NOAA Technical Memorandum ERL GLERL-46

A TWO-DIMENSIONAL LAKE CIRCULATION MODELING SYSTEM: PROGRAMS TO COMPUTE PARTICLE TRAJECTORIES AND THE MOTION OF DISSOLVED SUBSTANCES

John R. Bennett
Anne H. Clites
David J. Schwab

Great Lakes Environmental Research Laboratory
Ann Arbor, Michigan
May 1983

UNITED STATES DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
John V Byrne, Administrator

Environmental Research Laboratories
George H. Ludwig, Director

NOTICE

CONTENTS

FIGURE

A TWO-DIMENSIONAL LAKE CIRCULATION MODELING SYSTEM: PROGRAMS TO COMPUTE PARTICLE TRAJECTORIES AND THE MOTION OF DISSOLVED SUBSTANCES*

John R. Bennett, Anne Hutchinson Clites, and David J. Schwab

This report documents two computer programs that use currents from numerical lake circulation models to predict the motion of particles and dissolved substances. Movement of particles in the water is related to the vertically averaged current and to a percentage of the surface wind. Dissolved substances are assumed to be vertically mixed and thus to move only with the current. The user must provide bathymetric data, initial conditions, currents, winds, and a subroutine to generate output. The programs are very general and, though designed to be used with the hydrodynamic models and bathymetric grid generation programs documented in earlier GLERL technical memoranda, can be easily adapted to use input from other sources.

1. INTRODUCTION

One of the more practical applications of lake and ocean circulation models is the prediction of drifting object trajectories for search and rescue operations or oil and chemical spill cleanup. This type of forecasting involves four basic steps. First, the wind stress is determined from either predicted or observed wind and the air-water temperature difference. Second, the currents are calculated from the wind stress, the initial and boundary conditions, and the density field. Third, trajectories are computed from the currents and winds. Finally, when forecasting the movement of oil or chemicals, the effects of physical or chemical transformations must be estimated. A detailed discussion of these is given by Stolzenbach et al. (1977). At present there are operational methods that perform all four of these calculations (Pickett, 1981; Simons et al., 1975; Torgrimson, 1981). These methods, of necessity, make compromises between physical or numerical accuracy and ease of operation. Work continues to improve their accuracy while maintaining their broad applicability. The programs documented here are an attempt at making some of these refinements.

To test models of this type, the Great Lakes Environmental Research Laboratory (GLERL) has begun a field program using satellite-tracked current drifter buoys (Pickett et al., 1983). The drifter tracks will eventually be used to verify our model's predicted trajectories. Results of those tests will be published as they become available.

Our general lake circulation modeling system now consists of three parts: (1) a program to generate a bathymetric grid for any of the lakes and convert it to any size (Schwab and Sellers, 1980); (2) a set of programs to calculate

---

*GLERL Contribution No. 364.

wind stress fields and to predict currents in either a free surface or rigid-lid type of model (Schwab et al., 1981); and, with this report, (3) a set of programs to simulate the wind and current-driven motion of particles or of a dissolved substance in a lake. The programs documented in this memorandum were developed to use as input the bathymetric grid and current simulation developed earlier in the GLERL series; however, they can easily be adapted to use grids and currents generated elsewhere.

Within the main program, there are two diferent time steps in both programs. The large time step, DT, is supplied by the user. This increment determines how often the transport field will be updated and currents computed. Within this large time step, an internal time step, DTT, is calculated from the maximum current and wind speeds and knowledge of the stability criteria for the numerical methods. These smaller increments are used for computing the motion of particles or of a dissolved substance due to wind and currents.

The user is called upon to tailor these programs to his/her individual needs. Each program contains several user-supplied subroutines for this purpose. Both programs use a subroutine called UPDATE, which supplies the transport field at the midpoint of each large time step. Program PARTIC calls subroutine UPPART to introduce particles into the lake. Program ADVEC calls subroutine INIT to initialize the concentration field. Subroutine PHYSICS within ADVEC may be used to introduce other sources and sinks to the model. Each program also requires a user-supplied subroutine to generate output. In addition to these subroutines, the user of either program must supply the bathymetric grid, transport field, wind field, and control parameters (time step, duration of run, water level increment, and anemometer height).

2. PARTICLE TRAJECTORIES: Program PARTIC

2.1 Physical Assumptions

Several assumptions are basic to the physical processes of the model:

(1) The particles move with the vertically averaged current plus some user-supplied fraction of the wind.

(2) When particles are driven by currents alone (zero wind effect), they do not cross the shore boundary.

(3) When particles are driven onshore by wind, they stay there.

(4) Within the large time step (specified by the user), the current and wind stay constant. The user may choose a time interpolation of wind and current and the time step accordingly.

(5) The equations describing particle motion are:

$$\frac{dx}{dt} = u(x,y) + au_a(x,y) \tag{1}$$

$$\frac{dy}{dt} = v(x,y) + av_a(x,y) \tag{2}$$

where $\frac{dx}{dt}$ and $\frac{dy}{dt}$ are the particle velocity components, u and v are the x and y components of the current, a is the wind factor (the fraction of the surface wind with which the particle travels), and u_a and v_a are the x and y components of the wind.

2.2 Finite Difference Approximations

The variables are defined at the points shown in figure 1 for each grid box in the bathymetric grid. To ensure that particles can only be driven across the shore boundary by wind, never by current alone, the model deals with wind and current in separate steps. During the wind step, the following first-order differential equations govern particle motion:

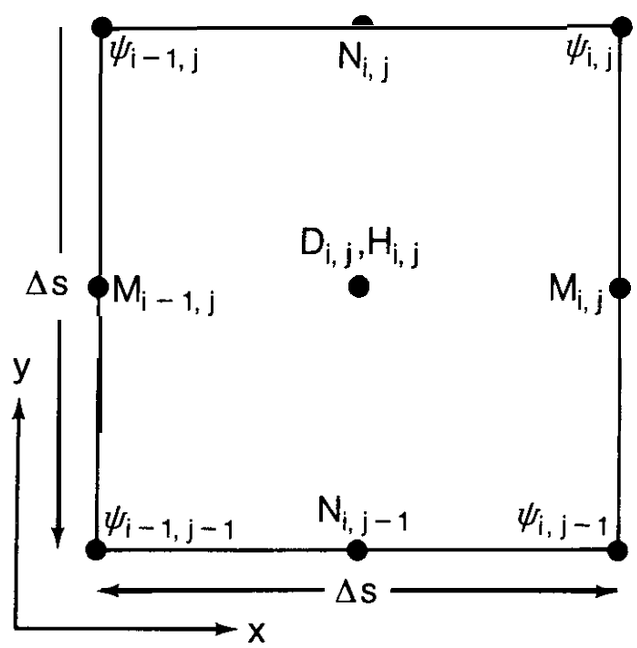

FIGURE 1.--*Definition of variables in* ***grid box*** *i, j* ***for lake*** *circulation* ***models.***

$$x^{n+1} - x^n = \Delta t \cdot a \cdot u_a(x^n, y^n) \tag{3}$$

$$y^{n+1} - y^n = \Delta t \cdot a \cdot v_a(x^n, y^n) \tag{4}$$

This straightforward method is adequate since the wind has a simple spatial structure. A spatial interpolation scheme allows for wind measurements at different heights, if necessary.

Predicting motion due to currents is more complex for several reasons. First, the simplest methods for interpolating the velocity field give rise to particle trajectories that either cross the shore boundary, or for which particles near the boundary eventually become trapped in the grid corners. Second, the first-order time derivatives have persistent errors that cause the particles to drift toward the boundary. We have developed a method for interpolating the velocity components that minimizes the first problem and a second-order time stepping procedure that solves the second problem.

Spatially, the currents are interpolated to the ϕ points in figure 1. In this process, care is taken to ensure that the boundary values are extrapolated from the interior. Referring to figure 1, every stream function point has north-south components of velocity on its right and left, and east-west components above and below it. Away from the shore boundary, interpolated values of the velocity components are formed by averaging these. However, if one of these components is zero (shore value), the nearest interior value of that component is used instead. This prevents artificial "dead" zones from developing in the grid corners. This step would not be necessary in a model that used a no-slip boundary condition. For the current interpolation step and the wind step, we choose a small time step, $\Delta t'$, so that the stability criteria are satisfied:

$$\Delta t' < \frac{\Delta S}{U_{max}} \tag{5}$$

There are $\frac{\Delta t}{\Delta t'}$ small time steps, each of which obeys the following set of equations:

$$\frac{x^{n+1} - x^n}{\Delta t'} = u(x^n, y^n) + 1/2 \frac{\partial u}{\partial x}(x^{n+1} - x^n) + 1/2 \frac{\partial u}{\Delta y}(y^{n+1} - y^n) \tag{6}$$

$$\frac{y^{n+1} - y^n}{\Delta t'} = v(x^n, y^n) + 1/2 \frac{\partial v}{\partial x}(x^{n+1} - x^n) + 1/2 \frac{\partial v}{\partial y}(y^{n+1} - y^n) \tag{7}$$

These formulas were derived from Taylor series expansion about the old particle position.

The values of u, v, and their derivatives are computed by bilinear interpolation from the four corner points of the grid square in which the particle begins the time step. Once this has been done, the pair of simultaneous linear equations (6) and (7) can be solved directly, yielding the new particle positions. The accuracy of this particle trajectory model is very dependent on grid size (Bennett et al., 1983).

3. MOTION OF DISSOLVED SUBSTANCES: Program ADVEC

3.1 Physical Assumptions

With the approximation that the water is incompressible and vertically well-mixed, the conservation equation for a dissolved substance or a property carried with the water is

$$\frac{\partial(C\cdot(D + H))}{\partial t} = -\frac{\partial}{\partial x}(U\cdot C) - \frac{\partial}{\partial y}(V\cdot C) + P(x,y,t) \tag{8}$$

In this equation, C(x,y,t) is the concentration of the substance in units of kilograms per cubic meter, D(x,y) is the equilibrium water depth, H(x,y,t) is the time-dependent water level fluctuation, and U and V are the x and y components of the volume transport, defined as:

$$U = \int_{-D}^{H} u\,dz, \qquad V = \int_{-D}^{H} v\,dz \tag{9}$$

P(x,y,t) is a function that describes sources, sinks, or other physical processes, such as decay or horizontal mixing. All initial conditions, volume transports, and sources and sinks must be provided by the user. The time-dependent fluctuation of the water level, however, is computed within the program:

$$\frac{\partial H}{\partial t} = -\frac{\partial U}{\partial x} - \frac{\partial V}{\partial y} \tag{10}$$

This ensures that the water level is always compatible with the transports.

3.2 Finite Difference Approximations

The program solves equation (8) in three steps according to the three terms that affect the total mass of C:

$$\frac{\partial C}{\partial t} = -\frac{1}{(D+H)}\frac{\partial}{\partial x}(UC) + \frac{\partial}{\partial y}(VC) \qquad \text{step 1}$$

$$-\frac{1}{(D+H)}\frac{\partial H}{\partial t} \qquad \text{step 2}$$

$$+ P \qquad \text{step 3}$$

In steps 1 and 2, the terms are evaluated in a conservative manner so that:

$$\frac{\partial}{\partial t}\left(\int_{\text{lake}} C\cdot(D+H)\Delta S^2\right) = \text{river fluxes}$$

If there are no river fluxes in the system, the total mass of C will not change.

To calculate the advective terms of step 1, the program alternates between the upwind method and the Lax-Wendroff method. Both of these are standard methods described in many textbooks (e.g., Roache, 1972; Richtmyer and Morton, 1967).

4. REFERENCES

Bennett, J. R., Campbell, J. E., and Clites, A. H. 1983. Accuracy of a finite difference method for computing Eulerian and Lagrangian currents. Submitted to *J. Comp.Phys.*

Pickett, R. L. 1981. Great Lakes spill model operating instructions, NOAA Tech. Memo. ERL GLERL-XX, National Technical Information Service, Sprinfield, Va. 22151. 9 pp.

Pickett, R. L., Campbell, J. E., Clites, A. H., and Partridge, R. M. 1983. Satellite-tracked current drifters in Lake Michigan. *J. Great Lakes Res.* 9(1):106-108.

Richtmyer, R. D., and Morton, K. W. 1967. *Difference methods for initial-value problems*, second edition, Interscience Publishers, J. Wiley and Sons, New York, N.Y., 419 pp.

Roache, P. J. 1972. *Computational fluid dynamics*, Hermosa Publishers, Albuquerque, N.M., 446 pp.

Schwab, D. J., and Sellers, D. L. 1980. Computerized bathymetry and shorelines of the Great Lakes, NOAA Data Rept. ERL GLERL-16, National Technical Information Service, Springfield, Va. 22151. 13 pp.

Schwab, D. J., Bennett, J. R., and Jessup, A. T. 1981. A two-dimensional lake circulation modeling system, NOAA Tech. Memo. ERL GLERL-38, National Technical Information Service, Springfield, Va. 22151. 79 pp.

Simons, T. J., Beal, G. S., Beal, K., El-Shaarawi, and Murty, T. S. 1975. Operational model for predicting the movement of oil in Canadian navigable waters, Rept. No. 37, Marine Sciences Directorate, Ottawa, Ont. 30 pp.

Stolzenbach, D. K., Madsen, O. S., Adams, E. E., Pollack, A. M., and Cooper, C. K. 1977. A review of basic techniques for predicting the behavior of surface oil slicks, M.I.T. Sea Grant Program Rept. No. 77-8, Massachusetts Institute of Technology, Cambridge, Mass. 322 pp.

