TWO-DIMENSIONAL ANALYTICAL MODEL (FORTRAN) FOR PREDICTION OF CONTAMINANT MOVEMENT IN GROUND WATER

by

Douglas C. Kent
School of Geology
Oklahoma State University
Stillwater, Oklahoma 74078

Jan Wagner
School of Chemical Engineering
Oklahoma State University
Stillwater, Oklahoma 74078

Fred E. Witz
School of Geology
Oklahoma State University
Stillwater, Oklahoma 74078

U.S. Environmental Protection Agency

Cooperative Agreement No. CR811142-01-0

Project Officer

Carl G. Enfield
Robert S. Kerr Environmental Research Laboratory
U.S. Environmental Protection Agency
Ada, Oklahoma 74820

Disclaimer

Although the information described in this document has been funded wholly or in part by the United States Environmental Protection Agency under cooperative agreement CR806931 to the National Center for Ground Water Research and cooperative agreement CR811142-01-0 to Oklahoma State University, it has not been subjected to the Agency's required peer and administrative review and therefore does not necessarily reflect the views of the Agency and no official endorsement should be inferred.

Part I

Program Description

Introduction

A FORTRAN computer program has been developed to calculate plume concentrations. The equation is derived from Wilson and Miller (1978). The program can calculate and display the concentration at a single point or as a grid map of concentration. The parameters needed to describe the problem are defined in Table I-1.

The program was developed and tested using Microsoft FORTRAN-80 (FORTRAN 66 standard) on the Kaypro microcomputer and Microsoft FORTRAN version 2.0 on (FORTRAN 77 standard) the IBM PC microcomputer. With only minor changes, the program should function using any corresponding FORTRAN compiler. The program should function within 40K bytes of random access memory (RAM) on most computers.

---

* CPM is a registered trademark of Digital Research.

TABLE I-1

DEFINITION OF TERMS

			(Units)
Primary Variables:			
C	=	Concentration of leachate at a specific time and distance.	(M/L^3)
X	=	Distance from source where concentration of leachate is computed. Distance is measured in direction of ground-water flow (gradient).	(L)
y	=	Transverse distance measured from the center-line of ground-water flow.	(L)
t	=	Sample time from beginning of leachate source flow.	(T)
Aquifer Parameters:			
m	=	Effective aquifer thickness or zone of mixing.	(L)
n	=	Effective porosity of aquifer or zone of mixing.	(dimensionless)
V	=	Velocity of ground-water flow within voids; estimated directly or from: $V = \frac{KI}{n}$ (or) $V = \frac{TI}{mn}$	(L/T)

where:

K	=	Coefficient of permeability or hydraulic conductivity of aquifer or zone of mixing.	(L/T)
T	=	Transmissivity of aquifer or zone of mixing.	(L^2/T)
I	=	Gradient of ground-water flow.	(dimensionless)

TABLE I-1
continued

Transport Parameters: (Units)

D_x = Longitudinal dispersion coefficient (mixing rate) in the x direction; estimated directly or from: (L^2/T)

$$D_x = a_x V + D_m$$

where:

a_x = Longitudinal dispersivity. (L)

D_m = Molecular diffusion coefficient, which is assumed to be negligible for velocities typical of permeable aquifers. D_m may be the dominant process in aquitards where $a_x V$ would be negligible ($V < 0.1$ cm/yr).

D_y = Transverse dispersion coefficient (mixing rate) in the y direction; estimated directly or from: (L^2/T)

$$D_y = a_y V + D_m \quad \text{(or)} \quad D_y = \frac{D_x}{D_r} + D_m$$

where:

a_y = Transverse dispersiviity (L)

D_r = a ratio which commonly ranges between 5 and 10 for medium to coarse sand aquifers. (dimensionless)

R_d = Retardation factor; estimated directly or from: (dimensionless)

$$R_d = 1 + \frac{p_b K_d}{n_t} \quad \text{(or)} \quad R_d = \frac{V}{V_d}$$

where:

p_b = Bulk density of aquifer medium (M/L^3)

n_t = Total porosity. (dimensionless)

K_d = Distribution factor for sorption on aquifer medium (from sorption isotherm column studies). (L^3/M)

V = Velocity of ground water. (L/T)

V_d = Observed velocity of leachate for a given concentration and chemical species. (L/T)

TABLE I-1
continued

Transport Parameters (continued): (Units)

γ = (Gamma) Coefficient for radioactive or biological decay. For no decay, the value is one. Calculated from: (dimensionless)

$$\gamma = 1 + \frac{4D_x\lambda}{V^2} \quad \text{(or)} \quad \gamma = 1 + \frac{4D_x \log(2)}{V^2\, t_{1/2}}$$

where:

λ = (Lambda) Decay constant. (1/T)

$t_{1/2}$ = Halflife; time when half of the original mass remains. (T)

Source Rate of Leachate:

Q_m = Mass flow rate estimated directly or obtained from: (M/T)

$$Q_m = AQ_r \quad \text{(or)} \quad Q_m = QC_o$$

$$\text{(or)} \quad Q_m = AQ_vC_o$$

where:

Q_r = Mass per area flow rate. (M/L^2T)

Q = Volume flow rate. (L^3/T)

Q_v = Recharge rate. (L/T)

A = Area of source. (L^2)

C_o = Initial concentration. (M/L^3)

TABLE I-1
continued

Intermediate Variables: (Units)

r = A weighted distance or radius; given by: (L)

$$r = \sqrt{\left(x^2 + \frac{D_x}{D_y} y^2\right)}$$

X_D = A characteristic dispersion length or scale factor; given by: (L)

$$X_D = \frac{D_x}{\sqrt{\gamma}\, V}$$

T_D = A characteristic dispersion time or scale factor; given by: (T)

$$T_D = \frac{R_d D_x}{\gamma\, V^2}$$

Q_D = A characteristic dilution-dispersion flow; given by: (L^3/T)

$$Q_D = nm\sqrt{D_x D_y}$$

r_m = Minimum distance from a non-point source for which equation has a certain accuracy; given by: (L)

$$r_m = \frac{V\sqrt{\gamma}\, L^2}{50\, D_x N}\left(1 + \frac{D_x}{D_y}\right)$$

(or)

$$r_m = \frac{L^2}{50\, X_D N}\left(1 + \frac{D_x}{D_y}\right)$$

where:

N = Allowable approximation accuracy. (dimensionless)

L = The greater of the source length and width.

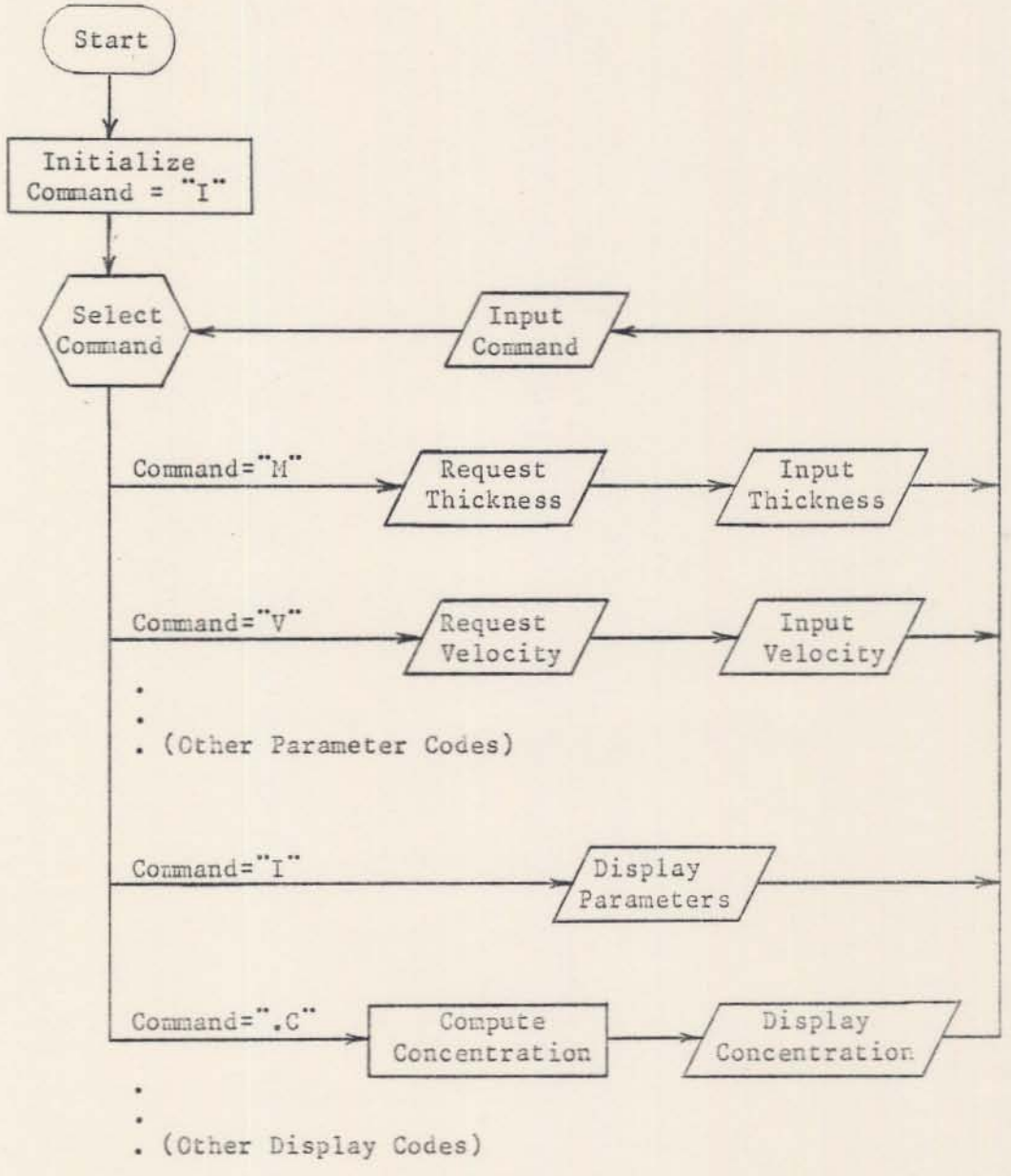

Figure I-1. Program Logic Flowchart

Assumptions and Limitations

The concentration equation is based on the following assumptions:

1. The ground-water flow regime is saturated.
2. The aquifer is unlimited (infinite) in areal extent (x and y directions).
3. All aquifer properties are homogeneous.
4. The ground-water flow is continuous and uniform in direction and velocity.
5. There is no dilution of the plume from recharge outside the source area.
6. The leachate source is a point in plain view.
7. The leachate is evenly distributed over the vertical dimension of the saturated zone.
8. The leachate source supplies a constant mass flow rate.

Although the program has been tested, the current version is subject to revision. As with any complex computer program, the results should be checked by professionals whenever safety is involved. The authors cannot assume any liability for damage resulting from the use of this program.

Program Description

Initialization

When the FORTRAN version is started, the user has the choice of entering all of the required parameters or recovering parameters saved during a previous execution of the program with the "OD" command. The units for any parameter (as listed in Appendix 1) can be modified in the BLOCK DATA section of the program, if the corresponding conversion factor (variable beginning with "U") is modified accordingly.

Commands

As shown by the flowchart in Figure I-1, once initialized, the program operates on the basis of requesting a command from the user, which designates a particular operation to be performed. In the FORTRAN version, all commands must be entered in upper case. (Many keyboards have a special shift for this purpose.) The commands are listed in Appendix I-A. The commands may be entered in any order, one at a time. A more detailed listing of the parameters and corresponding commands is also shown in Appendix I-A. The parameters are explained in Table I-1. During execution of the program, all parameters retain their values until changed by the user.

In some cases, more than one command is provided to enter a given parameter or to perform a given operation. Some commands provide a menu to select other commands (e.g. "D"), whereas some commands simply perform two or more other commands (e.g. ".IG"). When first using the program, most users will prefer to learn only one command for each operation. The commands used in the examples should serve this purpose and will be sufficient for most problems. Other commands for faster operation can be learned later.

Dispersion Coefficients

The program accepts either dispersion coefficient ("DX", "DY") or dipersivity ("AX", "AY"). If dispersivity is entered, then the dispersion coefficient will change whenever velocity is changed. Similarly, when the dispersion ratio ("DR") is used, the y dispersion coefficient ("DY") will change whenever the x dispersion coefficient ("DX") is changed. If the user enters a value for the y dispersion coefficient ("DY"), then the value entered will be retained until changed by the user.

Source data

The FORTRAN version accepts the source flow rate in any of the following forms:

1. Total mass flow
2. Mass per area flow
3. Volume flow and concentration
4. Volume per area flow and concentration

The basic equation for concentration assumes a constant source flow rate. However, the equation can be applied to a source with a number of time steps, each having a constant flow rate. Computer memory limits the total number of time steps from all sources. The limit, displayed by the "Q" command, will vary from computer to computer. The limit, displayed by the "Q" command, will vary from computer to computer. The concentration can be displayed for sample times during any time step. (Time steps that have not begun are ignored.)

When organizing data for the program, the user must select a zero reference time. Source starting times (beginning of time step) and sample

times (when concentration is calculated) are then described as the length of time before (negative time) or after (positive time) the reference time. The zero reference time must be chosen before the earliest desired sample time, because negative sample times are not allowed.

As time passes, the concentration at a given location reaches a constant value known as steady state. The steady state value for concentration can be useful, for example, as a "worst case" scenario. With the FORTRAN version, a negative value for sample time (usually -1) is used to request concentration at steady state. This is why actual sample times cannot be negative. (Negative values are allowed for starting times of source time steps.)

In the basic equation, the contaminant is assumed to enter the ground water directly below a point source. In practice, however, the equation can be used for locations far enough from a non-point source so that the source appears to be a point. Wilson and Miller (1978) provided two equations which relate the accuracy of approximation to the distance from the source, as shown in Table I-1. The FORTRAN version uses these equations to calculate an estimate of the accuracy of the results.

The FORTRAN version accepts data for either point or non-point sources. Since the equation requires sources to appear as point sources, the program divides non-point sources into subareas using one of the accuracy equations in the manner shown in Table I-1. The desired accuracy, initially 10% or 0.1, is set by the "QE" command. Sample locations near a large source could take considerable computation time (e.g 5 seconds on a microcomputer). To limit the computation to a reasonable time, a limit is placed on the number of subareas for each source. The limit is initially set at 100 subareas, but may be changed by the "QN" command.

The FORTRAN version accepts data for multiple sources. As a result, the source location is requested. As with time, the user must select an arbitrary zero reference location. Source and sample locations are then described as the distance down-gradient (+x) or up-gradient (-x) from the zero reference and the perpendicular distance from the x axis (+ or -y). After all information for one source has been entered, the program will allow adding another source. Another source can also be added after concentration has been displayed by re-entering the "Q" command. The number of sources is limited by the total number of time steps for all sources.

Source data may be changed during the use of the "Q" command by entering a negative number to "back up" or delete time steps. For example, if two sources have been entered each having three time steps, entering a negative number will have the following effect:

NUMBER ENTERED	RESULT
-1	First source unchanged. Second source with 2 time steps.
-2	First source unchanged. Second source with 1 time step.
-3	First source only.
-4	First source with 2 time steps.
-5	First source with 1 time step.
-6	No source data.
<-6	No source data.

The ".IQ" (or ".DQ") command may be used to observe these affects.

In addition to completely changing source data with the "Q" command, the FORTRAN version also provides the "QM" command which allows the user to change the mass flow rate. The "QM" command will prompt the user for a time step number to change. Entering a value of -1 will list the time steps with the current source flow rate. If a time step number is entered the program will prompt for the mass flow rate. The rate must be entered as a mass flow rate regardless of the original form.

Grid Map

The ".G" command will calculate and print a map of the concentration for locations on a grid. The first time the grid map command is entered, the FORTRAN version will prompt for the grid limits, if the "L" command has not been used. The FORTRAN version requires only

1. The x lower limit (left edge),
2. The x upper limit (right edge) or the x spacing (horizonal interval between nodes)
3. The y lower limit (top edge)
4. The y upper limit (bottom edge)

If the x upper limit or spacing is omitted, the number of nodes will be determined by the "SN" parameter. The "SN" parameter is normally set to zero, in which case the number of nodes will be adjusted to fill the length of line set by the "SL" parameter. The number of columns per node is set by the "SX" parameter. The number of lines per node is set by the "SY" parameter. The y spacing will default to the x spacing (square nodes). If the x upper limit and spacing of "SN" specify a map wider than the length of line ("SL"), then the map will be displayed in more than one section, which can be joined together. The initial values are

SL = 80 columns per line,

SN = 0 nodes per grid,

SX = 6 columns per node,

SY = 2 line per node.

If all of the concentration values on the grid map appear as zeros, most likely the values are too small to print. In this case, use the "SC" command to enter a multiplier of 10,100, etc. Values which are too large to print

will be converted to a (magnitude) + (one digit exponent of ten) and marked by a negative sign. (If the exponent exceeds ten, only the exponent is shown.) The larger values can be reduced with a multiplier of 0.1, 0.01, etc. The ".C" command can used to print the actual concentration value, in order to choose a scale factor. Values at or within a source location will be shown as "-1".

Prompting For All Parameters

The "IP" command prompts for all problem parameters. The following parameters are not prompted for:

1. The input units ("IR" and "IL" commands).
2. The output options and units ("OW", "OP", "OE" and "OT" commands).
3. The grid map scale parameters ("SC", "SL", "SN", "SX" and "SY" commands).

These parameters retain the same values as before the "IP" command. They all have initial values as shown in Appendix I-B.

Saving Parameter Values

The "OD" command saves all problem parameters to a file. The "IL" command restores the saved parameters. The following parameters are not saved:

1. The input units ("IR" and "IL" commands).
2. The output options and units ("OW", "OP", "OE" and "OT" commands).

These parameters retain the same values as before the "IL" command.

REFERENCES

Wilson, J. L. and P. J. Miller. 1978. Two-dimensional Plume in Uniform Ground-Water Flow. Journal of Hydraulics Div. Am. Soc. of Civil Eng. Paper No. 13665. HY4, pp. 503-514.

Appendix I-A

List of Commands

COMMAND	PARAMETERS SET OR ACTION TAKEN
A	Dispersivity menu for AX,AY or AX,AR.
AX	Dispersivity in x direction.
AY	Dispersivity in y direction.
AR	Dispersion ratio (Dx/Dy = Ax/Ay).
C	Case title.
D	Dispersion coefficient menu for DX,DY or DX,DR or AX,AY or AX,DR.
DX	Dispersion coefficient in x direction.
DY	Dispersion coefficient in y direction.
DR	Dispersion ratio (Dx/Dy).
DM	Molecular diffusion coefficient.
E	Exit from program.
G	Decay menu for GG, GL or GT.
GG	Decay coefficient (gamma).
GL	Decay lambda.
GT	Decay half-life time.
H	Help. List all one letter codes.
I	Input menu for IP, IL, IR, or I commands.
IP	Prompt for all parameters.
IL	Load parameters previously stored by OD command.
IR	Read input from another source.
L	Grid limits, LX and LY.
LX	Grid limits in x drection.
LY	Grid limits in y direction.
M	Aquifer thickness.
O	Output menu for OD, OW, OP, OE or OT.
OD	Dump all parameters to disk to be restored by IL command.
OW	Write results to another destination.
OP	Set prompting options.
OE	Set echo options.
OT	Set trace options.
P	Porosity
Q	Source data. (See Table X-3.)
QE	Desired number of subareas for non-point source.
QN	Maximum number of subareas for non-point source.
QM	Change source mass flow rate.

COMMAND	PARAMETERS SET OR ACTION TAKEN
R	Retardation factor.
S	Grid map scale parameters menu for SC, SL, SN, SX or SY.
SC	Grid map multiplier (concentration or steady state)
SL	Line length.
SN	Number of nodes per line.
SX	Node spacing in x direction.
SY	Node spacing in y direction.
T	Sample time.
TE	Steady State Time.
V	Velocity.
XY	Sample x and y location for .C command.
X	Sample x location for .C command.
Y	Sample y location for .C command.
ZM	Aquifer thickness (same as M command).
.C	Display single point concentration.
.D	Display parameters and source data.
.DP	Display parameters.
.DQ	Display souce data.
.DC	Display parameters, source data and single point concentration.
.DG	Display parameters, source data and concentration grid map.
.FF	Page printer. (Form feed.)
.G	Display concentration grid map.
.I	Display input parameters and source data.
.IP	Display input parameters.
.IQ	Display input source data.
.IC	Display input parameters, source data and single point concentration.
.IG	Display input parameters, source data and grid map.
.T	Display of Steady State Time.
.TG	Display of Steady State time grid map.

APPENDIX I-B

PARAMETER VARIABLES AND COMMANDS

COMMAND	DATA VARIABLE	UNIT VARIABLE	INITIAL UNIT	DESCRIPTION (INITIAL VALUE)
C	HHC1	-	-	Title, any 72 characters.
C	HHC2	-	-	Title, any 72 characters.
C	HHC3	-	-	Title, any 72 characters.
M,ZM*	ZM	UZM	FT	Aquifer thickness.
P	P	-	-	Porosity.
V	V	UV	FT/D	Velocity.
DX	DX	UD	FT2/D	X Dispersion coefficient.
DY	DY	UD	FT2/D	Y Dispersion coefficient.
AX	AX	UA	FT	Dispersivity in x direction.
AY	AY	UA	FT	Dispersivity in y direction.
DR,AR*	DR	-	-	Dispersion ratio, Dx/Dy = Ax/Ay.
DM	DM	UD	FT2/D	Molecular diffusion coefficient, (0.0)
R	R	-	-	Retardation factor.
GG	GG			Decay coefficient, gamma.
GL	GL	UGL	1/YR	Decay lambda.
GT	GT	UGT	YR	Decay half-life time.
Q	QQXL	UQL	FT	Source minimum x location.
Q	QQXM	UQL	FT	Source maximum x location.
Q	QQYL	UQL	FT	Source minimum y location.
Q	QQYM	QQL	FT	Source maximum y location.
Q	QQSL	UQL	FT	Source minimum size.
Q	QQSM	UQL	FT	Source maximum size.
Q	QQA	UQA	FT2	Source area.
Q	QQT	UQT	DAYS	Source time.
Q	QQV	UQV	FT/D	Source volume flow rate/area.
Q	QQ	UQ	FT3/D	Source volume flow rate.
Q	QQC	UQC	MG/L	Source concentration.
Q	QQR	UQR	LB/FT2/D	Source mass flow rate/area.
Q,QM*	QQM	UQM	LBM/D	Source mass flow rate.
QE	NQE	-	-	Desired accuracy. (0.1)
QN	NQN	-	-	Maximum number of subareas. (100)

* Either command may be used.

COMMAND	DATA VARIABLE	UNIT VARIABLE	INITIAL UNIT	DESCRIPTION (INITIAL VALUE)
T	TC	UTC	DAYS	Sample time.
TE	TE	-	percent	Percent of Steady State.
XC	X$C	ULC	FT	Sample x location.
YC	Y$C	ULC	FT	Sample y location.
LX	XGL	ULC	FT	Grid x minimum.
LX	XGM	ULC	FT	Grid x maximum.
LX	XGI	ULC	FT	Grid x increment.
LY	YGL	ULC	FT	Grid y minimum.
LY	YGM	ULC	FT	Grid y maximum.
LY	YGI	ULC	FT	Grid y increment.
SC	SC	-	-	Concentration multiplier. (1)
SL	NSL	-	-	Print line length. (80 characters)
SN	NSN	-	-	Nodes per line. (0)
SX	NSX	-	-	Grid x spacing. (6 characters/node)
SY	NSY	-	-	Grid y spacing. (3 lines/node)
.C	C	UC	MG/L	Result concentration.
IR	LUR	-	-	FORTRAN unit for input. (*)
	LUW	-	-	FORTRAN unit for results. (*)
	BOP,LUP	-	-	FORTRAN unit and option for prompting. (Prompting on, *)
OE	DOE,LUE	-	-	FORTRAN unit and option for echo. (Echo off, *)
OT	BT1-BT8	-	-	Trace options for program development. (All off)
	BATCH	-	-	Bath option: abort command when error occurs. (**)

* Value for unit depends on system. Unit 1 is used for most microcomputer systems.

** Batch option is set in program and cannot be changed by user.

PART II

PROGRAM OPERATIONS AND APPLICATIONS

The program begins by prompting the user with 4 options labeled 1,2,3, and -1.

```
 1    TO PROMPT FOR ALL REQUIRED PARAMETERS (IP),
 2    TO LOAD PREVIOUSLY SAVED PARAMETERS (IL),
 3    TO READ COMMANDS FROM ANOTHER SOURCE (IR),
-1    TO SET OUTPUT PARAMETERS (O):
```

In order to enter your hydrogeologic parameters, the user must prompt for them by entering:

```
? 1
```

and then pressing RETURN, the program will now ask you for "Three Title Lines". This allows the operator to document the case study being modeled. On each title line a specific characteristic for the case study can be documented, thus distinguishing one computer run from another. For example: Location of Problem, Type of contaminate, Source of data. The PROMPT for Problem title:

```
THREE TITLE LINES:
```

THE USER RESPONDS WITH (80 characters per line):

```
? Babylon site, N.Y.
? Chloride, single point source
? O.S.U. Consultants
```

Once documented, the program prompts the operator for the hydrogeologic parameters.

The first parameter prompted for is Saturated Thickness.

```
     THICKNESS          (FT):
```

THE USER RESPONDS WITH (5 characters per variable):

```
     ?  110
```

Next porosity is PROMPTED FOR.

```
     POROSITY           (UNITLESS):
```

THE USER RESPONDS WITH (5 characters per variable):

```
     ?  .35
```

Then you are PROMPTED for ground-water velocity

```
     VELOCITY           (FT/D):
```

THE USER RESPONDS WITH (5 characters per variable):

```
     ?  1.5
```

Now the program cues the modeler to construct a grid map. It is easiest to begin with a simple grid (i.e. 10 x 10). Superimpose the grid on the potentiometric map of the problem site. Grid squares are then assigned to points of interest. For example, a simplified schematic of a potentiometric map with a source and sample locations is shown in Figure 1. The model orients flow from left to right. Therefore, the map should be oriented perpendicular to the equipotential contours. Prepare a grid such as the 10 x 10 grid shown in Figure 2.

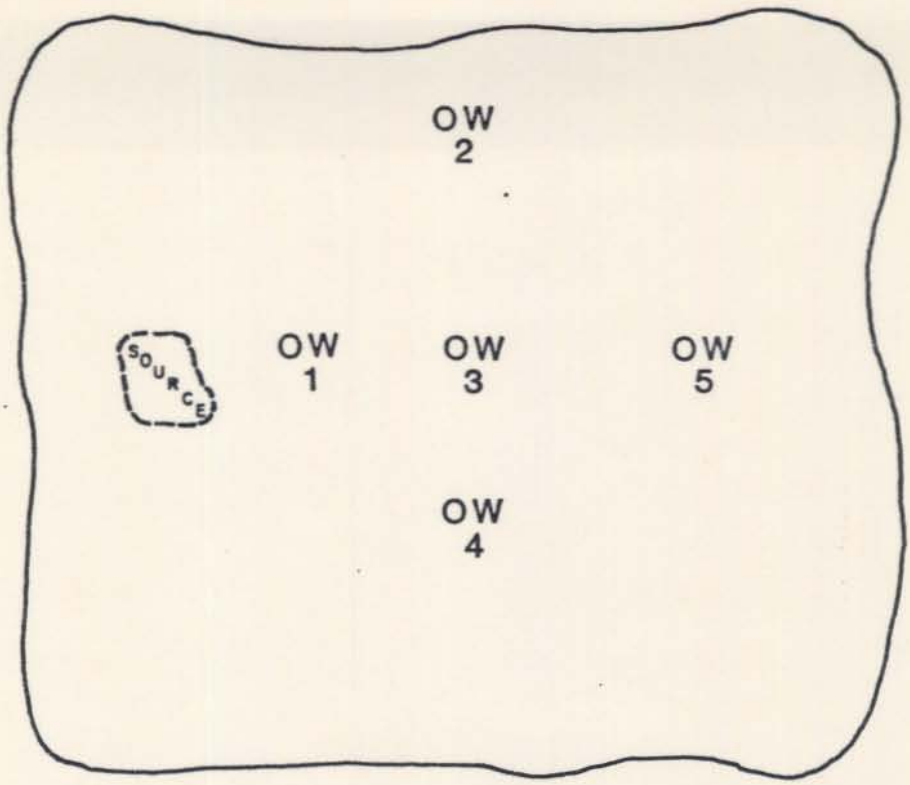

Figure 1
OW=observation wells for sampling

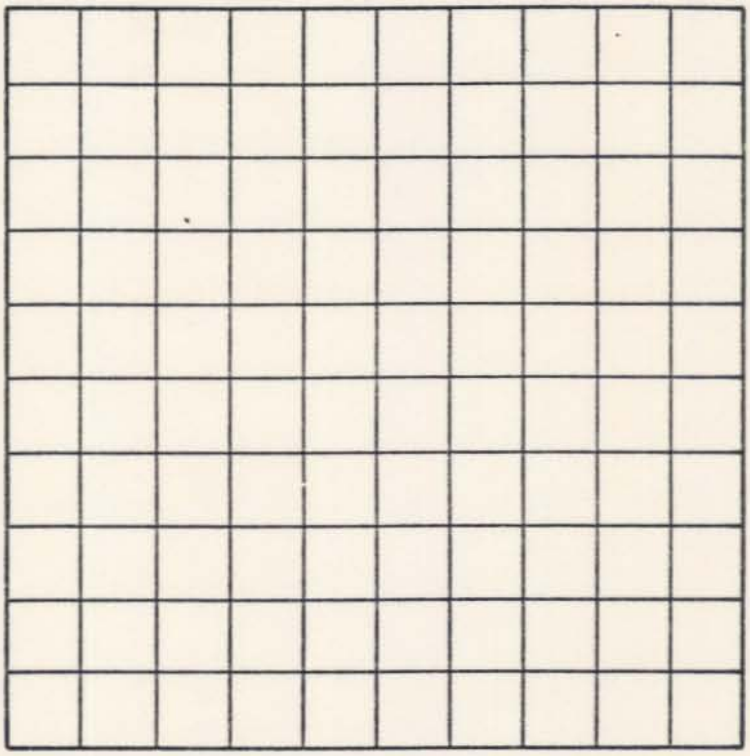

Figure 2

Overlay the grid onto the map as shown in Figure 3. Now the source and each observation well can be assigned to the center of the closest grid square. The location of each value will be referenced by an "X" and a "Y" value. You can arbitrarily set your origin (X=0, Y=0) in the lower left corner of the grid as shown in Figure 3. The source has been assigned to square (500, 1250) in Figure 3 (Over 2 squares in the X direction, up 5 squares in the Y direction). Observation well #1 has been assigned to square (1000, 1250); observation well #2 to (1500, 2000); observation well #3 to (1500, 1250); observation well #4 to (1500, 500); and observation well #5 to square (2250, 1250).

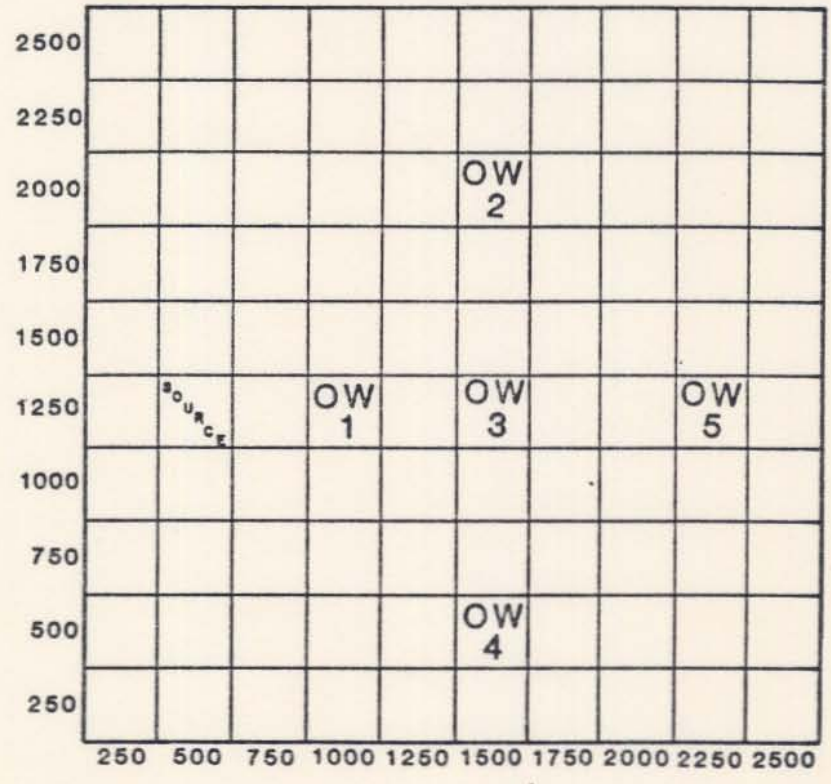

Figure 3

Once the grid map is aligned and coordinates are known for the source and sample locations, specifications can be provided for a grid map display. The map will represent the distribution of concentrations based on the location of the source and observation wells after data entry is completed. In order to generate the data necessary for the grid map display, the following PROMPTS will appear:

```
GRID MAP X LOCATIONS (FT)
MINIMUM, (MAXIMUM), (INTERVAL):
```

For the example used here, THE USER RESPONDS WITH (5 characters per variable):

```
?   0, 2500,  250
```

Next the Y dimensions are PROMPTED for. The following PROMPTS will be printed on the screen:

```
GRID MAP Y LOCATIONS (FT)
MINIMUM, MAXIMUM, (INTERVAL):
```

Again, for our example THE USER RESPONDS WITH (5 characters per variable):

```
?    0, 2500,  250
```

Now the program PROMPTS the user for information about the contaminant source. The following PROMPTS will appear from the main source menu:

```
****BEGIN SOURCE INFORMATION****
CURRENTLY USING 0 OF 10 TIME STEPS
0 TO END SOURCE INFORMATION
1 TO ADD POINT SOURCE
2 TO ADD NON-POINT SOURCE
-N TO DELETE LAST N TIME STEPS
```

For our example, THE USER RESPONDS WITH (5 characters per variable):

```
?    1
```

In order to preserve continuity in the sequential operation of the program, the source area and flow rate prompting routine shall be explained for a single point source. Multiple and non point source options will be explained later.

After choosing Option 1, the program will prompt for the location of the source and the source area (the approximate surface area that the contaminant source occupies). The following prompts will appear on the screen:

```
        LOCATION:
THE USER RESPONDS WITH (5 characters per variable):
        ?  500, 1250
        SOURCE AREA (FT2):
THE USER RESPONDS WITH (5 characters per variable):
        ? 1550
```

Next the program prompts the user for a flow rate. The rate can be calculated by using one of the four options shown below. Usually this parameter is unknown to the hydrogeologist. Therefore, this parameter is generally considered to be a variable. Four options are available for determining mass flow rate:

```
1 FOR MASS FLOW RATE (LBM/D),
2 FOR MASS/AREA RATE (LB/FT2/D),
3 FOR VOLUME FLOW RATE (FT3/D),
      AND CONCENTRATION (MG/L),
4 FOR VOLUME/AREA RATE (FT/D),
      AND CONCENTRATION (MG/L):
```

Option 1 MASS FLOW RATE, (LBM/DAY)

The following PROMPT will appear on the screen:

```
TIME, MASS FLOW RATE, (CONCENTRATION):
```

What is required of the user is a time (in days) at which the contaminant begins to flow through the system and the mass flow rate. When modeling only one source, set the time equal to 0. Do not let this confuse you. You are merely designating a reference point in time after which concentrations are to be observed. This concept is doubly

powerful. With one source being modeled, the contaminant can be introduced at time 0, and then again at specified time increments after 0 thus simulating slugs of contaminant flow in response to a series of recharge events.

After the time is requested, a mass flow rate and a concentration must follow (separated by commas). The mass flow rate is in pounds per day. (See Table II-1). Entering a concentration is optional which is indicated by parentheses in the prompt above. If a concentration is known and entered, then a volumetric flow rate will also be calculated. After this information is entered, the program is designed to prompt the user for the same information again. This involves the time step concept which will be discussed after all of the options have been explained. The mass flow rate data can also be entered by using one of the other options:

Option 2 MASS/AREA FLOW RATE (lB/FT2/D)

The following PROMPT will appear on the screen:

TIME, MASS/AREA RATE, (CONCENTRATION):

This option is similar to Option 1 except that a mass per area rate is prompted for rather than a mass rate. Again entering concentration is optional.

Option 3 VOLUME FLOW RATE (FT3/D and MG/L)

The following PROMPT will appear on the screen:

TIME, VOLUME FLOW RATE, CONCENTRATION:

Again, what is required of the user is similar to the previous options with the exception that the rate is now a volume flow rate. However, entering a concentration is mandatory.

Option 4 VOLUME/AREA FLOW RATE (FT/D and MG/L)

The following PROMPT will appear:

TIME, VOLUME/AREA RATE, CONCENTRATION:

This option is very similar to Option 3.

Now we will return to the main source menu and explain why and how each option is used in conjunction with the mass flow rate. The following is a list of the PROMPTS of the menu:

```
CURRENTLY USING 0 OF 10 TIME STEPS.
 0 TO END SOURCE INFORMATION.
 1 TO ADD POINT SOURCE,
 2 TO ADD NON-POINT SOURCE,
-N TO DELETE LAST N TIME STEPS:
```

Option 0 to end source information.

This option provides the means to stop entering source information and implies that the user wishes to proceed to the next step in the program.

Option 1 To add a point source

This option allows the user to add a contaminant source at a specific X, Y location. In the grid system the point source originates from a specific square whose location has been assigned by the user. Figure 4 is a simplified grid. For example, assume that two single point sources are located at (100,100) and (100,200) in Figure 4. Next, the program prompts the user for the source area. The source area may encompass the whole square or less than the square depending upon the scale of the grid set up by the user. After the source area is entered, the time, flow rate, and concentration is prompted for. If the contaminant source encompasses several adjacent squares and or parts of squares, then Option 2 should be used.

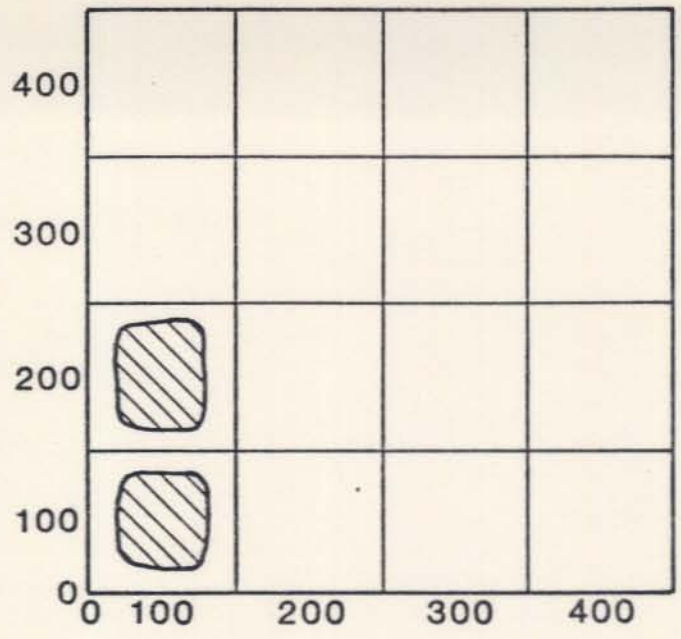

Figure 4

Option 2 To add a non-point source

The power of this option becomes evident when the area of the contamination source is very large with respect to the grid size. In the schematic grid system shown in Figure 5 the source originates from several squares and parts of squares.

