

**NWRI-UNPUBLISHED MANUSCR.**

SIMONS, T J

1982

Simons

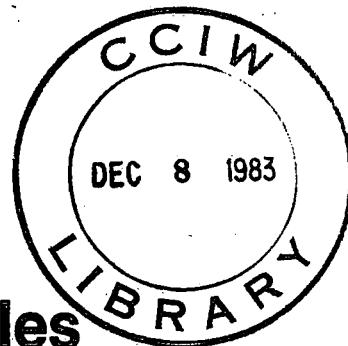


# **Environment Canada**

# **Environnement Canada**

# National Water Research Institute

# **Institut National de Recherche sur les Eaux**



T.J. Simons and D.C.L. Lam

National Water Research Institute

Burlington, Ontario, Canada

# Inland Waters Directorate      Direction Générale des Eaux Intérieures

**DOCUMENTATION OF A TWO-DIMENSIONAL X-Y MODEL PACKAGE  
FOR COMPUTING  
LAKE CIRCULATIONS AND POLLUTANT TRANSPORTS**

**by**

**T.J. Simons and D.C.L. Lam  
National Water Research Institute  
Burlington, Ontario, Canada**

**July 1982**

## **TABLE OF CONTENTS**

	<u>Page</u>
I.	1
Computational grid	1
Depths and boundaries	2
Active and Inactive grid points	4
II.	6
Circulation model	6
Free surface model	8
Stream function	10
Hydrodynamical model ONELAY	12
III.	13
Advection-diffusion model	13
Pollutant transport model POLTRA	16
IV.	18
Examples	18
V.	20
Program listings	20

## I. Computational Grid

The computational grid is a single Richardson lattice with a staggered distribution of variables (Figure 1). The water depth, H, the free surface elevation above the mean, Z, and any scalar variable such as temperature or pollutant concentration, C, are defined at the center of a grid square. They represent averages over this grid element such that, for example, the volume of the element is equal to  $(H+Z)(DS)^2$ , where DS is the mesh size, and the mass of pollutant is this volume multiplied by C. The currents are represented by vertically-integrated water transports per unit width. The I-component of the transport vector, U, is defined at the center of the left and right sides of the grid square, while the J-component, V, is defined at the lower and upper sides. These transport components represent averages over the respective sides of the grid element such that, for example, the water transport through the left side is equal to U.DS and the corresponding pollutant transport is U.C.DS. Finally, the transport stream function, S, is defined at the corners of the grid square.

For economy of computer storage, each variable array has its own allocation of subscripts in relation to grid location. A particular combination of subscripts (I,J), therefore, refers to different

FIG. 1

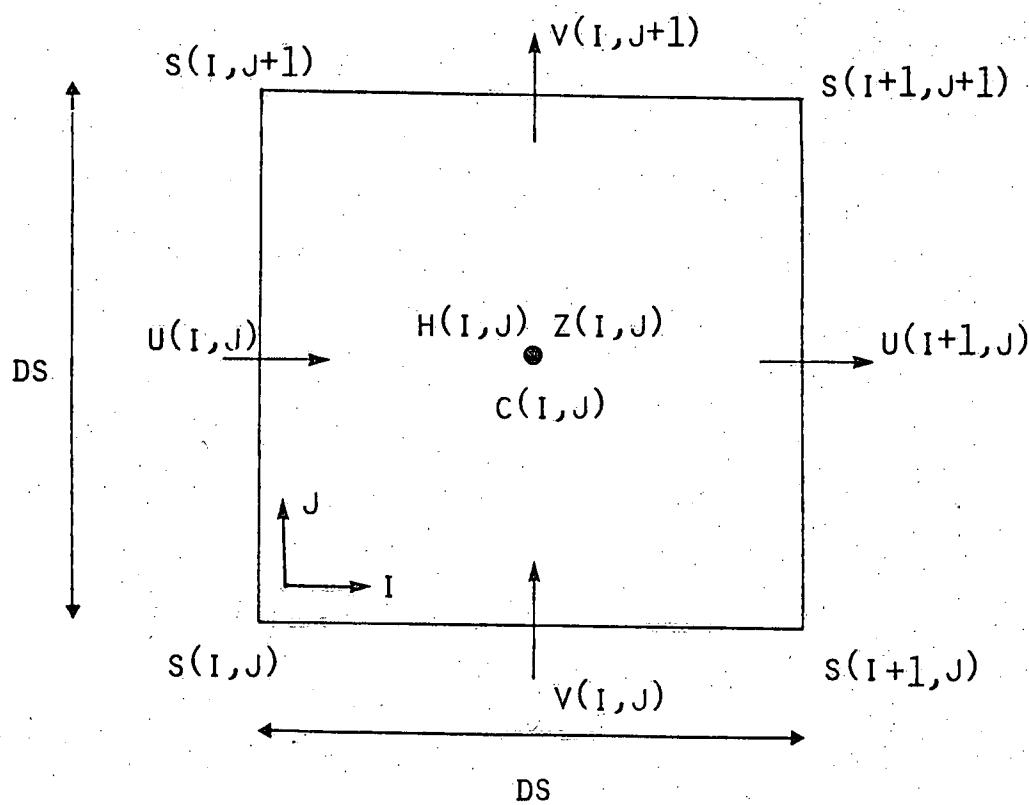
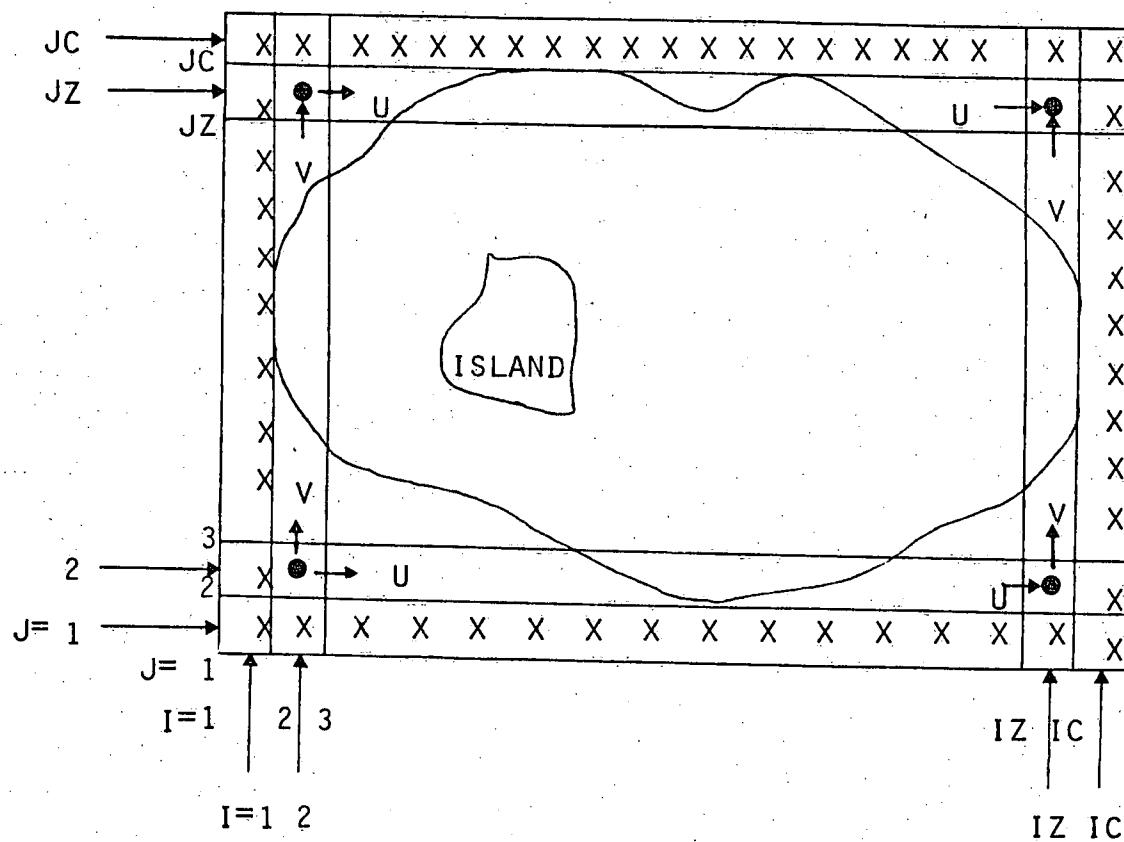


FIG. 2



locations depending on the variable concerned. The convention adopted here is that all variables at the lower left corner, the left side, and the lower side of the grid element have the same subscripts as those at the center (Figure 1). This means, for example, that the transport through the right side of the element has an I-index which is one higher than the center-index of this element and which is equal to the center-index of the adjacent element on the right.

#### Depths and Boundaries

The overall gridmesh has a rectangular shape (Figure 2). In principle, the dimensions of the arrays, H, Z, C, to which we will refer as elevation points, can be one less than the dimensions of U, V, S, which we call current or stream points, because the number of sides is one higher than the number of squares. To avoid this kind of complication, all variable arrays are given the same dimensions, namely, (IC, JC) where IC, JC represent the maximum number of current points. The maximum number of elevation points which are actually used are (IZ, JZ) where  $IZ = IC-1$ ,  $JZ = JC-1$ . Similarly, it can be verified from Figure 2 that the maximum number of U-components, including the boundaries, will be (IC, JZ) and the number of V-components will be (IZ, JC).

In order to simplify the computer code, it is convenient if the grid is surrounded by one row and one column of zero depths, H,

and concentrations, C. By virtue of the above convention adopted for the array dimensions, this condition is fulfilled on the right and the upper border of the grid. It remains to satisfy the same condition on the left and the lower border. This is done by setting the first row ( $J=1$ ) and the first column ( $I=1$ ) of the depth array equal to zero. Thus, the depths and pollutant concentrations are zero in the grid elements denoted by crosses in Figure 2, and the basin of interest is confined within this frame of crosses.

The shape and bathymetry of the basin are completely and exclusively determined by the depth array H, the depths being zero on land or on islands. The shorelines of the basin coincide locally with either one of the sides of the grid elements, resulting in a staircase boundary. Given the depths of the grid cells, the computer program determines which cell side is part of the shoreline by checking if one of the two adjacent cell depths is zero and the other nonzero. In order to locate open boundaries at the entrance or exit of rivers, the depths of gridcells located in such rivers must be set equal to zero while the transport normal to the boundary of the basin must be prescribed at such a river entrance or exit.