Torgrimson, G. M. 1981. A comprehensive model for oil spill simulation. In: *Proceedings, of the 1981 Oil Spill Conference*, pp. 423-428. American Petroleum Institute, Environmental Protection Agency, U.S. Coast Guard, Atlanta, Ga.

Appendix A.--FORTRAN COMPUTER PROGRAMS: PARTIC, ADVEC, AND ASSOCIATED SUBROUTINES

```
      PROGRAM PARTIC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE7,TAPE8,TAPE9)
C*****************************PROGRAM PARTIC***************************
C PURPOSE :
C                THE PURPOSE OF  THIS PROGRAM  IS  TO SIMULATE THE
C                MOVEMENT OF TRACER PARTICLES   IN A GRID MODEL OF A LAKE.
C                THE LAKE  IS REPRESENTED AS AN ARRAY OF SQUARE  GRID
C                BOXES.  THE USER MUST SUPPLY A DESCRIPTION OF THE LAKE
C                BATHYMETRY, THE WIND FIELD, THE VOLUME TRANSPORTS BETWEEN
C                BOXES, CONTROL PARAMETERS (TIME STEP, DURATION OF RUN,
C                WATER LEVEL INCREMENT, AND HEIGHT OF ANEMOMETER), AND
C                TRACER PARTICLE INITIAL POSITIONS.
C
C  INPUT
C   LOGICAL UNIT 5
C     RECORD 1 - CONTROL PARAMETER RECORD :
C                                                          FORMAT   CARD COLUMNS
C                DT - TIME STEP IN HOURS                    G10. 2       1 - 10
C                TT - DURATION OF RUN   IN HOURS            G10.2       11 - 20
C                DADD - MEAN WATER LEVEL RELATIVE
C                       TO LOW WATER DATUM IN METERS        G10.2       21 - 30
C                Z - HEIGHT ABOVE WATER   IN METERS
C                    AT WHICH WIND  IS MEASURED              G10.2       31 - 40
C
C   LOGICAL UNIT 7
C     BATHYMETRIC DATA FILE :
C                THE FORMAT OF THE BATHYMETRIC DATA FILE   IS DESCRIBED
C                IN SCHWAB AND SELLERS (1980): "COMPUTERIZED BATHY-
C                METRY AND SHORELINES OF THE GREAT LAKES", NOAA DATA
C                REPORT ERL-GLERL-16, AS DESCRIBED IN SCHWAB, BENNETT,
C                AND JESSUP  (1981)  : "A TWO-DIMENSIONAL LAKE CIRCULATION
C                MODELING  SYSTEM",  NOAA TECHNICAL MEMORANDUM ERL-GLERL-38.
C                FIVE ADDITIONAL FIELDS ARE PRESENT ON BATHYMETRIC
C                DATA HEADER RECORD NUMBER 2.  THESE ARE:
C                                                     FORMAT  CARD  COLUMNS
C                MINIMUM DEPTH                          I5       45-49
C                BASE GRID ROTATION FROM E-W            F6.2     50-55
C                I DISPLACEMENT                         I5       56-60
C                J DISPLACEMENT                         I5       61-65
C                ROTATION  ANGLE  FROM  BASE            F6.2     66-71
C                 (ANGLES MEASURED  IN DEGREES COUNTERCLOCKWISE)
C
C                THE ADDITIONAL FIELDS ARE REQUIRED FOR CONVERSIONS
C                BETWEEN LATITUDE, LONGITUDE PAIRS AND GRID DISTANCES.
C                ONLY THE BATHYMETRIC PART OF THE FILE   IS USED, SHORE-
C                LINE INFORMATION NEED NOT BE INCLUDED.
C
C   LOGICAL UNIT 8
C     METEOROLOGICAL DATA FILE
C                                                     FORMAT  CARD  COLUMNS
C                TLAST - TIME FROM BEGINNING          G10.4       1 - 10
C                        OF RUN (H)
C                PLAT - LATITUDE IN DEGREES NORTH     G10. 4      11 - 20
C                PLON - LONGITUDE  IN DEGREES WEST    G10. 4      21 - 30
```

```
C                Z - HEIGHT OF INSTRUMENT (M)        G10.4       31 - 40
C                TA - TEMPERATURE OF AIR (C)         G10.4       41 - 50
C                TW - TEMPERATURE OF WATER (C)       G10.4       51 - 60
C                WS - WIND SPEED (M/S)               G10.4       61 - 70
C                WD - WIND DIRECTION (DEG)           G10.4       71 - 80
C
C         ALL DATA FOR THE SAME TIME ARE GROUPED TOGETHER, WITH A
C         MAXIMUM OF 25 STATIONS IN A GROUP.
C
C       * NOTE: END-OF-FILE IS INDICATED BY A RECORD WITH A
C               NEGATIVE TIME.
C
C  OUTPUT :
C    LOGICAL UNIT 6 :
C      CONTROL PARAMETERS, BATHYMETRY, AND A LIST OF THE
C      METEOROLOGICAL DATA RECORDS.
C
C  COMMON BLOCKS
C                /CPARM/ DT, TT, DADD, Z - CONTROL PARAMETERS
C                /GPARM/ RPARM(23), IPARM(54) - REAL AND INTEGER
C                   PARAMETERS DESCRIBING THE BATHYMETRIC GRID.
C                   SEE SUBROUTINE RGRID FOR DETAILS.
C
C  SUBROUTINES
C
C        RGRID - READS THE BATHYMETRIC DATA FILE
C        PGPARM - PRINTS GRID PARAMETERS
C        PRNT - FORMATS AND PRINTS OUTPUT ON GRID
C        UZL - CALCULATES DRAG COEFFICIENTS AND WIND PROFILE
C              PARAMETERS USED BY FUNCTION WIND
C        FUNCTION WIND - READS METEOROLOGICAL DATA AND CALCULATES
C                        X AND Y COMPONENTS OF WIND
C        FUNCTION XDIST - RETURNS X DISTANCE FROM GRID ORIGIN GIVEN
C                         LATITUDE AND LONGITUDE
C        FUNCTION YDIST - RETURNS Y DISTANCE FROM GRID ORIGIN GIVEN
C                         LATITUDE AND LONGITUDE
C        FUNCTION UZ - CALCULATES WIND SPEED AT A DIFFERENT
C                      HEIGHT (ZWIND) THAN THE OBSERVATIONAL HEIGHT (Z)
C                      BASED ON WIND PROFILE PARAMETERS
C
C                THE USER MUST SUPPLY THREE SUBROUTINES TO UPDATE
C                TRANSPORTS AND GENERATE OUTPUT.  THEY ARE:
C
C        UPPART(T,PART,WFACTR,CFACTR,NPMAX) - CALLED AT THE  BEGINNING
C             OF EACH TIMESTEP TO INITIALIZE PARTICLE POSITIONS.  T IS
C             IN SECONDS FROM BEGINNING OF RUN.  PART DEFINES THE PARTICLE
C             POSITIONS IN METERS RELATIVE TO THE GRID  ORIGIN.  WFACTR IS
C             THE FRACTION OF THE WIND SPEED WITH WHICH PARTICLES MOVE IN
C             PURELY WIND DRIVEN MOTION. CFACTR IS THE FRACTION FOR
C             CURRENT. NPMAX IS THE MAXIMUM NUMBER OF PARTICLES THAT MAY BE
C             INTRODUCED IN ONE TIMESTEP.
C
C        UPDATE(T,D,U,V, WORK, IDIM) - SUPPLIES TRANSPORT FIELD AT THE MIDDLE OF
C             EACH TIMESTEP.  T IS IN SECONDS FROM BEGINNING OF RUN.
```

```
C             D IS THE DEPTH ARRAY.  U AND V ARE THE X AND Y COMPONENTS OF
C             TRANSPORT.  U(I,J), THE X COMPONENT OF TRANSPORT,   IS DEFINED AT
C             THE CENTER OF THE RIGHT SIDE OF  GRID  BOX  I,J.  V(I,J), THE Y
C             COMPONENT OF TRANSPORT, IS DEFINED AT THE CENTER OF THE TOP OF
C             GRID BOX  I,J.  WORK(I,J) IS A TEMPORARY STORAGE ARRAY CONTAIN-
C             ING STREAM FUNCTION FIELDS.  IDIM  IS THE FIRST DIMENSION OF
C             D, U, AND V.
C
C           * NOTE: TRANSPORTS ARE ONLY REQUIRED AT THE MIDPOINT OF
C                   EACH TIME STEP,   I.E., T = DT*(I+1/2), WHERE I=1,2,3,...
C
C        POUTP(T,D,PART,IDIM) - GENERATES USER-REQUIRED OUTPUT.
C             POUTP IS CALLED BY PARTIC EVERY TIME STEP (DT, SPECIFIED
C             BY THE USER).  T IS THE TIME IN SECONDS FROM THE BEGINNING
C             OF THE RUN.  D IS THE DEPTH ARRAY.  PART DEFINES THE
C             PARTICLE POSITIONS (IN METERS RELATIVE TO THE   GRID ORIGIN).
C             IDIM IS THE FIRST DIMENSION OF D.
C
C  HISTORY
C              WRITTEN BY   J. R. BENNETT AND A. H. CLITES, 1982, GREAT
C              LAKES ENVIRONMENTAL RESEARCH LABORATORY, ANN ARBOR, MI.
C
      REAL D(50,50),WORK(50,50),U(50,50),V(50,50),
     1     PART(2,100),WFACTR(100),CFACTR(100)
      COMMON/CPARM/DT, TT, DADD, Z
      COMMON/GPARM/RPARM(23), IPARM(54)
      DATA IDIM, JDIM/50,50/
      DATA NPMAX/100/
C
C READ BATHYMETRIC GRID INFORMATION
C
      CALL RGRID(7,D,IDIM,   JDIM)
C
C READ CONTROL PARAMETERS
C
      READ(5,100) DT, TT, DADD, Z
      WRITE(6,110) DT, TT, DADD, Z
      NSTEPS = TT/DT
      DT = DT*3600.
      IM = IPARM(1)
      JM = IPARM(2)
      DS = RPARM(3)
      DMIN = RPARM(5) + DADD
      IMM1 =  IM-1
      JMM1 = JM-1
C
C CLEAR ARRAYS AND ADD WATER LEVEL INCREMENT
C
      DO 20 I = 1,IM
         DO 20 J = 1,JM
            U(I,J) = 0.
            V(I,J) = 0.
            WORK(I,J) = 0.
            IF ( D(I,J) .LT. RPARM(5)) GO TO 20
```

```
            D(I, J) = D(I,J) + DADD
   20 CONTINUE
C
C IF WATER LEVEL INCREMENT RESULTS IN A NEGATIVE DMIN, STOP
C
      IF (DMIN . LE. 0.0) GO TO 90
C
C PRINT BATHYMETRIC DATA FILE HEADER INFORMATION
C
      CALL PGPARM (6)
C
C PRINT DEPTH FIELD
C
      CALL PRNT(6, D, IDIM, IM, JM, 0.)
      WRITE(6,130)
C
C MAIN ITERATION LOOP
C
      T = 0.
      DO 30 NS=1, NSTEPS
C
C GET PARTICLES FOR THIS TIMESTEP
C
      CALL UPPART(T,PART,WFACTR,CFACTR,NPART,NPMAX)
      IF (NPART .GT. NPMAX) GO TO 95
C
C MOVE TO MIDDLE OF TIMESTEP
C
      T = T + DT/2.
C
C UPDATE TRANSPORTS
C
      CALL UPDATE(T, D, U, V, WORK, IDIM)
C
C CONVERT TRANSPORTS TO HALF CURRENTS   BY   DIVIDING BY TWICE DEPTH
C
      DO 40  I=1,IMM1
         DO 40 J=1,JMM1
            IF  (D(I,J)  .GE.DMIN.AND.  D(I+1,J)  .GE.DMIN)
     1         U(I,J) = U(I,J)/(D(I,J) + D(I+1,J))
            IF  (D(I,J)  .GE.DMIN.AND.  D(I,J+1)  .GE.DMIN)
     1         V(I,J) = V(I,J)/(D(I,J) + D(I,J+1))
   40 CONTINUE
C
C INTERPOLATE CURRENTS TO STREAM FUNCTION POINTS, TAKING CARE
C TO USE INTERIOR CURRENTS AT THE SHORE BOUNDARY
C
      UMAX = 0.
      DO 51 I=1,IMM1
         IM1 = I-1
         IF(I.EQ. 1) IM1  =  1
         DO 51 J=1,JMM1
            UUP =  U(I,J+1)
            IF (D(I,J+1) .LT.DMIN)UUP  =  U(I+1,J+1)
```

```
            IF (D(I+1,J+1) .LT.  DMIN) UUP = U(IM1,J+1)
            UDN= U(I,J)
            IF (D(I,J).LT. DMIN)UDN = U(I+1,J)
            IF (D(I+1,J).LT. DMIN)UDN = U(IM1,J)
            IF (D(I,J).LT. DMIN.AND. D(I+1, J).LT. DMIN)UDN=UUP
            IF (D(I,J+1) .LT.  DMIN.AND.  D(I+1,J+1).LT.  DMIN)UUP=UDN
            WORK(I,J)  = UUP + UDN
C
C  CALCULATE MAXIMUM CURRENT SPEED
C
            UMAX = AMAX1(UMAX,ABS(WORK(I,J)))
   51 CONTINUE
      DO 50 I=1, IMM1
         DO 50 J=1, JMM1
         U(I,J) = WORK(I,J)
   50 CONTINUE

      DO 53 J=1, JMM1
         JM1 = J-1
         IF (J.EQ. 1) JM1 = 1
         DO 53 I=1, IMM1
            VR = V(I+1,J)
            IF (D(I+1, J) .LT. DMIN)VR = V(I+1,J+1)
            IF (D(I+1, J+1) .LT. DMIN)VR = V(I+1,JM1)
            VL = V(I,J)
            IF (D(I,J).LT. DMIN)VL = V(I,J+1)
            IF (D(I, J+1) .LT. DMIN)VL = V(I,JM1)
            IF  (D(I,J).LT.  DMIN.AND.  D(I,J+1).LT.DMIN)VL=VR
            IF (D(I+1, J) .LT.  DMIN  .AND.  D(I+1,J+1).LT.DMIN)VR=VL
            WORK(I,J) = VL + VR
            UMAX = AMAX1(UMAX,ABS(WORK(I,J)))
   53 CONTINUE
      DO 52 I=1, IMM1
         DO 52 J=1, JMM1
         V(I, J) = WORK(I, J)
   52 CONTINUE
C
C  CALCULATE MAXIMUM  WIND SPEED
C
      XMAX = IM*DS
      YMAX =  IM*DS
      WMAX = 0.0
      WMAX = AMAX1(WMAX,ABS(WIND(T,0.,0.,1,Z)))
      WMAX =  AMAX1(WMAX,ABS(WIND(T,0.,0.,2,Z)))
      WMAX = AMAX1(WMAX,ABS(WIND(T,0.,YMAX,1,Z)))
      WMAX = AMAX1(WMAX, ABS(WIND(T,0.,YMAX,2,Z)))
      WMAX = AMAX1(WMAX,ABS(WIND(T,XMAX,0.,1,Z)))
      WMAX =  AMAX1(WMAX,ABS(WIND(T,XMAX,0.,2,Z)))
      WMAX = AMAX1(WMAX,ABS(WIND(T,XMAX,YMAX,1,Z)))
      WMAX =  AMAX1(WMAX,ABS(WIND(T,XMAX,YMAX,2,Z)))
C
C  COMPUTE MAXIMUM CURRENT AND WIND FACTORS
C
      SPDMAX = 0.0
```

```
      CFMAX = 0.0
      WFMAX = 0.0
      DO 55 N=1, NPART
         CFMAX = AMAX1(CFMAX, CFACTR(N))
         WFMAX =  AMAX1(WFMAX,WFACTR(N))
   55 CONTINUE
C
C  USE MAXIMUM CURRENT AND WIND SPEEDS TO CALCULATE INTERNAL TIME STEP
C
      SPDMAX = AMAX1(CFMAX*UMAX,WFMAX*WMAX)
      NDTT = (IFIX((SPDMAX*DT)/DS) +1)*8
      DTT = DT/NDTT
      DTDS2 = (DTT*0.5) /DS
C
C  MOVE PARTICLES WITH CURRENTS
C
      DO 60 NDT=1, NDTT
         DO 70 N=1,NPART
            I = IFIX(PART(1,N)/DS) + 1
            J = IFIX(PART(2,N)/DS) + 1
            IF (D(I,J) .LT. DMIN) GO TO 70
            DX = (PART(1,N)/DS) - FLOAT(I-1)
            DY = (PART(2,N)/DS) - FLOAT(J-1)
            UOLD = CFACTR(N)*(U(I-1,J-1)*(1.0-DX)*(1.0-DY)+U(I,J-1)*DX*
     1         (1.0-DY)+U(I-1,J)*DY*(1.0-DX)+U(I,J)*DX*DY)
            VOLD = CFACTR(N)*(V(I-1,J-1)*(1.0-DX)*(1.0-DY)+V(I,J-1)*DX*
     1         (1.0-DY)+V(I-1,J)*DY*(1.0-DX)+V(I,J)*DX*DY)
            AA = CFACTR(N)*DTDS2*((U(I,J)-U(I-1,J))*DY +
     1          (U(I,J-1)-U(I-1,J-1))*(1.0-DY))
            BB = CFACTR(N)*DTDS2*((U(I,J)-U(I,J-1))*DX +
     1          (U(I-1,J)-U(I-1,J-1))*(1.0-DX))
            CC = CFACTR(N)*DTDS2*((V(I,J)-V(I-1,J))*DY +
     1          (V(I,J-1)-V(I-1,J-1))*(1.0-DY))
            DD = CFACTR(N)*DTDS2*((V(I,J)-V(I,J-1))*DX +
     1          (V(I-1,J)-V(I-1,J-1))*(1.0-DX))
            DENOM= 1.0 - AA - DD + AA*DD - BB*CC
            UPART = (UOLD* (1.0-DD) + VOLD*BB) /DENOM
            VPART = (UOLD*CC + VOLD*(1.0-AA))/DENOM
            PART(1,N) = PART(1,N)+DTT*UPART
            PART(2,N) = PART(2,N)+DTT*VPART
C
C   CHECK TO SEE WHETHER THE CURRENT HAS CAUSED THE PARTICLE
C   TO CROSS THE SHORE BOUNDARY.  IF IT HAS, MOVE IT BACK,
C   TAKING THE SHORTEST ROUTE.
C
            I = IFIX(PART(1,N)/DS)+1
            J = IFIX(PART(2,N)/DS)+1
            IF  (D(I,J)  .GE. DMIN) GO TO 70
            DX = (PART(1,N)/DS) - FLOAT(I-1)
            DY = (PART(2,N)/DS) - FLOAT(J-1)
            DXMIN  =AMIN1(DX,1.0-DX)
            DYMIN = AMIN1(DY,1.0-DY)
            IF (DYMIN .LT. DXMIN) GO TO 75
            IF (DX.LE. 0.5) PART(1,N)=PART(1,N)-1.01*DS*DX
```

```
            IF (DX .GT. 0.5) PART(1,N)=PART(1,N)+1.01*DS*(1.0-DX)
            GO TO 70
   75 CONTINUE
            IF (DY .LE. 0.5) PART(2,N)=PART(2,N)-1.01*DS*DY
            IF (DY .GT. 0.5) PART(2,N)=PART(2,N)+1.01*DS*(1.0-DY)
   70 CONTINUE
C
C  MOVE PARTICLES WITH WIND.   IF A PARTICLE CROSSES THE SHORE
C  BOUNDARY DRIVEN BY THE WIND,   IT STAYS THERE FOREVER.
C
      DO 80 N=1,NPART
         I = IFIX(PART(1,N)/DS) +   1
         J = IFIX(PART(2,N)/DS) +   1
         IF(D(I,J).LT.  DMIN)  GO  TO  80
         UPART = WFACTR(N)*WIND(T,PART(1,N),PART(2,N),1,Z)
         VPART = WFACTR(N)*WIND(T,PART(1,N),PART(2,N),2,Z)
         PART(1,N) = PART(1,N)+UPART*DTT
         PART(2,N) = PART(2,N)+VPART*DTT
   80 CONTINUE
   60 CONTINUE
C
C  UPDATE TIME
C
      T = NS*DT
C
C  CALL OUTPUT ROUTINE
C
      CALL POUTP(T, D, PART, NPART,   IDIM)
C
C  END MAIN LOOP
C
   30 CONTINUE
      STOP
   90 WRITE (6,120) DADD
      STOP
   95 WRITE (6,140)
      STOP
  100 FORMAT(4F10.2)
  110 FORMAT("1",//," DT = ",F6.2,"     TT = ", F6.2, "     DADD = ",F6.2,
     1    "     Z = ",F5.2)
  120 FORMAT("1",20X,"THE WATER LEVEL INCREMENT",F8.2,"RESULTS  IN  A",
     1        "NEGATIVE MINIMUM DEPTH - PROGRAM TERMINATED")
  130 FORMAT(50X, "DEPTH RELATIVE TO MEAN WATER LEVEL")
  140 FORMAT("1",20X, "THE NUMBER OF PARTICLES EXCEEDS THE MAXIMUM")
      END
```

```
      PROGRAM ADVEC (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT,
     1TAPE7,TAPE9)
C*************************PROGRAM ADVEC*********************************
C PURPOSE :
C                THE PURPOSE OF THIS PROGRAM IS TO COMPUTE THE
C                CONCENTRATION OF A CONSERVATIVE SUBSTANCE SUBJECT
C                ONLY TO ADVECTION IN A GRID MODEL OF A LAKE.
C                THE LAKE IS REPRESENTED AS AN ARRAY OF SQUARE GRID
C                BOXES.  THE USER MUST SUPPLY A DESCRIPTION OF THE
C                LAKE BATHYMETRY, THE VOLUME TRANSPORTS BETWEEN THE BOXES,
C                THE WIND FIELD, CONTROL PARAMETERS (TIME STEP,
C                DURATION OF RUN, WATER LEVEL INCREMENT) AND INITIAL
C                CONDITIONS.
C
C  INPUT :
C   LOGICAL UNIT 5 :
C     CONTROL PARAMETER RECORD :
C                                                    FORMAT  CARD  COLUMNS
C                DT - TIME STEP IN HOURS             G10.2        1 - 10
C                TT - DURATION OF RUN IN HOURS       G10.2       11 - 20
C                DADD - MEAN WATER LEVEL RELATIVE    G10.2       21 - 30
C                       TO LOW WATER DATUM IN METERS
C
C   LOGICAL UNIT 7 :
C     BATHYMETRIC DATA FILE :
C                THE FORMAT OF THE BATHYMETRIC DATA FILE   IS DESCRIBED
C                IN SCHWAB AND SELLERS (1980): "COMPUTERIZED BATHY-
C                METRY AND SHORELINES OF THE GREAT LAKES", NOAA DATA
C                REPORT ERL-GLERL-16, AS DESCRIBED IN SCHWAB, BENNETT,
C                AND JESSUP  (1981)  : "A TWO-DIMENSIONAL LAKE CIRCULATION
C                MODELING SYSTEM", NOAA TECHNICAL MEMORANDUM   ERL-GLERL-38.
C                FIVE ADDITIONAL FIELDS ARE PRESENT ON BATHYMETRIC
C                DATA HEADER RECORD NUMBER 2.  THESE ARE:
C                                                    FORMAT  CARD  COLUMNS
C                MINIMUM DEPTH                         I5       45-49
C                BASE GRID ROTATION FROM E-W           F6.2     50-55
C                I DISPLACEMENT                        I5       56-60
C                J DISPLACEMENT                        I5       61-65
C                ROTATION ANGLE FROM BASE              F6.2     66-71
C                 (ANGLES MEASURED IN DEGREES COUNTERCLOCKWISE)
C
C                THE ADDITIONAL FIELDS ARE REQUIRED FOR CONVERSIONS
C                BETWEEN LATITUDE,LONGITUDE PAIRS AND GRID DISTANCES.
C                ONLY THE BATHYMETRIC PART OF THE FILE   IS USED, SHORE-
C                LINE INFORMATION NEED NOT BE INCLUDED.
C
C OUTPUT :
C   LOGICAL UNIT 6 :
C     CONTROL PARAMETERS AND BATHYMETRY.
C
C COMMON BLOCKS :
C                /CPARM/ DT, TT, DADD - CONTROL PARAMETERS
C                /GPARM/ RPARM(23), IPARM(54) - REAL AND INTEGER
C                   PARAMETERS DESCRIBING THE BATHYMETRIC  GRID.
```

```
C                    SEE SUBROUTINE RGRID FOR DETAILS.
C
C  SUBROUTINES
C
C        RGRID - READS THE BATHYMETRIC DATA FILE
C        PGPARM - PRINTS  GRID PARAMETERS
C        PRNT - FORMATS AND PRINTS OUTPUT ON GRID
C        ADVECUP - CALCULATES ADVECTION ACCORDING TO THE UPWIND METHOD
C        ADVECLW - CALCULATES ADVECTION ACCORDING TO THE LAX-WENDROFF METHOD
C
C          * NOTE: THESE ADVECTION SUBROUTINES ARE ALTERNATED TO
C                  COMPUTE  IN-LAKE CALCULATIONS.  FOR RIVER INFLOWS
C                  AND OUTFLOWS, THE UPWIND METHOD  IS USED.
C
C             THE USER MUST SUPPLY SEVERAL SUBROUTINES TO UPDATE
C             TRANSPORTS, INITIALIZE CONDITIONS, ADD OTHER MASS
C             BALANCE TERMS, AND GENERATE OUTPUT.  THEY ARE
C
C        UPDATE(T,D,AM,AN,   IDIM) - SUPPLIES TRANSPORT FIELD AT TIME T
C             (IN SECONDS FROM  BEGINNING OF RUN).  D IS THE DEPTH ARRAY.
C             IDIM IS THE FIRST DIMENSION OF D,AM, AND AN.  AM AND AN ARE THE X AND
C             Y COMPONENTS OF TRANSPORT.  AM(I,J), THE X COMPONENT OF
C             TRANSPORT,  IS DEFINED AT THE CENTER OF THE RIGHT SIDE OF
C             GRID BOX  I,J.  AN(I, J), THE Y COMPONENT OF TRANSPORT, IS
C             DEFINED AT THE CENTER OF THE TOP OF  GRID  BOX  I,J.
C
C          * NOTE: TRANSPORTS ARE ONLY REQUIRED AT THE MIDPOINT OF
C                  EACH TIME STEP,  I.E., T = DT*(I+1/2), WHERE I=1,2,3 ,...
C
C        INIT(C,H,D,IDIM) - INITIALIZES THE CONCENTRATION FIELD AT THE
C             BEGINNING OF THE RUN.  D  IS THE DEPTH ARRAY.  C  IS THE
C             CONCENTRATION OF A DISSOLVED SUBSTANCE.  H  IS THE FREE
C             SURFACE FLUCTUATION FIELD  IN METERS.  IDIM  IS THE FIRST
C             DIMENSION OF C, H, AND D.
C
C        PHYSICS(C, H, D, U, V, T, IDIM) - PERMITS THE USER TO
C             INTRODUCE OTHER MASS BALANCE TERMS (SOURCES OR SINKS,
C             RESUSPENSION, SETTLING, ETC. ).  C  IS THE CONCENTRATION
C             OF A DISSOLVED SUBSTANCE.  H  IS THE FREE SURFACE
C             FLUCTUATION FIELD  IN METERS.  D  IS THE DEPTH ARRAY  IN
C             METERS.  U AND V ARE THE X AND Y COMPONENTS OF CURRENT
C             IN METERS PER SECOND.  T  IS TIME  IN SECONDS FROM THE
C             BEGINNING OF RUN.  IDIM  IS THE FIRST DIMENSION OF C,H,D,U AND V.
C
C          * NOTE: SUBROUTINE PHYSICS IS CALLED AT THE  MIDPOINT  OF
C                  EACH TIME STEP, I.E., T = DT*(I+1/2), WHERE I=1,2,3...
C
C        OUTP(C,H, D, T,  IDIM) - GENERATES USER-REQUIRED OUTPUT.
C             OUTP  IS CALLED BY ADVEC EVERY TIME STEP AT THE
C             MIDPOINT OF THE TIME INCREMENT, DT.  T  IS THE
C             TIME IN SECONDS FROM THE BEGINNING OF THE RUN.
C             D  IS THE DEPTH ARRAY  IN METERS.  C IS THE
C             CONCENTRATION OF A DISSOLVED SUBSTANCE.  H IS THE
C             FREE SURFACE FLUCTUATION IN METERS.  IDIM IS
```

```
C               THE   FIRST DIMENSION OF C, H, AND D.
C
   HISTORY:
               WRITTEN BY J. R. BENNETT AND A. H. CLITES, 1982,
               GREAT LAKES ENVIRONMENTAL RESEARCH LABORATORY,
               ANN ARBOR, MICHIGAN.

      REAL D(50,50), H(50,50), AM(50,50), AN(50,50), C(50,50),
     1FLUXX(50,50), FLUXY(50,50),U(50,50), V(50,50), HOLD(50,50)
      COMMON/CPARM/DT, TT, DADD
      COMMON/GPARM/RPARM(23), IPARM(54)
      COMMON/BOTTOM/SED(50,50)
      DATA IDIM, JDIM /50,50/
C
C READ BATHYMETRIC  GRID  INFORMATION
C
      CALL RGRID (7, D, IDIM, JDIM)
      READ (5,100) DT, TT, DADD
C
C  READ CONTROL PARAMETERS
C
      WRITE (6,110) DT, TT, DADD
      NSTEPS = TT / DT
      DT = DT * 3600.
      IM = IPARM(1)
      JM = IPARM(2)
      DS = RPARM(3)
      DMIN = RPARM(5) + DADD
      IMM1 = IM - 1
      JMM1 = JM - 1
C
C CLEAR ARRAYS AND ADD WATER LEVEL INCREMENT
C
      DO 10 I = 1, IM
        DO 10 J = 1, JM
          H(I,J) = 0.
          HOLD(I,J) = 0.
          AM(I,J) = 0.
          AN(I,J) = 0.
          C(I,J) = 0.
          FLUXX(I,J) = 0.
          FLUXY(I,J) = 0.
          U(I,J)=0.
          V(I,J)=0.
          ISTEP=0.
          IF (D(I,J) .LT. RPARM(5)) GO TO 10
          D(I,J) = D(I,J) + DADD
   10 CONTINUE
C