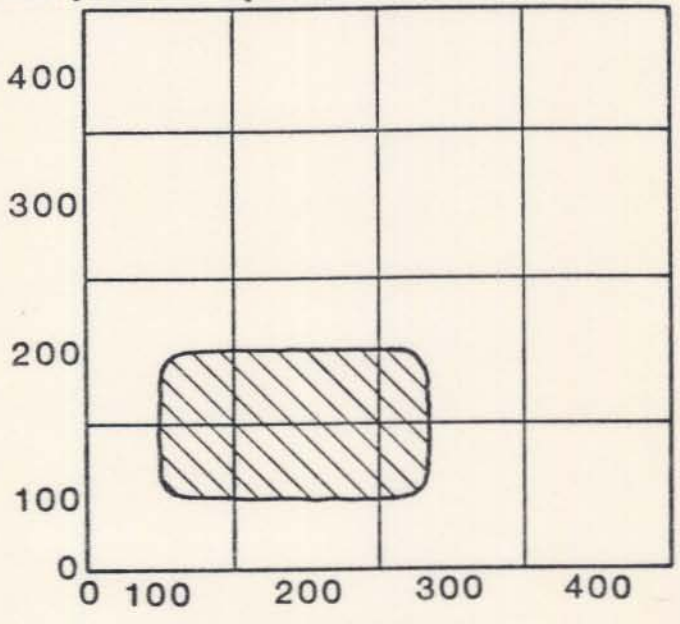

Figure 5

The following prompt will appear:

X LOCATION MINIMUM, MAXIMUM (FT):

For the schematic presented above THE USER WOULD RESPOND WITH (5 characters per variable):

? 50, 250

Next the Y location is prompted for. The following prompt will appear:

Y LOCATION, MINIMUM, MAXIMUM (FT):

For the schematic presented above THE USER WOULD RESPOND WITH (5 characters per variable):

? 50, 150

Again, the program prompts the user for a time, flow rate and concentration.

Option -N To delete last N time steps

An example will best illustrate the use of this option using several time steps. Let us initially model one point source. Let's assume that we have data for this example over a 20-year period. The contaminant has leached into the groundwater at an average rate of 52 pounds per day during the 20-year period. We also have observed concentrations at several observation wells located at various distances from the source. These data were gathered on an irregular time schedule over 20 years as shown in Table II-1 below. The concentration of the source is also shown for each time period.

Table II-1

Time (days)	Mass Flow rate (lbs/d)	Concentration (mg/l)	Time Step
0	52	100	1
365	52	110	2
548	52	130	3
730	52	160	4
1095	52	160	5
1643	52	180	6
2190	52	190	7
2920	52	200	8
4015	52	210	9
7300	52	250	10

The data is interpreted as follows. Initially, we observe a source concentration of 100 (mg/l). This will be our reference point in time, thus we designate it with a time = 0. This is our first time step. A year later we observe a concentration of 110 (mg/l). We have taken a second step in time which is equal to a period of 1 year. Thus our time step is #2. A year and a half from time = 0, we observe a concentration of 130 (mg/l). We have now taken our third step in time which is equal to a period of .5 years. Thus our time step is designated #3. Two years from our reference point in time we observed a concentration of 160 (mg/l). This is our fourth step in time with a period equal to .5 years. Three years later we again observe a concentration of 160 (mg/l). This is our fifth step in time with a period equal to 1 year. The remainder of the table follows in a similar manner.

In order to enter this data into the computer, Option 1 was selected for the multiple time step example. The following PROMPT will appear:

```
TIME, MASS FLOW RATE, (CONCENTRATION):
```

THE USER RESPONDS WITH (5 characters per variable):

```
?    0,   52,  100
```

The user enters the respective values and the program responds with another PROMPT:

```
ENTER 0 TO RETURN TO MAIN SOURCE MENU
TIME, MASS FLOW RATE, (CONCENTRATION):
```

THE USER RESPONDS WITH (5 characters per variable):

```
? 365,   52,  120
```

Suppose upon entering the data, the user enters an incorrect concentration at the next time step (120 instead of 110 while entering time step #2). To correct for this error, enter "0" to return to the

main source menu. The program responds with:

```
CURRENTLY USING 2 OF 10 TIME STEPS
 0 TO END SOURCE INFORMATION
 1 TO ADD POINT SOURCE
 2 TO ADD NON POINT SOURCE
-N TO DELETE LAST N TIME STEPS:
```

USER RESPONDS WITH (5 characters per variable):

```
?   -1
```

By entering -1 the information contained in the last time step (#2) is deleted.

The program responds with:

```
CURRENTLY USING 1 OF 10 TIME STEPS
 0 TO END SOURCE INFORMATION
 1 TO ADD POINT SOURCE
 2 TO ADD NON POINT SOURCE
-N TO DELETE LAST N TIME STEPS:
```

THE USER RESPONDS WITH (5 characters per variable):

```
?    1
```

In order to continue entering the data, the user must designate the point source again for time step 2 and subsequent time steps. Once this option has been entered, the user has returned to the time, flow rate, concentration mode for entering time step 2 data. When all the data has been entered, the program responds with:

```
ENTER 0 TO RETURN TO MAIN SOURCE MENU
TIME, MASS FLOW RATE, CONCENTRATION
```

THE USER RESPONDS WITH (5 characters per variable):

```
?    0
```

The program responds with:

```
CURRENTLY USING 10 OF 10 TIME STEPS
 0 TO END SOURCE INFORMATION
 1 TO ADD POINT SOURCE
 2 TO ADD NON-POINT SOURCE
-N TO DELETE LAST N TIME STEPS:
```

The above prompt confirms that the mass flow rate source has been

entered for all 10 time steps. In order to end source information THE USER RESPONDS WITH (5 characters per variable):

? 0

The program informs the user of this by the following statement:

****END SOURCE INFORMATION***

Next, the program prompts for a sample time:

SAMPLE TIME (DAYS)

THE USER RESPONDS WITH (5 characters per variable):

? 2330

The sample time is a point in time, after time 0, at which concentrations are to be computed and plotted on the grid. However, these concentrations will approach equilbrium (steady state) when the concentrations remain constant over time at a specific location on the grid. This is referred to as a "steady state" condition. If the user wishes to compute the concentrations throughout the grid under this condition, THE USER RESPONDS WITH (5 characters per variable):

? -1

The last prompts are for retardation, decay, and dispersion. Usually, the user lacks information about these parameters. Suppose the user has values for retardation and decay only.

The retardation PROMPT:

RETARDATION (UNITLESS)

THE USER RESPONDS WITH:

? 1

This indicates that the source is not "retarded" or influenced in any way by sorption on surfaces within the

earth material as it travels through the system.

The PROMPT for decay information is next:

0 FOR NO DECAY
1 FOR DECAY COEFFICIENT (GG)
2 FOR DECAY LAMBDA (GL)
3 FOR DECAY HALF-LIFE (GT)

Initially THE USER SHOULD RESPOND WITH (5 characters per variable):

? 0

Now the user has reduced the number of unknowns to 1 variable, dispersion. This parameter controls the shape of the plume. In other words, what is the rate at which the contaminant disperses in the X direction versus the rate it disperses in the Y direction? If the rates are the same then this ratio of X:Y (the dispersion ratio) equals 1. The user has 4 options for entering dispersion values depending upon the information available. The following is the list of prompt options for dispersion:

1 FOR X AND Y DISPERSION (DX, DY),
2 FOR X DISPERSION AND DISPERSION RATIO (DX, DR),
3 FOR X AND Y DISPERSIVITY (AX, AY),
4 FOR X DISPERSIVITY AND DISPERSION RATIO (AX, DR),
(USE DM FOR MOLECULAR DIFFUSION)

In order to consider these options (1-4), the user should refer to Table I-1 for the definition of dispersion. The options are described as follows:

Option 1 Allows the user to enter values for dispersion in the X and Y directions. The dispersion ratio and the X and Y dispersivities will be calculated for you.

Option 2 Allows the user to enter values for dispersion in the X direction and the dispersion ratio. Dispersion in the Y

direction and the X and Y dispersivities will be calculated for the user.

Option 3 Allows the user to enter the X and Y dispersivities. The X and Y dispersions and the dispersion ratio will be calculated automatically.

Option 4 Allows the user to enter the X dispersivity and the dispersion ratio. The Y dispersivity and the X and Y dispersions are calculated for the user.

If the dimensions of an existing plume can be measured, then this ratio of X and Y using Option 1 can be used as a starting point in order to generate an initial value for the dispersion ratio. The program responds with:

```
1 FOR X AND Y DISPERSION (DX, DY)
2 FOR X DISPERSION AND DISPERSION RATIO (DX,DR)
3 FOR X AND Y DISPERSIVITY (AX,AY)
4 FOR X DISPERSIVITY AND DISPERSION RATIO (AX,DR)
      (USE DM FOR MOLECULAR DIFFUSION)
```

THE USER RESPONDS WITH (5 characters per variable):

```
?   1
```

THE PROGRAM RESPONDS WITH;

```
X DISPERSION          (FT2/D)
```

THE USER RESPONDS WITH (5 characters per variable):

```
Y DISPERSION          (FT2/D)
```

THE USER RESPONDS WITH (5 characters per variable):

press return. The Display commands are used to reproduce the values of the data which you have entered, or to produce the solution as a single point concentration or as the grid map in Figure 3 showing all of the concentrations. The Display commands are:

.C	Display a single point (x,y) concentration
.D	Display all parameters
.G	Display grid map concentrations
.I	Display input parameters
.DG	Display all parameters and grid map concentrations
.T	Display of Steady State Time
.TG	Display grid map of Steady State Times

If the user enters .D, all parameters and values are displayed as entered for the single point example. The display is shown in Appendix A. Once displayed, the user can enter .G which will display grid map of concentrations for the solution (See Appendix A). Once the solution has been completed, the user can store the data. In order to store the data used in this example, the "OD" command is used. The following response will appear on the screen:

DUMP FILE UNIT (6 TO 10):

This means that a total of 5 different sets of data may be saved (numbered 6,7,8,9.and 10). The user must now assign a number (either 6,7,8,9, or 10) to the data set. Once the number is chosen, (i.e. 6), the input data is copied to the disk under the Fortran file name with that number.

At this point, the following PROMPT appears:

COMMAND

The user now has three choices; end the program by entering E; initiate a new problem by using the COMMAND I to recall the list of PROMPT options; or change selected parameters by using an EDIT command. These can be listed by

entering the HELP COMMAND, H. Some of the more useful edit commands are listed:

D	Dispersion
L	Grid Limits
M	Aquifer Thickness
P	Porosity
R	Retardation
V	Ground water velocity
Q	Source
QM	Source mass rate only
TE	Steady State Time

For example, the user will commonly want to make changes in the source term (Q), Dispersion (D) or Retardation (R).

Suppose the user ends the program using the COMMAND E and returns later, wanting to modify the data saved in data set 6. When the program begins, the initial PROMPT appears:

```
  1  TO PROMPT FOR ALL REQUIRED PARAMATERS (IP),
  2  TO LOAD PREVIOUSLY SAVED PARAMETERS (IL),
  3  TO READ COMMANDS FROM ANOTHER SOURCE (IR),
 -1  TO SET OUTPUT PARAMETERS (O)
```

To load information in data set 6, THE USER SHOULD RESPOND WITH (5 characters per variable):

```
?    2
```

THE PROGRAM RESPONSE IS:

```
LOAD FILE UNIT (6 TO 10):
```

The user responds with the data set number to be loaded. In this example, THE USER RESPONDS WITH (5 characters per variable):

```
?    6
```

The input parameters that had been previously saved in data set 6 are now loaded into the program.

THE PROGRAM RESPONDS WITH:

```
COMMAND
```

IN ORDER TO REVIEW THE DATA IN DATA SET 6, THE USER RESPONDS WITH (5 characaters per variable):

```
?    .I
```

Once the data set is loaded from the file, the concentrations at the observation wells can be matched to those of the computer run and the accuracy of the dispersion ratio can be tested.

The dispersion can be changed by entering the EDIT command D. One of the four options to enter dispersion is to be selected. The user will enter the new dispersion parameters indicated in the PROMPT. By varying the dispersion ratio, a "best fit" can be attempted. The process of matching computed and observed data is referred to as calibration. The procedure for changing one variable while all others are held constant is referred to as sensitivity analysis.

Once the general shape of the plume is achieved by adjusting the dispersion ratio, the velocity or retardation can be varied in order to shorten or lengthen the plume. An example of a sensitivity analysis run of retardation for calibration of dispersion is shown in Appendix B. Retardation must be greater than 1 in order to represent the effects of sorption phenomena. Let us suppose that the concentrations at X = 1750, Y = 750,100,1250, are all slightly higher than "actual" concentrations. By increasing the retardation, these values will be lowered for X = 1750. If values should be increased, then velocity should be increased instead. In the example, in Appendix B, retardation was changed from 1 to 1.3. The resulting concentrations at X = 1750, Y = 750,100,1250 have all slightly lowered in value. Retardation effects become more pronounced after a certain traveled distance and time.

Once the changes in Dispersion and Retardation are complete, the user will use the Display (.D, .C, .G, .DG) COMMANDS in order to

view the new solution on the screen. The user can save the new data for the corresponding solution by using the OD or I COMMANDS as described earlier.

The user may decide to begin a new problem, but will introduce several contaminant sources. The parameters and grid used in Figure 3 and Appendix A can also be used here except for the addition of another source using the main source menu. The step by step procedure and output are shown in Appendix C.

There are two point sources located at X = 500, Y = 500, and X = 500, Y = 1250 (See Figure 3 for grid location). The hydrologic parameters are identical, with the exception of one important change. Notice that the number of time steps (i.e. the number of start times for each source) is now 5. A total of 10 time steps and/or sources can be used. By using two sources, only 5 time steps are available. Combining both sources is essentially the same as over-laying the grid solution for each of the individual plumes on one another and adding the values which appear in the same grid square. The plumes begin to overlap at X =250, Y = 1250, to 1500. This feature is especially powerful and time saving.

Alternatively, the user may choose a non point source as shown in Figure 5 (large source area) rather than a multiple point source. To do so, Option 2 of the main source menu was used. The step by step procedure and output are shown in Appendix D.

STEADY-STATE

In certain instances, it may be desirable to determine both the concentration distribution of the plume after it has achieved equilibrium with its surroundings and the time required to reach that equilibrium. This time is referred to as "steady-state". The FPLUME model allows computation of these steady-state concentrations in the following manner:

(1) In response to the programs' "COMMAND?" prompt, a "T" is keyed in.

(2) The program then prompts for a time to be input. A value of "-1" (for steady-state) is keyed in.

(3) The program then prompts for a "COMMAND?". At this point the user can (a) print out the map of the plume using ".G" command or, (b) determine the time (in days) required for the system to reach equilibrium using the "TE" command.

When the "TE" command is used, the program prompts for input of "% STEADY STATE (UNITLESS):"? The input value should be between 1 and 100%. The higher the percentage, the more accurate the computation of the time required to achieve steady-state. Normally, 90% or 95% is used. After a value has been keyed in, the program again displays the "COMMAND?" prompt. The user then keys in ".T", and the program then prompts for the X and Y coordinates of an observation point. Once the desired coordinates have been entered, the program echo-prints the % steady-state, the X location, the Y location and the time (in days) required for the system to reach steady-state. The ".TG" command prints a grip map of the time (in days) required to reach the steady concentrations. The "S" or "SC" commands may be used to adjust the grid multiplier so that the values on the map are not written in exponential form.

Table of Contents
for Appendices

APPENDIX II-A

SINGLE POINT SOURCE

```
COMMAND>.D

SINGLE
POINT
SOURCE

THICKNESS        =  110.000      FT
POROSITY         =  .350000
VELOCITY         =  1.50000      FT/D

X DISPERSION     =  105.000      FT2/D
Y DISPERSION     =  21.0000      FT2/D
X DISPERSIVITY   =  70.0000      FT
Y DISPERSIVITY   =  14.0000      FT
DISPERSION RATIO=  5.00000
MOL. DIFFUSION   =  0.00000      FT2/D

RETARDATION      = 1.000000
DECAY GAMMA      = 1.000000
DECAY LAMBDA     =  0.00000      1/YR

ACCURACY         =  .100000E+00
MAXIMUM DIVISION=       100

X, Y LOCATION    =  500.000      ,    1250.00     FT
SOURCE AREA      =  1550.00      FT2
```

START TIME (DAYS)	VOLUME/ AREA RATE (FT/D)	VOLUME FLOW RATE (FT3/D)	SOURCE CONCENTR. (MG/L)	MASS/AREA RATE (LB/FT2/D)	MASS FLOW RATE (LBM/D)
0.00000	5.37394	8329.60	100.0000	.335484E-01	52.0000
365.000	4.88540	7572.36	110.000	.335484E-01	52.0000
548.000	4.13380	6407.39	130.000	.335484E-01	52.0000
730.000	3.35871	5206.00	160.000	.335484E-01	52.0000
1095.00	3.35871	5206.00	160.000	.335484E-01	52.0000
1641.00	2.98552	4627.56	180.000	.335484E-01	52.0000
2190.00	2.82839	4384.00	190.000	.335484E-01	52.0000
2920.00	2.68697	4164.80	200.000	.335484E-01	52.0000
4015.00	2.55902	3966.48	210.000	.335484E-01	52.0000
7300.00	2.14957	3331.84	250.000	.335484E-01	52.0000

```
COMMAND>.G

SAMPLE TIME      =  2333.30      DAYS
X SCALE          ( 1.000000      FT      )
Y SCALE          ( 1.000000      FT      )
CONCENTRATION    ( 1.000000      MG/L    )
```

Y \ X	0	250	500	750	1000	1250	1500	1750	2000	2250	2500
2500	0	0	0	0	0	0	0	0	0	0	0
2250	0	0	0	0	0	0	0	0	0	0	0
2000	0	0	0	0	0	0	0	0	0	0	0
1750	0	0	0	0	0	0	1	1	2	2	3
1500	0	0	1	3	7	9	11	13	13	13	14
1250	0	2	-1	69	49	40	34	31	29	26	24
1000	0	0	1	3	7	9	11	13	13	13	14
750	0	0	0	0	0	0	1	1	2	2	3
500	0	0	0	0	0	0	0	0	0	0	0
250	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

```
WORST APPROXIMATION = +- 6.66    %.
   1 SOURCE(S) SHOWN AS "-1".
```

APPENDIX II-B

SINGLE POINT SOURCE

SENSITIVITY ANALYSIS

APPENDIX II-C

TWO POINT SOURCES

```
COMMAND>.DG

SINGLE POINT SOURCE
SENSITIVITY ANALYSIS
DISPERSION

THICKNESS        =  110.000      FT
POROSITY         =  .350000
VELOCITY         =  1.50000      FT/D

X DISPERSION     =  105.000      FT2/D
Y DISPERSION     =  21.0000      FT2/D
X DISPERSIVITY   =  70.0000      FT
Y DISPERSIVITY   =  14.0000      FT
DISPERSION RATIO=  5.00000
MOL. DIFFUSION   =  0.00000      FT2/D

RETARDATION      =  1.30000
DECAY GAMMA      = 1.000000
DECAY LAMBDA     =  0.00000      1/YR

ACCURACY         =  .100000E+00
MAXIMUM DIVISION=       100

X, Y LOCATION    =  500.000      ,     1250.00      FT
SOURCE AREA      =  1550.00      FT2
```

START TIME (DAYS)	VOLUME/ AREA RATE (FT/D)	VOLUME FLOW RATE (FT3/D)	SOURCE CONCENTR. (MG/L)	MASS/AREA RATE (LB/FT2/D)	MASS FLOW RATE (LBM/D)
0.00000	5.37394	8329.60	100.0000	.335484E-01	52.0000
365.000	4.88540	7572.36	110.000	.335484E-01	52.0000
548.000	4.13380	6407.39	130.000	.335484E-01	52.0000
730.000	3.35871	5206.00	160.000	.335484E-01	52.0000
1095.00	3.35871	5206.00	160.000	.335484E-01	52.0000
1641.00	2.98552	4627.56	180.000	.335484E-01	52.0000
2190.00	2.82839	4384.00	190.000	.335484E-01	52.0000
2920.00	2.68697	4164.80	200.000	.335484E-01	52.0000
4015.00	2.55902	3966.48	210.000	.335484E-01	52.0000
7300.00	2.14957	3331.84	250.000	.335484E-01	52.0000

```
SAMPLE TIME      =  2333.30      DAYS
X SCALE          ( 1.000000      FT      )
Y SCALE          ( 1.000000      FT      )
CONCENTRATION    ( 1.000000      MG/L    )
```

Y \ X	0	250	500	750	1000	1250	1500	1750	2000	2250	2500
2500	0	0	0	0	0	0	0	0	0	0	0
2250	0	0	0	0	0	0	0	0	0	0	0
2000	0	0	0	0	0	0	0	0	0	0	0
1750	0	0	0	0	0	0	1	1	2	2	2
1500	0	0	1	3	7	9	11	12	13	13	12
1250	0	2	-1	69	49	40	34	30	27	24	21
1000	0	0	1	3	7	9	11	12	13	13	12
750	0	0	0	0	0	0	1	1	2	2	2
500	0	0	0	0	0	0	0	0	0	0	0
250	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

```
WORST APPROXIMATION = +- 6.66    %.
   1 SOURCE(S) SHOWN AS "-1".
```

```
GROUNDWATER PLUME CALCULATION PROGRAM
D.C. KENT, HYDROGEOLOGIST
FRED WITZ, PROGRAMMER
GEOLOGY DEPARTMENT, OKLAHOMA STATE UNIVERSITY
FORTRAN VERSION 1.0 (1983, MAY)


 1 TO PROMPT FOR ALL REQUIRED PARAMETERS (IP),
 2 TO LOAD PREVIOUSLY SAVED PARAMETERS (IL),
 3 TO READ COMMANDS FROM ANOTHER SOURCE (IR),
-1 TO SET OUTPUT PARAMETERS (O):
?1

THREE TITLE LINES:
?TWO

?POINT

?SOURCES

THICKNESS          (FT       ):
?110

POROSITY           (UNITLESS):
?.35

VELOCITY           (FT/D     ):
?1.5

GRID MAP X LOCATIONS (FT       ),
MINIMUM, (MAXIMUM), (INTERVAL):
?0,2500,250

GRID MAP Y LOCATIONS (FT       ),
MINIMUM, MAXIMUM, (INTERVAL):
?0,2500,250

**** BEGIN SOURCE INFORMATION ****
CURRENTLY USING   0 OF  10 TIME STEPS.
 0 TO END SOURCE INFORMATION,
 1 TO ADD POINT SOURCE,
 2 TO ADD NON-POINT SOURCE,
-N TO DELETE LAST N TIME STEPS:
?1

X LOCATION, Y LOCATION (FT       ):
?500,1000

SOURCE AREA        (FT2      ):
?1550

1 FOR MASS FLOW RATE     (LBM/D    ),
2 FOR MASS/AREA RATE     (LB/FT2/D),
3 FOR VOLUME FLOW RATE   (FT3/D    ),
      AND CONCENTRATION  (MG/L     ),
4 FOR VOLUME/AREA RATE   (FT/D     ),
      AND CONCENTRATION  (MG/L     ):
?1

TIME, MASS FLOW RATE, (CONCENTRATION):
?0,52,100

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?730,52,160

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?2190,52,190

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?4015,52,210

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?7300,52,250

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?0

CURRENTLY USING   5 OF  10 TIME STEPS.
 0 TO END SOURCE INFORMATION,
 1 TO ADD POINT SOURCE,
 2 TO ADD NON-POINT SOURCE,
-N TO DELETE LAST N TIME STEPS:
?1

X LOCATION, Y LOCATION (FT       ):
?500,1750

SOURCE AREA        (FT2      ):
?1550

1 FOR MASS FLOW RATE     (LBM/D    ),
2 FOR MASS/AREA RATE     (LB/FT2/D),
3 FOR VOLUME FLOW RATE   (FT3/D    ),
      AND CONCENTRATION  (MG/L     ),
4 FOR VOLUME/AREA RATE   (FT/D     ),
      AND CONCENTRATION  (MG/L     ):
?1
```

```
TIME, MASS FLOW RATE, (CONCENTRATION):
?0,52,100

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
??30,52,160

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?2190,52,190

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?4015,52,210

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
??300,52,250

ENTER    0 TO RETURN TO MAIN SOURCE MENU:
TIME, MASS FLOW RATE, (CONCENTRATION):
?0

CURRENTLY USING  10 OF  10 TIME STEPS.
 0 TO END SOURCE INFORMATION,
 1 TO ADD POINT SOURCE,
 2 TO ADD NON-POINT SOURCE,
-N TO DELETE LAST N TIME STEPS:
?-10

**** END SOURCE INFORMATION ****
-1 FOR STEADY STATE,
SAMPLE TIME      (DAYS    ):
?-1

RETARDATION      (UNITLESS):
?1

0 FOR NO DECAY,
1 FOR DECAY COEFFICIENT, GAMMA (GG),
2 FOR DECAY LAMBDA (GL),
3 FOR DECAY HALF-LIFE (GT):
?0

1 FOR X AND Y DISPERSION (DX, DY),
2 FOR X DISPERSION AND DISPERSION RATIO (DX, DR),
3 FOR X AND Y DISPERSIVITY (AX, AY),
4 FOR X DISPERSIVITY AND DISPERSION RATIO (AX, DR),
  (USE DM FOR MOLECULAR DIFFUSION):
?2

X DISPERSION     (FT2/D   ):
?105

DISPERSION RATIO (UNITLESS):
?5

COMMAND?
```

```
.DG

TWO
POINT
SOURCES

THICKNESS        =  110.000      FT
POROSITY         =  .350000
VELOCITY         =  1.50000      FT/D

X DISPERSION     =  105.000      FT2/D
Y DISPERSION     =  21.0000      FT2/D
X DISPERSIVITY   =  70.0000      FT
Y DISPERSIVITY   =  14.0000      FT
DISPERSION RATIO=  5.00000
MOL. DIFFUSION   =  0.00000      FT2/D

RETARDATION      = 1.000000
DECAY GAMMA      = 1.000000
DECAY LAMBDA     =  0.00000      1/YR

ACCURACY         =  .100000E+00
MAXIMUM DIVISION=       100

X, Y LOCATION    =  500.000      ,   1000.000     FT
SOURCE AREA      =  1550.00      FT2
  START          VOLUME/         VOLUME          SOURCE          MASS/AREA       MASS FLOW
  TIME           AREA RATE       FLOW RATE       CONCENTR.       RATE            RATE
  (DAYS     )    (FT/D     )     (FT3/D    )     (MG/L     )     (LB/FT2/D)      (LBM/D    )
  0.00000        5.37394         8329.60         100.0000        .335484E-01     52.0000
  730.000        3.35871         5206.00         160.000         .335484E-01     52.0000
  2190.00        2.82839         4384.00         190.000         .335484E-01     52.0000
  4015.00        2.55902         3966.48         210.000         .335484E-01     52.0000
  7300.00        2.14957         3331.84         250.000         .335484E-01     52.0000

X, Y LOCATION    =  500.000      ,   1750.00      FT
SOURCE AREA      =  1550.00      FT2
  START          VOLUME/         VOLUME          SOURCE          MASS/AREA       MASS FLOW
  TIME           AREA RATE       FLOW RATE       CONCENTR.       RATE            RATE
  (DAYS     )    (FT/D     )     (FT3/D    )     (MG/L     )     (LB/FT2/D)      (LBM/D    )
  0.00000        5.37394         8329.60         100.0000        .335484E-01     52.0000
  730.000        3.35871         5206.00         160.000         .335484E-01     52.0000
  2190.00        2.82839         4384.00         190.000         .335484E-01     52.0000
  4015.00        2.55902         3966.48         210.000         .335484E-01     52.0000
  7300.00        2.14957         3331.84         250.000         .335484E-01     52.0000

SAMPLE TIME      = STEADY STATE
X SCALE          ( 1.000000      FT      )
Y SCALE          ( 1.000000      FT      )
CONCENTRATION    ( 1.000000      MG/L    )

          X
    Y     0   250   500   750  1000  1250  1500  1750  2000  2250  2500

 2500     0     0     0     0     0     0     0     0     0     0     0

 2250     0     0     0     0     0     0     1     1     2     2     3

 2000     0     0     1     3     7     9    11    13    13    14    14

 1750     0     2    -1    69    49    40    34    31    28    26    25

 1500     0     0     1     3     7    10    12    14    15    16    17

 1250     0     0     1     3     7    10    12    14    15    16    17

 1000     0     2    -1    69    49    40    34    31    28    26    25

  750     0     0     1     3     7     9    11    13    13    14    14

  500     0     0     0     0     0     0     1     1     2     2     3

  250     0     0     0     0     0     0     0     0     0     0     0

    0     0     0     0     0     0     0     0     0     0     0     0

WORST APPROXIMATION = +- 1.06    %.
   2 SOURCE(S) SHOWN AS "-1".


COMMAND?E

END OF PLUME PROGRAM.
GOODBYE. STOP
```

APPENDIX II-D

NON POINT SOURCE

```
GROUNDWATER PLUME CALCULATION PROGRAM
D.C. KENT, HYDROGEOLOGIST
FRED WITZ, PROGRAMMER
GEOLOGY DEPARTMENT, OKLAHOMA STATE UNIVERSITY
FORTRAN VERSION 1.0 (1983, MAY)


 1 TO PROMPT FOR ALL REQUIRED PARAMETERS (IP),
 2 TO LOAD PREVIOUSLY SAVED PARAMETERS (IL),
 3 TO READ COMMANDS FROM ANOTHER SOURCE (IR),
-1 TO SET OUTPUT PARAMETERS (O):
?1

THREE TITLE LINES:
?NON

?POINT

?SOURCE

THICKNESS          (FT       ):
?110

POROSITY           (UNITLESS):
?.35

VELOCITY           (FT/D     ):
?1.5

GRID MAP X LOCATIONS (FT       ),
MINIMUM, (MAXIMUM), (INTERVAL):
?0,2500,250

GRID MAP Y LOCATIONS (FT       ),
MINIMUM, MAXIMUM, (INTERVAL):
?0,2500,250

**** BEGIN SOURCE INFORMATION ****
CURRENTLY USING   0 OF  10 TIME STEPS.
 0 TO END SOURCE INFORMATION,
 1 TO ADD POINT SOURCE,
 2 TO ADD NON-POINT SOURCE,
-N TO DELETE LAST N TIME STEPS:
?2

X LOCATION MINIMUM, MAXIMUM (FT       ):
?350,800

Y LOCATION MINIMUM, MAXIMUM (FT       ):
?1000,1300

1 FOR MASS FLOW RATE     (LBM/D    ),
2 FOR MASS/AREA RATE     (LB/FT2/D),
3 FOR VOLUME FLOW RATE   (FT3/D    ),
      AND CONCENTRATION  (MG/L     ),
4 FOR VOLUME/AREA RATE   (FT/D     ),
      AND CONCENTRATION  (MG/L     ):
?1

TIME, MASS FLOW RATE, (CONCENTRATION):
?0,52,200

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?730,52,300

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?2190,52,400

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?4015,52,425

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?7300,52,500

ENTER    0 TO RETURN TO MAIN SOURCE MENU;
TIME, MASS FLOW RATE, (CONCENTRATION):
?0

CURRENTLY USING   5 OF  10 TIME STEPS.
 0 TO END SOURCE INFORMATION,
 1 TO ADD POINT SOURCE,
 2 TO ADD NON-POINT SOURCE,
-N TO DELETE LAST N TIME STEPS:
?0

**** END SOURCE INFORMATION ****
-1 FOR STEADY STATE,
SAMPLE TIME        (DAYS     ):
?-1

RETARDATION        (UNITLESS):
?1

0 FOR NO DECAY,
1 FOR DECAY COEFFICIENT, GAMMA (GG),
2 FOR DECAY LAMBDA (GL),
3 FOR DECAY HALF-LIFE (GT):
?0

1 FOR X AND Y DISPERSION (DX, DY),
2 FOR X DISPERSION AND DISPERSION RATIO (DX, DR),
3 FOR X AND Y DISPERSIVITY (AX, AY),
4 FOR X DISPERSIVITY AND DISPERSION RATIO (AX, DR),
  (USE DM FOR MOLECULAR DIFFUSION):
?2

X DISPERSION       (FT2/D    ):
?105

DISPERSION RATIO (UNITLESS):
?5
```

```
COMMAND?.DG

NON
POINT
SOURCE

THICKNESS        =  110.000      FT
POROSITY         =  .350000
VELOCITY         =  1.50000      FT/D

X DISPERSION     =  105.000      FT2/D
Y DISPERSION     =  21.0000      FT2/D
X DISPERSIVITY   =  70.0000      FT
Y DISPERSIVITY   =  14.0000      FT
DISPERSION RATIO=  5.00000
MOL. DIFFUSION   =  0.00000      FT2/D

RETARDATION      = 1.000000
DECAY GAMMA      = 1.000000
DECAY LAMBDA     =  0.00000      1/YR

ACCURACY         =  .100000E+00
MAXIMUM DIVISION=       100

X LOCATION       =  350.000      TO   800.000      FT
Y LOCATION       = 1000.000      TO   1300.00      FT
SOURCE AREA      =  135000.      FT2
```

START TIME (DAYS)	VOLUME/ AREA RATE (FT/D)	VOLUME FLOW RATE (FT3/D)	SOURCE CONCENTR. (MG/L)	MASS/AREA RATE (LB/FT2/D)	MASS FLOW RATE (LBM/D)
0.00000	.308504E-01	4164.80	200.000	.385185E-03	52.0000
730.000	.205669E-01	2776.53	300.000	.385185E-03	52.0000
2190.00	.154252E-01	2082.40	400.000	.385185E-03	52.0000
4015.00	.145178E-01	1959.91	425.000	.385185E-03	52.0000
7300.00	.123401E-01	1665.92	500.000	.385185E-03	52.0000

```
SAMPLE TIME      = STEADY STATE
X SCALE          ( 1.000000      FT       )
Y SCALE          ( 1.000000      FT       )
CONCENTRATION    ( 1.000000      MG/L     )
```

Y \ X	0	250	500	750	1000	1250	1500	1750	2000	2250	2500
2500	0	0	0	0	0	0	0	0	0	0	0
2250	0	0	0	0	0	0	0	0	0	0	0
2000	0	0	0	0	0	0	0	0	0	0	0
1750	0	0	0	0	0	0	0	0	1	1	1
1500	0	0	0	1	2	4	5	6	7	8	8
1250	0	1	-1	-1	33	30	27	26	24	23	23
1000	0	1	-1	-1	24	23	23	22	21	20	20
750	0	0	0	0	1	2	3	4	5	6	6
500	0	0	0	0	0	0	0	0	0	1	1
250	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

```
WORST APPROXIMATION = +- 8.00    %.
    4 SOURCE(S) SHOWN AS '-1'.
```