There is another possibility to deal with open boundaries, that is, to prescribe the elevation, Z, in a grid point just outside the basin, say, in the river. Because of the way in which the program decides whether or not to perform computations in a given point (as

will be discussed next) the depth in this particular river point should be zero for the purpose of Z-calculations or C-calculations, but nonzero for the purpose of U- or V-calculations. While this kind of change is easy to implement, it is not done in the present version of the model since it obviously requires some familiarity with the details of the computer program and could easily lead to complications.

There are no restrictions on the shape of the basin or any islands in the basin. Channels may be as narrow as one single grid cell. However, it should be realized that no dynamical connection exists between adjacent grid squares unless they have at least one side in common. Thus two squares which touch each other in one corner only are essentially disconnected.

#### Active and Inactive Grid Points

The program differentiates between active points where calculations are to be performed and inactive points where variables are either zero or are defined by the user. Inspection of Figure 2 shows that the indices of the active points must be confined between the following limits:

z-points	I=2,IZ	J=2,JZ
U-points	I=3,IZ	J=2,JZ

V-points      I=2,IZ      J=3,JZ

S-points      I=3,IZ      J=3,JZ

Within these limits, the type of any one gridpoint is completely determined by the depth array H. Thus, variables Z or C are calculated only if the local value of H is nonzero. Variable U is computed only for those grid sides for which both adjacent squares have nonzero H-values and similarly for variable V. If either one of the adjacent H-values is zero, the transport across a grid side must be zero at a regular shoreline, including island boundaries, or it must be prescribed if one is dealing with a river entrance or exit as noted above.

The points outside the above limits can never be active grid points but they may be used as inactive points. Thus, for the elements denoted by crosses in Figure 2, the depths must always be zero but the variable C may have a prescribed nonzero value, for instance at the entrance point of a river (where, as noted above, the depth-array must be assigned a zero value). The same holds for the transports U or V at a river entrance or exit. Thus, for example, variable U with subscript I = 2 could never be an active point but it can be a river entrance or exit.

As a final word of caution, it should be emphasized that, if variables are to be prescribed, the corresponding indices refer to the

location of the particular variable, not necessarily to the grid square. Thus, if the grid square of Figure 1 were inside the lake such that a river would flow out of this square toward the right, then  $H(I+1,J)$  would be zero and one would prescribe  $U(I+1,J)$ , not  $U(I,J)$ .

## II Circulation Model

The present framework of the pollutant transport model is based on the assumption that the hydrodynamic computations are completed before the water quality calculations are started. Thus, the present package consists of two parts. The hydrodynamic program takes the depth array,  $H$ , any prescribed river inflows or outflows and any forcing by wind, if desired, and it computes the water levels,  $Z$ , and the water transport components  $U,V$ . These results are then stored in some fashion and perhaps averaged in time. The pollutant transport model takes the same depth array and the calculated water transports and carries out the desired water quality computations.

The circulation model included here computes vertically-averaged currents in a basin which is relatively well-mixed in the vertical. Such models can be formulated in two ways. The first type of model was originally designed for storm surge predictions and hence it computes free surface elevations as well as water circulations. The second type of model eliminates gravity waves by requiring the vertically integrated currents to be nondivergent, thus

allowing them to be represented by a streamfunction. From a computational viewpoint the main difference between the two models is that the free surface model allows for a longer step but requires a large number of iterations per time step. Thus, they are more or less equivalent in terms of economy.

If one is interested in advective processes, as is the case here, it is generally sufficient to use the nondivergent part of the vertically-integrated current. This is, in fact, advisable since it is of crucial importance to ensure conservation of water mass in a pollutant transport model. While this condition can be met by a free surface model, it is always satisfied by a streamfunction model. However, it is felt that the present water quality model should also be applicable to situations where the water circulation has been computed previously or is available from interpolation of observed currents. Therefore, the circulation model included here offers this option. First, the hydrodynamic calculations are performed. The model selected here is a free surface model. Next the nondivergent part of the computed vertically-integrated currents are obtained. Naturally, the second step can be eliminated if the hydrodynamic calculations have been done by a streamfunction model.

Free surface model

The model computes the free surface position and vertically integrated currents for given wind stress and prescribed transports of inflowing and outflowing rivers. Nonlinear accelerations are neglected, rotation is included, bottom friction can be linear or nonlinear, and horizontal diffusion is not needed for stability.

The equations are:

$$\frac{\partial U}{\partial t} = fV - gH \frac{\partial Z}{\partial x} - BU + \tau_{sx}$$

$$\frac{\partial V}{\partial t} = -fU - gH \frac{\partial Z}{\partial y} - BV + \tau_{sy}$$

$$\frac{\partial Z}{\partial t} = -\frac{\partial U}{\partial x} - \frac{\partial V}{\partial y}$$

where  $t$  is time,  $x, y$  are the horizontal coordinates ( $x$  clockwise from  $y$ ),  $U, V$  are the corresponding components of the vertically integrated current,  $Z$  is the upward displacement of the free surface from a mean level,  $H$  is the depth of the basin for this mean level,  $B$  is a bottom stress coefficient,  $f$  is the Coriolis parameter equal to twice the angular velocity of the earth's rotation times the sine of latitude,  $g$  is the gravitational acceleration, and  $\tau_s$  is the wind stress at the surface divided by the water density. The units are cm and sec.

The bottom stress coefficient is given by any one of the following formulations:

$$\text{nonlinear } B = C_D (U^2 + V^2)^{1/2}/H^2 \quad C_D \sim 0.002$$

$$\text{linear } B = a/H \quad a \sim 0.01 - 0.05 \text{ cm/sec}$$

$$\text{quasilinear } B = b/H^2 \quad b \sim 10 - 100 \text{ cm}^2/\text{sec}$$

The units of  $B$  are  $\text{sec}^{-1}$ .

The wind stress is assumed to have been computed previously, for example from the familiar relation

$$\vec{\tau}_s = C_D \frac{\rho_a}{\rho} \vec{W} \vec{W}$$

where  $\rho_a$  is the air density,  $\rho$  is the water density,  $\vec{W}$  is the wind vector, and  $C_D$  is a drag coefficient. The units of the stress are  $\text{cm}^2/\text{sec}^2$ .

The transports of inflowing or outflowing rivers are specified as vertically-integrated currents in units of  $\text{cm}^2/\text{sec}$ , that is, total river discharge divided by the grid size. The discharge from a single river may be distributed over more than one grid cell as long as the sum of the prescribed transports due to that river is equal to the total discharge divided by the grid size. The same holds with

regard to distribution over the two component of the vertically-integrated current, U and V.

The variables are defined on the staggered grid of Figure 2 and the derivatives in time and space are approximated by central differences. The Coriolis term requires spatial averaging, which is applied here to the vertically-integrated currents rather than the currents themselves. It is known that a more complicated averaging scheme has advantages in terms of energy conservation but the present scheme is more economical and probably satisfactory for most purposes.

Time extrapolation proceeds by first predicting Z, then U, then V, at all times utilizing the last available values of all variables. This method is stable as long as

$$\Delta t < DS/\sqrt{2gH_{\max}} \text{ and } \Delta t < 1/B_{\max}$$

where  $\Delta t$  is the time step and DS is the space step.

#### Streamfunction

In case the circulation is computed by a free surface model, the divergence of the vertically-integrated current is in general nonzero. While the divergent part of the transport determines the variations of the surface level, it is not important for pollutant

transport. Therefore, the advection program uses only the non-divergent part of the transport, that is, the part that can be represented by a streamfunction. Formally, this is done by writing

$$U = \frac{\partial P}{\partial x} - \frac{\partial S}{\partial y} \quad V = \frac{\partial P}{\partial y} + \frac{\partial S}{\partial x}$$

where  $P$  is the transport potential representing the divergent part of the transport and  $S$  is the streamfunction for the non-divergent part. Cross-differentiation gives the vorticity

$$F = \frac{\partial V}{\partial x} - \frac{\partial U}{\partial y} = \frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2}$$

Thus, given the transport,  $U$ ,  $V$ , the vorticity,  $F$ , may be computed in each point of the grid. Then the stream function,  $S$ , may be computed from the Poisson equation

$$\frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} = F$$

with  $S$  prescribed on the boundaries. This is done by relaxation.

The boundary values of the stream function are prescribed as follows. Along any boundary-element, be it open or closed, the stream function must increase in accordance with the above definition, thus

$$S(I+1,J) = S(I,J) + DS.V(I,J)$$

$$S(I,J+1) = S(I,J) - DS.U(I,J)$$

where V or U are the transport components normal to the boundary segment of interest. The stream function may be set to zero in an arbitrary point of the shoreline; its value along the whole boundary follows then from these relations. The value of the stream function in an arbitrary point of an island coast is found by applying the same relationships across any channel between the island and the outer boundary.

#### Hydrodynamical model ONELAY

The computer program ONELAY refers to the above one-layer free-surface circulation model with a single lattice. For definitions of variables refer to program listing. The following are inputs to be supplied by user in form of cards or tape file:

##### Main Routine:

1. TTHRS, DTMAP, DT

Routine DEPTH

2. IC, JC, DS
3. ((H(I,J), I=1,IC), J=1,JC)

Routine RIVERS

4. KU, KV
5. (IU(K), JU(K), K = 1,KU)
6. (UU(K), K = 1,KU)
7. (IV(K), JV(K), K = 1,KV)
8. (VV(K), K = 1,KV)

Routine STRESS

9. TSXX, TSYY

Routine PREDIC

10. F, G

**III. Advection-diffusion Model**

The calculations are performed on the single-lattice of

Figure 1. The governing equation is

$$\frac{\partial}{\partial t} (HC)_{ij} = X_{ij} - X_{i+1,j} + Y_{ij} - Y_{i,j+1} - W_{ij} C_{ij} + S_{ij}/DS^2$$

where H is the depth, C is the concentration of the pollutant of interest (grams/cm<sup>3</sup>), S is the source term (grams/sec), W is the settling velocity (cm/sec), and X and Y are the transport components through the sides of the grid square

$$X_{ij} = \frac{U_{ij}}{DS} (C_{i-1,j} \text{ or } C_{ij}) + \frac{A}{DS^2} \cdot \frac{H_{i-1,j} + H_{ij}}{2} (C_{i-1,j} - C_{ij})$$

$$Y_{ij} = \frac{V_{ij}}{DS} (C_{i,j-1} \text{ or } C_{ij}) + \frac{A}{DS^2} \cdot \frac{H_{i,j-1} + H_{ij}}{2} (C_{i,j-1} - C_{ij})$$

where U and V are the components of the vertically-integrated current and A is the horizontal diffusion coefficient (cm<sup>2</sup>/sec). Note that the transports across a closed boundary are zero.

