

ISWS/CIR-157/83

Circular 157

STATE OF ILLINOIS

DEPARTMENT OF ENERGY AND NATURAL RESOURCES



**GRAFPAC:
FDTOPC, GRID, CONVRT, CONTUR, SURFAC, PRTGRF, and SNKGRF Postprocessors
for Illinois State Water Survey Numerical Groundwater Models**

by Thomas G. Naymik, Bruce C. Mather,
and Matthew E. Chapman

ILLINOIS STATE WATER SURVEY
CHAMPAIGN

1983

CONTENTS

	Page
INTRODUCTION.....	1
PROCEDURE FILES.....	2
PLOTTING DEVICES.....	4
GRID DATA FILES.....	5
Program FDTOPC.....	7
Program GRID.....	9
FLOW OF DATA.....	14
Program CONVRT.....	16
Program CONTUR.....	22
Program SURFAC.....	31
Program PRTGRF.....	39
Program SNKGRF.....	49
REFERENCES.....	55

Funds derived from grants and contracts administered by the
University of Illinois were used to produce this report.

Printed by authority of the State of Illinois

INTRODUCTION

The Illinois State Water Survey has two publications which present groundwater and solute transport codes: Bulletin 55 (Prickett and Lonquist, 1971) and Bulletin 65 (Prickett, Naymik and Lonquist, 1981). These codes are used extensively by a variety of researchers. Graphical display of results has been recognized as a common need among users. This report was written to assist ISWS code users to more effective uses of graphical display. However, not all codes in this report can be used at all computer installations. Some codes favor larger systems where the NCAR (National Center for Atmospheric Research) graphics package is available. Still, portions of this information including the basic concepts should be useful to ISWS code users.

The function of each support code is presented as follows:

- FDTOPC** Routine to create a particle cell grid file from a finite difference grid file with the same data format.
- GRID** Plots finite difference and particle cell grids from delta X and delta Y values.
- CONVRT** A program for extracting nodal information from input files for the purposes of plotting contour maps or three-dimensional perspectives.
- CONTUR** An NCAR driven program for contour plots of model input or calculated water levels. Useful for checking the values of input parameters.
- SURFAC** An NCAR driven program for 3-D surface plots of model input or calculated water levels. Same uses as CONTUR but different form of plotting.
- PRTGRF** Plots particle locations calculated by the random-walk code. An extra subroutine (and corresponding CALL) are added to the random-walk code to produce the particle location file.
- SNKGRF** Produces breakthrough curves of solute concentration versus time at sinks which have been designated by the solute transport model or from raw field data.

PROCEDURE FILES

All of the programs in this package use PROCEDURAL or PROC files. Some systems call these COMMAND files. The PROC files in this package are specific to the NOS operating system for the CDC 6000 series computer. Details about each PROC command may be found in the NOS Version 1 (book 1 of 2) reference manual. A brief list of commands used follows:

Command	Explanation
.PROC,name	Name of called procedure. (Usually name of file)
NOTE,	CREATE a local file and place arguments in it. Always used here to hold DEVICE name for plotting.
IFE,	IF EQUAL, does a test and completes command if result equals zero or false.
ENDIF,	End of IFE statement, IF not equal, skip to here.
ADDLIB,	Used to attach NCAR graphics package to user program.
FTN,	COMPILE FORTRAN: Same as timesharing command.
GET,	GET a FILE: Same as timesharing command.
R.	REWIND local files: Same as timesharing command.
LOAD,	LOAD a FILE: Used for loading a library file (CONRECB).
GRAB,	GET a SYSTEM file: GCS, NCAR, ZETAVU
USE,	CHECK if file on disk: If not, reload and tell user to try again later.
RETURN,	RETURN a file: not REPLACE, clear FILE from local area.
REVERT.	Signal END of PROC file execution
.DATA,<file>	Make the following text into a local file named <file>.
EXIT.	If error occurs during PROC file execution, go to this point and continue execution.

PROCEDURE files are not necessary to the operation of the program, but they allow the user to set up routines without tedious and repetitious entry of commands. An example is FORTRAN compilation. After GETting the file, the source code must be compiled. After the first compilation, the binary load file is local and recompilation of the source is unnecessary. This significantly reduces cost. GRABbing the system routines GCS (Graphics Compatibility System) or NCAR are done with PROC files as well. Note that the PROC files compile or GRAB

programs once in each timesharing session. When the user signs on for another session, the PROCs are rerun.

For all programs in this package, the FORTRAN program described is contained in a PROC file of the same name, with a "P" appended to the file name, i.e., program GRID is contained in file GRIDP. It is assumed that the PROC file is local and that all GRID and DATA files are local before execution (see NOS handbook for the meaning of "local").

PLOTTING DEVICES

All graphic output for the programs in this package are device independent. The program may be run without changes, and the graphic output displayed at a standard terminal, a Tektronix graphics terminal, or a line printer, or plotted on a pen plotter such as the ZETA plotter. This can be accomplished two ways: using GCS (Graphics Compatibility System), or the NCAR (National Center for Atmospheric Research) graphics package. For programs interfaced with GCS, the device independent graphics data is sent to a GCS "device specific" routine which is loaded at execution time. This routine transforms the graphical output to a form suitable for the device specified. Currently some of the supported devices available to the ISWS are:

<device>	USE
ALPH	Interactive graphics on an alphanumeric terminal
TEKT	Interactive graphics on a Tektronix terminal (all models except the 4027 color terminal)
ZETA	Interactive graphics from any terminal. Produces plot file "TAPE99" which can be plotted using the system command: PLOT.
TK27	Color Tektronix terminal (not recommended)
ADDR	Cursor addressable terminal (not recommended)
PRNT	Line printer (not recommended)

For programs using NCAR, the graphical output is written to a plot file "NCARMC" (NCARM metacode), which is translated and plotted (either on a terminal or pen plotter) using an interpreter program (NCARTRN). Check NCAR documentation for details on running the translator and the various options available to the user.

GRID DATA FILES

Most of the post-processing programs presented here require information which describes the finite difference and particle cell grids. The finite difference grid defines the coordinates where raw data are entered and calculated head values are plotted by the routines CONTUR and SURFAC. The particle cell grid is used when computing the movement of particles. It can be derived from the finite difference grid by bisecting the distance between adjoining nodes. Once a finite difference file is set up, the particle cell file can be created with the program FDTOPC (described in the next section).

The grid for all routines can have variable spacings, to allow coverage of large areas and yet allow data to be concentrated in areas of interest. The maximum number of rows or columns for the grid is 100. The format for the grid files follows:

CARD	VARIABLES	FORMAT
1	NC NR SCB	(I10,I10,F10.0)
2 to n	SPI(i) (10 values per card)	(10F8.0)
n+1 to m	SPJ(j) (10 values per card)	(10F8.0)

where:

NC = the number of columns in the model
 NR = the number of rows in the model
 SCB = the "base" grid spacing (a scaling factor used in programs CONTUR and SURFAC). SCB must not equal 0.

SPI : (n-1) cards contain the column spacing information
 There are NC-1 spacing values on these cards
 SPJ : (m-n-1) cards contain the row spacing information
 There are NR-1 spacing values on these cards

An example of a finite difference grid file is shown below. This file was used to produce Figure 1.

	24	16	1.0							
200.0	200.0	175.0	125.0	100.0	90.0	65.0	50.0	50.0	50.0	
35.0	15.0	25.0	40.0	60.0	80.0	90.0	100.0	100.0	125.0	
175.0	200.0	400.0								
100.0	100.0	100.0	100.0	75.0	50.0	50.0	37.5	25.0	25.0	
25.0	37.5	75.0	100.0	400.0						

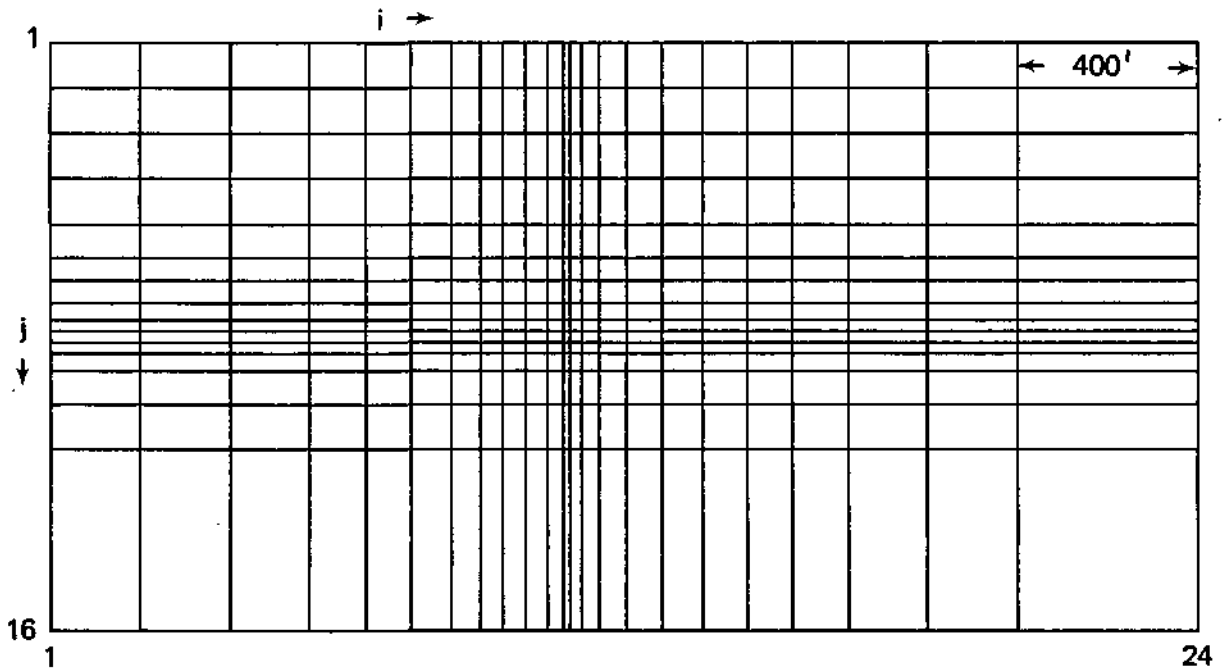


Figure 1. Finite difference grid corresponding to sample data file.

Program FDTOPC

This routine conveniently converts a finite difference (FD) file to the corresponding particle cell (PC) grid. It increases NR and NC by one and computes the nodal spacings for each column and row. All output conforms to each of the program formats in this package. The example PC file below was produced from the sample FD grid file (Figure 1) using program FDTOPC.