APPENDIX II-E

PROGRAM SOURCE FOR KAYPRO 2

```
**** TSO FOREGROUND HARDCOPY ****
DSNAME=U11236C.FPLUME.CNTL

C       GROUNDWATER PLUME CALCULATION PROGRAM                                  00000010
C       D.C. KENT, HYDROGEOLOGIST, PRINCIPAL INVESTIGATOR                      00000020
C       FRED WITZ, PROGRAMMER                                                  00000030
C         GEOLOGY DEPARTMENT, OKLAHOMA STATE UNIVERSITY                        00000040
C         STILLWATER, OKLAHOMA, 74078                                          00000050
C       FORTRAN VERSION (SEE VERSION BELOW)                                    00000060
C       TESTED WITH:                                                           00000070
C         MICROSOFT FORTRAN ON KAYPRO II (66 STANDARD)                         00000080
C       ================================ DECLARATIONS ==================00000090
        LOGICAL*1      BERR, BATCH,BOP , BOE ,BTSS                             00000100
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                        00000110
        LOGICAL*1      BBT , B   , BIP                                         00000120
        INTEGER*4                                                              00000130
     +    HHCMD                                                               00000140
     +  , H   , HA  , HAR , HHAX, HHAY                                        00000150
     +  , HC  , HD  , HHDX, HHDY, HHDR, HHDM                                  00000160
     +  , HE  , HF  , HG  , HHGG, HHGL, HHGT                                  00000170
     +  , HH  , HI  , HIP , HIL , HIR                                         00000180
        INTEGER*4                                                              00000190
     +    HL  , HLX , HLY , HM                                                00000200
     +  , HO  , HOD , HOP , HOE , HOT , HOW , HHP                            00000210
     +  , HQ  , HQN , HHQE, HHQL, HHQA, HHQT                                  00000220
     +  , HHQM, HHQR, HHQC, HHQQ, HHQV                                        00000230
     +  , HR  , HHRR, HHRK, HHRD                                              00000240
        INTEGER*4                                                              00000250
     +    HS  , HSC , HSL , HSN , HSX , HSY , HHTC , HHTE                     00000260
     +  , HU  , HHU , HHUL, HHUM, HHUT                                        00000270
     +  , HV  , HHVV, HHVG, HHVK, HHVT                                        00000280
     +  , HHVR, HHVI, HHVD, HHVU                                              00000290
     +  , HXY , HHXC, HHYC, HHZM, HH9C                                        00000300
     +  , HH9T                                                                00000310
        INTEGER*4                                                              00000320
     +    H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                             00000330
     +  , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                                  00000340
     +  , HHC1, HHC2, HHC3                                                    00000350
        INTEGER*4      HHCOM,HCMD, H6  , H7                                    00000360
        INTEGER*4      JJ  , J   , JY                                          00000370
        DIMENSION BBT(8), HHCOM(381), KKCOM(19), RRCOM(79), JJ(30)             00000380
        COMMON /CHAR/                                                          00000390
     +    HHCMD(30)                                                           00000400
     +  , H   , HA  , HAR       , HHAX  (7) , HHAY  (7)                      00000410
     +  , HC  , HD  , HHDX  (7) , HHDY  (7) , HHDR  (7) , HHDM  (7)          00000420
     +  , HE  , HF  , HG        , HHGG  (7) , HHGL  (7) , HHGT  (7)          00000430
     +  , HH  , HI  , HIP , HIL , HIR                                        00000440
     +  , HL  , HLX , HLY , HM                                               00000450
     +  , HO  , HOD , HOP , HOE , HOT , HOW , HHP   (7)                      00000460
     +  , HQ  , HQN , HHQE  (7) , HHQL  (7) , HHQA  (7) , HHQT  (7)          00000470
     +  , HHQM  (7) , HHQR  (7) , HHQC  (7) , HHQQ  (7) , HHQV  (7)          00000480
     +  , HR        , HHRR  (7) , HHRK  (7) , HHRD  (7)                      00000490
     +  , HS  , HSC , HSL , HSN , HSX , HSY , HHTC  (7) , HHTE  (7)          00000500
     +  , HU        , HHU   (7) , HHUL  (7) , HHUM  (7) , HHUT  (7)          00000510
     +  , HV        , HHVV  (7) , HHVG  (7) , HHVK  (7) , HHVT  (7)          00000520
     +  , HHVR  (7) , HHVI  (7) , HHVD  (7) , HHVU  (7)                      00000530
     +  , HXY       , HHXC  (7) , HHYC  (7) , HHZM  (7) , HH9C  (7)          00000540
     +  , HH9T  (7)                                                          00000550
     +  , H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                            00000560
     +  , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                                 00000570
     +  , HHC1 (18) , HHC2 (18) , HHC3 (18)                                  00000580
        COMMON /BIT /                                                          00000590
     +    BERR, BATCH                                                         00000600
     +  , BOP , BOE ,BTSS                                                     00000610
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                      00000620
```

```
      COMMON /INT /                                                     00000630
     +   LUL , LUM , LUR , LUP , LUE , LUW                              00000640
     +  , MCMD                                                          00000650
     +  , KDX , KDY , KG  , KLX , KLY , KR                            00000660
     +  , KTE , KVG , KVP , KVR , KXC , KYC                           00000670
     +  , NQ  , NQN , NSL , NSN , NSX , NSY                           00000680
     +  , MQ                                                          00000690
     +  , KKQX( 10) , KKQM( 10)                                       00000700
      COMMON /REAL/                                                   00000710
     +   UA  , UD  , UGL , UGT , ULC                                   00000720
     +  , UQL , UQA , UQT , UQM , UQR , UQC , UQQ , UQV               00000730
     +  , URK , URD , UTC , UVV , UVK , UVT , UVI , UVD , UVU , UZM   00000740
     +  , U9C                                                         00000750
     +  , AX  , AY  , C   , CE  , DX  , DY  , DR  , DM  , DXT , DYT   00000760
     +  , G   , GL  , GT  , G2  , P   , QE  , QD                      00000770
     +  , R   , RK  , RP  , RD  , SC  , TC  , TD  , TE  ,TPHI         00000780
     +  , V   , VN  , VG  , VP  , VK  , VKN , VT  , VTN , VM          00000790
     +  , VR  , VI  , VD  , VU                                        00000800
     +  , XC  , X9C , XGL , XGM , XGI , XD                            00000810
     +  , YC  , Y9C , YGL , YGM , YGI , YD  , ZM                      00000820
     +  , QQXL( 10) , QQXM( 10) , QQYL( 10) , QQYM( 10)               00000830
     +  , QQA ( 10) , QQT ( 10) , QQV ( 10) , QQ  ( 10)               00000840
     +  , QQC ( 10) , QQR ( 10) , QQM ( 10)                           00000850
      EQUIVALENCE (HHCOM(1), H), (KKCOM(1), KDX), (RRCOM(1), UA)      00000860
      EQUIVALENCE (BBT(1), BT1)                                       00000870
       DATA NACC, NEXP / 20, 80 /                                     00000880
      DATA MBT, MHCOM, MKCOM, MRCOM, MU / 8, 381, 19, 76, 30 /        00000890
      DATA BIP /.TRUE./                                               00000900
C     ================================ GLOBAL FORMATS ================00000910
 1001 FORMAT(1X)                                                      00000920
 1002 FORMAT('0')                                                     00000930
 1003 FORMAT('0'/)                                                    00000940
 1021 FORMAT(18A4)                                                    00000950
 1023 FORMAT(5E15.0)                                                  00000960
 1031 FORMAT(1X, 18A4)                                                00000970
 1033 FORMAT(1X, 5G15.7)                                              00000980
 1081 FORMAT(' INVALID OR MISSING VALUE.')                            00000990
C     ================================ BEGIN EXECUTION ===============00001000
 1101 FORMAT(' GROUNDWATER PLUME CALCULATION PROGRAM'                 00001010
     +     / ' D.C. KENT, HYDROGEOLOGIST, PRINCIPAL INVESTIGATOR'     00001020
     +     / ' FRED WITZ, PROGRAMMER'                                 00001030
     +     / ' GEOLOGY DEPARTMENT, OKLAHOMA STATE UNIVERSITY'         00001040
     +     / ' FORTRAN VERSION 1.1 (1984, JANUARY)' / )               00001050
      WRITE(LUP, 1101)                                                00001060
      GO TO 1500                                                      00001070
C     ================================ INPUT LOOP ====================00001080
 1400 CONTINUE                                                        00001090
 1401 FORMAT(' COMMAND?')                                             00001100
      IF(BOP) WRITE(LUP, 1401)                                        00001110
      READ (LUR, 1021) (HHCMD(I), I = 1, 18)                          00001120
      IF(BOE) CALL PUTH(HHCMD, LUE)                                   00001130
      HHCMD(19) = H                                                   00001140
C     ================================ NEXT COMMAND LOOP: POP ========00001150
 1500 CONTINUE                                                        00001160
      HCMD = HHCMD(1)                                                 00001170
      IF(HCMD.EQ.H) GO TO 1400                                        00001180
      DO 1580 I = 2, MCMD                                             00001190
        HHCMD(I - 1) = HHCMD(I)                                       00001200
        IF(HHCMD(I).EQ.H) GO TO 1590                                  00001210
 1580   CONTINUE                                                      00001220
      HHCMD(MCMD) = H                                                 00001230
 1590 CONTINUE                                                        00001240
C     ================================ RE-EXECUTE LOOP: SIEVE ========00001250
 1700 CONTINUE                                                        00001260
      IF(BT2) CALL PUTH(HHCMD, LUE)                                   00001270
      IF(BT1) WRITE(LUE,1031) HCMD                                    00001280
```

```
IF(HCMD.EQ. H       ) GO TO 1400                                    00001290
IF(HCMD.EQ. HA      ) GO TO 2100                                    00001300
IF(HCMD.EQ.HHAX (1) ) GO TO 2110                                    00001310
IF(HCMD.EQ.HHAY (1) ) GO TO 2120                                    00001320
IF(HCMD.EQ. HAR     ) GO TO      2330                               00001330
IF(HCMD.EQ. HC      ) GO TO 2200                                    00001340
IF(HCMD.EQ. HD      ) GO TO 2300                                    00001350
IF(HCMD.EQ.HHDX (1) ) GO TO 2310                                    00001360
IF(HCMD.EQ.HHDY (1) ) GO TO 2320                                    00001370
IF(HCMD.EQ.HHDR (1) ) GO TO 2330                                    00001380
IF(HCMD.EQ.HHDM (1) ) GO TO 2340                                    00001390
IF(HCMD.EQ. HE      ) GO TO 2400                                    00001400
IF(HCMD.EQ. HG      ) GO TO 2500                                    00001410
IF(HCMD.EQ.HHGG (1) ) GO TO 2510                                    00001420
IF(HCMD.EQ.HHGL (1) ) GO TO 2520                                    00001430
IF(HCMD.EQ.HHGT (1) ) GO TO 2530                                    00001440
IF(HCMD.EQ. HH      ) GO TO 2600                                    00001450
IF(HCMD.EQ. HI      ) GO TO 2700                                    00001460
IF(HCMD.EQ. HIP     ) GO TO 2710                                    00001470
IF(HCMD.EQ. HIL     ) GO TO 2720                                    00001480
IF(HCMD.EQ. HIR     ) GO TO 2730                                    00001490
IF(HCMD.EQ. HL      ) GO TO 2800                                    00001500
IF(HCMD.EQ. HLX     ) GO TO 2810                                    00001510
IF(HCMD.EQ. HLY     ) GO TO 2820                                    00001520
IF(HCMD.EQ. HM      ) GO TO      4700                               00001530
IF(HCMD.EQ. HO      ) GO TO 3000                                    00001540
IF(HCMD.EQ. HOD     ) GO TO 3010                                    00001550
IF(HCMD.EQ. HOW     ) GO TO 3020                                    00001560
IF(HCMD.EQ. HOP     ) GO TO 3030                                    00001570
IF(HCMD.EQ. HOE     ) GO TO 3040                                    00001580
IF(HCMD.EQ. HOT     ) GO TO 3050                                    00001590
IF(HCMD.EQ.HHP  (1) ) GO TO 3100                                    00001600
IF(HCMD.EQ. HQ      ) GO TO 3500                                    00001610
IF(HCMD.EQ.HHQE (1) ) GO TO 3810                                    00001620
IF(HCMD.EQ. HQN     ) GO TO 3820                                    00001630
IF(HCMD.EQ.HHQM (1) ) GO TO 3900                                    00001640
IF(HCMD.EQ. HR      ) GO TO 4100                                    00001650
IF(HCMD.EQ.HHRR (1) ) GO TO 4100                                    00001660
IF(HCMD.EQ. HS      ) GO TO 4200                                    00001670
IF(HCMD.EQ. HSC     ) GO TO 4210                                    00001680
IF(HCMD.EQ. HSL     ) GO TO 4220                                    00001690
IF(HCMD.EQ. HSN     ) GO TO 4230                                    00001700
IF(HCMD.EQ. HSX     ) GO TO 4240                                    00001710
IF(HCMD.EQ. HSY     ) GO TO 4250                                    00001720
IF(HCMD.EQ.HHTC (1) ) GO TO 4300                                    00001730
 IF(HCMD.EQ.HHTE (1) ) GO TO 4400                                   00001740
IF(HCMD.EQ. HV      ) GO TO 4500                                    00001750
IF(HCMD.EQ.HHVV (1) ) GO TO 4500                                    00001760
IF(HCMD.EQ. HXY     ) GO TO 4600                                    00001770
IF(HCMD.EQ.HHXC (1) ) GO TO 4610                                    00001780
IF(HCMD.EQ.HHYC (1) ) GO TO 4620                                    00001790
IF(HCMD.EQ.HHZM (1) ) GO TO 4700                                    00001800
IF(HCMD.EQ.HH9C (1) ) GO TO 5100                                    00001810
 IF(HCMD.EQ.HH9T (1) ) GO TO      5600                              00001820
IF(HCMD.EQ. H9D     ) GO TO 5200                                    00001830
IF(HCMD.EQ. H9DC    ) GO TO 5210                                    00001840
IF(HCMD.EQ. H9DG    ) GO TO 5220                                    00001850
IF(HCMD.EQ. H9DP    ) GO TO 5250                                    00001860
IF(HCMD.EQ. H9DQ    ) GO TO 5260                                    00001870
IF(HCMD.EQ. H9FF    ) GO TO 5300                                    00001880
IF(HCMD.EQ. H9G     ) GO TO 5400                                    00001890
IF(HCMD.EQ. H9I     ) GO TO 5500                                    00001900
IF(HCMD.EQ. H9IC    ) GO TO 5510                                    00001910
IF(HCMD.EQ. H9IG    ) GO TO 5520                                    00001920
IF(HCMD.EQ. H9IP    ) GO TO      5250                               00001930
IF(HCMD.EQ. H9IQ    ) GO TO 5560                                    00001940
```

```
        IF(HCMD.EQ. H9TG    ) GO TO 5620                                    00001950
        WRITE(LUP, 1801) HCMD                                               00001960
 1801 FORMAT(1X, A4, '?')                                                   00001970
        GO TO 1400                                                          00001980
C       ================================ COMMAND EXITS ==================00001990
C       ================================ INPUT AND MISC. ================00002000
C       -------------------------------- A -------------------------------00002010
 2100   CONTINUE                                                            00002020
 2101 FORMAT(' 1 FOR X AND Y DISPERSIVITY (AX, AY),'                        00002030
     +      / ' 2 FOR X DISPERSIVITY AND DISPERSION RATIO (AX, DR):'       00002040
     +      / ' ?')                                                        00002050
        IF(BOP) WRITE(LUP, 2101)                                            00002060
        CALL GETI(ICMD, 1, 2)                                               00002070
        IF(BERR) GO TO 8110                                                 00002080
        IF(ICMD.EQ.1) CALL PUSH(HHAY(1) )                                   00002090
        IF(ICMD.EQ.2) CALL PUSH(HHDR(1) )                                   00002100
C       GO TO 2110                                                          00002110
C       -------------------------------- AX ------------------------------00002120
 2110   CALL GETR(HHAX, AX, .TRUE.)                                         00002130
        IF(BERR) GO TO 8120                                                 00002140
        KDX = 2                                                             00002150
        GO TO 1500                                                          00002160
C       -------------------------------- AY ------------------------------00002170
 2120   CALL GETR(HHAY, AY, .TRUE.)                                         00002180
        IF(BERR) GO TO 8120                                                 00002190
        KDY = 2                                                             00002200
        GO TO 1500                                                          00002210
C       -------------------------------- C -------------------------------00002220
 2200   CONTINUE                                                            00002230
 2201 FORMAT(' THREE TITLE LINES:'/ ' ?')                                   00002240
 2202 FORMAT(' ?')                                                          00002250
        IF(BOP) WRITE(LUP, 2201)                                            00002260
        READ (LUR, 1021) HHC1                                               00002270
        IF(BOE) CALL PUTH(HHC1, LUE)                                        00002280
        IF(BOP) WRITE(LUP, 2202)                                            00002290
        READ (LUR, 1021) HHC2                                               00002300
        IF(BOE) CALL PUTH(HHC2, LUE)                                        00002310
        IF(BOP) WRITE(LUP, 2202)                                            00002320
        READ (LUR, 1021) HHC3                                               00002330
        IF(BOE) CALL PUTH(HHC3, LUE)                                        00002340
        GO TO 1500                                                          00002350
C       -------------------------------- D = D OR A ---------------------00002360
 2300   CONTINUE                                                            00002370
 2301 FORMAT(' 1 FOR X AND Y DISPERSION (DX, DY),'                          00002380
     +      / ' 2 FOR X DISPERSION AND DISPERSION RATIO (DX, DR),'         00002390
     +      / ' 3 FOR X AND Y DISPERSIVITY (AX, AY),'                      00002400
     +      / ' 4 FOR X DISPERSIVITY AND DISPERSION RATIO (AX, DR),'       00002410
     +      / '   (USE DM FOR MOLECULAR DIFFUSION):'                       00002420
     +      / ' ?')                                                        00002430
        IF(BOP) WRITE(LUP, 2301)                                            00002440
        CALL GETI(ICMD, 1, 4)                                               00002450
        IF(BERR) GO TO 8110                                                 00002460
        IF(ICMD.EQ.1) CALL PUSH(HHDY(1) )                                   00002470
        IF(ICMD.EQ.2) CALL PUSH(HHDR(1) )                                   00002480
        IF(ICMD.EQ.3) CALL PUSH(HHAY(1) )                                   00002490
        IF(ICMD.EQ.4) CALL PUSH(HHDR(1) )                                   00002500
        GO TO (2310, 2310, 2110, 2110), ICMD                                00002510
C       -------------------------------- DX ------------------------------00002520
 2310   CALL GETR(HHDX, DX, .TRUE.)                                         00002530
        IF(BERR) GO TO 8120                                                 00002540
        KDX = 1                                                             00002550
        GO TO 1500                                                          00002560
C       -------------------------------- DY ------------------------------00002570
 2320   CALL GETR(HHDY, DY, .TRUE.)                                         00002580
        IF(BERR) GO TO 8120                                                 00002590
        KDY = 1                                                             00002600
```

```
      GO TO 1500                                                        00002610
C     ------------------------------ DR ------------------------------00002620
 2330 CALL GETR(HHDR, DR, .TRUE.)                                       00002630
      IF(BERR) GO TO 8120                                               00002640
      KDY = 3                                                           00002650
      GO TO 1500                                                        00002660
C     ------------------------------ DM ------------------------------00002670
 2340 CALL GETR(HHDM, DM, .TRUE.)                                       00002680
      IF(BERR) GO TO 8120                                               00002690
      GO TO 1500                                                        00002700
C     ------------------------------ E -------------------------------00002710
 2400 CONTINUE                                                          00002720
 2401 FORMAT(' END OF PLUME PROGRAM.')                                  00002730
 2402 FORMAT(' GOODBYE.')                                               00002740
      WRITE(LUE, 2401)                                                  00002750
      WRITE(LUP, 2402)                                                  00002760
      STOP                                                              00002770
C     ------------------------------ G -------------------------------00002780
 2500 CONTINUE                                                          00002790
 2501 FORMAT(' 0 FOR NO DECAY,'                                         00002800
     +     / ' 1 FOR DECAY COEFFICIENT, GAMMA (GG),'                    00002810
     +     / ' 2 FOR DECAY LAMBDA (GL),'                                00002820
     +     / ' 3 FOR DECAY HALF-LIFE (GT):'                             00002830
     +     / ' ?')                                                      00002840
      IF(BOP) WRITE(LUP, 2501)                                          00002850
      CALL GETI(ICMD, 0, 3)                                             00002860
      IF(BERR) GO TO 8110                                               00002870
      ICMD = ICMD + 1                                                   00002880
      GO TO (2505, 2510, 2520, 2530), ICMD                              00002890
 2505 G  = 1.0                                                          00002900
      KG = 1                                                            00002910
      GO TO 1500                                                        00002920
C     ------------------------------ GG ------------------------------00002930
 2510 CALL GETR(HHGG, XC, .FALSE.)                                      00002940
      IF(XC.LT.1.0) GO TO 8120                                          00002950
      G  = XC                                                           00002960
      KG = 1                                                            00002970
      GO TO 1500                                                        00002980
C     ------------------------------ GL ------------------------------00002990
 2520 CALL GETR(HHGL, XC, .FALSE.)                                      00003000
      IF(XC.LT.0.0) GO TO 8120                                          00003010
      GL = XC                                                           00003020
      KG = 2                                                            00003030
      GO TO 1500                                                        00003040
C     ------------------------------ GT ------------------------------00003050
 2530 CALL GETR(HHGT, GT, .TRUE.)                                       00003060
      IF(BERR) GO TO 8120                                               00003070
      GL = 1.0                                                          00003080
      KG = 3                                                            00003090
      GO TO 1500                                                        00003100
C     ------------------------------ H -------------------------------00003110
 2600 CONTINUE                                                          00003120
 2601 FORMAT(                                                           00003130
     +  ' INPUT:              OUTPUT:'                                  00003140
     +/ ' C  CASE TITLE      .I  INPUT PARAM.'                         00003150
     +/ ' M  THICKNESS       .D  ALL PARAM.'                           00003160
     +/ ' P  POROSITY        .C  SINGLE'                               00003170
     +/ ' V  VELOCITY        .G  GRID MAP'                             00003180
     +/ ' D  DISPERSION      .FF PAGE PRINT'                           00003190
     +/ ' A  DISPERSIVITY'                                             00003200
     +/ ' R  RETARDATION     SPECIAL:'                                 00003210
     +/ ' G  DECAY           H  HELP'                                  00003220
     +/ ' Q  SOURCE          E  EXIT'                                  00003230
     +/ ' T  SAMPLE TIME     I  INPUT'                                 00003240
     +/ ' TE STEADY STATE    O  OUTPUT'                                00003250
     +/ ' XY SINGLE X,Y'                                               00003260
```

```
     +/ ' L  GRID LIMITS'                                                    00003270
     +/ ' S  GRID SCALES')                                                   00003280
        WRITE(LUP, 2601)                                                     00003290
        GO TO 1500                                                           00003300
C       ------------------------------- I ----------------------------00003310
 2700   CONTINUE                                                             00003320
 2701 FORMAT( / '  1 TO PROMPT FOR ALL REQUIRED PARAMETERS (IP),'            00003330
     +      / '  2 TO LOAD PREVIOUSLY SAVED PARAMETERS (IL),'               00003340
     +      / '  3 TO READ ALL INPUT FROM ANOTHER SOURCE (IR),'             00003350
     +      / ' -1 TO SET OUTPUT PARAMETERS (O):'                          00003360
     +      / ' ?')                                                        00003370
        IF(BOP) WRITE(LUP, 2701)                                             00003380
        CALL GETI(ICMD, -1, 3)                                               00003390
        IF(BERR) GO TO 8110                                                  00003400
        IF(ICMD.EQ. 0) GO TO 8110                                            00003410
        IF(ICMD.EQ. 1) GO TO 2710                                            00003420
        IF(ICMD.EQ. 2) GO TO 2720                                            00003430
        IF(ICMD.EQ. 3) HCMD = HIR                                            00003440
        IF(ICMD.EQ.-1) HCMD = HO                                             00003450
        CALL PUSH(HI)                                                        00003460
        GO TO 1700                                                           00003470
C       ------------------------------- IP ----------------------------00003480
C       (LAST COMMAND MUST RESET BIP)                                        00003490
 2710   BIP = .TRUE.                                                         00003500
        KLX = 0                                                              00003510
        KLY = 0                                                              00003520
        KXC = 0                                                              00003530
        KYC = 0                                                              00003540
        NQ  = 0                                                              00003550
        CALL PUSH(HD)                                                        00003560
        CALL PUSH(HG)                                                        00003570
        CALL PUSH(HR)                                                        00003580
        CALL PUSH(HHTC(1) )                                                  00003590
        CALL PUSH(HQ)                                                        00003600
         CALL PUSH(HL)                                                       00003610
        CALL PUSH(HV)                                                        00003620
        CALL PUSH(HHP (1) )                                                  00003630
        CALL PUSH(HHZM(1) )                                                  00003640
        HCMD = HC                                                            00003650
        GC TO 1700                                                           00003660
C       ------------------------------- IL ----------------------------00003670
 2720   CONTINUE                                                             00003680
 2721 FORMAT(' LOAD FILE UNIT (', I3, ' TO', I3, '):'/ ' ?')                 00003690
        WRITE(LUP, 2721) LUL, LUM                                            00003700
        CALL GETI(LU, LUL, LUM)                                              00003710
        IF(BERR) GO TO 8120                                                  00003720
        READ (LU) I, IX, IY, IQ                                              00003730
        IF(    I .NE.MHCOM                                                   00003740
     +    .OR. IX.NE.MKCOM                                                   00003750
     +    .OR. IY.NE.MRCOM                                                   00003760
     +    .OR. IQ.NE.MQ                                                      00003770
     +    ) GO TO 8120                                                       00003780
        READ (LU) HHCOM                                                      00003790
        READ (LU) KKCOM, KKQX, KKQM                                          00003800
        READ (LU) RRCOM                                                      00003810
        READ (LU) QQXL, QQXM, QQYL, QQYM                                     00003820
     +          , QQA , QQT , QQV , QQ                                      00003830
     +          , QQC , QQR , QQM                                           00003840
        ENDFILE LU                                                           00003850
        BIP = .FALSE.                                                        00003860
        GO TO 1500                                                           00003870
C       ------------------------------- IR ----------------------------00003880
 2730   CONTINUE                                                             00003890
 2731 FORMAT(' INPUT UNIT (1 TO', I3, '):'/ ' ?')                            00003900
        IF(BOP) WRITE(LUP, 2731) LUM                                         00003910
        CALL GETI(LU, 1, LUM)                                                00003920
```

```
        IF(BERR) GO TO 8120                                                   00003930
        LUR = LU                                                              00003940
        GO TO 1500                                                            00003950
C       ------------------------------- L ------------------------------00003960
 2800   CALL PUSH(HLY )                                                       00003970
C       GO TO 2810                                                            00003980
C       ------------------------------- LX -----------------------------00003990
 2810   CONTINUE                                                              00004000
 2811 FORMAT(' GRID MAP X LOCATIONS (',2A4,'),'                                00004010
     +     / ' MINIMUM, (MAXIMUM), (INTERVAL):'                              00004020
     +     / ' ?')                                                           00004030
        IF(BOP) WRITE(LUP, 2811) HHXC(6), HHXC(7)                             00004040
        READ (LUR, 1023) XGL, XGM, XGI                                        00004050
        IF(BOE) WRITE(LUE, 1033) XGL, XGM, XGI                                00004060
        KLX = 0                                                               00004070
        IF(XGM.LE.XGL .AND. XGI.LE.0.0) GO TO 8120                            00004080
        KLX = 1                                                               00004090
        GO TO 1500                                                            00004100
C       ------------------------------- LY -----------------------------00004110
 2820   CONTINUE                                                              00004120
 2821 FORMAT(' GRID MAP Y LOCATIONS (',2A4,'),'                                00004130
     +     / ' MINIMUM, MAXIMUM, (INTERVAL):'                                00004140
     +     / ' ?')                                                           00004150
        IF(BOP) WRITE(LUP, 2821) HHYC(6), HHYC(7)                             00004160
        READ (LUR, 1023) YGL, YGM, YGI                                        00004170
        IF(BOE) WRITE(LUE, 1033) YGL, YGM, YGI                                00004180
        KLY = 1                                                               00004190
        GO TO 1500                                                            00004200
C       ------------------------------- O ------------------------------00004210
 3000   CONTINUE                                                              00004220
 3001 FORMAT(' 1 TO DUMP ALL PARAMETERS TO DISK (OD),'                         00004230
     +     / ' 2 TO SET LOCATION FOR WRITING RESULTS (OW),'                   00004240
     +     / ' 3 TO SET PROMPTING (OP),'                                      00004250
     +     / ' 4 TO SET ECHO (OE),'                                           00004260
     +     / ' 5 TO SET TRACE (OT),'                                          00004270
     +     / ' ?')                                                           00004280
        IF(BOP) WRITE(LUP, 3001)                                              00004290
        CALL GETI(ICMD, 1, 5)                                                 00004300
        IF(BERR) GO TO 8110                                                   00004310
        GO TO (3010, 3020, 3030, 3040, 3050), ICMD                            00004320
C       ------------------------------- OD -----------------------------00004330
 3010   CONTINUE                                                              00004340
        IF(BIP) GO TO 8110                                                    00004350
 3011 FORMAT(' DUMP FILE UNIT (', I3, ' TO', I3, '):'/ ' ?')                   00004360
        WRITE(LUP, 3011) LUL, LUM                                             00004370
        CALL GETI(LU, LUL, LUM)                                               00004380
        IF(BERR) GO TO 8120                                                   00004390
        WRITE(LU) MHCOM, MKCOM, MRCOM, MQ                                     00004400
        WRITE(LU) HHCOM                                                       00004410
        WRITE(LU) KKCOM, KKQX, KKQM                                           00004420
        WRITE(LU) RRCOM                                                       00004430
        WRITE(LU) QQXL, QQXM, QQYL, QQYM                                      00004440
     +          , QQA , QQT , QQV , QQ                                       00004450
     +          , QQC , QQR , QQM                                            00004460
        ENDFILE LU                                                            00004470
        GO TO 1500                                                            00004480
C       ------------------------------- OW -----------------------------00004490
 3020   CONTINUE                                                              00004500
 3021 FORMAT(' 1 TO' I3, ' TO SET RESULTS UNIT:'/ ' ?')                        00004510
        IF(BOP) WRITE(LUP, 3021) LUM                                          00004520
        CALL GETI(LU, 1, LUM)                                                 00004530
        IF(BERR) GO TO 8120                                                   00004540
        LUW = LU                                                              00004550
        GO TO 1500                                                            00004560
C       ------------------------------- OP -----------------------------00004570
 3030   CONTINUE                                                              00004580
```

```
 3031 FORMAT(' -1 FOR NO PROMPTING,'                                    00004590
     +      / '  0 FOR PROMPTING,'                                     00004600
     +      / '  1 TO' I3, ' TO SET PROMPTING UNIT:'/ ' ?')            00004610
        IF(BOP) WRITE(LUP, 3031) LUM                                    00004620
        CALL GETI(LU, -1, LUM)                                          00004630
        IF(BERR) GO TO 8120                                             00004640
        BOP = LU.GE.0                                                   00004650
        IF(LU.GT.0) LUP = LU                                            00004660
        GO TO 1500                                                      00004670
C       ------------------------------ OE ----------------------------00004680
 3040   CONTINUE                                                        00004690
 3041 FORMAT(' -1 FOR NO ECHO,'                                         00004700
     +      / '  0 FOR ECHO,'                                          00004710
     +      / '  1 TO' I3, ' TO SET ECHO UNIT:'/ ' ?')                 00004720
        IF(BOP) WRITE(LUP, 3041) LUM                                    00004730
        CALL GETI(LU, -1, LUM)                                          00004740
        IF(BERR) GO TO 8120                                             00004750
        BOE = LU.GE.0                                                   00004760
        IF(LU.GT.0) LUE = LU                                            00004770
        GO TO 1500                                                      00004780
C       ------------------------------ OT ----------------------------00004790
 3050   CONTINUE                                                        00004800
 3051 FORMAT(' + FOR TRACE, - FOR NO TRACE:'                            00004810
     +      / ' 1 FOR COMMAND     TRACE,'                              00004820
     +      / ' 2 FOR STACK       TRACE,'                              00004830
     +      / ' 5 FOR CALCULATION TRACE:'/ ' ?')                       00004840
        IF(BOP) WRITE(LUP, 3051)                                        00004850
        CALL GETI(ICMD, -MBT, MBT)                                      00004860
        IF(BERR) GO TO 8120                                             00004870
        LU = IABS(ICMD)                                                 00004880
        IF(LU.NE. 0) BBT(LU) = ICMD.GT.0                                00004890
        GO TO 1500                                                      00004900
C       ------------------------------ P -----------------------------00004910
 3100   CALL GETR(HHP , XC, .TRUE.)                                     00004920
        IF(BERR) GO TO 8120                                             00004930
        IF(XC.GT.1.) GO TO 8120                                         00004940
        P = XC                                                          00004950
        GO TO 1500                                                      00004960
C       ============================== Q ============================00004970
 3500   CONTINUE                                                        00004980
 3511 FORMAT(' CURRENTLY USING', I4, ' OF', I4, ' TIME STEPS.'          00004990
     +      /'  0 TO END SOURCE INFORMATION,'                          00005000
     +      /'  1 TO ADD POINT SOURCE,'                                00005010
     +      /'  2 TO ADD NON-POINT SOURCE,'                            00005020
     +      /' -N TO DELETE LAST N TIME STEPS:'                        00005030
     +      /' ?')                                                     00005040
 3512 FORMAT(' **** BEGIN SOURCE INFORMATION ****')                     00005050
 3513 FORMAT(' **** END SOURCE INFORMATION ****')                       00005060
        WRITE(LUP, 3512)                                                00005070
 3510   IF(BOP) WRITE(LUP, 3511) NQ, MQ                                 00005080
        CALL GETI(ICMD, 0, 0)                                           00005090
        IF(ICMD.LT.0) GO TO 3520                                        00005100
        IF(ICMD.EQ.1) GO TO 3530                                        00005110
        IF(ICMD.EQ.2) GO TO 3540                                        00005120
        IF(NQ  .LE.0) GO TO 3790                                        00005130
        IF(ICMD.EQ.0) GO TO 3515                                        00005140
        GO TO 3790                                                      00005150
 3515   WRITE(LUP, 3513)                                                00005160
        GO TO 1500                                                      00005170
C       ------------------------------ REDUCE ------------------------00005180
 3520   NQ = MAXO(NQ + ICMD, 0)                                         00005190
        GO TO 3510                                                      00005200
C       ------------------------------ POINT -------------------------00005210
 3530   IF(NQ.GE.MQ) GO TO 3799                                         00005220
 3531 FORMAT(' X LOCATION, Y LOCATION (', 2A4, '):'/ ' ?')              00005230
        IF(BOP) WRITE(LUP, 3531)  HHQL(6), HHQL(7)                      00005240
```

```
      READ (LUR, 1023) QXL, QYL                                          00005250
      IF(BOE) WRITE(LUE, 1033) QXL, QYL                                  00005260
      CALL GETR(HHQA, QA, .FALSE.)                                       00005270
C     IF(QA.LT.0.) GO TO 3790                                            00005280
      KQX = 1                                                            00005290
      QXM = QXL                                                          00005300
      QYM = QYL                                                          00005310
      GO TO 3600                                                         00005320
C     -------------------------------- NON-POINT ---------------------00005330
 3540 IF(NQ.GE.MQ) GO TO 3799                                            00005340
 3541 FORMAT(' X LOCATION MINIMUM, MAXIMUM (', 2A4, '):'/ ' ?')         00005350
      IF(BOP) WRITE(LUP, 3541) HHQL(6), HHQL(7)                          00005360
      READ (LUR, 1023) QXL, QXM                                          00005370
      IF(BOE) WRITE(LUE, 1033) QXL, QXM                                  00005380
      IF(QXM.LE.QXL) GO TO 3790                                          00005390
 3542 FORMAT(' Y LOCATION MINIMUM, MAXIMUM (', 2A4, '):'/ ' ?')         00005400
      IF(BOP) WRITE(LUP, 3542) HHQL(6), HHQL(7)                          00005410
      READ (LUR, 1023) QYL, QYM                                          00005420
      IF(BOE) WRITE(LUE, 1033) QYM, QYM                                  00005430
      IF(QYM.LE.QYL) GO TO 3790                                          00005440
      KQX = 2                                                            00005450
      QA = (QXM - QXL)*(QYM - QYL)*UQL*UQL/UQA                           00005460
C     -------------------------------- TIME AND RATE -----------------00005470
 3600 CONTINUE                                                           00005480
 3601 FORMAT(' 1 FOR MASS FLOW RATE     (', 2A4, '),'                    00005490
     +     / ' 2 FOR MASS/AREA RATE     (', 2A4, '),'                    00005500
     +     / ' 3 FOR VOLUME FLOW RATE   (', 2A4, '),'                    00005510
     +     / '         AND CONCENTRATION (', 2A4, '),'                   00005520
     +     / ' 4 FOR VOLUME/AREA RATE   (', 2A4, '),'                    00005530
     +     / '         AND CONCENTRATION (', 2A4, '):'                   00005540
     +     / ' ?')                                                       00005550
      IF(BOP) WRITE(LUP, 3601)                                           00005560
     +   HHQM(6), HHQM(7), HHQR(6), HHQR(7)                             00005570
     +  , HHQQ(6), HHQQ(7), HHQC(6), HHQC(7)                            00005580
     +  , HHQV(6), HHQV(7), HHQC(6), HHQC(7)                            00005590
      CALL GETI(KQM, 1, 4)                                               00005600
      IF(BERR) GO TO 3790                                                00005610
      QTO = -1E20                                                        00005620
      QAU = QA*UQA                                                       00005630
      QV  = 0.0                                                          00005640
      Q   = 0.0                                                          00005650
      QC  = 0.0                                                          00005660
      QR  = 0.0                                                          00005670
 3700 CONTINUE                                                           00005680
      GO TO (3710, 3720, 3730, 3740), KQM                                00005690
 3710 CONTINUE                                                           00005700
 3711 FORMAT(' TIME, MASS FLOW RATE, (CONCENTRATION):'/ ' ?')            00005710
      IF(BOP) WRITE(LUP, 3711)                                           00005720
      READ (LUR, 1023) QT, QM, QC                                        00005730
      IF(BOE) WRITE(LUE, 1033) QT, QM, QC                                00005740
      Q  = 0.                                                            00005750
      IF(QC.GT.0.0) Q  = QM*UQM/(QC*UQC    )                             00005760