The decision to use  $C_{i-1,j}$  or  $C_{ij}$  etc. is based on the upstream differencing scheme (UDS), in which only upstream values are assumed to be essential in advective processes. This will add a numerical diffusion in the order of  $U_{ij} \cdot DS / H_{ij}$  and  $V_{ij} \cdot DS / H_{ij}$  to the physical diffusivity A. Thus, for systems where high accuracy of the physical diffusion is required, the grid spacing must be reduced to offset this numerical effect. On the other hand,

averaged concentrations  $(C_{i-1,j} + C_{ij})/2$  and  $(C_{i,j-1} + C_{ij})/2$  can be used throughout the computation, if a central differencing scheme (CDS) is used. However, spatial instability may occur, if  $U_{ij} DS / H_{ij} A$  exceeds 2, or at locations with strong sources or sinks. The advection-diffusion program POLTRA contains a parameter NUMTHD which the user may specify as 1 if the user prefers to use UDS, or as 2 if CDS is chosen.

As in the case of the hydrodynamic model, explicit time differencing is used in the advection-diffusion model. The following time stability criteria must be satisfied:

$$\frac{U_{ij} \Delta t}{H_{ij} DS} < 1$$

and  $\frac{A \Delta t}{DS^2} < \frac{1}{2}$

The above conditions on numerical spatial dispersion and time stability may be modified somewhat if more than one variable, which react kinetically with each other, are used.

The advection-diffusion program POLTRA allows a system of three variables in its present form, but extension to more than three variables is straightforward by changing the appropriate DIMENSION declaration statements. The arrays related to the Nth variables are

$C(I,J,N)$  and  $CC(I,J,N)$  for concentration values at the current and updated time steps, respectively.

The program POLTRA is structured in the same way as the program ONELAY in terms of grid convention and setup of lake bathymetry (routine DEPTH). It is assumed that the vertically-integrated currents U, V have been computed previously, for instance by program ONELAY, and have been stored on cards or tape. These integrated currents are now read by routine FLOW, whereupon the nondivergent part of the transports is determined to guarantee mass conservation with a streamfunction model (routine STREAM). Then, the program proceeds to compute the sources (routine SOURCE) including those from inflow and rivers and the sinks including outflows and settling effects. This is followed by the advection and diffusion (routine ADVECT) of the concentration as well as the biochemical reactions (routine BIOCHM), if any. The three routines, SOURCE, ADVECT and BIOCHM are called for each time step and each variable, one at a time. The routine BIOCHM, in particular, should be written by the user for any given problem, including the case where no biochemical reactions are present, for which case a dummy RETURN is needed.

#### Pollution Transport Model POLTRA

The computer program POLTRA is a one-layer two-dimensional x-y advection and diffusion model for pollutant transport. For

definitions refer to program listing. The following are inputs to be supplied by user in form of cards or tape files:

Main Routine:

1. NSTEP, NMAP, DT, NUMTHD, DIFF  
NC, (CO(N), N = 1, NC)

Routine DEPTH:

2. IC, JC, DS
3. ((H(I,J), I=1,IZ), J = 1,JZ)

Routine FLOWS:

4. ((U(I, J), I = 1, IU), J = 1, JV)
5. ((V (I,J), I = 1, IU), J = 1, JV)

Routine SOURCE:

6. KLOAD, KRIVER, KLOSS
7. (ILOAD (K), JLOAD (K), K = 1, KLOAD)
8. ((CLOAD (K, M), K = 1, KLOAD), M = 1, NC)
9. (IRIVER (K), JRIVER (K), K = 1, KRIVER)
10. ((CRIVER (K,M), K = 1, KRIVER), M = 1, NC)

11. (ILOSS (K), JLOSS (K), K = 1, KLOSS)
12. (WSET(M), M = 1, NC)

Routine BIOCHM:

13. RHO, TEMP, KI, CSAT

**IV. Examples**

The examples given in the program listing are for demonstrative purpose only; some model parameters may need to be adjusted, such as wind stress, water temperature and light factors. The 'lake' is about  $3375 \times 2440 \text{ m}^2$  in area with maximum depth of 6.35 m. There are two islands in the lake, one inflow affecting 5 grid points, one outflow affecting 4 grid points, and five source locations of pollutant entries other than those associated with inflowing rivers.

The time step is 6 sec. for the hydrodynamic calculations for a 'cold start' initial condition, i.e.  $Z=U=V=0$ . The time to reach steady-state is about 2 hours, for which the computational time is 224 sec. using a CDC Cyber 170. Zero wind stresses are used.

On the other hand, the pollutant transport model uses 30 sec. as one time step, and computes 10 hours' simulation in about

960 sec. using a CDC Cyber 170 for a system of two reacting variables. One variable is particulate phosphorus and the other is soluble reactive phosphorus, with initial concentration of  $0.5 \times 10^{-8}$  gm/c.c. and  $1.0 \times 10^{-8}$  gm/c.c. respectively. The former represents the phosphorus in planktons and the latter the nutrient (or pollutant) in the lake. The chemical kinetic equations are given in routine BIOCHM.

V. Program Listings

Program ONELAY  
Program POLTRA

PROGRAM ONELAY (INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT,TAPE1)

C ONE-LAYER FREE-SURFACE CIRCULATION MODEL WITH SINGLE LATTICE

C

C

C REFERENCE : DOCUMENTATION OF A TWO-DIMENSIONAL X-Y MODEL PACKAGE  
C FOR COMPUTING LAKE CIRCULATIONS AND POLLUTANT TRANSPORTS  
C BY T. J. SIMONS AND D. C. L. LAM (1982)

C

C

C H = DEPTH (CM), Z = SURFACE ELEVATION (CM), U,V = TRANSPORT (CM\*\*2/SEC)  
C DIMENSIONS OF ARRAYS ARE (IC,JC) AS DEFINED IN ROUTINE DEPTH

DIMENSION H(48,35),Z(48,35),U(48,35),V(48,35)

C TTHRS = TOTAL TIME FOR WHICH COMPUTATION IS CARRIED OUT (HOURS)

C DTMAP = OUTPUT INTERVAL (HOURS), DT = TIMESTEP (SECONDS)

READ (5,1001) TTHRS,DTMAP,DT

WRITE(6,1000)

WRITE(6,1001) TTHRS,DTMAP,DT

DTHRS=DT/3600.

NSTEP=TTHRS/DTHRS+.1

THRS=0

NMAPS=DTMAP/DTHRS+.1

KMAPS=0

CALL DEPTH(IC,JC,IZ,JZ,DS,H)

DO 20 I=1,IC

DO 20 J=1,JC

Z(I,J)=0.

U(I,J)=0.

20 V(I,J)=0.

DO 50 NTIME=1,NSTEP

THRS=THRS+DTHRS

CALL RIVERS(THRS,U,V)

CALL STRESS(THRS,DT,TSX,TSY)

CALL PREDIC(IZ,JZ,DT,DS,TSX,TSY,H,Z,U,V)

KMAPS=KMAPS+1

IF (KMAPS.LT.NMAPS) GO TO 50

KMAPS=0

CALL MAPOUT(IC,JC,THRS,H,Z,U,V)

50 CONTINUE

1000 FORMAT(1H1)

1001 FORMAT(8F10.2)

END

```
SUBROUTINE DEPTH(IC,JC,IZ,JZ,DS,H)
C
C READ DIMENSIONS OF ARRAYS (IC,JC) AND GRIDMESH DS IN CM
C READ DEPTHS OF GRIDSQUARES IN CM AND STORE IN ARRAY H(I,J)
C DEPTHS MUST BE ZERO FOR I=1, FOR I=IC, FOR J=1, AND FOR J=JC
C DEPTH OUTSIDE BASIN MUST BE ZERO EVEN IF GRID CELL LIES WITHIN A RIVER.
C
DIMENSION H(48,35)
C
READ (5,1001) IC,JC,DS
WRITE(6,1001) IC,JC,DS
IZ=IC+1
JZ=JC-1
C
DO 1 I=1,IC
DO 1 J=1,JC
1 H(I,J)=0.
DO 2 I=1,IZ
2 READ (1,1002) (H(I,J),J=1,JZ)
C
WRITE(6,1000)
DO 4 JJ=1,JC
J=JC-JJ+1
4 WRITE(6,1003) (H(I,J),I=1,24)
WRITE(6,1000)
DO 5 JJ=1,JC
J=JC-JJ+1
5 WRITE(6,1003) (H(I,J),I=25,IC)
C
1000 FORMAT(1H1)
1001 FORMAT(2I5,F10.0)
1002 FORMAT(24F3.0)
1003 FORMAT(1H0,24F5.0)
RETURN
END
```

SUBROUTINE RIVERS(THRS,U,V)