C  IF WATER LEVEL INCREMENT RESULTS  IN  A  NEGATIVE  DMIN, STOP
C
      IF (DMIN .LE. 0.0) GO TO 95
C
C PRINT BATHYMETRIC DATA FILE HEADER INFORMATION
```

```
C
      CALL PGPARM (6)
C
C   PRINT DEPTH FIELD
C
      CALL PRNT (6, D, IDIM, IM, JM, 0.)
      WRITE (6,140)
C
C   INITIALIZE CONCENTRATION FIELD
C
      CALL INIT (C, H, D, IDIM)
C
C  MAIN ITERATION LOOP
C
      T=0.
      DO 20 N = 1, NSTEPS
         T = T + DT/2.
C
C UPDATE TRANSPORTS
C
         CALL UPDATE  (T,  D, AM, AN,  IDIM  )
         UMAX = 0.
C
C CONVERT TRANSPORTS TO CURRENTS BY DIVIDING BY DEPTH
C
      DO 30 I=1, IMM1
         DO 30 J=1, JMM1
            IF(D(I,J).GE.DMIN   .OR.   D(I+1,J).GE.DMIN) U(I,J) =
     1         (2.0*AM(I,J))/(D(I,J) + D(I+1,J)+H(I,J)+H(I+1,J))
            IF(D(I,J).GE.DMIN.OR.  D(I,J+1)  .GE.  DMIN)  V(I,J)  =
     1         (2.0*AN(I,J))/(D(I,J) + D(I,J+1)+H(I,J)+H(I,J+1))
C
C CALCULATE MAXIMUM CURRENT SPEED
C
      UMAX=AMAX1(UMAX,ABS(U(I,J)))
      UMAX=AMAX1(UMAX,ABS(V(I,J)))

   30 CONTINUE
C
C USE MAXIMUM CURRENT SPEED TO CALCULATE INTERNAL TIME STEP
C
      NDTT=IFIX(1.5*(((UMAX*DT)/DS)+0.7))
      DTT=DT/NDTT
      DO 40 NT=1,NDTT
         ISTEP = ISTEP + 1
C
C CALL ADVECTION SUBROUTINE (ALTERNATES UPWIND METHOD WITH
C                                LAX-WENDROFF METHOD)
C
      IF(MOD(ISTEP,2). EQ.0) CALL ADVECUP(C,D,AM,AN,H,DTT,
     1    FLUXX,FLUXY,U,V,IDIM)
      IF(MOD(ISTEP,2).NE.0)  CALL ADVECLW(C,D,AM,AN,H,DTT,
     1    FLUXX,FLUXY,U,V, IDIM)
C
```

```
C CALCULATE WATER LEVEL   DISPLACEMENT
C
      DO 50 I=2, IMM1
         DO 50 J=2,  JMM1
            IF (D(I,J) .LT. DMIN) GO TO 50
            HOLD(I,J) = H(I,J)
            H(I, J) =   H(I,J) + (DTT/DS) * (AM(I-1,J) -   AM(I,J)
     1                + AN(I, J-1) - AN(I, J))
            IF  ((D(I,J)  +  H(I,J)).LE.0.0) GO TO 90
            C(I,J) = C(I,J) * ((HOLD(I,J) + D(I,J))
     1                / (H(I,J) + D(I,J)))
   50 CONTINUE
   40 CONTINUE
C
C CALL PHYSICS ROUTINE
C
      CALL PHYSICS (C, H, D, U, V, T, IDIM)
      T = N * DT
C
C CALL OUTPUT ROUTINE
C
      CALL OUTP  (C,  H,  D,T,  IDIM)
C
C END   MAIN LOOP
C
   20 CONTINUE
      STOP
   90 WRITE (6, 120)
      STOP
   95 WRITE (6,  130) DADD
      STOP
  100 FORMAT (F10.7,2F10.2)
  110 FORMAT ("1",//," DT =   ",F10.7,"  TT = ",F6.2,"   DADD = ",F6.2)
  120 FORMAT ("1", 20X, "NEGATIVE WATER DEPTH")
  130 FORMAT ("1", 20X, "THE WATER LEVEL INCREMENT",F8.2, "RESULTS IN A"
     1         "NEGATIVE MINIMUM DEPTH - PROGRAM TERMINATED")
  140 FORMAT (50X, "DEPTH RELATIVE TO MEAN WATER LEVEL")
      END
```

```
      SUBROUTINE PGPARM(LUN)
C
C PURPOSE
C                    TO PRINT THE GRID DESCRIPTION PARAMETERS (RPARM
C                    AND IPARM IN COMMON BLOCK /GPARM/)
C
C ARGUMENTS
C                    LUN - LOGICAL UNIT NUMBER ON WHICH TO PRINT
C
C COMMON BLOCK:
C                    /GPARM/ RPARM(23),IPARM(54)
C
      COMMON /GPARM/ RPARM (23), IPARM (54)
      WRITE  (LUN,10) (IPARM(I),I=5,54)
      WRITE  (LUN,20) (IPARM(I),I=1,4)
      WRITE  (LUN,30) (RPARM(I),I=1,7)
      WRITE  (LUN,40) (RPARM(I),I=8,23)
   10 FORMAT ("0BATHYMETRIC DATA FILE HEADER FOR", 50A1)
   20 FORMAT ("0IDIMENSION : IPARM(1) =   ", I5, 6X, "JDIMENSION : ",
     1       "IPARM(2) = ", 8X, I5/"0", 14X,
     2       "DISPLACEMENT OF ORIGIN IN ", "NUMBER OF GRID SQUARES"/
     3       "0IDISPLACEMENT : IPARM(3) =", I5, 6X,
     4       "JDISPLACEMENT : IPARM(4) =", 6X, I5)
   30 FORMAT ("0BASE LATITUDE : RPARM(1) = ", F12.7/"0BASE LONGITUDE ",
     1       ": RPARM(2) =", F12.7/"0GRID SIZE (M) : RPARM(3) = ", F6.0,
     2       5X, "MAX DEPTH (M) : RPARM(4) = ", 6X, F5.0/
     3       "0MIN DEPTH (M) : ", "RPARM(5) = ", F5.0, 5X,
     4       "BASE ROTATION : RPARM(6) = ", 8X, F5.2/
     5       "0ANGLE ROTATED : RPARM(7) = ", F5.0)
   40 FORMAT ("0", 11X, "GEOGRAPHIC-TO-MAP COORDINATE CONVERSION ",
     1       "COEFFICENTS FOR X"/"0RPARM(8) = ", 6X, E15.6, 5X,
     2       "RPARM(9) = ", 12X, E15.6/"0RPARM(10) = ", 5X, E15.6, 5X,
     3       "RPARM(11) = ", 11X, E15.6/"0", 11X,
     4       "GEOGRAPHIC-TO-MAP COORDINATE CONVERSION COEFFICIENTS",
     5       " FOR Y"/"0RPARM(12) = ", 5X, E15.6, 5X, "RPARM(13) = ",
     6       11X, E15.6/"0RPARM(14) = ", 5X, E15.6, 5X, "RPARM(15) = ",
     7       11X, E15.6/"0", 7X,
     8       "MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS",
     9       " FOR LONGITUDE"/"0RPARM(16) = ", 5X, E15.6, 5X,
     *       "RPARM(17) = ", 11X, E15.6/"0RPARM(18) = ", 5X, E15.6, 5X,
     1       "RPARM(19)=", 11X, E15.6/"0", 7X,
     2       "MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS",
     3       " FOR LATITUDE"/"0RPARM(20) = ", 5X, E15.6, 5X,
     4       "RPARM(21) = ",   11X, E15.6/"0RPARM(22) = ", 5X, E15.6, 5X,
     5       "RPARM(23) = ", 11X, E15.6)
      RETURN
      END
```

```
      FUNCTION WIND(T, XS, YS, K, ZWIND)
C
C PURPOSE :
C                TO RETURN WIND COMPONENT (METERS/SECOND) FOR
C                LOCATION (XS,YS) AT TIME T.  XS AND YS ARE IN METERS
C                RELATIVE TO THE GRID ORIGIN.  THE PARAMETER K
C                INDICATES WHICH COMPONENT OF WIND IS RETURNED.
C                K = 1 FOR THE X COMPONENT AND K = 2 FOR THE Y COMPONENT.
C                BOTH COMPONENTS ARE LINEAR FUNCTIONS OF X AND Y.
C                ZWIND IS THE HEIGHT ABOVE THE WATER IN METERS AT
C                WHICH WIND IS DETERMINED.
C
C ALGORITHM:     THIS FUNCTION READS METEOROLOGICAL DATA FROM A FILE,
C                CONVERTS IT TO WIND STRESS, AND INTERPOLATES THE WIND
C                STRESS IN TIME AND SPACE.  ALL DATA FOR THE SAME TIME
C                ARE GROUPED TOGETHER, WITH  A  MAXIMUM OF 25 STATIONS IN
C                A GROUP.  THE TWO COMPONENTS OF THE WIND ARE ASSUMED
C                TO BE LINEAR FUNCTIONS OF X AND Y AND THE BEST-FIT LINEAR
C                SURFACE FOR THE GIVEN DATA POINTS IS CALCULATED. EACH
C                TIME WIND IS CALLED, THE TIME PARAMETER IS CHECKED
C                AGAINST THE TIME OF THE LAST SET OF METEOROLOGICAL DATA
C                READ.  IF IT IS GREATER, ANOTHER GROUP IS READ.  IF IT
C                IS LESS, THE APPROPRIATE WIND COMPONENT IS CALCULATED
C                USING LINEAR INTERPOLATION BETWEEN THE LAST TWO LINEAR
C                SURFACES COMPUTED FOR THAT COMPONENT.  IF THE END-OF-FILE
C                IS ENCOUNTERED, WIND IS CALCULATED WITH THE LAST LINEAR
C                SURFACE COEFFICIENTS.
C
C                MANY OF THE VARIABLES USED IN THIS FUNCTION ARE ASSUM-
C                ED TO HAVE RETAINED THEIR VALUES FROM THE PREVIOUS
C                CALL.  IF YOUR FORTRAN SYSTEM DOES NOT DO THIS AUTO-
C                MATICALLY, PROVISION MUST BE MADE TO RETAIN VALUES
C                FOR THE VARIABLES:ISTA, AOLD1, BOLD1, COLD1, AOLD2,
C                BOLD2, COLD2, ANEW1, BNEW1, CNEW1, ANEW2, BNEW2, CNEW2,
C                TOLD, TNEW, TLAST, PLAT, PLON, WS, WD.
C
C  ARGUMENTS:
C                T - TIME IN SECONDS FROM BEGINNING OF RUN AT   WHICH
C                    STRESS IS TO BE CALCULATED
C                XS - X DISTANCE IN METERS FROM GRID ORIGIN AT WHICH
C                     WIND IS TO BE CALCULATED
C                YS - Y DISTANCE IN METERS FROM GRID ORIGIN AT WHICH
C                     WIND  IS TO BE CALCULATED
C                K - SPECIFIES WHETHER X COMPONENT OR Y COMPONENT
C                    OF STRESS IS RETURNED.  K = 1 FOR THE X COMPONENT
C                    AND K = 2 FOR THE Y COMPONENT
C                ZWIND - HEIGHT ABOVE WATER IN METERS AT WHICH WIND IS REQUIRED
C
C INPUT
C LOGICAL UNIT 8:
C      METEOROLOGICAL DATA FILE
C      CARD COLUMN          FORMAT      PARAMETER
C          1-10              G10.4      TLAST - TIME (HRS) FROM BEGINNING
C                                                 OF RUN
```

```
C         11-20            G10.4     PLAT - LATITUDE IN DEGREES NORTH
C         21-30            G10.4     PLON - LONGITUDE IN DEGREES WEST
C         31-40            G10.4     Z - HEIGHT OF INSTRUMENTS
C         41-50            G10.4     TA - TEMPERATURE OF AIR
C         51-60            G10.4     TW - TEMPERATURE OF WATER
C         61-70            G10.4     WS - WIND SPEED (M/S)
C         71-80            G10.4     WD - WIND DIRECTION (DEG)
C
C         ALL DATA FOR THE SAME TIME ARE GROUPED TOGETHER, WITH A
C         MAXIMUM OF 25 STATIONS IN A GROUP.
C
C    * NOTE: END-OF-FILE IS INDICATED BY A RECORD WITH A
C            NEGATIVE TIME
C
C OUTPUT
C LOGICAL UNIT 6
C      PRESENTATION OF METEOROLOGICAL DATA STATION BY STATION
C
C COMMON BLOCKS
C               /GPARM/RPARM(23),IPARM(54)
C
C SUBPROGRAMS :
C      FUNCTION XDIST - RETURNS X DISTANCE FROM GRID ORIGIN
C                          GIVEN LATITUDE AND LONGITUDE
C      FUNCTION YDIST - RETURNS Y DISTANCE FROM GRID ORIGIN
C                          GIVEN LATITUDE AND LONGITUDE
C      SUBROUTINE UZL - RETURNS DRAG COEFFICIENT AND WIND PROFILE
C                          PARAMETERS
C      FUNCTION UZ - RETURNS WIND SPEED AT HEIGHT ZWIND BASED ON
C                          WIND PROFILE PARAMETERS
C
C HISTORY :
C            WRITTEN BY J.R. BENNETT, 1982, GLERL, ANN ARBOR, MI BY
C            MODIFYING FUNCTION TAU.  SEE SCHWAB, BENNETT, AND JESSUP (1981).
C
      DIMENSION X(25), U(2), Y(25), AWIN(25,2)
      LOGICAL XEQ, YEQ
      COMMON /GPARM/ RPARM(23), IPARM(54)
      DATA RHOAW, ISTA /1.25E-3, 0/
      DATA LUN /8/
      DATA AOLD1, BOLD1, COLD1, AOLD2, BOLD2, COLD2, TOLD /7*0. /
C
C IF THIS IS THE FIRST TIME THROUGH, READ A RECORD
C
      IF (ISTA .EQ. 0) GO TO 30
C
C CHECK IF NECESSARY TO READ ANOTHER RECORD
C
      IF (T .LT. TNEW .OR. TNEW . LT. 0.) GO TO 90
   10 AOLD1 = ANEW1
      BOLD1 = BNEW1
      COLD1 = CNEW1
      AOLD2 = ANEW2
      BOLD2 = BNEW2
```

```
      COLD2 = CNEW2
      TOLD = TNEW
   20 TNEW = TLAST
      IF (TNEW .LT. 0) GO TO 90
C
C FIND AWIN, THE WIND FOR THE CURRENT STATION AT HEIGHT ZWIND ABOVE WATER
C
      X(ISTA) = XDIST(PLAT,PLON)
      Y (ISTA) = YDIST(PLAT, PLON)
      TD = TA - TW
      CALL UZL(WS, Z, TD, Z, CD, CH, Z0, FL)
      WSZ = UZ(ZWIND, WS, CD, Z0, FL)
      CDD = CD *     1.E3
      AWIN(ISTA,1) = WSZ * COS((270. - WD - RPARM(6) - RPARM(7))*
     1               ATAN(1.)/45.)
      AWIN(ISTA,2) = WSZ * SIN((270. - WD - RPARM(6) - RPARM(7))*
     1               ATAN(1.)/45.)
      TTLAST = TLAST / 3600.
      IF (ISTA .LE. 1) WRITE (6,160)
      XKM = X(ISTA) / 1000.
      YKM = Y (ISTA) / 1000.
      WRITE (6, 150) TTLAST, PLAT, PLON, Z, TA, TW, WS, WD, CDD,
     1 XKM, YKM
   30 READ (LUN, 130) TLAST, PLAT, PLON, Z, TA, TW, WS, WD
      TLAST = TLAST * 3600.
      IF (T .LT. TLAST .AND. ISTA .EQ. 0) GO TO 120
      ISTA = ISTA + 1
C
C  IF FIRST TIME THROUGH, FIND AWIN
C
      IF (ISTA .EQ. 1) GO TO 20
C
C CHECK IF LAST RECORD AT TIME TLAST HAS BEEN READ
C
      IF (TLAST .EQ. TNEW) GO TO 20
C
C NOW FIND THE BEST-FIT LINEAR SURFACE
C
      SX = 0.
      SY = 0.
      SXY = 0.
      SX2 = 0.
      SY2 = 0.
      SXWIN1 = 0.
      SYWIN1 = 0.
      SAWIN1 = 0.
      SXWIN2 = 0.
      SYWIN2 = 0.
      SAWIN2 = 0.
      N = ISTA - 1
      AN = FLOAT(N)
      XEQ = . TRUE.
      YEQ = .TRUE.
```

```
      DO 40 IN = 1, N
        SX = SX + X(IN)
        SY = SY + Y(IN)
        SXY = SXY + X(IN) * Y(IN)
        SX2 = SX2 + X(IN) ** 2
        SY2 = SY2 + Y(IN) * * 2
        SXWIN1 =SXWIN1 + X(IN) * AWIN(IN,1)
        SYWIN1 = SYWIN1 + Y(IN) * AWIN(IN,1)
        SXWIN2 = SXWIN2 + X(IN) * AWIN(IN,2)
        SYWIN2 = SYWIN2 + Y(IN) * AWIN(IN,2)
        SAWIN1 = SAWIN1 + AWIN(IN,1)
        SAWIN2 = SAWIN2 + AWIN(IN,2)
        IF  (X(IN).NE.  X(1)) XEQ = .FALSE.
        IF (Y(IN) .NE. Y(1)) YEQ = .FALSE.
   40 CONTINUE
C
C CALCULATE COEFFICIENTS ANEW, BNEW, CNEW, WHERE
C                        WIND = ANEW * X + BNEW * Y + CNEW
C                        FOR EACH COMPONENT.
C
      ANEW1 = 0.
      ANEW2 = 0.
      BNEW1 = 0.
      BNEW2 = 0.
      CNEW1 = AWIN(ISTA - 1,1)
      CNEW2 = AWIN(ISTA - 1,2)
C
C SOLVE THE SYSTEM OF EQUATIONS
C
C  1)     : SX2  SXY  SX :   : ANEW :   : SXWIN :
C  2)     : SXY  SY2  SY : * : BNEW : = : SYWIN :
C  3)     :  SX   SY AN  :   : CNEW :   : SAWIN :
C
C   BY   COMBINING EQUATIONS 1 AND 2 AND EQUATIONS 2 AND 3, WE
C ELIMINATE B AND OBTAIN 2 EQUATIONS  IN  A  AND  C.  THEN
C EXPRESSIONS FOR A AND C ARE OF THE FORM:
C              A = EXPRESSION1 / ACHK
C              C = EXPRESSION2 / (-ACHK)
C WHERE ACHK IS THE DETERMINANT OF THE 2 EQUATION SYSTEM.  IF ALL Y
C VALUES ARE EQUAL, WE SET A = 0 AND USE A DIFFERENT   EXPRESSION   TO
C  FIND  C.
C
C BY COMBINING EQUATIONS   1 AND 2 AND EQUATIONS 2 AND 3, WE
C ELIMINATE A AND OBTAIN 2 EQUATIONS  IN  B  AND  C.  THEN
C EXPRESSIONS FOR B AND C ARE OF THE FORM:
C               B = EXPRESSION3 / BCHK
C               C = EXPRESSION4 / (-BCHK)
C WHERE BCHK   IS THE DETERMINANT OF THE 2 EQUATION SYSTEM.  IF ALL X
C VALUES ARE EQUAL, WE SET B = 0 AND USE A DIFFERENT EXPRESSION TO
C  FIND C.
C
C   IF BOTH A AND B EQUAL 0, THEN C HAS THE DEFAULT VALUE ASSIGNED
C  ABOVE. (THIS WILL OCCUR IF ONLY ONE STATION   IS USED.)
C
```

```
      IF (YEQ) GO TO 50
      ACHK = (SY2*SX2 - SXY**2) * (SY**2 - AN*SY2) - (SY*SXY - SX*SY2) *
     1 (SX*SY2 - SY*SXY)
      ANEW1 = ((SXWIN1*SY2 - SYWIN1*SXY)*(SY**2 - AN*SY2) - (SYWIN1*SY -
     1 SAWIN1*SY2)*(SX*SY2 - SY*SXY)) / ACHK
      ANEW2 = ((SXWIN2*SY2 - SYWIN2*SXY)*(SY**2 - AN*SY2) - (SYWIN2*SY -
     1 SAWIN2*SY2)*(SX*SY2 - SY*SXY)) / ACHK
   50 IF (XEQ) GO TO 60
      BCHK = (SXY**2 - SX2*SY2) * (SY*SX - AN*SXY) - (SY2*SX - SY*SXY) *
     1 (SX*SXY - SY*SX2)
      IF (BCHK .EQ. 0. ) GO TO 60
      BNEW1 = ((SXWIN1*SXY - SYWIN1*SX2)*(SY*SX - AN*SXY) - (SYWIN1*SX -
     1 SAWIN1*SXY)*(SX*SXY - SY*SX2)) / BCHK
      BNEW2 = ((SXWIN2*SXY - SYWIN2*SX2)*(SY*SX - AN*SXY) - (SYWIN2*SX -
     1 SAWIN2*SXY)*(SX*SXY - SY*SX2)) / BCHK
   60  IF  (XEQ.AND.YEQ) GO TO 80
      IF  (YEQ)  GO TO 70
      CNEW1 = ((SXWIN1*SY2 - SYWIN1*SXY)*(SY*SXY - SX*SY2) - (SYWIN1*SY
     1- SAWIN1*SY2)*(SY2*SX2 - SXY**2)) / (-ACHK)
      CNEW2 = ((SXWIN2*SY2 - SYWIN2*SXY)*(SY*SXY - SX*SY2) - (SYWIN2*SY
     1- SAWIN2*SY2)*(SY2*SX2 - SXY**2)) / (-ACHK)
      GO TO 80
   70 CNEW1  =  ((SXWIN1*SXY  - SYWIN1*SX2)*(SY*SX - AN*SXY) - (SYWIN1*SX -
     1 SAWIN1*SXY)*(SX*SXY - SY*SX2)) / (-BCHK)
      CNEW2 = ((SXWIN2*SXY - SYWIN2*SX2)*(SY*SX - AN*SXY) - (SYWIN2*SX -
     1 SAWIN2*SXY)*(SX*SXY - SY*SX2)) /(-BCHK)
   80 CONTINUE
      ISTA = 1
      IF (T .GT. TNEW) GO TO   10
      WRITE (6,170)
C
C  CALCULATE WIND COMPONENTS AT LOCATION (XS,YS) AT NEW AND OLD TIMES
C
   90  IF  (K .EQ. 2) GO TO 100
      WINNEW = ANEW1 * XS + BNEW1 * YS + CNEW1
      WINOLD = AOLD1 * XS + BOLD1 * YS + COLD1
      GO TO  110
  100 WINNEW = ANEW2 * XS + BNEW2 * YS + CNEW2
      WINOLD = AOLD2 * XS + BOLD2 * YS + COLD2
C
C INTERPOLATE IN TIME
C
  110 WIND = WINOLD
      IF (TNEW .NE. TOLD) WIND = WIND + ((T-TOLD)/(TNEW-TOLD)) *
     1     (WINNEW-WINOLD)
      GO TO 180
  120 WRITE (6,140) TLAST, T
      STOP
  130 FORMAT  (8G10.4)
  140 FORMAT (" TIME OF FIRST METEOROLOGICAL DATA", G10.4,
     1        " SECONDS IS", " GREATER THAN T = ", G10.4,
     2        " - PROGRAM TERMINATED")
  150 FORMAT (" ", F5. 1, " * ", F10.7, " * ", F10.7, " * ", F4. 1, " * ",
     1        F5.2, " * ",F5.2," * ",F6.  2,  "  *  ", F4.0, " * ", F5.2,
```

```
   2        " * ",F5.0," * ", F5.0)
160 FORMAT (" ", 95("*")/
   1        "  TIME  *  LATITUDE  *  LONGITUDE  *  Z    * T-AIR * ",
   2        "T-H2O * W-SPD  *  W-DIR  *  CDE3  *  X  *    Y"/" ", 95("*"))
170 FORMAT (" ", 95("*"))
180 RETURN
    END
```

```
      FUNCTION UZ(Z,UM,CD,Z0,FL)
C
C  PURPOSE:
C
C        TO DETERMINE WIND SPEED AT HEIGHT Z GIVEN WIND SPEED (UM)
C        AND DRAG COEFFICIENT (CD) FROM ANOTHER HEIGHT AND THE
C        ROUGHNESS LENGTH (Z0) AND STABILITY LENGTH (FL) OF THE
C        PROFILE. USUALLY FUNCTION UZ IS USED AFTER SUBROUTINE UZL
C        IN ORDER TO FIND WIND SPEED AT A DIFFERENT HEIGHT THAN
C        THE OBSERVATION HEIGHT.
C
C  ALGORITHM:
C
C        THE WIND PROFILE IS ASSUMED TO CONFORM TO THE BUSINGER-
C        DYER FORMULATION USED IN SUBROUTINE UZL.
C
C  ARGUMENTS:
C
C        Z - HEIGHT AT WHICH WIND SPEED IS REQUIRED (METERS)
C        UM - WIND SPEED AT OBSERVATION HEIGHT (METERS PER SECOND)
C        CD - DRAG COEFFICIENT CORRESPONDING TO OBSERVATION HEIGHT
C        Z0 - ROUGHNESS LENGTH (METERS)
C        FL - STABILITY LENGTH (METERS)
C
C  HISTORY:
C
C        WRITTEN BY D.J SCHWAB, 1983, GLERL, ANN ARBOR, MI.
C        SEE SUBROUTINE UZL IN SCHWAB, BENNETT, AND JESSUP (1981)
C
      IF(FL.GT.0. ) GO TO 10
C
C UNSTABLE PROFILE
C
      X1=15./FL
      ARG1=(1.-X1*Z)**0.25
      ARG2=(1.-X1*Z0)**0.25
      B=ALOG((ARG1-1.)*(ARG2+1.)/((ARG1+1.)*(ARG2-1.)))+
     1   2.*(ATAN(ARG1)-ATAN(ARG2))
      GO TO 30
C
C STABLE SECTION
C
   10 IF(FL.LE. Z) GO TO 20
C
C MILDLY STABLE PROFILE
C
      B=ALOG(Z/Z0)+4.7*(Z-Z0)/FL
      GO TO 30
C
C STRONGLY STABLE PROFILE
C
   20 CONTINUE
      ARG1=FL/Z0
      X1=ALOG(ARG1)
```

```
      X2=ALOG(Z/FL)
      ARG1=1.-1./ARG1
      B=X1+4.7*ARG1+5.7*X2
C
C CALCULATE USTAR AND UZ
C
   30 CONTINUE
      USTAR=UM*SQRT(CD)
      UZ=USTAR*B/0.35
      RETURN
      END
```

```
      SUBROUTINE ADVECUP(C,D,AM,AN,H,DTT,FLUXX,FLUXY,U,V,IDIM)
C
C  PURPOSE:
C                    TO CALCULATE THE X AND Y COMPONENTS
C                    OF FLUX USING THE UPWIND METHOD
C
C  ARGUMENTS:
C                    C - CONCENTRATION OF SUBSTANCE
C                    D - DEPTH ARRAY (METERS)
C                    AM - COMPONENT OF TRANSPORT IN X-DIRECTION
C                    AN - COMPONENT OF TRANSPORT  IN Y-DIRECTION
C                    H - FREE SURFACE FLUCTUATION FIELD (METERS)
C                    DTT - INTERNAL TIME STEP (SECONDS)
C                    FLUXX, FLUXY - COMPONENTS OF FLUX
C                    U,V - COMPONENTS OF CURRENT (METERS/SECOND)
C                    IDIM - FIRST DIMENSION OF ALL PARAMETERS EXCEPT DTT
C
C  COMMON  BLOCK
C                    /GPARM/RPARM(23),IPARM(54)
C                    /CPARM/DT, TT, DADD
C
      COMMON /GPARM/ RPARM(23), IPARM(54)
      COMMON /CPARM/ DT, TT, DADD
      DIMENSION C(IDIM,1),D(IDIM,1),AM(IDIM,1),AN(IDIM,1),H(IDIM,1),
     1      FLUXX(IDIM,1),FLUXY(IDIM,1),U(IDIM,1),V(IDIM,1)
      DS=RPARM(3)
      DMIN=RPARM(5) + DADD
      IM=IPARM(1)
      JM=IPARM(2)
      IMM1=IM-1
      JMM1=JM-1
C
C CALCULATE X-COMPONENT OF FLUX
C
      DO 10 I=1,IMM1
         DO 10 J=2, JMM1
            IF(D(I,J) .LT. DMIN .AND. D(I+1,J) .LT. DMIN) GO TO 10
            IF (AM(I,J) .GE. 0.0)FLUXX(I,J)=  AM(I,J)  *  C(I,J)
            IF (AM(I,J) .LT. 0.0)FLUXX(I,J)=  AM(I,J)  *  C(I+1,J)
   10 CONTINUE
C
C CALCULATE Y-COMPONENT OF FLUX
C
      DO 20  J=1,JMM1
          DO 20 I=2, IMM1
            IF(D(I,J) .LT. DMIN .AND. D(I,J+1) .LT. DMIN) GO TO 20
            IF (AN(I, J) .GE. 0.0) FLUXY(I,J) = AN(I,J) * C(I,J)
            IF (AN(I,J) .LT. 0.0) FLUXY(I,J) = AN(I,J) • C(I,J+1)
   20 CONTINUE
C
C   CALCULATE NEW CONCENTRATION FIELD
C
      DO 30  I=2,IMM1
         DO 30 J=2,JMM1
```

```
      IF  (D(I,J).LT.  DMIN)  GO TO 30
      DMASS = -FLUXX(I,J) + FLUXX(I-1,J) -FLUXY(I,J)+FLUXY(I,J-1)
      C(I,J) = C(I,J) +DMASS *DTT / ((D(I,J) + H(I,J)) *DS)
30 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE ADVECLW(C,D,AM,AN,H,DTT,FLUXX,FLUXY,U,V, IDIM)
C
C  PURPOSE:
C                    TO CALCULATE THE X AND Y COMPONENTS
C                    OF FLUX USING THE LAX-WENDROFF METHOD
C
C  ARGUMENTS:
C                    C - CONCENTRATION OF SUBSTANCE
C                    D - DEPTH ARRAY (METERS)
C                    AM - COMPONENT OF TRANSPORT IN X-DIRECTION
C                    AN - COMPONENT OF TRANSPORT   IN Y-DIRECTION
C                    H - FREE SURFACE FLUCTUATION FIELD (METERS)
C                    DTT - INTERNAL TIME STEP (SECONDS)
C                    FLUXX,FLUXY - COMPONENTS OF FLUX
C                    U,V - COMPONENTS OF CURRENT (METERS/SECOND)
C                    IDIM - FIRST DIMENSION OF ALL PARAMETERS EXCEPT DTT
C
C  COMMON BLOCK
C                    /GPARM/RPARM(23),IPARM(54)
C                    /CPARM/DT, TT, DADD
C
      COMMON /GPARM/ RPARM(23), IPARM(54)
      COMMON /CPARM/ DT, TT, DADD
      DIMENSION C(IDIM,1),D(IDIM,1),AM(IDIM, 1),AN(IDIM,1),H(IDIM,1),
     1      FLUXX(IDIM,1),FLUXY(IDIM,1),U(IDIM,1),V(IDIM,1)
      DS=RPARM(3)
      DMIN=RPARM(5) + DADD
      IM=IPARM(1)
      JM=IPARM(2)
      IMM1=IM-1
      JMM1=JM-1
C
C CALCULATE X-COMPONENT OF FLUX
C
      DO 10 I=1, IMM1
         DO 10 J=2,JMM1
            IF(D(I,J)  .LT.  DMIN  .AND.  D(I+1,J).LT.DMIN) GO TO 10
            WTX=(DT*U(I,J))/DS
            IF (D(I,J) .LT. DMIN .OR. D(I,J+1) .LT. DMIN)
     1         WTX = SIGN(1.,U(I,J))
            FLUXX(I,J)=0.5*AM(I,J)*(C(I,J)*(1.0+WTX) +
     1                 C(I+1,J)*(1.0-WTX))
   10 CONTINUE
C
C  CALCULATE Y-COMPONENT OF FLUX
C
      DO 20  J=1,JMM1
          DO 20 I=2, IMM1
            IF(D(I,J) .LT. DMIN .AND. D(I,J+1) .LT. DMIN) GO TO 20
            WTY=(DT*V(I,J))/DS
            IF (D(I,J) .LT. DMIN .OR. D(I,J+1) .LT. DMIN)
     1         WTY = SIGN(1.,V(I,J))
            FLUXY(I,J)=0.5*AN(I,J)*(C(I,J)*(1.0+WTY) +
     1                 C(I, J+1)*(1.0-WTY))
```

```
   20 CONTINUE
C
C CALCUATE NEW CONCENTRATION FIELD
C
      DO 30 I=2,IMM1
         DO 30 J=2,JMM1
            IF (D(I,J) .LT. DMIN) GO TO 30
            DMASS = -FLUXX(I,J) + FLUXX(I-1,J) -FLUXY(I,J)+FLUXY(I,J-1)
            C(I,J)= C(I,J) + DMASS * DTT / ((D(I,J) + H(I,J)) * DS)
   30 CONTINUE
      RETURN
      END
```