      IF(QA.LE.0.0) GO TO 3715                                           00005770
      QR = QM*UQM/(QAU   *UQR)                                           00005780
      QV = Q     /(QAU   *UQV)                                           00005790
 3715 Q  = Q /UQQ                                                        00005800
      GO TO 3760                                                         00005810
 3720 CONTINUE                                                           00005820
      IF(QA.LE.0.0) GO TO 3790                                           00005830
 3721 FORMAT(' TIME, MASS/AREA RATE, (CONCENTRATION):'/ ' ?')            00005840
      IF(BOP) WRITE(LUP, 3721)                                           00005850
      READ (LUR, 1023) QT, QR, QC                                        00005860
      IF(BOE) WRITE(LUE, 1033) QT, QR, QC                                00005870
      QM = QR*UQR*QAU                                                    00005880
      IF(QC.LE.0.0) GO TO 3725                                           00005890
      Q  = QM    /(QC*UQC*UQQ)                                           00005900
```

```
        QV = Q *UQQ/(QAU   *UQV)                                          00005910
        GO TO 3750                                                        00005920
 3725   Q  = 0.                                                           00005930
        QV = 0.                                                           00005940
        GO TO 3750                                                        00005950
 3730   CONTINUE                                                          00005960
 3731 FORMAT(' TIME, VOLUME FLOW RATE, CONCENTRATION:'/ ' ?')             00005970
        IF(BOP) WRITE(LUP, 3731)                                          00005980
        READ (LUR, 1023) QT, Q, QC                                        00005990
        IF(BOE) WRITE(LUE, 1033) QT, Q, QC                                00006000
        QM = Q *UQQ*QC*UQC                                                00006010
        IF(QA.LE.0.0) GO TO 3750                                          00006020
        QR = QM    /(QAU   *UQR)                                          00006030
        QV = Q *UQQ/(QAU   *UQV)                                          00006040
        GO TO 3750                                                        00006050
 3740   CONTINUE                                                          00006060
        IF(QA.LE.0.0) GO TO 3790                                          00006070
 3741 FORMAT(' TIME, VOLUME/AREA RATE, CONCENTRATION:'/ ' ?')             00006080
        IF(BOP) WRITE(LUP, 3741)                                          00006090
        READ (LUR, 1023) QT, QV, QC                                       00006100
        IF(BOE) WRITE(LUE, 1033) QT, QV, QC                               00006110
        Q  = QV*UQV*QAU   /UQQ                                            00006120
        QR = QV*UQV*QC*UQC                                                00006130
        QM = QR    *QAU                                                   00006140
        QR = QR/UQR                                                       00006150
 3750   CONTINUE                                                          00006160
        QM = QM/UQM                                                       00006170
 3760   CONTINUE                                                          00006180
        IF(QT.LE.QTO) GO TO 3510                                          00006190
        IF(QM.LT.0.) GO TO 3790                                           00006200
C       IF(QC.LT.0.) GO TO 3790                                           00006210
        IF(NQ.GE.MQ) GO TO 3790                                           00006220
        NQ = NQ + 1                                                       00006230
        KKQX(NQ) = KQX                                                    00006240
        KKQM(NQ) = KQM                                                    00006250
        QQXL(NQ) = QXL                                                    00006260
        QQXM(NQ) = QXM                                                    00006270
        QQYL(NQ) = QYL                                                    00006280
        QQYM(NQ) = QYM                                                    00006290
        QQT (NQ) = QT                                                     00006300
        QQA (NQ) = QA                                                     00006310
        QQV (NQ) = QV                                                     00006320
        QQ  (NQ) = Q                                                      00006330
        QQC (NQ) = QC                                                     00006340
        QQR (NQ) = QR                                                     00006350
        QQM (NQ) = QM                                                     00006360
        QTO = QT                                                          00006370
        KQX = ISIGN(KQX, -1)                                              00006380
        ICMD = MIN1(QTO, 0.0)                                             00006390
 3771 FORMAT(1X,'ENTER', I5, ' TO RETURN TO MAIN SOURCE MENU:')           00006400
        IF(BOP) WRITE(LUP, 3771) ICMD                                     00006410
        GO TO 3700                                                        00006420
C       ------------------------------- ERRORS ------------------------00006430
 3790   WRITE(LUP, 1081)                                                  00006440
 3799   IF(BATCH) GO TO 1400                                              00006450
        GO TO 3510                                                        00006460
C       ================================ QE ============================00006470
 3810   CALL GETR(HHQE, QE, .TRUE.)                                       00006480
        IF(BERR) GO TO 8120                                               00006490
        GO TO 1500                                                        00006500
C       ------------------------------- QN ----------------------------00006510
 3820   CONTINUE                                                          00006520
 3821 FORMAT(' MAXIMUM NUMBER OF SUBAREAS PER SOURCE:'                    00006530
     +     / ' ?')                                                       00006540
        IF(BOP) WRITE(LUP, 3821)                                          00006550
        CALL GETI(ICMD, 0, 0)                                             00006560
```

```
        IF(ICMD.LE.0) GO TO 8120                                          00006570
        NQN = ICMD                                                        00006580
        GO TO 1500                                                        00006590
C       ================================ QM ==============================00006600
 3900   CONTINUE                                                          00006610
C3901 FORMAT(                                                             00006620
C    + '          X             Y             START        MASS      '    00006630
C    +/' LINE    LOCATION      LOCATION      TIME         FLOW RATE'     00006640
C    +/ 6X, 4(' (', 2A4, ') ') )                                          00006650
C3902 FORMAT(1X, I4, 1X, 4G13.6)                                          00006660
C       LU = LUW                                                          00006670
C       IF(.NOT. BOP) GO TO 3920                                          00006680
C       LU = LUP                                                          00006690
C3910   WRITE(LU, 3901) HHQL(6), HHQL(7), HHQL(6), HHQL(7)               00006700
C    +                , HHQT(6), HHQT(7), HHQM(6), HHQM(7)               00006710
C       DO 3919 IQ = 1, NQ                                                00006720
C         WRITE(LU, 3902) IQ, QQXL(IQ), QQYL(IQ), QQT(IQ), QQM(IQ)        00006730
C3919     CONTINUE                                                        00006740
 3920   CONTINUE                                                          00006750
 3921 FORMAT(' -1 TO LIST FLOW RATES,'                                    00006760
     +     / '  0 TO END QM,'                                            00006770
     +     / ' TIME STEP (LINE) NUMBER TO CHANGE FLOW RATE:'             00006780
     +     / ' ?')                                                       00006790
        IF(BOP) WRITE(LUP, 3921)                                          00006800
        CALL GETI(ICMD, -1, NQ)                                           00006810
        IF(BERR) GO TO 3990                                               00006820
C       IF(ICMD) 3910, 1500, 3950                                         00006830
        IF(ICMD) 3930, 1500, 3950                                         00006840
 3930   CALL PUSH(HCMD)                                                   00006850
        HCMD = H9IQ                                                       00006860
        GO TO 1700                                                        00006870
 3950   CALL GETR(HHQM, QM, .FALSE.)                                      00006880
        IF(QM.LT.0.) GO TO 3990                                           00006890
        KKQM(ICMD) = 1                                                    00006900
        QQM (ICMD) = QM                                                   00006910
        QQR (ICMD) = 0.                                                   00006920
        QQ  (ICMD) = 0.                                                   00006930
        QQV (ICMD) = 0.                                                   00006940
        GO TO 3920                                                        00006950
C       -------------------------------- ERRORS ------------------------00006960
 3990   WRITE(LUP, 1081)                                                  00006970
        IF(BATCH) GO TO 1400                                              00006980
        GO TO 3920                                                        00006990
C       ================================ R ===============================00007000
 4100   CALL GETR(HHRR, XC, .FALSE.)                                      00007010
        IF(XC.LT.1.0) GO TO 8120                                          00007020
        R = XC                                                            00007030
        GO TO 1500                                                        00007040
C       ================================ R ===============================00007050
 4200   CONTINUE                                                          00007060
 4201 FORMAT(' GRID MAP SCALE PARAMETERS:'                                00007070
     +     / ' 1 TO SET CONCENTRATION/STEADY STATE SCALE (SC),'          00007080
     +     / ' 2 TO SET LINE SIZE (SL),'                                 00007090
     +     / ' 3 TO SET NODES PER LINE (SN),'                            00007100
     +     / ' 4 TO SET X SPACING (SX),'                                 00007110
     +     / ' 5 TO SET Y SPACING (SY):'                                 00007120
     +     / ' ?')                                                       00007130
        IF(BOP) WRITE(LUP, 4201)                                          00007140
        CALL GETI(ICMD, 1, 5)                                             00007150
        IF(BERR) GO TO 8110                                               00007160
        GO TO (4210, 4220, 4230, 4240, 4250), ICMD                        00007170
C       -------------------------------- SC ----------------------------00007180
 4210   CONTINUE                                                          00007190
 4211 FORMAT(' GRID MAP MULTIPLIER:'/ ' ?')                               00007200
        IF(BOP) WRITE(LUP, 4211)                                          00007210
        READ (LUR, 1023) XC                                               00007220
```

```
      IF(BOE) WRITE(LUE, 1033) XC                                       00007230
      IF(XC.LE.0.0) GO TO 8120                                          00007240
      SC = 1.0/XC                                                       00007250
      GO TO 1500                                                        00007260
C     ------------------------------ SL ------------------------------00007270
 4220 CONTINUE                                                          00007280
 4221 FORMAT(' GRID MAP LINE SIZE (40-255 CHARACTERS/LINE):'/ ' ?')     00007290
      IF(BOP) WRITE(LUP, 4221)                                          00007300
      CALL GETI(ICMD, 40, 255)                                          00007310
      IF(BERR) GO TO 8120                                               00007320
      NSL = ICMD                                                        00007330
      GO TO 1500                                                        00007340
C     ------------------------------ SN ------------------------------00007350
 4230 CONTINUE                                                          00007360
 4231 FORMAT(' GRID MAP NODES PER LINE (0 FOR FULL LINE):'/ ' ?')       00007370
      IF(BOP) WRITE(LUP, 4231)                                          00007380
      CALL GETI(NSN, 0, 0)                                              00007390
      GO TO 1500                                                        00007400
C     ------------------------------ SX ------------------------------00007410
 4240 CONTINUE                                                          00007420
 4241 FORMAT(' GRID MAP X SPACING (3 TO 6 CHARACTERS/NODE):'/ ' ?')     00007430
      IF(BOP) WRITE(LUP, 4241)                                          00007440
      CALL GETI(ICMD, 3, 6)                                             00007450
      IF(BERR) GO TO 8120                                               00007460
      NSX = ICMD                                                        00007470
      GO TO 1500                                                        00007480
C     ------------------------------ SY ------------------------------00007490
 4250 CONTINUE                                                          00007500
 4251 FORMAT(' GRID MAP Y SPACING (1 TO 4 LINES/NODE):'/ ' ?')          00007510
      IF(BOP) WRITE(LUP, 4251)                                          00007520
      CALL GETI(ICMD, 1, 4)                                             00007530
      IF(BERR) GO TO 8120                                               00007540
      NSY = ICMD                                                        00007550
      GO TO 1500                                                        00007560
C     ------------------------------ T -------------------------------00007570
 4300 CONTINUE                                                          00007580
 4301 FORMAT(' -1 FOR STEADY STATE,')                                   00007590
      IF(BOP) WRITE(LUP, 4301)                                          00007600
      CALL GETR(HHTC, TC, .FALSE.)                                      00007610
      BIP = .FALSE.                                                     00007620
      GO TO 1500                                                        00007630
C     ============================== TE ==============================00007640
 4400  CALL GETR(HHTE, XC, .TRUE.)                                      00007650
       IF(BERR) GO TO 8120                                              00007660
       IF(XC .GT. 100.) GO TO 8120                                      00007670
       KTE = 1                                                          00007680
       YC = XC/50.                                                      00007690
       PO = 0.                                                          00007700
       P2 = PO                                                          00007710
       IF(ERFC(PO) - YC) 4425, 4480, 4426                               00007720
 4425  PO = -1.                                                         00007730
       GO TO 4429                                                       00007740
 4426  PO = 1.                                                          00007750
 4429  CONTINUE                                                         00007760
       DO 4448 I = 1, NEXP                                              00007770
         IF((ERFC(PO) - YC)*PO) 4449, 4480, 4445                        00007780
 4445  P2 = PO                                                          00007790
       PO = PO * 2.                                                     00007800
 4448  CONTINUE                                                         00007810
       GO TO 8120                                                       00007820
 4449  P1 = AMIN1(PO,P2)                                                00007830
       P2 = AMAX1(PO,P2)                                                00007840
       DO 4468 I = 1, NACC                                              00007850
         PO = (P1 + P2)/2.0                                             00007860
         IF(ERFC(PO) - YC) 4465, 4480, 4466                             00007870
 4465    P2 = PO                                                        00007880
```

```
          GO TO 4467                                                    00007890
 4466     P1 = PO                                                       00007900
 4467     CONTINUE                                                      00007910
 4468     CONTINUE                                                      00007920
        PO = (P1 + P2)/2.0                                             00007930
 4480   CONTINUE                                                       00007940
        TE = XC                                                        00007950
        TPHI = PO                                                      00007960
        GO TO 1500                                                     00007970
C       ================================ V ==============================00007980
 4500   CALL GETR(HHVV, V , .TRUE.)                                    00007990
        IF(BERR) GO TO 8120                                            00008000
        GO TO 1500                                                     00008010
C       -------------------------------- XY ------------------------------00008020
 4600   CALL PUSH(HHYC(1))                                             00008030
        HCMD = HHXC(1)                                                 00008040
        GO TO 1700                                                     00008050
C       -------------------------------- X -------------------------------00008060
 4610   CONTINUE                                                       00008070
        CALL GETR(HHXC, X9C, .FALSE.)                                  00008080
        KXC = 1                                                        00008090
        GO TO 1500                                                     00008100
C       -------------------------------- Y -------------------------------00008110
 4620   CONTINUE                                                       00008120
        CALL GETR(HHYC, Y9C, .FALSE.)                                  00008130
        KYC = 1                                                        00008140
        GO TO 1500                                                     00008150
C       -------------------------------- ZM ------------------------------00008160
 4700   CALL GETR(HHZM, ZM, .TRUE.)                                    00008170
        IF(BERR) GO TO 8120                                            00008180
        GO TO 1500                                                     00008190
C       ================================ OUTPUT COMMANDS ===============00008200
 5001 FORMAT(' SAMPLE TIME     = STEADY STATE')                        00008210
C       -------------------------------- .C ------------------------------00008220
 5100   CONTINUE                                                       00008230
 5101 FORMAT(18X, '  +-', G9.3, 1X, 2A4)                              00008240
 5102 FORMAT(' SAMPLE LOCATION WITHIN SOURCE.')                         00008250
        IF(KXC.NE.O .AND. KYC.NE.O) GO TO 5110                         00008260
        IF(BATCH) GO TO 8120                                           00008270
        CALL PUSH(HH9C(1))                                             00008280
        IF(KYC.EQ.O) CALL PUSH(HHYC(1))                                00008290
        IF(KXC.EQ.O) CALL PUSH(HHXC(1))                                00008300
        GO TO 1500                                                     00008310
 5110   CONTINUE                                                       00008320
        CALL SETUP                                                     00008330
        XC = X9C                                                       00008340
        YC = Y9C                                                       00008350
        B = TC.GE.O.O                                                  00008360
        IF(B) CALL PUTR(HHTC, TC)                                      00008370
        IF(.NOT.B) WRITE(LUW, 5001)                                    00008380
        CALL PUTR(HHXC, XC)                                            00008390
        CALL PUTR(HHYC, YC)                                            00008400
        CALL CALC                                                      00008410
        IF(.NOT.BERR) CALL PUTR(HH9C, C )                              00008420
        IF(CE.GT.O.O) WRITE(LUW, 5101) CE , HH9C(6), HH9C(7)           00008430
        IF(BERR     ) WRITE(LUW, 5102)                                 00008440
        WRITE(LUW, 1001)                                               00008450
        GO TO 1500                                                     00008460
C       -------------------------------- .D ------------------------------00008470
 5200   CALL PUSH(H9DQ)                                                00008480
        HCMD = H9DP                                                    00008490
        GO TO 1700                                                     00008500
C       -------------------------------- .DC -----------------------------00008510
 5210   CALL PUSH(HH9C(1))                                             00008520
        HCMD = H9D                                                     00008530
        GO TO 1700                                                     00008540
```

```
C       ------------------------------- .DG ---------------------------00008550
 5220   CALL PUSH(H9G)                                                 00008560
        HCMD = H9D                                                     00008570
        GO TO 1700                                                     00008580
C       ------------------------------- .DP, .IP ---------------------00008590
 5250   CONTINUE                                                       00008600
 5251 FORMAT(' MAXIMUM DIVISION=', I9)                                  00008610
        B = HCMD.EQ.H9DP                                               00008620
        IF(B) CALL SETUP                                               00008630
        CALL PUTH(HHC1, LUW)                                           00008640
        CALL PUTH(HHC2, LUW)                                           00008650
        CALL PUTH(HHC3, LUW)                                           00008660
        WRITE(LUW, 1001)                                               00008670
        CALL PUTR(HHZM, ZM)                                            00008680
        CALL PUTR(HHP , P )                                            00008690
        CALL PUTR(HHVV, V )                                            00008700
        WRITE(LUW, 1001)                                               00008710
        IF(B .OR. KDX.EQ.1 ) CALL PUTR(HHDX, DX)                       00008720
        IF(B .OR. KDY.EQ.1 ) CALL PUTR(HHDY, DY)                       00008730
        IF(B .OR. KDX.EQ.2 ) CALL PUTR(HHAX, AX)                       00008740
        IF(B .OR. KDY.EQ.2 ) CALL PUTR(HHAY, AY)                       00008750
        IF(B .OR. KDY.EQ.3 ) CALL PUTR(HHDR, DR)                       00008760
        IF(B .OR. DM.GT.0.0) CALL PUTR(HHDM, DM)                       00008770
        WRITE(LUW, 1001)                                               00008780
        CALL PUTR(HHRR, R )                                            00008790
        IF(B .OR. KG .EQ.1 ) CALL PUTR(HHGG, G )                       00008800
        IF(B .OR. KG .EQ.2 ) CALL PUTR(HHGL, GL)                       00008810
        IF( (B .OR. KG .EQ.3) .AND. GL.GT.0.0) CALL PUTR(HHGT, GT)     00008820
        WRITE(LUW, 1001)                                               00008830
        IF(B) CALL PUTR(HHQE, QE)                                      00008840
        IF(B) WRITE(LUW, 5251) NQN                                     00008850
        GO TO 1500                                                     00008860
C       ------------------------------- .DQ ---------------------------00008870
 5260   CONTINUE                                                       00008880
 5261 FORMAT(                                                           00008890
     +     / ' X, Y LOCATION   =', G13.6, ' ,  ', G13.6, 1X, 2A4)     00008900
 5262 FORMAT(                                                           00008910
     +     / ' X LOCATION      =', G13.6, ' TO ', G13.6, 1X, 2A4      00008920
     +     / ' Y LOCATION      =', G13.6, ' TO ', G13.6, 1X, 2A4 )    00008930
 5265 FORMAT(                                                           00008940
     + '   START         VOLUME/      VOLUME      '                   00008950
     +, '  SOURCE        MASS/AREA    MASS FLOW   '                   00008960
     +/'   TIME          AREA RATE    FLOW RATE   '                   00008970
     +, '  CONCENTR.     RATE         RATE        '                   00008980
     +/1X, 6('  (', 2A4, ') ') )                                      00008990
 5266 FORMAT(1X, 6G13.6)                                                00009000
        DO 5289 IQ = 1, NQ                                             00009010
          IF(KKQX(IQ)-1) 5280, 5271, 5272                              00009020
 5271       WRITE(LUW, 5261) QQXL(IQ), QQYL(IQ), HHQL(6), HHQL(7)      00009030
            GO TO 5275                                                 00009040
 5272       WRITE(LUW, 5262) QQXL(IQ), QQXM(IQ), HHQL(6), HHQL(7)      00009050
     +                      , QQYL(IQ), QQYM(IQ), HHQL(6), HHQL(7)     00009060
 5275       CALL PUTR(HHQA, QQA(IQ) )                                  00009070
            WRITE(LUW, 5265)                                           00009080
     +        HHQT(6), HHQT(7), HHQV(6), HHQV(7), HHQQ(6), HHQQ(7)     00009090
     +      , HHQC(6), HHQC(7), HHQR(6), HHQR(7), HHQM(6), HHQM(7)     00009100
 5280     CONTINUE                                                     00009110
          WRITE(LUW, 5266) QQT (IQ), QQV (IQ), QQ  (IQ)                00009120
     +                   , QQC (IQ), QQR (IQ), QQM (IQ)                00009130
 5289     CONTINUE                                                     00009140
        WRITE(LUW, 1001)                                               00009150
        GO TO 1500                                                     00009160
C       ------------------------------- .FF---------------------------00009170
 5300   CONTINUE                                                       00009180
 5301 FORMAT('1')                                                       00009190
        WRITE(LUW, 5301)                                               00009200
```

```
      GO TO 1500                                                        00009210
C     ================================ .G ================================00009220
 5400 CONTINUE                                                          00009230
      IF(KLX.NE.0 .AND. KLY.NE.0) GO TO 5410                            00009240
      IF(BATCH) GO TO 8120                                              00009250
      CALL PUSH(H9G)                                                    00009260
      IF(KLY.EQ.0) CALL PUSH(HLY)                                       00009270
      IF(KLX.EQ.0) CALL PUSH(HLX)                                       00009280
      GO TO 1500                                                        00009290
 5410 CONTINUE                                                          00009300
      CALL SETUP                                                        00009310
C                                                                       00009320
      NXL = MINO( (NSL - 5)/NSX, MJ)                                    00009330
      NX = NSN                                                          00009340
      IF(NX.LE.0) NX = NXL                                              00009350
      XI = XGI                                                          00009360
      IF(XGM.LE.XGL) GO TO 5425                                         00009370
        IF(XI.GT.0.) NX = (XGM - XGL)/XI + 1.0                          00009380
        IF(XI.LE.0.) XI = (XGM - XGL)/(NX - 1)                          00009390
 5425 CONTINUE                                                          00009400
      K  = ALOG10(AMAX1(ABS(XGL), ABS(XGL + XI*FLOAT(NX) ) ) )          00009410
      I  = 0                                                            00009420
      IF(K.GT.NSX-2) I = K - (NSX - 2)                                  00009430
      IF(K.LT.1    ) I = K - 3                                          00009440
      XS = 10.0**I                                                      00009450
C                                                                       00009460
      YI = YGI                                                          00009470
      IF(YI.LE.0.0) YI = XI                                             00009480
      NY = MAX1( (YGM - YGL)/YI, 0.0) + 1                               00009490
      K  = ALOG10(AMAX1(ABS(YGL), ABS(YGL + YI*FLOAT(NY) ) ) )          00009500
      I  = 0                                                            00009510
      IF(K.GT.3) I = K - 3                                              00009520
      IF(K.LT.1) I = K - 3                                              00009530
      YS = 10.0**I                                                      00009540
C                                                                       00009550
      CM = 10.0**(NSX - 1)                                              00009560
      XL = XGL                                                          00009570
 5440 NX1 = MINO(NX, NXL)                                               00009580
      XC = XL                                                           00009590
      DO 5449 IX = 1, NX1                                               00009600
        JJ(IX) = XC/XS                                                  00009610
        XC = XC + XI                                                    00009620
 5449   CONTINUE                                                        00009630
 5451 FORMAT(' X SCALE          (', G13.6, 1X, 2A4, ')'                 00009640
     +     / ' Y SCALE          (', G13.6, 1X, 2A4, ')'                 00009650
     +     / ' CONCENTRATION    (', G13.6, 1X, 2A4, ')'                 00009660
     +     / )                                                         00009670
 5452 FORMAT(' X SCALE          (', G13.6, 1X, 2A4, ')'                 00009680
     +     / ' Y SCALE          (', G13.6, 1X, 2A4, ')'                 00009690
     +     / ' T SCALE          (', G13.6, 1X, 2A4, ')'                 00009700
     +     / )                                                         00009710
 5453 FORMAT('         X'   /'     Y', 30I3)                          00009720
 5454 FORMAT('          X'  /'     Y', 30I4)                          00009730
 5455 FORMAT('           X' /'     Y', 30I5)                          00009740
 5456 FORMAT('            X'/'     Y', 30I6)                          00009750
       IF(BTSS) GO TO 5458                                              00009760
      B = TC.GE.0.0                                                     00009770
      IF(B) CALL PUTR(HHTC, TC)                                         00009780
      IF(.NOT.B) WRITE(LUW, 5001)                                       00009790
      WRITE(LUW, 5451)                                                  00009800
     +   XS, HHXC(6), HHXC(7)                                          00009810
     +  , YS, HHYC(6), HHYC(7)                                         00009820
     +  , SC, HH9C(7), HH9C(7)                                         00009830
       GO TO 5459                                                       00009840
 5458 CALL PUTR(HHTE,  TE)                                              00009850
      WRITE(LUW, 5452)                                                  00009860
```

```
     +    XS, HHXC(6), HHXC(7)                                              00009870
     +  , YS, HHYC(6), HHYC(7)                                              00009880
     +  , SC, HHTC(6), HHTC(7)                                              00009890
 5459    CONTINUE                                                          00009900
         IF(NSX.EQ.3) WRITE(LUW, 5453) (JJ(IX), IX = 1, NX1)               00009910
         IF(NSX.EQ.4) WRITE(LUW, 5454) (JJ(IX), IX = 1, NX1)               00009920
         IF(NSX.EQ.5) WRITE(LUW, 5455) (JJ(IX), IX = 1, NX1)               00009930
         IF(NSX.EQ.6) WRITE(LUW, 5456) (JJ(IX), IX = 1, NX1)               00009940
         IF(NSY.GT.1) WRITE(LUW, 1001)                                     00009950
         CEM = 0.0                                                         00009960
         NCE = 0                                                           00009970
         NCM = 0                                                           00009980
         YC = YGM                                                          00009990
         DO 5479 IY = 1, NY                                                00010000
           XC = XL                                                         00010010
           DO 5469 IX = 1, NX1                                             00010020
             CALL CALC                                                     00010030
             IF(C) 5465, 5462, 5461                                        00010040
 5461          CEM = AMAX1(CEM, 100.0*CE/C)                               00010050
 5462          C = C/SC                                                   00010060
               IF(C.GE.CM) GO TO 5463                                     00010070
               J = C + 0.5                                                00010080
               GO TO 5467                                                 00010090
 5463        CONTINUE                                                     00010100
               I = ALOG10(C)                                              00010110
               I = I - (NSX - 3)                                          00010120
               J = -I                                                     00010130
               IF(I.LE.9) J = J - 10.0*INT(C/(10.0**I) )                  00010140
               NCM = NCM + 1                                              00010150
               GO TO 5467                                                 00010160
 5465        CONTINUE                                                     00010170
               J = -1                                                     00010180
               NCE = NCE + 1                                              00010190
 5467        CONTINUE                                                     00010200
             JJ(IX) = J                                                   00010210
             XC = XC + XI                                                 00010220
 5469        CONTINUE                                                     00010230
 5473 FORMAT(1X, I5, 30I3)                                                00010240
 5474 FORMAT(1X, I5, 30I4)                                                00010250
 5475 FORMAT(1X, I5, 30I5)                                                00010260
 5476 FORMAT(1X, I5, 30I6)                                                00010270
           JY = YC/YS                                                     00010280
           IF(NSX.EQ.3) WRITE(LUW, 5473) JY, (JJ(IX), IX = 1, NX1)        00010290
           IF(NSX.EQ.4) WRITE(LUW, 5474) JY, (JJ(IX), IX = 1, NX1)        00010300
           IF(NSX.EQ.5) WRITE(LUW, 5475) JY, (JJ(IX), IX = 1, NX1)        00010310
           IF(NSX.EQ.6) WRITE(LUW, 5476) JY, (JJ(IX), IX = 1, NX1)        00010320
           IF(NSY.EQ.2) WRITE(LUW, 1001)                                  00010330
           IF(NSY.EQ.3) WRITE(LUW, 1002)                                  00010340
           IF(NSY.EQ.4) WRITE(LUW, 1003)                                  00010350
           YC = YC - YI                                                   00010360
 5479      CONTINUE                                                       00010370
 5481 FORMAT(' WORST APPROXIMATION = +-', G9.3, '%.')                     00010380
 5482 FORMAT(1X, I5, ' LARGE VALUE(S) IN -(MAGNITUDE+EXPONENT) FORM.')    00010390
 5483 FORMAT(1X, I5, ' SOURCE(S) SHOWN AS "-1".')                         00010400
        IF(CEM.GT.0.0) WRITE(LUW, 5481) CEM                               00010410
        IF(NCM.GT.0  ) WRITE(LUW, 5482) NCM                               00010420
        IF(NCE.GT.0  ) WRITE(LUW, 5483) NCE                               00010430
        WRITE(LUW, 1001)                                                  00010440
        XL = XC                                                           00010450
        NX = NX - NX1                                                     00010460
        IF(NX.GT.0) GO TO 5440                                            00010470
        WRITE(LUW, 1001)                                                  00010480
         BTSS = .FALSE.                                                   00010490
        GO TO 1500                                                        00010500
C       ------------------------------ .I ---------------------------00010510
 5500   CALL PUSH(H9IQ)                                                  00010520
```

```
        HCMD = H9IP                                                          00010530
        GO TO 1700                                                           00010540
C       -------------------------------- .IC ----------------------------00010550
 5510   CALL PUSH(HH9C(1))                                                   00010560
        HCMD = H9I                                                           00010570
        GO TO 1700                                                           00010580
C       -------------------------------- .IG ----------------------------00010590
 5520   CALL PUSH(H9G)                                                       00010600
        HCMD = H9I                                                           00010610
        GO TO 1700                                                           00010620
C       -------------------------------- .IQ ----------------------------00010630
 5560   CONTINUE                                                             00010640
 5561 FORMAT( /                                                             00010650
     + '   X             Y                          '                       00010660
     +, '  START         MASS          SOURCE       '                       00010670
     +/'   LOCATION      LOCATION      AREA         '                       00010680
     +, '  TIME          FLOW RATE     CONCENTR.    ')                      00010690
 5562 FORMAT( /                                                             00010700
     + '   X             Y                          '                       00010710
     +, '  START         MASS/AREA     SOURCE       '                       00010720
     +/'   LOCATION      LOCATION      AREA         '                       00010730
     +, '  TIME          RATE          CONCENTR.    ')                      00010740
 5563 FORMAT( /                                                             00010750
     + '   X             Y                          '                       00010760
     +, '  START         VOLUME        SOURCE       '                       00010770
     +/'   LOCATION      LOCATION      AREA         '                       00010780
     +, '  TIME          FLOW RATE     CONCENTR.    ')                      00010790
 5564 FORMAT( /                                                             00010800
     + '   X             Y                          '                       00010810
     +, '  START         VOLUME/       SOURCE       '                       00010820
     +/'   LOCATION      LOCATION      AREA         '                       00010830
     +, '  TIME          AREA RATE     CONCENTR.    ')                      00010840
 5565 FORMAT(1X, 6('  (', 2A4, ') ') )                                      00010850
 5566 FORMAT(1X, 6G13.6)                                                    00010860
        KQMO = 0                                                            00010870
        DO 5589 IQ = 1, NQ                                                  00010880
          KQX = KKQX(IQ)                                                    00010890
          KQM = KKQM(IQ)                                                    00010900
          GO TO (5571, 5572, 5573, 5574), KQM                               00010910
 5571       IF(KQM.NE.KQMO) WRITE(LUW, 5561)                               00010920
            H6 = HHQM(6)                                                    00010930
            H7 = HHQM(7)                                                    00010940
            Q  = QQM(IQ)                                                    00010950
            GO TO 5580                                                      00010960
 5572       IF(KQM.NE.KQMO) WRITE(LUW, 5562)                               00010970
            H6 = HHQR(6)                                                    00010980
            H7 = HHQR(7)                                                    00010990
            Q  = QQR(IQ)                                                    00011000
            GO TO 5580                                                      00011010
 5573       IF(KQM.NE.KQMO) WRITE(LUW, 5563)                               00011020
            H6 = HHQQ(6)                                                    00011030
            H7 = HHQQ(7)                                                    00011040
            Q  = QQ (IQ)                                                    00011050
            GO TO 5580                                                      00011060
 5574       IF(KQM.NE.KQMO) WRITE(LUW, 5564)                               00011070
            H6 = HHQV(6)                                                    00011080
            H7 = HHQV(7)                                                    00011090
            Q  = QQV(IQ)                                                    00011100
C           GO TO 5580                                                      00011110
 5580     IF(KQM.NE.KQMO) WRITE(LUW, 5565)                                  00011120
     +       HHQL(6), HHQL(7), HHQL(6), HHQL(7), HHQA(6), HHQA(7)           00011130
     +     , HHQT(6), HHQT(7), H6     , H7     , HHQC(6), HHQC(7)           00011140
          WRITE(LUW, 5566) QQXL(IQ), QQYL(IQ), QQA (IQ)                     00011150
     +                   , QQT (IQ), Q      , QQC (IQ)                     00011160
          IF(KQX.EQ.2) WRITE(LUW, 5566) QQXM(IQ), QQYM(IQ)                  00011170
          KQMO = KQM                                                        00011180
```

```
 5589     CONTINUE                                                          00011190
         WRITE(LUW, 1001)                                                  00011200
         GO TO 1500                                                        00011210
C        ================================ .T ============================00011220
 5600     IF(KTE .GT. 0 .AND. KXC .NE. 0 .AND. KYC .NE. 0) GO TO 5610      00011230
          IF(BATCH) GO TO 8120                                             00011240
          CALL PUSH(HH9T(1))                                               00011250
          IF(KYC .EQ. 0) CALL PUSH(HHYC(1))                                00011260
          IF(KXC .EQ. 0) CALL PUSH(HHXC(1))                                00011270
          IF(KTE .EQ. 0) CALL PUSH(HHTE(1))                                00011280
          GO TO 1500                                                       00011290
 5610     CONTINUE                                                         00011300
          BTSS = .TRUE.                                                    00011310
          CALL SETUP                                                       00011320
          XC = X9C                                                         00011330
          YC = Y9C                                                         00011340
          CALL PUTR(HHTE, TE)                                              00011350
          CALL PUTR(HHXC, XC)                                              00011360
          CALL PUTR(HHYC, YC)                                              00011370
          CALL CALC                                                        00011380
          CALL PUTR(HH9T, C)                                               00011390
          WRITE(LUW, 1001)                                                 00011400
          BTSS = .FALSE.                                                   00011410
          GO TO 1500                                                       00011420
C        ================================ .TG ===========================00011430
 5620    IF(KTE .GT. 0) GO TO 5630                                         00011440
          IF(BATCH) GO TO 8120                                             00011450
          CALL PUSH(H9TG)                                                  00011460
          HCMD = HHTE(1)                                                   00011470
          GO TO 1700                                                       00011480
 5630     CONTINUE                                                         00011490
          BTSS = .TRUE.                                                    00011500
          HCMD = H9G                                                       00011510
          GO TO 1700                                                       00011520
C        ================================ ERRORS ========================00011530
C        -------------------------------- PARAMETER ERROR (SERIOUS) -----00011540
 8110    CONTINUE                                                          00011550
         WRITE(LUP, 1081)                                                  00011560