	25	17	1.0							
200.0	200.0	187.5	150.0	112.5	95.0	77.5	57.5	50.0	50.0	
42.5	25.0	20.0	32.5	50.0	70.0	85.0	95.0	100.0	112.5	
150.0	187.5	300.0	400.0							
100.0	100.0	100.0	100.0	87.5	62.5	50.0	43.8	31.3	25.0	
25.0	31.3	56.3	87.5	250.0	400.0					

To use FDTOPC type:

```
/FDTOPCP,FD=<fdgrdfile>,PC=<pcgrdfile>
```

where <fdgrdfile> is the finite difference grid file (input)

<pcgrdfile> is the particle cell file (output)

An example of running FDTOPC with finite difference file SRFDG (sample input) to produce the particle cell grid file SRPCG is shown below:

```
/FDTOPCP,FD=SRFDG,PC=SRPCG
```

```
REVERT. SRFDG CONVERTED TO SRPCG
```

```

.PROC,FDTOPC,FD=FDGRID,PC=PCGRID.
FTN,I=TEMP,L=0,A,B=FDBIN.
FDBIN,FD,PC.
REVERT. CONVERT FD TO PC
.DATA,TEMP

      PROGRAM FDTOPC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      REAL SPI(100),SPJ(100),OUTI(100),OUTJ(100)
C
C INPUT NUMBERS OF ROWS AND COLUMNS IN MODEL, AND BASE DISTANCE
C
      READ (5,*) NC,NR,SCB
C
C INPUT VARIABLE GRID SPACINGS (FREE FORMAT)
C
      M=NC-1
      N=NR-1
      READ (5,*) (SPI(I),I=1,M)
      READ (5,*) (SPJ(NR-I),I=1,N)
C
C COMPUTE AND WRITE PARTICLE CELL GRID
C
      WRITE (6,910) NC+1,NR+1,SCB
      OUTI(1)=SPI(1)
      DO 20 I=2,M
        OUTI(I)=(SPI(I-1)+SPI(I))/2
20  CONTINUE
      OUTI(NC)=SPI(NC-1)
      WRITE (6,920) (OUTI(I),I=1,NC)
      OUTJ(1)=SPJ(1)
      DO 30 I=2,N
        OUTJ(I)=(SPJ(I-1)+SPJ(I))/2
30  CONTINUE
      OUTJ(NR)=SPJ(NR-1)
      WRITE (6,920) (OUTJ(NR+1-I),I=1,NR)
      STOP
C
910 FORMAT (I10,I10,F10.2)
920 FORMAT (10F8.1)
      END

```

Figure 2. Listing of program FDTOPC.

Program GRID

Program GRID is used to draw finite difference grids and particle cell grids. It has been discussed partially in the previous two sections. The grids can have variable nodal spacings as described in the GRID DATA FILE section. Program GRID is contained in the procedure file GRIDP and is run interactively:

```
/GRIDP,GRDFIL=<grdfile>,PLOT=<device>
```

where <grdfile> is the FD or PC grid file (see GRID DATA FILES)

<device> specifies the plotting device (see PLOTTING DEVICES)
If PLOT is not specified, the default is PLOT=ZETA.

When using GRIDP to display the grid at the terminal (PLOT=TEKT, or PLOT=ALPH) the size of the plot cannot be controlled. If PLOT=ZETA, however, the user is prompted for size information. If the specified size, which corresponds to the total width of a row, creates a plot height (total length of one column) larger than the width of a small plotter (11 inches), the user is told that the wide plotter must be used. If the height of the plot is larger than the wide plotter (34 inches) then an error message is printed and no plot is produced. In either case, the plot is centered and the height is appropriately scaled. The program listing for GRID is shown in Figure 3.

Examples of GRIDP:

EXAMPLE 1:

```
/GET,GRIDP
/GET,PCGRID
/GRIDP,GRDFIL=PCGRID,PLOT=ZETA
  ENTER THE DESIRED WIDTH OF GRID, IN INCHES.
? 6.0
REVERT.  GRIDP,PCGRID
/
```

At this point the plot file is in TAPE99.

EXAMPLE 2:

```
/GRIDP,GRDFIL=PCGRID                ;Default to PLOT=ZETA
  ENTER THE DESIRED WIDTH OF GRID, IN INCHES.
? 13.5
  WIDE PLOTTER REQUIRED FOR THIS PLOT.
REVERT.  GRIDP,PCGRID
/
```

EXAMPLE 3:

```
/GRIDP,GRDFIL=PCGRID
  ENTER THE DESIRED WIDTH OF GRID, IN INCHES.
? 45
  OUTPUT TOO BIG FOR ANY PLOTTER ~ NO OUTPUT
REVERT.  GRIDP,PCGRID
/
```

EXAMPLE 4:

```
/GRIDP,GRDFIL=FDGRID,PLOT=TEKT                ;Local plot on terminal
  (plot is on terminal at this point)
  (user can copy with a VERSATEC now. When done, type any character)
REVERT.  GRIDP, FDGRID
/
```

An example of an FDGRID and the resulting plot (Figure 1) are shown in the section GRID DATA FILES.

```

.PROC,GRIDP,GRDFIL=GRIDX,PLOT=ZETA.
NOTE(ZZZDEV)/PLOT
IFE,.NOT.FILE(GCSBASE,PM),A1.
GRAB,GCS PLOT/F.
ENDIF(A1).
IFE,.NOT.FILE(GRIDB,LO),A2.
R.
FTN,I=TEMP,B=GRIDB,L=0,A.
ENDIF(A2).
R.
GRIDB,,GRDFIL.
RETURN,ZZZDEV,TEMP.
REVERT. GRIDP,GRDFIL
.DATA,TEMP

      PROGRAM GRID(INPUT,OUTPUT,DATA,ZZZDEV,TAPE5=INPUT,TAPE6=OUTPUT,
1          TAPE7=DATA,TAPE8=ZZZDEV)
      REAL SPI(100),SPJ(100)
      DATA XDIS/0.0/,YDIS/0.0/,HMAX/11.0/
C
C  READ DEVICE TYPE (TEKT, ALPH, ZETA, ETC.)
C
C      READ (8,910) DEV
C
C  INPUT NUMBERS OF ROWS AND COLUMNS IN MODEL, AND BASE DISTANCE
C
C      READ (7,*) NC,NR,SCB
C
C  INPUT VARIABLE GRID SPACINGS
C
C      M=NC-1
C      N=NR-1
C      READ (7,*) (SPI(I),I=1,M)
C      READ (7,*) (SPJ(NR-I),I=1,N)
C
C  CALCULATE MAXIMUM HORIZONTAL AND VERTICAL DISTANCES
C
C      DO 20 I=1,M
C          XDIS=XDIS+SPI(I)
20  CONTINUE
C      DO 30 I=1,N
C          YDIS=YDIS+SPJ(I)
30  CONTINUE
C      IF (DEV.NE."Z") GO TO 90

```

Figure 3. Listing of program GRID.

```

C
C   IF ZETA PLOTTER, INPUT SIZE AND DIMENSIONS OF PLOT.
C
      WRITE (6,920)
      READ (5,*) W
      H=W*(YDIS/XDIS)
      IF (H.LE.10.5) GO TO 90
      IF (H.LE.33.5) GO TO 80
      WRITE (5,930)
      STOP
80  WRITE (5,940)
      HMAX=34.0

C
C   INITIALIZE GRAPHING ROUTINES
C
90  CALL USTART
      WIND=AMAX1(XDIS,YDIS)
      CALL UWINDO (0.,WIND,0.,WIND)
      IF (DEV.NE."Z") GO TO 100

C
C   SET UP PLOTTING AREA.
C
      CALL UDIMEN (W+5,HMAX)
      CALL UDAREA (.5,W+0.5,(HMAX-H)/2,(HMAX+H)/2)
      CALL UWINDO (0.,XDIS,0.,YDIS)

C
C   PLOT THE GRID
C
100 CALL UMOVE(0.0,0.0)
      CALL UPEN(XDIS,0.0)
      CALL UPEN(XDIS,YDIS)
      CALL UPEN(0.0,YDIS)
      CALL UPEN(0.0,0.0)
      XTEMP=0
      M2=NC-2
      DO 110 K=1,M2
          XTEMP=XTEMP+SPI(K)
          CALL UMOVE(XTEMP,0.0)
          CALL UPEN(XTEMP,YDIS)
110  CONTINUE

C
      YTEMP=0
      N2=NR-2
      DO 120 K=1,N2
          YTEMP=YTEMP+SPJ(K)
          CALL UMOVE(0.0,YTEMP)
          CALL UPEN(XDIS,YTEMP)
120  CONTINUE

```

Figure 3. Continued.


```
C
C  WRAP-UP GRAPHING
C
      CALL UFLUSH
      IF (DEV.EQ."Z") GO TO 130
      CALL UBELL
      CALL UPAUSE
130  CALL UERASE
      CALL UEND
      STOP
C
C  FORMAT STATEMENTS
C
910  FORMAT(A1)
920  FORMAT(" ENTER THE DESIRED WIDTH OF GRID, IN INCHES.")
930  FORMAT(" OUTPUT TOO BIG FOR ANY PLOTTER - NO OUTPUT")
940  FORMAT(" WIDE PLOTTER NEEDED FOR THIS GRID")
      END
```

Figure 3. Concluded.

FLOW OF DATA

The data files for the remaining plotting routines come from several sources. Figure 4 shows graphically where data comes from and where it is routed for a specific routine. The programs are depicted inside the boxes and the files are illustrated as input and output. TRANS is the program from Bulletin 65 (see program statement at the beginning of random-walk code). File <trin> is an example input file to TRANS (angle brackets indicate user chosen file NAMES). Files in bold face indicate the actual file names produced by TRANS.

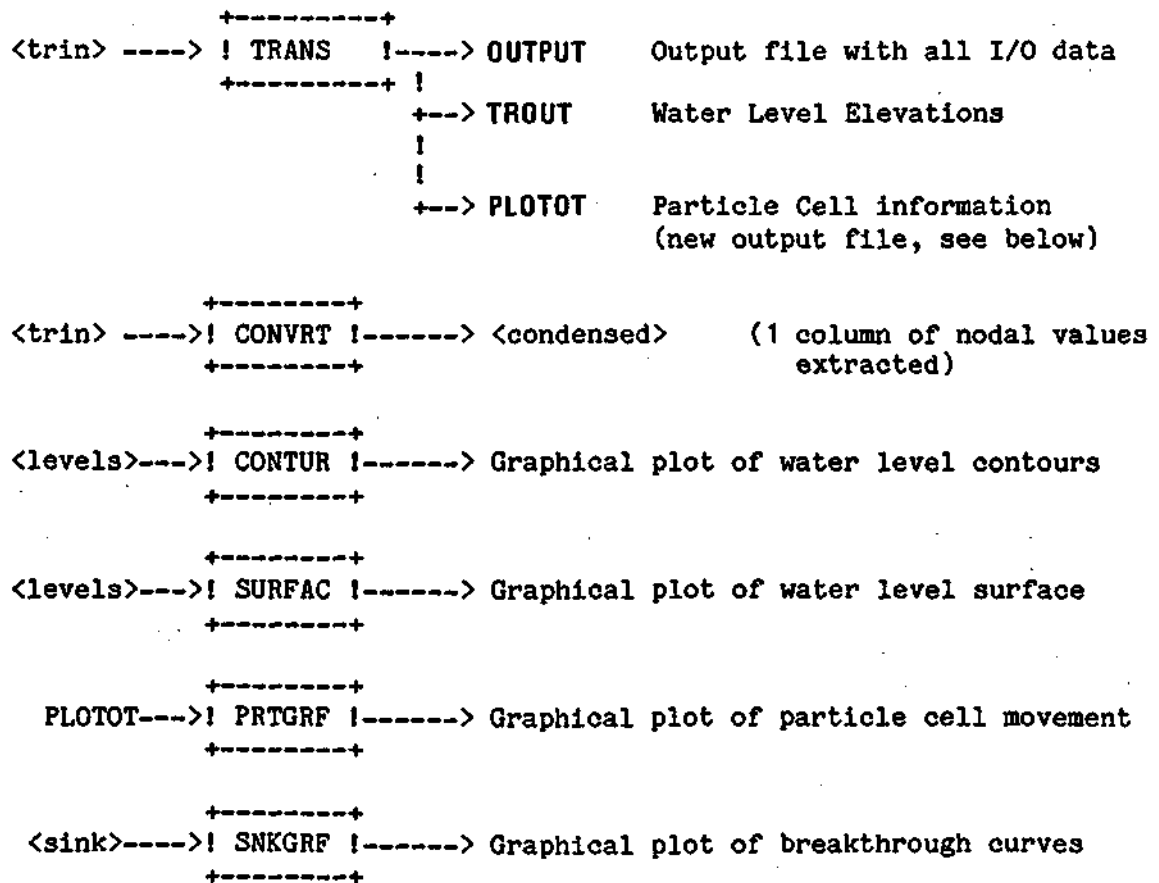


Figure 4. Flow of Data.