C THIS ROUTINE SPECIFIES INFLOWS OR OUTFLOWS DUE TO RIVERS  
C NO DISTINCTION IS MADE BETWEEN INFLOWS OR OUTFLOWS.  
C KU = NUMBER OF POINTS WHERE U - COMPONENTS ARE TO BE PRESCRIBED  
C IU,JU = INDICES OF THESE POINTS IN TWO-DIMENSIONAL GRID  
C UU = VALUE OF U - COMPONENT IN CM\*\*2/SEC  
C KV,IV,JV,VV = SAME FOR V - COMPONENTS  
C

```
DIMENSION U(48,35),V(48,35)
DIMENSION IU(10),JU(10),UU(10),IV(10),JV(10),VV(10)

DATA      NCALL/0/
IF(NCALL.GT.0) GO TO 1

READ (5,1001) KU,KV
WRITE(6,1001) KU,KV
READ (5,1001) (IU(K),JU(K),K=1,KU)
WRITE(6,1001) (IU(K),JU(K),K=1,KU)
READ (5,1002) (UU(K),K=1,KU)
WRITE(6,1002) (UU(K),K=1,KU)
READ (5,1001) (IV(K),JV(K),K=1,KV)
WRITE(6,1001) (IV(K),JV(K),K=1,KV)
READ (5,1002) (VV(K),K=1,KV)
WRITE(6,1002) (VV(K),K=1,KV)

1 NCALL=NCALL+1
IF(THRS.GT.1.) RETURN

DO 10 K=1,KU
I=IU(K)
J=JU(K)
10 U(I,J)=UU(K)*THRS
DO 20 K=1,KV
I=IV(K)
J=JV(K)
20 V(I,J)=VV(K)*THRS

1001 FORMAT(16I5)
1002 FORMAT(8E10.3)
RETURN
END
```

```
C SUBROUTINE STRESS(THRS,DT,TSX,TSY)
C THIS ROUTINE SPECIFIES WINDSTRESS COMPONENTS TSX,TSY (CM**2/SEC**2)
C DATA NCALI/0/
C IF(NCALL.GT.0) GO TO 1
C
C READ (5,1001) TSXX,TSYY
C WRITE(6,1001) TSXX,TSYY
C
1 NCALL=NCALL+1
TSX=DT*TSXX
TSY=DT*TSYY
C
1001 FORMAT(8F10.2)
RETURN
END
```

```
SUBROUTINE PREDIC(IZ,JZ,DT,DS,TSX,TSY,H,Z,U,V)
```

```
C PREDICTION OF SURFACE ELEVATION AND INTEGRATED WATER TRANSPORT  
C SINGLE LATTICE
```

```
DIMENSION H(48,35),Z(48,35),U(48,35),V(48,35)  
DIMENSION GDHU(1680),GDHV(1680),BKHU(1680),BKHV(1680)
```

```
DATA NCALL/0/  
IF (NCALL.GT.0) GO TO 40
```

```
C F = CORIOLIS PARAMETER, G = GRAVITY
```

```
READ(5,1001) F,G  
WRITE(6,1001) F,G  
DTDS=DT/DS  
FDT=F*DT/4.  
GDS=G*DT/DS
```

```
C BFRIC = BOTTOM FRICTION COEFFICIENT  
C HERE LINEAR FRICTION IS USED  
C FOR NONLINEAR BOTTOM FRICTION USE CARDS STARTING WITH CCC
```

```
BFRIC=.05*DT  
CCC BFRIC=.0025*DT
```

```
C FOR ECONOMY OF CALCULATION USE DUMMY ARRAYS
```

```
N=0  
DO 20 I=3,IZ  
DO 20 J=2,JZ  
N=N+1  
GDHU(N)=0.  
IF(H(I-1,J).LE.0..OR.H(I,J).LE.0.) GO TO 20  
HU=(H(I-1,J)+H(I,J))/2.  
GDHU(N)=GDS*HU  
BKHU(N)=BFRIC/HU  
CCC BKHU(N)=BFRIC/(HU*HU)  
20 CONTINUE
```

```
N=0  
DO 30 I=2,IZ  
DO 30 J=3,JZ  
N=N+1  
GDHV(N)=0.  
IF(H(I,J-1).LE.0..OR.H(I,J).LE.0.) GO TO 30  
HV=(H(I,J-1)+H(I,J))/2.  
GDHV(N)=GDS*HV  
BKHV(N)=BFRIC/HV  
CCC BKHV(N)=BFRIC/(HV*HV)  
30 CONTINUE
```

40 NCALL=NCALL+1

C SURFACE PREDICTION

```
DO 50 I=2,IZ
DO 50 J=2,JZ
IF(H(I,J).LE.0.) GO TO 50
Z(I,J)=Z(I,J)+DTDS*(U(I,J)-U(I+1,J)+V(I,J)-V(I,J+1))
50 CONTINUE
```

C PREDICTION OF U-COMPONENT

```
N=0
DO 60 I=3,IZ
DO 60 J=2,JZ
N=N+1
IF(GDHU(N).LE.0.) GO TO 60
UBAR=V(I-1,J)+V(I-1,J+1)+V(I,J)+V(I,J+1)
OMB=1.-BKHU(N)
CCC OMB=1.-BKHU(N)*SQRT(U(I,J)**2+UBAR**2/16.)
U(I,J)=OMB*U(I,J)+FDT*UBAR+GDHU(N)*(Z(I-1,J)-Z(I,J))+TSX
60 CONTINUE
```

C PREDICTION OF V-COMPONENT

```
N=0
DO 70 I=2,IZ
DO 70 J=3,JZ
N=N+1
IF(GDHV(N).LE.0.) GO TO 70
UBAR=U(I,J-1)+U(I,J)+U(I+1,J-1)+U(I+1,J)
OMB=1.-BKHV(N)
CCC OMB=1.-BKHV(N)*SQRT(UBAR**2/16.+U(I,J)**2)
U(I,J)=OMB*U(I,J)-FDT*UBAR+GDHV(N)*(Z(I,J-1)-Z(I,J))+TSY
70 CONTINUE
```

1001 FORMAT(8E10.2)

RETURN

END

SUBROUTINE MAPOUT(IC,JC,THRS,H,Z,U,V)

C MAPS OF SURFACE ELEVATION AND VELOCITY COMPONENTS

DIMENSION H(48,35),Z(48,35),U(48,35),V(48,35)  
DIMENSION D(48,35),C(48,35)

```
DO 1 I=1,IC
DO 1 J=1,JC
D(I,J)=U(I,J)/100.
C(I,J)=V(I,J)/100.
1 CONTINUE

1 WRITE(6,1011) THRS
DO 2 JJ=1,JC
J=JC-JJ+1
2 WRITE(6,1003) (Z(I,J),I=1,24)
WRITE(6,1011) THRS
DO 3 JJ=1,JC
J=JC-JJ+1
3 WRITE(6,1003) (Z(I,J),I=25,IC)

WRITE(6,1012) THRS
DO 4 JJ=1,JC
J=JC-JJ+1
4 WRITE(6,1003) (D(I,J),I=1,24)
WRITE(6,1012) THRS
DO 5 JJ=1,JC
J=JC-JJ+1
5 WRITE(6,1003) (D(I,J),I=25,IC)

WRITE(6,1013) THRS
DO 6 JJ=1,JC
J=JC-JJ+1
6 WRITE(6,1003) (C(I,J),I=1,24)
WRITE(6,1013) THRS
DO 7 JJ=1,JC
J=JC-JJ+1
7 WRITE(6,1003) (C(I,J),I=25,IC)

1003 FORMAT(1H0,24F5.0)
1011 FORMAT(1H1,' TIME IN HOURS'F6.2,'-DT      SURFACE ELEVATION (CM)')
1012 FORMAT(1H1,' TIME 'F6.2,' HRS    U-TRANSPORT (100 CM**2/S)')
1013 FORMAT(1H1,' TIME 'F6.2,' HRS    V-TRANSPORT (100 CM**2/S)')
RETURN
END
```

48 200 100 66  
35 7161.

0.	J.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	31.	76.	104.	104.	126.	111.	121.	124.	138.	124.	114.	121.	141.	134.	124.	124.	124.	131.	69.	0.	0.	0.	0.	0.	0.	0.		
0.	0.	31.	104.	135.	155.	155.	168.	155.	168.	168.	168.	165.	162.	155.	158.	185.	175.	172.	172.	175.	131.	104.	0.	0.	0.	0.	0.	0.	
0.	0.	61.	165.	209.	216.	216.	216.	216.	216.	216.	216.	216.	216.	216.	216.	226.	226.	216.	216.	216.	216.	216.	216.	216.	216.	216.	155.		
0.	0.	0.	64.	185.	236.	257.	270.	273.	267.	253.	257.	270.	270.	270.	270.	263.	250.	250.	250.	250.	250.	250.	250.	250.	250.	250.	206.		
0.	0.	0.	21.	111.	196.	256.	301.	311.	297.	277.	270.	290.	317.	338.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	246.	
0.	0.	0.	0.	42.	141.	219.	280.	338.	334.	297.	297.	324.	362.	375.	362.	345.	348.	321.	314.	321.	317.	297.	311.	307.	307.	304.	0.		
0.	0.	0.	0.	0.	71.	156.	223.	277.	355.	335.	304.	287.	704.	317.	338.	331.	338.	338.	341.	345.	365.	363.	334.	317.	317.	317.	317.	317.	
0.	0.	0.	0.	0.	0.	64.	125.	199.	331.	469.	423.	365.	345.	348.	345.	334.	345.	362.	363.	372.	375.	375.	366.	355.	355.	338.	338.	338.	
0.	0.	0.	0.	0.	0.	100.	331.	433.	473.	443.	413.	419.	419.	406.	399.	402.	416.	416.	402.	399.	395.	375.	365.	361.	361.	361.	361.	361.	355.
0.	0.	0.	0.	0.	0.	103.	267.	355.	412.	419.	405.	422.	433.	446.	453.	436.	429.	419.	412.	416.	402.	399.	399.	399.	399.	399.	399.	399.	
0.	0.	0.	0.	0.	0.	39.	113.	165.	243.	307.	354.	355.	361.	361.	361.	361.	361.	361.	422.	422.	422.	422.	422.	422.	422.	422.	422.	383.	
0.	0.	0.	0.	0.	0.	42.	143.	199.	253.	297.	317.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	331.	
0.	0.	0.	0.	0.	0.	14.	125.	213.	267.	290.	300.	334.	358.	355.	351.	345.	375.	433.	470.	443.	429.	436.	456.	456.	456.	456.	456.	399.	
0.	0.	0.	0.	0.	0.	57.	169.	236.	229.	159.	233.	307.	338.	324.	314.	301.	348.	436.	430.	456.	435.	446.	467.	467.	473.	473.	473.	429.	
0.	0.	0.	0.	0.	0.	57.	139.	125.	62.	132.	233.	240.	222.	236.	240.	307.	339.	453.	460.	4F3.	477.	504.	521.	548.	570.	570.	570.		
0.	0.	0.	0.	0.	0.	0.	35.	121.	111.	0.	32.	91.	135.	165.	133.	213.	301.	395.	450.	453.	494.	507.	541.	564.	592.	592.	538.		
0.	0.	0.	0.	0.	0.	0.	38.	127.	117.	3.	0.	0.	0.	25.	77.	169.	229.	331.	416.	473.	517.	531.	533.	565.	582.	582.	582.	582.	538.
0.	0.	0.	0.	0.	0.	0.	41.	132.	122.	0.	0.	0.	24.	114.	216.	267.	316.	426.	437.	541.	589.	555.	572.	575.	575.	575.	575.	497.	
0.	0.	0.	0.	0.	0.	43.	136.	116.	3.	0.	0.	63.	142.	145.	226.	294.	379.	450.	545.	533.	546.	531.	531.	531.	531.	531.	531.		
0.	0.	0.	0.	0.	0.	56.	142.	122.	107.	134.	133.	126.	129.	141.	186.	24t.	316.	399.	430.	534.	507.	490.	470.	460.	460.	460.	460.	423.	
0.	0.	0.	0.	0.	0.	99.	153.	115.	152.	145.	114.	51.	43.	53.	125.	209.	290.	341.	332.	412.	413.	429.	409.	396.	396.	396.	396.	385.	
0.	0.	0.	0.	0.	0.	119.	175.	155.	51.	0.	0.	0.	105.	149.	250.	314.	355.	372.	355.	382.	382.	351.	351.	351.	351.	351.	351.		
0.	0.	0.	0.	0.	0.	52.	157.	171.	146.	3.	0.	0.	53.	92.	131.	172.	219.	237.	342.	362.	368.	362.	345.	321.	328.	321.	321.	321.	
0.	0.	0.	0.	0.	0.	54.	161.	158.	70.	0.	0.	43.	99.	114.	123.	132.	152.	291.	267.	314.	304.	307.	297.	257.	263.	264.	264.	264.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.</td																									