Appendix B.--SAMPLE RUNS OF PROGRAMS PARTIC AND ADVEC

```
      SUBROUTINE UPPART(T,PART,WFACTR,CFACTR,NPART,NPMAX)
C
C PURPOSE:
C               TO  INITIALIZE THE PARTICLES AT THE BEGINNING
C               OF A TIMESTEP.
C
C ARGUMENTS:
C               T - DURATION OF RUN (SECONDS)
C               PART - ARRAY DEFINING PARTICLE LOCATIONS
C               WFACTR - WIND FACTOR (PERCENT)
C               CFACTR - CURRENT FACTOR (PERCENT)
C               NPART - NUMBER OF TRACER PARTICLES
C               NPMAX - MAXIMUM NUMBER OF PARTICLES
C
      DIMENSION PART(2,100), WFACTR(100), CFACTR(100)
      NPART = 4
      IF (T .GT. 2000) GO TO 20
C
C   READ PARTICLE POSITIONS, WIND AND CURRENT FACTORS
C
      DO 10 N=1,NPART
         READ(5, 100)PLAT,PLON,PWIND,PCURR
         PART(1,N) = XDIST(PLAT,PLON)
         PART(2,N) = YDIST(PLAT,PLON)
         WFACTR(N) = PWIND
         CFACTR(N) = PCURR
   10 CONTINUE

   20 RETURN
  100 FORMAT (2F12.5, 2F6.3)
      END
```