Program CONVRT

Program CONVRT is used to convert nodal parameters from the input file for the ISWS program TRANS to a format required by ISWS programs CONTUR and SURFAC. Node card data in the TRANS input file is formatted as one card per node point in the finite difference grid. CONVRT takes this input file and compresses the data to 10 values per record in a format suitable for CONTUR and SURFAC. The user has the option to select which column of data is to be converted. It is assumed that the input file for CONVRT has the same data setup as required in the ISWS Bulletin 65 program TRANS. Users of programs in Bulletin 55 should remove the FORTRAN statements in CONVRT which READ in the additional B65 data, i.e., variable grid spacings, pollution cards, etc. The program listing for CONVRT is shown in Figure 6.

To run CONVRT interactively:

```
/CONVRTP,IN=<trin>,OUT=<outfil>
```

where <trin> is the input data file to TRANS

<outfil> is the file to which the compressed data is written

The format of <trin> is described in ISWS Bulletin 65.

The format of <outfil> is described as follows:

CARD	VARIABLES	FORMAT	
1	FLO,HI,FINC,NHI,ILAB,NULBLL	(3F10.0,3I10)	(see CONTUR)
2	SWITCH,C1,C2,R1,R2	(I2,4I10)	(see SURFAC)
3 to n	<data(i,j)>	(1X,10E10.4)	

The data array is rearranged into blocks of 10 columns by NR rows until all columns are output. If Z(i,j) specifies row i, column j, the data become:

```

Z(1,1)  Z(1,2)  .....  Z(1,10)
Z(2,1)  Z(2,2)  .....  Z(2,10)
Z(NR,1) Z(NR,2) ..... Z(NR,10)
Z(1,11) Z(1,12) ..... Z(1,20)
Z(NR,11) Z(NR,12) ..... Z(NR,20)
.      .      .
Z(1,x)  Z(1,x+1) ..... Z(1,NC)
Z(NR,x)  Z(NR,x+1) ..... Z(NR,NC)

```

An example of using CONVRT to extract the node data corresponding to the BOTTOM of the aquifer from a sample TRANS input file TRINX and putting the compressed form in file BOTTX is shown below:

```
/CONVRTP,IN=TRINX,OUT=BOTTX
```

```
DATA SELECTION AND CONVERSION
```

- 1 T1
- 2 T2
- 3 SF1
- 4 H
- 5 Q
- 6 R
- 7 RH
- 8 RD
- 9 SF2
- 10 CH
- 11 PERM1
- 12 PERM2
- 13 BOTT

```
ENTER DESIRED COLUMN NUMBER
```

```
? 13
```

```
REVERT. TRINX CONVERTED TO BOTTX
```

Note: The input file TRINX is not modified by program CONVRT.

```
.PROC,CONVRTP,IN=TRIN,OUT=XXX.
R.
FTN,I=TEMP,L=0,B=CONVRTB.
CONVRTB,,,IN,OUT.
RETURN,TEMP.
REVERT. CONVERT IN TO OUT
.DATA,TEMP
```

```
PROGRAM CONVRT(INPUT,OUTPUT,TRIN,NEWFIL,
1          TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=TRIN,TAPE8=NEWFIL)
C*****
C THIS PROGRAM IS USED TO CONVERT INPUT DATA FROM THE. *
C TRANS INPUT FILE "TRIN" TO A FORMAT USED BY "SURFAC" AND *
C "CONTUR", THE SURFACE AND CONTOUR PLOTTING PROGRAMS. *
C THE DATA IS REARRANGED INTO ROWS, WITH TEN NUMBERS ON A LINE. *
C THE DATA IS ARRANGED HORIZONTALLY BY COLUMNS AND VERTICALLY *
C BY ROWS. *
C *
C NODES IS THE NODE CARD DATA ARRAY THAT HOLDS ALL OF THE INFO IN *
C TRIN. IT CAN BE MANIPULATED TO DUMP A SPECIFIC ROW INTO A FILE *
C *
C 1 - T1 *
C 2 - T2 *
C 3 - SF1 *
C 4 - H *
C 5 - Q *
C 6 - R *
C 7 - RH *
C 8 - RD *
C 9 - SF2 *
C 10 - CH *
C 11 - PERM1 *
C 12 - PERM2 *
C 13 - BOTT *
C *
C*****
REAL NODES(50,50),ANODE(13)
C
C QUERY USER FOR NODE DATA DESIRED
C
20 WRITE (6,1080)
   READ (5,*) ICOL
   IF (ICOL.EQ.0) STOP
   IF ((ICOL.LT.1).OR.(ICOL.GT.13)) GO TO 20
C
C READ AND DISCARD PARMETER CARD
C READ NC,NR AND DISCARD REST OF DEFAULT VALUE CARD
C
   READ (7,940) (IJUNK,I=1,4),NC,NR,(IJUNK,I=1,11)
```

Figure 6. Listing of program CONVRT.

```

C
C READ IN PUMP PARAMETER CARD (DISCARD NSP)
C
C   READ (7,950) NPUMP,IJUNK,NRT
C
C READ AND DISCARD PUMPING SCHEDULE CARDS
C
C   DO 30 I=1, NPUMP
C     READ (7,960) IJUNK,IJUNK,(RJUNK,J=1,NRT)
30  CONTINUE
C
C READ AND DISCARD DELX AND DELY VARIABLE GRID SPACINGS
C
C   READ (7,970) (RJUNK,II=1,NC)
C   READ (7,970) (RJUNK,JJ=1,NR)
C
C READ AND DISCARD X1,DX,Y1,DY,DELP
C
C   READ (7,980) (RJUNK,I=1,5)
C
C READ AND DISCARD POLLUTION PARAMETER CARD
C
C   READ (7,990) RJUNK,IJUNK,(RJUNK,I=1,8)
C
C READ AND DISCARD SINK LOCATION CARDS
C
C   DO 40 II=1,100
C     READ (7,1000) I,J,IJUNK
C     IF ((I.EQ.0).OR.(J.EQ.0)) GO TO 50
40  CONTINUE
C
C READ AND DISCARD SOURCE CONCENTRATION CARDS
C
C   50 DO 60 II=1,100
C     READ (7,1010) I,J,IJUNK
C     IF ((I.EQ.0).OR.(J.EQ.0)) GO TO 70
60  CONTINUE
C
C   FINALLY READ IN DATA THAT WE WILL USE IN DATA REDUCTION
C   NODE ARRAY NEED ONLY BE 2-D SINCE ONLY 1 PARAMETER IS STORED
C
C   70 READ (7,*)
C     DO 80 II=1,2000
C       IF (EOF(7).NE.0) GO TO 90
C       READ (7,910) I,J,(ANODE(K),K=1,13)
C
C DO SOME SIMPLE DATA VERIFICATION TO SEE IF NODE INFORMATION
C IS GEOMETRICALLY POSSIBLE (NO BOTT'S ABOVE LAND SURFACE!)
C THIS IS NOT NECESSARY, BUT HELPS IN DATA VERIFICATION.
C

```

Figure 6. Continued.

```

      IF (ANODE(13).GT.ANODE(8)) WRITE (6,1030) I,J
      IF (ANODE(8).GT.ANODE(10)) WRITE (6,1040) I,J
      IF (ANODE(10).GT.ANODE(7)) WRITE (6,1050) I,J
      IF (ANODE(13).GT.ANODE(4)) WRITE (6,1060) I,J
      IF (ANODE(4).GT.ANODE(10)) WRITE (6,1070) I,J
      NODES(I,J)=ANODE(ICOL)
80    CONTINUE
C
C    OUTPUT SOME INFORMATION FOR PROGRAM CONTUR
C
C    90 WRITE (8,920) 0,0,5,-1,1,2
C
C    OUTPUT SOME INFORMATION FOR PROGRAM SURFAC
C
C    WRITE (8,930) 0,1,NC,1,NR
C
C    OUTPUT COMPRESSED NODE INFORMATION
C    COLUMNS 1 TO 10 ALL ROWS
C    COLUMNS 11 TO 20 ALL ROWS
C    COLUMNS 21 TO 30 ALL ROWS
C    "      31 TO 40 " "
C    UNTIL ALL COLUMNS ARE OUTPUT
C
      M=NC/10
      IF (MOD(M,10).GT.0) M=M+1
      DO 110 II=1,M
        IJ=(II-1)*10
        IIJ=MINO(10,NC-IJ)
        DO 100 JJ=1,NR
          WRITE (8,1020) (NODES(K+IJ,JJ),K=1,IIJ)
100    CONTINUE
110    CONTINUE
      STOP
C
C
910  FORMAT (2I3,2F6.0,2F4.0,9F6.0)
920  FORMAT (3F10.0,3I10)
930  FORMAT (1X,I2,4I10)
940  FORMAT (I6,2F6.0,I6/2I6,11F6.0)
950  FORMAT (3I6)
960  FORMAT (2I3,12F6.0)
970  FORMAT (10F8.0)
980  FORMAT (5F10.0)
990  FORMAT (F7.1,I7,8F7.1)
1000 FORMAT (3I3)
1010 FORMAT (2I3,1F14.4)
1020 FORMAT (1X,10E10.4)
1030 FORMAT (" ERROR: BOTT > RD AT (" ,I3," ,",I3,")")

```

Figure 6. Continued.


```

1040 FORMAT (" ERROR: RD  > CH AT (" ,I3," ,",I3,")")
1050 FORMAT (" ERROR: CH  > RH AT (" ,I3," ,",I3,")")
1060 FORMAT (" ERROR: BOTT > HO AT (" ,I3," ,",I3,")")
1070 FORMAT (" ERROR: HO  > CH AT (" ,I3," ,",I3,")")
1080 FORMAT (/" DATA SELECTION AND CONVERSION ",
1      /" 1 T1"
2      /" 2 T2"
3      /" 3 SF1"
4      /" 4 H"
5      /" 5 Q"
6      /" 6 R"
7      /" 7 RH"
8      /" 8 RD"
9      /" 9 SF2"
$      /" 10 CH"
$      /" 11 PERM1"
$      /" 12 PERM2"
$      /" 13 BOTT"//" ENTER DESIRED COLUMN NUMBER")
END

```

Figure 6. Concluded.

Program CONTUR

Program CONTUR is used to draw contour plots from data with uniform or variable spaced grids. The input data to CONTUR is either the output file "TROUT" from the ISWS program TRANS or from one of the columns of data in the input file to TRANS (see FLOW OF DATA, CONVRT). CONTUR uses a modified version of the NCAR plotting subroutine CONREC. Most plotting parameters are contained in the contour data file; however, the user is queried during program execution for some additional information. CONTUR is run interactively:

```
/CONTURP,CONDAT=<levels>,FDGRID=<fdgrid>
```

where <fdgrid> is the finite difference grid (see GRID DATA FILES)

<levels> is the data values to be contoured (see FLOW OF DATA)

The file <levels> which is either the output of TRANS or CONVRT has the following format:

CARD	VARIABLES	FORMAT
1	FLO,HI,FINC,NHI,ILAB,NULBLL	(3F10.0,3I10)
2	DUMMY	(used in SURFAC)
3	<contour data>	(10E10.4)

where: FLO = Value of the lowest contour level.
 HI = Value of the highest countour level.