THE HOUSE - 100-27

### Surface elevations (cm)

SURFACE ELEVATION (cm)

卷之三







TIME 1.00 sec V-THERM (10 cm<sup>2</sup>/15 min)

TIME IN HOURS 20.00-21 SURFACE ELEVATION (M)







TELE 2.00 MF 3 V-TECH 110 GND/51

1

2

PROGRAM POLTRAC(INPUT,OUTPUT,TAPES,TAPE6=OUTPUT)

C POLLUTANT TRANSPORT MODEL

C

C REFERENCE: DOCUMENTATION OF A TWO-DIMENSIONAL X-Y MODEL PACKAGE  
C FOR COMPUTING LAKE CIRCULATIONS AND POLLUTANT TRANSPORTS  
C BY T. J. SIMONS AND D. C. L. LAM (1982)

C

C C = VARIABLE CONCENTRATION AVERAGED OVER GRID SQUARE (GRAMS/CM\*\*3)  
C U,V = VERTICALLY INTEGRATED WATER TRANSPORT IN X,Y DIRECTION (CM\*\*2/SEC)  
C H = WATER DEPTH AVERAGED OVER A GRID SQUARE (CM)  
C STAGGERED GRID WITH H AND C AT CENTER OF GRIDSQUARE, U AT LEFT HAND SIDE,  
C AND V AT LOWER SIDE HAVING THE SAME INDICES I,J  
C ALL TWO-DIMENSIONAL ARRAYS HAVE DIMENSIONS (IU,JV)  
C IZ,JZ = NUMBER OF GRIDSQUARES IN X,Y DIRECTION  
C IU,JV = NUMBER OF SIDES OF GRIDSQUARES IN X,Y DIRECTION (=IZ+1,JZ+1)  
C DS = GRIDSPACING (CM), DT = TIMESTEP (SEC), NSTEP = NUMBER OF STEPS  
C NC = NUMBER OF VARIABLES, CO = INITIAL VALUES(GRAM/CM\*\*3)  
C NMAP = NO. OF TIME STEPS IN BETWEEN PRINTOUTS  
C DIFF = HORIZONTAL EDDY DIFFUSIVITY (CM\*\*2/SEC)  
C NUMTHD = 1 IF UPSTREAM DIFFERENCING SCHEME IS USED  
C (NUMERICAL DIFFUSION IS ABOUT U\*DS/H;  
C TEMPORALLY STABLE IF U\*DT/DS\*H IS LESS THAN 1  
C AND DIFF\*DT/DS\*\*2 IS LESS THAN 0.5)  
C NUMTHD = 2 IF CENTRAL DIFFERENCING SCHEME IS USED  
C (SPATIALLY STABLE IF U\*DS/H\*DIFF IS LESS THAN 2;  
C TEMPORAL STABILITY CONDITION IS SAME AS NUMTHD=1)

COMMON TLOAD(3),TGAIN(3),TLOSS(3),TWSET(3)  
COMMON /TEMP/ TT(48,35),NUMTHD,DIFF  
DIMENSION H(48,35),U(48,35),V(48,35)  
DIMENSION C(48,35,3),CO(3),CC(48,35,3)

C

C READ IN MODEL PARAMETERS: NSTEP,NMAP,DT,DIFF,NC,CO

C

READ (5,1004) NSTEP,NMAP,DT,NUMTHD,DIFF  
READ (5,1005) NC,(CO(I),I=1,NC)  
WRITE(6,1000)  
WRITE(6,1004) NSTEP,NMAP,DT,NUMTHD,DIFF  
WRITE(6,1005) NC,(CO(I),I=1,NC)

C READ DEPTHS OF GRID SQUARES IN CM

C DEPTH OUTSIDE LAKE MUST BE ZERO EVEN IF GRID CELL LIES WITHIN A RIVER  
C FIRST ROW AND FIRST COLUMN OF DEPTH ARRAY ARE ASSUMED TO BE ZERO  
C LAST ROW AND LAST COLUMN ARE ALWAYS ZERO SINCE DIMENSIONS ARE IU,JV,  
C WHILE NUMBER OF GRIDSQUARES ARE IZ=IU-1, JZ=JV-1

CALL DEPTH(IU,JV,IZ,JZ,DS,H)

CALL FLOWS(IZ,JZ,IU,JV,DS,H,U,V)

DO 10 N=1,NC

DO 10 I=1,IZ

DO 10 J=1,JZ

C(I,J,N)=0.

IF(H(I,J).EQ.0.) GO TO 10

C(I,J,N)=CO(N)

10 CONTINUE

TIME=0.

NMAP=0

```

DO 50 ISTEP=1,NSTEP
TIME=TIME+DT
DO 30 N=1,NC

DO 15 I=1,IZ
DO 15 J=1,JZ
15 TT(I,J)=0.

CALL SOURCE(N,NC,IZ,JZ,DT,DS,H,U,V,C(1,1,N))
CALL ADVECT(IZ,JZ,IU,JV,DT,DS,H,U,V,C(1,1,N))
CALL BIOCHM(N,IZ,JZ,DT,H,C)

DO 20 I=1,IZ
DO 20 J=1,JZ
20 CC(I,J,N)=TT(I,J)

30 CONTINUE

DO 35 N=1,NC
DO 35 I=1,IZ
DO 35 J=1,JZ
C(I,J,N)=C(I,J,N)+CC(I,J,N)
35 CC(I,J,N)=C(I,J,N)*1.E9

```

C PRINT COMPUTED CONCENTRATIONS AND VOLUME INTEGRALS OF POLLUTANTS

```

KMAP=KMAP+1
IF(KMAP.LT.NMAP) GO TO 50
KMAP=0
THRS=TIME/3600.
DO 42 N=1,NC
CSUM=0.
DO 40 I=1,IZ
DO 40 J=1,JZ
40 CSUM=CSUM+H(I,J)*C(I,J,N)
CSUM=CSUM*DS*DS
WRITE(6,1002) THRS,CSUM,TLOAD(N),TGAIN(N),TLOSS(N),TWSET(N)

DO 41 JJ=1,35
J=36-JJ
41 WRITE(6,1003) (CC(I,J,N),I=1,24)
WRITE(6,1000)
DO 42 JJ=1,35
J=36-JJ
42 WRITE(6,1003) (CC(I,J,N),I=25,48)

50 CONTINUE
1000 FORMAT(1H1)
1001 FORMAT(24F3.0)
1002 FORMAT(1H1,F10.3,5E20.6/)
1003 FORMAT(1H0,24F5.1)
1004 FORMAT(2I10,E10.3,I10,E10.3)
1005 FORMAT(I10,6E10.3)
END

```

SUBROUTINE DEPTH(IC,JC,IZ,JZ,DS,H)

C  
C READ DIMENSIONS OF ARRAYS (IC,JC) AND GRIDMESH DS IN CM  
C READ DEPTHS OF GRIDSQUARES IN CM AND STORE IN ARRAY H(I,J)  
C DEPTHS MUST BE ZERO FOR I=1, FOR I=IC, FOR J=1, AND FOR J=JC  
C DEPTH OUTSIDE BASIN MUST BE ZERO EVEN IF GRID CELL LIES WITHIN A RIVER.  
C  
DIMENSION H(48,35)  
C  
C  
READ (5,1001) IC,JC,DS  
WRITE(6,1001) IC,JC,DS  
IZ=IC-1  
JZ=JC-1  
C  
DO 1 I=1,IC  
DO 1 J=1,JC  
1 H(I,J)=0.  
DO 2 I=1,IZ  
2 READ (5,1002) (H(I,J),J=1,JZ)  
C  
WRITE(6,1000)  
DO 4 JJ=1,JC  
J=JC-JJ+1  
4 WRITE(6,1003) (H(I,J),I=1,24)  
WRITE(6,1000)  
DO 5 JJ=1,JC  
J=JC-JJ+1  
5 WRITE(6,1003) (H(I,J),I=25,IC)  
C  
1000 FORMAT(1H1)  
1001 FORMAT(2I5,F10.0)  
1002 FORMAT(24F3.0)  
1003 FORMAT(1H0,24F5.0)  
RETURN  
END

SUBROUTINE FLOWS(IZ,JZ,IU,JV,DS,H,U,V)

C READ VERTICALLY INTEGRATED TRANSPORT COMPONENTS IN CM\*\*2/SEC

DIMENSION H(48,35),U(48,35),V(48,35)

```
DO 1 I=1,IU
DO 1 J=1,JV
U(I,J)=0.
1 V(I,J)=0.
DO 2 I=1,IU
2 READ (5,1002) (U(I,J),J=1,JV)
DO 3 I=1,IU
3 READ (5,1002) (V(I,J),J=1,JV)

UNIT=100.
DO 4 I=1,IU
DO 4 J=1,JV
U(I,J)=U(I,J)/UNIT
4 V(I,J)=V(I,J)/UNIT
IWRITE=0
5 IWRITE=IWRITE+1

WRITE(6,1000)
DO 6 JJ=1,35
J=36-JJ
6 WRITE(6,1003) (U(I,J),I=1,24)
WRITE(6,1000)
DO 7 JJ=1,35
J=36-JJ
7 WRITE(6,1003) (U(I,J),I=25,48)
WRITE(6,1000)
DO 8 JJ=1,35
J=36-JJ
8 WRITE(6,1003) (V(I,J),I=1,24)
WRITE(6,1000)
DO 9 JJ=1,35
J=36-JJ
9 WRITE(6,1003) (V(I,J),I=25,48)
IF(IWRITE.GT.1) GO TO 10

CALL STREAM(IZ,JZ,IU,JV,DS,H,U,V)
GO TO 5
10 CONTINUE
DO 11 I=1,IU
DO 11 J=1,JV
U(I,J)=U(I,J)*UNIT
11 V(I,J)=V(I,J)*UNIT
1000 FORMAT(1H1)
1002 FORMAT(12F6.0)
1003 FORMAT(1H0,24F5.0)
```

RETURN

END

SUBROUTINE STREAM(IZ,JZ,IU,JV,DS,H,U,V)

C THIS ROUTINE FINDS THE NONDIVERGENT PART OF THE VERTICALLY INTEGRATED  
C WATER TRANSPORT BY DETERMINING THE STREAMFUNCTION S AND VORTICITY F.

DIMENSION H(48,35),U(48,35),V(48,35),S(48,35),F(48,35)

```
DO 10 I=1,IU
DO 10 J=1,JV
S(I,J)=0.
10 F(I,J)=0.
```

C F.EQ.0 ON LAND AND ON ANY BOUNDARY CONTOUR

C FIND STREAMPOINTS INSIDE THE LAKE AND SET F TEMPORARILY GREATER THAN ZERO

```
DO 20 I=2,IZ
DO 20 J=2,JZ
IF(H(I,J).EQ.0.) GO TO 20
IF(H(I-1,J).EQ.0.) GO TO 20
IF(H(I,J-1).EQ.0.) GO TO 20
IF(H(I-1,J-1).EQ.0.) GO TO 20
F(I,J)=1.
20 CONTINUE
```

C SPECIFY STREAMFUNCTION ON ALL BOUNDARY CONTOURS

CALL SHORE(IZ,JZ,IU,JV,DS,H,U,V,S,F)

C COMPUTE VORTICITY OF WATER TRANSPORT

```
DO 30 I=2,IZ
DO 30 J=2,JZ
IF(F(I,J).EQ.0.) GO TO 30
F(I,J)=(U(I,J-1)-U(I,J)-V(I-1,J)+V(I,J))*DS
IF(F(I,J).EQ.0.) STOP
30 CONTINUE
```

C FIND STREAMFUNCTION INSIDE THE LAKE

CALL POISON(IU,JV,S,F)

C DETERMINE NONDIVERGENT TRANSPORTS

```
DO 50 I=2,IZ
DO 50 J=2,JZ
IF(H(I-1,J).EQ.0..AND.H(I,J).EQ.0.) GO TO 40
U(I,J)=(S(I,J)-S(I,J+1))/DS
40 IF(H(I,J-1).EQ.0..AND.H(I,J).EQ.0.) GO TO 50
V(I,J)=(S(I+1,J)-S(I,J))/DS
50 CONTINUE
```

C

C

RETURN  
END

SUBROUTINE SOURCE(N,NC,IZ,JZ,DT,DS,H,U,V,T)

C THIS ROUTINE SPECIFIES INPUT OF POLLUTANTS FROM ALL SOURCES INCLUDING  
C INFLOWING RIVERS AND ALL LOSSES DUE TO OUTFLOWING RIVERS OR SETTLING.  
C FOR COMPUTING THE RIVER CONTRIBUTIONS IT IS ASSUMED THAT THE RIVER  
C FLOWS ARE KNOWN FROM THE CIRCULATION MODEL.  
C KLOAD = NUMBER OF GRIDPOINTS WITH LOADING OTHER THAN RIVERS  
C ILOAD,JLOAD = INDICES OF THESE GRIDPOINTS IN TWO-DIMENSIONAL GRID  
C CLOAD(K,N) = LOADING OF VARIABLE N IN POINT K (IN GRAMS/SEC)  
C KRIVER = NUMBER OF GRIDPOINTS ADJACENT TO INFLOWING RIVERS  
C IRIVER,JRIVER = INDICES OF THESE GRIDPOINTS IN TWO-DIMENSIONAL GRID  
C CRIVER(K,N) = RIVER CONCENTRATION OF VARIABLE N ADJACENT TO POINT K  
C (IN GRAMS/CM\*\*3)  
C KLOSS = NUMBER OF GRIDPOINTS ADJACENT TO OUTFLOWING RIVERS  
C ILOSS,JLOSS = INDICES OF THESE GRIDPOINTS IN TWO-DIMENSIONAL GRID  
C WSET(N) = SETTLING VELOCITY OF VARIABLE N (CM/SEC)

COMMON TLOAD(3),TGAIN(3),TLOSS(3),TWSET(3)  
COMMON /TEMP/TT(48,35)  
DIMENSION H(48,35),U(48,35),V(48,35),T(48,35)  
DIMENSION ILOAD(10),JLOAD(10),CLOAD(10,3)  
DIMENSION IRIVER(10),JRIVER(10),CRIVER(10,3)  
DIMENSION ILOSS(10),JLOSS(10)  
DIMENSION WSET(3)  
DATA NCALL/0/

IF(NCALL.GT.0) GO TO 2  
READ (5,1001) KLOAD,KRIVER,KLOSS  
READ (5,1001) (ILOAD(K),JLOAD(K),K=1,KLOAD)  
READ (5,1002) ((CLOAD(K,M),K=1,KLOAD),M=1,NC)  
READ (5,1001) (IRIVER(K),JRIVER(K),K=1,KRIVER)  
READ (5,1003) ((CRIVER(K,M),K=1,KRIVER),M=1,NC)  
READ (5,1001) (ILOSS(K),JLOSS(K),K=1,KLOSS)  
READ (5,1004) (WSET(M),M=1,NC)  
WRITE(6,1001) KLOAD,KRIVER,KLOSS  
WRITE(6,1001) (ILOAD(K),JLOAD(K),K=1,KLOAD)  
WRITE(6,1002) ((CLOAD(K,M),K=1,KLOAD),M=1,NC)  
WRITE(6,1001) (IRIVER(K),JRIVER(K),K=1,KRIVER)  
WRITE(6,1003) ((CRIVER(K,M),K=1,KRIVER),M=1,NC)  
WRITE(6,1001) (ILOSS(K),JLOSS(K),K=1,KLOSS)  
WRITE(6,1004) (WSET(M),M=1,NC)  
DO 1 M=1,NC  
1 TLOAD(M)=0.  
1 TGAIN(M)=0.  
1 TLOSS(M)=0.  
1 TWSET(M)=0.  
2 NCALL=NCALL+1

```

    IF(KLOAD.EQ.0) GO TO 15
    DO 10 K=1,KLOAD
    I=ILOAD(K)
    J=JLOAD(K)
    VOLUM=DS*DS*KH(I,J)
    TT(I,J)=TT(I,J)+DT*CLOAD(K,N)/VOLUM
10   TLOAD(N)=TLOAD(N)+DT*CLOAD(K,N)
15   CONTINUE

    IF(KRIVER.EQ.0) GO TO 25
    DO 20 K=1,KRIVER
    I=IRIVER(K)
    J=JRIVER(K)
    CGAIN=0.
    IF(H(I-1,J).EQ.0.) CGAIN=CGAIN+DS*U(I,J)*CRIVER(K,N)
    IF(H(I+1,J).EQ.0.) CGAIN=CGAIN-DS*U(I+1,J)*CRIVER(K,N)
    IF(H(I,J-1).EQ.0.) CGAIN=CGAIN+DS*V(I,J)*CRIVER(K,N)
    IF(H(I,J+1).EQ.0.) CGAIN=CGAIN-DS*V(I,J+1)*CRIVER(K,N)
    VOLUM=DS*DS*KH(I,J)
    TT(I,J)=TT(I,J)+DT*CGAIN/VOLUM
20   TGAIN(N)=TGAIN(N)+DT*CGAIN
25   CONTINUE
C
    IF(KLOSS.EQ.0) GO TO 35
    DO 30 K=1,KLOSS
    I=ILLOSS(K)
    J=JLOSS(K)
    CLOSS=0.
    IF(H(I+1,J).EQ.0.) CLOSS=CLOSS+DS*U(I+1,J)*T(I,J)
    IF(H(I-1,J).EQ.0.) CLOSS=CLOSS-DS*U(I,J)*T(I,J)
    IF(H(I,J+1).EQ.0.) CLOSS=CLOSS+DS*V(I,J+1)*T(I,J)
    IF(H(I,J-1).EQ.0.) CLOSS=CLOSS-DS*V(I,J)*T(I,J)
    VOLUM=DS*DS*KH(I,J)
    TT(I,J)=TT(I,J)-DT*CLOSS/VOLUM
30   TLOSS(N)=TLOSS(N)+DT*CLOSS
35   CONTINUE
C
    IF(WSET(N).EQ.0.) GO TO 45
    DO 40 I=1,IZ
    DO 40 J=1,JZ
    IF(H(I,J).EQ.0.) GO TO 40
    CWSET=WSET(N)*T(I,J)
    TT(I,J)=TT(I,J)-DT*CWSET/H(I,J)
    TWSET(N)=TWSET(N)+DT*DS*DS*CWSET
40   CONTINUE
45   CONTINUE
C
C
1001 FORMAT(16I5)
1002 FORMAT(8E10.3)
1003 FORMAT(8E10.3)
1004 FORMAT(8F10.5)
    RETURN
    END

```

SUBROUTINE ADVECT(IZ,JZ,IU,JV,DT,DS,H,U,V,T)

C THIS ROUTINE COMPUTES TWO-DIMENSIONAL TRANSPORT BY ADVECTION AND DIFFUSION  
C USING UPSTREAM DIFFERENCES (NUMTHD=1) OR CENTRAL DIFFERENCES (NUMTHD=2)  
C T = VARIABLE AVERAGED OVER A GRID SQUARE  
C H = WATER DEPTH AVERAGED OVER A GRID SQUARE  
C U,V = VERTICALLY INTEGRATED WATER TRANSPORT IN X,Y DIRECTION  
C STAGGERED GRID WITH H AND T AT CENTER OF GRID SQUARE, U AT LEFT HAND SIDE,  
C AND V AT LOWER SIDE HAVING THE SAME INDICES  
C DEPTH H MUST BE ZERO OUTSIDE THE LAKE, INCLUDING RIVER POINTS  
C CONTRIBUTIONS FROM INFLOWING OR OUTFLOWING RIVERS ARE COMPUTED IN  
C ROUTINE SOURCE AND THUS ARE NOT INCLUDED HERE.  
C

DIMENSION H(48,35),U(48,35),V(48,35),T(48,35)  
COMMON /TEMP/ TT(48,35),NUMTHD,DIFF

C DIFF = DIFFUSION COEFFICIENT IN CM\*\*2/SEC

```
DX=DS
DY=DS
A=DIFF/(DX*DX)
B=DIFF/(DY*DY)
```

C TRANSPORTS ACROSS EAST-WEST SIDES OF GRID SQUARES

```
DO 20 I=2,IZ
DO 20 J=2,JZ
IF(H(I,J-1).EQ.0..OR.H(I,J).EQ.0.) GO TO 20
IF(NUMTHD.EQ.2) GO TO 10
IF(V(I,J).GE.0.) VT=V(I,J)*T(I,J-1)/DY
IF(V(I,J).LT.0.) VT=V(I,J)*T(I,J)/DY
GO TO 15
10 VT=V(I,J)*(T(I,J)+T(I,J-1))*0.5/DY
15 CONTINUE
HV=(H(I,J-1)+H(I,J))/2.
VT=VT+B*HV*(T(I,J-1)-T(I,J))
TT(I,J) =TT(I,J)+DT*VT/H(I,J)
TT(I,J-1)=TT(I,J-1)-DT*VT/H(I,J-1)
20 CONTINUE
```

C TRANSPORTS ACROSS NORTH-SOUTH SIDES OF GRID SQUARES

```
DO 40 I=2,IZ
DO 40 J=2,JZ
IF(H(I-1,J).EQ.0..OR.H(I,J).EQ.0.) GO TO 40
IF(NUMTHD.EQ.2) GO TO 30
IF(U(I,J).GE.0.) UT=U(I,J)*T(I-1,J)/DX
IF(U(I,J).LT.0.) UT=U(I,J)*T(I,J)/DX
GO TO 35
30 UT=U(I,J)*(T(I-1,J)+T(I,J))*0.5/DX
35 CONTINUE
HU=(H(I-1,J)+H(I,J))/2.
UT=UT+A*HU*(T(I-1,J)-T(I,J))
TT(I,J) =TT(I,J)+DT*UT/H(I,J)
TT(I-1,J)=TT(I-1,J)-DT*UT/H(I-1,J)
40 CONTINUE
```

```
RETURN
END
```

```
SUBROUTINE BIOCHM(N,IZ,JZ,DT,H,C)
REAL K1
DIMENSION H(48,35),C(48,35,3)
COMMON /TEMP/TT(48,35),NUMTHD,DIFF
DATA NCALL/0/
```

```
C
C AN EXAMPLE OF TWO-COMPARTMENT PHOSPHORUS MODEL:
DC1/DT = -PHI*C1/(C1+CSAT)*C2+K1*1.07**TEMP*C2
DC2/DT = +PHI*C1/(C1+CSAT)*C2-K1*1.07**TEMP*C2
```

```
C WHERE C1 = SOLUBLE REACTIVE PHOSPHORUS (GM/CM**3)
C C2 = PARTICULATE PHOSPHORUS (GM/CM**3)
C RHO = LIGHT FACTOR ( 0 TO 1)
C TEMP = TEMPERATURE (DEGREE C)
C PHI = LIGHT AND TEMPERATURE FACTOR
C K1 = RESPIRATION RATE (PER SEC)
C CSAT = HALF-SATURATED CONSTANT (GM/CM**3)
```

```
C
C NOTE THAT WHILE THESE MODEL PARAMETERS ARE TREATED AS
C CONSTANTS HERE, THEY MAY BE FUNCTIONS OF TIME AND SPACE
C (SEE, E.G., SIMONS AND LAM 1980, WAT. RESOUR. RES., 16, 105-116)
```

```
C
C IF(NCALL.GT.0) GO TO 10
READ (5,1000) RHO,TEMP,K1,CSAT
WRITE(6,1000) RHO,TEMP,K1,CSAT
10 NCALL=NCALL+1
TEX=1.07**TEMP
PHI=5.E-6*RHO*TEX
TEX=K1*TEX
```

```
C
C KINETICS OF C1
```

```
C
C IF(N.EQ.2) GO TO 30
DO 20 I=1,IZ
DO 20 J=1,JZ
IF(H(I,J).EQ.0.) GO TO 20
C1=C(I,J,1)
C2=C(I,J,2)
TT(I,J)=TT(I,J)+DT*(-PHI*C1/(C1+CSAT)+TEX)*C2
20 CONTINUE
RETURN
```

```
C
C KINETICS OF C2
```

```
C
C 30 DO 40 I=1,IZ
DO 40 J=1,JZ
IF(H(I,J).EQ.0.) GO TO 40
C1=C(I,J,1)
C2=C(I,J,2)
TT(I,J)=TT(I,J)+DT*(PHI*C1/(C1+CSAT)-TEX)*C2
40 CONTINUE
```

```
C
C 1000 FORMAT(2F10.1,2E10.3)
RETURN
END
```

```
SUBROUTINE SHORE(IZ,JZ,IU,JV,DS,H,U,V,S,F)
```

```
C THIS ROUTINE SETS THE VALUES OF THE STREAMFUNCTION ON BOUNDARY CONTOURS  
C INCLUDING ISLANDS AND OPEN BOUNDARIES.
```

```
DIMENSION H(48,35),U(48,35),V(48,35),S(48,35),F(48,35)
```

```
C ARRAY F IS USED TEMPORARILY TO DIFFERENTIATE BETWEEN STREAMPOINTS IN THE  
C LAKE (F.GT.0), ON LAND (F.EQ.0) AND ON ANY CONTOUR (F.LT.0)
```

```
DO 10 I=2,IU  
DO 10 J=2,JV  
IF(F(I,J).GT.0.) GO TO 10  
IF(H(I-1,J-1).GT.0.) F(I,J)=-1.  
IF(H(I-1,J).GT.0.) F(I,J)=-1.  
IF(H(I,J-1).GT.0.) F(I,J)=-1.  
IF(H(I,J).GT.0.) F(I,J)=-1.  
10 CONTINUE
```

```
C FIND THE FIRST POINT OF EACH CONTOUR
```

```
NB=0  
20 NB=NB+1  
DO 30 I=1,IU  
DO 30 J=1,JV  
IF(F(I,J).LT.0.) GO TO 40  
30 CONTINUE  
GO TO 90
```

```
C SET STREAMFUNCTION TO ZERO IF THIS IS THE OUTER CONTOUR (NB.EQ.1)  
C DETERMINE STREAMFUNCTION IF THIS IS AN ISLAND CONTOUR (NB.GT.1)
```

```
40 SS=0.  
IF(NB.EQ.1) GO TO 60  
50 I=I-1  
IF(F(I,J).GT.0.) GO TO 50  
SS=S(I,J)  
55 SS=SS+DS*V(I,J)  
I=I+1  
IF(F(I,J).GT.0.) GO TO 55  
60 S(I,J)=SS  
F(I,J)=0.
```

```
C FIND STREAMFUNCTION IN REMAINING POINTS OF THIS CONTOUR
```

```
C SET F=0 IN POINTS WHERE STREAMFUNCTION HAS BEEN DETERMINED
```

```
70 CONTINUE  
WRITE(6,1000) I,J,S(I,J)  
1000 FORMAT(2I5,E12.5)
```

```
II=I+1  
IF(II.GT.IU) GO TO 75  
IF(F(II,J).GE.0.) GO TO 75  
IF(H(I,J-1).EQ.0..AND.H(I,J).EQ.0..) GO TO 75  
IF(H(I,J-1).GT.0..AND.H(I,J).GT.0..) GO TO 75  
S(II,J)=S(I,J)+DS*V(I,J)  
F(II,J)=0.  
I=II  
GO TO 70
```

75 JJ=J+1  
IF(F(I,JJ).GE.0.) GO TO 80  
IF(H(I-1,J).EQ.0..AND.H(I,J).EQ.0.) GO TO 80  
IF(H(I-1,J).GT.0..AND.H(I,J).GT.0.) GO TO 80  
S(I,JJ)=S(I,J)-DS\*U(I,J)  
F(I,JJ)=0.  
J=JJ  
GO TO 70

80 II=I-1  
IF(II.EQ.0) GO TO 85  
IF(F(II,J).GE.0.) GO TO 85  
IF(H(II,J-1).EQ.0..AND.H(II,J).EQ.0.) GO TO 85  
IF(H(II,J-1).GT.0..AND.H(II,J).GT.0.) GO TO 85  
S(II,J)=S(I,J)-DS\*V(II,J)  
F(II,J)=0.  
I=II  
GO TO 70

85 JJ=J-1  
IF(JJ.EQ.0) GO TO 20  
IF(F(I,JJ).GE.0.) GO TO 20  
IF(H(I-1,JJ).EQ.0..AND.H(I,JJ).EQ.0.) GO TO 20  
IF(H(I-1,JJ).GT.0..AND.H(I,JJ).GT.0.) GO TO 20  
S(I,JJ)=S(I,J)+DS\*U(I,JJ)  
F(I,JJ)=0.  
J=JJ  
GO TO 70

90 CONTINUE

C  
C

RETURN  
END

SUBROUTINE POISON(IU,JV,S,F)

C SOLUTION OF POISON EQUATION BY RELAXATION, S = UNKNOWN, F = RIGHT HAND SIDE  
C F MUST BE EQUAL TO ZERO OUTSIDE LAKE, INCLUDING BOUNDARY CONTOURS  
C BUT F CANNOT BE ZERO INSIDE THE LAKE  
C S MUST BE PRESCRIBED ON ALL BOUNDARY CONTOURS  
C EPS = ACCURACY, RO = OVERRELAXATION

DIMENSION S(48,35),F(48,35)

EPS=.001

RO=1.5

DO 10 I=1,IU  
DO 10 J=1,JV  
IF(F(I,J).EQ.0.) GO TO 10  
S(I,J)=F(I,J)  
10 CONTINUE

DO 40 ITER=1,200  
SMAX=0.  
DSMAX=0.

DO 30 N=1,2  
DO 30 I=1,IU  
DO 30 J=1,JV  
IF(F(I,J).EQ.0.) GO TO 30  
IJN=I+J+N  
IF(MOD(IJN,2)) 30,30,20  
20 RES=S(I-1,J)+S(I+1,J)+S(I,J-1)+S(I,J+1)-4.\*S(I,J)-F(I,J)  
DS=RES\*RO/4.  
S(I,J)=S(I,J)+DS

SS=S(I,J)\*\*2  
IF(SS.GT.SMAX) SMAX=SS  
DSS=DS\*DS  
IF(DSS.GT.DSMAX) DSMAX=DSS

30 CONTINUE

DS=SQRT(DSMAX/SMAX)  
IF(DS.LT.EPS) GO TO 50  
40 CONTINUE

50 WRITE(6,1000) ITER,DS  
1000 FORMAT(I5,E12.5)  
RETURN  
END

1200 600 300E+02 i 100E+05

-07 • 500E-08

2 161.

48 35



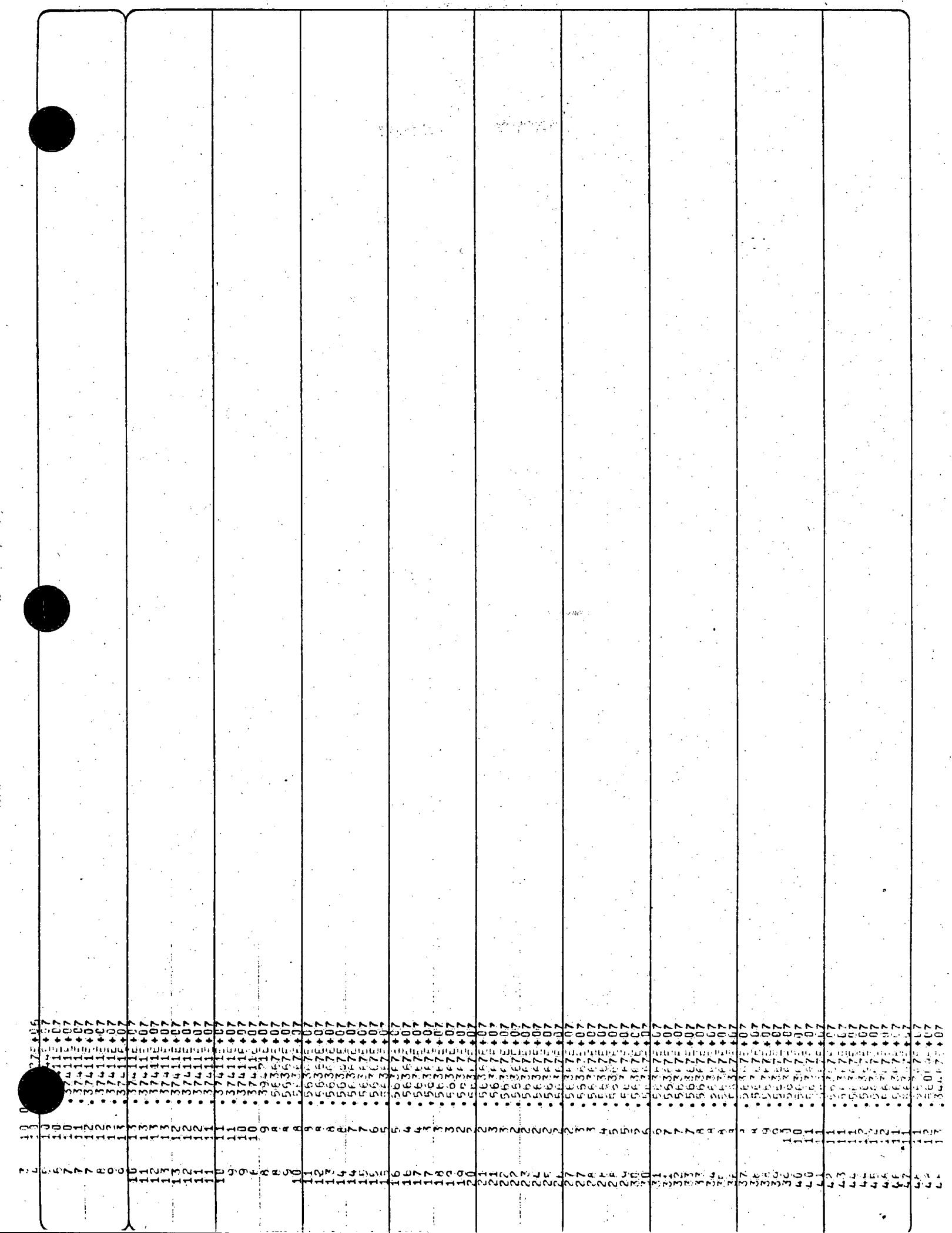


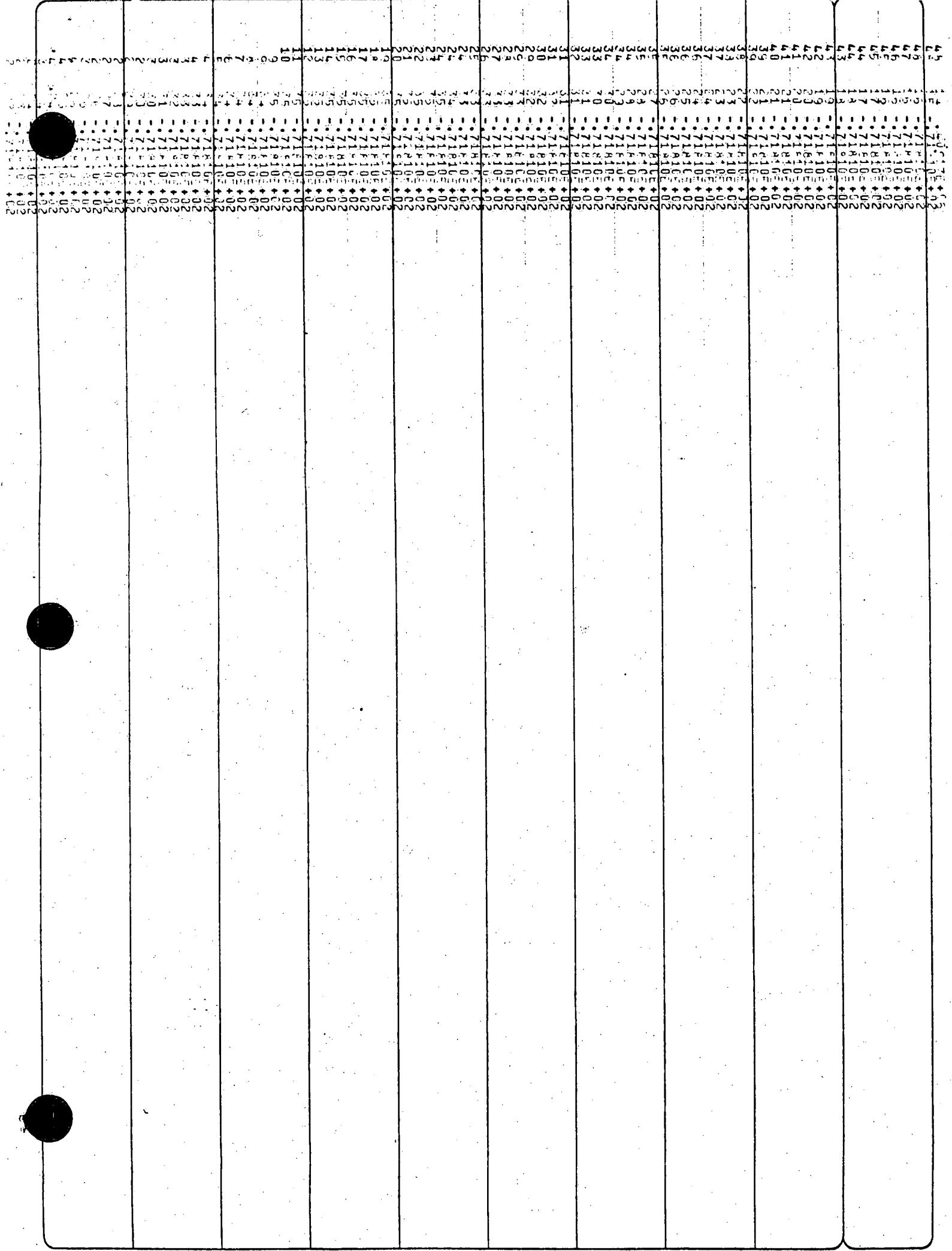






















۷

-134886E+06

•360000=403

• 101.23034

2



5.000 . . . . .  
• 676158E+05 . 360000F+03 . 507305E+05

• 507366E+05      • 105564E+04



10.00

•72000E•03

•20222E+06      •202271E+06

1



• 676305E+05

• 181461E+06 • 720300E+03 •

• 1013614804



10066