Program PARTIC
Sample user-supplied subroutine

```
      SUBROUTINE UPDATE (T, D, U, V, WORK, IDIM)
C
C PURPOSE :
C                    TO UPDATE THE TRANSPORTS BY READING
C                    THE STREAM FUNCTION FIELDS AT INTERVALS
C                    SPECIFIED BY THE USER.
C ARGUMENTS :
C                    T - DURATION OF RUN (SECONDS)
C                    D - DEPTH ARRAY (METERS)
C                    U - COMPONENT OF TRANSPORT   IN X-DIRECTION
C                    V - COMPONENT OF TRANSPORT IN Y-DIRECTION
C                    WORK - STORAGE ARRAY CONTAINING STREAM FUNCTION FIELDS
C                    IDIM - FIRST DIMENSION OF D,U, AND V   IN DIMENSION STATEMENT
C                           OF CALLING PROGRAM
C COMMON BLOCK :
C                    /GPARM/ RPARM(23),  IPARM(54)
C                    /CPARM/ DT, TT, DADD, Z
      COMMON/GPARM/RPARM(23), IPARM(54)
      COMMON/CPARM/DT, TT, DADD, Z
      DIMENSION D(IDIM, 1), U(IDIM, 1), V(IDIM, 1), WORK(IDIM, 1)
      DS = RPARM(3)
      DMIN = RPARM(5) + DADD
      IM = IPARM(1)
      JM = IPARM(2)
      IMM1 = IM - 1
      JMM1 = JM - 1
      READ(9)((WORK(I,J),I=1,IM),J=1,JM)
      DO 10 I=1, IM
         DO 10 J=1,JM
            U(I,J) = 0.
            V(I,J) = 0.
   10 CONTINUE
      DO20 I=1, IM
         DO 20 J=2, JM
            U(I, J)=(WORK(I,J-1) - WORK(I,J))/DS
   20 CONTINUE
      DO 30 I=2, IM
         DO 30 J=1,JM
            V(I, J)=(WORK(I, J) - WORK(I-1,J))/DS
   30 CONTINUE
      RETURN
      END
```

