NUREG/CR-5656
PNL-7510
1A, 1B

EXTRAN: A Computer Code for Estimating Concentrations of Toxic Substances at Control Room Air Intakes

Manuscript Completed: February 1991
Date Published: March 1991

Prepared by
J. V. Ramsdell

Pacific Northwest Laboratory
Richland, WA 99352

Prepared for
Division of Safety Issue Resolution
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555
NRC FIN B2929

ABSTRACT

This report presents the NRC staff with a tool for assessing the potential effects of accidental releases of radioactive materials and toxic substances on habitability of nuclear facility control rooms. The tool is a computer code that estimates concentrations at nuclear facility control room air intakes given information about the release and the environmental conditions.

The name of the computer code is EXTRAN. EXTRAN combines procedures for estimating the amount of airborne material, a Gaussian puff dispersion model, and the most recent algorithms for estimating diffusion coefficients in building wakes. It is a modular computer code, written in FORTRAN-77, that runs on personal computers. It uses a math coprocessor, if present, but does not require one. Code output may be directed to a printer or disk files.

EXECUTIVE SUMMARY

The U.S. Nuclear Regulatory Commission (NRC) staff assesses the potential effects of accidental releases of radioactive material and toxic and hazardous substances on habitability of nuclear facility control rooms. This report, prepared by the Pacific Northwest Laboratory (PNL), provides the NRC staff with improved information and a new tool for use in those assessments. It proposes a new procedure for estimating the concentrations of gases or volatile liquids at nuclear facility control room air intakes that might result from accidental releases. The procedure is incorporated in a computer code called EXTRAN. A second report (Owczarski 1990) describes a procedure for estimating the transport and deposition of material within buildings.

The PNL developed EXTRAN after reviewing procedures used by the NRC for estimating atmospheric transport and diffusion for control room habitability assessments (Murphy and Campe 1974). The review showed that those procedures do not predict variations in the concentrations in building wakes associated with changes in meteorological conditions (Ramsdell 1988). New building-wake diffusion algorithms (Ramsdell 1988, 1990a, 1990b) predict more than half the observed variability. The EXTRAN code combines procedures for estimating the amount of airborne material, a Gaussian puff model, and the most recent of the building-wake diffusion coefficient algorithms.

Given hazardous material in a tank, users postulate a release scenario, the environmental conditions, and the physical and chemical characteristics of the material. From this information, EXTRAN computes a concentration time history at the control room air intake. Specifically, it estimates concentrations from the time the substance first arrives at the air intake until actions can be taken to protect control room occupants.

EXTRAN is a modular computer code that runs on personal computers using the MSDOS operating system. It is written in FORTRAN-77 and uses a math coprocessor, if present. It does not require a coprocessor. Code output may be directed to a printer or disk files. The "Good Practices Standard" of the PNL Quality Assurance program governed development of the EXTRAN computer code. Early versions of the code were tested by the NRC staff. This version incorporates revisions based on their comments and suggestions.

The three major components of the report are found in Chapters 2 through 4. Chapter 2 is a user's guide to the EXTRAN code. Chapter 3 describes the technical bases for the EXTRAN code, and Chapter 4 describes the code.

ACKNOWLEDGMENTS

This report was prepared for the Division of Reactor Accident Analysis, Office of Nuclear Regulatory Research, of the U.S. Nuclear Regulatory Commission. The NRC Project Manager was Charles Ferrell. Charles Ferrell and other members of the NRC staff contributed significantly to the development of the EXTRAN computer code. They tested and suggested revisions to various versions of the code. Their assistance is appreciated.

CONTENTS

FIGURES

TABLES

1.0 INTRODUCTION

The U.S. Nuclear Regulatory Commission (NRC) staff assesses the potential effects of accidental releases of radioactive and toxic substances on the habitability of nuclear facility control rooms. This report proposes a new procedure for estimating the concentrations at nuclear facility control room air intakes. A computer code called EXTRAN presents the procedure in an easy-to-use form. EXTRAN runs on personal computers using the MSDOS operating system.

The Pacific Northwest Laboratory developed EXTRAN after reviewing procedures used by the NRC for estimating atmospheric transport and diffusion for control room habitability assessments (Murphy and Campe 1974). The review showed that those procedures do not predict variations in the concentrations in building wakes associated with changes in meteorological conditions (Ramsdell 1988). New building-wake diffusion algorithms (Ramsdell 1988, 1990a, 1990b) predict more than half the observed variability. The EXTRAN code combines procedures for estimating the amount of airborne material, a Gaussian puff model, and the most recent of the building-wake diffusion coefficient algorithms.

Given hazardous material in a tank, users postulate a release scenario, the environmental conditions, and the physical and chemical characteristics of the material. From this information, EXTRAN computes a concentration time history at the control room air intake. Specifically, it estimates concentrations from the time the substance first arrives at the air intake until actions can be taken to protect control room occupants.

This report has three main parts. Chapter 2 is a user's guide to the EXTRAN code. