/A(JLM)
05030        SS(JLM)=B(JLM)
05040        B(JLM)=B(JLM)*Q(JLM)
05050        IF(CONSTE) GAMAF=H(LM)/BETA(IS)*(SIN(A(JLM))/A(JLM)-PHII)
05060        IF(CONNV) AP=-(6.-6.*COS(A(JLM))-4.*SIN(A(JLM))*A(JLM))/ASQRD
05070        E(JLM)=((PSI-GAMA)*H(LM)/NB(LM)+GAMAF)*Q(JLM)
05080        D1(JLM)=AP/H(LM)*Q(JLM)
05090 400    CONTINUE
05100        RETURN
```

```
05110       END
05120 C
05130 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
05140 C
05150       SUBROUTINE DS(JLM,KLM,ASORD)
05160 C
05170 C      ROUTINE WHICH CALCULATES DELSIG FOR CASE WHEN NV IS INDEPENDENT
05180 C      OF Z.
05190 C
05200       COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
05210      &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),O(864),
05220      &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
05230      &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
05240      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
05250      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
05260       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
05270 C
05280       DELSIG(JLM)=-EDDYV(KLM)*ASORD*0.5/H(LM)
05290       RETURN
05300       END
05310 C
05320 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
05330 C
05340 C
05350       SUBROUTINE OUTPUT (LP,MP,Z,INDEX)
05360 C
05370 C
05380 C  ROUTINE WHICH PRINTS A TABLE.  MAIN OUTPUT ROUTINE USED IN THIS PROGR
05390 C
05400       DIMENSION Z(1)
05410       LPAGE=0
05420       MPAGE=(LP-1)/10+1
05430       LPI=LP*INDEX
05440       DO 800 MM=1,MPAGE
05450       LSTART=10*(MM-1)*INDEX+1
05460       LSTOP=10*MM*INDEX
05470       JSTOP=10*MM
05480       JSTART=10*(MM-1)+1
05490       IF(MM.EQ.MPAGE) LSTOP=LP*INDEX
05500       IF(MM.GT.1) WRITE(3,100)
05510 100   FORMAT('1',////,30X,'CONTINUED FROM PREVIOUS PAGE')
05520       WRITE(3,200)(J,J=JSTART,JSTOP)
05530 200   FORMAT(53X,'  X DIRECTION',/,' -Y-',6X,10(I2,10X))
05540       WRITE(3,300)
05550 300   FORMAT(' ---',6X,10('--',10X))
05560       DO 700 M=1,MP
05570       NN=MP+1-M
05580       LM=(NN-1)*LPI
```

```
05590       DO 500 N=1,INDEX
05600       LMINDX=LM+N-1
05610       WRITE(3,400) NN,(Z(LMINDX+L),L=LSTART,LSTOP,INDEX)
05620 400   FORMAT(' ',I2,3X,10(G10.3,2X))
05630 500   CONTINUE
05640       NPAGE=(M*INDEX)/48
05650       IF(NPAGE.EQ.LPAGE) GO TO 700
05660       LPAGE=NPAGE
05670       WRITE(3,600)
05680 600    FORMAT('1',////)
05690 700   CONTINUE
05700 800   CONTINUE
05710       RETURN
05720       END
05730 C
05740 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
05750 C
05760 C
05770       SUBROUTINE HEADER
05780 C
05790 C
05800 C     ROUTINE WHICH PRINTS INFORMATION SUMMARIZING EXECUTION OF
05810 C     PROGRAM CONSTANT.
05820 C
05830       COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
05840      &   B(864),E(864),D1(864),ALPHA(864),BETA(864),NB(288),Q(864),
05850      &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
05860      &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
05870      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
05880      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
05890       REAL NB,NBP
05900       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
05910 C
05920       DATA ASTER/'* * '/
05930       WRITE(3,10) (ASTER,I=1,30),VERSIN
05940 10    FORMAT(///,'1',30A4,//,' ',14A4,/,' ',56(1H-))
05950       WRITE(3,15) TITLE
05960 15    FORMAT(/,' ',20A4)
05970       WRITE(3,20)JP,KP,LP,MP,DL
05980 20    FORMAT(/,' JP=',I1,3X,'KP=',I1,3X,'LP=',I2,3X,'MP=',I2,3X,
05990      &       'DL=',G13.3)
06000       IF(CONNV) WRITE(3,30)
06010 30    FORMAT(/,' NV HAS BEEN TAKEN AS CONSTANT IN THE VERTICAL.')
06020       WRITE(3,40) PHI
06030 40    FORMAT(/,' LATITUDE OF GRID ORIGIN=',F3.0,' DEG')
06040       IF(FXY) WRITE(3,45) THETA
06050 45    FORMAT(/,' BETA-PLANE APPROXIMATION USED - ANGLE BETWEEN',
06060      &       ' X-AXIS AND TRUE NORTH=',F4.0)
```

```
06070       WRITE(3,50)
06080 50    FORMAT(//'1STILL WATER DEPTHS IN METERS FOLLOW:')
06090       CALL OUTPUT(LP,MP,H,1)
06100       WRITE(3,60)
06110 60    FORMAT(//,'1THE LAYER DEPTHS FOR THE EDDY VISCOSITY FOLLOW ',
06120      &   '(NONDIMENSIONAL):')
06130       CALL OUTPUT(LP,MP,CK,KP)
06140       WRITE(3,130)
06150 130   FORMAT(//,'1FRICTION COEFFICIENTS (M/SEC):')
06160       CALL OUTPUT(LP,MP,CB,1)
06170       WRITE(3,10)(ASTER,I=1,30)
06180       RETURN
06190       END
06200 C
06210 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
06220 C
06230       SUBROUTINE PRINTS
06240 C
06250 C     ROUTINE WHICH PRINTS NV(KLM) AND A(JLM) WHENEVER NV CHANGES.
06260 C
06270       COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
06280      &   B(864),E(864),D(864),ALPHA(864),BETA(864),NB(288),Q(864),
06290      &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
06300      &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
06310      &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
06320      &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
06330 C
06340       REAL NB,NBP
06350       LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
06360 C
06370       DATA ASTER/'* * '/
06380 C
06390       WRITE(3,10) (ASTER,I=1,30)
06400 10    FORMAT(///,'1',30A4)
06410       TIME=ETIME/3600.
06420       WRITE(3,100) TIME
06430 100   FORMAT(//5X,21(1H-)/4X,'/ TIME =',
06440      &F6.2,' HOURS',2X,'/',/,4X,21(1H-))
06450       IF(EDDYS.OR.EDDYT) WRITE(3,200) REYNO
06460 200   FORMAT(/,' REYNOLDS NO.=',F5.1)
06470       IF (EDDYS.AND.(.NOT.EDDYT)) WRITE(3,300) WA2
06480 300   FORMAT(/,' THE AVG. WIND STRESS USED IN NV MODEL=',E10.3)
06490       WRITE(3,400)
06500 400   FORMAT(//,'1NV (SQ MTS/SEC):')
06510       CALL OUTPUT(LP,MP,EDDYV,KP)
06520       WRITE(3,500)
06530 500   FORMAT(//,'1A(JLM):')
06540       CALL OUTPUT(LP,MP,A,JP)
```

```
      WRITE(3,600)
600   FORMAT(//,'1F(L,M) - CORIOLIS PARAMETER')
      CALL OUTPUT(LP,MP,F,1)
      WRITE(3,10)(ASTER,I=1,30)
      RETURN
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE ERROR(IERR,LM,K)
C
C
      WRITE(3,100) IERR,LM,K
100   FORMAT(///,' * * ERROR NO.',I2,' HAS OCCURED EXECUTION ABORTED'
     &       ,/,' LM=',I5,5X,'K=',I2)
      STOP
      END
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE CORIOL
C
C
C     ROUTINE TO CALCULATE THE CORIOLIS PARAMETER, F.  IF FXY=TRUE THEN
C     F IS CALUCULATED AS A FUNCTION OF LATITUDE.  IF FALSE THEN F IS
C     CALCULATED BASED ON THE VALUE OF CPHI.
C
      COMMON/AA/CK(1552),DELSIG(2592),SS(864),H(288),FMASS(288),
     &   B(864),E(864),D1(864),ALPHA(864),BETA(864),HP(288),Q(864),
     &   A(864),CB(288),EDDYV(1552),JP,KP,LP,MP,LPMP,LM,KPM1,
     &   JLMP,KLMP,G,NBP,REYNO,WA2,NS,IS,KT,DL,
     &   ETIME,NUCHG,UA2(288),VA2(288),VERSIN(14),TITLE(20),
     &   CONSTE,EDDYT,EDDYS,CONNV,LAND(288),FXY,PHI,THETA,F(288)
C
      REAL NB,NBP
      LOGICAL CONSTE,LAND*1,EDDYT,EDDYS,CONNV,FXY
C
      DATA OMEGA/0.00014544/,RADIUS/6378000./,PI/3.14159/
C
      CONV=PI/180.
      F(1)=OMEGA*SIN(PHI*CONV)
      DO 500 M=1,MP
      LMP=(M-1)*LP
      MM1=M-1.
      DO 500 L=1,LP
      LM=LMP+L
      F(LM)=F(1)
      IF(.NOT.FXY) GOTO 500
      LM1=L-1
      DIST=SQRT(FLOAT(LM1*LM1+MM1*MM1))*DL
```

```
07030        PSI=90.
07040        IF(LM1.EQ.0) GOTO 50
07050        PSI=ATAN(FLOAT(MM1/LM1))/CONV
07060 50     PSI=PSI+90.-THETA
07070        ZETA=DIST*SIN(PSI*CONV)
07080        F(LM)=OMEGA*SIN(ARSIN(ZETA/RADIUS)+PHI*CONV)
07090 500    CONTINUE
07100        RETURN
07110        END
07120 C$ENTRY
EOF:
```

```
00010 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
00020 C
00030 C                      PROGRAM DENSITY (GAL)
00040 C
00050 C          VERSION 7 - JULY 1982 - REVISION A
00060 C
00070 C                      3-D CIRCULATION MODEL
00080 C
00090 C                    BY CORTIS COOPER, NECE
00100 C
00110 C     THIS PROGRAM CALCULATES THE DENSITY GRADIENT TERM WHICH IS
00120 C     USED IN PROGRAM CIRC.  PROGRAM VISCOUS MUST BE RUN BEFORE
00130 C     DENSITY.  SEE GAL USER'S MANUAL FOR INFORMATION ON HOW TO
00140 C     USE THIS PROGRAM.  THE MANUAL ALSO CONSTAINS A LIST OF REVISIONS.
00150 C
00160 C     REVISION 6.A IS FOR MULTICS.  ONLY DIFFERENCE BETWEEN 6.A AND
00170 C     5.F IS THAT MULTIPLE RETURN IN SUBROUTINE BOTTOM WAS
00180 C     REMOVED.  WAS NOT NEEDED ANYWAY.
00190 C
00200 C     REVISION 7.A FOR IBM.  SAME AS 6.A.
00210 C
00220       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
00230      &         SAL(864),RHOWP(864),RHOW(864),ROWX(864),
00240      &         ROWY(864),A(864),Q(864),CALDEN,
00250      &         LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
00260      &         KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
00270      &         ASTER,ETIME,LAND(288),BOTT(864)
00280       DIMENSION DROWX(10),DROWY(10)
00290       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
00300 C
00310       DATA NUCHG/1/
00320 C
00330       NAMELIST/MISC/NUCHG,CALDEN
00340       NAMELIST/SALIN/SAL
00350       NAMELIST/TEMP/T
00360       NAMELIST/LAYER/HP,IPP,DTIME
00370       NAMELIST/RHO/RHOWP
00380 C
00390       READ(1,100) TITLE1
00400 100   FORMAT(20A4)
00410       READ(1,MISC)
00420       READ(10) TITLE2,LP,MP,JP,KP,CONNV
00430       LPM1=LP-1
00440       KPLPMP=KP*LP*MP
00450       MPM1=MP-1
00460       LPMP=LP*MP
00470       JPLPMP=JP*LPMP
```

```
00480       READ(10)(H(LM),LM=1,LPMP),(LAND(LM),LM=1,LPMP),
00490      &        (DUM,LM=1,LPMP),(DUM,LM=1,KPLPMP),(DUM,LM=1,LPMP)
00500       CALL HEADER
00510       WRITE(11) TITLE1
00520       DO 999 II=1,NUCHG
00530       IF(DTIME.GT.ETIME) CALL RDNV
00540       READ(1,LAYER)
00550       IF(CALDEN) READ(1,SALIN)
00560       IF(CALDEN) READ(1,TEMP)
00570       IF(.NOT.CALDEN) READ(1,RHO)
00580       IPLP=IPP*LP
00590       IPLPMP=IPP*LPMP
00600       IPPP=IPP
00610       CALL SETRHO
00620       CALL CALCR(DROWX,DROWY,IPPP)
00630       CALL SETBCD
00640       CALL PRINTS
00650       IF(II.EQ.NUCHG) DTIME=1.0E30
00660       CALL WRTDSK
00670 999   CONTINUE
00680       STOP
00690       END
00700 C
00710 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
00720 C
00730       BLOCK DATA
00740 C
00750       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
00760      &         SAL(864),RHOWP(864),RHOW(864),ROWX(864),
00770      &         ROWY(864),A(864),Q(864),CALDEN,
00780      &         LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
00790      &         KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
00800      &         ASTER,ETIME,LAND(288),BOTT(864)
00810       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
00820 C
00830       DATA ASTER/'* * '/
00840       DATA T,SAL,RHOW/864*0.0,864*0.0,864*0.0/
00850       DATA BOTT/864*.FALSE./,CALDEN/.FALSE./
00860       DATA H,LAND/288*0.0,288*.TRUE./
00870       DATA ROWX,ROWY,A/864*0.0,864*0.0,864*0.0/
00880       DATA HP,DELHP,PHI,S/10.,500.,8*0.0,10*0.0,10*0.0,10*0.0/
00890       DATA DTIME/1.0E30/,ETIME/0.0/,IPP/2/
00900       DATA VERSIN/'GAL:','VERS','ION ','7 - ','JUNE',' 16 ',
00910      &     '1982',' - R','EV. ','A - ','DENS','ITY ',2*'    '/
00920       DATA TITLE1,TITLE2/40*'    '/
00930       END
00940 C
00950 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
00960 C
00970       SUBROUTINE RDNV
00980 C
00990 C      ROUTINE WHICH READS IN VARIABLES ASSOCIATED WITH THE EDDY VISC.
01000 C
01010       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
01020      &          SAL(864),RHOWP(864),RHOW(864),ROWX(864),
01030      &          ROWY(864),A(864),Q(864),CALDEN,
01040      &          LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
01050      &          KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
01060      &          ASTER,ETIME,LAND(288),BOTT(864)
01070 C
01080       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
01090 C
01100       KLMM1=LPMP*(KP-1)
01110       JJLPMP=JP*JPLPMP
01120       IF (CONNV) JJLPMP=JPLPMP
01130       READ(10) ETIME,(A(JLM),JLM=1,JPLPMP),(Q(JLM),JLM=1,JPLPMP),
01140      &          (DUM,JLM=1,JPLPMP),(DUM,JLM=1,JJLPMP),
01150      &          (DUM,JLM=1,JPLPMP),(DUM,JLM=1,JPLPMP),
01160      &          (DUM,JLM=1,JPLPMP),(DUM,LM=1,LPMP).
01170      &          (DUM,KLM=1,KPLPMP),(DUM,LM=1,LPMP),
01180      &          (DUM,KLM=1,KLMM1),(DUM,KLM=1,KLMM1)
01190       RETURN
01200       END
01210 C
01220 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01230 C
01240       SUBROUTINE SETRHO
01250 C
01260 C     ROUTINE WHICH:  (1) SETS THE VARIABLE BOTT TO TRUE AT THE LEVEL
01270 C     WHERE THE BOTTOM OCCURS, (10) SETS RHOW(IP,L,M)=0 FOR ALL LAYERS
01280 C     DEEPER THAN H(L,M) AND (10) CALCULATES RHOW(IP,L,M) ELSEWHERE.
01290 C
01300       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
01310      &          SAL(864),RHOWP(864),RHOW(864),ROWX(864),
01320      &          ROWY(864),A(864),Q(864),CALDEN,
01330      &          LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
01340      &          KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
01350      &          ASTER,ETIME,LAND(288),BOTT(864)
01360       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
01370 C
01380       DO 100 I=1,IPP
01390       ILMP=LPMP*(I-1)
01400       DO 100 LM=1,LPMP
01410       IPLM=(LM-1)*IPP+I
01420       ILM=ILMP+LM
01430       IF (HP(I).GT.H(LM)) CALL BOTTOM(IPLM,I)
```

```
01440       IF(CALDEN) RHOWP(ILM)=DEN(T(ILM),SAL(ILM))
01450 100   CONTINUE
01460       DO 300 LM=1,LPMP
01470       IPLMP=(LM-1)*IPP
01480       DO 300 I=1,IPP
01490       IPLM=IPLMP+I
01500       ILM=LPMP*(I-1)+LM
01510       RHOW(IPLM)=RHOWP(ILM)
01520 300   CONTINUE
01530 C     WRITE(3,111)
01540 C     WRITE(3,222) (RHOW(IACH), IACH=1,IPLPMP)
01550  222  FORMAT(1X,10G12.5)
01560  111  FORMAT(1X,'RHOW FROM SUBROUTINE SETRHO',/)
01570       RETURN
01580       END
01590 C
01600 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01610 C
01620 C
01630       SUBROUTINE BOTTOM(IPLM,I)
01640 C
01650 C     ROUTINE WHICH SETS THE VARIABLES BOTT(IP,L,M) AND RHOW(IP,L,M)
01660 C     ONCE THE BOTTOM FOR GRID (L,M) HAS BEEN LOCATED.
01670 C
01680       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
01690      &          SAL(864),RHOWP(864),RHOW(864),ROWX(864),
01700      &          ROWY(864),A(864),Q(864),CALDEN,
01710      &          LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
01720      &          KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
01730      &          ASTER,ETIME,LAND(288),BOTT(864)
01740       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
01750       DO 100 J=I,IPP
01760       BOTT(IPLM)=.TRUE.
01770       RHOW(IPLM)=0.0
01780       IPLM=IPLM+1
01790 100   CONTINUE
01800       RETURN
01810       END
01820 C
01830 C
01840 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01850 C
01860       FUNCTION DEN(T,S)
01870 C
01880 C     ROUTINE WHICH COMPUTES DENSITY GIVEN SALINITY(PPT) AND TEMPERATURE
01890 C     (DEGREES CELSIUS).  NOTE ALGORITHIM USED APPLIES ONLY TO SHALLOW
01900 C     WATER.
01910 C
```

```
01920 C  NEXT EQN FROM NEUMANN & PIERSON, PRINC. OF PHYS. OCEANOGRAPHY, P.40
01930       CL=(S-0.030)/1.805
01940 C  NEXT EQN FROM DEFANT, PHYS. OCEANOGRAPHY, P. 41
01950       SIG=-0.069+1.4708*CL-0.00157*CL*CL+0.0000398*CL**3
01960 C  REMAINING RELATIONSHIPS FROM H.O. PUB. 615, P. VII
01970       A=T*(4.7867-0.098185*T+0.0010843*T*T)/1000.0
01980       B=T*(18.030-0.8164*T+0.01667*T*T)/1000000.0
01990       D1=-((T-3.98)**2)/503.570
02000       D2=(T+283.0)/(T+67.26)
02010       D=D1*D2
02020       SIGMA=D+(SIG+0.1324)*(1.0-A+B*(SIG-0.1324))
02030       DEN=(1+.001*SIGMA)*1000.
02040       RETURN
02050       END
02060 C
02070 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02080 C
02090 C
02100       SUBROUTINE CALCR(DROWX,DROWY,IPPP)
02110 C
02120 C     ROUTINE WHICH CALCULATES THE DENSITY COEFFICIENTS, ROWX(J,L,M) &
02130 C     ROWY(J,L,M).
02140 C
02150       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
02160      &          SAL(864),RHOWP(864),RHOW(864),ROWX(864),
02170      &          ROWY(864),A(864),Q(864),CALDEN,
02180      &          LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
02190      &          KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
02200      &          ASTER,ETIME,LAND(288),BOTT(864)
02210       DIMENSION DROWX(IPPP),DROWY(IPPP)
02220       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
02230 C
02240       DELHP(1)=HP(1)
02250       IF(IPP.EQ.1) GO TO 200
02260       DO 100 IP=2,IPP
02270       DELHP(IP)=HP(IP)-HP(IP-1)
02280 100   CONTINUE
02290 C
02300 200   CONTINUE
02310       DO 1000 M=2,MPM1
02320       LMP=(M-1)*LP
02330       DO 999 L=2,LPM1
02340       LM=L+LMP
02350       IF (LAND(LM)) GO TO 999
02360       JLMP=(LM-1)*JP
02370       IPLMP=(LM-1)*IPP
02380       DO 999 J=1,JP
02390       JLM=J+JLMP
```

```
02400        HA=H(LM)/A(JLM)
02410        AH=1.0/HA
02420        HASQRD=HA*HA
02430        ZETA2=0.0
02440        DO 300 I=1,IPP
02450        IPLM=I+IPLMP
02460        IF(BOTT(IPLM)) GO TO 400
02470        ZETA1=ZETA2
02480        ZETA2=HP(I)*AH
02490        DROWX(I)=FROW(IPLM-IPP,IPLM,IPLM+IPP)
02500        DROWY(I)=FROW(IPLM-IPLP,IPLM,IPLM+IPLP)
02510        PHI(I)=(COS(ZETA2)+ZETA2*SIN(ZETA2)-COS(ZETA1)-ZETA1*SIN(ZETA1))
02520       &       *HASQRD
02530        S(I)=HA*(SIN(ZETA2)-SIN(ZETA1))
02540  300   CONTINUE
02550  400   IPLM=IPLMP+1
02560        IF(BOTT(IPLM)) GO TO 1000
02570        ROWX(JLM)=SUM(DROWX,IPPP,IPLMP)*Q(JLM)
02580        ROWY(JLM)=SUM(DROWY,IPPP,IPLMP)*Q(JLM)
02590 C      WRITE(3,111) JLM
02600 C      WRITE(3,222) (DROWY(IACH), IACH=1,IPP)
02610  999   CONTINUE
02620  1000  CONTINUE
02630   222  FORMAT(1X,10G12.5)
02640    111  FORMAT(1X,'JLM=  ',I5)
02650        RETURN
02660        END
02670 C
02680 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02690 C
02700        FUNCTION FROW(LOC1,LOC2,LOC3)
02710 C
02720 C
02730 C      FUNCTION WHICH CALCULATES THE EXPRESSION DELTA-RHOW.
02740 C
02750        COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
02760       &         SAL(864),RHOWP(864),RHOW(864),ROWX(864),
02770       &         ROWY(864),A(864),Q(864),CALDEN,
02780       &         LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
02790       &         KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
02800       &         ASTER,ETIME,LAND(288),BOTT(864)
02810        LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
02820        RHO1=RHOW(LOC1)
02830        RHO3=RHOW(LOC3)
02840        CONST=1.0
02850        IF(BOTT(LOC1)) RHO1=EROW(CONST,LOC2)
02860        IF(BOTT(LOC3)) RHO3=EROW(CONST,LOC2)
02870        FROW=(RHO3-RHO1)*CONST
```