         IF(BATCH) GO TO 1400                                              00011570
         IF(BIP) GO TO 1700                                                00011580
         GO TO 1500                                                        00011590
C        -------------------------------- PARAMETER ERROR (NON-SERIOUS) -00011600
 8120    CONTINUE                                                          00011610
         WRITE(LUP, 1081)                                                  00011620
         IF(BATCH) GO TO 1500                                              00011630
         IF(BIP) GO TO 1700                                                00011640
         GO TO 1500                                                        00011650
C        ================================ END ===========================00011660
         END                                                               00011670
C        ================================ BLOCK DATA ====================00011680
         BLOCK DATA INIT                                                   00011690
C        ================================ DECLARATIONS ==================00011700
         LOGICAL*1      BERR, BATCH,BOP , BOE ,BTSS                       00011710
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                   00011720
         LOGICAL*1      BBT , B   , BIP                                   00011730
         INTEGER*4                                                        00011740
     +     HHCMD                                                          00011750
     +   , H   , HA  , HAR , HHAX, HHAY                                   00011760
     +   , HC  , HD  , HHDX, HHDY, HHDR, HHDM                             00011770
     +   , HE  , HF  , HG  , HHGG, HHGL, HHGT                             00011780
     +   , HH  , HI  , HIP , HIL , HIR                                    00011790
         INTEGER*4                                                        00011800
     +     HL  , HLX , HLY , HM                                           00011810
     +   , HO  , HOD , HOP , HOE , HOT , HOW , HHP                        00011820
     +   , HQ  , HQN , HHQE, HHQL, HHQA, HHQT                             00011830
     +   , HHQM, HHQR, HHQC, HHQQ, HHQV                                   00011840
```

```
     +   , HR   , HHRR, HHRK, HHRD                                                    00011850
         INTEGER*4                                                                    00011860
     +     HS   , HSC , HSL , HSN , HSX , HSY , HHTC , HHTE                         00011870
     +   , HU   , HHU , HHUL, HHUM, HHUT                                            00011880
     +   , HV   , HHVV, HHVG, HHVK, HHVT                                            00011890
     +   , HHVR, HHVI, HHVD, HHVU                                                   00011900
     +   , HXY , HHXC, HHYC, HHZM, HH9C                                             00011910
     +   , HH9T                                                                     00011920
         INTEGER*4                                                                  00011930
     +     H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                                  00011940
     +   , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                                       00011950
     +   , HHC1, HHC2, HHC3                                                         00011960
         INTEGER*4      HHCOM,HCMD, H6   , H7                                       00011970
         INTEGER*4      JJ   , J  , JY                                              00011980
         COMMON /CHAR/                                                              00011990
     +     HHCMD(30)                                                                00012000
     +   , H    , HA   , HAR        , HHAX  (7) , HHAY  (7)                        00012010
     +   , HC   , HD   , HHDX  (7) , HHDY  (7) , HHDR  (7) , HHDM  (7)            00012020
     +   , HE   , HF   , HG         , HHGG  (7) , HHGL  (7) , HHGT  (7)            00012030
     +   , HH   , HI   , HIP , HIL , HIR                                          00012040
     +   , HL   , HLX , HLY , HM                                                  00012050
     +   , HO   , HOD , HOP , HOE , HOT , HOW , HHP   (7)                         00012060
     +   , HQ   , HQN , HHQE  (7) , HHQL  (7) , HHQA  (7) , HHQT  (7)            00012070
     +   , HHQM  (7) , HHQR  (7) , HHQC  (7) , HHQQ  (7) , HHQV  (7)             00012080
     +   , HR         , HHRR  (7) , HHRK  (7) , HHRD  (7)                         00012090
     +   , HS   , HSC , HSL , HSN , HSX , HSY , HHTC  (7) , HHTE  (7)            00012100
     +   , HU         , HHU   (7) , HHUL  (7) , HHUM  (7) , HHUT  (7)             00012110
     +   , HV         , HHVV  (7) , HHVG  (7) , HHVK  (7) , HHVT  (7)             00012120
     +   , HHVR  (7) , HHVI  (7) , HHVD  (7) , HHVU  (7)                          00012130
     +   , HXY        , HHXC  (7) , HHYC  (7) , HHZM  (7) , HH9C  (7)             00012140
     +   , HH9T  (7)                                                              00012150
     +   , H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                                00012160
     +   , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                                     00012170
     +   , HHC1 (18) , HHC2 (18) , HHC3 (18)                                      00012180
         COMMON /BIT /                                                            00012190
     +     BERR, BATCH                                                            00012200
     +   , BOP , BOE ,BTSS                                                        00012210
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                          00012220
         COMMON /INT /                                                            00012230
     +     LUL , LUM , LUR , LUP , LUE , LUW                                      00012240
     +   , MCMD                                                                   00012250
     +   , KDX , KDY , KG  , KLX , KLY , KR                                       00012260
     +   , KTE , KVG , KVP , KVR , KXC , KYC                                      00012270
     +   , NQ   , NQN , NSL , NSN , NSX , NSY                                     00012280
     +   , MQ                                                                     00012290
     +   , KKQX( 10) , KKQM( 10)                                                  00012300
         COMMON /REAL/                                                            00012310
     +     UA   , UD   , UGL , UGT , ULC                                          00012320
     +   , UQL , UQA , UQT , UQM , UQR , UQC , UQQ , UQV                          00012330
     +   , URK , URD , UTC , UVV , UVK , UVT , UVI , UVD , UVU , UZM              00012340
     +   , U9C                                                                    00012350
     +   , AX   , AY   , C    , CE   , DX   , DY   , DR   , DM   , DXT , DYT     00012360
     +   , G    , GL   , GT   , G2   , P    , QE   , QD                          00012370
     +   , R    , RK   , RP   , RD   , SC   , TC   , TD   , TE   , TPHI          00012380
     +   , V    , VN   , VG   , VP   , VK   , VKN , VT   , VTN , VM              00012390
     +   , VR   , VI   , VD   , VU                                               00012400
     +   , XC   , X9C , XGL , XGM , XGI , XD                                     00012410
     +   , YC   , Y9C , YGL , YGM , YGI , YD   , ZM                              00012420
     +   , QQXL( 10) , QQXM( 10) , QQYL( 10) , QQYM( 10)                          00012430
     +   , QQA ( 10) , QQT ( 10) , QQV ( 10) , QQ  ( 10)                          00012440
     +   , QQC ( 10) , QQR ( 10) , QQM ( 10)                                      00012450
C        ================================ CHARACTER ====================00012460
         DATA HHCMD /'I    ',     29*'    '/                                     00012470
         DATA  H     , HA    , HAR  /'    ','A   ','AR  '/                      00012480
         DATA HHAX  /'AX  ','X DI','SPER','SIVI','TY  ','FT  ','    '/          00012490
         DATA HHAY  /'AY  ','Y DI','SPER','SIVI','TY  ','FT  ','    '/          00012500
```

```
      DATA  HC    , HD    /'C   ','D   '/                                        00012510
      DATA HHDX  /'DX  ','X DI','SPER','SION','    ','FT2/','D   '/           00012520
      DATA HHDY  /'DY  ','Y DI','SPER','SION','    ','FT2/','D   '/           00012530
      DATA HHDR  /'DR  ','DISP','ERSI','ON R','ATIO','    ','    '/           00012540
      DATA HHDM  /'DM  ','MOL.',' DIF','FUSI','ON  ','FT2/','D   '/           00012550
      DATA  HE    , HF    , HG    /'E   ','F   ','G   '/                       00012560
      DATA HHGG  /'GG  ','DECA','Y GA','MMA ','    ','    ','    '/           00012570
      DATA HHGL  /'GL  ','DECA','Y LA','MBDA','    ','1/YR','    '/           00012580
      DATA HHGT  /'GT  ','DECA','Y HA','LF-L','IFE ','YR  ','    '/           00012590
      DATA  HH    , HI    /'H   ','I   '/                                      00012600
      DATA  HIP   , HIL   , HIR  /'IP  ','IL  ','IR  '/                       00012610
      DATA  HL    , HLX   , HLY   , HM   /'L   ','LX  ','LY  ','M   '/        00012620
      DATA  HO    , HOD   , HOW  /'O   ','OD  ','OW  '/                       00012630
      DATA  HOP   , HOE   , HOT  /'OP  ','OE  ','OT  '/                       00012640
      DATA HHP   /'P   ','PORO','SITY','    ','    ','    ','    '/           00012650
      DATA  HQ    , HQN  /'Q   ','QN  '/                                       00012660
      DATA HHQL  /'QL  ','SOUR','CE L','OCAT','ION ','FT  ','    '/           00012670
      DATA HHQA  /'QA  ','SOUR','CE A','REA ','    ','FT2 ','    '/           00012680
      DATA HHQT  /'QT  ','SOUR','CE T','IME ','    ','DAYS','    '/           00012690
      DATA HHQM  /'QM  ','MASS',' FLO','W RA','TE  ','LBM/','D   '/           00012700
      DATA HHQR  /'QR  ','MASS','/ARE','A RA','TE  ','LB/F','T2/D'/           00012710
      DATA HHQC  /'QC  ','SOUR','CE C','ONCE','NTR.','MG/L','    '/           00012720
      DATA HHQQ  /'QQ  ','VOLU','ME F','LOW ','RATE','FT3/','D   '/           00012730
      DATA HHQV  /'QV  ','VOLU','ME/A','REA ','RATE','FT/D','    '/           00012740
      DATA HHQE  /'QE  ','ACCU','RACY','    ','    ','    ','    '/           00012750
      DATA  HR   /'R   '/                                                      00012760
      DATA HHRR  /'RR  ','RETA','RDAT','ION ','    ','    ','    '/           00012770
C     DATA HHRK  /'RK  ','SORP','TION',' RAT','E   ','FT3/','LBM '/           00012780
C     DATA HHRD  /'RD  ','AQUI','FER ','DENS','ITY ','LBM/','FT3 '/           00012790
      DATA  HS    , HSC   , HSL  /'S   ','SC  ','SL  '/                       00012800
      DATA  HSX   , HSN   , HSY  /'SX  ','SN  ','SY  '/                       00012810
      DATA HHTC  /'T   ','SAMP','LE T','IME ','    ','DAYS','    '/           00012820
       DATA HHTE  /'TE  ','% ST','EADY',' STA','TE  ','    ','    '/          00012830
      DATA  HU   /'U   '/                                                      00012840
C     DATA HHUL  /'UL  ','LENG','TH  ','    ','    ','FT  ','    '/          00012850
C     DATA HHUM  /'UM  ','MASS','    ','    ','    ','LBM ','    '/          00012860
C     DATA HHUT  /'UT  ','TIME','    ','    ','    ','DAYS','    '/          00012870
      DATA  HV   /'V   '/                                                      00012880
      DATA HHVV  /'VV  ','VELO','CITY','    ','    ','FT/D','    '/          00012890
C     DATA HHVG  /'VG  ','GRAD','IENT','    ','    ','    ','    '/          00012900
C     DATA HHVK  /'VK  ','PERM','EABI','LITY','    ','FT/D','    '/          00012910
C     DATA HHVT  /'VT  ','TRAN','SMIS','SIVI','TY  ','FT2/','D   '/          00012920
C     DATA HHVR  /'VR  ','VELO','CITY','/V.W','ATER','    ','    '/          00012930
C     DATA HHVI  /'VI  ','INTR','INSI','C PE','RM. ','FT2 ','    '/          00012940
C     DATA HHVD  /'VD  ','DENS','ITY ','    ','    ','LBM/','FT3 '/          00012950
C     DATA HHVU  /'VU  ','VISC','OSIT','Y   ','    ','FT3/','LBM '/          00012960
      DATA  HXY  /'XY  '/                                                      00012970
      DATA HHXC  /'X   ','X LO','CATI','ON  ','    ','FT  ','    '/          00012980
      DATA HHYC  /'Y   ','Y LO','CATI','ON  ','    ','FT  ','    '/          00012990
      DATA HHZM  /'Z   ','THIC','KNES','S   ','    ','FT  ','    '/          00013000
      DATA HH9C  /'.C  ','CONC','ENTR','ATIO','N   ','MG/L','    '/          00013010
       DATA HH9T  /'.T  ','STEA','DY S','TATE','    ','DAYS','    '/          00013020
      DATA  H9D   , H9DC  , H9DG /'.D  ','.DC ','.DG '/                       00013030
      DATA  H9DP  , H9DQ        /'.DP ','.DQ '/                               00013040
      DATA  H9FF  , H9G         /'.FF ','.G  '/                               00013050
      DATA  H9I   , H9IC  , H9IG /'.I  ','.IC ','.IG '/                       00013060
      DATA  H9IP  , H9IQ  , H9TG /'.IP ','.IQ ','.TG '/                       00013070
C     ================================ BIT ============================00013080
C     -------------------------------- I/O OPTIONS --------------------00013090
      DATA                                                                     00013100
     +   BATCH                 / .FALSE./                                      00013110
     +  , BOP , BOE ,BTSS      / .TRUE. , .FALSE., .FALSE./                    00013120
     +  , BT1 , BT2 , BT3 , BT4 / .FALSE., .FALSE., .FALSE., .FALSE./          00013130
     +  , BT5 , BT6 , BT7 , BT8 / .FALSE., .FALSE., .FALSE., .FALSE./          00013140
C     ================================ INTEGER ========================00013150
C     -------------------------------- I/O UNITS, DIMENSIONS ---------00013160
```

```
      DATA                                                                  00013170
     +   LUL , LUM             /    3.   10/                               00013180
     +  , LUR , LUP , LUE , LUW /    5.    6.    6.    6/                  00013190
     +  , MCMD, MQ              /   30.   10/                               00013200
C         ------------------------------- FLAGS AND COUNTS -------------00013210
      DATA                                                                  00013220
     +   KDX , KDY , KG  , KLX , KLY /    0.    0.    0.    0.    0/      00013230
     +  , KR  , KTE , KVG , KVP , KVR /    0.    0.    0.    0.    0/      00013240
     +  , KXC , KYC , NQ  , NQN       /    0.    0.    0.  100/           00013250
     +  , NSL , NSN , NSX , NSY       /   80.    0.    6.    2/           00013260
C         ================================ REAL ========================00013270
C         ------------------------------- UNITS, PARAMETERS ------------00013280
      DATA                                                                  00013290
     +   UA  , UD        / 1.0          , 1.0          /                   00013300
     +  , UGL , UGT , ULC / 2.737909E- 3, 365.2422    , 1.0         /     00013310
     +  , UQL , UQA , UQT / 1.0         , 1.0         , 1.0         /     00013320
     +  , UQM , UQR , UQC / 1.0         , 1.0         , 62.42796E- 6/     00013330
     +  , UQQ , UQV       / 1.0         , 1.0         /                   00013340
     +  , URK , URD , UTC / 1.0         , 1.0         , 1.0         /     00013350
     +  , UVV , UVK , UVT / 1.0         , 1.0         , 1.0         /     00013360
     +  , UVI , UVD , UVU / 1.0         , 1.0         , 1.0         /     00013370
     +  , UZM , U9C       / 1.0         , 62.42796E- 6/                   00013380
     +  , DM  , QE  , SC  / 0.0         , 0.1         , 1.0         /     00013390
      END                                                                   00013400
C         ================================ PUSH ========================00013410
      SUBROUTINE PUSH(HO)                                                   00013420
      LOGICAL*1     BERR, BATCH,BOP , BOE ,BTSS                             00013430
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                     00013440
      INTEGER*4                                                             00013450
     +   HHCMD                                                             00013460
     +  , H   , HA  , HAR , HHAX, HHAY                                     00013470
     +  , HC  , HD  , HHDX, HHDY, HHDR, HHDM                               00013480
     +  , HE  , HF  , HG  , HHGG, HHGL, HHGT                               00013490
     +  , HH  , HI  , HIP , HIL , HIR                                      00013500
      INTEGER*4                                                             00013510
     +   HL  , HLX , HLY , HM                                              00013520
     +  , HO  , HOD , HOP , HOE , HOT , HOW , HHP                          00013530
     +  , HQ  , HQN , HHQE, HHQL, HHQA, HHQT                               00013540
     +  , HHQM, HHQR, HHQC, HHQQ, HHQV                                     00013550
     +  , HR  , HHRR, HHRK, HHRD                                           00013560
      INTEGER*4                                                             00013570
     +   HS  , HSC , HSL , HSN , HSX , HSY , HHTC , HHTE                   00013580
     +  , HU  , HHU , HHUL, HHUM, HHUT                                     00013590
     +  , HV  , HHVV, HHVG, HHVK, HHVT                                     00013600
     +  , HHVR, HHVI, HHVD, HHVU                                           00013610
     +  , HXY , HHXC, HHYC, HHZM, HH9C                                     00013620
     +  , HH9T                                                             00013630
      INTEGER*4                                                             00013640
     +   H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                           00013650
     +  , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                               00013660
     +  , HHC1, HHC2, HHC3                                                 00013670
      INTEGER*4     HO  , H1  , H2                                          00013680
      COMMON /CHAR/                                                         00013690
     +   HHCMD(30)                                                         00013700
     +  , H   , HA  , HAR       , HHAX  (7) , HHAY  (7)                   00013710
     +  , HC  , HD  , HHDX  (7) , HHDY  (7) , HHDR  (7) , HHDM  (7)       00013720
     +  , HE  , HF  , HG        , HHGG  (7) , HHGL  (7) , HHGT  (7)       00013730
     +  , HH  , HI  , HIP , HIL , HIR                                     00013740
     +  , HL  , HLX , HLY , HM                                            00013750
     +  , HO  , HOD , HOP , HOE , HOT , HOW , HHP   (7)                   00013760
     +  , HQ  , HQN , HHQE  (7) , HHQL  (7) , HHQA  (7) , HHQT  (7)       00013770
     +  , HHQM  (7) , HHQR  (7) , HHQC  (7) , HHQQ  (7) , HHQV  (7)       00013780
     +  , HR        , HHRR  (7) , HHRK  (7) , HHRD  (7)                   00013790
     +  , HS  , HSC , HSL , HSN , HSX , HSY , HHTC  (7) , HHTE  (7)       00013800
     +  , HU        , HHU   (7) , HHUL  (7) , HHUM  (7) , HHUT  (7)       00013810
     +  , HV        , HHVV  (7) , HHVG  (7) , HHVK  (7) , HHVT  (7)       00013820
```

```
     +   , HHVR  (7) , HHVI  (7) , HHVD  (7) , HHVU  (7)                    00013830
     +   , HXY       , HHXC  (7) , HHYC  (7) , HHZM  (7) , HH9C  (7)        00013840
     +   , HH9T  (7)                                                        00013850
     +   , H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                          00013860
     +   , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                               00013870
     +   , HHC1 (18) , HHC2 (18) , HHC3 (18)                                00013880
         COMMON /BIT /                                                      00013890
     +     BERR, BATCH                                                      00013900
     +   , BOP , BOE , BTSS                                               00013910
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00013920
         COMMON /INT /                                                      00013930
     +     LUL , LUM , LUR , LUP , LUE , LUW                              00013940
     +   , MCMD                                                           00013950
     +   , KDX , KDY , KG  , KLX , KLY , KR                               00013960
     +   , KTE , KVG , KVP , KVR , KXC , KYC                              00013970
     +   , NQ  , NQN , NSL , NSN , NSX , NSY                              00013980
     +   , MQ                                                             00013990
     +   , KKQX( 10) , KKQM( 10)                                          00014000
         H1 = HO                                                          00014010
         DO 5900 I = 1, MCMD                                              00014020
           H2 = HHCMD(I)                                                  00014030
           HHCMD(I) = H1                                                  00014040
           IF(H1.EQ.H) RETURN                                             00014050
           H1 = H2                                                        00014060
 5900      CONTINUE                                                       00014070
         WRITE(LUP, 9001) MCMD, H1                                        00014080
 9001 FORMAT(' STACK SPACE OF', I5, ' COMMANDS EXCEEDED.'                 00014090
     +     / ' COMMAND ', A4, ' LOST.')                                   00014100
         END                                                              00014110
C        ================================ GETI ============================00014120
         SUBROUTINE GETI(KO, LO, MO)                                      00014130
         LOGICAL*1     BERR, BATCH,BOP , BOE , BTSS                       00014140
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00014150
         COMMON /BIT /                                                    00014160
     +     BERR, BATCH                                                    00014170
     +   , BOP , BOE , BTSS                                               00014180
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00014190
         COMMON /INT /                                                    00014200
     +     LUL , LUM , LUR , LUP , LUE , LUW                              00014210
     +   , MCMD                                                           00014220
     +   , KDX , KDY , KG  , KLX , KLY , KR                               00014230
     +   , KTE , KVG , KVP , KVR , KXC , KYC                              00014240
     +   , NQ  , NQN , NSL , NSN , NSX , NSY                              00014250
     +   , MQ                                                             00014260
     +   , KKQX( 10) , KKQM( 10)                                          00014270
 1022 FORMAT(10I5)                                                        00014280
 1032 FORMAT(1X, 10I5)                                                    00014290
         READ(LUR, 1022) K1                                               00014300
         IF(BOE) WRITE(LUE, 1032) K1                                      00014310
         BERR = .TRUE.                                                    00014320
         IF(LO.LT.MO .AND. (K1.LT.LO .OR. K1.GT.MO) ) RETURN              00014330
         BERR = .FALSE.                                                   00014340
         KO = K1                                                          00014350
         RETURN                                                           00014360
         END                                                              00014370
C        ================================ GETR ============================00014380
         SUBROUTINE GETR(HHO, RO, BO)                                     00014390
         LOGICAL*1     BERR, BATCH,BOP , BOE , BTSS                       00014400
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00014410
         LOGICAL*1     BO                                                 00014420
         INTEGER*4     HHO , H                                            00014430
         DIMENSION HHO(7)                                                 00014440
         COMMON /BIT /                                                    00014450
     +     BERR, BATCH                                                    00014460
     +   , BOP , BOE , BTSS                                               00014470
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00014480
```

```
      COMMON /INT /                                                     00014490
     +   LUL , LUM , LUR , LUP , LUE , LUW                              00014500
     +  , MCMD                                                          00014510
     +  , KDX , KDY , KG  , KLX , KLY , KR                            00014520
     +  , KTE , KVG , KVP , KVR , KXC , KYC                           00014530
     +  , NQ  , NQN , NSL , NSN , NSX , NSY                           00014540
     +  , MQ                                                           00014550
     +  , KKQX( 10) , KKQM( 10)                                        00014560
      DATA H /'    '/                                                 00014570
 1011 FORMAT(1X, 4A4, ' (',2A4, '):' / ' ?')                          00014580
 1012 FORMAT(1X, 4A4, ' (UNITLESS):' / ' ?')                          00014590
 1023 FORMAT(E15.0)                                                   00014600
 1033 FORMAT(1X, G15.7)                                               00014610
      IF(.NOT.BOP) GO TO 3000                                         00014620
      IF(HHO(6).EQ.H) GO TO 2500                                      00014630
      WRITE(LUP, 1011) (HHO(I), I = 2, 7)                             00014640
      GO TO 3000                                                      00014650
 2500 WRITE(LUP, 1012) (HHO(I), I = 2, 5)                             00014660
 3000 READ(LUR, 1023) R1                                              00014670
      IF(BOE) WRITE(LUE, 1033) R1                                     00014680
      BERR = .TRUE.                                                   00014690
      IF(BO .AND. R1.LE.0) RETURN                                     00014700
      BERR = .FALSE.                                                  00014710
      RO = R1                                                         00014720
      RETURN                                                          00014730
      END                                                             00014740
C     ================================ PUTR ============================00014750
      SUBROUTINE PUTR(HHO, RO)                                        00014760
      LOGICAL*1     BERR, BATCH,BOP , BOE , BTSS                      00014770
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8               00014780
      INTEGER*4     HHO                                               00014790
      DIMENSION HHO(7)                                                00014800
      COMMON /BIT /                                                   00014810
     +   BERR, BATCH                                                  00014820
     +  , BOP , BOE , BTSS                                           00014830
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8              00014840
      COMMON /INT /                                                   00014850
     +   LUL , LUM , LUR , LUP , LUE , LUW                            00014860
     +  , MCMD                                                        00014870
     +  , KDX , KDY , KG  , KLX , KLY , KR                           00014880
     +  , KTE , KVG , KVP , KVR , KXC , KYC                          00014890
     +  , NQ  , NQN , NSL , NSN , NSX , NSY                          00014900
     +  , MQ                                                          00014910
     +  , KKQX( 10) , KKQM( 10)                                       00014920
 1001 FORMAT(1X, 4A4, '=', G13.6, 1X, 2A4)                           00014930
      WRITE(LUW, 1001) (HHO(I), I = 2, 5), RO, HHO(6), HHO(7)         00014940
      RETURN                                                          00014950
      END                                                             00014960
C     ================================ PUTH ============================00014970
      SUBROUTINE PUTH(HHO, LUO)                                       00014980
      LOGICAL*1     BERR, BATCH,BOP , BOE , BTSS                      00014990
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8               00015000
      INTEGER*4     HHO , H                                           00015010
      DIMENSION HHO(18)                                               00015020
      COMMON /BIT /                                                   00015030
     +   BERR, BATCH                                                  00015040
     +  , BOP , BOE , BTSS                                           00015050
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8              00015060
      COMMON /INT /                                                   00015070
     +   LUL , LUM , LUR , LUP , LUE , LUW                            00015080
     +  , MCMD                                                        00015090
     +  , KDX , KDY , KG  , KLX , KLY , KR                           00015100
     +  , KTE , KVG , KVP , KVR , KXC , KYC                          00015110
     +  , NQ  , NQN , NSL , NSN , NSX , NSY                          00015120
     +  , MQ                                                          00015130
     +  , KKQX( 10) , KKQM( 10)                                       00015140
```

```
      DATA H /'    '/                                                    00015150
 1001 FORMAT(1X, 18A4)                                                   00015160
      N = 18                                                             00015170
      DO 2900 I = 1, 17                                                  00015180
        IF(HHO(N).NE.H) GO TO 3000                                       00015190
        N = N - 1                                                        00015200
 2900   CONTINUE                                                         00015210
 3000 CONTINUE                                                           00015220
      WRITE(LUO, 1001) (HHO(I), I = 1, N)                                00015230
      RETURN                                                             00015240
      END                                                                00015250
C     ================================ SETUP ==========================00015260
      SUBROUTINE SETUP                                                   00015270
      LOGICAL*1     BERR, BATCH,BOP , BOE , BTSS                         00015280
     +, BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                    00015290
      COMMON /BIT /                                                      00015300
     +  BERR, BATCH                                                      00015310
     +, BOP , BOE , BTSS                                                 00015320
     +, BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                    00015330
      COMMON /INT /                                                      00015340
     +  LUL , LUM , LUR , LUP , LUE , LUW                                00015350
     +, MCMD                                                             00015360
     +, KDX , KDY , KG  , KLX , KLY , KR                                 00015370
     +, KTE , KVG , KVP , KVR , KXC , KYC                                00015380
     +, NQ  , NQN , NSL , NSN , NSX , NSY                                00015390
     +, MQ                                                               00015400
     +, KKQX( 10) , KKQM( 10)                                            00015410
      COMMON /REAL/                                                      00015420
     +  UA  , UD  , UGL , UGT , ULC                                       00015430
     +, UQL , UQA , UQT , UQM , UQR , UQC , UQQ , UQV                    00015440
     +, URK , URD , UTC , UVV , UVK , UVT , UVI , UVD , UVU , UZM        00015450
     +, U9C                                                              00015460
     +, AX  , AY  , C   , CE  , DX  , DY  , DR  , DM  , DXT , DYT        00015470
     +, G   , GL  , GT  , G2  , P   , QE  , QD                           00015480
     +, R   , RK  , RP  , RD  , SC  , TC  , TD  , TE  , TPHI             00015490
     +, V   , VN  , VG  , VP  , VK  , VKN , VT  , VTN , VM               00015500
     +, VR  , VI  , VD  , VU                                             00015510
     +, XC  , X9C , XGL , XGM , XGI , XD                                 00015520
     +, YC  , Y9C , YGL , YGM , YGI , YD  , ZM                           00015530
     +, QQXL( 10) , QQXM( 10) , QQYL( 10) , QQYM( 10)                    00015540
     +, QQA ( 10) , QQT ( 10) , QQV ( 10) , QQ  ( 10)                    00015550
     +, QQC ( 10) , QQR ( 10) , QQM ( 10)                                00015560
C     -------------------------------- CALCULATE BASIC PARAMETERS ----00015570
      V1 = V *UVV                                                        00015580
      V2 = V1*V1                                                         00015590
      IF(KDX.EQ.2 ) DX = AX*UA*V1/    UD                                 00015600
      IF(KDX.EQ.1 ) AX = DX*UD   /(V1*UA)                                00015610
      IF(KDY.EQ.3 ) DY = DX      / DR                                    00015620
      IF(KDY.EQ.2 ) DY = AY*UA*V1/    UD                                 00015630
      IF(KDY.NE.2 ) AY = DY*UD   /(V1*UA)                                00015640
      IF(KDY.LT.3 ) DR = DX      / DY                                    00015650
      DXT = DX + DM                                                      00015660
      DYT = DY + DM                                                      00015670
      D1  = DXT*UD                                                       00015680
      D2  = SQRT(D1*DYT*UD)                                              00015690
      IF(KG .EQ.3 ) GL = ALOG(2.0)/(GT*UGT*UGL)                          00015700
      IF(KG .GE.2 ) G  = 1.0 + 4.0*GL*UGL*D1/V2                          00015710
      IF(KG .LT.2 ) GL = (G - 1.0)*V2/(4.0*D1*UGL)                       00015720
      IF(KG .LT.3 .AND. GL.GT.0.) GT = ALOG(2.0)/(GL*UGL*UGT)            00015730
      G2 = SQRT(G)                                                       00015740
C     -------------------------------- NORMALIZING VARIABLES ---------00015750
      XD = D1   /(G2*V1)                                                 00015760
      YD = D2   /(G2*V1)                                                 00015770
      TD = R *D1/(G *V2)                                                 00015780
      QD = P *ZM*UZM*D2                                                  00015790
      RETURN                                                             00015800
```

```
      END                                                                   00015810
C     ================================ CALC ============================00015820
      SUBROUTINE CALC                                                       00015830
      LOGICAL*1     BERR, BATCH,BOP , BOE , BTSS                            00015840
     +, BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                       00015850
      COMMON /BIT /                                                         00015860
     +  BERR, BATCH                                                         00015870
     +, BOP , BOE , BTSS                                                    00015880
     +, BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                       00015890
      COMMON /INT /                                                         00015900
     +  LUL , LUM , LUR , LUP , LUE , LUW                                   00015910
     +, MCMD                                                                00015920
     +, KDX , KDY , KG  , KLX , KLY , KR                                    00015930
     +, KTE , KVG , KVP , KVR , KXC , KYC                                   00015940
     +, NQ  , NQN , NSL , NSN , NSX , NSY                                   00015950
     +, MQ                                                                  00015960
     +, KKQX( 10) , KKQM( 10)                                               00015970
      COMMON /REAL/                                                         00015980
     +  UA  , UD  , UGL , UGT , ULC                                         00015990
     +, UQL , UQA , UQT , UQM , UQR , UQC , UQQ , UQV                       00016000
     +, URK , URD , UTC , UVV , UVK , UVT , UVI , UVD , UVU , UZM           00016010
     +, U9C                                                                 00016020
     +, AX  , AY  , C   , CE  , DX  , DY  , DR  , DM  , DXT , DYT           00016030
     +, G   , GL  , GT  , G2  , P   , QE  , QD                              00016040
     +, R   , RK  , RP  , RD  , SC  , TC  , TD  , TE  , TPHI                00016050
     +, V   , VN  , VG  , VP  , VK  , VKN , VT  , VTN , VM                  00016060
     +, VR  , VI  , VD  , VU                                                00016070
     +, XC  , X9C , XGL , XGM , XGI , XD                                    00016080
     +, YC  , Y9C , YGL , YGM , YGI , YD  , ZM                              00016090
     +, QQXL( 10) , QQXM( 10) , QQYL( 10) , QQYM( 10)                       00016100
     +, QQA ( 10) , QQT ( 10) , QQV ( 10) , QQ  ( 10)                       00016110
     +, QQC ( 10) , QQR ( 10) , QQM ( 10)                                   00016120
      DATA WPI / 3.14159265 /                                               00016130
      X  = XC*ULC                                                           00016140
      Y  = YC*ULC                                                           00016150
      T  = TC*UTC                                                           00016160
      E  = 2.0                                                              00016170
      C  = 0.                                                               00016180
      CE = 0.                                                               00016190
       CR = 0.                                                              00016200
       TM = QQT(1)*UQT                                                      00016210
      DO 5390 IQ = 1, NQ                                                    00016220
        TQ = QQT(IQ)*UQT                                                    00016230
         TM = AMAX1(TM,TQ)                                                  00016240
        IF(T.GE.0. .AND. TQ.GE.T) GO TO 5390                                00016250
        KQX = KKQX(IQ)                                                      00016260
        IF(KQX.LT.0) GO TO 3000                                             00016270
        XL = QQXL(IQ)*UQL                                                   00016280
        YL = QQYL(IQ)*UQL                                                   00016290
        GO TO (2100, 2200), KQX                                             00016300
2100    CONTINUE                                                            00016310
          NX = 1                                                            00016320
          XI = 0.                                                           00016330
          NY = 1                                                            00016340
          YI = 0.                                                           00016350
          S2 = QQA (IQ)*UQA                                                 00016360
          QDA = QD                                                          00016370
          GO TO 2900                                                        00016380
2200    CONTINUE                                                            00016390
          XM = QQXM(IQ)*UQL                                                 00016400
          YM = QQYM(IQ)*UQL                                                 00016410
          RN = AMAX1(X-XM, XL-X, Y-YM, YL-Y)                                00016420
          IF(RN.LE.0.) GO TO 9100                                           00016430
          XI = XM - XL                                                      00016440
          YI = YM - YL                                                      00016450
          XN = NQN                                                          00016460
```

```
            S   = SQRT(AMAX1(50.0*XD*QE*RN/(1.0 + DR), XI*YI/XN) )        00016470
            NX = NQN - MAX1(XN - XI/S, 0.)                               00016480
            XI = XI/FLOAT(NX)                                            00016490
            NY = NQN - MAX1(XN - YI/S, 0.)                               00016500
            YI = YI/FLOAT(NY)                                            00016510
            S2 = AMAX1(XI, YI)                                           00016520
            S2 = S2*S2                                                   00016530
            QDA = FLOAT(NX)*FLOAT(NY)*QD                                 00016540
2900      CONTINUE                                                       00016550
          ER = 0.02*S2*(1.0 + DR)/(XD*XD)                                00016560
          IF(T.GE.0.) ER = ER + 0.2                                      00016570
          QMO = 0.                                                       00016580
3000      CONTINUE                                                       00016590
          QM = QQM(IQ)*UQM                                               00016600
          QN = (QM - QMO)/QDA                                            00016610
          YQ = YL + YI*0.5                                               00016620
          DO 5290 IY = 1, NY                                             00016630
            YN = (Y - YQ)/YD                                             00016640
            XQ = XL + XI*0.5                                             00016650
            DO 5190 IX = 1, NX                                           00016660
              XN = (X - XQ)/XD                                           00016670
              RN = SQRT(XN*XN + YN*YN)                                   00016680
               IF(BTSS) GO TO 5000                                       00016690
              IF(RN.LE.0.) GO TO 9100                                    00016700
              IF(T.LT.0.) GO TO 4500                                     00016710
                TN = (T - TQ)/TD                                         00016720
                WO = (RN - TN)/SQRT(4.0*TN)                              00016730
                   E = ERFC(WO)                                          00016740
4500          CONTINUE                                                   00016750
              CY = EXP( (XN/G2 - RN)*0.5)                                00016760
              CR = 4.0*SQRT(WPI*RN)                                      00016770
              W   = QN*CY*E/CR                                           00016780
5001 FORMAT(1X, 5G15.7)                                                  00016790
              IF(BT5) WRITE(LUW, 5001)                                   00016800
     +          X , XQ, XD, XN, CR                                      00016810
     +        , Y , YQ, YD, YN, CY                                      00016820
     +        , T , TQ, TD, TN, E                                       00016830
     +        , QM, QMO,QDA,QN, W                                       00016840
              C   = C + W                                                00016850
              CE = CE + W*ER/RN                                          00016860
               GO TO 5090                                                00016870
5000          CONTINUE                                                   00016880
              CR = AMAX1(CR, RN)                                         00016890
5090          CONTINUE                                                   00016900
              XQ = XQ + XI                                               00016910
5190          CONTINUE                                                   00016920
            YQ = YQ + YI                                                 00016930
5290        CONTINUE                                                     00016940
           QMO = QM                                                      00016950
5390       CONTINUE                                                     00016960
          IF(BTSS) GO TO 5500                                            00016970
         C   = C/U9C                                                     00016980
         CE = CE/U9C                                                     00016990
          GO TO 5990                                                     00017000
5500      CONTINUE                                                       00017010
          IF(CR .GT. 0.) C = CR / (SQRT(TPHI * TPHI + CR) + TPHI)        00017030
          C = (TM + C * C * TD) / UTC                                    00017040
5990      CONTINUE                                                       00017050
         BERR = .FALSE.                                                  00017060
         RETURN                                                          00017070
9100     CONTINUE                                                        00017080
         C   = -1.0                                                      00017090
         CE = 0.0                                                        00017100
         BERR = .TRUE.                                                   00017110
         RETURN                                                          00017120
         END                                                             00017130
```

```
C       ================================ ERFC ============================00017140
         FUNCTION ERFC(WO)                                                 00017150
        DATA EO, E1, E2, E3, E4, E5                                        00017160
     +   / .32759, .25438, -.28540, 1.42141, -1.45315, 1.06141 /           00017170
        W  = 1.0/(1.0 + EO*ABS(WO) )                                       00017180
        W  = W*(E1 + W*(E2 + W*(E3 + W*(E4 + W*E5) ) ) )                   00017190
        E  = W*EXP(-(WO*WO) )                                              00017200
        IF(WO.LT.O.) E = 2.0 - E                                           00017210
         ERFC = E                                                          00017220
         RETURN                                                            00017230
         END                                                               00017240
```