Program PARTIC
Sample user-supplied subroutine

```
      SUBROUTINE POUTP (T, D, PART, NPART,   IDIM)
C
C PURPOSE :
C                  TO PRINT THE OUTPUT
C
C ARGUMENTS :
C                  T - DURATION OF RUN (SECONDS)
C                  D - DEPTH ARRAY (METERS)
C                  PART - ARRAY DEFINING PARTICLE LOCATIONS
C                  NPART - NUMBER OF TRACER PARTICLES
C                  IDIM - FIRST DIMENSION OF D IN DIMENSION STATEMENT
C                         OF CALLING PROGRAM
C
C COMMON BLOCK:
C                  /GPARM/ RPARM(23), IPARM(54)
C                  /CPARM/ DT, TT, DADD, Z
C
      DIMENSION D(IDIM,1),PART(2,1)
      COMMON/GPARM/RPARM(23),IPARM(54)
      COMMON/CPARM/DT, TT, DADD, Z
      DMIN = RPARM(5) + DADD
      DS = RPARM(3)
      RCENT = 50000 + DS
      HRS = T/3600.
      WRITE (6,100) HRS

      DO 10 N=1,NPART
         X = PART(1,N) - RCENT
         Y = PART(2,N) - RCENT
         WRITE (6,110) N, X,Y
   10 CONTINUE

      RETURN
  100 FORMAT (1X,F12.0)
  110 FORMAT (1X,I6,2F12.1)
      END
```