```
02880       RETURN
02890       END
02900 C
02910 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02920 C
02930 C
02940       FUNCTION EROW(CONST,LOC2)
02950 C
02960 C     FUNCTION WHICH CORRECTS DENSITY GRADIENT FOR CASE OF OPEN
02970 C     WATER BOUNDARY OR LAND BOUNDARY CONDITION.
02980 C
02990       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
03000      &          SAL(864),RHOWP(864),RHOW(864),ROWX(864),
03010      &          ROWY(864),A(864),Q(864),CALDEN,
03020      &          LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
03030      &          KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
03040      &          ASTER,ETIME,LAND(288),BOTT(864)
03050       DIMENSION DROWX(10),DROWY(10)
03060       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
03070 C
03080       EROW=RHOW(LOC2)
03090       CONST=2.0
03100       RETURN
03110       END
03120 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
03130 C
03140       FUNCTION SUM(DROW,IPPP,IPLMP)
03150 C
03160 C     FUNCTION WHICH COMPUTES THE COEFFICIENT ROW(J,L,M) IN EITHER THE-
03170 C     X OR Y DIRECTION.
03180 C
03190       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
03200      &          SAL(864),RHOWP(864),RHOW(864),ROWX(864),
03210      &          ROWY(864),A(864),Q(864),CALDEN,
03220      &          LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
03230      &          KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
03240      &          ASTER,ETIME,LAND(288),BOTT(864)
03250       DIMENSION DROW(IPPP)
03260       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
03270 C
03280       SUM=0.0
03290       IF(IPP.EQ.1) GOTO 500
03300       SUM1=0.0
03310       DO 400 IP=2,IPP
03320       IPLM=IPLMP+IP
03330       IF(BOTT(IPLM)) GO TO 500
03340       SUM1=0.0
03350       IPM1=IP-1
```

```
03360       DO 200 II=1,IPM1
03370       SUM1=DELHP(II)*DROW(II)+SUM1
03380 200   CONTINUE
03390       SUM=((SUM1-HP(IP-1)*DROW(IP))*S(IP)+DROW(IP)*PHI(IP))/RHOW(IPLM)+
03400      &        SUM
03410 400   CONTINUE
03420 500   SUM=SUM+DROW(1)*PHI(1)/RHOW(IPLMP+1)
03430       RETURN
03440       END
03450 C
03460 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
03470 C
03480       SUBROUTINE SETBCD
03490 C
03500 C     ROUTINE WHICH SETS ROWX & ROWY AT (J,L,MP) AND (J,LP,M)
03510 C
03520       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(238),T(864),
03530      &         SAL(864),RHOWP(864),RHOW(864),ROWX(864),
03540      &         ROWY(864),A(864),Q(864),CALDEN,
03550      &         LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
03560      &         KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
03570      &         ASTER,ETIME,LAND(288),BOTT(864)
03580       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
03590       JPLP=JP*LP
03600       JLMP=(MPM1)*JPLP
03610       DO 100 I=1,JPLP
03620       JLM=JLMP+I
03630       ROWX(JLM)=ROWX(JLM-JPLP)
03640       ROWY(JLM)=ROWY(JLM-JPLP)
03650 100   CONTINUE
03660       DO 200 M=1,MP
03670       JLMP=M*JPLP-JP
03680       DO 200 J=1,JP
03690       JLM=J+JLMP
03700       ROWX(JLM)=ROWX(JLM-JP)
03710       ROWY(JLM)=ROWY(JLM-JP)
03720 200   CONTINUE
03730       RETURN
03740       END
03750 C
03760 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
03770 C
03780       SUBROUTINE HEADER
03790 C
03800 C
03810 C     ROUTINE WHICH PRINTS INFORMATION SUMMARIZING EXECUTION OF
03820 C     PROGRAM DENSITY.
03830 C
```

```
03840       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
03850      &          SAL(864),RHOWP(864),RHOW(864),ROWX(864),
03860      &          ROWY(864),A(864),Q(864),CALDEN,
03870      &          LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
03880      &          KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
03890      &          ASTER,ETIME,LAND(288),BOTT(864)
03900       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
03910 C
03920       WRITE(3,10) (ASTER,I=1,30),VERSIN
03930 10    FORMAT(///,'1',30A4,//,' ',10X,14A4,/,' ',10X,56(1H-))
03940       WRITE(3,15) TITLE1
03950 15    FORMAT(/,' ',20A4)
03960       WRITE(3,18) TITLE2
03970 18    FORMAT(/,' VISCOSITY TITLE:  ',20A4)
03980       WRITE(3,20)JP,LP,MP
03990 20    FORMAT(/,' JP=',I1,3X,'LP=',I2,3X,'MP=',I2)
04000       WRITE(3,50)
04010 50    FORMAT(//'1STILL WATER DEPTHS IN METERS FOLLOW:')
04020       CALL OUTPUT(LP,MP,H,1)
04030       RETURN
04040       END
04050 C
04060 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
04070 C
04080       SUBROUTINE PRINTS
04090 C
04100 C     ROUTINE WHICH PRINTS A(JLM),T(ILM),SAL(ILM) AND RHOW(ILM) EACH
04110 C     TIME THE DENSITY FIELD OR NV CHANGES.
04120 C
04130       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
04140      &          SAL(864),RHOWP(864),RHOW(864),ROWX(864),
04150      &          ROWY(864),A(864),Q(864),CALDEN,
04160      &          LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
04170      &          KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
04180      &          ASTER,ETIME,LAND(288),BOTT(864)
04190       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
04200 C
04210       WRITE(3,10)(ASTER,I=1,30)
04220 10    FORMAT('1',30A4)
04230       TIME=DTIME/3600.
04240       WRITE(3,20) TIME
04250 20    FORMAT(//5X,21(1H-)/4X,'/ TIME =',
04260      &F6.2,' HOURS',2X,'/',/,4X,21(1H-))
04270       WRITE(3,30)(HP(I),I=1,IPP)
04280 30    FORMAT(//,' THE LAYER DEPTHS FOLLOW(METERS):',/,' ',10(G10.3,3X))
04290       WRITE(3,105)
04300 105   FORMAT(//,'1A(JLM):')
04310       CALL OUTPUT(LP,MP,A,JP)
```

```
04320       WRITE(3,90)
04330 90    FORMAT(//,'1 THE DENSITY PROFILES FOLLOW(G/CUBIC CM):')
04340       CALL OUTPUT(LP,MP,RHOW,IPP)
04350       CALL OUTPUT( LP,MP,ROWX,JP)
04360       CALL OUTPUT(LP,MP,ROWY,JP)
04370       WRITE(3,10)(ASTER,I=1,30)
04380       RETURN
04390       END
04400 C
04410 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
04420 C
04430       SUBROUTINE WRTDSK
04440 C
04450 C     ROUTINE WHICH WRITES CONSTANTS CALULATED BY DENSITY ONTO DISK.
04460 C
04470       COMMON/AA/DELHP(10),PHI(10),S(10),HP(10),H(288),T(864),
04480      &          SAL(864),RHOWP(864),RHOW(864),ROWX(864),
04490      &          ROWY(864),A(864),Q(864),CALDEN,
04500      &          LP,MP,JP,IPP,IPLP,LPM1,MPM1,JPLPMP,LPMP,IPLPMP,
04510      &          KP,KPLPMP,CONNV,VERSIN(14),TITLE1(20),TITLE2(20),DTIME,
04520      &          ASTER,ETIME,LAND(288),BOTT(864)
04530       LOGICAL CALDEN,CONNV,LAND*1,BOTT*1
04540       INDEX=1
04550       DO 100 LM=1,IPLPMP,IPP
04560       RHOW(INDEX)=RHOW(LM)
04570       INDEX=INDEX+1
04580 100   CONTINUE
04590       WRITE(11)DTIME,IPP,(ROWX(JLM),JLM=1,JPLPMP),
04600      &         (ROWY(JLM),JLM=1,JPLPMP),(RHOW(LM),LM=1,LPMP)
04610 C     WRITE(3,900) (ROWX(I),I=1,JPLPMP)
04620 C     WRITE(3,900) (ROWY(I), I=1,JPLPMP)
04630 C     WRITE(3,900) (RHOW(I), I=1,LPMP)
04640   900 FORMAT(1X,10G10.3)
04650       RETURN
04660       END
04670 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
04680 C
04690 C
04700       SUBROUTINE OUTPUT (LP,MP,Z,INDEX)
04710 C
04720 C
04730 C  ROUTINE WHICH PRINTS A TABLE.  MAIN OUTPUT ROUTINE USED IN THIS PROGR
04740 C
04750       DIMENSION Z(1)
04760       LPAGE=0
04770       MPAGE=(LP-1)/10+1
04780       LPI=LP*INDEX
04790       DO 800 MM=1,MPAGE
```

```
04800         LSTART=10*(MM-1)*INDEX+1
04810         LSTOP=10*MM*INDEX
04820         JSTOP=10*MM
04830         JSTART=10*(MM-1)+1
04840         IF(MM.EQ.MPAGE) LSTOP=LP*INDEX
04850         IF(MM.GT.1) WRITE(3,100)
04860  100    FORMAT('1',////,30X,'CONTINUED FROM PREVIOUS PAGE')
04870         WRITE(3,200)(J,J=JSTART,JSTOP)
04880  200    FORMAT(53X,'  X DIRECTION',/,' -Y-',6X,10(I2,10X))
04890         WRITE(3,300)
04900  300    FORMAT(' ---',6X,10('--',10X))
04910         DO 700 M=1,MP
04920         NN=MP+1-M
04930         LM=(NN-1)*LPI
04940         DO 500 N=1,INDEX
04950         LMINDX=LM+N-1
04960         WRITE(3,400) NN,(Z(LMINDX+L),L=LSTART,LSTOP,INDEX)
04970  400    FORMAT(' ',I2,3X,10(F10.4,2X))
04980  500    CONTINUE
04990         NPAGE=(M*INDEX)/48
05000         IF(NPAGE.EQ.LPAGE) GO TO 700
05010         LPAGE=NPAGE
05020         WRITE(3,600)
05030  600    FORMAT('1',////)
05040  700    CONTINUE
05050  800    CONTINUE
05060         RETURN
05070         END
05090
EOF:
```

```
00010 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
00020 C
00030 C                        PROGRAM CIRC(GAL)
00040 C
00050 C                   VERSION 7.A - JULY 1982
00060 C
00070 C                FLORIDA SHELF 3-D CIRCULATION MODEL
00080 C
00090 C                     BY CORTIS COOPER, NECE
00100 C
00110 C                SEE USERS MANUAL FOR DIRECTIONS
00120 C
00130 C  6.A:  10/1/81 VERSION 5, REV I OF CIRC MODIFIED TO
00140 C        RUN ON MIT MULTICS.
00150 C  6.B:  11/1/81 ATMOSPHERIC PRESSURE GRADIENT TERMS REMOVED.
00160 C  6.C:  12/1/81 UPDATED TO INCLUDE LATERAL SHEAR TERMS.
00170 C  6.D:  1/8/82 BOUNDARY CONDITIONS ARE SPECIFICALLY SET
00180 C        FOR THE FLORIA SHELF GRID.  SEE SUBROUTINE BNDRY.
00190 C  6.E:  1/29/82 VARIATION IN THE CORIOLIS PARAMETER INCLUDED.
00200 C        OUTPUTS VELOCITY AND SURFACE ELEVATION IN F4.0 FORMAT
00210 C  6.F:  4/10/82 REMOVED SUBROUTINE CALFLO, SINCE IT NO LONGER WAS
00220 C        NEEDED.  SIMPLIFIED BCFLOW.  INCORPORATED SINUSOIDAL VARIATION
00230 C        OF DBC AT WESTERN BOUNDARY.  PUT IN REFLECTIONAL SYMETRY ON
00240 C        WESTERN BOUNDARY.  CORRECTED BUG IN HOT START OPTION.
00250 C  6.G:  5/15/82 CORRECT BUG IN ROUTINE BNDRY TO IMPORVE CALCULATION
00260 C        AT GRID (11,5) AND (11,23)
00270 C  6.H:  5/29/82 CHANGED OUTPUT SO THAT ROW 2, COLS 7-11 WERE SET
00280 C        EQUAL TO ROW 3, SAME COLS.
00290 C  7.A:  7/3/82 MODIFIED TO RUN ON IBM.  REMOVED OUTPUT ROUTINES.
00300 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
00310 C
00320 C              DEFININITON OF MAJOR VARIABLES USED IN PROGRAM
00330 C USES NOTATIONS FROM THEORECTICAL DESCRIPTION AND USER'S MANUAL FOR GAL
00340 C
00350 C  DELSIG= CONSTANT CALCULATED IN PROGRAM VISCOUS.
00360 C  SS    = CONSTANT CALCULATED IN PROGRAM VISCOUS.
00370 C   H    = STILL WATER DEPTH
00380 C  FMASS = CONSTANT CALCULATED IN PROGRAM VISCOUS.
00390 C  B     = CONSTANT CALCULATED IN PROGRAM VISCOUS.
00400 C  E     = CONSTANT CALCULATED IN PROGRAM VISCOUS.
00410 C  D1    = CONSTANT CALCULATED IN PROGRAM VISCOUS.
00420 C  CONNV = LOGICAL VARIABLE INDICATING WHETHER THE VERTICAL EDDY VISCOSITY
00430 C                  IS TO BE ASSUMED CONSTANT.
00440 C  LAND  = LOGICAL VARIABLE INDICATED WHETHER AN ELEMENT IS LAND OR WATER.
00450 C  F     = CORIOLIS PARAMTER
00460 C  C     = UNDETERMINED PARAMETER IN THE X DIRECTION.
```

```
00470 C  D      = UNDETERMINED PARAMETER IN THE Y DIRECTION.
00480 C  ETA    = SURFACE ELEVATION ABOVE STILL WATER LEVEL.
00490 C  UB     = MASS FLUXES IN THE X DIRECTION  'U-BAR'.
00500 C  VB     = MASS FLUXES IN THE Y DIRECTION, 'V-BAR'.
00510 C  LP     = NO. OF GRID ELEMENTS IN THE X DIRECTION.
00520 C  MP     = NO. OF GRID ELEMENTS IN THE Y DIRECTION.
00530 C  JP     = NO. OF COSINE TERMS USED IN BASIS FCN.
00540 C  KP     = NO. OF LINEAR SEGMENTS USED FOR THE VERTICAL EDDY VISCOSITY
00550 C  EDDYH  = THE HORIZNTAL EDDY VISCOSITY PARAMETER.
00560 C  VERSIN= CONTAINS REVISION LABEL WHICH IS PRINTED OUT
00570 C  PTIME  = PROTOTYPE TIME IN SECONDS
00580 C  AMP    = THE SPECIFIED AMPLITUDE AT EACH ELEMENT.
00590 C  IAMP   = SPECIFIES THE TYPE OF BOUNDARY CONDITION TO BE APPLIED AT THE EL.
00600 C  TITCIR= TITLE OF THE CIRC RUN.
00610 C  CBC    = ARRAY CONTAINING OLD VALUES OF C.  USED PRIMARILY FOR BOUNDARY C'S.
00620 C  DBC    = ARRAY CONTAINING OLD VALUES OF D.  USED PRIMARILY FOR BOUNDARY C'S.
00630 C  HOTIME= THE TIME FOR A HOT START.
00640 C  PRTFLO= LOGICAL VARIABLE INDICATING WHETHER MASS FLUXES ARE TO BE PRINTED.
00650 C  ISTPR = STARTING TIME FOR PRINTER AND PLOTTED OUTPUT.
00660 C  DTT    = TIME STEP.
00670 C  DL     = ELEMENT SIZE.
00680 C  TITWND= TITLE OF THE WIND RUN.
00690 C  TITVIS= TITLE OF PROGRAM VISCOUS RUN.
00700 C  TITDEN= TITLE OF PROGRAM DENSITY RUN.
00710 C  PHI    = LATITUDE
00720 C  WLNGTH= THE WAVE LENGTH OF THE OSCILLATION ON THE WESTERN BOUNDARY.
00730 C  WPEROD= THE WAVE PERIOD OF THE OSCILLATION ON THE WESTERN BOUNDARY.
00740 C  UA2    = THE FRICTION VELOCITY SQUARED IN THE X DIRECTION.
00750 C  VA2    = THE FRICTION VELOCITY SQUARED IN THE Y DIRECTION.
00760 C  FX     = CONSTANT ASSOCIATED WITH THE WIND SHEAR STRESS IN THE X-DIRECTION.
00770 C  FY     = CONSTANT ASSOCIATED WITH THE WIND SHEAR STRESS IN THE Y-DIRECTION.
00780 C  TAUX   = CONSTANT ASSOCIATED WITH THE WIND SHEAR STRESS IN THE X-DIRECTION.
00790 C  TAUY   = CONSTANT ASSOCIATED WITH THE WIND SHEAR STRESS IN THE Y-DIRECTION.
00800 C  ROWX   = CONSTANT READ FROM PROGRAM DENSITY.  PROPORTIONAL TO THE DENSITY
00810 C           GRADIENT IN THE X-DIRECTION.
00820 C  ROWY   = CONSTANT READ FROM PROGRAM DENSITY.  PROPORTIONAL TO THE DENSITY
00830 C           GRADIENT IN THE Y-DIRECTION.
00840 C  RHOW   = DENSITY OF SEA WATER IN THE SURFACE LAYER.  READ FROM PROG. DENSITY
00850 C  IPP    = NO. OF DENSITY LAYERS.  READ FROM PROG. DENSITY.
00860 C  DENSE  = LOGICAL VARIABLE INDICATING WHETHER DENSITY GRADIENT IS TO BE
00870 C           CONSIDERED.
00880 C  DTIME  = TIME OF LAST DENSITY FIELD CHANGE.
00890 C  CK     = THE LAYER DEPTH OF THE VERTICAL EDDY VISCOSITY.
00900 C  ALPHA  = THE SLOPE OF THE LINEAR VERTICAL EDDY VISCOSITY SEGEMENT.  READ
00910 C           FROM PROG. VISCOUS.
00920 C  BETA   = THE INTERCEPT OF THE LINEAR VERTICAL EDDY VISCOSITY SEGEMENT.
00930 C           READ FROM PROG. VISCOUS.
00940 C  NB     = THE VALUE OF THE VERTICAL EDDY VISCOSITY AT THE BOTTOM
```

```
00950 C  A      = THE PERIOD OF THE BASIS FUNCTION
00960 C  CB     = THE BOTTOM FRICTION COEFFICIENT.
00970 C  EDDYV  = THE VERTICAL EDDY VISCOSITY.
00980 C
00990        COMMON/EDDE/DELSIG(2592),SS(864),H(288),FMASS(288),
01000       &   B(864),E(864),D1(864),CONNV,LAND(288),F(288)
01010        COMMON/UNKNOW/C(864),D(864),ETA(288),UB(13,25),VB(13,25)
01020        COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
01030       &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
01040       &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
01050       &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
01060        COMMON/WINDY/UA2(288),VA2(288),FX(288),FY(288),TAUX(864),TAUY(864)
01070        COMMON/DEN/ROWX(864),ROWY(864),RHOW(288),IPP,DENSE,DTIME
01080        COMMON/VISC/CK(1552),ALPHA(865),BETA(865),NB(288),A(864),
01090       &   CB(288),EDDYV(1552)
01100 C
01110        INTEGER IAMP*2,AMP*2
01120        LOGICAL NVCHG,WDCHG RLID,DENSE,DENCHG,CONNV,HOTSTR,
01130       &        LATRAL,PRTFLO,LAND*1
01140        REAL NB
01150 C
01160        NAMELIST/MISC/EDDYH,DENSE
01170        NAMELIST/TIME/TLAST,DTT,ISTPR,IDTPR
01180        NAMELIST/BCS/PERIOD,AMP,IAMP,UB,VB,RLID,CBC,DBC,WLNGTH,WPEROD
01190       &            ,WAMP
01200        NAMELIST/INOUT/PRTFLO,HOTIME,HOTSTR
01210 C
01220        DATA NVCHG/.TRUE./,WDCHG/.TRUE./,ETIME/0.0/,DENCHG/.FALSE./,
01230       &     COUNT/0.0/,WTIME/0.0/,PERIOD/45400.0/,RLID/.FALSE./,
01240       &     LATRAL/.FALSE./,GRAV/9.8/,IDTPR/900000000/
01250 C
01260 C  READ IN NECESSARY PARAMETERS USING NAMELIST AND INITIALIZE OTHER CONSTANTS
01270        READ(1,100) TITCIR
01280  100   FORMAT(20A4)
01290        READ(10) TITVIS,LP,MP,JP,KP,CONNV,DUM,DUM,DUM,DUM,DL,PHI
01300        LPMP=LP*MP
01310        KPLPMP=KP*LPMP
01320        LPM1=LP-1
01330        MPM1=MP-1
01340        LPP1=LP+1
01350        MPP1=MP+1
01360        JPLPMP=JP*LPMP
01370        JPLP=JP*LP
01380        KPM1=KP-1
01390        READ(10) (H(LM),LM=1,LPMP),(LAND(LM),LM=1,LPMP),
01400       &(DUM,LM=1,LPMP),(CK(KLM),KLM=1,KPLPMP),(F(LM),LM=1,LPMP)
01410        READ(1,MISC)
01420        IF(EDDYH.GT.0.01) LATRAL=.TRUE.
```