APPENDIX II-F

PROGRAM SOURCE FOR IBM PC

```
**** TSO FOREGROUND HARDCOPY ****
DSNAME=U11236C.FPLUMEPC.CNTL

C       GROUNDWATER PLUME CALCULATION PROGRAM                                  00000010
C       D.C. KENT, HYDROGEOLOGIST, PRINCIPAL INVESTIGATOR                      00000020
C       FRED WITZ AND LORRAINE LEMASTER, PROGRAMMERS                           00000030
C         GEOLOGY DEPARTMENT, OKLAHOMA STATE UNIVERSITY                        00000040
C         STILLWATER, OKLAHOMA, 74078                                          00000050
C       FORTRAN VERSION (SEE VERSION BELOW)                                    00000060
C       TESTED WITH:                                                           00000070
C         MICROSOFT FORTRAN ON IBM PC (77 STANDARD)                            00000080
C       ================================ DECLARATIONS ==================00000090
        LOGICAL*2     BERR, BATCH,BOP , BOE ,BTSS                              00000100
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                        00000110
        LOGICAL*2     BBT , B   , BIP                                          00000120
        CHARACTER*4                                                            00000130
     +    HHCMD                                                               00000140
     +  , H    , HA   , HAR        , HHAX       , HHAY                        00000150
     +  , HC   , HD   , HHDX       , HHDY       , HHDR       , HHDM           00000160
     +  , HE   , HF   , HG         , HHGG       , HHGL       , HHGT           00000170
     +  , HH   , HI   , HIP , HIL , HIR                                       00000180
        CHARACTER*4                                                            00000190
     +    HL   , HLX , HLY , HM                                               00000200
     +  , HO   , HOD , HOP , HOE , HOT , HOW , HHP                           00000210
     +  , HQ   , HQN , HHQE       , HHQL       , HHQA       , HHQT           00000220
     +  , HHQM       , HHQR       , HHQC       , HHQQ       , HHQV           00000230
     +  , HR         , HHRR       , HHRK       , HHRD                        00000240
        CHARACTER*4                                                            00000250
     +    HS   , HSC , HSL , HSN , HSX , HSY , HHTC       , HHTE             00000260
     +  , HU         , HHU        , HHUL       , HHUM       , HHUT           00000270
     +  , HV         , HHVV       , HHVG       , HHVK       , HHVT           00000280
     +  , HHVR       , HHVI       , HHVD       , HHVU                        00000290
     +  , HXY        , HHXC       , HHYC       , HHZM       , HH9C           00000300
     +  , HH9T                                                               00000310
        CHARACTER*4                                                            00000320
     +    H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                             00000330
     +  , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                                 00000340
     +  , HHC1       , HHC2       , HHC3                                      00000350
        CHARACTER*4   HHCOM,HCMD, H6  , H7                                     00000360
        INTEGER*4     JJ   , J  , JY                                           00000370
        DIMENSION BBT(8), HHCOM(381), KKCOM(19), RRCOM(79), JJ(30)             00000380
        COMMON /CHAR/                                                          00000390
     +    HHCMD(30)                                                           00000400
     +  , H    , HA   , HAR        , HHAX  (7) , HHAY  (7)                    00000410
     +  , HC   , HD   , HHDX  (7) , HHDY  (7) , HHDR  (7) , HHDM  (7)        00000420
     +  , HE   , HF   , HG         , HHGG  (7) , HHGL  (7) , HHGT  (7)        00000430
     +  , HH   , HI   , HIP , HIL , HIR                                       00000440
     +  , HL   , HLX , HLY , HM                                               00000450
     +  , HO   , HOD , HOP , HOE , HOT , HOW , HHP   (7)                     00000460
     +  , HQ   , HQN , HHQE  (7) , HHQL  (7) , HHQA  (7) , HHQT  (7)        00000470
     +  , HHQM  (7) , HHQR  (7) , HHQC  (7) , HHQQ  (7) , HHQV  (7)        00000480
     +  , HR         , HHRR  (7) , HHRK  (7) , HHRD  (7)                     00000490
     +  , HS   , HSC , HSL , HSN , HSX , HSY , HHTC  (7) , HHTE  (7)        00000500
     +  , HU         , HHU   (7) , HHUL  (7) , HHUM  (7) , HHUT  (7)        00000510
     +  , HV         , HHVV  (7) , HHVG  (7) , HHVK  (7) , HHVT  (7)        00000520
     +  , HHVR  (7) , HHVI  (7) , HHVD  (7) , HHVU  (7)                     00000530
     +  , HXY        , HHXC  (7) , HHYC  (7) , HHZM  (7) , HH9C  (7)        00000540
     +  , HH9T  (7)                                                         00000550
     +  , H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                           00000560
     +  , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                               00000570
     +  , HHC1 (18) , HHC2 (18) , HHC3 (18)                                00000580
        COMMON /BIT /                                                          00000590
     +    BERR, BATCH                                                         00000600
     +  , BOP , BOE ,BTSS                                                    00000610
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                     00000620
```

```
      COMMON /INTE/                                                     00000630
     +   LUL , LUM , LUR , LUP , LUE , LUW                              00000640
     +  , MCMD                                                          00000650
     +  , KDX , KDY , KG  , KLX , KLY , KR                            00000660
     +  , KTE , KVG , KVP , KVR , KXC , KYC                           00000670
     +  , NQ  , NQN , NSL , NSN , NSX , NSY                           00000680
     +  , MQ                                                          00000690
     +  , KKQX( 10) , KKQM( 10)                                       00000700
      COMMON /REAL/                                                   00000710
     +   UA  , UD  , UGL , UGT , ULC                                   00000720
     +  , UQL , UQA , UQT , UQM , UQR , UQC , UQQ , UQV               00000730
     +  , URK , URD , UTC , UVV , UVK , UVT , UVI , UVD , UVU , UZM   00000740
     +  , U9C                                                         00000750
     +  , AX  , AY  , C   , CE  , DX  , DY  , DR  , DM  , DXT , DYT   00000760
     +  , G   , GL  , GT  , G2  , P   , QE  , QD                      00000770
     +  , R   , RK  , RP  , RD  , SC  , TC  , TD  , TE  ,TPHI         00000780
     +  , V   , VN  , VG  , VP  , VK  , VKN , VT  , VTN , VM          00000790
     +  , VR  , VI  , VD  , VU                                        00000800
     +  , XC  , X9C , XGL , XGM , XGI , XD                            00000810
     +  , YC  , Y9C , YGL , YGM , YGI , YD  , ZM                      00000820
     +  , QQXL( 10) , QQXM( 10) , QQYL( 10) , QQYM( 10)               00000830
     +  , QQA ( 10) , QQT ( 10) , QQV ( 10) , QQ  ( 10)               00000840
     +  , QQC ( 10) , QQR ( 10) , QQM ( 10)                           00000850
      EQUIVALENCE (HHCOM(1), H), (KKCOM(1), KDX), (RRCOM(1), UA)      00000860
      EQUIVALENCE (BBT(1), BT1)                                       00000870
       DATA NACC, NEXP / 20, 80 /                                     00000880
      DATA MBT, MHCOM, MKCOM, MRCOM, MJ / 8, 381, 19, 76, 30 /        00000890
      DATA BIP /.TRUE./                                               00000900
C     ================================ GLOBAL FORMATS ================00000910
 1001 FORMAT(1X)                                                      00000920
 1002 FORMAT('0')                                                     00000930
 1003 FORMAT('0'/)                                                    00000940
 1021 FORMAT(18A4)                                                    00000950
 1023 FORMAT(5E15.0)                                                  00000960
 1031 FORMAT(1X, 18A4)                                                00000970
 1033 FORMAT(1X, 5G15.7)                                              00000980
 1081 FORMAT(' INVALID OR MISSING VALUE.')                            00000990
C     ================================ BEGIN EXECUTION ===============00001000
 1101 FORMAT(' GROUNDWATER PLUME CALCULATION PROGRAM'                 00001010
     +     / ' D.C. KENT, HYDROGEOLOGIST, PRINCIPAL INVESTIGATOR'     00001020
     +     / ' FRED WITZ AND LORRAINE LEMASTER, PROGRAMMERS'          00001030
     +     / ' GEOLOGY DEPARTMENT, OKLAHOMA STATE UNIVERSITY'         00001040
     +     / ' FORTRAN VERSION 1.2 (1985, JANUARY)' / )               00001050
      CALL INIT                                                       00001060
      WRITE(LUP, 1101)                                                00001070
      GO TO 1500                                                      00001080
C     ================================ INPUT LOOP ====================00001090
 1400 CONTINUE                                                        00001100
 1401 FORMAT(' COMMAND?')                                             00001110
      IF(BOP) WRITE(LUP, 1401)                                        00001120
      READ (LUR, 1021) (HHCMD(I), I = 1, 18)                          00001130
      IF(BOE) CALL PUTH(HHCMD, LUE)                                   00001140
      HHCMD(19) = H                                                   00001150
C     ================================ NEXT COMMAND LOOP: POP ========00001160
 1500 CONTINUE                                                        00001170
      HCMD = HHCMD(1)                                                 00001180
      IF(HCMD.EQ.H) GO TO 1400                                        00001190
      DO 1580 I = 2, MCMD                                             00001200
        HHCMD(I - 1) = HHCMD(I)                                       00001210
        IF(HHCMD(I).EQ.H) GO TO 1590                                  00001220
 1580   CONTINUE                                                      00001230
      HHCMD(MCMD) = H                                                 00001240
 1590 CONTINUE                                                        00001250
C     ================================ RE-EXECUTE LOOP: SIEVE ========00001260
 1700 CONTINUE                                                        00001270
      IF(BT2) CALL PUTH(HHCMD, LUE)                                   00001280
```

```
IF(BT1) WRITE(LUE,1031) HCMD                                    00001290
IF(HCMD.EQ. H       ) GO TO 1400                                00001300
IF(HCMD.EQ. HA      ) GO TO 2100                                00001310
IF(HCMD.EQ.HHAX (1) ) GO TO 2110                                00001320
IF(HCMD.EQ.HHAY (1) ) GO TO 2120                                00001330
IF(HCMD.EQ. HAR     ) GO TO      2330                           00001340
IF(HCMD.EQ. HC      ) GO TO 2200                                00001350
IF(HCMD.EQ. HD      ) GO TO 2300                                00001360
IF(HCMD.EQ.HHDX (1) ) GO TO 2310                                00001370
IF(HCMD.EQ.HHDY (1) ) GO TO 2320                                00001380
IF(HCMD.EQ.HHDR (1) ) GO TO 2330                                00001390
IF(HCMD.EQ.HHDM (1) ) GO TO 2340                                00001400
IF(HCMD.EQ. HE      ) GO TO 2400                                00001410
IF(HCMD.EQ. HG      ) GO TO 2500                                00001420
IF(HCMD.EQ.HHGG (1) ) GO TO 2510                                00001430
IF(HCMD.EQ.HHGL (1) ) GO TO 2520                                00001440
IF(HCMD.EQ.HHGT (1) ) GO TO 2530                                00001450
IF(HCMD.EQ. HH      ) GO TO 2600                                00001460
IF(HCMD.EQ. HI      ) GO TO 2700                                00001470
IF(HCMD.EQ. HIP     ) GO TO 2710                                00001480
IF(HCMD.EQ. HIL     ) GO TO 2720                                00001490
IF(HCMD.EQ. HIR     ) GO TO 2730                                00001500
IF(HCMD.EQ. HL      ) GO TO 2800                                00001510
IF(HCMD.EQ. HLX     ) GO TO 2810                                00001520
IF(HCMD.EQ. HLY     ) GO TO 2820                                00001530
IF(HCMD.EQ. HM      ) GO TO      4700                           00001540
IF(HCMD.EQ. HO      ) GO TO 3000                                00001550
IF(HCMD.EQ. HOD     ) GO TO 3010                                00001560
IF(HCMD.EQ. HOW     ) GO TO 3020                                00001570
IF(HCMD.EQ. HOP     ) GO TO 3030                                00001580
IF(HCMD.EQ. HOE     ) GO TO 3040                                00001590
IF(HCMD.EQ. HOT     ) GO TO 3050                                00001600
IF(HCMD.EQ.HHP  (1) ) GO TO 3100                                00001610
IF(HCMD.EQ. HQ      ) GO TO 3500                                00001620
IF(HCMD.EQ.HHQE (1) ) GO TO 3810                                00001630
IF(HCMD.EQ. HQN     ) GO TO 3820                                00001640
IF(HCMD.EQ.HHQM (1) ) GO TO 3900                                00001650
IF(HCMD.EQ. HR      ) GO TO 4100                                00001660
IF(HCMD.EQ.HHRR (1) ) GO TO 4100                                00001670
IF(HCMD.EQ. HS      ) GO TO 4200                                00001680
IF(HCMD.EQ. HSC     ) GO TO 4210                                00001690
IF(HCMD.EQ. HSL     ) GO TO 4220                                00001700
IF(HCMD.EQ. HSN     ) GO TO 4230                                00001710
IF(HCMD.EQ. HSX     ) GO TO 4240                                00001720
IF(HCMD.EQ. HSY     ) GO TO 4250                                00001730
IF(HCMD.EQ.HHTC (1) ) GO TO 4300                                00001740
 IF(HCMD.EQ.HHTE (1) ) GO TO 4400                               00001750
IF(HCMD.EQ. HV      ) GO TO 4500                                00001760
IF(HCMD.EQ.HHVV (1) ) GO TO 4500                                00001770
IF(HCMD.EQ. HXY     ) GO TO 4600                                00001780
IF(HCMD.EQ.HHXC (1) ) GO TO 4610                                00001790
IF(HCMD.EQ.HHYC (1) ) GO TO 4620                                00001800
IF(HCMD.EQ.HHZM (1) ) GO TO 4700                                00001810
IF(HCMD.EQ.HH9C (1) ) GO TO 5100                                00001820
 IF(HCMD.EQ.HH9T (1) ) GO TO      5600                          00001830
IF(HCMD.EQ. H9D     ) GO TO 5200                                00001840
IF(HCMD.EQ. H9DC    ) GO TO 5210                                00001850
IF(HCMD.EQ. H9DG    ) GO TO 5220                                00001860
IF(HCMD.EQ. H9DP    ) GO TO 5250                                00001870
IF(HCMD.EQ. H9DQ    ) GO TO 5260                                00001880
IF(HCMD.EQ. H9FF    ) GO TO 5300                                00001890
IF(HCMD.EQ. H9G     ) GO TO 5400                                00001900
IF(HCMD.EQ. H9I     ) GO TO 5500                                00001910
IF(HCMD.EQ. H9IC    ) GO TO 5510                                00001920
IF(HCMD.EQ. H9IG    ) GO TO 5520                                00001930
IF(HCMD.EQ. H9IP    ) GO TO      5250                           00001940
```

```
      IF(HCMD.EQ. H9IQ    ) GO TO 5560                                  00001950
       IF(HCMD.EQ. H9TG    ) GO TO 5620                                 00001960
      WRITE(LUP, 1801) HCMD                                            00001970
 1801 FORMAT(1X, A4, '?')                                              00001980
      GO TO 1400                                                       00001990
C     ================================ COMMAND EXITS ================00002000
C     ================================ INPUT AND MISC. ==============00002010
C     -------------------------------- A -----------------------------00002020
 2100 CONTINUE                                                         00002030
 2101 FORMAT(' 1 FOR X AND Y DISPERSIVITY (AX, AY),'                   00002040
     +     / ' 2 FOR X DISPERSIVITY AND DISPERSION RATIO (AX, DR):'    00002050
     +     / ' ?')                                                     00002060
      IF(BOP) WRITE(LUP, 2101)                                         00002070
      CALL GETI(ICMD, 1, 2)                                            00002080
      IF(BERR) GO TO 8110                                              00002090
      IF(ICMD.EQ.1) CALL PUSH(HHAY(1) )                                00002100
      IF(ICMD.EQ.2) CALL PUSH(HHDR(1) )                                00002110
C     GO TO 2110                                                       00002120
C     -------------------------------- AX ----------------------------00002130
 2110 CALL GETR(HHAX, AX, .TRUE.)                                      00002140
      IF(BERR) GO TO 8120                                              00002150
      KDX = 2                                                          00002160
      GO TO 1500                                                       00002170
C     -------------------------------- AY ----------------------------00002180
 2120 CALL GETR(HHAY, AY, .TRUE.)                                      00002190
      IF(BERR) GO TO 8120                                              00002200
      KDY = 2                                                          00002210
      GO TO 1500                                                       00002220
C     -------------------------------- C -----------------------------00002230
 2200 CONTINUE                                                         00002240
 2201 FORMAT(' THREE TITLE LINES:'/ ' ?')                              00002250
 2202 FORMAT(' ?')                                                     00002260
      IF(BOP) WRITE(LUP, 2201)                                         00002270
      READ (LUR, 1021) HHC1                                            00002280
      IF(BOE) CALL PUTH(HHC1, LUE)                                     00002290
      IF(BOP) WRITE(LUP, 2202)                                         00002300
      READ (LUR, 1021) HHC2                                            00002310
      IF(BOE) CALL PUTH(HHC2, LUE)                                     00002320
      IF(BOP) WRITE(LUP, 2202)                                         00002330
      READ (LUR, 1021) HHC3                                            00002340
      IF(BOE) CALL PUTH(HHC3, LUE)                                     00002350
      GO TO 1500                                                       00002360
C     -------------------------------- D = D OR A --------------------00002370
 2300 CONTINUE                                                         00002380
 2301 FORMAT(' 1 FOR X AND Y DISPERSION (DX, DY),'                     00002390
     +     / ' 2 FOR X DISPERSION AND DISPERSION RATIO (DX, DR),'      00002400
     +     / ' 3 FOR X AND Y DISPERSIVITY (AX, AY),'                   00002410
     +     / ' 4 FOR X DISPERSIVITY AND DISPERSION RATIO (AX, DR),'    00002420
     +     / '   (USE DM FOR MOLECULAR DIFFUSION):'                    00002430
     +     / ' ?')                                                     00002440
      IF(BOP) WRITE(LUP, 2301)                                         00002450
      CALL GETI(ICMD, 1, 4)                                            00002460
      IF(BERR) GO TO 8110                                              00002470
      IF(ICMD.EQ.1) CALL PUSH(HHDY(1) )                                00002480
      IF(ICMD.EQ.2) CALL PUSH(HHDR(1) )                                00002490
      IF(ICMD.EQ.3) CALL PUSH(HHAY(1) )                                00002500
      IF(ICMD.EQ.4) CALL PUSH(HHDR(1) )                                00002510
      GO TO (2310, 2310, 2110, 2110), ICMD                             00002520
C     -------------------------------- DX ----------------------------00002530
 2310 CALL GETR(HHDX, DX, .TRUE.)                                      00002540
      IF(BERR) GO TO 8120                                              00002550
      KDX = 1                                                          00002560
      GO TO 1500                                                       00002570
C     -------------------------------- DY ----------------------------00002580
 2320 CALL GETR(HHDY, DY, .TRUE.)                                      00002590
      IF(BERR) GO TO 8120                                              00002600
```

```
      KDY = 1                                                          00002610
      GO TO 1500                                                       00002620
C     -------------------------------- DR ----------------------------00002630
 2330 CALL GETR(HHDR, DR, .TRUE.)                                      00002640
      IF(BERR) GO TO 8120                                              00002650
      KDY = 3                                                          00002660
      GO TO 1500                                                       00002670
C     -------------------------------- DM ----------------------------00002680
 2340 CALL GETR(HHDM, DM, .TRUE.)                                      00002690
      IF(BERR) GO TO 8120                                              00002700
      GO TO 1500                                                       00002710
C     -------------------------------- E -----------------------------00002720
 2400 CONTINUE                                                         00002730
 2401 FORMAT(' END OF PLUME PROGRAM.')                                 00002740
 2402 FORMAT(' GOODBYE.')                                              00002750
      WRITE(LUE, 2401)                                                 00002760
      WRITE(LUP, 2402)                                                 00002770
      STOP                                                             00002780
C     -------------------------------- G -----------------------------00002790
 2500 CONTINUE                                                         00002800
 2501 FORMAT(' 0 FOR NO DECAY,'                                        00002810
     +     / ' 1 FOR DECAY COEFFICIENT, GAMMA (GG),'                   00002820
     +     / ' 2 FOR DECAY LAMBDA (GL),'                               00002830
     +     / ' 3 FOR DECAY HALF-LIFE (GT):'                            00002840
     +     / ' ?')                                                     00002850
      IF(BOP) WRITE(LUP, 2501)                                         00002860
      CALL GETI(ICMD, 0, 3)                                            00002870
      IF(BERR) GO TO 8110                                              00002880
      ICMD = ICMD + 1                                                  00002890
      GO TO (2505, 2510, 2520, 2530), ICMD                             00002900
 2505 G  = 1.0                                                         00002910
      KG = 1                                                           00002920
      GO TO 1500                                                       00002930
C     -------------------------------- GG ----------------------------00002940
 2510 CALL GETR(HHGG, XC, .FALSE.)                                     00002950
      IF(XC.LT.1.0) GO TO 8120                                         00002960
      G  = XC                                                          00002970
      KG = 1                                                           00002980
      GO TO 1500                                                       00002990
C     -------------------------------- GL ----------------------------00003000
 2520 CALL GETR(HHGL, XC, .FALSE.)                                     00003010
      IF(XC.LT.0.0) GO TO 8120                                         00003020
      GL = XC                                                          00003030
      KG = 2                                                           00003040
      GO TO 1500                                                       00003050
C     -------------------------------- GT ----------------------------00003060
 2530 CALL GETR(HHGT, GT, .TRUE.)                                      00003070
      IF(BERR) GO TO 8120                                              00003080
      GL = 1.0                                                         00003090
      KG = 3                                                           00003100
      GO TO 1500                                                       00003110
C     -------------------------------- H -----------------------------00003120
 2600 CONTINUE                                                         00003130
 2601 FORMAT(                                                          00003140
     +  ' INPUT:             OUTPUT:'                                 00003150
     +/ ' C  CASE TITLE     .I  INPUT PARAM.'                         00003160
     +/ ' M  THICKNESS      .D  ALL PARAM.'                           00003170
     +/ ' P  POROSITY       .C  SINGLE'                               00003180
     +/ ' V  VELOCITY       .G  GRID MAP'                             00003190
     +/ ' D  DISPERSION     .FF PAGE PRINT'                           00003200
     +/ ' A  DISPERSIVITY'                                            00003210
     +/ ' R  RETARDATION    SPECIAL:'                                 00003220
     +/ ' G  DECAY          H  HELP'                                  00003230
     +/ ' Q  SOURCE         E  EXIT'                                  00003240
     +/ ' T  SAMPLE TIME    I  INPUT'                                 00003250
     +/ ' TE STEADY STATE   O  OUTPUT'                                00003260
```

```
     +/ ' XY SINGLE X,Y'                                                    00003270
     +/ ' L  GRID LIMITS'                                                   00003280
     +/ ' S  GRID SCALES')                                                  00003290
        WRITE(LUP, 2601)                                                    00003300
        GO TO 1500                                                          00003310
C       ------------------------------ I ------------------------------00003320
 2700   CONTINUE                                                           00003330
 2701 FORMAT( / '  1 TO PROMPT FOR ALL REQUIRED PARAMETERS (IP),'          00003340
     +     / '  2 TO LOAD PREVIOUSLY SAVED PARAMETERS (IL),'               00003350
     +     / '  3 TO READ ALL INPUT FROM ANOTHER SOURCE (IR),'             00003360
     +     / ' -1 TO SET OUTPUT PARAMETERS (O):'                          00003370
     +     / ' ?')                                                        00003380
        IF(BOP) WRITE(LUP, 2701)                                            00003390
        CALL GETI(ICMD, -1, 3)                                              00003400
        IF(BERR) GO TO 8110                                                 00003410
        IF(ICMD.EQ. 0) GO TO 8110                                           00003420
        IF(ICMD.EQ. 1) GO TO 2710                                           00003430
        IF(ICMD.EQ. 2) GO TO 2720                                           00003440
        IF(ICMD.EQ. 3) HCMD = HIR                                           00003450
        IF(ICMD.EQ.-1) HCMD = HO                                            00003460
        CALL PUSH(HI)                                                       00003470
        GO TO 1700                                                          00003480
C       ------------------------------ IP -----------------------------00003490
C       (LAST COMMAND MUST RESET BIP)                                       00003500
 2710   BIP = .TRUE.                                                       00003510
        KLX = 0                                                             00003520
        KLY = 0                                                             00003530
        KXC = 0                                                             00003540
        KYC = 0                                                             00003550
        NQ  = 0                                                             00003560
        CALL PUSH(HD)                                                       00003570
        CALL PUSH(HG)                                                       00003580
        CALL PUSH(HR)                                                       00003590
        CALL PUSH(HHTC(1) )                                                 00003600
        CALL PUSH(HQ)                                                       00003610
         CALL PUSH(HL)                                                      00003620
        CALL PUSH(HV)                                                       00003630
        CALL PUSH(HHP (1) )                                                 00003640
        CALL PUSH(HHZM(1) )                                                 00003650
        HCMD = HC                                                           00003660
        GO TO 1700                                                          00003670
C       ------------------------------ IL -----------------------------00003680
 2720   CONTINUE                                                           00003690
 2721 FORMAT(' LOAD FILE UNIT (', I3, ' TO', I3, '):'/ ' ?')               00003700
        WRITE(LUP, 2721) LUL, LUM                                           00003710
        CALL GETI(LU, LUL, LUM)                                             00003720
        IF(BERR) GO TO 8120                                                 00003730
        READ (LU) I, IX, IY, IQ                                             00003740
        IF(     I .NE.MHCOM                                                 00003750
     +    .OR. IX.NE.MKCOM                                                  00003760
     +    .OR. IY.NE.MRCOM                                                  00003770
     +    .OR. IQ.NE.MQ                                                     00003780
     +    ) GO TO 8120                                                      00003790
        READ (LU) HHCOM                                                     00003800
        READ (LU) KKCOM, KKQX, KKQM                                         00003810
        READ (LU) RRCOM                                                     00003820
        READ (LU) QQXL, QQXM, QQYL, QQYM                                    00003830
     +          , QQA , QQT , QQV , QQ                                     00003840
     +          , QQC , QQR , QQM                                          00003850
        ENDFILE LU                                                          00003860
        BIP = .FALSE.                                                       00003870
        GO TO 1500                                                          00003880
C       ------------------------------ IR -----------------------------00003890
 2730   CONTINUE                                                           00003900
 2731 FORMAT(' INPUT UNIT (1 TO', I3, '):'/ ' ?')                           00003910
        IF(BOP) WRITE(LUP, 2731) LUM                                        00003920
```

```
      IF(BOP) WRITE(LUP, 4211)                                          00007230
      READ (LUR, 1023) XC                                               00007240
      IF(BOE) WRITE(LUE, 1033) XC                                       00007250
      IF(XC.LE.0.0) GO TO 8120                                          00007260
      SC = 1.0/XC                                                       00007270
      GO TO 1500                                                        00007280
C     ------------------------------ SL ----------------------------00007290
 4220 CONTINUE                                                          00007300
 4221 FORMAT(' GRID MAP LINE SIZE (40-255 CHARACTERS/LINE):'/ ' ?')     00007310
      IF(BOP) WRITE(LUP, 4221)                                          00007320
      CALL GETI(ICMD, 40, 255)                                          00007330
      IF(BERR) GO TO 8120                                               00007340
      NSL = ICMD                                                        00007350
      GO TO 1500                                                        00007360
C     ------------------------------ SN ----------------------------00007370
 4230 CONTINUE                                                          00007380
 4231 FORMAT(' GRID MAP NODES PER LINE (0 FOR FULL LINE):'/ ' ?')       00007390
      IF(BOP) WRITE(LUP, 4231)                                          00007400
      CALL GETI(NSN, 0, 0)                                              00007410
      GO TO 1500                                                        00007420
C     ------------------------------ SX ----------------------------00007430
 4240 CONTINUE                                                          00007440
 4241 FORMAT(' GRID MAP X SPACING (3 TO 6 CHARACTERS/NODE):'/ ' ?')     00007450
      IF(BOP) WRITE(LUP, 4241)                                          00007460
      CALL GETI(ICMD, 3, 6)                                             00007470
      IF(BERR) GO TO 8120                                               00007480
      NSX = ICMD                                                        00007490
      GO TO 1500                                                        00007500
C     ------------------------------ SY ----------------------------00007510
 4250 CONTINUE                                                          00007520
 4251 FORMAT(' GRID MAP Y SPACING (1 TO 4 LINES/NODE):'/ ' ?')          00007530
      IF(BOP) WRITE(LUP, 4251)                                          00007540
      CALL GETI(ICMD, 1, 4)                                             00007550
      IF(BERR) GO TO 8120                                               00007560
      NSY = ICMD                                                        00007570
      GO TO 1500                                                        00007580
C     ------------------------------ T -----------------------------00007590
 4300 CONTINUE                                                          00007600
 4301 FORMAT(' -1 FOR STEADY STATE,')                                   00007610
      IF(BOP) WRITE(LUP, 4301)                                          00007620
      CALL GETR(HHTC, TC, .FALSE.)                                      00007630
      BIP = .FALSE.                                                     00007640
      GO TO 1500                                                        00007650
C     ============================== TE ============================00007660
 4400  CALL GETR(HHTE, XC, .TRUE.)                                      00007670
       IF(BERR) GO TO 8120                                              00007680
       IF(XC .GT. 100.) GO TO 8120                                      00007690
       KTE = 1                                                          00007700
       YC = XC/50.                                                      00007710
       P0 = 0.                                                          00007720
       P2 = P0                                                          00007730
       IF(ERFC(P0) - YC) 4425, 4480, 4426                               00007740
 4425  P0 = -1.                                                         00007750
       GO TO 4429                                                       00007760
 4426  P0 = 1.                                                          00007770
 4429  CONTINUE                                                         00007780
       DO 4448 I = 1, NEXP                                              00007790
         IF((ERFC(P0) - YC)*P0) 4449, 4480, 4445                        00007800
 4445  P2 = P0                                                          00007810
       P0 = P0 * 2.                                                     00007820
 4448  CONTINUE                                                         00007830
       GO TO 8120                                                       00007840
 4449  P1 = AMIN1(P0,P2)                                                00007850
       P2 = AMAX1(P0,P2)                                                00007860
       DO 4468 I = 1, NACC                                              00007870
         P0 = (P1 + P2)/2.0                                             00007880
```

```
      CALL GETI(LU, 1, LUM)                                                00003930
      IF(BERR) GO TO 8120                                                  00003940
      LUR = LU                                                             00003950
      GO TO 1500                                                           00003960
C     ------------------------------- L ----------------------------------00003970
 2800 CALL PUSH(HLY )                                                      00003980
C     GO TO 2810                                                           00003990
C     ------------------------------- LX ---------------------------------00004000
 2810 CONTINUE                                                             00004010
 2811 FORMAT(' GRID MAP X LOCATIONS (',2A4,'),'                            00004020
     +     / ' MINIMUM, (MAXIMUM), (INTERVAL):'                           00004030
     +     / ' ?')                                                        00004040
      IF(BOP) WRITE(LUP, 2811) HHXC(6), HHXC(7)                            00004050
      READ (LUR, *) XGL, XGM, XGI                                          00004060
      IF(BOE) WRITE(LUE, 1033) XGL, XGM, XGI                               00004070
      KLX = 0                                                              00004080
      IF(XGM.LE.XGL .AND. XGI.LE.0.0) GO TO 8120                           00004090
      KLX = 1                                                              00004100
      GO TO 1500                                                           00004110
C     ------------------------------- LY ---------------------------------00004120
 2820 CONTINUE                                                             00004130
 2821 FORMAT(' GRID MAP Y LOCATIONS (',2A4,'),'                            00004140
     +     / ' MINIMUM, MAXIMUM, (INTERVAL):'                             00004150
     +     / ' ?')                                                        00004160
      IF(BOP) WRITE(LUP, 2821) HHYC(6), HHYC(7)                            00004170
      READ (LUR, *) YGL, YGM, YGI                                          00004180
      IF(BOE) WRITE(LUE, 1033) YGL, YGM, YGI                               00004190
      KLY = 1                                                              00004200
      GO TO 1500                                                           00004210
C     ------------------------------- O ----------------------------------00004220
 3000 CONTINUE                                                             00004230
 3001 FORMAT(' 1 TO DUMP ALL PARAMETERS TO DISK (OD),'                     00004240
     +     / ' 2 TO SET LOCATION FOR WRITING RESULTS (OW),'               00004250
     +     / ' 3 TO SET PROMPTING (OP),'                                  00004260
     +     / ' 4 TO SET ECHO (OE),'                                       00004270
     +     / ' 5 TO SET TRACE (OT),'                                      00004280
     +     / ' ?')                                                        00004290
      IF(BOP) WRITE(LUP, 3001)                                             00004300
      CALL GETI(ICMD, 1, 5)                                                00004310
      IF(BERR) GO TO 8110                                                  00004320
      GO TO (3010, 3020, 3030, 3040, 3050), ICMD                           00004330
C     ------------------------------- OD ---------------------------------00004340
 3010 CONTINUE                                                             00004350
      IF(BIP) GO TO 8110                                                   00004360
 3011 FORMAT(' DUMP FILE UNIT (', I3, ' TO', I3, '):'/ ' ?')               00004370
      WRITE(LUP, 3011) LUL, LUM                                            00004380
      CALL GETI(LU, LUL, LUM)                                              00004390
      IF(BERR) GO TO 8120                                                  00004400
      WRITE(LU) MHCOM, MKCOM, MRCOM, MQ                                    00004410
      WRITE(LU) HHCOM                                                      00004420
      WRITE(LU) KKCOM, KKQX, KKQM                                          00004430
      WRITE(LU) RRCOM                                                      00004440
      WRITE(LU) QQXL, QQXM, QQYL, QQYM                                     00004450
     +        , QQA , QQT , QQV , QQ                                      00004460
     +        , QQC , QQR , QQM                                           00004470
      ENDFILE LU                                                           00004480
      CLOSE(LU)                                                            00004490
      GO TO 1500                                                           00004500
C     ------------------------------- OW ---------------------------------00004510
 3020 CONTINUE                                                             00004520
 3021 FORMAT(' 1 TO' I3, ' TO SET RESULTS UNIT:'/ ' ?')                    00004530
      IF(BOP) WRITE(LUP, 3021) LUM                                         00004540
      CALL GETI(LU, 1, LUM)                                                00004550
      IF(BERR) GO TO 8120                                                  00004560
      LUW = LU                                                             00004570
      GO TO 1500                                                           00004580
```

```
C       ------------------------------- OP ----------------------------00004590
 3030   CONTINUE                                                        00004600
 3031 FORMAT(' -1 FOR NO PROMPTING,'                                     00004610
     +      / '  0 FOR PROMPTING,'                                      00004620
     +      / '  1 TO' I3, ' TO SET PROMPTING UNIT:'/ ' ?')             00004630
        IF(BOP) WRITE(LUP, 3031) LUM                                     00004640
        CALL GETI(LU, -1, LUM)                                           00004650
        IF(BERR) GO TO 8120                                              00004660
        BOP = LU.GE.0                                                    00004670
        IF(LU.GT.0) LUP = LU                                             00004680
        GO TO 1500                                                       00004690
C       ------------------------------- OE ----------------------------00004700
 3040   CONTINUE                                                        00004710
 3041 FORMAT(' -1 FOR NO ECHO,'                                          00004720
     +      / '  0 FOR ECHO,'                                           00004730
     +      / '  1 TO' I3, ' TO SET ECHO UNIT:'/ ' ?')                  00004740
        IF(BOP) WRITE(LUP, 3041) LUM                                     00004750
        CALL GETI(LU, -1, LUM)                                           00004760
        IF(BERR) GO TO 8120                                              00004770
        BOE = LU.GE.0                                                    00004780
        IF(LU.GT.0) LUE = LU                                             00004790
        GO TO 1500                                                       00004800
C       ------------------------------- OT ----------------------------00004810
 3050   CONTINUE                                                        00004820
 3051 FORMAT(' + FOR TRACE, - FOR NO TRACE:'                             00004830
     +      / ' 1 FOR COMMAND     TRACE,'                               00004840
     +      / ' 2 FOR STACK       TRACE,'                               00004850
     +      / ' 5 FOR CALCULATION TRACE:'/ ' ?')                        00004860
        IF(BOP) WRITE(LUP, 3051)                                         00004870
        CALL GETI(ICMD, -MBT, MBT)                                       00004880
        IF(BERR) GO TO 8120                                              00004890
        LU = IABS(ICMD)                                                  00004900
        IF(LU.NE. 0) BBT(LU) = ICMD.GT.0                                 00004910
        GO TO 1500                                                       00004920
C       ------------------------------- P ----------------------------00004930
 3100   CALL GETR(HHP , XC, .TRUE.)                                     00004940
        IF(BERR) GO TO 8120                                              00004950
        IF(XC.GT.1.) GO TO 8120                                          00004960
        P = XC                                                           00004970
        GO TO 1500                                                       00004980
C       ============================== Q ============================00004990
 3500   CONTINUE                                                        00005000
 3511 FORMAT(' CURRENTLY USING', I4, ' OF', I4, ' TIME STEPS.'           00005010
     +      /'  0 TO END SOURCE INFORMATION,'                           00005020
     +      /'  1 TO ADD POINT SOURCE,'                                 00005030
     +      /'  2 TO ADD NON-POINT SOURCE,'                             00005040
     +      /' -N TO DELETE LAST N TIME STEPS:'                         00005050
     +      /' ?')                                                      00005060
 3512 FORMAT(' **** BEGIN SOURCE INFORMATION ****')                      00005070
 3513 FORMAT(' **** END SOURCE INFORMATION ****')                        00005080
        WRITE(LUP, 3512)                                                 00005090
 3510   IF(BOP) WRITE(LUP, 3511) NQ, MQ                                 00005100
        CALL GETI(ICMD, 0, 0)                                            00005110
        IF(ICMD.LT.0) GO TO 3520                                         00005120
        IF(ICMD.EQ.1) GO TO 3530                                         00005130
        IF(ICMD.EQ.2) GO TO 3540                                         00005140
        IF(NQ  .LE.0) GO TO 3790                                         00005150
        IF(ICMD.EQ.0) GO TO 3515                                         00005160
        GO TO 3790                                                       00005170
 3515   WRITE(LUP, 3513)                                                00005180
        GO TO 1500                                                       00005190
C       ------------------------------- REDUCE -----------------------00005200
 3520   NQ = MAXO(NQ + ICMD, 0)                                         00005210
        GO TO 3510                                                       00005220
C       ------------------------------- POINT ------------------------00005230
 3530   IF(NQ.GE.MQ) GO TO 3799                                         00005240
```

```
 3531 FORMAT(' X LOCATION, Y LOCATION (', 2A4, '):'/ ' ?')                00005250
       IF(BOP) WRITE(LUP, 3531)  HHQL(6), HHQL(7)                         00005260
       READ (LUR, *) QXL, QYL                                             00005270
       IF(BOE) WRITE(LUE, 1033) QXL, QYL                                  00005280
       CALL GETR(HHQA, QA, .FALSE.)                                       00005290
C      IF(QA.LT.0.) GO TO 3790                                            00005300
       KQX = 1                                                            00005310
       QXM = QXL                                                          00005320
       QYM = QYL                                                          00005330
       GO TO 3600                                                         00005340
C      -------------------------------- NON-POINT ----------------------00005350
 3540  IF(NQ.GE.MQ) GO TO 3799                                            00005360
 3541 FORMAT(' X LOCATION MINIMUM, MAXIMUM (', 2A4, '):'/ ' ?')           00005370
       IF(BOP) WRITE(LUP, 3541) HHQL(6), HHQL(7)                          00005380
       READ (LUR, *) QXL, QXM                                             00005390
       IF(BOE) WRITE(LUE, 1033) QXL, QXM                                  00005400
       IF(QXM.LE.QXL) GO TO 3790                                          00005410
 3542 FORMAT(' Y LOCATION MINIMUM, MAXIMUM (', 2A4, '):'/ ' ?')           00005420
       IF(BOP) WRITE(LUP, 3542) HHQL(6), HHQL(7)                          00005430
       READ (LUR, *) QYL, QYM                                             00005440
       IF(BOE) WRITE(LUE, 1033) QYM, QYM                                  00005450
       IF(QYM.LE.QYL) GO TO 3790                                          00005460
       KQX = 2                                                            00005470
       QA = (QXM - QXL)*(QYM - QYL)*UQL*UQL/UQA                           00005480
C      -------------------------------- TIME AND RATE -----------------00005490
 3600  CONTINUE                                                           00005500
 3601 FORMAT(' 1 FOR MASS FLOW RATE    (', 2A4, '),'                      00005510
     +     / ' 2 FOR MASS/AREA RATE    (', 2A4, '),'                    00005520
     +     / ' 3 FOR VOLUME FLOW RATE  (', 2A4, '),'                    00005530
     +     / '         AND CONCENTRATION (', 2A4, '),'                  00005540
     +     / ' 4 FOR VOLUME/AREA RATE  (', 2A4, '),'                    00005550
     +     / '         AND CONCENTRATION (', 2A4, '):'                  00005560
     +     / ' ?')                                                      00005570
       IF(BOP) WRITE(LUP, 3601)                                           00005580
     +   HHQM(6), HHQM(7), HHQR(6), HHQR(7)                             00005590
     +  , HHQQ(6), HHQQ(7), HHQC(6), HHQC(7)                            00005600
     +  , HHQV(6), HHQV(7), HHQC(6), HHQC(7)                            00005610
       CALL GETI(KQM, 1, 4)                                               00005620
       IF(BERR) GO TO 3790                                                00005630
       QTO = -1E20                                                        00005640
       QAU = QA*UQA                                                       00005650
       QV  = 0.0                                                          00005660
       Q   = 0.0                                                          00005670
       QC  = 0.0                                                          00005680
       QR  = 0.0                                                          00005690
 3700  CONTINUE                                                           00005700
       GO TO (3710, 3720, 3730, 3740), KQM                                00005710
 3710  CONTINUE                                                           00005720
 3711 FORMAT(' TIME, MASS FLOW RATE, (CONCENTRATION):'/ ' ?')             00005730
       IF(BOP) WRITE(LUP, 3711)                                           00005740
       READ (LUR, *) QT, QM, QC                                           00005750
       IF(BOE) WRITE(LUE, 1033) QT, QM, QC                                00005760
       Q  = 0.                                                            00005770
       IF(QC.GT.0.0) Q  = QM*UQM/(QC*UQC    )                             00005780
       IF(QA.LE.0.0) GO TO 3715                                           00005790
       QR = QM*UQM/(QAU   *UQR)                                           00005800
       QV = Q     /(QAU   *UQV)                                           00005810
 3715  Q  = Q /UQQ                                                        00005820
       GO TO 3760                                                         00005830
 3720  CONTINUE                                                           00005840
       IF(QA.LE.0.0) GO TO 3790                                           00005850
 3721 FORMAT(' TIME, MASS/AREA RATE, (CONCENTRATION):'/ ' ?')             00005860
       IF(BOP) WRITE(LUP, 3721)                                           00005870
       READ (LUR, *) QT, QR, QC                                           00005880
       IF(BOE) WRITE(LUE, 1033) QT, QR, QC                                00005890
       QM = QR*UQR*QAU                                                    00005900
```

```
      IF(QC.LE.0.0) GO TO 3725                                          00005910
      Q  = QM    /(QC*UQC*UQQ)                                          00005920
      QV = Q *UQQ/(QAU   *UQV)                                          00005930
      GO TO 3750                                                        00005940
 3725 Q  = 0.                                                           00005950