Program PARTIC
Sample user-supplied subroutine

```
  1.00       3.00       0.00      10.00   }   Input for program PARTIC on LUN 5
41.57235    81.00000   .020 1.000         ⎫
41.78617    81.00000   .020 1.000         ⎬   Input for subroutine UPPART (program PARTIC) on LUN 5
42.00000    81.57330   .020 1.000         ⎪
42.00000    81.00000   .020 1.000         ⎭
```

Other input for program PARTIC include:

(1) bathymetric grid file on LUN 7
(2) meteorological data file on LUN 8
(3) streamfunction field file on LUN 9

For documentation of these files, refer to:

Schwab, D.J., J.R. Bennett, and A.T. Jessup (1981): A two-dimensional lake circulation modeling system. NOAA Technical Memorandum ERL-GLERL-38.

```
DT =   1.00    TT =   3.00    DADD =   0.00    Z = 10.00
BATHYMETRIC DATA FILE HEADER FOR      CIRCULAR PARABOLOID LAKE
IDIMENSION : IPARM(1) =      22      JDIMENSION : IPARM(2) =          22
             DISPLACEMENT OF ORIGIN IN NUMBER OF GRID SQUARES
IDISPLACEMENT  :  IPARM(3)  =  0      JDISPLACEMENT : IPARM(4) =        0
BASE LATITUDE : RPARM(1) =   41.5048257
BASE LONGITUDE : RPARM(2) = 81.6638263
GRID SIZE (M) : RPARM(3) = 5000.     MAX DEPTH (M) : RPARM(4) =       100.
MIN DEPTH (M) : RPARM(5) =    5.     BASE ROTATION : RPARM(6) =        0.00
ANGLE ROTATED : RPARM(7) =    0.
          GEOGRAPHIC-TO-MAP COORDINATE CONVERSION COEFFICENTS FOR X
RPARM(8) =          .833330E+02     RPARM(9) =              0.
RPARM(10) =         0.              RPARM(11) =             0.
          GEOGRAPHIC-TO-MAP COORDINATE CONVERSION COEFFICIENTS FOR Y
RPARM(12) =         0.              RPARM(13) =              .111111E+03
RPARM(14) =         0.              RPARM(15) =             0.
      MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS FOR LONGITUDE
RPARM(16) =         .120000E-01     RPARM(17) =             0.
RPARM(18) =         0.              RPARM(19) =             0.
      MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS FOR LATITUDE
RPARM(20) =         0.              RPARM(21) =              .900000E-02
RPARM(22) =         0.              RPARM(23) =             0.
```

Output from program PARTIC

Output from subroutine PGPARM

Page 1 of pr inted output from program PARTIC

```
VALUES MULTIPLIED BY 10** 0
*****************************************************************************************
********************************  8  12  14  14  12   8********************************
************************ 12  20  25  29  31  31  29  25  20  12************************
**************** 6  18  27  35  41  44  46  46  44  41  35  27  18  6****************
************ 6  20  31  41  48  54  58  60  60  58  54  48  41  31  20  6************
************ 18  31  43  52  60  65  69  71  71  69  65  60  52  43  31  18************
******** 12  27  41  52  62  69  75  79  81  81  79  75  69  62  52  41  27  12********
******** 20  35  48  60  69  77  82  86  88  88  86  82  77  69  60  48  35  20********
**** 8  25  41  54  65  75  82  88  92  94  94  92  88  82  75  65  54  41  25  8****
**** 12  29  44  58  69  79  86  92  96  98  98  96  92  86  79  69  58  44  29  12****
•  *** 14  31  46  60  71  81  88  94  98 100 100  98  94  88  81  71  646  31  14*****
**** 14  31  46  60  71  81  88  94  98 100 100  98  94  88  81  71  60  46  31  14****
**** 12  29  44  58  69  79  86  92  96  98  98  96  92  86  79  69  58  44  29  12****
•  ⊠⊠⊠  8  25  41  54  65  75  82  88  92  94  94  92  88  82  75  65  54  41  25  8****
******** 20  35  48  60  69  77  82  86  88  88  86  82  77  69  60  48  35  20********
•  ******* 12  27  41  52  62  69  75  79  81  81  79  75  69  62  52  41  27  12********
************ 18  31  43  52  60  65  69  71  71  69  65  60  52  43  31  18************
************ 6  20  31  41  48  54  58  60  60  58  54  48  41  31  20 6************
**************** 6  18  27  35  41  44  46  46  44  41  35  27  18 6****************
************************ 12  20  25  29  31  31  29  25  20  12************************
********************************  8  12  14  14  12   8********************************
*****************************************************************************************
                                   DEPTH RELATIVE TO MEAN WATER LEVEL
```

Depth grid

```
*****************************************************************************************
TIME  •  LATITUDE  •  LONGITUDE * Z   * T-AIR * T-H2O * W-SPD  * W-DIR * CDE3    •    X * Y
*****************************************************************************************
  0.0 * 42.0000000 • 81.0000000 • 10.0 * 4.00 * 4.00 * 10.00 * 270. * 0.00 *  ⊠⊠⊠  •  55.
```

Meteorological data file

Page 2 of printed output from program PARTIC

```
     1.
1      1096.0    -47496.3
2      1036.4    -23739.6
3    -46741.2        20.2
4      1034.0        19.2
     2.
1      1926.9    -47491.8
2      1752.2    -23739.8
3    -46030.7        23.4
4      1745.0        18.5
     3.
1      2807.8    -47480.8
2      2466.2    -23740.3
3    -45324.3        30.1
4      2451.7        17.2
```

Output from subroutine POUTP
(x and y coordinates of new particle
positions for each timestep)

Page 3 of ***printed*** output from program PARTIC

```
      SUBROUTINE INIT (C, H, D, IDIM)
C
C  PURPOSE:
C                A USER-SUPPLIED SUBROUTINE TO INITIALIZE THE
C                CONCENTRATION FIELD IN THE LAKE.
C
C ARGUMENTS :
C                C - CONCENTRATION OF SUBSTANCE
C                H - FREE SURFACE FLUCTUATION FIELD (METERS)
C                D - DEPTH ARRAY (METERS)
C                IDIM - FIRST DIMENSION OF C,H, AND D IN DIMENSION STATEMENT
C                       OF CALLING PROGRAM
C
C COMMON BLOCK :
C                /GPARM/ RPARM(23), IPARM(54)
C                /CPARM/ DT, TT, DADD
C                /BOTTOM/ SED(50,50)
C
      COMMON/CPARM/DT, TT, DADD
      COMMON/GPARM/RPARM(23), IPARM(54)
      COMMON/BOTTOM/SED(50,50)
      DIMENSIONC(IDIM, 1),H(IDIM, 1),D(IDIM, 1)
      DMIN = RPARM(5) + DADD
      IM = IPARM(1)
      JM = IPARM(2)
C
C  INITIALIZE PHYSICS
C
      DO 10 I=1,IM
         DO 10 J=1,JM
            SED(I,J) = 0.
            IF (D(I,J).LT.DMIN) SED(I,J) = -1.
            IF (D(I,J).GE.DMIN) C(I,J) = 1.
   10 CONTINUE

C
C DISPLAY INITIAL CONDITIONS ON LAKE GRID
C
      CALL PRNT(6,C,IDIM, IPARM(1), IPARM(2),-1.E+8)
      RETURN
      END
```

Program ADVEC
Sample user-supplied subroutine

```
      SUBROUTINE UPDATE(T, D, AM, AN, IDIM)
C
C  PURPOSE :
C                  TO UPDATE THE TRANSPORTS BY READING
C                  THE STREAM FUNCTION FIELDS AT INTERVALS
C                  SPECIFIED BY THE USER
C
C  ARGUMENTS:
C                  T - DURATION OF RUN (SECONDS)
C                  D - DEPTH ARRAY (METERS)
C                  AM - COMPONENT OF TRANSPORT IN X-DIRECTION
C                  AN - COMPONENT OF TRANSPORT  IN Y-DIRECTION
C                  IDIM - FIRST DIMENSION OF D, AM, AND AN IN DIMENSION STATEMENT
C                         OF CALLING PROGRAM
C
C  COMMON BLOCK :
C                   /GPARM/ RPARM(23), IPARM(54)
C                   /CPARM/ DT, TT, DADD
C
      COMMON/GPARM/RPARM(23), IPARM(54)
      COMMON/CPARM/DT, TT, DADD
      DIMENSION D(IDIM,1),AM(IDIM,1),  AN(IDIM,1)
      DIMENSION PSI (22, 22)
      DS = RPARM(3)
      DMIN = RPARM(5) + DADD
      IM = IPARM(1)
      JM = IPARM(2)
      IMM1 = IM - 1
      JMM1 =JM - 1
      READ(9) ((PSI(I,J),I=1,IM),J=1,JM)

      DO 10 I=2,IMM1
         DO 10 J=2, JMM1
            AM(I,J) = (PSI(I,J-1) - PSI(I,J))/DS

   10 AN(I,J) = (PSI(I,J) - PSI(I-1,J))/DS
      IF(T .LT. 12000.) CALL PRNT(6,PSI,IDIM,IPARM(1),IPARM(2),-1.)
      RETURN
      END
```

Program ADVEC
Sample user-supplied subroutine

```
      SUBROUTINE PHYSICS  (C,H,D,  U,V,  T,  IDIM)
C
C  PURPOSE:
C                    TO ALLOW THE USER TO INTRODUCE OTHER
C                    MASS-BALANCE TERMS TO THE MODEL
C  ARGUMENTS:
C                    C - CONCENTRATION OF SUBSTANCE
C                    H - FREE SURFACE FLUCTUATION FIELD (METERS)
C                    D - DEPTH ARRAY (METERS)
C                    U - COMPONENT OF CURRENT IN X-DIRECTION (METERS/SECOND)
C                    V - COMPONENT OF CURRENT   IN Y-DIRECTION (METERS/SECOND)
C                    T - DURATION OF RUN (SECONDS)
C                    IDIM - FIRST DIMENSION OF C,H, D,U, AND V
C  COMMON  BLOCK:
C                    /GPARM/RPARM(23),IPARM(54)
C                    /CPARM/DT, TT, DADD
C                    /BOTTOM/SED(50,50)
C
      DIMENSION C(IDIM,1), D(IDIM,1), U(IDIM,1), V(IDIM,1)
      COMMON/GPARM/RPARM(23), IPARM(54)
      COMMON/CPARM/DT, TT, DADD
      COMMON/BOTTOM/SED(50,50)
      DATA WSET, RESUS, FLUX /2.6636E-6, 2.6636E-7, 2.6636E-9/
      IM = IPARM(1)
      JM = IPARM(2)
      IMM1 = IM - 1
      JMM1 = JM - 1
      DS = RPARM(3)
      DMIN = RPARM(5) + DADD
      DO 10 I=2,IMM1
         DO 10 J=2,JMM1
            IF (D(I,J) .LT. DMIN) GO TO 10
            SPDSQ = 0.25*((U(I,J)+U(I-1,J))**2 +
     1              (V(I,J)+V(I,J-1))**2)
            FLUXB = WSET*C(I,J) - RESUS*SPDSQ
            SED(I,J) = SED(I,J) + DT*FLUXB
            C(I,J) = C(I,J) + (DT/D(I,J))*(FLUX-FLUXB)
   10 CONTINUE
      RETURN
      END
```