```
01430       READ(1,TIME)
01440       CALL SETBCS
01450       READ(1,BCS)
01460       READ(1,INOUT)
01470       READ(4) TITWND
01480       IF(DENSE) READ(11) TITDEN
01490       IF(HOTSTR) CALL HOT
01500 C
01510 C  INITIALIZE VARIOUS CONSTANTS.
01520       DO 200 LM=1,LPMP
01530       F(LM)=DTT*F(LM)
01540 200   CONTINUE
01550       C2=DTT/DL
01560       C4=2.0*DTT
01570       C5=2.0*C2*GRAV
01580       C7=GRAV/(2.0*DL)
01590 C     THE ATMOSPHERIC PRESSURE GRADIENTS ARE MULTIPLIED BY C8.
01600 C     C8 IS MULTIPLIED BY 0.1 TO CONVERT FROM MB/KM TO KG/SQ
01610 C     MT/SQ SEC.
01620       C8=DL/GRAV*0.1
01630       RTIME=ITIME
01640       MAXIT=INT((TLAST-RTIME)/DTT)
01650       EH=EDDYH*DTT/(DL*DL)
01660       CALL HEDER(PERIOD,RLID,TLAST,IDTPR)
01670       CALL BCFLOW
01680 C
01690 C  START MAIN TIME LOOP
01700       DO 999 IT=1,MAXIT
01710       T=FLOAT(IT)*DTT+RTIME
01720       ITIME=INT(T)
01730       IF(WTIME.LT.T) CALL RDWIND(WTIME,WDCHG)
01740       IF(ETIME.LT.T) CALL RDNV(NVCHG,ETIME)
01750       IF(DENSE.AND.DTIME.LT.T) CALL RDROW (DENCHG,C7)
01760       IF(WDCHG.OR.NVCHG.OR.DENCHG) CALL CHANGE(C8,WDCHG,NVCHG,DENCHG)
01770       CALL NEWCD(LATRAL)
01780       IF(.NOT.RLID) CALL NEWETA
01790       CALL BNDRY(T,PERIOD)
01800       COUNT=COUNT+DTT
01810       IF(ITIME.GE.ISTPR.AND.COUNT.GE.IDTPR) CALL PRINTS(COUNT)
01820 999   CONTINUE
01830       STOP
01840       END
01850 C
01860 C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01870       BLOCK DATA
01880       COMMON/EDDE/DELSIG(2592),SS(864),H(288),FMASS(288),
01890      &   B(864),E(864),D1(864),CONNV,LAND(288),F(288)
01900       COMMON/UNKNOW/C(864),D(864),ETA(288),UB(13,25),VB(13,25)
```

```
01910       COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
01920      &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
01930      &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
01940      &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
01950       COMMON/WINDY/UA2(288),VA2(288),FX(288),FY(288),TAUX(864),TAUY(864)
01960       COMMON/DEN/ROWX(864),ROWY(864),RHOW(288),IPP,DENSE,DTIME
01970       COMMON/VISC/CK(1552),ALPHA(865),BETA(865),NB(288),A(864),
01980      &   CB(288),EDDYV(1552)
01990 C
02000       LOGICAL NVCHG,WDCHG,DENSE,DENCHG,CONNV,PRTFLO,LAND*1
02010       INTEGER IAMP*2,AMP*2
02020       REAL NB
02030 C
02040       DATA DELSIG/2592*0.0/
02050       DATA ALPHA,BETA/865*0.0,865*0.0/
02060       DATA CK,EDDYV/1552*0.0,1552*0.0/
02070       DATA UB,VB/325*0.0,325*0.0/
02080       DATA CBC,DBC/975*0.0,975*0.0/
02090       DATA FX,FY,UA2,VA2,FMASS,H,ETA,AMP,IAMP,RHOW,LAND,F,NB,CB/
02100      &    288*0.0,288*0.0,288*0.0,288*0.0,288*0.0,
02110      &    288*0.0,288*0.0,288*0,288*4,288*1022.85,
02120      &    288*.FALSE.,288*0.0,288*0.0,288*0.0/
02130       DATA B,E,D1,SS,TAUX,TAUY,C,D,ROWX,ROWY,A/
02140      &     864*0.,864*0.,864*0.,864*0.,864*0.,864*0.0,864*0.0,
02150      &     864*0.,864*0.0,864*0.0,864*0.0/
02160       DATA VERSIN/'GAL ','- VE','R. 7','.A -',' JUL','Y 19',
02170      &     '82  ','PROG','RAM ','CIRC',4*'    '/
02180       DATA TITCIR,TITDEN,TITWND,TITVIS/80*'    '/
02190       DATA PRTFLO/.FALSE./,DENSE/.FALSE./,DTIME/0.0/,
02200      &     CONNV/.TRUE./,ITIME/0/,ISTPR/9000000/
02210       DATA EDDYH,WLNGTH,WPEROD,WAMP/1.E+05,3*0./
02220       END
02230 C
02240 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02250 C
02260       SUBROUTINE NEWCD(LATRAL)
02270 C
02280 C
02290 C   ROUTINE WHICH UPDATES C(JLM) AND D(JLM) FOR EACH ELEMENT.
02300 C
02310       COMMON/EDDE/DELSIG(2592),SS(864),H(288),FMASS(288),
02320      &   B(864),E(864),D1(864),CONNV,LAND(288),F(288)
02330       COMMON/UNKNOW/C(864),D(864),ETA(288),UB(13,25),VB(13,25)
02340       COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
02350      &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
02360      &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
02370      &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
02380       COMMON/WINDY/UA2(288),VA2(288),FX(288),FY(288),TAUX(864),TAUY(864)
```

```
02390       COMMON/DEN/ROWX(864),ROWY(864),RHOW(288),IPP,DENSE,DTIME
02400 C
02410       INTEGER IAMP*2,AMP*2
02420       LOGICAL HOTSTR,PRTFLO,LAND*1,CONNV,DENSE,LATRAL
02430       DATA EDDH/0.0/
02440 C
02450       DO 900 M=2,MP
02460       LMP=(M-1)*LP
02470       DO 900 L=2,LP
02480       LM=L+LMP
02490       JLMP=(LM-1)*JP
02500 C  CHECKS TO SEE IF LAND.  IF SO NO NEED TO UPDATE C(J)'S AND D(J)'S.
02510          IF(LAND(LM)) GO TO 900
02520 C  CHECKS TO SEE IF C IS SPECIFIED AT GRID.  IF SO DOES NOT UPDATE C.
02530          IF(IAMP(L,M).GE.5) GOTO 900
02540 C  CHECKS TO SEE IF NO-FLOW BOUNDARY IN X DIRECTION.  IF SO NO NEED TO
02550 C  UPDATE C(J)'S FOR THE GRID.
02560          IF(LAND(LM-1).OR.IAMP(L-1,M).EQ.6) GO TO 500
02570 C        CHECKS TO SEE IF M.EQ.MP.  IF SO C IS NOT UPDATED BUT RATHER SET
02580 C        TO C AT MP-1.
02590          IF(M.EQ.MP) GOTO 500
02600 C        LOOP WHICH UPDATES C(J)'S FOR GRID L,M.
02610          SUM=0.0
02620          ETAX=ETA(LM)-ETA(LM-1)
02630          JJLM=1+JLMP
02640          DO 100 J=1,JP
02650          JLM=J+JLMP
02660          IF(LATRAL) EDDH=(CBC(J,L-1,M)+CBC(J,L+1,M)-4.0*CBC(J,L,M)+
02670         &           CBC(J,L,M-1)+CBC(J,L,M+1))*EH
02680          DBAR=D(JLM)
02690          IF(LAND(LM-LP)) DBAR=D(JLM+JPLP)
02700          CONSTS=-B(JLM)*ETAX+F(LM)*DBAR+ROWX(JLM)+TAUX(JLM)+EDDH
02710          IF(.NOT.CONNV) C(JLM)=CJDJ(C(JJLM),JP,J,JLM,CONSTS)
02720          IF(CONNV) C(JLM)=C(JLM)*DELSIG(JLM)+CONSTS
02730          SUM=SUM+C(JLM)*SS(JLM)
02740 100      CONTINUE
02750 C        UPDATES MASS FLUX IN X DIRECTION FOR GRID L,M.
02760          UB(L,M)=FX(LM)+H(LM)*SUM
02770 C
02780 C        CHECKS FOR NO-FLOW BOUNDARY IN Y DIRECTION.  IF SO  NO NEED
02790 C        TO UPDATE D(JLM) SO PROGRAM SKIPS TO NEXT GRID.
02800 500      IF(LAND(LM-LP).OR.IAMP(L,M-1).EQ.6) GOTO 900
02810 C        LOOP WHICH UPDATES D(J)'S FOR GRID L,M.
02820          ETAY=ETA(LM)-ETA(LM-LP)
02830          JJLM=1+JLMP
02840          SUM=0.0
02850          DO 200 J=1,JP
02860          JLM=J+JLMP
```

```
02870       IF(LATRAL) EDDH=(DBC(J,L-1,M)+DBC(J,L+1,M)-4.0*DBC(J,L,M)+
02880      &           DBC(J,L,M-1)+DBC(J,L,M+1))*EH
02890       CBAR=CBC(J,L,M)
02900       IF(LAND(LM-1)) CBAR=CBC(J,L+1,M)
02910       CONSTS=-B(JLM)*ETAY-F(LM)*CBAR+ROWY(JLM)+TAUY(JLM)+EDDH
02920       IF(.NOT.CONNV) D(JLM)=CJDJ(D(JJLM),JP,J,JLM,CONSTS)
02930       IF(CONNV) D(JLM)=D(JLM)*DELSIG(JLM)+CONSTS
02940       SUM=SUM+D(JLM)*SS(JLM)
02950 200   CONTINUE
02960 C     UPDATES MASS FLUX IN Y DIRECTION FOR GRID L,M.
02970       VB(L,M)=FY(LM)+H(LM)*SUM
02980 900   CONTINUE
02990       RETURN
03000       END
03010 C
03020 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
03030 C
03040       FUNCTION CJDJ(DUM,JP,J,JLM,CONSTS)
03050 C
03060 C     ROUTINE WHICH CALCULATES THE SUMMATION OF DELSIG(I,J,L,M) FOR
03070 C     I TO IP FOR THE CASE WHERE NV=NV(Z).
03080 C
03090       COMMON/EDDE/DELSIG(2592),SS(864),H(288),FMASS(288),
03100      &   B(864),E(864),D1(864),CONNV,LAND(288),F(288)
03110       DIMENSION DUM(JP)
03120 C
03130       LOGICAL CONNV,PRTFLO,LAND*1
03140 C
03150       IJLMP=(JLM-1)*JP
03160       PS=0.0
03170       DO 100 I=1,JP
03180       IJLM=I+IJLMP
03190       PS=PS+DUM(I)*DELSIG(IJLM)
03200 100   CONTINUE
03210       CJDJ=DUM(J)+CONSTS+PS
03220       RETURN
03230       END
03240 C
03250 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
03260 C
03270       SUBROUTINE NEWETA
03280 C
03290 C
03300 C     ROUTINE WHICH UPDATES SURFACE ELEVATION, ETA.
03310 C
03320       COMMON/EDDE/DELSIG(2592),SS(864),H(288),FMASS(288),
03330      &   B(864),E(864),D1(864),CONNV,LAND(288),F(288)
03340       COMMON/UNKNOW/C(864),D(864),ETA(288),UB(13,25),VB(13,25)
```

```
      COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
     &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
     &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
     &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
C
      INTEGER IAMP*2,AMP*2
      LOGICAL CONNV,PRTFLO,LAND*1
C
      DO 100 M=2,MPM1
      LMP=(M-1)*LP
      DO 100 L=2,LPM1
      LM=L+LMP
      IF(LAND(LM)) GO TO 100
      ETA(LM)=FCNETA(L,M,LM,C2)
100   CONTINUE
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      FUNCTION FCNETA(L,M,LM,C2)
C
C     ROUTINE WHICH CALCULATES NEW ETA IN TERMS OF OLD ETAS
C     AND OLD FLOWS.
C
      COMMON/UNKNOW/C(864),D(864),ETA(288),UB(13,25),VB(13,25)
C
      FCNETA=ETA(LM)+C2*(UB(L,M)-UB(L+1,M)+VB(L,M)-VB(L,M+1))
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE RDROW(DENCHG,C7)
C
C     ROUTINE WHICH READS IN DENSITY GRADIENT TERMS CREATED BY PROGRAM DENSE.
C
      COMMON/EDDE/DELSIG(2592),SS(864),H(288),FMASS(288),
     &   B(864),E(864),D1(864),CONNV,LAND(288),F(288)
      COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
     &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
     &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
     &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
      COMMON/DEN/ROWX(864),ROWY(864),RHOW(288),IPP,DENSE,DTIME
C
      LOGICAL DENCHG,DENSE,CONNV,PRTFLO,LAND*1
      INTEGER IAMP*2,AMP*2
C
```

```
03830        DENCHG=.TRUE.
03840        CONST1=-C4*C7
03850 50     READ(11) DTIME,IPP,(ROWX(JLM),JLM=1,JPLPMP),
03860       &         (ROWY(JLM),JLM=1,JPLPMP),(RHOW(LM),LM=1,LPMP)
03870        IF(DTIME.LT.ITIME) GOTO 50
03880        DO 150 LM=1,LPMP
03890        JLMP=(LM-1)*JP
03900        IF(LAND(LM)) GO TO 150
03910        CONST2=CONST1/H(LM)
03920        DO 100 J=1,JP
03930        JLM=J+JLMP
03940        ROWX(JLM)=CONST2*ROWX(JLM)
03950        ROWY(JLM)=CONST2*ROWY(JLM)
03950 100    CONTINUE
03970 150    CONTINUE
03980        RETURN
03990        END
04000 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
04010 C
04020        SUBROUTINE RDNV(NVCHG,ETIME)
04030 C
04040 C
04050 C      ROUTINE WHICH READS IN CONSTANTS ASSOCIATED WITH VERTICAL
04060 C          EDDY VISCOSITY AND CORIOLIS PARAMTER.  CONSTANTS WERE GENERATED
04070 C      BY PROGRAM VISCOUS.
04080 C
04090        COMMON/EDDE/DELSIG(2592),SS(864),H(288),FMASS(288),
04100       &   B(864),E(864),D1(864),CONNV,LAND(288),F(288)
04110        COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
04120       &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
04130       &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
04140       &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
04150        COMMON/VISC/CK(1552),ALPHA(865),BETA(865),
04160       &       NB(288),A(864),CB(288),EDDYV(1552)
04170 C
04180        INTEGER IAMP*2,AMP*2
04190        LOGICAL CONNV,NVCHG,PRTFLO,LAND*1
04200        REAL NB
04210 C
04220        JJLPMP=JP*JPLPMP
04230        KPLPMP=KP*LPMP
04240        KLMM1=LPMP*(KP-1)
04250        NVCHG=.TRUE.
04260        IF(CONNV) JJLPMP=JPLPMP
04270 10     READ(10) ETIME,(A(JLM),JLM=1,JPLPMP),(DUM,JLM=1,JPLPMP),
04280       &         (B(JLM),JLM=1,JPLPMP),(DELSIG(JJLM),JJLM=1,JJLPMP),
04290       &         (D1(JLM),JLM=1,JPLPMP),(E(JLM),JLM=1,JPLPMP),
04300       &         (SS(JLM),JLM=1,JPLPMP),(FMASS(LM),LM=1,LPMP),
```

```
     &         (EDDYV(KLM),KLM=1,KPLPMP),(NB(LM),LM=1,LPMP),
     &         (ALPHA(KLM),KLM=1,KLMM1),(BETA(KLM),KLM=1,KLMM1)
      IF(ETIME.LT.ITIME) GOTO 10
      DO 100 M=1,MP
      LMP=(M-1)*LP
      DO 100 L=1,LP
      LM=L+LMP
      JLMP=(LM-1)*JP
      IF (LAND(LM)) GO TO 100
      C1=C4/H(LM)
      DO 50 J=1,JP
      JLM=J+JLMP
      D1(JLM)=C4*D1(JLM)
      B(JLM)=C5*B(JLM)
      E(JLM)=F(LM)*2.*E(JLM)
      IF(.NOT.CONNV) GOTO 25
      DELSIG(JLM)=DELSIG(JLM)*C1+1.0
      GOTO 50
 25   IJLMP=(JLM-1)*JP
      DO 30 I=1,JP
      IJLM=I+IJLMP
      DELSIG(IJLM)=DELSIG(IJLM)*C1
 30   CONTINUE
 50   CONTINUE
 100  CONTINUE
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE RDWIND(WTIME,WDCHG)
C
C     ROUTINE WHICH READS IN THE WIND FIELD EACH TIME IT CHANGES.
C     WIND FIELD CAN BE GENERATED BY A NUMBER OF PROGRAMS INCLUDING
C     PROGRAM WIND.
C
      COMMON/CON/JP,KP,LP,MP.KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
     &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
     &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
     &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
      COMMON/WINDY/UA2(288),VA2(288),FX(288),FY(288),TAUX(864),TAUY(864)
C
      LOGICAL WDCHG,PRTFLO
      INTEGER IAMP*2,AMP*2
C
      WDCHG=.TRUE.
 100  READ(4) WTIME,(UA2(LM),LM=1,LPMP),(VA2(LM),LM=1,LPMP)
      WTIME=WTIME*3600.
```

```
04790       IF(WTIME.LT.ITIME) GOTO 100
04800       RETURN
04810       END
04820 C
04830 C
04840 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
04850 C
04860       SUBROUTINE CHANGE(C8,WDCHG,NVCHG,DENCHG)
04870 C
04880 C
04890 C   ROUTINE WHICH CHANGES VARIOUS CONSTANTS.  IT IS NECESSARY TO UPDATE
04900 C   THESE CONSTANTS WHENEVER THE WIND,EDDY VISCOSITY OR DENSITY CHANGES.
04910 C
04920       COMMON/EDDE/DELSIG(2592),SS(864),H(288),FMASS(288),
04930      &   B(864),E(864),D1(864),CONNV,LAND(288),F(288)
04940       COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
04950      &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
04960      &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
04970      &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
04980       COMMON/WINDY/UA2(288),VA2(288),FX(288),FY(288),TAUX(864),TAUY(864)
04990       COMMON/DEN/ROWX(864),ROWY(864),RHOW(288),IPP,DENSE,DTIME
05000       INTEGER IAMP*2,AMP*2
05010       LOGICAL CONNV,WDCHG,NVCHG,DENCHG,PRTFLO,LAND*1
05020 C
05030       DENCHG=.FALSE.
05040       NVCHG=.FALSE.
05050       WDCHG=.FALSE.
05060       DO 100 M=1,MP
05070       LMP=(M-1)*LP
05080       DO 100 L=1,LP
05090       LM=L+LMP
05100       IF (LAND(LM)) GO TO 100
05110       JLMP=(LM-1)*JP
05120       FX(LM)=UA2(LM)*FMASS(LM)
05130       FY(LM)=VA2(LM)*FMASS(LM)
05140       DO 50 J=1,JP
05150       JLM=J+JLMP
05160       TAUX(JLM)=D1(JLM)*UA2(LM)+E(JLM)*VA2(LM)
05170       TAUY(JLM)=D1(JLM)*VA2(LM)-E(JLM)*UA2(LM)
05180 50    CONTINUE
05190 100   CONTINUE
05200       RETURN
05210       END
05220 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
05230 C
05240       SUBROUTINE BCFLOW
05250 C
05260 C     ROUTINE WHICH SETS APPROPRIATE MASS FLUXES WHEN RIVER INFLOW OR
```

```
05270 C     OUTFLOW OPTIONS HAS BEEN SPECIFIED, I.E. IAMP=6.
05280 C
05290       COMMON/UNKNOW/C(864),D(864),ETA(288),UB(13,25),VB(13,25)
05300       COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
05310      &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
05320      &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
05330      &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
05340       INTEGER IAMP*2,AMP*2
05350       LOGICAL PRTFLO
05360 C
05370       DO 100 L=1,LPM1
05380       M=1
05390       IF(IAMP(L,M).NE.6)GOTO 100
05400       UB(L,M+1)=UB(L,M)
05410       VB(L,M+1)=VB(L,M)
05420 100   CONTINUE
05430       DO 200 M=1,MPM1
05440       L=1
05450       IF(IAMP(L,M).NE.6)GOTO 200
05460       UB(L+1,M)=UB(L,M)
05470       VB(L+1,M)=VB(L,M)
05480 200   CONTINUE
05490       RETURN
05500       END
05510 C
05520 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
05530 C
05540 C
05550       SUBROUTINE SETBCS
05560 C
05570 C
05580 C     ROUTINE WHICH INITIALLY SETS THE ELEMENT TYPE (IAMP).
05590 C     SEE USER'S MANUAL FOR MORE COMPLETE DEFINITIONS.
05600 C
05610 C
05620       COMMON/EDDE/DELSIG(2592),SS(864),H(288),FMASS(288),
05630      &   B(864),E(864),D1(864),CONNV,LAND(288),F(288)
05640       COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
05650      &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
05660      &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
05670      &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
05680 C
05690       INTEGER IAMP*2,AMP*2
05700       LOGICAL CONNV,PRTFLO,LAND*1
05710 C
05720       DO 100 M=1,MP
05730       LMP=(M-1)*LP
05740       DO 100 L=1,LP
```

```
05750       LM=L+LMP
05760       IF(L.EQ.1.OR.L.EQ.LP.OR.M.EQ.1.OR.M.EQ.MP) IAMP(L,M)=1
05770       IF(H(LM).GT.0.01) GO TO 100
05780       IAMP(L,M)=0
05790 100   CONTINUE
05800       RETURN
05810       END
05820 C
05830 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
05840 C
05850       SUBROUTINE BNDRY(T,PERIOD)
05860 C
05870 C
05880 C     ROUTINE WHICH SETS THE BOUNDARY CONDITIONS AT OPEN WATER BNDRYS.
05890 C     MODIFIED 1/8/82 TO SPECIFICALLY FIT THE WEST FLORIDA SHELF.
05900 C
05910       COMMON/EDDE/DELSIG(2592),SS(864),H(288),FMASS(288),
05920      &   B(864),E(864),D1(864),CONNV,LAND(288),F(288)
05930       COMMON/UNKNOW/C(864),D(864),ETA(288),UB(13,25),VB(13,25)
05940       COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
05950      &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
05960      &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
05970      &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
05980 C
05990       INTEGER IAMP*2,AMP*2
06000       LOGICAL HOTSTR,PRTFLO,LAND*1,CONNV
06010 C
06020       FACTOR=(1-COS(T*6.28/PERIOD))*.1
06030       DO 100 L=2,LP
06040       M=1
06050       JLM=(L-1)*JP+1
06060 C  SET SOUTHERN BOUNDARY:  D(ETA)/DY=0, IF NOT LAND THEN SET SYMMETRY
06070 C  ON C, SYMMETRY ON D.  IF LAND THEN SET D=0, ANTISYMMETRY ON C.
06080       ETA(L)=ETA(L+LP)
06090       UB(L,1)=UB(L,2)
06100       VB(L,1)=VB(L,3)
06110       CALL SETCD(JP,1.,CBC(1,L,1),C(JLM+JPLP))
06120       CALL SETCD(JP,1.,DBC(1,L,1),D(JLM+2*JPLP))
06130       IF(.NOT.LAND(L)) GOTO 50
06140       ETA(L)=0.0
06150       CALL SETCD(JP,-1.,CBC(1,L,1),C(JLM+JPLP))
06160       CALL SETCD(JP, 0.,DBC(1,L,M),D(JLM))
06170       VB(L,M)=0.0
06180 C  SET NORTHERN BOUNDARY - SAME AS SOUTHERN.
06190 50    M=MP
06200       LMP=(M-1)*LP
06210       LM=L+LMP
06220       JLM=(LM-1)*JP+1
```