      QV = 0.                                                           00005960
      GO TO 3750                                                        00005970
 3730 CONTINUE                                                          00005980
 3731 FORMAT(' TIME, VOLUME FLOW RATE, CONCENTRATION:'/ ' ?')           00005990
      IF(BOP) WRITE(LUP, 3731)                                          00006000
      READ (LUR, *) QT, Q, QC                                           00006010
      IF(BOE) WRITE(LUE, 1033) QT, Q, QC                                00006020
      QM = Q *UQQ*QC*UQC                                                00006030
      IF(QA.LE.0.0) GO TO 3750                                          00006040
      QR = QM    /(QAU   *UQR)                                          00006050
      QV = Q *UQQ/(QAU   *UQV)                                          00006060
      GO TO 3750                                                        00006070
 3740 CONTINUE                                                          00006080
      IF(QA.LE.0.0) GO TO 3790                                          00006090
 3741 FORMAT(' TIME, VOLUME/AREA RATE, CONCENTRATION:'/ ' ?')           00006100
      IF(BOP) WRITE(LUP, 3741)                                          00006110
      READ (LUR, *) QT, QV, QC                                          00006120
      IF(BOE) WRITE(LUE, 1033) QT, QV, QC                               00006130
      Q  = QV*UQV*QAU   /UQQ                                            00006140
      QR = QV*UQV*QC*UQC                                                00006150
      QM = QR    *QAU                                                   00006160
      QR = QR/UQR                                                       00006170
 3750 CONTINUE                                                          00006180
      QM = QM/UQM                                                       00006190
 3760 CONTINUE                                                          00006200
      IF(QT.LE.QTO) GO TO 3510                                          00006210
      IF(QM.LT.0.) GO TO 3790                                           00006220
C     IF(QC.LT.0.) GO TO 3790                                           00006230
      IF(NQ.GE.MQ) GO TO 3790                                           00006240
      NQ = NQ + 1                                                       00006250
      KKQX(NQ) = KQX                                                    00006260
      KKQM(NQ) = KQM                                                    00006270
      QQXL(NQ) = QXL                                                    00006280
      QQXM(NQ) = QXM                                                    00006290
      QQYL(NQ) = QYL                                                    00006300
      QQYM(NQ) = QYM                                                    00006310
      QQT (NQ) = QT                                                     00006320
      QQA (NQ) = QA                                                     00006330
      QQV (NQ) = QV                                                     00006340
      QQ  (NQ) = Q                                                      00006350
      QQC (NQ) = QC                                                     00006360
      QQR (NQ) = QR                                                     00006370
      QQM (NQ) = QM                                                     00006380
      QTO = QT                                                          00006390
      KQX = ISIGN(KQX, -1)                                              00006400
      ICMD = MIN1(QTO, 0.0)                                             00006410
 3771 FORMAT(1X,'ENTER', I5, ' TO RETURN TO MAIN SOURCE MENU;')         00006420
      IF(BOP) WRITE(LUP, 3771) ICMD                                     00006430
      GO TO 3700                                                        00006440
C     ------------------------------- ERRORS ------------------------00006450
 3790 WRITE(LUP, 1081)                                                  00006460
 3799 IF(BATCH) GO TO 1400                                              00006470
      GO TO 3510                                                        00006480
C     =============================== QE ============================00006490
 3810 CALL GETR(HHQE, QE, .TRUE.)                                       00006500
      IF(BERR) GO TO 8120                                               00006510
      GO TO 1500                                                        00006520
C     ------------------------------- QN ----------------------------00006530
 3820 CONTINUE                                                          00006540
 3821 FORMAT(' MAXIMUM NUMBER OF SUBAREAS PER SOURCE:'                  00006550
     +     / ' ?')                                                     00006560
```

```
          IF(ERFC(PO) - YC) 4465, 4480, 4466                                00007890
 4465     P2 = PO                                                           00007900
          GO TO 4467                                                        00007910
 4466     P1 = PO                                                           00007920
 4467     CONTINUE                                                          00007930
 4468     CONTINUE                                                          00007940
         PO = (P1 + P2)/2.0                                                 00007950
 4480    CONTINUE                                                           00007960
         TE = XC                                                            00007970
         TPHI = PO                                                          00007980
         GO TO 1500                                                         00007990
C        ================================ V ==============================00008000
 4500    CALL GETR(HHVV, V , .TRUE.)                                        00008010
         IF(BERR) GO TO 8120                                                00008020
         GO TO 1500                                                         00008030
C        -------------------------------- XY ------------------------------00008040
 4600    CALL PUSH(HHYC(1))                                                 00008050
         HCMD = HHXC(1)                                                     00008060
         GO TO 1700                                                         00008070
C        -------------------------------- X -------------------------------00008080
 4610    CONTINUE                                                           00008090
         CALL GETR(HHXC, X9C, .FALSE.)                                      00008100
         KXC = 1                                                            00008110
         GO TO 1500                                                         00008120
C        -------------------------------- Y -------------------------------00008130
 4620    CONTINUE                                                           00008140
         CALL GETR(HHYC, Y9C, .FALSE.)                                      00008150
         KYC = 1                                                            00008160
         GO TO 1500                                                         00008170
C        -------------------------------- ZM ------------------------------00008180
 4700    CALL GETR(HHZM, ZM, .TRUE.)                                        00008190
         IF(BERR) GO TO 8120                                                00008200
         GO TO 1500                                                         00008210
C        ================================ OUTPUT COMMANDS ================00008220
 5001 FORMAT(' SAMPLE TIME      = STEADY STATE')                           00008230
C        -------------------------------- .C ------------------------------00008240
 5100    CONTINUE                                                           00008250
 5101 FORMAT(18X, '  +-', G9.3, 1X, 2A4)                                    00008260
 5102 FORMAT(' SAMPLE LOCATION WITHIN SOURCE.')                             00008270
         IF(KXC.NE.0 .AND. KYC.NE.0) GO TO 5110                             00008280
         IF(BATCH) GO TO 8120                                               00008290
         CALL PUSH(HH9C(1))                                                 00008300
         IF(KYC.EQ.0) CALL PUSH(HHYC(1))                                    00008310
         IF(KXC.EQ.0) CALL PUSH(HHXC(1))                                    00008320
         GO TO 1500                                                         00008330
 5110    CONTINUE                                                           00008340
         CALL SETUP                                                         00008350
         XC = X9C                                                           00008360
         YC = Y9C                                                           00008370
         B = TC.GE.0.0                                                      00008380
         IF(B) CALL PUTR(HHTC, TC)                                          00008390
         IF(.NOT.B) WRITE(LUW, 5001)                                        00008400
         CALL PUTR(HHXC, XC)                                                00008410
         CALL PUTR(HHYC, YC)                                                00008420
         CALL CALC                                                          00008430
         IF(.NOT.BERR) CALL PUTR(HH9C, C )                                  00008440
         IF(CE.GT.0.0) WRITE(LUW, 5101) CE , HH9C(6), HH9C(7)               00008450
         IF(BERR      ) WRITE(LUW, 5102)                                    00008460
         WRITE(LUW, 1001)                                                   00008470
         GO TO 1500                                                         00008480
C        -------------------------------- .D ------------------------------00008490
 5200    CALL PUSH(H9DQ)                                                    00008500
         HCMD = H9DP                                                        00008510
         GO TO 1700                                                         00008520
C        -------------------------------- .DC -----------------------------00008530
 5210    CALL PUSH(HH9C(1))                                                 00008540
```

```
        IF(BOP) WRITE(LUP, 3821)                                        00006570
        CALL GETI(ICMD, 0, 0)                                           00006580
        IF(ICMD.LE.0) GO TO 8120                                        00006590
        NQN = ICMD                                                      00006600
        GO TO 1500                                                      00006610
C       ================================ QM ============================00006620
 3900   CONTINUE                                                        00006630
C3901 FORMAT(                                                           00006640
C    + '        X            Y            START        MASS     '      00006650
C    +/' LINE   LOCATION     LOCATION     TIME         FLOW RATE'      00006660
C    +/ 6X, 4('  (', 2A4, ') ') )                                       00006670
C3902 FORMAT(1X, I4, 1X, 4G13.6)                                        00006680
C       LU = LUW                                                        00006690
C       IF(.NOT. BOP) GO TO 3920                                        00006700
C       LU = LUP                                                        00006710
C3910   WRITE(LU, 3901) HHQL(6), HHQL(7), HHQL(6), HHQL(7)             00006720
C    +                 , HHQT(6), HHQT(7), HHQM(6), HHQM(7)           00006730
C       DO 3919 IQ = 1, NQ                                              00006740
C         WRITE(LU, 3902) IQ, QQXL(IQ), QQYL(IQ), QQT(IQ), QQM(IQ)      00006750
C3919     CONTINUE                                                      00006760
 3920   CONTINUE                                                        00006770
 3921 FORMAT(' -1 TO LIST FLOW RATES,'                                  00006780
     +     / '  0 TO END QM,'                                           00006790
     +     / ' TIME STEP (LINE) NUMBER TO CHANGE FLOW RATE:'           00006800
     +     / ' ?')                                                     00006810
        IF(BOP) WRITE(LUP, 3921)                                        00006820
        CALL GETI(ICMD, -1, NQ)                                         00006830
        IF(BERR) GO TO 3990                                             00006840
C       IF(ICMD) 3910, 1500, 3950                                       00006850
        IF(ICMD) 3930, 1500, 3950                                       00006860
 3930   CALL PUSH(HCMD)                                                 00006870
        HCMD = H9IQ                                                     00006880
        GO TO 1700                                                      00006890
 3950   CALL GETR(HHQM, QM, .FALSE.)                                    00006900
        IF(QM.LT.0.) GO TO 3990                                         00006910
        KKQM(ICMD) = 1                                                  00006920
        QQM (ICMD) = QM                                                 00006930
        QQR (ICMD) = 0.                                                 00006940
        QQ  (ICMD) = 0.                                                 00006950
        QQV (ICMD) = 0.                                                 00006960
        GO TO 3920                                                      00006970
C       ------------------------------ ERRORS ------------------------00006980
 3990   WRITE(LUP, 1081)                                                00006990
        IF(BATCH) GO TO 1400                                            00007000
        GO TO 3920                                                      00007010
C       ================================ R =============================00007020
 4100   CALL GETR(HHRR, XC, .FALSE.)                                    00007030
        IF(XC.LT.1.0) GO TO 8120                                        00007040
        R = XC                                                          00007050
        GO TO 1500                                                      00007060
C       ================================ R =============================00007070
 4200   CONTINUE                                                        00007080
 4201 FORMAT(' GRID MAP SCALE PARAMETERS:'                              00007090
     +     / ' 1 TO SET CONCENTRATION/STEADY STATE SCALE (SC),'        00007100
     +     / ' 2 TO SET LINE SIZE (SL),'                               00007110
     +     / ' 3 TO SET NODES PER LINE (SN),'                          00007120
     +     / ' 4 TO SET X SPACING (SX),'                               00007130
     +     / ' 5 TO SET Y SPACING (SY):'                               00007140
     +     / ' ?')                                                     00007150
        IF(BOP) WRITE(LUP, 4201)                                        00007160
        CALL GETI(ICMD, 1, 5)                                           00007170
        IF(BERR) GO TO 8110                                             00007180
        GO TO (4210, 4220, 4230, 4240, 4250), ICMD                      00007190
C       ------------------------------ SC ----------------------------00007200
 4210   CONTINUE                                                        00007210
 4211 FORMAT(' GRID MAP MULTIPLIER:'/ ' ?')                             00007220
```

```
        HCMD = H9D                                                              00008550
        GO TO 1700                                                              00008560
C       ------------------------------ .DG ---------------------------00008570
 5220   CALL PUSH(H9G)                                                         00008580
        HCMD = H9D                                                              00008590
        GO TO 1700                                                              00008600
C       ------------------------------ .DP, .IP ----------------------00008610
 5250   CONTINUE                                                               00008620
 5251 FORMAT(' MAXIMUM DIVISION=', I9)                                          00008630
        B = HCMD.EQ.H9DP                                                        00008640
        IF(B) CALL SETUP                                                        00008650
        CALL PUTH(HHC1, LUW)                                                    00008660
        CALL PUTH(HHC2, LUW)                                                    00008670
        CALL PUTH(HHC3, LUW)                                                    00008680
        WRITE(LUW, 1001)                                                        00008690
        CALL PUTR(HHZM, ZM)                                                     00008700
        CALL PUTR(HHP , P )                                                     00008710
        CALL PUTR(HHVV, V )                                                     00008720
        WRITE(LUW, 1001)                                                        00008730
        IF(B .OR. KDX.EQ.1 ) CALL PUTR(HHDX, DX)                                00008740
        IF(B .OR. KDY.EQ.1 ) CALL PUTR(HHDY, DY)                                00008750
        IF(B .OR. KDX.EQ.2 ) CALL PUTR(HHAX, AX)                                00008760
        IF(B .OR. KDY.EQ.2 ) CALL PUTR(HHAY, AY)                                00008770
        IF(B .OR. KDY.EQ.3 ) CALL PUTR(HHDR, DR)                                00008780
        IF(B .OR. DM.GT.0.0) CALL PUTR(HHDM, DM)                                00008790
        WRITE(LUW, 1001)                                                        00008800
        CALL PUTR(HHRR, R )                                                     00008810
        IF(B .OR. KG .EQ.1 ) CALL PUTR(HHGG, G )                                00008820
        IF(B .OR. KG .EQ.2 ) CALL PUTR(HHGL, GL)                                00008830
        IF( (B .OR. KG .EQ.3) .AND. GL.GT.0.0) CALL PUTR(HHGT, GT)              00008840
        WRITE(LUW, 1001)                                                        00008850
        IF(B) CALL PUTR(HHQE, QE)                                               00008860
        IF(B) WRITE(LUW, 5251) NQN                                              00008870
        GO TO 1500                                                              00008880
C       ------------------------------ .DQ ---------------------------00008890
 5260   CONTINUE                                                               00008900
 5261 FORMAT(                                                                   00008910
     +      / ' X, Y LOCATION    =', G13.6, ' ,  ', G13.6, 1X, 2A4)             00008920
 5262 FORMAT(                                                                   00008930
     +      / ' X LOCATION       =', G13.6, ' TO ', G13.6, 1X, 2A4              00008940
     +      / ' Y LOCATION       =', G13.6, ' TO ', G13.6, 1X, 2A4 )            00008950
 5265 FORMAT(                                                                   00008960
     + '   START         VOLUME/       VOLUME      '                           00008970
     +, '  SOURCE        MASS/AREA     MASS FLOW   '                           00008980
     +/'   TIME          AREA RATE     FLOW RATE   '                           00008990
     +, '  CONCENTR.     RATE          RATE        '                           00009000
     +/1X, 6('  (', 2A4, ') ') )                                               00009010
 5266 FORMAT(1X, 6G13.6)                                                        00009020
        DO 5289 IQ = 1, NQ                                                      00009030
          IF(KKQX(IQ)-1) 5280, 5271, 5272                                       00009040
 5271       WRITE(LUW, 5261) QQXL(IQ), QQYL(IQ), HHQL(6), HHQL(7)              00009050
            GO TO 5275                                                          00009060
 5272       WRITE(LUW, 5262) QQXL(IQ), QQXM(IQ), HHQL(6), HHQL(7)              00009070
     +                     , QQYL(IQ), QQYM(IQ), HHQL(6), HHQL(7)             00009080
 5275       CALL PUTR(HHQA, QQA(IQ) )                                          00009090
            WRITE(LUW, 5265)                                                    00009100
     +        HHQT(6), HHQT(7), HHQV(6), HHQV(7), HHQQ(6), HHQQ(7)              00009110
     +      , HHQC(6), HHQC(7), HHQR(6), HHQR(7), HHQM(6), HHQM(7)              00009120
 5280     CONTINUE                                                           00009130
          WRITE(LUW, 5266) QQT (IQ), QQV (IQ), QQ  (IQ)                      00009140
     +                   , QQC (IQ), QQR (IQ), QQM (IQ)                       00009150
 5289     CONTINUE                                                           00009160
        WRITE(LUW, 1001)                                                        00009170
        GO TO 1500                                                              00009180
C       ------------------------------ .FF---------------------------00009190
 5300   CONTINUE                                                               00009200
```

```
 5301 FORMAT('1')                                                        00009210
      WRITE(LUW, 5301)                                                   00009220
      GO TO 1500                                                         00009230
C     ================================ .G ==============================00009240
 5400 CONTINUE                                                           00009250
      IF(KLX.NE.0 .AND. KLY.NE.0) GO TO 5410                             00009260
      IF(BATCH) GO TO 8120                                               00009270
      CALL PUSH(H9G)                                                     00009280
      IF(KLY.EQ.0) CALL PUSH(HLY)                                        00009290
      IF(KLX.EQ.0) CALL PUSH(HLX)                                        00009300
      GO TO 1500                                                         00009310
 5410 CONTINUE                                                           00009320
      CALL SETUP                                                         00009330
C                                                                        00009340
      NXL = MINO( (NSL - 5)/NSX, MJ)                                     00009350
      NX = NSN                                                           00009360
      IF(NX.LE.0) NX = NXL                                               00009370
      XI = XGI                                                           00009380
      IF(XGM.LE.XGL) GO TO 5425                                          00009390
        IF(XI.GT.0.) NX = (XGM - XGL)/XI + 1.0                           00009400
        IF(XI.LE.0.) XI = (XGM - XGL)/(NX - 1)                           00009410
 5425 CONTINUE                                                           00009420
      K  = ALOG10(AMAX1(ABS(XGL), ABS(XGL + XI*FLOAT(NX) ) ) )           00009430
      I  = 0                                                             00009440
      IF(K.GT.NSX-2) I = K - (NSX - 2)                                   00009450
      IF(K.LT.1    ) I = K - 3                                           00009460
      XS = 10.0**I                                                       00009470
C                                                                        00009480
      YI = YGI                                                           00009490
      IF(YI.LE.0.0) YI = XI                                              00009500
      NY = MAX1( (YGM - YGL)/YI, 0.0) + 1                                00009510
      K  = ALOG10(AMAX1(ABS(YGL), ABS(YGL + YI*FLOAT(NY) ) ) )           00009520
      I  = 0                                                             00009530
      IF(K.GT.3) I = K - 3                                               00009540
      IF(K.LT.1) I = K - 3                                               00009550
      YS = 10.0**I                                                       00009560
C                                                                        00009570
      CM = 10.0**(NSX - 1)                                               00009580
      XL = XGL                                                           00009590
 5440 NX1 = MINO(NX, NXL)                                                00009600
      XC = XL                                                            00009610
      DO 5449 IX = 1, NX1                                                00009620
        JJ(IX) = XC/XS                                                   00009630
        XC = XC + XI                                                     00009640
 5449   CONTINUE                                                         00009650
 5451 FORMAT(' X SCALE          (', G13.6, 1X, 2A4, ')'                  00009660
     +      / ' Y SCALE          (', G13.6, 1X, 2A4, ')'                00009670
     +      / ' CONCENTRATION    (', G13.6, 1X, 2A4, ')'                00009680
     +      / )                                                         00009690
 5452 FORMAT(' X SCALE          (', G13.6, 1X, 2A4, ')'                  00009700
     +      / ' Y SCALE          (', G13.6, 1X, 2A4, ')'                00009710
     +      / ' T SCALE          (', G13.6, 1X, 2A4, ')'                00009720
     +      / )                                                         00009730
 5453 FORMAT('         X'   /'      Y', 3013)                          00009740
 5454 FORMAT('          X'  /'      Y', 3014)                          00009750
 5455 FORMAT('           X' /'      Y', 3015)                          00009760
 5456 FORMAT('            X'/'      Y', 3016)                          00009770
        IF(BTSS) GO TO 5458                                              00009780
      B = TC.GE.0.0                                                      00009790
      IF(B) CALL PUTR(HHTC, TC)                                          00009800
      IF(.NOT.B) WRITE(LUW, 5001)                                        00009810
      WRITE(LUW, 5451)                                                   00009820
     +   XS, HHXC(6), HHXC(7)                                           00009830
     +  , YS, HHYC(6), HHYC(7)                                          00009840
     +  , SC, HH9C(6), HH9C(7)                                          00009850
        GO TO 5459                                                       00009860
```

```
5458    CALL PUTR(HHTE,  TE)                                                  00009870
        WRITE(LUW, 5452)                                                      00009880
     +    XS, HHXC(6), HHXC(7)                                               00009890
     +  , YS, HHYC(6), HHYC(7)                                               00009900
     +  , SC, HHTC(6), HHTC(7)                                               00009910
5459     CONTINUE                                                           00009920
        IF(NSX.EQ.3) WRITE(LUW, 5453) (JJ(IX), IX = 1, NX1)                   00009930
        IF(NSX.EQ.4) WRITE(LUW, 5454) (JJ(IX), IX = 1, NX1)                   00009940
        IF(NSX.EQ.5) WRITE(LUW, 5455) (JJ(IX), IX = 1, NX1)                   00009950
        IF(NSX.EQ.6) WRITE(LUW, 5456) (JJ(IX), IX = 1, NX1)                   00009960
        IF(NSY.GT.1) WRITE(LUW, 1001)                                         00009970
        CEM = 0.0                                                             00009980
        NCE = 0                                                               00009990
        NCM = 0                                                               00010000
        YC = YGM                                                              00010010
        DO 5479 IY = 1, NY                                                    00010020
          XC = XL                                                             00010030
          DO 5469 IX = 1, NX1                                                 00010040
            CALL CALC                                                         00010050
            IF(C) 5465, 5462, 5461                                            00010060
5461          CEM = AMAX1(CEM, 100.0*CE/C)                                    00010070
5462          C = C/SC                                                        00010080
              IF(C.GE.CM) GO TO 5463                                          00010090
              J = C + 0.5                                                     00010100
              GO TO 5467                                                      00010110
5463          CONTINUE                                                        00010120
              I = ALOG10(C)                                                   00010130
              I = I - (NSX - 3)                                               00010140
              J = -I                                                          00010150
              IF(I.LE.9) J = J - 10.0*INT(C/(10.0**I) )                       00010160
              NCM = NCM + 1                                                   00010170
              GO TO 5467                                                      00010180
5465          CONTINUE                                                        00010190
              J = -1                                                          00010200
              NCE = NCE + 1                                                   00010210
5467          CONTINUE                                                        00010220
              JJ(IX) = J                                                      00010230
              XC = XC + XI                                                    00010240
5469          CONTINUE                                                        00010250
5473 FORMAT(1X, I5, 3OI3)                                                     00010260
5474 FORMAT(1X, I5, 3OI4)                                                     00010270
5475 FORMAT(1X, I5, 3OI5)                                                     00010280
5476 FORMAT(1X, I5, 3OI6)                                                     00010290
          JY = YC/YS                                                          00010300
          IF(NSX.EQ.3) WRITE(LUW, 5473) JY, (JJ(IX), IX = 1, NX1)             00010310
          IF(NSX.EQ.4) WRITE(LUW, 5474) JY, (JJ(IX), IX = 1, NX1)             00010320
          IF(NSX.EQ.5) WRITE(LUW, 5475) JY, (JJ(IX), IX = 1, NX1)             00010330
          IF(NSX.EQ.6) WRITE(LUW, 5476) JY, (JJ(IX), IX = 1, NX1)             00010340
          IF(NSY.EQ.2) WRITE(LUW, 1001)                                       00010350
          IF(NSY.EQ.3) WRITE(LUW, 1002)                                       00010360
          IF(NSY.EQ.4) WRITE(LUW, 1003)                                       00010370
          YC = YC - YI                                                        00010380
5479      CONTINUE                                                            00010390
5481 FORMAT(' WORST APPROXIMATION = +-', G9.3, '%.')                          00010400
5482 FORMAT(1X, I5, ' LARGE VALUE(S) IN -(MAGNITUDE+EXPONENT) FORM.')         00010410
5483 FORMAT(1X, I5, ' SOURCE(S) SHOWN AS "-1".')                              00010420
        IF(CEM.GT.0.0) WRITE(LUW, 5481) CEM                                   00010430
        IF(NCM.GT.0  ) WRITE(LUW, 5482) NCM                                   00010440
        IF(NCE.GT.0  ) WRITE(LUW, 5483) NCE                                   00010450
        WRITE(LUW, 1001)                                                      00010460
        XL = XC                                                               00010470
        NX = NX - NX1                                                         00010480
        IF(NX.GT.0) GO TO 5440                                                00010490
        WRITE(LUW, 1001)                                                      00010500
         BTSS = .FALSE.                                                       00010510
        GO TO 1500                                                            00010520
```

```
C       ------------------------------- .I ----------------------------00010530
 5500   CALL PUSH(H9IQ)                                                00010540
        HCMD = H9IP                                                    00010550
        GO TO 1700                                                     00010560
C       ------------------------------- .IC ---------------------------00010570
 5510   CALL PUSH(HH9C(1))                                             00010580
        HCMD = H9I                                                     00010590
        GO TO 1700                                                     00010600
C       ------------------------------- .IG ---------------------------00010610
 5520   CALL PUSH(H9G)                                                 00010620
        HCMD = H9I                                                     00010630
        GO TO 1700                                                     00010640
C       ------------------------------- .IQ ---------------------------00010650
 5560   CONTINUE                                                       00010660
 5561 FORMAT( /                                                       00010670
     + '   X             Y                          '                 00010680
     +, '  START         MASS          SOURCE       '                 00010690
     +/'   LOCATION      LOCATION      AREA         '                 00010700
     +, '  TIME          FLOW RATE     CONCENTR.    ')                00010710
 5562 FORMAT( /                                                       00010720
     + '   X             Y                          '                 00010730
     +, '  START         MASS/AREA     SOURCE       '                 00010740
     +/'   LOCATION      LOCATION      AREA         '                 00010750
     +, '  TIME          RATE          CONCENTR.    ')                00010760
 5563 FORMAT( /                                                       00010770
     + '   X             Y                          '                 00010780
     +, '  START         VOLUME        SOURCE       '                 00010790
     +/'   LOCATION      LOCATION      AREA         '                 00010800
     +, '  TIME          FLOW RATE     CONCENTR.    ')                00010810
 5564 FORMAT( /                                                       00010820
     + '   X             Y                          '                 00010830
     +, '  START         VOLUME/       SOURCE       '                 00010840
     +/'   LOCATION      LOCATION      AREA         '                 00010850
     +, '  TIME          AREA RATE     CONCENTR.    ')                00010860
 5565 FORMAT(1X, 6('  (', 2A4, ') ') )                                00010870
 5566 FORMAT(1X, 6G13.6)                                              00010880
        KQMO = 0                                                      00010890
        DO 5589 IQ = 1, NQ                                            00010900
          KQX = KKQX(IQ)                                              00010910
          KQM = KKQM(IQ)                                              00010920
          GO TO (5571, 5572, 5573, 5574), KQM                         00010930
 5571       IF(KQM.NE.KQMO) WRITE(LUW, 5561)                         00010940
            H6 = HHQM(6)                                              00010950
            H7 = HHQM(7)                                              00010960
            Q  = QQM(IQ)                                              00010970
            GO TO 5580                                                00010980
 5572       IF(KQM.NE.KQMO) WRITE(LUW, 5562)                         00010990
            H6 = HHQR(6)                                              00011000
            H7 = HHQR(7)                                              00011010
            Q  = QQR(IQ)                                              00011020
            GO TO 5580                                                00011030
 5573       IF(KQM.NE.KQMO) WRITE(LUW, 5563)                         00011040
            H6 = HHQQ(6)                                              00011050
            H7 = HHQQ(7)                                              00011060
            Q  = QQ (IQ)                                              00011070
            GO TO 5580                                                00011080
 5574       IF(KQM.NE.KQMO) WRITE(LUW, 5564)                         00011090
            H6 = HHQV(6)                                              00011100
            H7 = HHQV(7)                                              00011110
            Q  = QQV(IQ)                                              00011120
C           GO TO 5580                                                00011130
 5580     IF(KQM.NE.KQMO) WRITE(LUW, 5565)                           00011140
     +        HHQL(6), HHQL(7), HHQL(6), HHQL(7), HHQA(6), HHQA(7)    00011150
     +      , HHQT(6), HHQT(7), H6     , H7     , HHQC(6), HHQC(7)    00011160
          WRITE(LUW, 5566) QQXL(IQ), QQYL(IQ), QQA (IQ)               00011170
     +                   , QQT (IQ), Q      , QQC (IQ)               00011180
```

```
          IF(KQX.EQ.2) WRITE(LUW, 5566) QQXM(IQ), QQYM(IQ)                    00011190
          KQMO = KQM                                                          00011200
 5589     CONTINUE                                                            00011210
         WRITE(LUW, 1001)                                                     00011220
         GO TO 1500                                                           00011230
C        ================================ .T ============================00011240
 5600     IF(KTE .GT. 0 .AND. KXC .NE. 0 .AND. KYC .NE. 0) GO TO 5610        00011250
          IF(BATCH) GO TO 8120                                                00011260
          CALL PUSH(HH9T(1))                                                  00011270
          IF(KYC .EQ. 0) CALL PUSH(HHYC(1))                                   00011280
          IF(KXC .EQ. 0) CALL PUSH(HHXC(1))                                   00011290
          IF(KTE .EQ. 0) CALL PUSH(HHTE(1))                                   00011300
          GO TO 1500                                                          00011310
 5610     CONTINUE                                                            00011320
          BTSS = .TRUE.                                                       00011330
          CALL SETUP                                                          00011340
          XC = X9C                                                            00011350
          YC = Y9C                                                            00011360
          CALL PUTR(HHTE, TE)                                                 00011370
          CALL PUTR(HHXC, XC)                                                 00011380
          CALL PUTR(HHYC, YC)                                                 00011390
          CALL CALC                                                           00011400
          CALL PUTR(HH9T, C)                                                  00011410
          WRITE(LUW, 1001)                                                    00011420
          BTSS = .FALSE.                                                      00011430
          GO TO 1500                                                          00011440
C        ================================ .TG ===========================00011450
 5620    IF(KTE .GT. 0) GO TO 5630                                            00011460
          IF(BATCH) GO TO 8120                                                00011470
          CALL PUSH(H9TG)                                                     00011480
          HCMD = HHTE(1)                                                      00011490
          GO TO 1700                                                          00011500
 5630     CONTINUE                                                            00011510
          BTSS = .TRUE.                                                       00011520
          HCMD = H9G                                                          00011530
          GO TO 1700                                                          00011540
C        ================================ ERRORS ========================00011550
C        -------------------------------- PARAMETER ERROR (SERIOUS) -----00011560
 8110    CONTINUE                                                            00011570
         WRITE(LUP, 1081)                                                     00011580
         IF(BATCH) GO TO 1400                                                00011590
         IF(BIP) GO TO 1700                                                  00011600
         GO TO 1500                                                          00011610
C        -------------------------------- PARAMETER ERROR (NON-SERIOUS) -00011620
 8120    CONTINUE                                                            00011630
         WRITE(LUP, 1081)                                                     00011640
         IF(BATCH) GO TO 1500                                                00011650
         IF(BIP) GO TO 1700                                                  00011660
         GO TO 1500                                                          00011670
C        ================================ END ===========================00011680
         END                                                                 00011690
C        ================================ INIT ==========================00011700
         SUBROUTINE INIT                                                     00011710
         LOGICAL*2     BERR, BATCH,BOP , BOE ,BTSS                           00011720
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                     00011730
         CHARACTER*4                                                         00011740
     +     HHCMD                                                             00011750
     +   , H    , HA   , HAR        , HHAX       , HHAY                      00011760
     +   , HC   , HD   , HHDX       , HHDY       , HHDR       , HHDM         00011770
     +   , HE   , HF   , HG         , HHGG       , HHGL       , HHGT         00011780
     +   , HH   , HI   , HIP , HIL , HIR                                     00011790
         CHARACTER*4                                                         00011800
     +     HL   , HLX , HLY , HM                                             00011810
     +   , HO   , HOD , HOP , HOE , HOT , HOW , HHP                          00011820
     +   , HQ   , HQN , HHQE       , HHQL       , HHQA       , HHQT         00011830
     +   , HHQM       , HHQR       , HHQC       , HHQQ       , HHQV         00011840
```

```
     +   , HR          , HHRR        , HHRK        , HHRD                          00011850
         CHARACTER*4                                                               00011860
     +     HS    , HSC , HSL , HSN , HSX , HSY , HHTC        , HHTE              00011870
     +   , HU          , HHU         , HHUL        , HHUM        , HHUT            00011880
     +   , HV          , HHVV        , HHVG        , HHVK        , HHVT            00011890
     +   , HHVR        , HHVI        , HHVD        , HHVU                          00011900
     +   , HXY         , HHXC        , HHYC        , HHZM        , HH9C            00011910
     +   , HH9T                                                                    00011920
         CHARACTER*4                                                               00011930
     +     H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                                 00011940
     +   , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                                      00011950
     +   , HHC1        , HHC2        , HHC3                                        00011960
         COMMON /CHAR/                                                             00011970
     +     HHCMD(30)                                                               00011980
     +   , H     , HA    , HAR         , HHAX  (7) , HHAY  (7)                     00011990
     +   , HC    , HD    , HHDX  (7) , HHDY  (7) , HHDR  (7) , HHDM  (7)           00012000
     +   , HE    , HF    , HG          , HHGG  (7) , HHGL  (7) , HHGT  (7)         00012010
     +   , HH    , HI    , HIP , HIL , HIR                                         00012020
     +   , HL    , HLX , HLY , HM                                                  00012030
     +   , HO    , HOD , HOP , HOE , HOT , HOW , HHP   (7)                         00012040
     +   , HQ    , HQN , HHQE  (7) , HHQL  (7) , HHQA  (7) , HHQT  (7)           00012050
     +   , HHQM  (7) , HHQR  (7) , HHQC  (7) , HHQQ  (7) , HHQV  (7)             00012060
     +   , HR          , HHRR  (7) , HHRK  (7) , HHRD  (7)                       00012070
     +   , HS    , HSC , HSL , HSN , HSX , HSY , HHTC  (7) , HHTE  (7)           00012080
     +   , HU          , HHU   (7) , HHUL  (7) , HHUM  (7) , HHUT  (7)           00012090
     +   , HV          , HHVV  (7) , HHVG  (7) , HHVK  (7) , HHVT  (7)           00012100
     +   , HHVR  (7) , HHVI  (7) , HHVD  (7) , HHVU  (7)                         00012110
     +   , HXY         , HHXC  (7) , HHYC  (7) , HHZM  (7) , HH9C  (7)           00012120
     +   , HH9T  (7)                                                             00012130
     +   , H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                               00012140
     +   , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                                    00012150
     +   , HHC1 (18) , HHC2 (18) , HHC3 (18)                                     00012160
         COMMON /BIT /                                                           00012170
     +     BERR, BATCH                                                           00012180
     +   , BOP , BOE ,BTSS                                                       00012190
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                         00012200
         COMMON /INTE/                                                           00012210
     +     LUL , LUM , LUR , LUP , LUE , LUW                                     00012220
     +   , MCMD                                                                  00012230
     +   , KDX , KDY , KG  , KLX , KLY , KR                                      00012240
     +   , KTE , KVG , KVP , KVR , KXC , KYC                                     00012250
     +   , NQ  , NQN , NSL , NSN , NSX , NSY                                     00012260
     +   , MQ                                                                    00012270
     +   , KKQX( 10) , KKQM( 10)                                                 00012280
         COMMON /REAL/                                                           00012290
     +     UA  , UD  , UGL , UGT , ULC                                           00012300
     +   , UQL , UQA , UQT , UQM , UQR , UQC , UQQ , UQV                         00012310
     +   , URK , URD , UTC , UVV , UVK , UVT , UVI , UVD , UVU , UZM             00012320
     +   , U9C                                                                   00012330
     +   , AX  , AY  , C   , CE  , DX  , DY  , DR  , DM  , DXT , DYT             00012340
     +   , G   , GL  , GT  , G2  , P   , QE  , QD                                00012350
     +   , R   , RK  , RP  , RD  , SC  , TC  , TD  , TE  ,TPHI                   00012360
     +   , V   , VN  , VG  , VP  , VK  , VKN , VT  , VTN , VM                    00012370
     +   , VR  , VI  , VD  , VU                                                  00012380
     +   , XC  , X9C , XGL , XGM , XGI , XD                                      00012390
     +   , YC  , Y9C , YGL , YGM , YGI , YD  , ZM                                00012400
     +   , QQXL( 10) , QQXM( 10) , QQYL( 10) , QQYM( 10)                         00012410
     +   , QQA ( 10) , QQT ( 10) , QQV ( 10) , QQ  ( 10)                         00012420
     +   , QQC ( 10) , QQR ( 10) , QQM ( 10)                                     00012430
         OPEN(1,FILE='BLOCK.DAT')                                                00012440
C        ================================ CHARACTER ====================00012450
 7000 FORMAT(20A4)                                                              00012460
         READ(1,7000)(HHCMD(I),I=1,30)                                          00012470
         READ(1,7000) H,HA,HAR                                                  00012480
         READ(1,7000)(HHAX(I),I=1,7)                                            00012490
         READ(1,7000)(HHAY(I),I=1,7)                                            00012500
```

```
      READ(1,7000) HC,HD                                                00012510
      READ(1,7000)(HHDX(I),I=1,7)                                       00012520
      READ(1,7000)(HHDY(I),I=1,7)                                       00012530
      READ(1,7000)(HHDR(I),I=1,7)                                       00012540
      READ(1,7000)(HHDM(I),I=1,7)                                       00012550
      READ(1,7000) HE,HF,HG                                             00012560
      READ(1,7000)(HHGG(I),I=1,7)                                       00012570
      READ(1,7000)(HHGL(I),I=1,7)                                       00012580
      READ(1,7000)(HHGT(I),I=1,7)                                       00012590
      READ(1,7000) HH,HI                                                00012600
      READ(1,7000) HIP,HIL,HIR                                          00012610
      READ(1,7000) HL,HLX,HLY,HM                                        00012620
      READ(1,7000) HO,HOD,HOW                                           00012630
      READ(1,7000) HOP,HOE,HOT                                          00012640
      READ(1,7000)(HHP(I),I=1,7)                                        00012650
      READ(1,7000) HQ,HQN                                               00012660
      READ(1,7000)(HHQL(I),I=1,7)                                       00012670
      READ(1,7000)(HHQA(I),I=1,7)                                       00012680
      READ(1,7000)(HHQT(I),I=1,7)                                       00012690
      READ(1,7000)(HHQM(I),I=1,7)                                       00012700
      READ(1,7000)(HHQR(I),I=1,7)                                       00012710
      READ(1,7000)(HHQC(I),I=1,7)                                       00012720
      READ(1,7000)(HHQQ(I),I=1,7)                                       00012730
      READ(1,7000)(HHQV(I),I=1,7)                                       00012740
      READ(1,7000)(HHQE(I),I=1,7)                                       00012750
      READ(1,7000) HR                                                   00012760
      READ(1,7000)(HHRR(I),I=1,7)                                       00012770
      READ(1,7000) HS,HSC,HSL                                           00012780
      READ(1,7000) HSX,HSN,HSY                                          00012790
      READ(1,7000)(HHTC(I),I=1,7)                                       00012800
      READ(1,7000)(HHTE(I),I=1,7)                                       00012810
      READ(1,7000) HU                                                   00012820
      READ(1,7000) HV                                                   00012830
      READ(1,7000)(HHVV(I),I=1,7)                                       00012840
      READ(1,7000) HXY                                                  00012650
      READ(1,7000)(HHXC(I),I=1,7)                                       00012860
      READ(1,7000)(HHYC(I),I=1,7)                                       00012870
      READ(1,7000)(HHZM(I),I=1,7)                                       00012880
      READ(1,7000)(HH9C(I),I=1,7)                                       00012890
      READ(1,7000)(HH9T(I),I=1,7)                                       00012900
      READ(1,7000) H9D,H9DC,H9DG                                        00012910
      READ(1,7000) H9DP,H9DQ                                            00012920
      READ(1,7000) H9FF,H9G                                             00012930
      READ(1,7000) H9I,H9IC,H9IG                                        00012940
      READ(1,7000) H9IP,H9IQ,H9TG                                       00012950
C     ================================ BIT ============================00012960
C     -------------------------------- I/O OPTIONS --------------------00012970
 7010 FORMAT(4L2)                                                       00012980
      READ(1,7010) BATCH                                                00012990
      READ(1,7010) BDP,BOE,BTSS                                         00013000