If FLO=HI=0.0, FLO = minimum(data array)
 HI = maximum(data array)

FINC > 0 Increment between contour levels
 = 0 CONREC produces FINC to create 10 to 30 contour levels.

NHI = 0 Highs and Lows marked H and L with value plotted under H/L.
 > 0 The value of each contour datum is plotted.
 < 0 Neither above occurs

ILAB = 0 Label lines with contour values
= 0 Do not label lines

NULBUL = Number of unlabeled lines BETWEEN labeled lines.

CONTUR uses a specially modified version of the NCAR subroutine CONREC to account for the variable grid spacing of the contour data. This modified subroutine is in the file CONRECS, and consists of replacing two statement functions FX(X,Y) and FY(X,Y), by calls to function subprograms (of the same name). There also is an interactive query added (to CONREC) pertaining to the inclusion of a plot perimeter. Figure 7 shows the modifications to CONREC, and Figure 8 shows the functions added to the end of CONREC. CONRECS is compiled and the resulting binary file is saved in CONRECB. After GRABbing the NCAR package, it is necessary to force the loader to use this modified CONRECB instead of the NCAR version. This is done in PROC file CONTURP with the LOAD command. All other NCAR calls use the standard NCAR library.

During program execution, the user is prompted for additional plot information regarding the following items:

ENLARGEMENT FACTOR:

A scaling factor to determine how much of the page or screen is filled by the plot. Usually 1 or greater.

PLOT LABEL:

The label for the plot has a maximum length of 40 characters. It is output at the top of the plot in high quality letters.

PERIMETER:

As part of the CONREC modifications, an outline may be drawn with tic marks along the side.

PEN COLOR: Self explanatory.

Subroutines deleted from CONREC include: CLGEN, REORD, STLINE, CALCNT, EZCNTR, CONBD.

The following is added after statement label 110 in CONREC:

```
C*****
C
  PRINT*," "
  PRINT*,"DO YOU WANT A PERIMETER AROUND THE PLOT? (Y OR N)"
  READ 1199,ANS
1199  FORMAT(A1)
      IF(ANS.EQ."N") GOTO 111
      CALL PERIM(NX-1,1,NY-1,1)
111  IF (ICNST .NE. 0) GOTO 124
C
C*****
```

The following are statements removed from subroutine DRLINE:

```
FX(X,Y) = X
FY(X,Y) = Y
```

The following statements are removed from subroutine MINMAX:

```
FX(X,Y) = X
FY(X,Y) = Y
```

Figure 7. Modifications to NCAR subroutine CONREC.

An example using CONTUR with data file TROUTZ (a renamed output file from TRANS) and finite difference grid file SRFDG is shown below. An example of a contour plot produced by CONTUR is shown in Figure 9. The listing for program CONTUR is shown in Figure 10.

```
/CONTURP,CONDAT=TROUTZ,FDGRID=SRFDG
```

```
*****  
*   CONTOUR PLOTTER   *  
*****
```

```
ENTER FIGURE LABEL  
1234567890123456789012345678901234567890  
? SAMPLE OF PROGRAM CONTOUR - TROUTZ
```

```
ENTER ENLARGEMENT FACTOR  
? 2
```

```
ENTER PEN COLOR  
1) BLACK  
2) BLUE  
3) GREEN  
4) RED  
? 1
```

```
DO YOU WANT A PERIMETER AROUND THE PLOT? (Y OR N)  
? N
```

```
FINISHED. FILE 'NCARMC' CONTAINS METACODE  
REVERT. CONTUR,TROUTZ
```

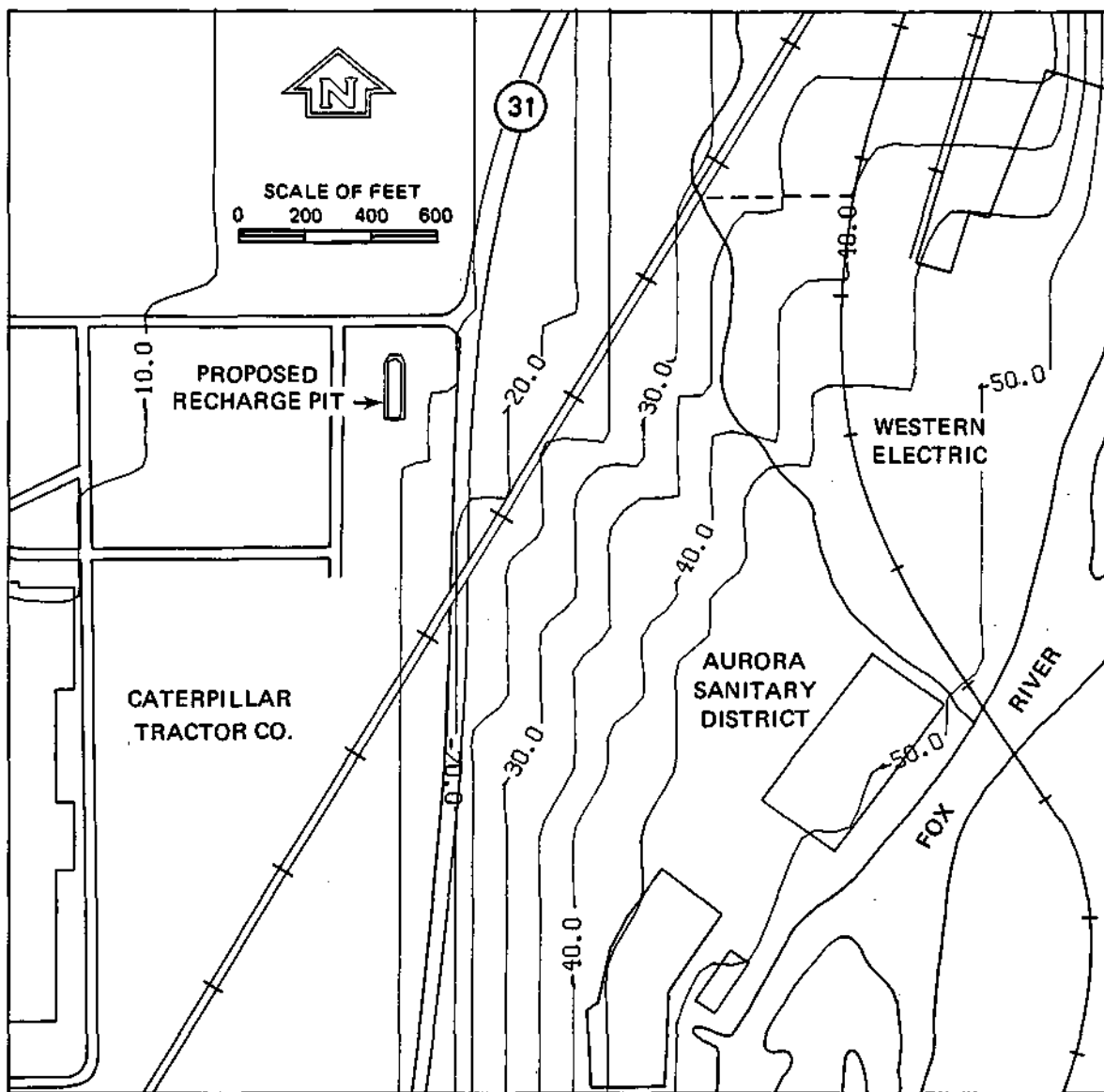


Figure 9. An example of contoured initial head values (blocked data) on demographic features (see Naymik, 1982)

```

.PROC,CONTURP,FDGRID=GRID,CONDAT=HEADS.
GET,CONRECB.
R.
IFE,.NOT.FILE(CONB,LO),A.
FTN,I=TEMP,L=0,B=CONB.
ENDIF,A.
IFE,.NOT.FILE(NCAR,PM),B.
GRAB,NCARFNT.
ENDIF,B.
LDSET(USEP=SMOOTH).
SATISFY.
LOAD,CONRECB.
CONB,,FDGRID,CONDAT.
RETURN,TEMP.
REVERT. CONTURP,CONDAT
.DATA,TEMP

PROGRAM CONTUR(INPUT,OUTPUT,GRID,DATA,NCARMC,NCRFNT,
1          TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=GRID,TAPE8=DATA)
REAL Z(100,100),SPI(99),SPJ(99),SCB,LABEL(4),A,B,SCF,PENCOL(4)
INTEGER NC,NR,M,N,IWORK(4),PEN
COMMON /CONR1/ SPI(99),SPJ(99),SCB,A,B
COMMON /CONRE1/ IOFFP,SPVAL
COMMON /CONRE4/ SIZEL,SIZEM,SIZEP,NREP,NCRT,ILAB,NULBLL,IOFFD,EXT,
1IOFFM,ISOLID,NLA,NLM,XLT,YBT,SIDE
DATA PENCOL/5HBLACK,4HBLUE,5HGREEN,3HRED/
DATA NSET,NDOT/0,0/

C
C PRINT PROGRAM HEADER
C
C WRITE (6,910)
C
C INPUT NUMBER OF COLUMNS AND ROWS IN MODEL, AND BASE GRID SPACING
C
C READ (7,960) NC,NR,SCB
C M=NC-1
C N=NR-1
C
C INPUT VARIABLE GRID SPACINGS
C
C READ (7,970) (SPI(J),J=1,M)
C READ (7,970) (SPJ(NR-J),J=1,N)
C
C INPUT PARAMETERS FOR SUBROUTINE CONREC; SEE NCAR DOCUMENTATION
C FOR FULL DETAILS
C
C READ (8,980) FLO,HI,FINC,NHI,ILAB,NULBLL
C READ (8,980) DUMMY

```

Figure 10. Listing of program CONTUR.


```

C
C INPUT DATA TO BE CONTOURED
C
      M=NC/10
      IF (MOD(NC,10).GT.0) M=M+1
      DO 30 I=1,M
        IJ=(I-1)*10
        IIJ=MINO(10,NC-IJ)
        DO 20 J=1,NR
          READ (8,990) (Z(K+IJ,NR+1-J),K=1,IIJ)
20      CONTINUE
30      CONTINUE

C
C INPUT PLOT LABEL, ENLARGEMENT FACTOR, AND PEN COLOR FROM TERMINAL.
C
      WRITE (6,920)
      READ (5,1000) (LABEL(I),I=1,4)
      WRITE (6,930)
      READ (5,*) SCALE
      A=SCALE
      WRITE (6,940)
      READ (5,*) PEN

C
      B=A
      SCF=SCB/A
      CALL ENTSRC (IOLD,1)

C
C WRITE LABEL ON PLOT
C
      CALL PWRITX (500,1000,5H"PRU",5,24,0,0)
      CALL PWRITX (500,1000,LABEL,40,24,0,0)

C
C CONTOUR DATA
C
      CALL OPTN ("COLOR",PENCOL(PEN))
      CALL CONREC (Z,100,NC,NR,FLO,HI,FINC,NSET,NHI,NDOT)

C
C WRITE SCALING LABEL ON PLOT
C
      CALL OPTN ("COLOR","BLACK")
      ENCODE (38,1010,IWORK) SCF
      CALL PWRIT (512,36,IWORK,38,8,0,0)

C
C CHECK FOR ERRORS
C
      IF (NERROR(IERROR).NE.0) GO TO 40
      CALL FRAME
      WRITE (6,950)
      STOP
40 CALL EPRINT
      STOP

```

Figure 10. Continued.

```
C
C   FORMAT STATEMENTS
C
910 FORMAT (/*****
      1      /*#   CONTOUR PLOTTER   **
      2      /***)
920 FORMAT ("  ENTER FIGURE LABEL"
      1      /*  123456789012345678901234567890")
930 FORMAT (/ "ENTER ENLARGEMENT FACTOR")
940 FORMAT (/ "ENTER PEN COLOR"/
      1      "1) BLACK"/
      2      "2) BLUE"/
      3      "3) GREEN"/
      4      "4) RED")
950 FORMAT (/ "FINISHED. FILE 'NCARMC' CONTAINS METACODE")
960 FORMAT (2I10,F10.0)
970 FORMAT (10F8.0)
980 FORMAT (3F10.0,3I10)
990 FORMAT (1X,10F10.2)
1000 FORMAT (5A10)
1010 FORMAT ("EACH TIC REPRESENTS ",G13.5,"FEET.")
      END
```

Figure 10. Concluded.