```
06230 C       ETA(LM)=0.0
06240         ETA(LM)=ETA(LM-LP)
06250         UB(L,MP)=UB(L,MP-1)
06260         VB(L,MPP1)=VB(L,MP-1)
06270         CALL SETCD(JP,1.,DBC(1,L,MPP1),D(JLM-JPLP))
06280         CALL SETCD(JP,1.,CBC(1,L,MP),C(JLM-JPLP))
06290         CALL SETCD(JP,1.,DBC(1,L,MP),D(JLM))
06300         CALL SETCD(JP,1.,C(JLM),C(JLM-JPLP))
06310         IF(.NOT.LAND(LM)) GOTO 100
06320         ETA(LM)=0.0
06330         VB(L,MP)=0.0
06340         CALL SETCD(JP,-1.,CBC(1,L,MP),C(JLM-JPLP))
06350         CALL SETCD(JP,0.,DBC(1,L,MP),D(JLM))
06360 100     CONTINUE
06370         DO 200 M=1,MP
06380         LMP=(M-1)*LP
06390 C   SET WESTERN OPEN OCEAN BOUNDARY: SET ETA=0, REFLECTIONAL SYMMETRY ON C,
06400 C   AND IF D IS NOT SPECIFIED(I.E. IAMP.NE.5) REFLECTIONAL SYMMETRY ON D.
06410         L=1
06420         LM=L+LMP
06430         JLM=(LM-1)*JP+1
06440         ETA(LM)=0.0
06450         IF(IAMP(L,M).NE.5) CALL SETCD(JP,1.,DBC(1,1,M),D(JLM+JP))
06460         CALL SETCD(JP,1.,CBC(1,1,M),C(JLM+JP))
06470 C     NEXT STATEMENTS SET SINUSOIDAL CONDITION AT WESTERN BNDRY.
06480         IF(IAMP(L,M).NE.5) GOTO 200
06490         IF(WLNGTH.LE.0.01.OR.WPEROD.LE.0.01) GOTO 200
06500         DBC(1,1,M)=WAMP*SIN(6.28/WLNGTH*M*DL-6.28/WPEROD*ITIME)
06510 200     CONTINUE
06520 C   SET CBC & DBC ELSEWHERE IN THE GRID.  FOR LAND
06530 C   ELEMENTS, USE NEGATIVE REFLECTIONAL SYMMETRY AND ZERO ORTHOGONAL FLOW.
06540         DO 300 M=2,MPM1
06550         LMP=(M-1)*LP
06560         DO 300 L=2,LP
06570         LM=LMP+L
06580         JLM=(LM-1)*JP+1
06590         CALL SETCD(JP,1.,CBC(1,L,M),C(JLM))
06600         CALL SETCD(JP,1.,DBC(1,L,M),D(JLM))
06610         IF(.NOT.LAND(LM)) GOTO 300
06620         CALL SETCD(JP, 0.,CBC(1,L,M),C(JLM))
06630         IF(LAND(LM-LP)) CALL SETCD(JP,-1.,DBC(1,L,M),D(JLM-JP))
06640         IF(LAND(LM-LP)) VB(L,M)=-VB(L-1,M)
06650         UB(L,M)=0.0
06660 300     CONTINUE
06670 C       SET DBC AND CBC AT SOUTHWESTERN BOUNDARY.
06680         IF(IAMP(1,1).EQ.5) GOTO 999
06690         CALL SETCD(JP,1.0,CBC(1,1,1),C(JPLP+1))
06700         CALL SETCD(JP,1.0,DBC(1,1,1),D(JPLP+1))
```

```
06710 999    RETURN
06720        END
06730 C
06740 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
06750 C
06760        SUBROUTINE SETCD(JP,SIGNN,X,Y)
06770 C
06780 C      ROUTINE WHICH SETS THE VARIABLES CBC AND DBC EQUAL TO THE
06790 C      APPRORIATE C(J)'S AND D(J)'S.
06800 C
06810        DIMENSION X(20),Y(20)
06820 C
06830        DO 100 J=1,JP
06840        X(J)=Y(J)*SIGNN
06850 100    CONTINUE
06860        RETURN
06870        END
06880 C
06900 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
06910 C
06920 C
06930        SUBROUTINE PRINTS(COUNT)
06940 C
06950 C
06960 C  ROUTINE WHICH PRINTS PARAMETERS CHOSEN BY USER.
06970 C
06980        COMMON/UNKNOW/C(864),D(864),ETA(288),UB(13,25),VB(13,25)
06990        COMMON/EDDE/DELSIG(2592),SS(864),H(288),FMASS(288),
07000       &   B(864),E(864),D1(864),CONNV,LAND(288),F(288)
07010        COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
07020       &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
07030       &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
07040       &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
07050        COMMON/WINDY/UA2(288),VA2(288),FX(288),FY(288),TAUX(864),TAUY(864)
07060 C
07070        LOGICAL PRTFLO,LAND*1
07080        INTEGER AMP*2,IAMP*2
07090 C
07100        COUNT=0.0
07110        TIME=FLOAT(ITIME)/3600.
07120        WRITE(3,50) (ASTER,I=1,30)
07130 50     FORMAT ('1',//30A4)
07140        WRITE(3,100) TIME
07150 100    FORMAT(//5X,21(1H-)/4X,'/ TIME=',
07160       &F6.2,' HOURS',2X,'/',/,4X,21(1H-))
07190        CALL PRINT1(6,12)
07200        DO 600 M=1,MP
07210        LMP=(M-1)*LP
```

```
07220       DO 600 L=1,LP
07230       LM=LMP+L
07240       IF(.NOT.LAND(LM)) GOTO 600
07250       UB(L,M)=0.0
07260       VB(L,M)=0.0
07270       JLMP=(LM-1)*JP
07280       DO 500 J=1,JP
07290       JLM=JLMP+J
07300       C(JLM)=0.0
07310       D(JLM)=0.0
07320 500   CONTINUE
07330 600   CONTINUE
07340       IF(PRTFLO) WRITE(12) ITIME,((UB(L,M),L=1,LP),M=1,MP),
07350      &                          ((VB(L,M),L=1,LP),M=1,MP)
07360       WRITE(12) ITIME,(((CBC(J,L,M),J=1,JP),L=1,LPP1),M=1,MPP1),
07370      & (((DBC(J,L,M),J=1,JP),L=1,LPP1),M=1,MPP1),(ETA(LM),LM=1,LMP)
07380       WRITE(3,50)(ASTER,I=1,30)
07390       RETURN
07400       END
07410 C
07420 C * * * * * * * * * * * * * * * * * * * * * * * * * * * *
07430 C
07440       SUBROUTINE PRINT1(L,M)
07450 C
07460 C     ROUTINE WHICH LISTS VARIOUS COFFICIENTS FOR ELEMENT L,M.
07470 C
07480       COMMON/UNKNOW/C(864),D(864),ETA(288),UB(13,25),VB(13,25)
07490       COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
07500      &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
07510      &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
07520      &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
07530       COMMON/WINDY/UA2(288),VA2(288),FX(288),FY(288),TAUX(864),TAUY(864)
07540 C
07550       INTEGER IAMP*2,AMP*2
07560       LM=(M-1)*LP+L
07570       JLM=(LM-1)*JP
07580       WRITE(3,150) L,M
07590 150   FORMAT(//,45X,' SUMMARY OUTPUT FOR GRID(',I2,',',I2,')')
07600       WRITE(3,200) UA2(LM),VA2(LM)
07610 200   FORMAT(/,40X,' U*2=',G10.3,5X,'V*2=',G10.3,' SQ M/SQ S')
07620       WRITE(3,250)
07630 250   FORMAT(20X,' UNDETERMINED PARAMETERS')
07640       WRITE(3,300) (I,I=1,JP)
07650 300   FORMAT(//,' ',31X,8(I2,10X))
07660       WRITE(3,310)(C(I+JLM),I=1,JP)
07670 310   FORMAT (22X,' C(J)',2X,8(E10.3,2X))
07680       WRITE(3,350) (D(I+JLM),I=1,JP)
07690 350   FORMAT (22X,' D(J)',2X,8(E10.3,2X))
```

```
07700       WRITE(3,400)
07710 400   FORMAT(/,20X,' MASS FLUXES-SQ MTS/SEC:')
07720       WRITE(3,450) UB(L,M),VB(L,M)
07730 450   FORMAT(22X,' QX=',E10.3,6X,'QY=',E10.3)
07740       WRITE(3,460) ETA(LM)
07750 460   FORMAT(/,20X,' SURFACE ELEVATION(M)=',G10.3)
07760       RETURN
07770       END
07780 C
07790 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
07800 C
07810       SUBROUTINE HOT
07820 C
07830 C
07840 C     ROUTINE WHICH INITIALIZES THE UNKNOWNS C(JLM), D(JLM) AND ETA(LM)
07850 C     TO VALUES ACCESSED FROM DISK FILE12.
07860 C
07870       COMMON/UNKNOW/C(864),D(864),ETA(288),UB(13,25),VB(13,25)
07880       COMMON/CON/JP,KP,LP,MP,KPM1,EH,EDDYH,C2,C4,C5,VERSIN(14),ITIME,
07890      &   AMP(12,24),IAMP(12,24),TITCIR(20),JPLPMP,LPMP,JPLP,LPP1,MPP1,
07900      &   MPM1,LPM1,CBC(3,13,25),DBC(3,13,25),HOTIME,PRTFLO,ISTPR,WAMP,
07910      &   DTT,DL,TITWND(20),TITVIS(20),TITDEN(20),PHI,WLNGTH,WPEROD
07920       INTEGER IAMP*2,AMP*2
07930       LOGICAL PRTFLO,RLID
07940       READ(12) (DUM,I=1,30),
07950      &         ((AMP(L,M),L=1,LP),M=1,MP),((IAMP(L,M),L=1,LP),M=1,MP)
07960 200   IF(PRTFLO) READ(12) ITIME,(DUM,LM=1,LPMP),(DUM,LM=1,LPMP)
07970       READ(12) ITIME,(((CBC(J,L,M),J=1,JP),L=1,LPP1),M=1,MPP1),
07980      &               (((DBC(J,L,M),J=1,JP),L=1,LPP1),M=1,MPP1),
07990      &               (ETA(LM),LM=1,LPMP)
08000       IF(ITIME.LT.HOTIME) GOTO 200
08010       DO 300 M=1,MP
08020       LMP=(M-1)*LP
08030       DO 300 L=1,LP
08040       LM=LMP+L
08050       JLMP=(LM-1)*JP
08060       DO 300 J=1,JP
08070       JLM=J+JLMP
08080       C(JLM)=CBC(J,L,M)
08090       D(JLM)=DBC(J,L,M)
08100 300   CONTINUE
08110       WRITE(3,400) ITIME,HOTIME
08120 400   FORMAT(///,' ***** PROGRAM IS READING VALUES OF C(JLM), D(JLM) & '
08130      &          ,'ETA(LM) FROM DISK FILE 12.',/,'STARTING TIME=',I9,
08140      &       ' SECONDS',/,'SPECIFIED STARTING TIME=',G10.3,' SECONDS')
08150       RETURN
08160       END
08170 C
```

```
08180 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
08190 C
08200       SUBROUTINE ERROR(IERR)
08210 C
08220       WRITE(3,100) IERR
08230 100   FORMAT(///,' * * ERROR NO.',I2,' HAS OCCURED IN PROGRAM CIRC.')
08240       RETURN
08250       END
08260 C * * * * * * * * * * * * * *
08390       INTEGER IAMP*2,AMP*2
08400       LOGICAL RLID,DENSE,PRTFLO
08410       DATA ASTER/'* * '/
08420 C
08430       REWIND 12
08440       WRITE(3,100) (ASTER,I=1,30),VERSIN
08450 100   FORMAT(///,'1',30A4,//,' ',40X,14A4,/,' ',40X,40(1H-))
08460       WRITE(12) TITCIR,DTT,PHI,DL,PERIOD,EDDYH,RLID,TLAST,
08470      &          DENSE,PRTFLO,IDTPR,
08480      &          ((AMP(L,M),L=1,LP),M=1,MP),((IAMP(L,M),L=1,LP),M=1,MP)
08490       WRITE(3,200) TIT1
08500 200   FORMAT(/,' ',30A4)
08510       RETURN
08520       END
08530 C$ENTRY
EOF:
```

```
00010 C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
00020 C
00030 C
00040 C                    PROGRAM PRTVEL (GAL)
00050 C
00060 C            VERSION 7 - JULY 1982 - REVISION A
00070 C
00080 C                  3-D CIRCULATION MODEL
00090 C
00100 C                  BY CORTIS COOPER
00110 C
00120 C      PART V OF PROGRAM GAL, THIS PROGRAM PRINTS VARIOUS INFORMATION
00130 C      SUMMARIZING THE RUN.  PROGRAMS VISCOUS, WIND (OR HURR), CIRC,
00140 C      AND POSSIBLY DENSITY SHOULD HAVE BEEN RUN BEFORE EXECUTING
00150 C      PROGRAM PRTVEL.  SEE GAL USERS MANUAL FOR DIRECTIONS AND FOR
00160 C      LIST OF REVISIONS.
00170 C
00180       COMMON/KNOW/H(288),CK(1552),ALPHA(864),BETA(864),
00190      &       NB(288),A(864),CB(288),EDDYV(1552),CONNV,LAND(288)
00200       COMMON/TTIME/ITIME,ETIME,DTIME,WTIME,RTIME,ICOUNT
00210       COMMON/UNKNOW/C(3,13,25),D(3,13,25),ETA(288),UB(288),VB(288),
00220      &  SPEED(288),DIREC(288)
00230       COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
00240      &   EDDYT,REYNO,WA2,AMP(288),IAMP(288),TIT1(20),TIT2(20),
00250      &   TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
00260       COMMON/WINDY/UA2(288),VA2(288)
00270       COMMON/DEN/ROWX(864),ROWY(864),RHOW(288),IPP,DENSE
00280       COMMON/PRNTR/LOCATE(10),NOGRDS,LAYERS,PRTETA,PRTFLO,
00290      &   LEVELS,Z(10,16),ZEND(16),LAY(16),LEV(16),RLID
00300 C
00310       INTEGER IAMP*2,AMP*2
00320       LOGICAL CONNV,EDDYS.EDDYT,DENSE,PRTETA,PRTFLO,RLID,PRTWND,
00330      &        PRTNV,PRTDEN,LAND*1
00340       REAL NB,LAY,LEV
00350       COMPLEX LOCATE
00360       DATA PRTNV,PRTWND,PRTDEN/3*.FALSE./
00370       NAMELIST/OUTOPS/PRTWND,PRTNV,PRTDEN,LOCATE,NOGRDS,LAYERS,PRTETA,
00380      &                LEVELS
00390 C
00400       READ(1,OUTOPS)
00410       READ(10) TIT2,LP,MP,JP,KP,CONNV,EDDYS,EDDYT,REYNO,WA2
00420       LPMP=LP*MP
00430       KPLPMP=KP*LPMP
00440       JPLPMP=JP*LPMP
00450       LPP1=LP+1
00460       MPP1=MP+1
00470       KPM1=KP-1
```

```
00480       READ(4) TIT4
00490       READ(10) (H(LM),LM=1,LPMP),(LAND(LM),LM=1,LPMP),
00500      &      (CB(LM),LM=1,LPMP),(CK(KLM),KLM=1,KPLPMP)
00510       READ(12) TIT1,DTT,CPHI,DL,PERIOD,EDDYH,RLID,TLAST,DENSE,
00520      &   PRTFLO,IDTPR,
00530      &   (AMP(LM),LM=1,LPMP),(IAMP(LM),LM=1,LPMP)
00540       IF(DENSE) READ(11) TIT3
00550       CALL GOPTS
00560       CALL HEDER
00570       ITLAST=INT(TLAST/IDTPR)*IDTPR-DTT
00580       IDTT=DTT
00590       DO 700 IT=1,ITLAST,IDTT
00600       TIME=IT
00610       IF(RTIME.GT.TIME) GOTO 700
00620       IF(PRTFLO) CALL RDFLO
00630       READ(12) ITIME,(((C(J,L,M),J=1,JP),L=1,LPP1),M=1,MPP1),
00640      &         (((D(J,L,M),J=1,JP),L=1,LPP1),M=1,MPP1),
00650      &         (ETA(LM),LM=1,LPMP)
00660       RTIME=ITIME
00670       CALL WRTIME(RTIME,' =')
00680       IF(ETIME.LT.RTIME) CALL RDNV(PRTNV)
00690       IF(WTIME.LT.RTIME) CALL RDWIND(LP,MP,LPMP,PRTWND)
00700       IF(DENSE.AND.DTIME.LT.TIME) CALL RDDEN(PRTDEN)
00710       IF(PRTETA) CALL WRTETA
00720       IF(NOGRDS.GT.0) CALL WRTPRF
00730       IF(LAYERS.GT.0) CALL PRTLAY
00740       IF(LEVELS.GT.0) CALL WRTPLT
00750 700   CONTINUE
00760       IF(LEVELS.GT.0) WRITE(3,800) ICOUNT
00770 800   FORMAT(//,' * * * * *  A TOTAL OF',I4,' PLOTS WERE CREATED')
00780 900   STOP
00790       END
00800 C
00810 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
00820 C
00830       BLOCK DATA
00840 C
00850       COMMON/KNOW/H(288),CK(1552),ALPHA(864),BETA(864)
00860      &       NB(288),A(864),CB(288),EDDYV(1552),CONNV,LAND(288)
00870       COMMON/PRNTR/LOCATE(10),NOGRDS,LAYERS,PRTETA,PRTFLO,
00880      &   LEVELS,Z(10,16),ZEND(16),LAY(16),LEV(16),RLID
00890       COMMON/TTIME/ITIME,ETIME,DTIME,WTIME,RTIME,ICOUNT
00900       COMMON/UNKNOW/C(3,13,25),D(3,13,25),ETA(288),UB(288),VB(288),
00910      &  SPEED(288),DIREC(288)
00920       COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
00930      &   EDDYT,REYNO,WA2,AMP(288),IAMP(288),TIT1(20),TIT2(20),
00940      &   TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
00950       COMMON/WINDY/UA2(288),VA2(288)
```

```
00960       COMMON/DEN/ROWX(864),ROWY(864),RHOW(288),IPP,DENSE
00970 C
00980       INTEGER IAMP*2,AMP*2
00990       LOGICAL CONNV,EDDYS,EDDYT,DENSE,PRTETA,PRTFLO,RLID,PRTWND,
01000      &        PRTNV,PRTDE,LAND*1
01010       REAL NB,LAY,LEV
01020       COMPLEX LOCATE
01030 C
01040 C
01050       DATA ITIME,ICOUNT,RTIME,ETIME,DTIME,WTIME/2*0,4*0.0/
01060       DATA EDDYS,EDDYT,DENSE,PRTETA,PRTFLO,RLID/
01070      &     T,F,F,T,F,F/
01080       DATA VERSIN/'GAL:','VERS','ION ','7 - ','JULY',' 4, ',
01090      &     '1982',' - R','EV. ','A - ','PROG','RAM ','PRTV','EL  '/
01100       END
01110 C
01120 C
01130 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01140 C
01150       SUBROUTINE HEDER
01160 C
01170 C
01180 C  ROUTINE WHICH PRINTS ESSENTIAL INFO THAT SUMMARIZES RUN.
01190 C
01200       COMMON/UNKNOW/C(3,13,25),D(3,13,25),ETA(288),UB(288),VB(288),
01210      &  SPEED(288),DIREC(288)
01220       COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
01230      &   EDDYT,REYNO,WA2,AMP(288),IAMP(288),TIT1(20),TIT2(20),
01240      &   TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
01250       COMMON/KNOW/H(288),CK(1552),ALPHA(864),BETA(864),
01260      &       NB(288),A(864),CB(288),EDDYV(1552),CONNV,LAND(288)
01270       COMMON/DEN/ROWX(864),ROWY(864),RHOW(288),IPP,DENSE
01280       INTEGER IAMP*2,AMP*2
01290       REAL NB
01300       LOGICAL CONNV,EDDYS,EDDYT,DENSE,LAND*1
01310       DATA ASTER/'* * '/
01320       WRITE(3,50) (ASTER,I=1,30),VERSIN
01330 50    FORMAT(///,'1',30A4,//,' ',40X,14A4,/,' ',40X,52(1H-))
01340       WRITE(3,100) TIT1
01350 100   FORMAT(/,' ',30X,20A4)
01360       WRITE(3,150) TIT2
01370 150   FORMAT(/,30X,' VISCOUS TITLE:  ',20A4)
01380       IF(DENSE) WRITE(3,200) TIT3
01390 200   FORMAT(/,30X,' DENSITY TITLE:  ',20A4)
01400       WRITE(3,250) TIT4
01410 250   FORMAT(/,30X,' WIND TITLE:',20A4)
01420       WRITE(3,300)JP,KP,LP,MP,EDDYH
01430 300   FORMAT(/,30X,' JP=',I1,3X,'KP=',I1,3X,'LP=',I2,3X,'MP=',I2,
```

```
01440       &     3X,' EDDYH=',E10.3)
01450        WRITE(3,350) DTT,DL,CPHI
01460 350    FORMAT(/,30X,' DT=',F4.0,'SECS',3X,'DL=',F7.1,' METERS',
01470       &     3X,'LATITUDE=',F4.0,'DEG N')
01480        WRITE(3,500)
01490 500    FORMAT(//,48X,'1STILL WATER DEPTHS (METERS)',/,48X,
01500       &     ' ',25(1H-),/)
01510        CALL OUTPUT(LP,MP,H,1)
01520        WRITE(3,600)
01530 600    FORMAT(//,48X,'1FRICTION COEFFICIENTS (M/SEC)',/,48X,30(1H-),/)
01540        CALL OUTPUT(LP,MP,CB,1)
01550        WRITE(3,700)
01560 700    FORMAT(//,41X,'1LAYER DEPTHS FOR THE EDDY VISCOSITY ',
01570       &   '(NONDIMENSIONAL):',/,' ',41X,51(1H-),/)
01580        CALL OUTPUT(LP,MP,CK,KP)
01590        WRITE(3,800)
01600 800    FORMAT(//,50X,'1INITIAL AMPLITUDES(MTS) ',/,50X,24(1H-))
01610        CALL EQUATE(AMP,LP,MP)
01620        WRITE(3,850) PERIOD
01630 850    FORMAT(/,' PERIOD OF THE TIDAL OSCILLATION=',G10.3,
01640       &    'SECS')
01650        WRITE(3,900)
01660 900    FORMAT(//,53X,'1ELEMENT TYPES ',/,53X,12(1H-),/)
01670        CALL EQUATE(IAMP,LP,MP)
01680        WRITE(3,950) (ASTER,I=1,30)
01690 950    FORMAT(/,' ',30A4)
01700        RETURN
01710        END
01720 C
01730 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01740 C
01750        SUBROUTINE EQUATE(IZ,LP,MP)
01760 C
01770 C
01780 C      ROUTINE WHICH CONVERTS INTEGER*2 VARIABLE TO REAL*4 SO
01790 C      THAT IT MAY BE PRINTED USING TINE OUTPUT.
01800 C
01810        DIMENSION IZ(1),Z(288)
01820        INTEGER*2 IZ
01830 C
01840        LPMP=LP*MP
01850        DO 500 LM=1,LPMP
01860        Z(LM)=IZ(LM)
01870 500    CONTINUE
01880        CALL OUTPUT(LP,MP,Z,1)
01890        RETURN
01900        END
01910 C
```

```
01920 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01930 C
01940 C
01950       SUBROUTINE OOPTS
01960 C
01970 C
01980 C  ROUTINE WHICH READS DATA NEEDED FOR OUTPUT OPTIONS.
01990 C
02000       COMMON/PRNTR/LOCATE(10),NOGRDS,LAYERS,PRTETA,PRTFLO,
02010      &   LEVELS,Z(10,16),ZEND(16),LAY(16),LEV(16),RLID
02020       LOGICAL PRTETA,PRTFLO
02030       COMPLEX LOCATE
02040       REAL LAY,LEV
02050       IF (NOGRDS.EQ.0) GO TO 200
02060 C READS IN DEPTHS AT WHICH VELOCITY WILL BE CALCULATED AND
02070 C PRINTED FOR A PARTICULAR WATER COLUMN.
02080       DO 100 J=1,NOGRDS
02090       READ(1,10)(Z(J,I),I=1,16)
02100 10    FORMAT (16F5.1)
02110       DO 20 I=2,16
02120       IF(Z(J,I).LT.0.001) GO TO 100
02130 20    CONTINUE
02140 100   ZEND(J)=I-1
02150 200   IF(LAYERS.EQ.0)GO TO 300
02160       READ(1,10)(LAY(I),I=1,LAYERS)
02170 C     WRITE(3,10)(LAY(I),I=1,LAYERS)
02180 300   IF(LEVELS.EQ.0) GO TO 500
02190       READ(1,10)(LEV(I),I=1,LEVELS)
02200 500   RETURN
02210       END
02220 C
02230 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02240 C
02250 C
02260       SUBROUTINE OUTPUT (LP,MP,Z,INDEX)
02270 C
02280 C
02290 C  ROUTINE WHICH PRINTS A TABLE.  MAIN OUTPUT ROUTINE USED IN THIS PROGR
02300 C
02310       DIMENSION Z(1)
02320       LPAGE=0
02330       MPAGE=(LP-1)/10+1
02340       LPI=LP*INDEX
02350       DO 800 MM=1,MPAGE
02360       LSTART=10*(MM-1)*INDEX+1
02370       LSTOP=10*MM*INDEX
02380       JSTOP=10*MM
02390       JSTART=10*(MM-1)+1
```

```
02400       IF(MM.EQ.MPAGE) LSTOP=LP*INDEX
02410       IF(MM.GT.1) WRITE(3,100)
02420 100   FORMAT('1',////,30X,'CONTINUED FROM PREVIOUS PAGE')
02430       WRITE(3,200)(J,J=JSTART,JSTOP)
02440 200   FORMAT(53X,'  X DIRECTION',/,' -Y-',6X,10(I2,10X))
02450       WRITE(3,300)
02460 300   FORMAT(' ---',6X,10('--',10X))
02470       DO 700 M=1,MP
02480       NN=MP+1-M
02490       LM=(NN-1)*LPI
02500       DO 500 N=1,INDEX
02510       LMINDX=LM+N-1
02520       WRITE(3,400) NN,(Z(LMINDX+L),L=LSTART,LSTOP,INDEX)
02530 400   FORMAT(' ',I2,3X,10(G11.4,1X))
02540 500   CONTINUE
02550       NPAGE=(M*INDEX)/48
02560       IF(NPAGE.EQ.LPAGE) GO TO 700
02570       LPAGE=NPAGE
02580       WRITE(3,600)
02590 600    FORMAT('1',////)
02600 700   CONTINUE
02610 800   CONTINUE
02620       RETURN
02630       END
02640 C
02650 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  *
02660 C
02670 C
02680       SUBROUTINE VELOCY (L,M,Z,U,V)
02690 C
02700 C
02710 C  ROUTINE WHICH CALCULATES VELOCITIES FOR TIME T, LOCATION L,M AND DEP
02720 C
02730       COMMON/KNOW/H(288),CK(1552),ALPHA(864),BETA(864),
02740      &       NB(288),A(864),CB(288),EDDYV(1552),CONNV,LAND(288)
02750       COMMON/UNKNOW/C(3,13,25),D(3,13,25),ETA(288),UB(288),VB(288),
02760      &  SPEED(288),DIREC(288)
02770       COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
02780      &   EDDYT,REYNO,WA2,AMP(288),IAMP(288),TIT1(20),TIT2(20),
02790      &   TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
02800       COMMON/WINDY/UA2(288),VA2(288)
02810 C
02820       INTEGER IAMP*2,AMP*2
02830       LOGICAL CONNV,LAND*1
02840       REAL NB
02850       LMP=(M-1)*LP
02860       LM=L+LMP
02870       KLMP=(LM-1)*KPM1
```

```
02880       IS=KLMP+1
02890 C
02900       UO1=Z*Z*H(LM)*(Z-1)/NB(LM)
02910       UO2=1./BETA(IS)*H(LM)*(1.-Z)
02920       DIF=ABS((BETA(IS)-NB(LM))/NB(LM))
02930       IF(DIF.LT.0.01) GO TO 50
02940       UO2=ALOG(NB(LM)/(ALPHA(IS)*Z*H(LM)+BETA(IS)))/ALPHA(IS)
02950 50    UO=UA2(LM)*(UO1+UO2)
02960       VO=VA2(LM)*(UO1+UO2)
02970       UJ=0.0
02980       VJ=0.0
02990       JLMP=(LM-1)*JP
03000       DO 100 J=1,JP
03010       JLM=J+JLMP
03020       UJ=C(J,L,M)*COS(A(JLM)*Z)+UJ
03030 100   VJ=D(J,L,M)*COS(A(JLM)*Z)+VJ
03040 C
03050       U=UO+UJ
03060       V=VO+VJ
03070       ABVJ=ABS(VJ)
03080       JLM=JLM-2
03090       ABUJ=ABS(UJ)
03100       IF(ABVJ.LT.1.0E-10) V=0.0
03110       IF(ABUJ.LT.1.0E-10) U=0.0
03120 C
03130       RETURN
03140       END
03150 C
03160 C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
03170 C
03180 C
03190       SUBROUTINE SPEDRC(DEPTH,LP,MP,KOUNT)
03200 C
03210 C
03220 C  ROUTINE WHICH CALCULATES THE SPEED AND DIRECTION FOR THE ENTIRE
03230 C  GRID FOR A SPECIFIED WATER DEPTH.
03240 C
03250       COMMON/UNKNOW/C(3,13,25),D(3,13,25),ETA(288),UB(288),VB(288),
03260      &  SPEED(288),DIREC(288)
03270       COMMON/KNOW/H(288),CK(1552),ALPHA(864),BETA(864),
03280      &       NB(288),A(864),CB(288),EDDYV(1552),CONNV,LAND(288)
03290       LOGICAL CONNV,LAND*1
03300 C
03310       KOUNT=0.0
03320       DO 600 M=1,MP
03330       LMP=(M-1)*LP
03340       DO 600 L=1,LP
03350       LM=L+LMP
```

```
03360       SPEED(LM)=0.0
03370       DIREC(LM)=0.0
03380       IF(LAND(LM).OR.DEPTH.GT.H(LM)) GO TO 600
03390       CALL VELOCY(L,M,DEPTH,U,V)
03400       SPEED(LM)=SQRT(U*U+V*V)
03410       ABV=ABS(V)
03420       ABU=ABS(U)
03430       IF(ABV.GT.1.0E-10.OR.ABU.GT.1.0E-10)
03440      &DIREC(LM)=ATAN2(V,U)*180./3.14159
03450   600 CONTINUE
03460       RETURN
03470       END
03480 C
03490 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
03500 C
03510       SUBROUTINE RDNV(PRTNV)
03520 C
03530 C   ROUTINE WHICH READS CONSTANTS ASSOCIATED WITH EDDY VISCOSITY AND
03540 C   PRINTS TABLES SUMMARIZING THE CONSTANTS.
03550 C
03560       COMMON/KNOW/H(288),CK(1552),ALPHA(864),BETA(864),
03570      &       NB(288),A(864),CB(288),EDDYV(1552),CONNV,LAND(288)
03580       COMMON/TTIME/ITIME,ETIME,DTIME,WTIME,RTIME,ICOUNT
03590       COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
03600      &   EDDYT,REYNO,WA2,AMP(288),IAMP(288),TIT1(20),TIT2(20),
03610      &   TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
03620 C
03630       INTEGER IAMP*2,AMP*2
03640       LOGICAL CONNV,EDDYS,EDDYT,DENSE,PRTETA,PRTFLO,RLID,
03650      &        PRTNV,LAND*1
03660       REAL NB
03670 C
03680       JJLPMP=JP*JPLPMP
03690       KLMM1=LPMP*KPM1
03700       IF(CONNV) JJLPMP=JPLPMP
03710 50    READ(10) ETIME,(A(JLM),JLM=1,JPLPMP),(DUM,JLM=1,JPLPMP),
03720      &         (DUM,JLM=1,JPLPMP),(DUM,JLM=1,JJLPMP),
03730      &         (DUM,JLM=1,JPLPMP),(DUM,JLM=1,JPLPMP),
03740      &         (DUM,JLM=1,JPLPMP),(DUM,LM=1,LPMP),
03750      &         (EDDYV(KLM),KLM=1,KPLPMP),(NB(LM),LM=1,LPMP),
03760      &         (ALPHA(KLM),KLM=1,KLMM1),(BETA(KLM),KLM=1,KLMM1)
03770       IF(ETIME.LT.RTIME) GOTO 50
03780       IF(.NOT.PRTNV) RETURN
03790       IF(EDDYS.OR.EDDYT) WRITE(3,400) REYNO
03800 400   FORMAT(/30X,' REYNOLDS NO.=',F5.1)
03810       IF (EDDYS.AND.(.NOT.EDDYT)) WRITE(3,450) WA2
03820 450   FORMAT(/30X,' THE AVG. WIND STRESS USED IN NV MODEL=',E10.3)
03830       WRITE(3,200)
```

```
03840 200   FORMAT(//,52X,'INV (SQ MTS/SEC)',/,52X,16(1H-),/)
03850       CALL OUTPUT(LP,MP,EDDYV,KP)
03860       WRITE(3,300)
03870 300   FORMAT(//,57X,'IA(JLM)',/,57X,7(1H-),/)
03880       CALL OUTPUT(LP,MP,A,JP)
03890       RETURN
03900       END
03910 C
03920 C
03930 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
03940 C
03950       SUBROUTINE RDFLO
03960 C
03970 C     ROUTINE WHICH READS IN MASS FLUXES AND PRINTS OUT TABLE.
03980 C
03990       COMMON/TTIME/ITIME,ETIME,DTIME,WTIME,RTIME,ICOUNT
04000       COMMON/UNKNOW/C(3,13,25),D(3,13,25),ETA(288),UB(288),VB(288),
04010      &  SPEED(288),DIREC(288)
04020       COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
04030      &   EDDYT,REYNO,WAS,AMP(288),IAMP(288),TIT1(20),TIT2(20),
04040      &   TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
04050       COMMON/DEN/ROWX(864),ROWY(864),RHOW(288),IPP,DENSE
04060 C
04070       INTEGER IAMP*2,AMP*2
04080       LOGICAL CONNV,EDDYS,EDDYT,DENSE,PRTETA,PRTFLO,RLID,PRTWND,
04090      &        PRTNV,LAND*1
04100 C
04110       READ(12) ITIME,(UB(LM),LM=1,LPMP),(VB(LM),LM=1,LPMP)
04120       RTIME=ITIME
04130       CALL WRTIME(ETIME,' =')
04140       WRITE(3,140)
04150 140   FORMAT('1',///,41X,'MASS FLUXES IN X DIRECTION - SQ MTS/SEC',
04160      &   /,41X,39(1H-))
04170       CALL OUTPUT(LP,MP,UB,1)
04180       WRITE(3,150)
04190 150   FORMAT('1',///,41X,'MASS FLUXES IN Y DIRECTION - SQ MTS/SEC',
04200      &  /,41X,39(1H-))
04210       CALL OUTPUT(LP,MP,VB,1)
04220       RETURN
04230       END
04240 C
04250 C
04260 C
04270 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
04280 C
04290 C
04300       SUBROUTINE RDWIND(LP,MP,LPMP,PRTWND)
04310 C
```

```
04320 C     ROUTINE WHICH READS IN WIND SHEAR VELOCITY AND PRINTS A TABLE.
04330 C
04340 C
04350       COMMON/TTIME/ITIME,ETIME,DTIME,WTIME,RTIME,ICOUNT
04360       COMMON/WINDY/UA2(288),VA2(288)
04370 C
04380       LOGICAL PRTWND
04390 C
04400 50    READ(4) WTIME,(UA2(LM),LM=1,LPMP),(VA2(LM),LM=1,LPMP)
04410       WTIME=WTIME*3600.
04420       IF(WTIME.LT.RTIME) GOTO 50
04430       IF(.NOT.PRTWND) RETURN
04440       WRITE(3,100)
04450 100   FORMAT(//,'1',30X,'1WIND SHEAR VELOCITY SQRD IN X DIRECTION',
04460      &       ' (SQ MTS/SQ SEC)',/,' ',30X,54(1H-))
04470       CALL OUTPUT(LP,MP,UA2,1)
04480       WRITE(3,200)
04490 200   FORMAT(//,'1',30X,'1WIND SHEAR VELOCITY SQRD IN Y DIRECTION',
04500      &       ' (SQ MTS/SQ SEC)',/,' ',30X,54(1H-))
04510       CALL OUTPUT(LP,MP,VA2,1)
04520       RETURN
04530       END
04540 C
04550 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
04560 C
04570 C
04580       SUBROUTINE RDDEN(PRTDEN)
04590 C
04600 C
04610       COMMON/TTIME/ITIME,ETIME,DTIME,WTIME,RTIME,ICOUNT
04620       COMMON/DEN/ROWX(864),ROWY(864),RHOW(288),IPP,DENSE
04630       COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
04640      &   EDDYT,REYNO,WA2,AMP(288),IAMP(288),TIT1(20),TIT2(20),
04650      &   TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
04660 C
04670       LOGICAL PRTDEN,EDDYS,EDDYT
04680       INTEGER IAMP*2,AMP*2
04690 C
04700 50    READ(11) DTIME,IPP,(DUM,JLM=1,JPLPMP),(DUM,JLM=1,JPLPMP),
04710      &         (RHOW(LM),LM=1,LPMP)
04720       IF(DTIME.LT.RTIME) GOTO 50
04730       IF(.NOT.PRTDEN) RETURN
04740       WRITE(3,100)
04750 100   FORMAT(//,30X,'1WATER DENSITY IN THE SURFACE LAYER ',
04760      &      ' (G/CUBIC CM)',/,30X,' ',48(1H-))
04770       CALL OUTPUT(LP,MP,RHOW,1)
04780       RETURN
04790       END
```

```
04800 C
04810 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
04820 C
04830 C
04840       SUBROUTINE WRTIME(TIME,CHAR)
04850 C
04860 C     ROUTINE WHICH WRITES OUT TIME HEADING.
04870 C
04880       INTEGER CHAR*2
04890       DATA ASTER/'* * '/
04900 C
04910       IF(TIME.GT.1.0E32) TIME=0.0
04920       TIMEH=TIME/3600.
04930       WRITE(3,50)(ASTER,I=1,30)
04940 50    FORMAT(//,'1',30A4)
04950       WRITE(3,100) CHAR,TIMEH
04960 100   FORMAT(//5X,21(1H-)/4X,'/ TIME',A2,
04970      &F6.2,' HOURS',2X,'/',/,4X,21(1H-))
04980       RETURN
04990       END
05000 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
05010 C
05020       SUBROUTINE WRTPLT
05030 C
05040 C     ROUTINE WHICH WRITES A DISK FILE CONTAINING SPEED AND DIRECTION
05050 C     DATA WHICH CAN BE PLOTTED USING PROGRAM CURPLOT.
05060 C
05070 C
05080       COMMON/TTIME/ITIME,ETIME,DTIME,TTIME,RTIME,ICOUNT
05090       COMMON/UNKNOW/C(3,13,25),D(3,13,25),ETA(288),UB(288),VB(288),
05100      &  SPEED(288),DIREC(288)
05110       COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
05120      &   EDDYT,REYNO,WA2,AMP(288),IAMP(288),TIT1(20),TIT2(20),
05130      &   TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
05140       COMMON/PRNTR/LOCATE(10),NOGRDS,LAYERS,PRTETA,PRTFLO,
05150      &   LEVELS,Z(10,16),ZEND(16),LAY(16),LEV(16),RLID
05160 C
05170       INTEGER IAMP*2,AMP*2
05180       LOGICAL CONNV,EDDYS,EDDYT,DENSE,PRTETA,PRTFLO,RLID,PRTWND,
05190      &        PRTNV,PRTDEN,LAND*1
05200       REAL LAY,LEV
05210       COMPLEX LOCATE
05220 C
05230       DO 200 J=1,LEVELS
05240       ICOUNT=ICOUNT+1
05250       CALL SPEDRC(LEV(J),LP,MP,DUM)
05260       DEPTH=10.*LEV(J)
05270       WRITE(3,100) DEPTH
```

```
05280 100   FORMAT('1',////,20X,' DISK FILE CONTAINING CURRENT SPEED AND',
05290      &      'DIRECTION FOR LEVEL',F5.1,' % WAS CREATED')
05300       WRITE(13) RTIME,LEV(J),(SPEED(LM),DIREC(LM),LM=1,LPMP)
05310 200   CONTINUE
05320       RETURN
05330       END
05340 C
05350 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
05360 C
05370       SUBROUTINE PRTLAY
05380 C
05390 C     ROUTINE WHICH PRINTS PLAN VIEW OF VELOCITIES.
05400 C
05410 C
05420       COMMON/TTIME/ITIME,ETIME,DTIME,WTIME,RTIME,ICOUNT
05430       COMMON/UNKNOW/C(3,13,25),D(3,13,25),ETA(288),UB(288),VB(288),
05440      &  SPEED(288),DIREC(288)
05450       COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
05460      &   EDDYT,REYNO,WA2,AMP(288),IAMP(288),TIT1(20),TIT2(20),
05470      &   TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
05480       COMMON/PRNTR/LOCATE(10),NOGRDS,LAYERS,PRTETA,PRTFLO,
05490      &   LEVELS,Z(10,16),ZEND(16),LAY(16),LEV(16),RLID
05500 C
05510       INTEGER IAMP*2,AMP*2
05520       LOGICAL CONNV,EDDYS,EDDYT,DENSE,PRTETA,PRTFLO,RLID,PRTWND,
05530      &        PRTNV,PRTDEN,LAND*1
05540       REAL LAY,LEV
05550       COMPLEX LOCATE
05560 C
05570       DO 1000 J=1,LAYERS
05580       CALL SPEDRC(LAY(J),LP,MP,DUM)
05590       DEPTH=LAY(J)
05600       WRITE(3,620) DEPTH
05610 620   FORMAT ('1'////,39X,' CURRENT SPEEDS AT LEVEL',F5.1,' % ',
05620      & 'IN MTS/SEC',/,39X,' ',43(1H-))
05630       CALL OUTPUT (LP,MP,SPEED,1)
05640       WRITE(3,640) DEPTH
05650 640   FORMAT ('1'///,32X,' CURRENT DIRECTIONS AT LEVEL',F5.1,' % IN '
05660      &, 'DEG FROM THE X AXIS',/,32X,' ',59(1H-))
05670       CALL OUTPUT (LP,MP,DIREC,1)
05680  1000 CONTINUE
05690       RETURN
05700       END
05710 C
05720 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
05730 C
05740       SUBROUTINE WRTETA
05750 C
```

```
05760       COMMON/TTIME/ITIME,ETIME,DTIME,WTIME,RTIME,ICOUNT
05770       COMMON/UNKNOW/C(3,13,25),D(3,13,25),ETA(288),UB(288),VB(288),
05780      &  SPEED(288),DIREC(288)
05790       COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
05800      &   EDDYT,REYNO,WA2,AMP(288),IAMP(288),TIT1(20),TIT2(20),
05810      &   TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
05820       COMMON/WINDY/UA2(288),VA2(288)
05830 C
05840       INTEGER IAMP*2,AMP*2
05850       LOGICAL CONNV,EDDYS,EDDYT,DENSE,PRTETA,PRTFLO,RLID,PRTWND,
05860      &        PRTNV,LAND*1
05870       WRITE(3,120)
05880 120   FORMAT('1',///,41X,'WATER ELEVATION(MTS) ABOVE STILL WATER',
05890      &  /,41X,38(1H-))
05900       CALL OUTPUT(LP,MP,ETA,1)
05910       RETURN
05920       END
05930 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
05940 C
05950       SUBROUTINE WRTPRF
05960 C
05970 C     ROUTINE WHICH WRITES TABLE CONTAINING VELOCITY PROFILE.
05980 C
05990       COMMON/KNOW/H(288),CK(1552),ALPHA(864),BETA(864),
06000      &       NB(288),A(864),CB(288),EDDYV(1552),CONNV,LAND(288)
06010       COMMON/PRNTR/LOCATE(10),NOGRDS,LAYERS,PRTETA,PRTFLO,
06020      &   LEVELS,Z(10,16),ZEND(16),LAY(16),LEV(16),RLID
06030       COMMON/TTIME/ITIME,ETIME,DTIME,WTIME,RTIME,ICOUNT
06040       COMMON/UNKNOW/C(3,13,25),D(3,13,25),ETA(288),UB(288),VB(288),
06050      &  SPEED(288),DIREC(288)
06060       COMMON/CON/JP,KP,LP,MP,KPM1,CPHI,DTT,DL,PERIOD,EDDYS,
06070      &   EDDYT,REYNO,WA2,AMP(288),IAMP(288),TIT1(20),TIT2(20),
06080      &   TIT3(20),TIT4(20),VERSIN(14),EDDYH,JPLPMP,LPMP,KPLPMP
06090 C
06100       INTEGER IAMP*2,AMP*2
06110       LOGICAL LAND*1,CONNV,EDDYS,EDDYT,DENSE,PRTETA,PRTFLO,RLID,PRTWND,
06120      &        PRTNV
06130       REAL NB,LAY,LEV
06140       COMPLEX LOCATE
06150 C
06160 C  PRINTS THE VELOCITY PROFILES FOR THE SPECIFIED GRIDS.
06170       DO 999 N=1, NOGRDS
06180       L=REAL (LOCATE(N))
06190       M=AIMAG(LOCATE(N))
06200       LMP=(M-1)*LP
06210       LM=L+LMP
06220       WRITE(3,210) L,M
06230 210   FORMAT('1',//,45X,' VELOCITY PROFILE - GRID(',I2,
```