      READ(1,7010) BT1,BT2,BT3,BT4                                      00013010
      READ(1,7010) BT5,BT6,BT7,BT8                                      00013020
C     ================================ INTEGER ========================00013030
C     -------------------------------- I/O UNITS, DIMENSIONS ---------00013040
 7020 FORMAT(5I5)                                                       00013050
      READ(1,7020) LUL,LUM                                              00013060
      READ(1,7020) LUR,LUP,LUE,LUW                                      00013070
      READ(1,7020) MCMD,MQ                                              00013080
C     -------------------------------- FLAGS AND COUNTS --------------00013090
      READ(1,7020) KDX,KDY,KG,KLX,KLY                                   00013100
      READ(1,7020) KR,KTE,KVG,KVP,KVR                                   00013110
      READ(1,7020) KXC,KYC,NQ,NQN                                       00013120
      READ(1,7020) NSL,NSN,NSX,NSY                                      00013130
C     ================================ REAL ===========================00013140
C     -------------------------------- UNITS, PARAMETERS -------------00013150
 7030 FORMAT(3F13.10)                                                   00013160
```

```
      READ(1,7030) UA,UD                                                    00013170
      READ(1,7030) UGL,UGT,ULC                                              00013180
      READ(1,7030) UQL,UQA,UQT                                              00013190
      READ(1,7030) UQM,UQR,UQC                                              00013200
      READ(1,7030) UQQ,UQV                                                  00013210
      READ(1,7030) URK,URD,UTC                                              00013220
      READ(1,7030) UVV,UVK,UVT                                              00013230
      READ(1,7030) UVI,UVD,UVU                                              00013240
      READ(1,7030) UZM,U9C                                                  00013250
      READ(1,7030) DM,QE,SC                                                 00013260
      END                                                                   00013270
C     ================================ PUSH ============================00013280
      SUBROUTINE PUSH(HO)                                                   00013290
      LOGICAL*2     BERR, BATCH,BOP , BOE ,BTSS                             00013300
     + , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                      00013310
      CHARACTER*4                                                           00013320
     +   HHCMD                                                              00013330
     + , H   , HA  , HAR       , HHAX      , HHAY                          00013340
     + , HC  , HD  , HHDX      , HHDY      , HHDR      , HHDM              00013350
     + , HE  , HF  , HG        , HHGG      , HHGL      , HHGT              00013360
     + , HH  , HI  , HIP , HIL , HIR                                        00013370
      CHARACTER*4                                                           00013380
     +   HL  , HLX , HLY , HM                                               00013390
     + , HO  , HOD , HOP , HOE , HOT , HOW , HHP                            00013400
     + , HQ  , HQN , HHQE      , HHQL      , HHQA      , HHQT              00013410
     + , HHQM      , HHQR      , HHQC      , HHQQ      , HHQV              00013420
     + , HR        , HHRR      , HHRK      , HHRD                          00013430
      CHARACTER*4                                                           00013440
     +   HS  , HSC , HSL , HSN , HSX , HSY , HHTC      , HHTE              00013450
     + , HU        , HHU       , HHUL      , HHUM      , HHUT              00013460
     + , HV        , HHVV      , HHVG      , HHVK      , HHVT              00013470
     + , HHVR      , HHVI      , HHVD      , HHVU                          00013480
     + , HXY       , HHXC      , HHYC      , HHZM      , HH9C              00013490
     + , HH9T                                                               00013500
      CHARACTER*4                                                           00013510
     +   H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                            00013520
     + , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                                 00013530
     + , HHC1      , HHC2      , HHC3                                       00013540
      CHARACTER*4   HO  , H1  , H2                                          00013550
      COMMON /CHAR/                                                         00013560
     +   HHCMD(30)                                                          00013570
     + , H   , HA  , HAR       , HHAX  (7) , HHAY  (7)                     00013580
     + , HC  , HD  , HHDX  (7) , HHDY  (7) , HHDR  (7) , HHDM  (7)         00013590
     + , HE  , HF  , HG        , HHGG  (7) , HHGL  (7) , HHGT  (7)         00013600
     + , HH  , HI  , HIP , HIL , HIR                                        00013610
     + , HL  , HLX , HLY , HM                                               00013620
     + , HO  , HOD , HOP , HOE , HOT , HOW , HHP   (7)                      00013630
     + , HQ  , HQN , HHQE  (7) , HHQL  (7) , HHQA  (7) , HHQT  (7)         00013640
     + , HHQM  (7) , HHQR  (7) , HHQC  (7) , HHQQ  (7) , HHQV  (7)         00013650
     + , HR        , HHRR  (7) , HHRK  (7) , HHRD  (7)                     00013660
     + , HS  , HSC , HSL , HSN , HSX , HSY , HHTC  (7) , HHTE  (7)         00013670
     + , HU        , HHU   (7) , HHUL  (7) , HHUM  (7) , HHUT  (7)         00013680
     + , HV        , HHVV  (7) , HHVG  (7) , HHVK  (7) , HHVT  (7)         00013690
     + , HHVR  (7) , HHVI  (7) , HHVD  (7) , HHVU  (7)                      00013700
     + , HXY       , HHXC  (7) , HHYC  (7) , HHZM  (7) , HH9C  (7)         00013710
     + , HH9T  (7)                                                          00013720
     + , H9D , H9DC, H9DG, H9DP, H9DQ, H9FF, H9G                            00013730
     + , H9I , H9IC, H9IG, H9IP, H9IQ, H9TG                                 00013740
     + , HHC1 (18) , HHC2 (18) , HHC3 (18)                                  00013750
      COMMON /BIT /                                                         00013760
     +   BERR, BATCH                                                        00013770
     + , BOP , BOE ,BTSS                                                    00013780
     + , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                      00013790
      COMMON /INTE/                                                         00013800
     +   LUL , LUM , LUR , LUP , LUE , LUW                                  00013810
     + , MCMD                                                               00013820
```

```
     +   , KDX , KDY , KG  , KLX , KLY , KR                                  00013830
     +   , KTE , KVG , KVP , KVR , KXC , KYC                                  00013840
     +   , NQ  , NQN , NSL , NSN , NSX , NSY                                  00013850
     +   , MQ                                                                 00013860
     +   , KKQX( 10) , KKQM( 10)                                              00013870
         H1 = HO                                                              00013880
         DO 5900 I = 1, MCMD                                                  00013890
           H2 = HHCMD(I)                                                      00013900
           HHCMD(I) = H1                                                      00013910
           IF(H1.EQ.H) RETURN                                                 00013920
           H1 = H2                                                            00013930
 5900      CONTINUE                                                           00013940
         WRITE(LUP, 9001) MCMD, H1                                            00013950
 9001 FORMAT(' STACK SPACE OF', I5, ' COMMANDS EXCEEDED.'                     00013960
     +     / ' COMMAND ', A4, ' LOST.')                                       00013970
         END                                                                  00013980
C        ================================ GETI ============================00013990
         SUBROUTINE GETI(KO, LO, MO)                                          00014000
         LOGICAL*2     BERR, BATCH,BOP , BOE , BTSS                           00014010
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                      00014020
         COMMON /BIT /                                                        00014030
     +     BERR, BATCH                                                        00014040
     +   , BOP , BOE , BTSS                                                   00014050
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                      00014060
         COMMON /INTE/                                                        00014070
     +     LUL , LUM , LUR , LUP , LUE , LUW                                  00014080
     +   , MCMD                                                               00014090
     +   , KDX , KDY , KG  , KLX , KLY , KR                                  00014100
     +   , KTE , KVG , KVP , KVR , KXC , KYC                                  00014110
     +   , NQ  , NQN , NSL , NSN , NSX , NSY                                  00014120
     +   , MQ                                                                 00014130
     +   , KKQX( 10) , KKQM( 10)                                              00014140
 1022 FORMAT(10I5)                                                            00014150
 1032 FORMAT(1X, 10I5)                                                        00014160
         READ(LUR, 1022) K1                                                   00014170
         IF(BOE) WRITE(LUE, 1032) K1                                          00014180
         BERR = .TRUE.                                                        00014190
         IF(LO.LT.MO .AND. (K1.LT.LO .OR. K1.GT.MO) ) RETURN                  00014200
         BERR = .FALSE.                                                       00014210
         KO = K1                                                              00014220
         RETURN                                                               00014230
         END                                                                  00014240
C        ================================ GETR ============================00014250
         SUBROUTINE GETR(HHO, RO, BO)                                         00014260
         LOGICAL*2     BERR, BATCH,BOP , BOE , BTSS                           00014270
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                      00014280
         LOGICAL*2     BO                                                     00014290
         CHARACTER*4   HHO , H                                                00014300
         DIMENSION HHO(7)                                                     00014310
         COMMON /BIT /                                                        00014320
     +     BERR, BATCH                                                        00014330
     +   , BOP , BOE , BTSS                                                   00014340
     +   , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                      00014350
         COMMON /INTE/                                                        00014360
     +     LUL , LUM , LUR , LUP , LUE , LUW                                  00014370
     +   , MCMD                                                               00014380
     +   , KDX , KDY , KG  , KLX , KLY , KR                                  00014390
     +   , KTE , KVG , KVP , KVR , KXC , KYC                                  00014400
     +   , NQ  , NQN , NSL , NSN , NSX , NSY                                  00014410
     +   , MQ                                                                 00014420
     +   , KKQX( 10) , KKQM( 10)                                              00014430
         DATA H /'    '/                                                      00014440
 1011 FORMAT(1X, 4A4, ' (',2A4, '):' / ' ?')                                  00014450
 1012 FORMAT(1X, 4A4, ' (UNITLESS):' / ' ?')                                  00014460
 1023 FORMAT(E15.0)                                                           00014470
 1033 FORMAT(1X, G15.7)                                                       00014480
```

```
            QM = QQM(IQ)*UQM                                                    00016470
            QN = (QM - QMO)/QDA                                                 00016480
            YQ = YL + YI*0.5                                                    00016490
            DO 5290 IY = 1, NY                                                  00016500
              YN = (Y - YQ)/YD                                                  00016510
              XQ = XL + XI*0.5                                                  00016520
              DO 5190 IX = 1, NX                                                00016530
                XN = (X - XQ)/XD                                                00016540
                RN = SQRT(XN*XN + YN*YN)                                        00016550
                 IF(BTSS) GO TO 5000                                            00016560
                IF(RN.LE.0.) GO TO 9100                                         00016570
                IF(T.LT.0.) GO TO 4500                                          00016580
                  TN = (T - TQ)/TD                                              00016590
                  WO = (RN - TN)/SQRT(4.0*TN)                                   00016600
                     E = ERFC(WO)                                               00016610
 4500           CONTINUE                                                        00016620
                CY = EXP( (XN/G2 - RN)*0.5)                                     00016630
                CR = 4.0*SQRT(WPI*RN)                                           00016640
                W  = QN*CY*E/CR                                                 00016650
 5001 FORMAT(1X, 5G15.7)                                                        00016660
                IF(BT5) WRITE(LUW, 5001)                                        00016670
     +            X , XQ, XD, XN, CR                                           00016680
     +          , Y , YQ, YD, YN, CY                                           00016690
     +          , T , TQ, TD, TN, E                                            00016700
     +          , QM, QMO,QDA,QN, W                                            00016710
                C  = C + W                                                      00016720
                CE = CE + W*ER/RN                                               00016730
                 GO TO 5090                                                     00016740
 5000            CONTINUE                                                       00016750
                 CR = AMAX1(CR, RN)                                             00016760
 5090           CONTINUE                                                        00016770
                XQ = XQ + XI                                                    00016780
 5190           CONTINUE                                                        00016790
              YQ = YQ + YI                                                      00016800
 5290         CONTINUE                                                        00016810
            QMO = QM                                                            00016820
 5390       CONTINUE                                                          00016830
           IF(BTSS) GO TO 5500                                                00016840
          C  = C/U9C                                                          00016850
          CE = CE/U9C                                                         00016860
           GO TO 5990                                                         00016870
 5500      CONTINUE                                                          00016880
           IF(CR .GT. 0.) C = CR / (SQRT(TPHI * TPHI + CR) + TPHI)           00016890
           C = (TM + C * C * TD) / UTC                                       00016900
 5990      CONTINUE                                                          00016910
          BERR = .FALSE.                                                     00016920
          RETURN                                                             00016930
 9100     CONTINUE                                                          00016940
          C  = -1.0                                                          00016950
          CE = 0.0                                                           00016960
          BERR = .TRUE.                                                      00016970
          RETURN                                                             00016980
          END                                                                00016990
C         ================================ ERFC ============================00017000
           FUNCTION ERFC(WO)                                                 00017010
          DATA E0, E1, E2, E3, E4, E5                                        00017020
     +     / .32759, .25438, -.28540, 1.42141, -1.45315, 1.06141 /           00017030
          W  = 1.0/(1.0 + E0*ABS(WO) )                                       00017040
          W  = W*(E1 + W*(E2 + W*(E3 + W*(E4 + W*E5) ) ) )                   00017050
          E  = W*EXP(-(WO*WO) )                                              00017060
          IF(WO.LT.0.) E = 2.0 - E                                           00017070
           ERFC = E                                                          00017080
           RETURN                                                            00017090
           END                                                               00017100
```

```
**** TSO FOREGROUND HARDCOPY ****
DSNAME=U11236C.BLOCKPC.DATA

I

    A   AR
AX  X DISPERSIVITY  FT
AY  Y DISPERSIVITY  FT
C   D
DX  X DISPERSION    FT2/D
DY  Y DISPERSION    FT2/D
DR  DISPERSION RATIO
DM  MOL. DIFFUSION  FT2/D
E   F   G
GG  DECAY GAMMA
GL  DECAY LAMBDA    1/YR
GT  DECAY HALF-LIFE YR
H   I
IP  IL  IR
L   LX  LY  M
O   OD  OW
OP  OE  OT
P   POROSITY
Q   QN
QL  SOURCE LOCATION FT
QA  SOURCE AREA     FT2
QT  SOURCE TIME     DAYS
QM  MASS FLOW RATE  LBM/D
QR  MASS/AREA RATE  LB/FT2/D
QC  SOURCE CONCENTR.MG/L
QQ  VOLUME FLOW RATEFT3/D
QV  VOLUME/AREA RATEFT/D
QE  ACCURACY
R
RR  RETARDATION
S   SC  SL
SX  SN  SY
T   SAMPLE TIME     DAYS
TE  % STEADY STATE
U
V
VV  VELOCITY        FT/D
XY
X   X LOCATION      FT
Y   Y LOCATION      FT
Z   THICKNESS       FT
.C  CONCENTRATION   MG/L
.T  STEADY STATE    DAYS
.D  .DC .DG
.DP .DQ
.FF .G
.I  .IC .IG
.IP .IQ .TG
 F
 T F F
 F F F F
 F F F F
    6   10
    0    0    0    0
   30   10
    0    0    0    0    0
    0    0    0    0    0
    0    0    0  100
   80    0    6    2
1.0          1.0
```

```
2.737909E-3  365.2422      1.0
1.0          1.0           1.0
1.0          1.0           62.42796E-6
1.0          1.0
1.0          1.0           1.0
1.0          1.0           1.0
1.0          1.0           1.0
1.0          62.42796E-6
0.0          0.1           1.0
```

```
      IF(.NOT.BOP) GO TO 3000                                          00014490
      IF(HHO(6).EQ.H) GO TO 2500                                       00014500
      WRITE(LUP, 1011) (HHO(I), I = 2, 7)                              00014510
      GO TO 3000                                                       00014520
 2500 WRITE(LUP, 1012) (HHO(I), I = 2, 5)                              00014530
 3000 READ(LUR, 1023) R1                                               00014540
      IF(BOE) WRITE(LUE, 1033) R1                                      00014550
      BERR = .TRUE.                                                    00014560
      IF(BO .AND. R1.LE.0) RETURN                                      00014570
      BERR = .FALSE.                                                   00014580
      RO = R1                                                          00014590
      RETURN                                                           00014600
      END                                                              00014610
C     ================================ PUTR ============================00014620
      SUBROUTINE PUTR(HHO, RO)                                         00014630
      LOGICAL*2     BERR, BATCH,BOP , BOE , BTSS                       00014640
     +, BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00014650
      CHARACTER*4   HHO                                                00014660
      DIMENSION HHO(7)                                                 00014670
      COMMON /BIT /                                                    00014680
     +  BERR, BATCH                                                    00014690
     +, BOP , BOE , BTSS                                               00014700
     +, BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00014710
      COMMON /INTE/                                                    00014720
     +  LUL , LUM , LUR , LUP , LUE , LUW                             00014730
     +, MCMD                                                          00014740
     +, KDX , KDY , KG  , KLX , KLY , KR                              00014750
     +, KTE , KVG , KVP , KVR , KXC , KYC                             00014760
     +, NQ  , NQN , NSL , NSN , NSX , NSY                             00014770
     +, MQ                                                            00014780
     +, KKQX( 10) , KKQM( 10)                                         00014790
 1001 FORMAT(1X, 4A4, '=', G13.6, 1X, 2A4)                            00014800
      WRITE(LUW, 1001) (HHO(I), I = 2, 5), RO, HHO(6), HHO(7)          00014810
      RETURN                                                           00014820
      END                                                              00014830
C     ================================ PUTH ============================00014840
      SUBROUTINE PUTH(HHO, LUO)                                        00014850
      LOGICAL*2     BERR, BATCH,BOP , BOE , BTSS                       00014860
     +, BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00014870
      CHARACTER*4   HHO , H                                            00014880
      DIMENSION HHO(18)                                                00014890
      COMMON /BIT /                                                    00014900
     +  BERR, BATCH                                                    00014910
     +, BOP , BOE , BTSS                                               00014920
     +, BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00014930
      COMMON /INTE/                                                    00014940
     +  LUL , LUM , LUR , LUP , LUE , LUW                             00014950
     +, MCMD                                                          00014960
     +, KDX , KDY , KG  , KLX , KLY , KR                              00014970
     +, KTE , KVG , KVP , KVR , KXC , KYC                             00014980
     +, NQ  , NQN , NSL , NSN , NSX , NSY                             00014990
     +, MQ                                                            00015000
     +, KKQX( 10) , KKQM( 10)                                         00015010
      DATA H /'    '/                                                 00015020
 1001 FORMAT(1X, 18A4)                                                00015030
      N = 18                                                           00015040
      DO 2900 I = 1, 17                                                00015050
        IF(HHO(N).NE.H) GO TO 3000                                     00015060
        N = N - 1                                                      00015070
 2900   CONTINUE                                                       00015080
 3000 CONTINUE                                                        00015090
      WRITE(LUO, 1001) (HHO(I), I = 1, N)                              00015100
      RETURN                                                           00015110
      END                                                              00015120
C     ================================ SETUP ===========================00015130
      SUBROUTINE SETUP                                                 00015140
```

```
      LOGICAL*2     BERR, BATCH,BOP , BOE , BTSS                        00015150
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00015160
      COMMON /BIT /                                                     00015170
     +    BERR, BATCH                                                  00015180
     +  , BOP , BOE , BTSS                                             00015190
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                00015200
      COMMON /INTE/                                                     00015210
     +    LUL , LUM , LUR , LUP , LUE , LUW                            00015220
     +  , MCMD                                                         00015230
     +  , KDX , KDY , KG  , KLX , KLY , KR                            00015240
     +  , KTE , KVG , KVP , KVR , KXC , KYC                           00015250
     +  , NQ  , NQN , NSL , NSN , NSX , NSY                           00015260
     +  , MQ                                                           00015270
     +  , KKQX( 10) , KKQM( 10)                                       00015280
      COMMON /REAL/                                                     00015290
     +    UA  , UD  , UGL , UGT , ULC                                  00015300
     +  , UQL , UQA , UQT , UQM , UQR , UQC , UQQ , UQV               00015310
     +  , URK , URD , UTC , UVV , UVK , UVT , UVI , UVD , UVU , UZM   00015320
     +  , U9C                                                          00015330
     +  , AX  , AY  , C   , CE  , DX  , DY  , DR  , DM  , DXT , DYT   00015340
     +  , G   , GL  , GT  , G2  , P   , QE  , QD                      00015350
     +  , R   , RK  , RP  , RD  , SC  , TC  , TD  , TE  , TPHI        00015360
     +  , V   , VN  , VG  , VP  , VK  , VKN , VT  , VTN , VM          00015370
     +  , VR  , VI  , VD  , VU                                        00015380
     +  , XC  , X9C , XGL , XGM , XGI , XD                            00015390
     +  , YC  , Y9C , YGL , YGM , YGI , YD  , ZM                      00015400
     +  , QQXL( 10) , QQXM( 10) , QQYL( 10) , QQYM( 10)               00015410
     +  , QQA ( 10) , QQT ( 10) , QQV ( 10) , QQ  ( 10)               00015420
     +  , QQC ( 10) , QQR ( 10) , QQM ( 10)                           00015430
C     ------------------------------ CALCULATE BASIC PARAMETERS ----00015440
      V1 = V *UVV                                                       00015450
      V2 = V1*V1                                                        00015460
      IF(KDX.EQ.2 ) DX = AX*UA*V1/    UD                              00015470
      IF(KDX.EQ.1 ) AX = DX*UD   /(V1*UA)                             00015480
      IF(KDY.EQ.3 ) DY = DX      / DR                                 00015490
      IF(KDY.EQ.2 ) DY = AY*UA*V1/    UD                              00015500
      IF(KDY.NE.2 ) AY = DY*UD   /(V1*UA)                             00015510
      IF(KDY.LT.3 ) DR = DX      / DY                                 00015520
      DXT = DX + DM                                                     00015530
      DYT = DY + DM                                                     00015540
      D1  = DXT*UD                                                      00015550
      D2  = SQRT(D1*DYT*UD)                                             00015560
      IF(KG .EQ.3 ) GL = ALOG(2.0)/(GT*UGT*UGL)                       00015570
      IF(KG .GE.2 ) G  = 1.0 + 4.0*GL*UGL*D1/V2                       00015580
      IF(KG .LT.2 ) GL = (G - 1.0)*V2/(4.0*D1*UGL)                    00015590
      IF(KG .LT.3 .AND. GL.GT.0.) GT = ALOG(2.0)/(GL*UGL*UGT)         00015600
      G2 = SQRT(G)                                                      00015610
C     ------------------------------ NORMALIZING VARIABLES ---------00015620
      XD = D1   /(G2*V1)                                                00015630
      YD = D2   /(G2*V1)                                                00015640
      TD = R *D1/(G *V2)                                                00015650
      QD = P *ZM*UZM*D2                                                 00015660
      RETURN                                                            00015670
      END                                                               00015680
C     ================================ CALC ========================00015690
      SUBROUTINE CALC                                                   00015700
      LOGICAL*2     BERR, BATCH,BOP , BOE , BTSS                        00015710
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                00015720
      COMMON /BIT /                                                     00015730
     +    BERR, BATCH                                                  00015740
     +  , BOP , BOE , BTSS                                             00015750
     +  , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                00015760
      COMMON /INTE/                                                     00015770
     +    LUL , LUM , LUR , LUP , LUE , LUW                            00015780
     +  , MCMD                                                         00015790
     +  , KDX , KDY , KG  , KLX , KLY , KR                            00015800
```

```
     +   , KTE , KVG , KVP , KVR , KXC , KYC                                    00015810
     +   , NQ  , NQN , NSL , NSN , NSX , NSY                                    00015820
     +   , MQ                                                                   00015830
     +   , KKQX( 10) , KKQM( 10)                                                00015840
         COMMON /REAL/                                                          00015850
     +     UA  , UD  , UGL , UGT , ULC                                          00015860
     +   , UQL , UQA , UQT , UQM , UQR , UQC , UQQ , UQV                        00015870
     +   , URK , URD , UTC , UVV , UVK , UVT , UVI , UVD , UVU , UZM            00015880
     +   , U9C                                                                  00015890
     +   , AX  , AY  , C   , CE  , DX  , DY  , DR  , DM  , DXT , DYT           00015900
     +   , G   , GL  , GT  , G2  , P   , QE  , QD                              00015910
     +   , R   , RK  , RP  , RD  , SC  , TC  , TD  , TE  , TPHI                00015920
     +   , V   , VN  , VG  , VP  , VK  , VKN , VT  , VTN , VM                  00015930
     +   , VR  , VI  , VD  , VU                                                 00015940
     +   , XC  , X9C , XGL , XGM , XGI , XD                                     00015950
     +   , YC  , Y9C , YGL , YGM , YGI , YD  , ZM                               00015960
     +   , QQXL( 10) , QQXM( 10) , QQYL( 10) , QQYM( 10)                        00015970
     +   , QQA ( 10) , QQT ( 10) , QQV ( 10) , QQ  ( 10)                        00015980
     +   , QQC ( 10) , QQR ( 10) , QQM ( 10)                                    00015990
         DATA WPI / 3.14159265 /                                                00016000
         X  = XC*ULC                                                            00016010
         Y  = YC*ULC                                                            00016020
         T  = TC*UTC                                                            00016030
         E  = 2.0                                                               00016040
         C  = 0.                                                                00016050
         CE = 0.                                                                00016060
          CR = 0.                                                               00016070
          TM = QQT(1)*UQT                                                       00016080
         DO 5390 IQ = 1, NQ                                                     00016090
           TQ = QQT(IQ)*UQT                                                     00016100
            TM = AMAX1(TM,TQ)                                                   00016110
           IF(T.GE.0. .AND. TQ.GE.T) GO TO 5390                                 00016120
           KQX = KKQX(IQ)                                                       00016130
           IF(KQX.LT.0) GO TO 3000                                              00016140
           XL = QQXL(IQ)*UQL                                                    00016150
           YL = QQYL(IQ)*UQL                                                    00016160
           GO TO (2100, 2200), KQX                                              00016170
2100       CONTINUE                                                            00016180
             NX = 1                                                             00016190
             XI = 0.                                                            00016200
             NY = 1                                                             00016210
             YI = 0.                                                            00016220
             S2 = QQA (IQ)*UQA                                                  00016230
             QDA = QD                                                           00016240
             GO TO 2900                                                         00016250
2200       CONTINUE                                                            00016260
             XM = QQXM(IQ)*UQL                                                  00016270
             YM = QQYM(IQ)*UQL                                                  00016280
             RN = AMAX1(X-XM, XL-X, Y-YM, YL-Y)                                 00016290
             IF(RN.LE.0.) GO TO 9100                                            00016300
             XI = XM - XL                                                       00016310
             YI = YM - YL                                                       00016320
             XN = NQN                                                           00016330
             S  = SQRT(AMAX1(50.0*XD*QE*RN/(1.0 + DR), XI*YI/XN) )              00016340
             NX = NQN - MAX1(XN - XI/S, 0.)                                     00016350
             XI = XI/FLOAT(NX)                                                  00016360
             NY = NQN - MAX1(XN - YI/S, 0.)                                     00016370
             YI = YI/FLOAT(NY)                                                  00016380
             S2 = AMAX1(XI, YI)                                                 00016390
             S2 = S2*S2                                                         00016400
             QDA = FLOAT(NX)*FLOAT(NY)*QD                                       00016410
2900       CONTINUE                                                            00016420
           ER = 0.02*S2*(1.0 + DR)/(XD*XD)                                      00016430
           IF(T.GE.0.) ER = ER + 0.2                                            00016440
           QMO = 0.                                                             00016450
3000       CONTINUE                                                            00016460
```

```
      LOGICAL*2     BERR, BATCH,BOP , BOE , BTSS                        00015150
     + , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                   00015160
      COMMON /BIT /                                                      00015170
     +   BERR, BATCH                                                     00015180
     + , BOP , BOE , BTSS                                               00015190
     + , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00015200
      COMMON /INTE/                                                      00015210
     +   LUL , LUM , LUR , LUP , LUE , LUW                              00015220
     + , MCMD                                                           00015230
     + , KDX , KDY , KG  , KLX , KLY , KR                               00015240
     + , KTE , KVG , KVP , KVR , KXC , KYC                              00015250
     + , NQ  , NQN , NSL , NSN , NSX , NSY                              00015260
     + , MQ                                                             00015270
     + , KKQX( 10) , KKQM( 10)                                          00015280
      COMMON /REAL/                                                      00015290
     +   UA  , UD  , UGL , UGT , ULC                                    00015300
     + , UQL , UQA , UQT , UQM , UQR , UQC , UQQ , UQV                  00015310
     + , URK , URD , UTC , UVV , UVK , UVT , UVI , UVD , UVU , UZM      00015320
     + , U9C                                                            00015330
     + , AX  , AY  , C   , CE  , DX  , DY  , DR  , DM  , DXT , DYT      00015340
     + , G   , GL  , GT  , G2  , P   , QE  , QD                         00015350
     + , R   , RK  , RP  , RD  , SC  , TC  , TD  , TE  , TPHI           00015360
     + , V   , VN  , VG  , VP  , VK  , VKN , VT  , VTN , VM             00015370
     + , VR  , VI  , VD  , VU                                           00015380
     + , XC  , X9C , XGL , XGM , XGI , XD                               00015390
     + , YC  , Y9C , YGL , YGM , YGI , YD  , ZM                         00015400
     + , QQXL( 10) , QQXM( 10) , QQYL( 10) , QQYM( 10)                  00015410
     + , QQA ( 10) , QQT ( 10) , QQV ( 10) , QQ  ( 10)                  00015420
     + , QQC ( 10) , QQR ( 10) , QQM ( 10)                              00015430
C     ------------------------------- CALCULATE BASIC PARAMETERS ----00015440
      V1 = V *UVV                                                        00015450
      V2 = V1*V1                                                         00015460
      IF(KDX.EQ.2 ) DX = AX*UA*V1/    UD                                 00015470
      IF(KDX.EQ.1 ) AX = DX*UD   /(V1*UA)                                00015480
      IF(KDY.EQ.3 ) DY = DX      / DR                                    00015490
      IF(KDY.EQ.2 ) DY = AY*UA*V1/    UD                                 00015500
      IF(KDY.NE.2 ) AY = DY*UD   /(V1*UA)                                00015510
      IF(KDY.LT.3 ) DR = DX      / DY                                    00015520
      DXT = DX + DM                                                      00015530
      DYT = DY + DM                                                      00015540
      D1  = DXT*UD                                                       00015550
      D2  = SQRT(D1*DYT*UD)                                              00015560
      IF(KG .EQ.3 ) GL = ALOG(2.0)/(GT*UGT*UGL)                          00015570
      IF(KG .GE.2 ) G  = 1.0 + 4.0*GL*UGL*D1/V2                          00015580
      IF(KG .LT.2 ) GL = (G - 1.0)*V2/(4.0*D1*UGL)                       00015590
      IF(KG .LT.3 .AND. GL.GT.0.) GT = ALOG(2.0)/(GL*UGL*UGT)            00015600
      G2 = SQRT(G)                                                       00015610
C     ------------------------------- NORMALIZING VARIABLES ---------00015620
      XD = D1   /(G2*V1)                                                 00015630
      YD = D2   /(G2*V1)                                                 00015640
      TD = R *D1/(G *V2)                                                 00015650
      QD = P *ZM*UZM*D2                                                  00015660
      RETURN                                                             00015670
      END                                                                00015680
C     ================================ CALC ======================00015690
      SUBROUTINE CALC                                                    00015700
      LOGICAL*2     BERR, BATCH,BOP , BOE , BTSS                        00015710
     + , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                   00015720
      COMMON /BIT /                                                      00015730
     +   BERR, BATCH                                                     00015740
     + , BOP , BOE , BTSS                                               00015750
     + , BT1 , BT2 , BT3 , BT4 , BT5 , BT6 , BT7 , BT8                  00015760
      COMMON /INTE/                                                      00015770
     +   LUL , LUM , LUR , LUP , LUE , LUW                              00015780
     + , MCMD                                                           00015790
     + , KDX , KDY , KG  , KLX , KLY , KR                               00015800
```

```
     +   , KTE , KVG , KVP , KVR , KXC , KYC                                  00015810
     +   , NQ  , NQN , NSL , NSN , NSX , NSY                                  00015820
     +   , MQ                                                                 00015830
     +   , KKQX( 10) , KKQM( 10)                                              00015840
         COMMON /REAL/                                                        00015850
     +     UA  , UD  , UGL , UGT , ULC                                        00015860
     +   , UQL , UQA , UQT , UQM , UQR , UQC , UQQ , UQV                      00015870
     +   , URK , URD , UTC , UVV , UVK , UVT , UVI , UVD , UVU , UZM          00015880
     +   , U9C                                                                00015890
     +   , AX  , AY  , C   , CE  , DX  , DY  , DR  , DM  , DXT , DYT          00015900
     +   , G   , GL  , GT  , G2  , P   , QE  , QD                             00015910
     +   , R   , RK  , RP  , RD  , SC  , TC  , TD  , TE  , TPHI               00015920
     +   , V   , VN  , VG  , VP  , VK  , VKN , VT  , VTN , VM                 00015930
     +   , VR  , VI  , VD  , VU                                               00015940
     +   , XC  , X9C , XGL , XGM , XGI , XD                                   00015950
     +   , YC  , Y9C , YGL , YGM , YGI , YD  , ZM                             00015960
     +   , QQXL( 10) , QQXM( 10) , QQYL( 10) , QQYM( 10)                      00015970
     +   , QQA ( 10) , QQT ( 10) , QQV ( 10) , QQ  ( 10)                      00015980
     +   , QQC ( 10) , QQR ( 10) , QQM ( 10)                                  00015990
         DATA WPI / 3.14159265 /                                              00016000
         X  = XC*ULC                                                          00016010
         Y  = YC*ULC                                                          00016020
         T  = TC*UTC                                                          00016030
         E  = 2.0                                                             00016040
         C  = 0.                                                              00016050
         CE = 0.                                                              00016060
          CR = 0.                                                             00016070
          TM = QQT(1)*UQT                                                     00016080
         DO 5390 IQ = 1, NQ                                                   00016090
           TQ = QQT(IQ)*UQT                                                   00016100
            TM = AMAX1(TM,TQ)                                                00016110
           IF(T.GE.0. .AND. TQ.GE.T) GO TO 5390                              00016120
           KQX = KKQX(IQ)                                                    00016130
           IF(KQX.LT.0) GO TO 3000                                           00016140
           XL = QQXL(IQ)*UQL                                                 00016150
           YL = QQYL(IQ)*UQL                                                 00016160
           GO TO (2100, 2200), KQX                                           00016170
2100       CONTINUE                                                          00016180
             NX = 1                                                          00016190
             XI = 0.                                                         00016200
             NY = 1                                                          00016210
             YI = 0.                                                         00016220
             S2 = QQA (IQ)*UQA                                               00016230
             QDA = QD                                                        00016240
             GO TO 2900                                                      00016250
2200       CONTINUE                                                          00016260
             XM = QQXM(IQ)*UQL                                               00016270
             YM = QQYM(IQ)*UQL                                               00016280
             RN = AMAX1(X-XM, XL-X, Y-YM, YL-Y)                              00016290
             IF(RN.LE.0.) GO TO 9100                                         00016300
             XI = XM - XL                                                    00016310
             YI = YM - YL                                                    00016320
             XN = NQN                                                        00016330
             S  = SQRT(AMAX1(50.0*XD*QE*RN/(1.0 + DR), XI*YI/XN) )           00016340
             NX = NQN - MAX1(XN - XI/S, 0.)                                  00016350
             XI = XI/FLOAT(NX)                                               00016360
             NY = NQN - MAX1(XN - YI/S, 0.)                                  00016370
             YI = YI/FLOAT(NY)                                               00016380
             S2 = AMAX1(XI, YI)                                              00016390
             S2 = S2*S2                                                      00016400
             QDA = FLOAT(NX)*FLOAT(NY)*QD                                    00016410
2900       CONTINUE                                                          00016420
           ER = 0.02*S2*(1.0 + DR)/(XD*XD)                                   00016430
           IF(T.GE.0.) ER = ER + 0.2                                         00016440
           QMO = 0.                                                          00016450
3000       CONTINUE                                                          00016460
```

```
          QM = QQM(IQ)*UQM                                                  00016470
          QN = (QM - QMO)/QDA                                               00016480
          YQ = YL + YI*0.5                                                  00016490
          DO 5290 IY = 1, NY                                                00016500
            YN = (Y - YQ)/YD                                                00016510
            XQ = XL + XI*0.5                                                00016520
            DO 5190 IX = 1, NX                                              00016530
              XN = (X - XQ)/XD                                              00016540
              RN = SQRT(XN*XN + YN*YN)                                      00016550
               IF(BTSS) GO TO 5000                                          00016560
              IF(RN.LE.0.) GO TO 9100                                       00016570
              IF(T.LT.0.) GO TO 4500                                        00016580
                TN = (T - TQ)/TD                                            00016590
                WO = (RN - TN)/SQRT(4.0*TN)                                 00016600
                   E = ERFC(WO)                                             00016610
 4500         CONTINUE                                                     00016620
              CY = EXP(-(XN/G2 - RN)*0.5)                                   00016630
              CR = 4.0*SQRT(WPI*RN)                                         00016640
              W  = QN*CY*E/CR                                               00016650
 5001 FORMAT(1X, 5G15.7)                                                    00016660
              IF(BT5) WRITE(LUW, 5001)                                      00016670
     +          X , XQ, XD, XN, CR                                          00016680
     +        , Y , YQ, YD, YN, CY                                          00016690
     +        , T , TQ, TD, TN, E                                           00016700
     +        , QM, QMO,QDA,QN, W                                           00016710
              C  = C + W                                                    00016720
              CE = CE + W*ER/RN                                             00016730
               GO TO 5090                                                   00016740
 5000          CONTINUE                                                    00016750
               CR = AMAX1(CR, RN)                                          00016760
 5090         CONTINUE                                                     00016770
              XQ = XQ + XI                                                  00016780
 5190         CONTINUE                                                     00016790
            YQ = YQ + YI                                                    00016800
 5290       CONTINUE                                                       00016810
          QMO = QM                                                          00016820
 5390     CONTINUE                                                         00016830
         IF(BTSS) GO TO 5500                                                00016840
        C  = C/USC                                                          00016850
        CE = CE/USC                                                         00016860
         GO TO 5990                                                         00016870
 5500    CONTINUE                                                         00016880
         IF(CR .GT. 0.) C = CR / (SQRT(TPHI * TPHI + CR) + TPHI)            00016890
         C = (TM + C * C * TD) / UTC                                        00016900
 5990    CONTINUE                                                         00016910
        BERR = .FALSE.                                                      00016920
        RETURN                                                              00016930
 9100   CONTINUE                                                           00016940
        C  = -1.0                                                           00016950
        CE = 0.0                                                            00016960
        BERR = .TRUE.                                                       00016970
        RETURN                                                              00016980
        END                                                                 00016990
C       ******************************** ERFC ************************** 00017000
         FUNCTION ERFC(WO)                                                  00017010
        DATA E0, E1, E2, E3, E4, E5                                         00017020
     +   / .32759, .25438, -.28540, 1.42141, -1.45315, 1.06141 /            00017030
        W  = 1.0/(1.0 + E0*ABS(WO) )                                        00017040
        W  = W*(E1 + W*(E2 + W*(E3 + W*(E4 + W*E5) ) ) )                    00017050
        E  = W*EXP(-(WO*WO) )                                               00017060
        IF(WO.LT.0.) E = 2.0 - E                                            00017070
         ERFC = E                                                           00017080
         RETURN                                                             00017090
         END                                                                00017100
```

```
**** TSO FOREGROUND HARDCOPY ****
DSNAME=U11236C.BLOCKPC.DATA

I

    A   AR
AX  X DISPERSIVITY  FT
AY  Y DISPERSIVITY  FT
C   D
DX  X DISPERSION    FT2/D
DY  Y DISPERSION    FT2/D
DR  DISPERSION RATIO
DM  MOL. DIFFUSION  FT2/D
E   F   G
GG  DECAY GAMMA
GL  DECAY LAMBDA    1/YR
GT  DECAY HALF-LIFE YR
H   I
IP  IL  IR
L   LX  LY  M
O   OD  OW
OP  OE  OT
P   POROSITY
Q   QN
QL  SOURCE LOCATION FT
QA  SOURCE AREA     FT2
QT  SOURCE TIME     DAYS
QM  MASS FLOW RATE  LBM/D
QR  MASS/AREA RATE  LB/FT2/D
QC  SOURCE CONCENTR.MG/L
QQ  VOLUME FLOW RATEFT3/D
QV  VOLUME/AREA RATEFT/D
QE  ACCURACY
R
RR  RETARDATION
S   SC  SL
SX  SN  SY
T   SAMPLE TIME     DAYS
TE  % STEADY STATE
U
V
VV  VELOCITY        FT/D
XY
X   X LOCATION      FT
Y   Y LOCATION      FT
Z   THICKNESS       FT
.C  CONCENTRATION   MG/L
.T  STEADY STATE    DAYS
.D  .DC .DG
.DP .DQ
.FF .G
.I  .IC .IG
.IP .IQ .TG
 F
 T F F
 F F F F
 F F F F
    6   10
    0    0    0    0
   30   10
    0    0    0    0    0
    0    0    0    0    0
    0    0    0  100
   80    0    6    2
1.0          1.0
```

```
2.737909E-3  365.2422      1.0
1.0          1.0           1.0
1.0          1.0           62.42796E-6
1.0          1.0
1.0          1.0           1.0
1.0          1.0           1.0
1.0          1.0           1.0
1.0          62.42796E-6
0.0          0.1           1.0
```