Program ADVEC
Sample user-supplied subroutine

```
      SUBROUTINE OUTP(C,  H, D, T,  IDIM)
C
C  PURPOSE:
C                    TO GENERATE OUTPUT
C  ARGUMENTS:
C                    C - CONCENTRATION OF SUBSTANCE
C                    H - FREE SURFACE FLUCTUATION FIELD (METERS)
C                    D - DEPTH ARRAY (METERS)
C                    T - DURATION OF RUN (SECONDS)
C                    IDIM - FIRST DIMENSION OF C,H, AND D
C COMMON  BLOCK:
C                    /GPARM/RPARM(23),IPARM(54)
C                    /CPARM/DT, TT, DADD
C                    /BOTTOM/SED(50,50)
C
      COMMON/CPARM/DT, TT, DADD
      COMMON/GPARM/RPARM(23), IPARM(54)
      COMMON/BOTTOM/SED(50,50)
      DIMENSION C(IDIM, 1), D(IDIM, 1), CC(50,50)
      DMIN = RPARM(5) + DADD
      IM = IPARM(1)
      JM = IPARM(2)
      DO 10 I = 1, IM
         DO 10 J = 1, JM
            CC(I,J) = C(I,J)
            IF (D(I,J) .LT. DMIN) CC(I,J) = -1.
   10 CONTINUE
      HRS = T/3600.
      CALL PRNT (6, CC, IDIM, IPARM(1), IPARM(2), -1.)
      WRITE(6,100) HRS
      CALL PRNT (6, SED, IDIM, IPARM(1),  IPARM(2),  -1.)
      WRITE(6,110) HRS
  100 FORMAT ("0", 20X, "CONCENTRATION FIELD AT TIME T = ",F6.2," HOURS")
  110 FORMAT ("0",20X, "SEDIMENT   IN KG/SQUARE METER AT TIME T = ",F6.2,
     1" HOURS")
      RETURN
      END
```

Program ADVEC
Sample user-supplied subroutine

```
24.          24.        0.00
     -1.
```

} Input for program ADVEC on LUN 5

```
DT = 24.          TT  =  24.     DADD =   0.00
BATHYMETRIC DATA FILE HEADER FOR       CIRCULAR PARABOLOID LAKE
IDIMENSION : IPARM(1) =      22      JDIMENSION : IPARM(2) =          22
              DISPLACEMENT OF ORIGIN IN NUMBER OF GRID SQUARES
IDISPLACEMENT  :  IPARM(3)  =  0      JDISPLACEMENT : IPARM(4) =        0
BASE LATITUDE : RPARM(1) =   41.5273336
BASE LONGITUDE : RPARM(2) =  81.6336524
GRID SIZE (M) : RPARM(3) = 5000.     MAX DEPTH (M) : RPARM(4) =       100.
MIN DEPTH (M) : RPARM(5) =    5.     BASE ROTATION : RPARM(6) =        0.00
ANGLE ROTATED : RPARM(7) =    0.
          GEOGRAPHIC-TO-MAP COORDINATE CONVERSION COEFFICENTS FOR X
RPARM(8) =          .833330E+02     RPARM(9) =                0.
RPARM(10) =         0.              RPARM(11) =               0.
          GEOGRAPHIC-TO-MAP COORDINATE CONVERSION COEFFICIENTS FOR Y
RPARM(12) =         0.              RPARM(13) =               .111111E+03
RPARM(14) =         0.              RPARM(15) =               0.
      MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS FOR LONGITUDE
RPARM(16) =         .120000E-01     RPARM(17) =               0.
RPARM(18) =         0.              RPARM(19) =               0.
      MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS FOR LATITUDE
RPARM (20) =        0.              RPARM(21) =               .900000E-02
RPARM(22) =         0.              RPARM (23) =              0.
```

} Output for program ADVEC

} Output from subroutine PGPARM

Page 1 of printed output from program ADVEC

```
VALUES MULTIPLIED BY 10** 0
*****************************************************************************************
*******************************  8  12  14  14  12   8*******************************
************************* 12  20  25  29  31  31  29  25  20  12*************************
***************  6  18  27  35  41  44  46  46  44  41  35  27  18  6***************
************  6  20  31    41  48  54  58  60  60  58  54  48  41  31  20    6************
************  18  31  43  52  60  65  69  71  71  69  65  60  52  43  31  18************
********  12  27  41  52  62  69  75  79  81  81  79  75  69  62  52  41  27  12********
•   ⊠⊠⊠⊠⊠⊠⊠   20 35 48 60 69 77  82  86  88  88  86  82  77  69  60  48  35  20********
**** 8 25 41   54   65   75   82  88  92  94  94  92  88  82  75  65  54  41  25  8****
****  12  29  44  58  69  79  86  92  96  98  98  96  92  86  79  69  58  44  29  12****
****  14  31  46  60  71  81  88  94  98 100 100  98  94  88  81  71  60  46  31  14****
****  14  31  46  60  71  81  88  94  98 100 100  98  94  88  81  71  60  46  31  14****

****  12  29  44  58  69  79  86  92  96  98  98  96  92  86  79  69  58  44  29  12****
****  8  25  41  54  65  75  82  88  92  94  94  92  88  82  75  65  54  41  25  8****
********  227 354148  60  69  77  82  86  88  88  86  82  77  69  60  48  35  20********
********  12        52   62  69  75  79  81  81  79  75  69  62  52  41  27  12********
************  18  31  43  52  60  65  69  71  71  69  65  60  52  43  31 18************
************  6  20  31    41    48  54  58  60  60  58  54  48  41  31   20 6************
***************  6  18  27  35  41  44  46  46  44  41  35  27  18  6***************
************************* 12  20  25  29  31  31  29  25  20  12*************************
*******************************  8  12  14  14  12   8*******************************
*****************************************************************************************
                                                DEPTH RELATIVE TO MEAN WATER LEVEL
```

Depth grid

Page 2 of printed output from program ADVEC

```
VALUES MULTIPLIED BY  10**  3
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0100010001000100010001000   0   0   0   0   0   0   0   0
0   0   0   0   0   0100010001000100010001000100010001000100010001000   0   0   0   0   0
0   0   0   0100010001000100010001000100010001000100010001000100010001000   0   0   0   0
0   0   0100010001000100010001000100010001000100010001000100010001000100010001000   0   0   0
0   0   0100010001000100010001000100010001000100010001000100010001000100010001000   0   0   0
0   0100010001000100010001000100010001000100010001000100010001000100010001000100010001000   0   0
0   0100010001000100010001000100010001000100010001000100010001000100010001000100010001000   0   0
0100010001000100010001000100010001000100010001000100010001000100010001000100010001000100010001000   0
0100010001000100010001000100010001000100010001000100010001000100010001000100010001000100010001000   0
0100010001000100010001000100010001000100010001000100010001000100010001000100010001000100010001000   0
0100010001000100010001000100010001000100010001000100010001000100010001000100010001000100010001000   0
0100010001000100010001000100010001000100010001000100010001000100010001000100010001000100010001000   0
0100010001000100010001000100010001000100010001000100010001000100010001000100010001000100010001000   0
0   0100010001000100010001000100010001000100010001000100010001000100010001000100010001000   0   0
0   0100010001000100010001000100010001000100010001000100010001000100010001000100010001000   0   0
0   0   0100010001000100010001000100010001000100010001000100010001000100010001000   0   0   0
0   0   0100010001000100010001000100010001000100010001000100010001000100010001000   0   0   0
0   0   0   0100010001000100010001000100010001000100010001000100010001000   0   0   0   0
0   0   0   0   0   0100010001000100010001000100010001000100010001000   0   0   0   0   0   0
0   0   0   0   0   0   0   0100010001000100010001000   0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

Concentration:
Initial conditions
from subroutine INIT

Page 3 of printed output from program ADVEC

VALUES MULTIPLIED BY 10** -1

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	82	129	137	127	82	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	81	138	175	205	213	203	174	135	81	0	0	0	0	0	0	0
0	0	0	0	74	127	159	196	226	246	252	244	222	193	157	125	72	0	0	0	0	0
0	0	0	64	116	160	193	222	244	258	262	256	240	216	187	154	111	61	0	0	0	0
0	0	0	50	120	165	196	221	238	248	251	245	232	213	188	156	113	48	0	0	0	0
0	0	59	82	120	154	180	199	212	220	221	216	206	190	170	144	112	77	57	0	0	0
0	0	32	74	105	130	149	163	173	178	178	174	165	153	138	118	95	67	30	0	0	0
0	47	47	62	80	96	108	117	123	125	124	121	115	106	95	83	68	54	42	45	0	0
0	18	28	37	46	54	59	63	65	65	64	61	57	52	46	40	33	27	21	14	0	0
0	2	4	6	7	7	7	6	4	2	0	-2	-4	-6	-7	-7	-7	-6	-4	-2	0	0
0	-14	-21	-27	-33	-40	-46	-52	-57	-61	-64	-66	-65	-63	-59	-54	-46	-37	-28	-18	0	0
0	-45	-42	-54	-69	-83	-95	-106	-115	-121	-125	-125	-123	-117	-108	-96	-80	-62	-47	-47	0	0
0	0	-30	-67	-95	-118	-138	-153	-165	-174	-178	-178	-173	-163	-149	-130	-105	-74	-32	0	0	0
0	0	-57	-77	-112	-144	-170	-190	-206	-216	-221	-220	-213	-199	-180	-154	-120	-82	-59	0	0	0
0	0	0	-48	-113	-156	-188	-213	-232	-245	-251	-248	-238	-221	-196	-165	-120	-50	0	0	0	0
0	0	0	-61	-111	-154	-187	-216	-240	-256	-262	-258	-244	-222	-193	-160	-116	-64	0	0	0	0
0	0	0	0	-72	-125	-157	-193	-223	-244	-252	-246	-226	-196	-159	-127	-74	0	0	0	0	0
0	0	0	0	0	0	-81	-135	-174	-203	-213	-205	-175	-138	-81	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	-82	-127	-137	-129	-82	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Streamfunction field
from subroutine UPDATE

Page 4 of printed output from program ADVEC

```
VALUES MULTIPLIED BY  10**  3
*******************************************************************************
******************************** 971 981 984 984 981 971*******************************
************************* 981 989 991 992 993 993 992 991 989 981*************************
****************962 987  991  993 994 995 995 995 995 994 993 991 987 962****************
************ 962 989 993 994 995 996 996 996 996 996 996 995 994 993 989 962************
************ 987 993 995 996 996 996 997 997 997 997 996 996 996 995 993 987************
******** 981 991 994 996 996 997 997 997 997 997 997 997 997 996 996 994 991 981********
******** 989 993 995 996 997 997 997 997 997 997 997 997 997 997 996 995 993 989********
•    *** 971 991 994 996 996 997 997  997 998 998 998 998 997 997 997 996 996 994 991 971****
**** 981 992 995 996 997 997 997 998 998 998 998 998 998 997 997 997 996 995 992 981****
**** 984 993 995 996 997 997 997 998 998 998 998 998 998 997 997 997 996 995 993 984****
•    *** 984 993 995 996 997 997 997 998 998 998 998 998 998 997 997 997 996 995 993 984****
**** 981 992 995 996 997 997 997 998 998 998 998 998 998 997 997 997 996 995 992 981****
**** 971 991 994 996 996 997997 997 998 998 998 998 997 997 997 996 996 994 991 971****
******** 989 993 995 996 997 997 997 997 997 997 997 997 997 997 996 995 993 989********
******** 981 991 994 996 996 997 997 997 997 997 997 997 997 996 996 994 991 981********
************ 987 993 995 996 996 996 997 997 997 997 996 996 996 995 993 987************
************ 962 989 993 994 995 996 996 996 996 996 996 995 994 993 989 962************
**************** 962 987 991 993 994 995 995 995 995 994 993 991 987 962****************
************************* 981 989 991 992 993 993 992 991 989 981*************************
******************************** 971 981 984 984 981 971*******************************
*******************************************************************************

CONCENTRATION FIELD AT TIME T = 24.00 HOURS
```

Concentration in water column from subroutine PHYSICS

Page 5 of printed output from program ADVEC

```
   VALUES MULTIPLIED BY 10**  3
*********************************************************************************
************************************* 230 230 230 230 230 230*******************************
************************ 230 230 230 230 230 230 230 230 230 230************************
*************** 230 230 230 230 230 230 230 230 230 230 230 230 230 230***************
*********** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230***********
*********** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230***********
******** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230********
******** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230********
**** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230****
**** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230****
**** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230****
**** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230****
**** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230****
**** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230****
******** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230********
******** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230********
*********** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230***********
*********** 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230 230***********
*************** 230 230 230 230 230 230 230 230 230 230 230 230 230 230***************
************************ 230 230 230 230 230 230 230 230 230************************
****************************** 230 230 230 230 230 230******************************
************************************************** ********************************

                SEDIMENT IN KG/SQUARE METER AT TIME T =  24.00 HOURS
```

Accumulation
at the bottom from
subroutine PHYSICS

Page 6 of printed output from program ADVEC