```
06240      &',',I2,')',/,45X,30(1H-))
06250       WRITE(3,220)
06260 220   FORMAT (/,20X,' Z-MTS',T35,'U(Z)-M/S',
06270      & T55,'V(Z)-M/S',T75,'SPEED-M/S',T95,'DIREC-DEG')
06280       WRITE(3,230)
06290 230   FORMAT (' ',20X,5(1H-),T35,8(1H-),T55,8(1H-),
06300      & T75,9(1H-),T95,9(1H-))
06310       IF(LAND(LM)) GO TO 450
06320       IP=ZEND(N)
06330       DO 280 I=1,IP
06340       IF(Z(N,I).LE.H(LM)) GO TO 240
06350       WRITE(3,235) Z(N,I),H(LM)
06360 235   FORMAT(' ',19X,F5.1,T35,'DEPTH SPECIFIED IS BELOW THE ',
06370      &  'STILL WATER DEPTH FOR THE GRID(',F5.1,')')
06380       GO TO 280
06390 240   DEPTH=Z(N,I)/H(LM)
06400       CALL VELOCY (L,M,DEPTH,U,V)
06410       ABV=ABS(V)
06420       ABU=ABS(U)
06430       SPEED(I)=SQRT (U*U+V*V)
06440       IF(ABV.GT.1.0E-10.OR.ABU.GT.1.0E-10)
06450      &DIREC(I)=ATAN2(V,U)*180./3.14159
06460       WRITE(3,250) Z(N,I),U,V,SPEED(I),DIREC(I)
06470 250   FORMAT (' ',19X,F5.1,T34,E10.3,T54,E10.3,
06480      & T75,E10.3,T96,F6.1)
06490 280   CONTINUE
06500       WRITE(3,300) (I,I=1,JP)
06510 300   FORMAT (//,20X,' UNDETERMINED PARAMETERS:',/,
06520      & ' ',31X,8(I2,10X))
06530       WRITE(3,310)(C(I,L,M),I=1,JP)
06540 310   FORMAT (22X,' C(J)',2X,8(E10.3,2X))
06550       WRITE(3,320) (D(I,L,M),I=1,JP)
06560 320   FORMAT (22X,' D(J)',2X,8(E10.3,2X))
06570       WRITE(3,330)
06580 330   FORMAT(/,20X,' MASS FLUXES-SQ MTS/SEC:')
06590       WRITE(3,340) UB(LM),VB(LM)
06600 340   FORMAT(22X,' QX=',E10.3,6X,'QY=',E10.3)
06610       WRITE(3,400) ETA(LM)
06620 400   FORMAT(/,20X,' SURFACE HEIGHT - MTS:',E10.3)
06630       GO TO 999
06640 450   WRITE(3,460)
06650 460   FORMAT(//,40X,' -- LAND BOUNDARY - ZERO VELOCITY --',////)
06660 999   CONTINUE
06670       RETURN
06680       END
06690 C$ENTRY
EOF:
```

```
00010 C              * * * * *    PLOTVEL 7.A  - * * * * *
00020 C
00030 C
00040 C  AUTHOR:  C.K. COOPER, NECE AND B. FIDLER, UMO.
00050 C  DATE:    ORIGINAL VERSION BY FIDLER IN JUNE 1979, MAJOR MODIFICATION
00060 C           APRIL 1982 BY COOPER.
00070 C PURPOSE:  PROGRAM PLOTS A PLAN VIEW OF THE VELOCITY FIELD ON A RECT.
00080 C           GRID SYSTEM OF LP BY MP ELEMENTS.  EACH ELEMENT IS A SQUARE
00090 C           OF ACTUAL LENGTH DL AND PLOTTED LENGTH GRDSZ.
00100 C
00110 C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
00120 C
00130 C  HISTORY OF REVISIONS:
00140 C
00150 C   REV 1.2:  READS IN VELOCITY IN F4.0 FORMAT AND READS PLAN VEIW
00153 C            DEPTH AS A % OF LOCAL DEPTH, NOT METERS.
00155 C  REV 7.A:  SAME AS 1.2 EXCEPT READS IN VARIABLE SPANNED RECORD.
00160 C
00170 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
00180 C
00190 C  DESCRIPTION OF INPUT:
00200 C
00210 C   TWO FILES ARE NECESSARY, ONE CONSISTING OF A FORTRAN NAMELIST
00220 C   AND THE OTHER OF SPEED AND DIRECTION DATA TO BE PLOTTED.
00230 C   THE SPEED/DIREC FILE FORMAT IS DEFINED IN ROUTINE RDSPD.  EACH
00240 C   ARRAY OF SPEED AND DIRECTION SHOULD BE PRECEEDED BY ONE RECORD
00250 C   CONTAINING THE TIME AND DEPTH (M) OF THE PLAN VIEW.
00260 C   THE NAMELIST FILE SHOULD CONTAIN THE FOLLOWING VARIABLES:
00270 C
00280 C    GRDSZ = THE INDIVICUAL ELEMENT SIZE OF THE GRID SYSTEM IN INCHES.
00290 C            ACTUAL SIZE PLOTTED WILL BE GRDSZ*FACT INCHES.
00300 C    LP    = THE NUMBER OF ELEMENTS IN THE X-DIRECTION, I.E. THE
00310 C            DIRECTION ALIGNED WITH THE WIDTH OF THE PLOTTING PAPER.
00320 C    MP    = THE NUMBER OF ELEMENTS IN THE Y-DIRECTION, I.E. THE
00330 C            THE DIRECTION ALIGNED WITH THE LENGTH OF THE PAPER.
00340 C    DL    = THE INDIVIDUAL ELEMENT SIZE IN METERS.
00350 C    ISKIP = THE NUMBER OF PLAN VIEWS TO BE INITIALLY SKIPPED BEFORE
00360 C            DRAWING BEGINS.  SHOULD BE 0 IF 1ST PLAN VIEW IS TO BE
00370 C            DRAWN.
00380 C    INC   = THE NUMBER OF PLAN VIEWS TO BE SKIPPED BETWEEN DRAWINGS.
00390 C            SHOULD BE 0 IF EACH SUCCESSIVE PLAN VIEW IS TO BE DRAWN.
00400 C    VECTOR = LOGICAL VARIABLE.  TRUE INDICATES VELOCITY VECTORS WILL
00410 C            BE DRAWN AS VECTORS WHOSE LENGTH WILL BE PROPORTIONAL
00420 C            TO THE SPEED.  FALSE INDICATES THE VELOCITY WILL BE SHOWN
00430 C            AND AN ARROW OF CONSTANT LENGTH WITH FEATHERS ON THE TAIL.
00440 C            EACH FEATHER WILL EQUAL SPDSCL (SEE BELOW).
```

```
00450 C    FACT  = THE FACTOR BY WHICH THE PLOTS WILL BE MODIFIED.  IF .LT.1
00460 C            THE PLOTS WILL BE REDUCED.  IF .GT.1 THE PLOTS WILL BE
00470 C            AMPLIFIED.
00480 C    SCALE = LOGICAL VARIABLE. TRUE INDICATES PLOTS WILL BE SCALED
00490 C            AUTOMATICALLY BASED ON THE MAXIMUM SPEED EQUAL GRDSZ.
00500 C            FALSE INDICATES THAT SPDSCL WILL BE USED TO SCALE.
00510 C   SPDSCL = ONLY USED IF SCALE EQUAL FALSE.  SPDSCL IS THE LENGTH OF
00520 C            ONE FEATHER ON THE VELOCITY ARROW IF VECTOR.EQ.FALSE.  IF
00530 C            VECTOR.EQ.TRUE SPDSCL SHOULD BE APPROXIMATELY EQUAL TO
00540 C            THE MAXIMUM SPEED TO OCCUR.
00550 C   ANGT   = THE ANGLE FROM THE X-AXIS TO TRUE NORTH IN DEGREES
00560 C            CCW FROM THE X-AXIS.
00570 C   NOPLTS = THE NUMBER OF PLAN VIEWS TO BE DRAWN.
00580 C   NWF1   = THE DEVICE NUMBER WHERE THE SPEED/DIREC DATA IS RESIDENT
00590 C   PRT    = PRINT SWITCH.  TRUE INDICATES TABLES OF THE SPEED AND
00600 C            DIRECTION ARRAYS WILL BE CREATED.  FALSE, NO TABLES.
00610 C   DIST   = THE DISTANCE IN INCHES BETWEEN TWO CONSECUTIVE PLAN VIEWS.
00620 C   SUNITS = TELLS PROGRAM WHAT UNITS THE SPEED WILL BE INPUT.  IF
00623 C            M/S THEN SUNITS=1., IF CM/S THEN .01, IF MM/S THEN .001, EC.
00625 C   DUNITS = TELLS PROGRAM WHAT UNITS THE DIRECTION WILL BE INPUT.
00627 C            IF IN TENS OF DEGREES, THEN DUNITS=10., IF TO THE
00628 C            NEAREST WHOLE DEGREE, THEN 1.
00630 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
00640 C
00650       COMMON /AA/GRDSZ,GRD,NWF1,MAXSPD,SCALE,SPDSCL,NOPLTS,LP,MP,
00660      & DL,ANGT,SPEED(650),DIREC(650),TITLE(20),TIME,LEV,
00670      & ISKIP,INC,VECTOR,LPMP,PRT,FEATHL,FEATHD,ARROWL,SUNITS,DUNITS
00680       LOGICAL PRT,VECTOR,SCALE
00690       REAL MAXSPD,LEV
00700       NAMELIST /MISC/ GRDSZ,LP,MP,DL,ISKIP,INC,VECTOR,FACT,
00710      &                SCALE,SPDSCL,ANGT,NOPLTS,NWF1,PRT,DIST,
00713      &                SUNITS,DUNITS
00720       EXTERNAL TEKQIK
00730 C   READ PLOTTING INFORMATION
00740       READ(1,10) TITLE
00750 10    FORMAT(20A4)
00760       READ(1,MISC)
00770       LPMP=LP*MP
00780       ANGT=360.-ANGT
00790 C   SET LENGTH OF PAPER USED.
00800       PLTLEN=(NOPLTS*((GRDSZ*FLOAT(MP)+5.)+5.)+10.)*FACT
00810 C   INITIALIZE PLOTTER AND SET UP PLOT.
00820       CALL UMPLOT(PLTLEN)
00830       CALL FORM('   ')
00840       CALL HEADER('BOX 705 : COOPER',16)
00850       CALL PLOTS(0,0,14)
00860 C   NEXT CALL PLOT SETS DISTANCE FROM EDGE OF PAPER TO GRID EDGE.
00870       CALL PLOT(0.0,1.5,-3)
```

```
00880       CALL FACTOR(FACT)
00890 C   SKIP UNWANTED DATA FILES TO DESIRED FIRST PLOT
00900       IF(ISKIP.GT.0)  CALL SKIP (ISKIP,&998)
00910 C   MAIN ITERATIVE LOOP.  LOOP IS REPEATED FOR EACH NEW PLOT.
00920       DO 20 NOP=1,NOPLTS
00930 C   NEXT CALL PLOT SETS DISTANCE BETWEEN PLOTS.
00940       CALL PLOT(DIST,0.0,-3)
00950       CALL GRID(0.,0.,GRDSZ,GRDSZ,MP,LP)
00960 C   CALL SUBROUTINES WHICH PLOT THE VELOCITY FIELD.
00970       PLEN= GRDSZ*FLOAT(MP)
00980 C   NEXT CALL PLOT RESETS ORIGION AT GRID L=1, M=1.
00990       CALL PLOT (PLEN,0.,-3)
01000       CALL RDSPD(&998)
01010       CALL VELVEC
01020 C   IDENTIFY THE PLOT BY TITLE, TIME AND DEPTH AND PLOT NORTH ARROW.
01030 C   NEXT CALL PLOT SETS DISTANCE OF TITLE FROM BOTTOM OF GRID.
01040       CALL PLOT(.5,1.0,-3)
01050       CALL PLABL
01060 C   NEXT CALL PLOT SETS DISTANCE FROM TIME AND DEPTH LABEL TO
01070 C   THE NORTH ARROW AND THE SCALES.
01080       CALL PLOT(1.25,0.,-3)
01090       CALL NASPLT
01100 C   SKIP RECORDS IF INC IS NOT ZERO.
01110       IF(INC.GT.0)  CALL SKIP(INC,LP,MP,NWF1,&998)
01120       CALL PLOT(0.0,-1.0,-3)
01130 20    CONTINUE
01140       CALL PLOT(1.,0.,999)
01150       GOTO 1000
01160 998   WRITE(3,999) NOP
01170 999   FORMAT(//,'* * * * *  EXECUTION TERMINATING - PREMATURE',
01180      & ' END OF DATA.',/,' * * * * * RECORDS READ=',I5)
01190 1000  CONTINUE
01200       RETURN
01210       END
01220 C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01230 C
01240       BLOCK DATA
01250       COMMON /AA/GRDSZ,GRD,NWF1,MAXSPD,SCALE,SPDSCL,NOPLTS,LP,MP,
01260      & DL,ANGT,SPEED(650),DIREC(650),TITLE(20),TIME,LEV,
01270      & ISKIP,INC,VECTOR,LPMP,PRT,FEATHL,FEATHD,ARROWL,SUNITS,DUNITS
01280       REAL MAXSPD,LEV
01290       LOGICAL PRT,VECTOR,SCALE
01300       DATA GRDSZ,DL,SUNITS,DUNITS,NOPLTS,LP,MP/
01303      &     .5,30000.,1.,1.,1,12,24/
01310       DATA ANGT,MAXSPD/65.,.1/,FEATHL,FEATHD,ARROWL/3.,10.,.75/
01320       DATA NWF1/13/,SCALE/F/,VECTOR/F/,FACT,DIST/.6,4./
01330       DATA ISKIP,INC /0,0/
01340       END
```

```
01350 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01360 C
01370       SUBROUTINE VELVEC
01380 C
01390 C   ROUTINE WHICH PLOTS THE VELOCITY VECTORS.
01400 C
01410       COMMON /AA/GRDSZ,GRD,NWF1,MAXSPD,SCALE,SPDSCL,NOPLTS,LP,MP,
01420      & DL,ANGT,SPEED(650),DIREC(650),TITLE(20),TIME,LEV,
01430      & ISKIP,INC,VECTOR,LPMP,PRT,FEATHL,FEATHD,ARROWL,SUNITS,DUNITS
01440       REAL MAXSPD,LEV
01450       LOGICAL PRT,VECTOR,SCALE
01460 C   PLOT THE VELOCITY FIELD EXCLUDING THOSE VECTORS
01470 C   WHICH ARE TOO SMALL OR ARE ACTUALLY OVER LAND.
01480       LM=0
01490       DO 20 M=1,MP
01500       DO 20 L=1,LP
01510       LM=LM+1
01520       DIR=DIREC(LM)*0.0174533
01530       X1=(FLOAT(L)-0.5)*GRDSZ
01540       Y1=(FLOAT(M)-0.5)*GRDSZ
01550 C   PLOT VELOCITY VECTOR IN ONE OF TWO WAYS DEPENDING ON
01560 C   USER PREFERENCE.
01570       IF(VECTOR) CALL VECTR(X1,Y1,DIR,LM)
01580       IF(.NOT.VECTOR) CALL ARRW(X1,Y1,DIR,LM)
01590 20    CONTINUE
01600       RETURN
01610       END
01620 C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
01630 C
01640       SUBROUTINE NASPLT
01650 C
01660 C ROUTINE WHICH PLOTS THE NORTH ARROWS
01670 C AND DISTANCE AND SPEED SCALES.
01680 C
01690       COMMON /AA/GRDSZ,GRD,NWF1,MAXSPD,SCALE,SPDSCL,NOPLTS,LP,MP,
01700      & DL,ANGT,SPEED(650),DIREC(650),TITLE(20),TIME,LEV,
01710      & ISKIP,INC,VECTOR,LPMP,PRT,FEATHL,FEATHD,ARROWL,SUNITS,DUNITS
01720       REAL MAXSPD,LEV
01730       LOGICAL PRT,VECTOR,SCALE
01740 C
01750       RANGT=0.0174533*ANGT
01760       ATLEN=.5
01770       ICODE=16
01780 C   DETERMINE THE ORIENTATION OF THE NORTH ARROW
01790 C   FOR PROPER SCALE PLACEMENT.
01800       ORIENT=1.
01810       IF((-1.0.LT.TAN(RANGT)).AND.(TAN(RANGT).LE.1.0)) ORIENT=2.
01820       IF(ORIENT.EQ.1.) GO TO 10
```

```
01830       IF(COS(RANGT).LT.0.) ORIENT=(-2.)
01840       GO TO 20
01850    10 IF(SIN(RANGT).LT.0.) ORIENT=(-1.)
01860 C    PLOT THE NORTH ARROW.
01870    20 AHLEN=0.16*ATLEN
01880       AHWID=0.4*AHLEN
01890       X2=ATLEN*SIN(RANGT)
01900       Y2=ATLEN*COS(RANGT)
01910       CALL ARROW(0.0,0.0,X2,Y2,AHLEN,AHWID,ICODE)
01920 C    PLOT THE SCALES
01930       DELTAV = DL/(1000.*GRDSZ)
01940       CALL AXIS(0.,1.25,'KM',2,1.0,90.,0.,DELTAV)
01950       DELTAV = GRDSZ/MAXSPD
01960       IF(.NOT.VECTOR) DELTAV=FEATHL*SPDSCL/GRDSZ
01970       CALL AXIS(0.,3.5,'VEL (M/S)',9,1.0,90.,0.,DELTAV)
01980       IF(VECTOR)GOTO 999
01990       CALL SYMBOL(.75,0.0,0.14,'1 FEATHER = ',90.,12)
02000       CALL NUMBER(999.,999.,0.14,SPDSCL,90.,3)
02010       CALL SYMBOL(999.,999.,0.14,' M/S',90.,4)
02020 999   RETURN
02030       END
02040 C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02050 C
02060       SUBROUTINE PLABL
02070 C
02080 C ROUTINE WHICH IDENTIFIES THE PLOT IN TIME
02090 C AND BY DEPTH LEVEL AND BY TITLE.
02100 C
02110       COMMON /AA/GRDSZ,GRD,NWF1,MAXSPD,SCALE,SPDSCL,NOPLTS,LP,MP,
02120      & DL,ANGT,SPEED(650),DIREC(650),TITLE(20),TIME,LEV,
02130      & ISKIP,INC,VECTOR,LPMP,PRT,FEATHL,FEATHD,ARROWL,SUNITS,DUNITS
02140       REAL LEV,MAXSPD
02150       LOGICAL PRT,VECTOR,SCALE
02160 C
02170 C   PLOT THE TITLE
02180       CALL SYMBOL(0.0,0.0,0.14,TITLE,90.,80)
02190 C   PLOT THE TIME.
02200       CALL SYMBOL(0.5,0.0,0.14,'TIME =',90.,6)
02210       CALL NUMBER(999.,999.,0.14,TIME,90.,2)
02220       CALL SYMBOL(999.,999.,0.14,' H    ',90.,6)
02230 C   PLOT THE DEPTH.
02240       CALL SYMBOL(999.,999.,0.14,'  DEPTH =',90.,9)
02250       CALL NUMBER(999.,999.,0.14,LEV,90.,1)
02260       CALL SYMBOL(999.,999.,0.14,' %',90.,2)
02270       RETURN
02280       END
02290 C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02300 C
```

```
02310       SUBROUTINE SKIP (NSKIP,*)
02320 C
02330 C  ROUTINE WHICH SKIPS DATA ON WORK FILE WHICH IS NOT TO BE PLOTTED
02340 C
02350       COMMON /AA/GRDSZ,GRD,NWF1,MAXSPD,SCALE,SPDSCL,NOPLTS,LP,MP,
02360      & DL,ANGT,SPEED(650),DIREC(650),TITLE(20),TIME,LEV,
02370      & ISKIP,INC,VECTOR,LPMP,PRT,FEATHL,FEATHD,ARROWL,SUNITS,DUNITS
02380       REAL LEV,MAXSPD
02390       LOGICAL PRT,VECTOR,SCALE
02400 C
02410       DO 10 I=1,NSKIP
02420       CALL RDSPD(&998)
02430 10    CONTINUE
02440       GOTO 999
02450 998   RETURN1
02460 999   RETURN
02470       END
02480 C   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02490 C
02500       SUBROUTINE OUTPUT (LP,MP,Z,INDEX)
02510 C
02520 C
02530 C  ROUTINE WHICH PRINTS A TABLE.  MAIN OUTPUT ROUTINE USED IN THIS PROGR
02540 C
02550       DIMENSION Z(1)
02560       LPAGE=0
02570       MPAGE=(LP-1)/10+1
02580       LPI=LP*INDEX
02590       DO 800 MM=1,MPAGE
02600       LSTART=10*(MM-1)*INDEX+1
02610       LSTOP=10*MM*INDEX
02620       JSTOP=10*MM
02630       JSTART=10*(MM-1)+1
02640       IF(MM.EQ.MPAGE) LSTOP=LP*INDEX
02650       IF(MM.GT.1) WRITE(3,100)
02660 100   FORMAT('1',////,30X,'CONTINUED FROM PREVIOUS PAGE')
02670       WRITE(3,200)(J,J=JSTART,JSTOP)
02680 200   FORMAT(53X,'  X DIRECTION',/,' -Y-',6X,10(I2,10X))
02690       WRITE(3,300)
02700 300   FORMAT(' ---',6X,10('--',10X))
02710       DO 700 M=1,MP
02720       NN=MP+1-M
02730       LM=(NN-1)*LPI
02740       DO 500 N=1,INDEX
02750       LMINDX=LM+N-1
02760       WRITE(3,400) NN,(Z(LMINDX+L),L=LSTART,LSTOP,INDEX)
02770 400   FORMAT(' ',I2,3X,10(G11.4,1X))
02780 500   CONTINUE
```

```
02790       NPAGE=(M*INDEX)/48
02800       IF(NPAGE.EQ.LPAGE) GO TO 700
02810       LPAGE=NPAGE
02820       WRITE(3,600)
02830 600    FORMAT('1',////)
02840 700   CONTINUE
02850 800   CONTINUE
02860       RETURN
02870       END
02880 C
02890 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
02900 C
02910       SUBROUTINE GRID(XO,YO,DX,DY,LX,LY)
02920 C
02930 C  ROUTINE PLOTS A GRID.
02940 C
02950       COMMON /AA/GRDSZ,GRD,NWF1,MAXSPD,SCALE,SPDSCL,NOPLTS,LP,MP,
02960      & DL,ANGT,SPEED(650),DIREC(650),TITLE(20),TIME,LEV,
02970      & ISKIP,INC,VECTOR,LPMP,PRT,FEATHL,FEATHD,ARROWL,SUNITS,DUNITS
02980       REAL MAXSPD,LEV
02990       LOGICAL PRT,VECTOR,SCALE
03000 C
03010       CALL PLOT (XO,YO,3)
03020       LXP1=LX+1
03030       LYP1=LY+1
03040       XEND=LX*DX
03050       YEND=LY*DY
03060       DO 10 I=1,LYP1
03070       Y=(I-1)*DY
03080       CALL PLOT(0.,Y,3)
03090       CALL PLOT(XEND,Y,2)
03100 10    CONTINUE
03110       DO 20 I=1,LXP1
03120       X=(I-1)*DX
03130       CALL PLOT(X,0.,3)
03140       CALL PLOT(X,YEND,2)
03150 20    CONTINUE
03160       RETURN
03170       END
03180 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
03190 C
03200       SUBROUTINE VECTR(X1,Y1,DIR,LM)
03210 C
03220 C   ROUTINE TO DRAW A VELOCITY VECTOR AT GRID X1,Y1.
03230 C
03240       COMMON /AA/GRDSZ,GRD,NWF1,MAXSPD,SCALE,SPDSCL,NOPLTS,LP,MP,
03250      & DL,ANGT,SPEED(650),DIREC(650),TITLE(20),TIME,LEV,
03260      & ISKIP,INC,VECTOR,LPMP,PRT,FEATHL,FEATHD,ARROWL,SUNITS,DUNITS
```

```
      REAL MAXSPD,LEV
      LOGICAL PRT,VECTOR,SCALE
      SCLSPD=SPEED(LM)*GRDSZ/MAXSPD
      X2=X1+SCLSPD*COS(DIR)
      Y2=Y1+SCLSPD*SIN(DIR)
      AHLEN=0.375*SCLSPD
      CALL ARROW(-Y1,X1,-Y2,X2,AHLEN,0.,12)
C     CALL NUMBER (-Y1,X1,0.1,FLOAT(LM),90.,-1)
      RETURN
      END
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE ARRW(X1,Y1,DIR,LM)
C
C     ROUTINE WHICH PLOTS A CONSTANT LENGTH ARROW IN THE DIRECTION
C     OF THE VELOCITY VECTOR AND PLACES FEATHERS ON THE END WHICH
C     ARE PROPORTIONAL TO THE SPEED AT GRID X1,Y1.
C
      COMMON /AA/GRDSZ,GRD,NWF1,MAXSPD,SCALE,SPDSCL,NOPLTS,LP,MP,
     & DL,ANGT,SPEED(650),DIREC(650),TITLE(20),TIME,LEV,
     & ISKIP,INC,VECTOR,LPMP,PRT,FEATHL,FEATHD,ARROWL,SUNITS,DUNITS
      REAL MAXSPD,LEV
      LOGICAL PRT,VECTOR,SCALE
C
C   FLNGTH IS THE PLOTTED LENGTH OF A FEATHER.  ONE FEATHER LENGTH IS
C   EQUAL TO SPDSCL.  WHEN PLOTTED ONE FEATHER LNGTH EQUAL GRDSZ/FEATHL.
C   ALNGHT EQUAL THE LENGTH OF THE VELOCITY ARROW, A CONSTANT.
C   FEATHD EQUAL THE CONSTANT THAT DETERMINES HOW FAR APART THE FEATHERS
C   WILL BE DRAWN ON THE VELOCITY VECTOR.  ARROWL DETERMINES THE LENGTH
C   OF THE VELOCITY ARROW.
      IF(SPEED(LM).LT.(SPDSCL/50.)) GOTO 300
      ALNGTH=GRDSZ*ARROWL
      DIRC=DIREC(LM)+90.
      CALL SYMBOL(-Y1,X1,ALNGTH,13,DIREC(LM),-1)
      RNUM=SPEED(LM)/SPDSCL
      NUM=RNUM
      FLNGTH=GRDSZ/5.0
      X2=X1-ALNGTH*COS(DIR)*0.5
      Y2=Y1-ALNGTH*SIN(DIR)*0.5
      HEADL=GRDSZ/5.
      X3=X1+(ALNGTH)*COS(DIR)*0.5
      Y3=Y1+(ALNGTH)*SIN(DIR)*0.5
      CALL SYMBOL(-Y3,X3,HEADL,2,DIREC(LM),-1)
      IF(NUM.EQ.0) GOTO 200
      DO 100 N=1,NUM
      CALL SYMBOL(-Y2,X2,FLNGTH,13,DIRC,-1)
      X2=X2+ALNGTH/FEATHD*COS(DIR)
```

```
03750       Y2=Y2+ALNGTH/FEATHD*SIN(DIR)
03760 100   CONTINUE
03770 200   FLNGTH=FLNGTH*(SPEED(LM)-NUM*SPDSCL)/SPDSCL
03780       CALL SYMBOL(-Y2,X2,FLNGTH,13,DIRC,-1)
03790 300   RETURN
03800       END
03810 C
03820 C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
03830 C
03840       SUBROUTINE RDSPD(*)
03850 C
03860 C     READS IN SPEED AND DIREC ARRAYS AND
03870 C     DETERMINES THE MAXIMUM SPEED FOR SCALING.
03880 C
03890       COMMON /AA/GRDSZ,GRD,NWF1,MAXSPD,SCALE,SPDSCL,NOPLTS,LP,MP,
03900      & DL,ANGT,SPEED(650),DIREC(650),TITLE(20),TIME,LEV,
03910      & ISKIP,INC,VECTOR,LPMP,PRT,FEATHL,FEATHD,ARROWL,SUNITS,DUNITS
03920       REAL MAXSPD,LEV
03930       LOGICAL PRT,VECTOR,SCALE
03940 C
03950       READ(NWF1,END=998) TIME,LEV,(SPEED(LM),DIREC(LM),LM=1,LPMP)
03990       DO 5 LM=1,LPMP
04000       SPEED(LM)=SPEED(LM)*SUNITS
04010       DIREC(LM)=DIREC(LM)*DUNITS
04020 5     CONTINUE
04030       IF(.NOT.PRT) GOTO 9
04040       WRITE(3,6) TIME,LEV
04050 6     FORMAT(////,'1 TIME=',G10.3,10X,'PLAN VIEW AT LEVEL = ',G10.3)
04060       CALL OUTPUT(LP,MP,SPEED,1)
04070       CALL OUTPUT(LP,MP,DIR,1)
04080 9     MAXSPD = SPDSCL
04090       IF(.NOT.SCALE) GO TO 999
04100       MAXSPD=0.
04110       DO 10 LM=1,LPMP
04120       MAXSPD=AMAX1(MAXSPD,SPEED(LM))
04130       IF(.NOT.VECTOR) SPDSCL=MAXSPD/5.0
04140 10    CONTINUE
04150       GOTO 999
04160 998   RETURN1
04170 999   RETURN
04180       END
04190 C$ENTRY
EOF:
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                    PROGRAM WIND(GAL)
C
C          VERSION 7 - July 1982 - REVISION A
C
C                 BY CORTIS COOPER, NECE
C
C     PROGRAM WHICH CALULATES WIND FRICTION VELOCITIES SQUARED FOR
C     CASE WHERE WIND DOES NOT VARY SPATIALLY BUT DOES VARY
C     TEMPORALLY.
C
      DIMENSION UA2(600),VA2(600),TIT(20)
      NAMELIST/MISC/IT0,WTHETA,WT0,IDTT,MAXT,LP,MP
      DATA UA2,VA2,/600*0.0,600*0.0/,IT0,WTHETA,WT0,IDTT,MAXT,LP,MP/
     &     3600,0,10,3600,72000,12,24/
C
      READ(1,100) TIT
100   FORMAT(20A4)
      READ(1,MISC)
      WRITE(4) TIT
      LPMP=LP*MP
      MAXIT=MAXT/IDTT+1
      IF(IT0.LT.1) IT0=1
      IF(IT0.LE.MAXT) GOTO 120
      WRITE(3,50)
50    FORMAT(///,' * * * * * ERROR:  IT0.GT.MAXT * * * * * *')
      GOTO 600
120   DO 500 I=1,MAXIT
      ITIME=I*IDTT
      WTIME=ITIME/3600.
C  CALCULATES WIND AT NEW TIME USING A LINEAR RAMP FUNCTION IN TIME.
      W=WT0*ITIME/(FLOAT(IT0))
      IF(ITIME.LE.IT0) GOTO 150
      I=MAXIT
      WTIME=1.0E32
      W=WT0
C  CALCULATES WIND STRESS COEFFICIENTS ACCORDING TO WU'S FORMULAS.
150   RKAPPA=(.95375+0.0775*W)*1.E-06
C  CALCULATES WIND FRICTION VELOCITY SQUARED.
      U=COS(WTHETA*3.14159/180.0)*RKAPPA*W**2
      V=SIN(WTHETA*3.14159/180.0)*RKAPPA*W**2
      DO 200 LM=1,LPMP
      UA2(LM)=U
      VA2(LM)=V
200   CONTINUE
      WRITE(4) WTIME,(UA2(LM),LM=1,LPMP),(VA2(LM),LM=1,LPMP)
```

```
      WRITE(3,300) WTIME,W
300   FORMAT(' TIME=,',G14.4,5X,'SPEED=',G14.4)
500   CONTINUE
600   STOP
      END
```

```
00010 C            *** * ***  SPATIAL 7.A   *** * ***
00020 C  MAKES SPATIAL INTERPOLATION OF WIND FIELD OVER A GAL MODEL GRID
00030 C  GIVEN DATA AT SEVERAL KNOWN POINTS.  PROGRAM IS SET UP
00040 C  FOR FIVE (5) MET STATION INPUTS.  FOR MORE MET STATIONS,
00050 C  REDIMENSION VARIABLE 'WT' AND REVISE I/O FORMATS.
00060 C
00070 C  LAST REV: 13 APRIL 1982, CKC, CHANGED STRESS CALC.
00080 C                    FROM WU (1969) TO WU (1980)
00090 C  PREVIOUS REV: 6 JAN 1982
00100 C  AUTHOR:  ADRIAN C. HUMPHREYS, III, NECE
00110 C
00120 C  DEFINITION OF VARIABLES:
00130 C  LP            = NO. OF GRID ELEMENTS IN Y DIRECTION.
00140 C  MP            = NO. OF GRID ELEMENTS IN X DIRECTION.
00150 C  N             = NO. OF METEOROLOGICAL STATIONS (MAX. OF 10)
00160 C  IX(I),IY(I)   = GRID COORDINATES OF MET STATIONS.
00170 C  UM(I),VM(I)   = U&V COMPONENTS OF VELOCITY AT MET STATION 'I'.
00180 C  IUM(I),IVM(I)= DITTO,  BUT IN INTEGER NODC FORMAT.
00190 C  W(I,K)        = WEIGHTING FACTOR FOR MET STA 'I', AT GRID ELEMENT 'K'.
00200 C  LTIME         = NO. OF SEQUENTIAL WIND FIELDS TO BE CREATED.
00210 C  U(I),V(I)     = INTERPOLATED VELOCITY COMPONENTS AT GRID
00220 C                   ELEMENT 'I'.  WRITTEN TO DISK AND DISCARDED
00230 C                   FROM MEMORY AFTER EACH TIME STEP.
00240 C
00250       IMPLICIT REAL (A-H,O-Z)
00260       COMMON /ONE/  LP,MP,LPMP,IX(5),IY(5),N,SYSW(5),INT
00270       COMMON /TWO/  IUM(5),IVM(5),UM(5),VM(5),W(5,288)
00280       COMMON /THREE/  ICARD,IN1,IOUT1,IOUT2,IPRT
00290       INTEGER A(20)
00300       DIMENSION U(288),V(288)
00310       DIMENSION WT(5)
00320       DATA U,V /288*0.0,288*0.0/
00330       DATA KNT/0/,LIMIT/999/
00340       NAMELIST /GRID/LP,MP,N,IX,IY,SYSW,INT
00350 C
00360       READ(ICARD,GRID)
00370       LPMP = LP*MP
00380       KNT=KNT+1

00390 C  COMPUTE WEIGHTING FACTORS (N*LPMP VALUES).  ONE FOR EACH GRID
00400 C  ELEMENT FOR EACH MET STATION. . .
00410       DO 100 L=1,LP
00420       DO 100 M=1,MP
00430       LM=MP*(L-1)+M
00440       CALL WEIGHT (L,M,LM,WT)
00450       DO 50 INDEX=1,N
00460       W(INDEX,LM)=WT(INDEX)*SYSW(INDEX)
```

```
00470 C     IF(LM.LE.9)  WRITE(IPRT,444) INDEX,LM,W(INDEX,LM)
00480   444 FORMAT(1X,'WEIGHT(',I1, ',', I3,') = ',F10.5)
00490    50 CONTINUE
00500   100 CONTINUE
00510 C
00520 C  READ HEADER FROM DATA FILE . . .
00530       READ(IN1,810,END=900,ERR=900) A
00540   810 FORMAT(20A4)
00550 C  WRITE HEADER TO UNFORMATTED DISK . . .
00560       WRITE(IOUT1) A
00570 C  WRITE HEADER TO OUTPUT .MONITOR FILE . . .
00580       WRITE(IPRT,810) A
00590 C  BEGIN MAIN INTERPOLATION LOOP . . .
00600       DO 300 ITIME=1,LIMIT,INT
00610 C  READ WIND VELOCITY FOR THIS TIME STEP AT EACH MET STATION FROM
00620 C  SPECIAL DATA FILE CONTAINING U & V COMPONENTS RELATIVE TO
00630 C  GRID NORTH . . .
00640       READ(IN1,820,END=900,ERR=900) (IUM(ISTA),IVM(ISTA),ISTA=1,N)
00650   820 FORMAT(10I5)
00660 C  SKIPS NUMBER IN INPUT RECORD = INT-1
00670       IF(INT.EQ.1) GO TO 120
00680       DO 115 I=2,INT
00690       READ(IN1,820,END=900,ERR=900) DUM
00700   115 CONTINUE
00710 C
00720   120 DO 130  J1=1,N
00730       UM(J1) = FLOAT(IUM(J1))/100.
00740       VM(J1) = FLOAT(IVM(J1))/100.
00750   130 CONTINUE
00760 C  INTERPOLATE . . . 'I' IS THE MET STA COUNTER, 'K' IS THE
00770 C  GRID ELEMENT COUNTER . . .
00780       DO 200 K=1,LPMP
00790       VSUM=0.0
00800       USUM=0.0
00810       DO 150  I=1,N
00820       USUM = USUM + W(I,K)*UM(I)
00830       VSUM = VSUM + W(I,K)*VM(I)
00840   150 CONTINUE
00850       U(K)=USUM
00860       V(K)=VSUM
00870   200 CONTINUE
00880       CALL STRESS(U,V,ITIME)
00890       KNT=KNT+1
00900   300 CONTINUE
00910       GO TO 999
00920   900 WRITE(IPRT,910)  KNT
00930   910 FORMAT(//1X,'READING TERMINATED DUE TO EOF OR ERROR',
00940      &/1X,' AFTER ',I6,' RECORDS.')
```

```
00950  999  STOP
00960       END
00970 C  *** * *** * *** * ***
00980       BLOCK DATA
00990       IMPLICIT REAL (A-H,O-Z)
01000       COMMON /ONE/  LP,MP,LPMP,IX(5),IY(5),N,SYSW(5),INT
01010       COMMON /TWO/  IUM(5),IVM(5),UM(5),VM(5),W(5,288)
01020       COMMON /THREE/  ICARD,IN1,IOUT1,IOUT2,IPRT
01030       DATA ICARD,IN1,IOUT1,IOUT2,IPRT /1,1,4,3,3/
01040       DATA IX,IY /5*0,5*0/
01050       DATA UM /5*0.0/, VM /5*0.0/, W /1440*0.0/
01060       DATA IUM /5*0/, IVM /5*0/
01070       DATA SYSW /5*1.0/
01080       DATA INT,LP,MP,N/1,12,24,5/
01090       END
01100 C  *** * *** * *** *
01110       SUBROUTINE WEIGHT (L,M,LM,WT)
01120       IMPLICIT REAL (A-H,O-Z)
01130       COMMON /ONE/  LP,MP,LPMP,IX(5),IY(5),N,SYSW(5),INT
01140       COMMON /TWO/  IUM(5),IVM(5),UM(5),VM(5),W(5,288)
01150       COMMON /THREE/  ICARD,IN1,IOUT1,IOUT2,IPRT
01160       DIMENSION R(5),WT(5)
01170 C  COMPUTES RADIAL DISTANCE FROM CENTER OF EACH GRID
01180 C  ELEMENT TO POINTS WHOSE LOCATION IS KNOWN WRT THE
01190 C  GRID AXES.  THEN COMPUTES WEIGHTING FACTORS, WHERE:
01200 C  EPS    = SMALLNESS TEST FACTOR FOR PIVOT ELEMENT IN 'SIMUL'.
01210 C  IX(I),IY(I)= THE COORDINATES IN GRID SPACE OF MET STA 'I'.
01220 C  L, M         = THE COORDINATES OF THE CENTER OF A GRID ELEMENT.
01230 C
01240       EPS=1.0E-07
01250       NP1 = N+ 1
01260 C  INITIALIZE WEIGHTS . . .
01270       DO 50 I=1,N.
01280       WT(I)=0.0
01290    50 CONTINUE
01300 C  COMPUTE RADIAL DISTANCES . . .
01310       DO 100 K=1,N
01320       R(K) = SQRT( FLOAT(M-IX(K))**2 + FLOAT(L-IY(K))**2 )
01330 C  IF RADIAL DISTANCE FROM THIS GRID ELEMENT TO MET STA 'K' IS
01340 C  SMALL, WEIGHT THIS MET STA 100 PERCENT.
01350       IF(R(K) .GT. 0.01)  GO TO 100
01360       WT(K) = 1.0
01370       RETURN
01380   100 CONTINUE
01390 C
01400 C  NOW SET UP MATRIX & SOLVE L.EQ'S FOR WEIGHTING FACTORS.
01410 C       WT(1) +      WT(2) +      WT(3) +      WT(4) = 1.0
01420 C  R(1)*WT(1) - R(2)*WT(2)                          = 0.0
```

```
01430 C  R(1)*WT(1)              - R(3)*WT(3)              = 0.0
01440 C  R(1)*WT(1)                                - R(4)*WT(4) = 0.0
01450 C
01460 C  THESE REDUCE TO . . .
01470 C  WT(2) = ( R(1)/R(2) )*WT(1)
01480 C  WT(3) = ( R(1)/R(3) )*WT(1)
01490 C  WT(4) = ( R(1)/R(4) )*WT(1), ETC . . .
01500 C  AND . . .
01510 C  WT(1)*( 1 + R(1)/R(2) + R(1)/R(3) + R(1)/R(4) + . . . = 1.0
01520 C
01530 C  SOLVING FOR WT(1) . . .
01540       SUM = 1.0
01550       DO 250 I=2,N
01560       SUM = SUM + R(1)/R(I)
01570   250 CONTINUE
01580       WT(1) = 1.0/SUM
01590 C  SOLVE FOR REMAINING WEIGHTS . . .
01600       DO 350 I=2,N
01610       WT(I) = ( R(1)/R(I) )*WT(1)
01620   350 CONTINUE
01630       RETURN
01640       END
01650 C   *** * *** * *** * *** *
01660       SUBROUTINE STRESS(U,V,ITIME)
01670 C  COMPUTES WIND STRESS AT EACH GRID ELEMENT GIVEN VELOCITY COMPONENTS
01680 C  AND WRITES VALUES TO UNFORMATTED DISK.
01690 C
01700       IMPLICIT REAL (A-H,O-Z)
01710       COMMON /ONE/  LP,MP,LPMP,IX(5),IY(5),N,SYSW(5),INT
01720       COMMON /THREE/  ICARD,IN1,IOUT1,IOUT2,IPRT
01730       DIMENSION U(288),V(288),USTAR(288),VSTAR(288)
01740       WTIME=FLOAT(ITIME)
01750       DO 100  I=1,LPMP
01760       SPD=SQRT( U(I)**2 + V(I)**2 )
01770       RKAPPA=(.95375+0.0775*SPD)*1.E-06
01780       USTAR(I) = U(I)*RKAPPA*SPD
01790       VSTAR(I) = V(I)*RKAPPA*SPD
01800       IF(USTAR(I).LE.1.0E-02 .AND. VSTAR(I).LE.1.0E-02)  GO TO 100
01810       WRITE(IPRT,870) WTIME,I,USTAR(I),VSTAR(I)
01820   870 FORMAT(//1X,'JOB TERMINATED, USTAR OR VSTAR TOO LARGE',
01830      &/1X,'TIME = ',F5.0,'  GRID ELEMENT = ',I3,
01840      &/1X,'USTAR = ',1E12.3,'  VSTAR = ',1E12.3,//)
01850       STOP
01860   100 CONTINUE
01870       WRITE(IOUT1) WTIME,(USTAR(LM),LM=1,LPMP),(VSTAR(LM),LM=1,LPMP)
01880       WRITE(IPRT,851) WTIME
01890   851 FORMAT(1X,'TIME = ',F5.0)
01900 C     WRITE(IPRT,854)
```

```
01910   854 FORMAT(1X,'USTAR(I),VSTAR(I),I=1,240')
01920 C     WRITE(IPRT,852) (USTAR(LM),VSTAR(LM), LM=1,LPMP)
01930   852 FORMAT(1X,6E12.3)
01940 C     WRITE(IPRT,853)
01950   853 FORMAT(/1X,'WIND SPEED, U & V COMPONENTS')
01960 C     WRITE(IPRT,852) (U(LM),V(LM), LM=1,LPMP)
01970       RETURN
01980       END
01990 C  *** * *** * *** * *** * *** *
02000 C$ENTRY
EOF:
2.
```

```
C               ***  **   DENSTAT   **  ***
C  COMPUTES STATISTICS ON DENSITY FROM NODC OCEANOGRAPHIC TAPE.
C  AUTHOR:   A. C. HUMPHREYS, NEW ENGLAND COASTAL ENGINEERS
C  DATE:     23 MARCH 1982
C  LAST REV: 16 APRIL 1982
C
      COMMON /ONE/ SIGSUM(10,8,4),KNTS(10,8,4),ISIGMA(10,8,4,2000),
     &     VAR(10,8,4),DEV(10,8,4),RANGE(10,8,4),AVG(10,8,4),ISUM(10,8)
      COMMON /TWO/ IERR1,IERR2,IERR3,IERR4,IERR5,ICODE
      COMMON /TRE/ ISTIME,IETIME
      INTEGER*2 ISIGMA
      INTEGER IDEP(4),ISIG(4)
      NAMELIST /MISC/ISTIME,IETIME
      DATA  LOCN,LOCW /2*99/
      DATA JDEP,JSIG /2*9999/
C
      READ(1,MISC)
      ITEST=1
      DO 500 ITER=1,36000
      READ(13,10,END=600) LAT,LONG,MODAY,IDEP,ISIG,IREC
   10 FORMAT(4X,I5,I6,5X,I4,3X,4A1,11X,4A1,33X,I1)
      IF(IREC .EQ. 1) GO TO 500
C
C  WRITE A COUNTER TO DISK FOR INFO IN CASE OF CRASH . .
      ITER2=ITER/1000
      IF(ITER2 .LT. ITEST) GO TO 20
      ITEST=ITEST+1
      WRITE(23,15) ITER
   15 FORMAT('PROGRAM DENSTAT2 PROCESSED AT LEAST ', I10,' RECORDS')
   20 CONTINUE
C
C  REJECT DATA POINT IF NOT WITHIN DESIRED TIME SPAN . . .
      IF(IABS(ISTIME-IETIME) .LT. 28) GO TO 800
      IF(ISTIME .LT. IETIME) GO TO 25
C  . . . FOR SPANS BEGINNING LATE IN YEAR, ENDING EARLY . . .
      IF(MODAY .GE. ISTIME .OR.  MODAY .LE. IETIME) GO TO 30
      GO TO 497
C  . . . FOR SPANS BEGINNING EARLY, ENDING LATE . . .
   25 IF(MODAY .GE. ISTIME .AND. MODAY .LE. IETIME) GO TO 30
      GO TO 497
   30 CALL LOCATE(LAT,LONG,LOCN,LOCW)
      IF(LOCN .EQ. 99 .OR. LOCW .EQ. 99) GO TO 470
      CALL ALFNUM(IDEP,JDEP)
      IF(JDEP .EQ. 9999) GO TO 480
      CALL DEEPS(JDEP,LOCD)
      IF(LOCD .EQ. 5) GO TO 495
      CALL ALFNUM(ISIG,JSIG)
```

```
      IF(JSIG .EQ. 9999) GO TO 490
      IF(JSIG .LT. 1000 .OR. JSIG .GT. 3000) GO TO 490
C     IF(MODAY .GE. 0401 .AND. MODAY .LT. 0930)
C    & ISUM(LOCN,LOCW)=ISUM(LOCN,LOCW)+1
C
      KNTS(LOCN,LOCW,LOCD)       = KNTS(LOCN,LOCW,LOCD) + 1
      KNT                        = KNTS(LOCN,LOCW,LOCD)
      ISIGMA(LOCN,LOCW,LOCD,KNT) = JSIG
      SIGSUM(LOCN,LOCW,LOCD) = SIGSUM(LOCN,LOCW,LOCD)
     &                       + FLOAT(JSIG)/100.0
C
      GO TO 500
  470 IERR1=IERR1 + 1
      GO TO 500
  480 IERR2=IERR2 + 1
      GO TO 500
  490 IERR3=IERR3 + 1
      GO TO 500
  495 IERR4=IERR4 + 1
      GO TO 500
  497 IERR5=IERR5 + 1
C
  500 CONTINUE
      ICODE=1
      ITER=ITER-1
      GO TO 610
  600 CONTINUE
      ITER = ITER-1
  610 CALL REPORT(ITER)
C
      CALL STAT2(SIGSUM,KNTS,ISIGMA)
C     CALL STAT(SIGSUM,KNTS,ISIGMA,AVG,VAR,DEV,RANGE)
C     CALL OUTPUT(KNTS,AVG,VAR,DEV,RANGE)
C
      GO TO 999
  800 WRITE(3,810) ISTIME,IETIME
  810 FORMAT(//1X,'LESS THAN ONE MONTH DATA REQUESTED, EXECUTION TERMINATED',
     &/,'FIRST DATE= ',I4,'  SECOND DATE= ',i4)
  999 STOP
      END
C
C  ***  **  ***  **  ***  **  ***
C
      BLOCK DATA
C
      COMMON /ONE/ SIGSUM(10,8,4),KNTS(10,8,4),ISIGMA(10,8,4,2000),
     &     VAR(10,8,4),DEV(10,8,4),RANGE(10,8,4),AVG(10,8,4),ISUM(10,8)
      COMMON /TWO/ IERR1,IERR2,IERR3,IERR4,IERR5,ICODE
```

```
      COMMON /TRE/ ISTIME,IETIME
C
      INTEGER*2 ISIGMA
      DATA SIGSUM /320*0.0/, VAR/320*0.0/,DEV/320*0.0/,RANGE/320*0.0/
      DATA KNTS   /320*0/, AVG/320*0.0/
      DATA ISIGMA /640000*0/
      DATA IERR1,IERR2,IERR3,IERR4,IERR5,ICODE /6*0/
      DATA ISUM /80*0/
      DATA ISTIME, IETIME /0401,0930/
      END
C
C  ***  **  ***  **  ***  **  ***
C
      SUBROUTINE LOCATE(LAT,LONG,LOCN,LOCW)
C
C  ESTABLISHES GRID LOCATION BASED ON LATITUDE AND LONGITUDE
C
      CALL CONVRT(LAT,DEGN)
      CALL CONVRT(LONG,DEGW)
C
C  25 DEGREES IS MINIMUM LATITUDE, 30 IS MAX . . .
      A=25.0
      DO 30 I=1,10
      B = A + 0.5
      IF(DEGN .GE. A .AND. DEGN .LT. B) GO TO 50
      A=B
  30  CONTINUE
C
C  DATA POINT OUT OF GEOGRAPHIC RANGE, ASSIGN ERROR VALUE TO LOCN
      LOCN=99
      RETURN
C
  50  LOCN = I
C  85 DEGREES IS MAXIMUM LONGITUDE, 81 IS MIN . . .
      A=85.0
      DO 60 J=1,8
      B = A - 0.5
      IF(DEGW .LE. A .AND. DEGW .GT. B) GO TO 70
      A=B
  60  CONTINUE
C
C  DATA POINT OUT OF GEOGRAPHIC RANGE, ASSIGN ERROR VALUE TO LOCW
      LOCW=99
      RETURN
C
  70  LOCW=J
C
C  REPORT LOCN & LOCW AS GRID LOCATIONS OF THIS DATA POINT . . .
```

```
      RETURN
      END
C
C  ***  **  ***  **  ***  **  ***
C
      SUBROUTINE CONVRT(INTEG,DEG)
C
C  CONVERTS INTEGER DEGREES & MINUTES (I6) TO DECIMAL DEGREES
C
      NDEG = INTEG/1000
      MIN  = INTEG - NDEG*1000
      DEG  = FLOAT(NDEG) + FLOAT(MIN)/600.0
      RETURN
      END
C  ***  **  ***  **  ***  **  ***
C
      SUBROUTINE ALFNUM(ALF,NUM)
C
C  CONVERTS AN ALPHANUMERIC VALUE TO INTEGER
C
      INTEGER A(10) /'&','J','K','L','M','N','O','P','Q','R'/
      INTEGER D(10) /'0','1','2','3','4','5','6','7','8','9'/
      INTEGER B     /' '/
      INTEGER ALF(4),INTEG(4)
C
      DO 100 I=1,4
      DO  50 J=1,10
      IF( ALF(I) .NE. D(J) ) GO TO 50
      INTEG(I) = J-1
      GO TO 100
  50  CONTINUE
C     IF(I .NE. 4) GO TO 90
C CHECK LAST CHARACTER FOR 11 OVERPUNCH & TRANSLATE
C     DO 60 K=1,10
C     IF(ALF(I) .NE. A(K)) GO TO 60
C     INTEG(I)=K-1
C     GO TO 100
C 60  CONTINUE
  90  IF( ALF(I) .NE. B) GO TO 200
      INTEG(I) = 0
 100  CONTINUE
C
      NUM = 1000*INTEG(1)+100*INTEG(2)+10*INTEG(3)+INTEG(4)
      RETURN
C
C  ORIGINAL VALUE WAS NOT NUMBER, REPORT ERROR CODE . . .
 200  NUM=9999
      RETURN
```

```
      END
C
C  ***  **  ***  **  ***  **  ***
C
      SUBROUTINE DEEPS(JDEP,LD)
C
C  DETERMINES WHERE IN DEPTH RANGE THIS DATA POINT LIES.  ASSIGNS
C  DATA POINT TO A DEPTH CATEGORY FROM ONE (SHALLOW) TO FIVE (OVER
C  100 M. DEPTH).
C
      INTEGER A,B
      A=0
C
      DO 100 I=1,4
      B = A + 20
      IF(I .EQ. 4) B=100
      IF(JDEP .GE. A .AND. JDEP .LT. B) GO TO 150
      A=B
  100 CONTINUE
C
C  DEPTH IS GREATER THAN 100 M.
      LD=5
      RETURN
C
  150 LD=I
      RETURN
      END
C
C  ***  **  ***  **  ***  **  ***
      SUBROUTINE REPORT(ITER)
C
C  REPORTS ERRORS IN PROCESSING AT TOP OF PRINTOUT
C
      COMMON /TWO/ IERR1,IERR2,IERR3,IERR4,IERR5,ICODE
C
      IF(ICODE .EQ. 0)  WRITE(3,5) ITER
      IF(ICODE .EQ. 1)  WRITE(3,7) ITER
    5 FORMAT(//'1','END OF FILE AFTER ',I6,' RECORDS')
    7 FORMAT(//1X,'NORMAL TERMINATION AFTER ',I6,' RECORDS',
     &/1X,'END OF FILE NOT REACHED')
      WRITE(3,10) IERR1
   10 FORMAT(//1X,I6,' DATA POINTS WERE OUT OF GEOGRAPHIC RANGE')
      WRITE(3,20) IERR2
   20 FORMAT(1X,I6,' DATA POINTS REFERRED TO GARBLED DEPTHS')
      WRITE(3,30) IERR3
   30 FORMAT(1X,I6,' DATA POINTS REFERRED TO GARBLED SIGMA TEE VALUES')
      WRITE(3,35) IERR4
   35 FORMAT(1X,I6,' DATA POINTS WERE BELOW 100 METRES DEPTH')
```

```
      WRITE(3,37) IERR5
   37 FORMAT(1X,I6,' DATA POINTS WERE NOT IN DESIRED TIME SPAN')
      ITOT=IERR1+IERR2+IERR3+IERR4+IERR5
      WRITE(3,40) ITOT
   40 FORMAT(1X,I6,' TOTAL VALUES WERE NOT COUNTED IN THIS COMPILATION')
C
      RETURN
      END
C
C  *** **  *** **  *** **  ***
C
      SUBROUTINE STAT(SIGSUM,KNTS,ISIGMA,AVG,VAR,DEV,RANGE)
C
C  COMPUTES ELEMENTARY STATISTICS ON ACCUMULATED DATA VALUES.
C
      REAL SIGSUM(10,8,4),AVG(10,8,4),
     &     VAR(10,8,4),DEV(10,8,4),RANGE(10,8,4)
      INTEGER*2 ISIGMA(10,8,4,2000)
      INTEGER KNTS(10,8,4)
C
      DO 200 J=1,10
      DO 180 K=1,8
      DO 160 L=1,4
C
      IF(KNTS(J,K,L) .LT. 1) GO TO 120
      AVG(J,K,L) = SIGSUM(J,K,L)/FLOAT(KNTS(J,K,L))
      GO TO 125
  120 AVG(J,K,L) = 0.0
  125 CONTINUE
      SIGMIN = 100.0
      SIGMAX = 0.0
      SS     = 0.0
      IEND   = KNTS(J,K,L)
      IF(IEND .EQ. 0)  IEND=1
C
      DO 140 M=1,IEND
      SIGMA = ISIGMA(J,K,L,M)/100.0
      SS = SS + (SIGMA - AVG(J,K,L))**2
      SIGMAX = AMAX1(SIGMAX,SIGMA)
      SIGMIN = AMIN1(SIGMIN,SIGMA)
  140 CONTINUE
C
      IF(KNTS(J,K,L) .LE. 1) GO TO 145
      VAR(J,K,L) = SS/(KNTS(J,K,L)-1)
      GO TO 150
  145 VAR(J,K,L) = 0.0
  150 DEV(J,K,L) = SQRT(VAR(J,K,L))
      RANGE(J,K,L)= SIGMAX-SIGMIN
```

```
C
  160 CONTINUE
  180 CONTINUE
  200 CONTINUE
C
      RETURN
      END
C
C  ***  **  ***  **  ***  **  ***
C
      SUBROUTINE OUTPUT(KNTS,AVG,VAR,DEV,RANGE)
C
C  WRITES SUMMARY OF DATA IN TABULAR FORM, BY DEPTH
C
      COMMON /TRE/ ISTIME,IETIME
      REAL SIGSUM(10,8,4),AVG(10,8,4),
     &     VAR(10,8,4),DEV(10,8,4),RANGE(10,8,4)
      INTEGER KNTS(10,8,4)
      INTEGER A,B
C
      A=0
      DO 200 L=1,4
      B = A + 10
      IF(L .EQ. 4) B=100
C     IF(L .EQ. 5) GO TO 15
      WRITE(3,10) A,B
   10 FORMAT(//'1','STATISTICS ON SIGMATEE FOR DEPTHS BETWEEN ',
     &      I6,' AND',I6,' METRES')
C     GO TO 20
C 15  WRITE(3,17)
C 17  FORMAT(//1X,'STATISTICS ON SIGMATEE FOR DEPTHS GREATER THAN 100',
C    &      ' METRES')
C 20  CONTINUE
C
      J=11
      DO 180 J1=1,10
      J=J-1
      WRITE(3,80)  (AVG(J,K,L),   K=1,8)
      WRITE(3,82)  (VAR(J,K,L),   K=1,8)
      WRITE(3,84)  (DEV(J,K,L),   K=1,8)
      WRITE(3,86)  (RANGE(J,K,L), K=1,8)
      WRITE(3,88)  (KNTS(J,K,L),  K=1,8)
   80 FORMAT(/1X,'MEAN',     T15,8F8.2)
   82 FORMAT(1X,'VARIANCE', T15,8F8.2)
   84 FORMAT(1X,'STD. DEV.',T15,8F8.2)
   86 FORMAT(1X,'RANGE',    T15,8F8.2)
   88 FORMAT(1X,'NO. OBS.', T15,8I8)
C
```

```
  180 CONTINUE
      A=B
  200 CONTINUE
      RETURN
      END
C  ***  **  ***  **  ***  **  ***
C
      SUBROUTINE SUMMER (ISUM)
C
      INTEGER ISUM(10,8)
      WRITE(3,10)
   10 FORMAT(//1X,'NUMBER OF SUMMER SIGMA-T OBS IN EACH SQUARE',//)
      J=11
      DO 200 LOCN=1,10
      J=J-1
      WRITE(3,20) (ISUM(J,LOCW), LOCW=1,8)
   20 FORMAT(/1X,'NO. OBS.',T15,8I8)
  200 CONTINUE
      RETURN
      END
C  ***  **  ***  **  ***  **  ***
C
      SUBROUTINE STAT2 (SIGSUM,KNTS,ISIGMA)
C
C  COMPUTES ELEMENTARY STATISTICS ON ACCUMULATED DATA VALUES.
C  ACCUMULATES DATA FROM EACH DEPTH LAYER ABOVE LAYER 4.
C
      REAL SIGSUM(10,8,4),AVG1(10,8),VAR1(10,8),DEV1(10,8),RNG1(10,8)
     &SUM(10,8)
      INTEGER KNTS(10,8,4)
      INTEGER*2 ISIGMA
      DIMENSION ISIGMA(10,8,4,2000)
      INTEGER N(10,8)
C
      DO 200 L=1,4
      DO 180 K=1,8
      DO 160 J=1,10
C
C  COMPUTES INDIVIDUAL STATISTICS ON SURFACE AND BOTTOM LAYERS,
C  CUMULATIVE STATISTICS FROM SURFACE TO BOTTOM OF LAYER 3.
      IF(L .EQ. 2 .OR. L .EQ. 4) GO TO 50
      N(J,K) = KNTS(J,K,L)
      SUM(J,K)=SIGSUM(J,K,L)
      GO TO 60
   50 N(J,K) = N(J,K) + KNTS(J,K,L)
      SUM(J,K) = SUM(J,K) + SIGSUM(J,K,L)
   60 IF(N(J,K) .LT. 1) GO TO 120
      AVG1(J,K) = SUM(J,K)/FLOAT(N(J,K))
```

```
      GO TO 125
  120 AVG1(J,K) = 0.0
  125 CONTINUE
      SIGMIN = 100.0
      SIGMAX = 0.0
      SS     = 0.0
C
      DO 142 I1=1,L
      IEND   = KNTS(J,K,L)
      IF(IEND .EQ. 0)  IEND=1
      DO 140 M=1,IEND
      SIGMA = ISIGMA(J,K,L,M)/100.0
      SS = SS + (SIGMA - AVG1(J,K))**2
      SIGMAX = AMAX1(SIGMAX,SIGMA)
      SIGMIN = AMIN1(SIGMIN,SIGMA)
  140 CONTINUE
  142 CONTINUE
C
      IF(N(J,K) .LE. 1) GO TO 145
      VAR1(J,K) = SS/FLOAT(N(J,K)-1)
      GO TO 150
  145 VAR1(J,K) = 0.0
  150 DEV1(J,K) = SQRT(VAR1(J,K))
      RNG1(J,K) = SIGMAX-SIGMIN
C
  160 CONTINUE
  180 CONTINUE
C
      CALL RITER(AVG1,VAR1,DEV1,RNG1,N,L)
C
  200 CONTINUE
C
      RETURN
      END
C
C  ***  **  ***  **  ***  **  ***
C
      SUBROUTINE RITER(AVG1,VAR1,DEV1,RNG1,N,L)
C
C  WRITES SUMMARY OF CUMULATIVE STATS IN TABULAR FORM, BY DEPTH
C
      COMMON /TRE/ ISTIME,IETIME
      REAL AVG1(10,8),VAR1(10,8),DEV1(10,8),RNG1(10,8)
      INTEGER A,B,N(10,8)
C
      WRITE(3,5) ISTIME,IETIME
  5   FORMAT(//1X,'FOR MONTHS BEGINNING: ',I4,', AND ENDING: ',I4)
      A=0
```

```
      B = L*10
      IF(L .EQ. 4) A=30
      IF(L .EQ. 4) B=100
C     IF(L .EQ. 5) GO TO 15
      WRITE(3,10) A,B
   10 FORMAT(//1X,'STATISTICS ON SIGMATEE FOR DEPTHS BETWEEN ',
     &      I6,' AND',I6,' METRES')
C
      J=11
      DO 180 J1=1,10
      J=J-1
      WRITE(3,80)  (AVG1(J,K),   K=1,8)
      WRITE(3,82)  (VAR1(J,K),   K=1,8)
      WRITE(3,84)  (DEV1(J,K),   K=1,8)
      WRITE(3,86)  (RNG1(J,K), K=1,8)
      WRITE(3,88)  (N(J,K),      K=1,8)
   80 FORMAT(/1X,'MEAN',    T15,8F8.2)
   82 FORMAT(1X,'VARIANCE', T15,8F8.2)
   84 FORMAT(1X,'STD. DEV.',T15,8F8.2)
   86 FORMAT(1X,'RANGE',    T15,8F8.2)
   88 FORMAT(1X,'NO. OBS.', T15,8I8)
C
  180 CONTINUE
  200 CONTINUE
      RETURN
      END
C  [illegible]
C
C  NOTES ON PROGRAM:
C
C  PROGRAM READS LATITUDE, LONGITUDE, DEPTH, DATE, SIGMA-T, AND
C  RECORD TYPE.  RECORD IS REJECTED (NO FURTHER PROCESSING OCCURS)
C  IF:
C              * RECORD TYPE = 1,
C              * LAT OR LONG IS OUT OF DESIRED RANGE,
C              * DEPTH IS BELOW 100 METRES,
C              * VALUES FOR DEPTH OR SIGMA-T COULD NOT BE
C                TRANSLATED FROM CODED VALUE ON TAPE.
C
C  OTHERWISE, ISIGMA(J,K,L,M) IS CLASSIFIED BY GRID LOCATION NORTH (J),
C  GRID LOCATION WEST (K), DEPTH RANGE (L), AND NUMBER OF THIS
C  RECORD IN THIS CATEGORY (M).  ISIGMA IS ASSIGNED AS A HALF-WORD
C  INTEGER TO REDUCE THE TOTAL MEMORY REQUIREMENT.
C  NUMBER OF VALID VALUES OCCURING DURING THE SUMMER PERIOD IS
C  ACCUMULATED AS 'ISUM'.
C  'STAT2' CALCULATES CUMULATIVE STATISTICS FOR THE UPPER THREE LAYERS.
C  IF STAT2 IS CALLED, 'RITER' IS CALLED FROM WITHIN STAT2, AND SUBROUTINE
C  'OUTPUT' IS NOT CALLED.
```

```
/*
/*
TAPE RUN
PUNCH DLOG DATA G (NOH
O DISK G
L e o G (DAT
/*
/*
/*
```

The Department of the Interior Mission

As the Nation's principal conservation agency, the Department of the Interior has responsibility for most of our nationally owned public lands and natural resources. This includes fostering sound use of our land and water resources; protecting our fish, wildlife, and biological diversity; preserving the environmental and cultural values of our national parks and historical places; and providing for the enjoyment of life through outdoor recreation. The Department assesses our energy and mineral resources and works to ensure that their development is in the best interests of all our people by encouraging stewardship and citizen participation in their care. The Department also has a major responsibility for American Indian reservation communities and for people who live in island territories under U.S. administration.

The Minerals Management Service Mission

As a bureau of the Department of the Interior, the Minerals Management Service's (MMS) primary responsibilities are to manage the mineral resources located on the Nation's Outer Continental Shelf (OCS), collect revenue from the Federal OCS and onshore Federal and Indian lands, and distribute those revenues.

Moreover, in working to meet its responsibilities, the **Offshore Minerals Management Program** administers the OCS competitive leasing program and oversees the safe and environmentally sound exploration and production of our Nation's offshore natural gas, oil and other mineral resources. The MMS **Minerals Revenue Management** meets its responsibilities by ensuring the efficient, timely and accurate collection and disbursement of revenue from mineral leasing and production due to Indian tribes and allottees, States and the U.S. Treasury.

The MMS strives to fulfill its responsibilities through the general guiding principles of: (1) being responsive to the public's concerns and interests by maintaining a dialogue with all potentially affected parties and (2) carrying out its programs with an emphasis on working to enhance the quality of life for all Americans by lending MMS assistance and expertise to economic development and environmental protection.