Program SURFAC

Program SURFAC is used to draw surface plots from data on uniform or variable spaced grids. The input data to SURFAC is either the output file "TROUT" from the ISWS program TRANS or from the columns of data in the input file to TRANS (see FLOW OF DATA, CONVRT.) Most plotting parameters are contained in the surface data file; however, the user is queried during program execution for some additional information. SURFAC is run interactively:

```
/SURFACP,SRFDAT=<levels>,FDGRID=<fdgrid>
```

where <levels> is the data values to be surfaced (see FLOW OF DATA)

<fdgrid> is the finite difference grid file (see GRID DATA FILES)

The file <levels> which is either the output of TRANS or CONVRT has the following format:

CARD	VARIABLES	FORMAT
1	DUMMY	(used in CONTUR)
2	SWITCH,C1,C2,R1,R2	(I2,4I10)
3	<surface data>	(see CONVRT)

where: SWITCH = 1 : Produce close-up of surface (see C1,C2,R1,R2)
 0 : No close-up produced, use entire grid

C1	=	First column of data for close-up
C2	=	Last " " " " "
R1	=	First row " " " "
R2	=	Last " " " " "

As an example, if the user has a 50 column by 30 row model, but wants only the surface columns 22-45, rows 1-24, the variables are: SWITCH=0,C1=22,C2=45,R1=1,R2=24.

During program execution, the user is prompted for additional plot information regarding the following items:

REDUCTION FACTOR:

A factor which scales the plot in the z-direction to produce a plot with all dimensions of approximately the same magnitude. This is necessary since the Z dimension may not have the same unit of measure as the X-Y dimension which is feet.

LOOK AT POINT:

The 3 dimensional point in space at which the viewer's "eye" is said to be looking. Since X and Y distances have been scaled by the value SCB (see GRID DATA FILES) and the Z distance has been scaled by the REDUCTION FACTOR, the coordinates of this are (X,Y,Z) after scaling.

FROM POINT:

The 3 dimensional point in space at which the viewer's "eye" is located. The coordinates of the viewer's "eye" and "looking at" point effectively define a line-of-sight.

PLOT LABEL:

The label for the plot has a maximum length of 40 characters. It is output on the plot in high quality letters.

LABEL ANGLE/POSITION:

The label can now be positioned anywhere and at any angle on the plot. The user should take care not to write over the plot. The angle is measured in degrees defined counterclockwise from the horizontal. The X and Y coordinates have limits 0 to 1024 inclusive with (X,Y) = (0,0) defined as the lower left corner of the plot.

PEN COLOR: Self explanatory.

An example using SURFAC with data file HEADSX (calculated water levels) and finite difference file AFDGRID is shown below:

```
/SURFACP,SRFDAT=HEADSX,FDGRID=AFDGRID
```

```
*****
*   SURFACE PLOTTER   *
*****
```

```
ENTER REDUCTION FACTOR
? 10.0
LOOK AT POINT (X,Y,Z)?
? 8,8,-10
FROM POINT (X,Y,Z)?
? 25,-5,3
  ENTER LABEL
  1234567890123456789012345678901234567890
? WATER TABLE ELEVATIONS - AURORA MODEL
ENTER LABEL ANGLE, X, and Y POSITION
? 0,500,90
ENTER PEN COLOR
1) BLACK
2) BLUE
3) GREEN
4) RED
? 2
FINISHED - FILE 'NCARMC' CONTAINS METACODE
REVERT. SURFAC,HEADSX
```

At this point, the metacode in file NCARMC may be translated and displayed on a Tektronix terminal or plotted on the ZETA plotter. See Figure 11 for an example of a surface plot produced from SURFAC. The program listing for SURFAC is shown in Figure 12. /

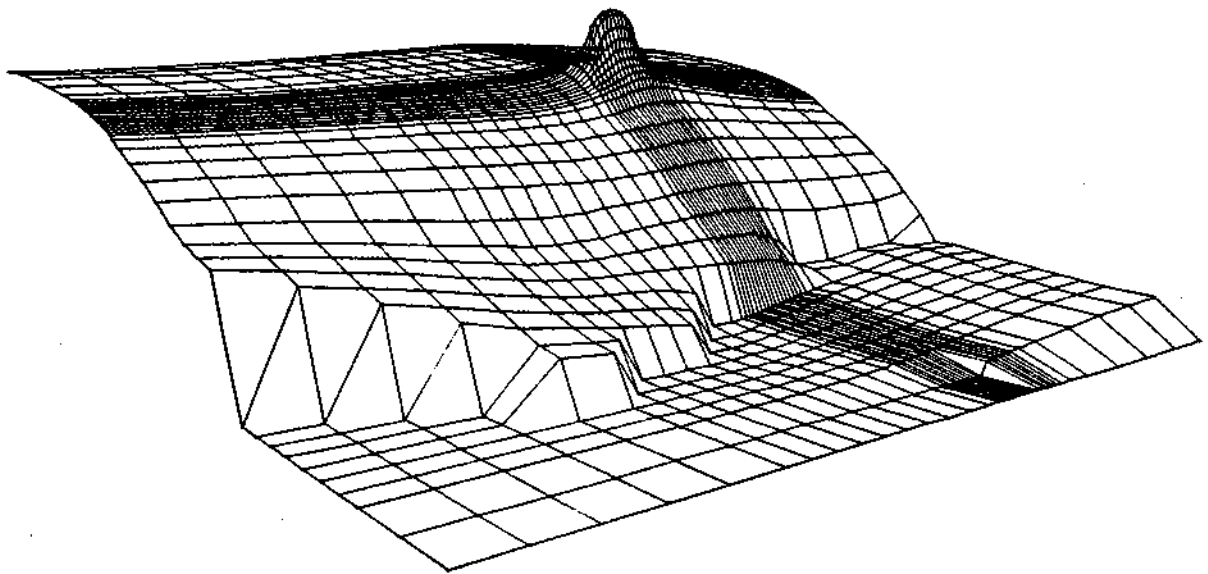


Figure 11. An example of a water table surface plot (see Naymik, 1982)

```

.PROC,SURFACP,SRFDAT=DATA,FDGRID=GRID.
R.
IFE,.NOT.FILE(SRFB,LO),A.
FTN,I=TEMP,L=0,ER,T,PMD,B=SRFB.
ENDIF,A.
IFE,.NOT.FILE(NCAR,PM),B.
ADDLIB.
USE,NCAR,NCARTRN,NCARTRB,NCARLOG/UN=LIBRARY.
ATTACH,NCAR,NCRFNT/UN=LIBRARY.
GET,NCARTRN/UN=LIBRARY.
ADDLIB+NCAR.
ENDIF,B.
SRFB,,SRFDAT,FDGRID.
RETURN,TEMP.
REVERT. SURFACP,SRFDAT
.DATA,TEMP

PROGRAM SURFAC(INPUT,OUTPUT,DATA,VARGRD,NCARMC,NCRFNT,
1          TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=VARGRD,TAPE8=DATA)
REAL Z(43,39),SPI(42),SPJ(38),SCB,LABEL(4),X(43),Y(39),S(6)
REAL MMM(2,43,39),STEREO
REAL ZZ(43,39),XX(43),YY(39),PENCIL(4)
INTEGER NC,NR,M,N,THETA,SWITCH,IWORK(7),C1,C2,R1,R2,PEN
COMMON /SRFBLK/ LIMU(1024),LIML(1024),CL(41),NCL,LL,FACT,IROT,NDRZ
1,NUPPER,NRSWT,BIGD,UMIN,UMAX,VMIN,VMAX,RZERO,IOFFP,NSPVAL,SPVAL,
2 BIGEST
COMMON /SRFIP1/ IFR,ISTP,IROTS,IDRX>IDRY>IDRZ,IUPPER,ISKIRT,
1          NCLA,THETA,HSKIRT,CHI,CLO,CINC,ISPVAL
DATA PENCIL/5HBLACK,4HBLUE,5HGREEN,3HRED/,STEREO/0/

C
C DISPLAY PROGRAM HEADER
C
C WRITE (6,910)
C
C INPUT NUMBER OF ROWS AND COLUMNS IN MODEL AND BASE GRID DISTANCE
C
C READ (7,990) NC,NR,SCB
C
C INPUT VARIABLE GRID SPACINGS
C
C M=NC-1
C N=NR-1
C READ (7,1020) (SPI(J),J=1,M)
C READ (7,1020) (SPJ(NR-J),J=1,N)
C
C CALCULATE ACTUAL X AND Y VALUES FOR EACH DATA POINT
C FROM GRID SPACINGS
C
C X(1)=0.0
C Y(1)=0.0

```

Figure 12. Listing of program SURFAC.

```

DO 20 I=1,M
  X(I+1)=X(I)+(SPI(I)/SCB)
20 CONTINUE
DO 30 I=1,N
  Y(I+1)=Y(I)+(SPJ(I)/SCB)
30 CONTINUE
C
C FIRST CARD IS USED IN PROGRAM CONTUR, SO IGNORE ON INPUT
C
C READ (8,1000) DUMMY
C
C INPUT SWITCH FOR CLOSEUP, AND COLUMN AND ROW NUMBERS OF CLOSEUP
C IF SWITCH IS ANY NUMBER OTHER THAN ZERO, A CLOSEUP PLOT
C WILL BE PRODUCED.
C
C READ (8,1010) SWITCH,C1,C2,R1,R2
C
C INPUT DATA TO BE SURFACE PLOTTED
C
C M=NC/10
C IF (MOD(NC,10).GT.0) M=M+1
C DO 50 I=1,M
C   IJ=(I-1)*10
C   IIJ=MINO(10,NC-IJ)
C   DO 40 J=1,NR
C     READ (8,1030) (Z(K+IJ,NR+1-J),K=1,IIJ)
40 CONTINUE
50 CONTINUE
C
C INPUT REDUCTION FACTOR, IF NEEDED (ENTER 1 IF NOT NEEDED)
C
C WRITE (6,920)
C READ (5,*) SCALE
C
C SCALE DATA BY REDUCTION FACTOR. THIS FACTOR IS CHOSEN
C SO AS TO PRODUCE A PLEASING PLOT, WITH NO DIMENSION
C EXCEEDINGLY LARGE COMPARED TO THE OTHER DIMENSIONS.
C
C DO 70 I=1,NC
C   DO 60 J=1,NR
C     Z(I,J)=Z(I,J)/SCALE
60 CONTINUE
70 CONTINUE
  IF (SWITCH) 80,120,80
C
C FILL ARRAYS FOR CLOSEUP
C
C 80 R1=NR+1-R1
C   R2=NR+1-R2
C   MM=C2-C1+1
C   NN=R1-R2+1

```

Figure 12. Continued.


```

DO 90 I=1,NN
  YY(I)=Y(R2+I-1)
90 CONTINUE
DO 110 I=1,MM
  XX(I)=X(C1+I-1)
  DO 100 J=1,NN
    ZZ(I,J)=Z(C1+I-1,R2+J-1)
100 CONTINUE
110 CONTINUE
C
C INPUT POINT "LOOKED AT"
C
120 WRITE (6,930)
  READ (5,*) S(4),S(5),S(6)
C
C INPUT POSITION OF "EYE"
C
  WRITE (6,940)
  READ (5,*) S(1),S(2),S(3)
C
C INPUT PLOT LABEL
C
  WRITE (6,950)
  READ 1040, (LABEL(I),I=1,4)
C
C INPUT LABEL ANGLE (DEGREES COUNTERCLOCKWISE FROM HORIZONTAL)
C LABEL X AND Y POSITION ( 0 < IX < 1024 )
C ( 0 < IY < 1024 )
C
C
C THETA = 0, IX = 500, IY = 90 PRODUCES NICE POSITIONING
C OF PLOT LABEL.
C
C
  WRITE (6,960)
  READ (5,*) THETA,IX,IY
C
C INPUT PEN COLOR
C
  WRITE (6,970)
  READ (5,*) PEN
  CALL ENTSRC (IOLD,1)
C
C FIGURE LABELING SEGMENTS
C
  CALL PWRITX (IX,IY,LABEL,40,24,THETA,0)
  CALL PWRIT (300,56,13HLINE OF SIGHT,13,8,0,0)
  ENCODE (65,1050,IWORK) S(1),S(2),S(3)
  CALL PWRIT (300,36,IWORK,65,8,0,0)
  ENCODE (65,1060,IWORK) S(4),S(5),S(6)
  CALL PWRIT (300,16,IWORK,65,8,0,0)
  ENCODE (31,1070,IWORK) SCALE
  CALL PWRIT (720,16,IWORK,31,8,0,0)

```

Figure 12. Continued.

```

C
C SURFACE PLOTTING CALLS
C
  CALL OPTN ("COLOR",PENCOL(PEN))
  IF (SWITCH) 140,130,140
C
C CALL 130 FOR FULL ARRAY PLOT, 140 FOR CLOSEUP
C
130 CALL SRFACE (X,Y,Z,MMM,43,NC,NR,S,STEREO)
  GO TO 150
140 CALL SRFACE (XX,YY,ZZ,MMM,43,MM,NN,S,STEREO)
C
C CHECK FOR ERRORS
C
150 IF (NERROR(IERROR).NE.0) GO TO 160
C
  WRITE (6,980)
  STOP
160 CALL EPRINT
  STOP
C
C FORMAT STATEMENTS
C
C
910 FORMAT (/" *****"/
1      " * SURFACE PLOTTER **"/
2      " *****"/)
920 FORMAT (" ENTER REDUCTION FACTOR")
930 FORMAT (" LOOK AT POINT (X,Y,Z)?")
940 FORMAT (" FROM POINT (X,Y,Z)?")
950 FORMAT (" ENTER PLOT LABEL. "/
1      " 1234567890123456789012345678901234567890")
960 FORMAT (" ENTER LABEL ANGLE, X, AND Y POSITION")
970 FORMAT ("ENTER PEN COLOR"/
1      "1) BLACK"/
2      "2) BLUE"/
3      "3) GREEN"/
4      "4) RED")
980 FORMAT ("FINISHED - FILE 'NCARMC' CONTAINS METACODE")
990 FORMAT (2I10,F10.0)
1000 FORMAT (3F10.0,3I10,F10.0)
1010 FORMAT (I2,4I10,F10.0)
1020 FORMAT (10F8.0)
1030 FORMAT (1X,10F10.2)
1040 FORMAT (5A10)
1050 FORMAT (18HLOOK FROM POINT X=,G13.5,4H ,Y=,G13.5,4H ,Z=,G13.5)
1060 FORMAT (18H LOOK AT POINT X=,G13.5,4H ,Y=,G13.5,4H ,Z=,G13.5)
1070 FORMAT (18HREDUCTION FACTOR =,G13.5)
  END

```

Figure 12. Concluded.

Program PRTGRF

Program PRTGRF is used to plot the positions of the solute particles after transport time steps (DELP) in the random-walk program. Particle positions are prepared for plotting through an added subroutine PLOTPT (described later in this section.) At execution time, the user has the option of having the particle cell grid plotted along with the particles. This flexibility allows for an acetate plot of a grid to be overlain on particle plots or for having both particles and the grid plotted together if acetate is not available. PRTGRF is run interactively:

```
/PRTGRFP, PAR DAT=<plotot>, FDGRID=<fdgrid>, PLOT=<device>
```

where <plotot> is the particle position data file

<fdgrid> is the finite difference cell grid file (see GRID DATA FILES)

<device> is the output device for plot (see PLOTTING DEVICES)

The particle position data file has the following format:

CARD	VARIABLE	FORMAT
1	NP, ITIME	(2I10)
2	<particle data> iteration 1	(10F7.2)
	blank	
	blank	
	blank	
	NP, ITIME	
	<particle data> iteration 2	
⋮	⋮	

where NP = Number of particles in iteration

ITIME = Iteration number - DO LOOP 620 I counter in program
TRANS. Not used in PRTGRF, only for user check

The particle data are arranged with all X-coordinate values first, 10 values per card. The corresponding Y-coordinates are next, 10 per card. Three blank lines separate DELPS. This is done simply for ease of scanning the file. See the PLOTPT subroutine for more detail.

To produce the file PLOTOT from TRANS, it is necessary to make a slight change in the TRANS "PROGRAM" statement, some additions to the TRANS 620 DO loop, and the addition of a subroutine PLOTPT (see Figure 13).

```

C*** CALL APPROPRIATE SUBROUTINES
C*** FOR AQUIFER SITUATION
C
      DO 620 I=1,NPITS
        ITIME=I
        CALL PLOTPT(ITIME)
        CALL SUMMRY
        CALL SNKCON
        CALL CLEAR
        CALL ADVAN(DELP)
        CALL GENP4
      CALL MAP
      CALL CONMAP
620    CONTINUE

```

The modification to TRANS "PROGRAM STATEMENT":

```

\PROGRAM TRANS (OUTPUT, TRIN, TROUT, PLOTOT, TAPE5=TRIN, TAPE1=TROUT,
X TAPE6=OUTPUT, DEBUG=OUTPUT, TAPE8=PLOTOT)

```

The subroutine PLOTPT to format particle information and output to UNIT 8;

```

SUBROUTINE PLOTPT(ITIME)
COMMON/TRACE/NP,MAXP,PM,DISPL,DISPT,X(5001),Y(5001)
IF(NP.LE.0) RETURN
II=NP/10
IF(MOD(NP,10).GT.0) II=II+1
WRITE(8,910) NP,ITIME
DO 10 K=1,II
  IJ=(K-1)*10
  IIJ=MIN0(10,NP-IJ)
  WRITE(8,920) (X(KK+IJ),KK=1,IIJ)
10  CONTINUE
DO 20 K=1,II
  IJ=(K-1)*10
  IIJ=MIN0(10,NP-IJ)
  WRITE(8,920) (Y(KK+IJ),KK=1,IIJ)
20  CONTINUE
WRITE(8,930)
RETURN
910  FORMAT(I10,I10)
920  FORMAT(10F7.2)
930  FORMAT(//)
END

```

Figure 13. Modifications/additions to ISWS program TRANS.

An example of running program PRTGRF is shown below. A sample output plot from PRTGRF is shown in Figure 14. The program listing for PRTGRF is shown in Figure 15.

```
/PRTGRFP, PARDAT=SRPARTS, FDGRID=SRFDG, PLOT=ZETA
```

```
*****
# PROGRAM PART-GRAF #
*****
```

```
DO YOU WANT OVERLAID GRID? (Y/N)
? Y
PARTICLES WILL BE DRAWN AS CIRCLES
ENTER RADIUS OF PARTICLES IN VIRTUAL FEET (0 IF POINT)
0
ENTER THE DESIRED WIDTH OF GRID, IN INCHES
? 6.0
```

```
ENTER TRANSPORT ITERATION NUMBER TO BE PLOTTED
ENTER 0 TO EXIT
? 4
ENTER PEN COLOR:
1) BLACK 2)BLUE 3)GREEN 4)RED
? 1
```

```
ENTER TRANSPORT ITERATION NUMBER TO BE PLOTTED
ENTER 0 TO EXIT
? 5
ENTER PEN COLOR:
DBLACK 2)BLUE 3)GREEN 4)RED
? 4
```

```
ENTER TRANSPORT ITERATION NUMBER TO BE PLOTTED
ENTER 0 TO EXIT
? 10
ENTER PEN COLOR:
DBLACK 2)BLUE 3)GREEN 4)RED
? 3
```

```
ENTER TRANSPORT ITERATION NUMBER TO BE PLOTTED
ENTER 0 TO EXIT
? 0
REVERT. GRDPRT, SRPARTS
```

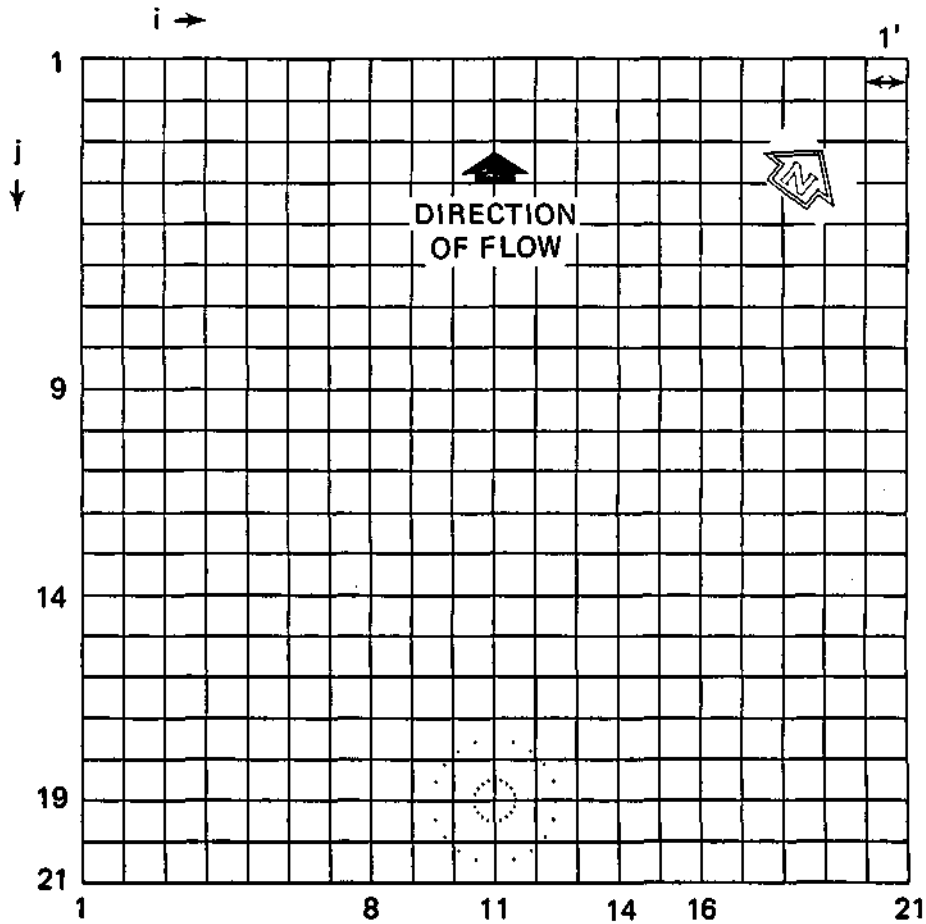


Figure 14. An example of particles plotted on a finite difference grid (see Naymik and Sievers, 1983).

```

.PROC ,PRTGRFP ,PARDAT=DATA ,FDGRID=GRID ,PLOT=ZETA .
NOTE(ZZZDEV)/PLOT
R.
IFE ,.NOT.FILE(GCS_PLOT,PM) ,A.
GRAB,GCS_PLOT/F.
ENDIF ,A.
IFE ,.NOT.FILE(PARTB,LO) ,B.
FTN ,I=TEMP ,L=0 ,B=PARTB.
RETURN ,TEMP.
ENDIF ,B.
SETTL(40).
PARTB , , PARDAT ,FDGRID.
RETURN ,ZZZDEV.
REVERT.  PRTGRF ,PARDAT
.DATA ,TEMP

      PROGRAM PRTGRF(INPUT,OUTPUT,DATA,VARGRD,ZZZDEV,TAPE5=INPUT,
1          TAPE6=OUTPUT,TAPE7=VARGRD,TAPE8=ZZZDEV,TAPE9=DATA)
C
C THIS PROGRAM DRAWS A PLOT OF THE PARTICLE POSITIONS AT THE
C TIME OF THE CALL OF 'PLOTPT' IN PROGRAM 'TRANS'.
C THE POSITIONS OF THE PARTICLES ARE STORED IN THE ARRAYS
C X AND Y IN THEIR ORIGINAL FORM AS PRODUCED BY THE "ADDED" TRANS
C SUBROUTINE "PLOTPT". THIS PROGRAM APPLIES THE NECESSARY
C TRANSFORMATIONS TO THE PARTICLE POSITIONS AS DETERMINED BY
C THE VARIABLE GRID SPACINGS, AND PLOTS THE PARTICLES AS
C REPRESENTED BY A CHARACTER CHOSEN BY THE USER.
C THE USER HAS THE OPTION AT RUN TIME TO PLOT THE PARTICLE CELL
C GRID ITSELF ON TOP OF THE PARTICLE DATA. THIS ALLOWS PLOTTING
C PARTICLE DATA ON ACETATE TO BE OVERLAID ON THE GRID, OR PLOTTING
C BOTH DATA AND GRID TOGETHER.
C
      LOGICAL FT
      REAL X(3001),Y(3001),XDIS,YDIS,PENCOL(4)
      REAL PCI(100),PCJ(100)
      COMMON /TRANSF/ FDI(100),FDJ(100),NR
      DATA PENCOL/"BLACK","BLUE","GREEN","RED"/
      DATA FDI/100*0.0/,FDJ/100*0.0/,PCI/100*0.0/,PCJ/100*0.0/
      DATA FT/.TRUE./, IPEN/1/, W/5.71/, HMAX/11.0/
C
C INPUT NUMBER OF ROWS AND COLUMNS IN MODEL, AND BASE GRID DISTANCE.
C
      READ (7,*) NC,NR,SCB
C
C INPUT DEVICE TYPE
C
      READ (8,980) DEV

```

Figure 15. Listing of program PRTGRF.

```

C
C INPUT VARIABLE GRID SPACINGS
C
M=NC-1
N=NR-1
READ (7,*) (FDI(I),I=1,M)
READ (7,*) (FDJ(NR-J),J=1,N)
C
C COMPUTE PARTICLE CELL GRID AND
C DETERMINE MAXIMUM HORIZONTAL AND VERTICAL DIMENSIONS OF
C MODEL AREA
C
FDI(NC)=FDI(M)
FDJ(NR)=FDJ(N)
C
PCI(1)=FDI(1)
XDIS=PCI(1)
DO 10 K=2,NC
  PCI(K)=(FDI(K-1)+FDI(K))*0.5
  XDIS=XDIS+PCI(K)
10 CONTINUE
C
PCJ(1)=FDJ(1)
YDIS=PCJ(1)
DO 20 K=2,NR
  PCJ(K)=(FDJ(K-1)+FDJ(K))*0.5
  YDIS=YDIS+PCJ(K)
20 CONTINUE
C
NC=NC+1
NR=NR+1
C
C PRINT OUT HEADER AND QUERY IF WANT OVERLAID GRID
C IF NO GRID IS PRODUCED, THEN A SEPARATE GRID SHOULD
C BE MADE WITH PROGRAM "GRID" ON ACETATE.
C
WRITE (6,910)
READ (5,980) IGRID
WRITE (6,970)
READ (5,*) RADIUS
IF (DEV.NE."Z") GO TO 40

```

Figure 15. Continued.


```

C
C IF ZETA PLOTTER, INPUT WIDTH OF PLOT.
C THE REST OF THE PLOT IS SCALED RELATIVE TO "WIDTH" AND
C THEN CENTERED ON THE PLOTTER. IF THE PLOT IS LARGER THAN
C WOULD FIT WELL ON THE NARROW PLOTTER, THE USER IS INFORMED
C THAT THE LARGE PLOTTER IS NEEDED AND CENTERS IT FOR 34 IN. WIDE.
C
  WRITE (6,920)
  READ (5,*) W
  H=W*(YDIS/XDIS)
  IF (H.LE.10.5) GO TO 40
  IF (H.LE.33.5) GO TO 30
  WRITE (6,930)
  STOP
30 WRITE (6,940)
  HMAX=34.0
C
C INITIALIZE GRAPHING ROUTINES
C
40 CALL USTART
  CALL UHOME
  CALL UWAIT (2.0)
  IF (DEV.EQ."Z") GO TO 50
  IF (XDIS.GE.YDIS) CALL UDAREA (0.0,W,0.0,W*(YDIS/XDIS))
  IF (XDIS.LT.YDIS) CALL UDAREA (0.0,W*(XDIS/YDIS),0.0,W)
  GO TO 60
50 IF (W.GT.7.5) CALL UDIMEN (W+1.0,HMAX)
  IF (W.LE.7.5) CALL UDIMEN (8.5,HMAX)
  CALL UDAREA (.5,W+0.5,(HMAX-H)/2,(HMAX+H)/2)
60 CALL UWINDO (0.,XDIS,0.,YDIS)
  IF (RADIUS.LE.0.0) RADIUS=XDIS/W*0.005
C
C INPUT NUMBER OF TRANSPORT ITERATION TO BE DISPLAYED.
C PROGRAM WILL READ THROUGH NPLOT-1 SECTIONS OF THE DATA FILE,
C RETAINING THE NPLOT-TH SECTION FOR PLOTTING.
C
70 WRITE (6,950)
  READ (5,*) NPLOT
  IF ((DEV.NE."Z").OR.(.NOT.FT)) CALL UERASE
  IF (NPLOT.EQ.0) GO TO 110
  IF (DEV.NE."Z") GO TO 80
C
C IF ZETA PLOTTER, INPUT PEN COLOR AND CHARACTER REPRESENTATION
C FOR PARTICLES.
C
  WRITE (6,960)
  READ (5,*) IPEN
80 REWIND 9
  DO 90 I=1,NPLOT

```

Figure 15. Continued.


```

910 FORMAT("*****"/
1      "** PROGRAM PART-GRAF **"/
2      "*****"/
3      "DO YOU WANT OVERLAID GRID? (Y/N)")
920 FORMAT("ENTER THE DESIRED WIDTH OF GRID, IN INCHES")
930 FORMAT("OUTPUT TOO BIG FOR ANY PLOTTER - NO OUTPUT")
940 FORMAT("WIDE PLOTTER NEEDED FOR THIS GRID")
950 FORMAT("/ENTER TRANSPORT ITERATION NUMBER TO BE PLOTTED"/
1      "ENTER 0 TO QUIT")
960 FORMAT ("ENTER PEN COLOR:"/
1      "1)BLACK 2)BLUE 3)GREEN 4)RED")
970 FORMAT("PARTICLES WILL BE DRAWN AS CIRCLES"/
1      "ENTER RADIUS OF PARTICLES IN VIRTUAL FEET (0 IF POINT)")
980 FORMAT (A1)
990 FORMAT (10F7.0)
1000 FORMAT (I1/I1/I1)
1010 FORMAT (I10)
      END

C
C X TRANSFORMATION
C
      FUNCTION FX (X)
      COMMON /TRANSF/ FDI(100),FDJ(100),NR
      Q=FDI(1)*0.5
      IX=X-1
      IF (IX.EQ.0) GO TO 20
      DO 10 I=1,IX
          Q=Q+FDI(I)
10     CONTINUE
20     Q=Q+(X-1-IX)*FDI(IX+1)
      FX=Q
      RETURN
      END

C
C Y TRANSFORMATION
C
      FUNCTION FY (Y)
      COMMON /TRANSF/ FDI(100),FDJ(100),NR
      Q=FDJ(1)*0.5
      IY=NR-Y-1
      IF (IY.EQ.0) GO TO 20
      DO 10 I=1,IY
          Q=Q+FDJ(I)
10     CONTINUE
20     Q=Q+(NR-1-Y-IY)*FDJ(IY+1)
      FY=Q
      RETURN
      END

```

Figure 15. Continued.

```
C
C THIS SUBROUTINE PLOTS THE GRID SO THAT
C PARTICLES MAY BE PLOTTED ON TOP TO SCALE
C
  SUBROUTINE GRID(PCI,PCJ,NR,NC,XDIS,YDIS)
  REAL PCI(100),PCJ(100),XDIS,YDIS
C
  CALL USET ("LNULL")
  CALL USET ("BLACK")
  XTEMP=0.0
  M2=NC-2
  DO 10 K=1,M2
    XTEMP=XTEMP+PCI(K)
    CALL UMOVE (XTEMP,0.0)
    CALL UPEN (XTEMP,YDIS)
10  CONTINUE
C
  YTEMP=0.0
  N2=NR-2
  DO 20 K=1,N2
    YTEMP=YTEMP+PCJ(K)
    CALL UMOVE (0.0,YTEMP)
    CALL UPEN (XDIS,YTEMP)
20  CONTINUE
  RETURN
  END
```

Figure 15. Concluded.

Program SNKGRF

Program SNKGRF is used to construct breakthrough curves of concentration versus time from data generated by program TRANS or from raw field data. The time intervals do not have to be regularly spaced. SNKGRF is run interactively:

```
/SNKGRFP,SINKDAT=<sink>,PLOT=<device>
```

where <sink> is the breakthrough data (see FLOW OF DATA)

<device> is the output device for plotting (see PLOTTING DEVICES)

The file <sink> has the following format:

CARD	VARIABLE	FORMAT
all	TIME,CONC	(2F10.0)

where: TIME = Time in DAYS

CONC = Concentration in mg/liter

Note: The data cards should be in increasing TIME order because the data are plotted in the order of the file. This will not make a difference if lines do not connect the data points, but if lines are used, the lines may cross and render the plot meaningless.

An example of running SNKGRF with the following data file SKTEST (note that in this case, the TIMES are of constant intervals, but this is not necessary):

```
SKTEST:
  0.0      0
 30.0      0
 60.0    11.7130
 90.0    323.5725
120.0    427.5256
150.0    506.5886
180.0    557.8331
210.0    604.6852
240.0     638.40
270.0     628.1
300.0     625.2
330.0     626.6
360.0     635.43
390.0     651.50
420.0     647.1
450.0     329.4
480.0     194.7
510.0     120.0
540.0      65.9
570.0      19.0
600.0      23.4
630.0      13.2
660.0       4.4
690.0       1.5
720.0       5.9
```

```
/SNKGRF,SINKDAT=SKTEST,PLOT=ZETA ; ZETA is the default
```

```
*****
* BREAKTHROUGH CURVES *
*****
```

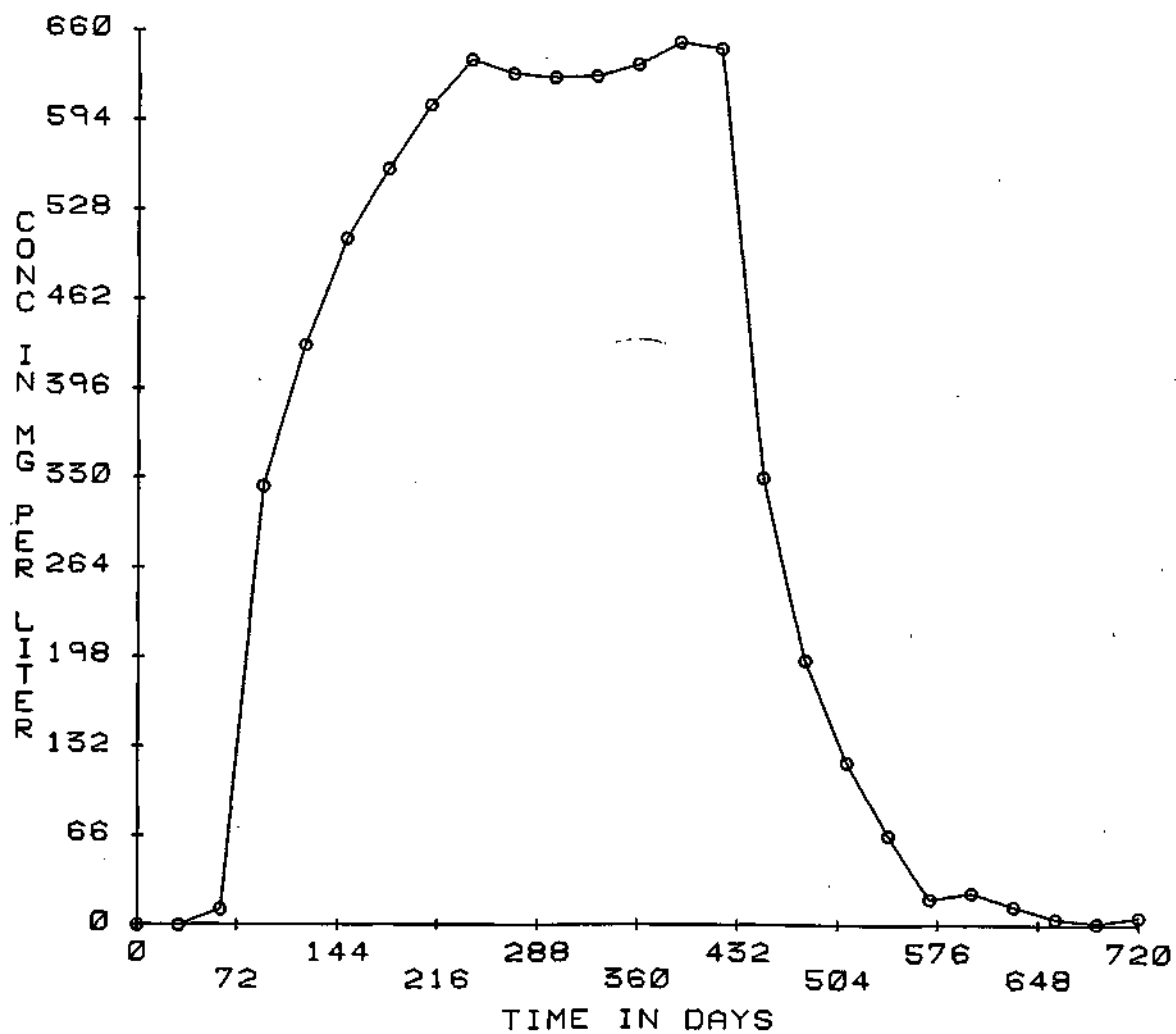
```
XMAX= 720.0
YMAX= 651.50
```

```
ENTER XLIMIT,YLIMIT
? 720,660 ;Give round numbers for nice plot
LINES CONNECTING DATUM (Y/N)
? Y ;Connect points
ENTER FIGURE LABEL
? 1234567890123456789012345678901234567890
EXAMPLE SINK #1 FROM AURORA MODEL
```

```
REVERT. SNKGRF,SKTEST
```

At this point, the plot is in file "TAPE99" to be plotted on ZETA.

The sample input file is plotted using SNKGRF in Figure 16. The program listing for SNKGRF is shown in Figure 17.



EXAMPLE SINK 1 FROM AURORA MODEL

Figure 16. An example of a SNKGRF plot from a model sink (see Naymik, 1982).

```

.PROC,SNKGRFP,SINKDAT=DATA,PLOT=ZETA.
IFE,.NOT.FILE(GCS_PLOT,PM),A.
GRAB,GCS_PLOT/F.
ENDIF,A.
R.
NOTE(ZZZDEV)/PLOT
IFE,.NOT.FILE(SINKB,LO),B.
FTN,I=TEMP,L=0,B=SINKB,ER,T,PMD,A.
ENDIF,B.
RETURN,TEMP.
SINKB,,,SINKDAT.
RETURN,ZZZDEV.
REVERT.SNKGRFP,SINKDAT
.DATA,TEMP

```

```

PROGRAM SNKGRF(INPUT,OUTPUT,DATA,ZZZDEV,
1          TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=DATA,TAPE8=ZZZDEV)
C
C THIS PROGRAM PLOTS THE POLLUTANT CONCENTRATIONS AT A GIVEN
C SINK VS TIME FOR DATA PRODUCED BY "TRANS", OR FIELD DATA.
C
REAL X(100),Y(100),LABEL(5),STUD(8)
DATA YMAX/0.0/,TX/5/
LABEL(5)=";"
C
C INPUT THE DATA FROM THE INPUT FILE.
C
N=0
20 N=N+1
READ (7,920) X(N),Y(N)
IF (EOF(7).NE.0.0) GO TO 30
YMAX=AMAX1(YMAX,Y(N))
GO TO 20
30 N=N-1
XMAX=X(N)
C
C IN TIMESHARING, THE VALUE OF THE LARGEST DATA POINTS ARE DISPLAYED
C TO ALLOW PROPER DIMENSIONING OF THE PLOT.
C
WRITE (6,930) XMAX,YMAX
READ (5,*) XMAX,YMAX
TX=XMAX/10.0
TY=YMAX/10.0

```

Figure 17. Listing of program SNKGRF.


```

C
C THE DATA POINTS MAY OR MAY NOT BE CONNECTED WITH LINES
C THIS IS AN OPTION ANSWERED HERE BY USER DURING EXECUTION
C
  WRITE (6,940)
  READ (5,910) ILINE
C
C INPUT LABEL FOR PLOT. THIS WILL BE DISPLAYED ON ZETA PLOTS
C AND PLOTS GENERATED ON TERMINAL SCREENS.
C
  WRITE (6,950)
  READ (5,960) (LABEL(I),I=1,4)
C
C INPUT PLOT DEVICE TYPE. (TEKT, ALPH, ZETA, ETC.)
C
  READ (8,910) DEV
C
C INITIALIZE GRAPHING ROUTINES
C
  CALL USTART
  CALL USTUD (STUD)
  CALL UDAREA (0.5,STUD(6)*0.80,STUD(8)*0.25,STUD(8)*0.90)
  CALL UWINDO (0.,XMAX,0.,YMAX)
  CALL USET ("OWNSCALE")
  CALL USET ("XBOTH")
  CALL USET ("YBOTH")
C
C SET "TICX" TO BE THE NUMBER OF DAYS PER TIME INTERVAL
C
  CALL UPSET ("TICX",TX)
  CALL UPSET ("TICY",TY)
C
C AXIS LABELS ARE SET HERE.
C
  CALL UPSET ("XLABEL","TIME IN DAYS;")
  CALL UPSET ("YLABEL","CONC IN MG PER LITER;")
C
C PLOT THE DATA
C
  CALL USET ("LNULL")
  IF(ILINE.EQ."N") CALL USET("NNULL")
  CALL UPLOT1 (X,Y,FLOAT(N))
  CALL USET("INCHES")
  DO 10 I=1,N
  CALL UMOVE(X(I),Y(I))
  CALL USET("DEVICE")
  CALL UWHERE(XTEMP,YTEMP)
  CALL UCRCLE(XTEMP,YTEMP,0.03)
  CALL USET("VIRTUAL")
10 CONTINUE

```

Figure 17. Continued.

```

C
C   SET UP THE PLOT MESSAGE
C
      CALL UMOVE(X(1),Y(1))
      CALL USET ("ACENTER")
      CALL UDAREA (STUD(5),STUD(6),STUD(7),STUD(8))
      CALL UWINDO (0.,300.,0.,300.)
      CALL UPRINT (150.,60.,LABEL)
C
C   WRAP UP GRAPHING.
C
      CALL UFLUSH
      IF (DEV.EQ."Z") GO TO 40
      CALL UBELL
      CALL UPAUSE
40 CALL UERASE
      CALL UEND
      STOP
C
910 FORMAT(A1)
920 FORMAT(F10.0,F10.0)
930 FORMAT(/"*****"/
1      " * BREAKTHROUGH CURVES *"/
2      "*****"/
3      " XMAX= ",F7.2/
4      " YMAX= ",F7.2//
5      "ENTER  XLIMIT,YLIMIT")
940 FORMAT("LINES CONNECTING DATUM (Y/N)")
950 FORMAT (" ENTER FIGURE LABEL"/
1      " 1234567890123456789012345678901234567890")
960 FORMAT (4A10)
      END

```

Figure 17. Concluded.

REFERENCES

- CDC NOS Version 1 Reference Manual, Volume 1 of 2, Publication No. 60435400, revision J, Literature and Distribution Services, 308 North Dale Street, St. Paul, Minn. 35103.
- McArthur, G.R. 1978. An Introduction to the SCD Graphics System, National Center for Atmospheric Research/TN-161+1A, Boulder, Colorado, Contributor Lofton R. Henderson, paging by sections.
- Naymik, T.G. 1982. Modeling as a tool in monitoring network design. Proceedings of Second National Symposium on Aquifer Restoration and Ground Water Monitoring, D.M. Neilsen, editor, pp. 151-155.
- Naymik, T.G. and M.E. Sievers. 1983. Groundwater tracer experiment (I) at Sand Ridge State Forest, Illinois. Illinois State Water Survey Contract Report 317, 69 p.
- Prickett, T.A. and C.G. Lonquist. 1971. Selected digital computer techniques for groundwater resource evaluation. Illinois State Water Survey Bulletin 55, 62 p.
- Prickett, T.A., T.G. Naymik, and C.G. Lonquist. 1981. A "random-walk" solute transport model for selected groundwater quality evaluations. Illinois State Water Survey Bulletin 65, 103 p.
- U.S. Army Eng. Waterways Experimental Station. 1978. Graphics Compatibility System (GCS), Automatic Data Processing Center, P.O. Box 631, Vicksburg, Miss. 39108.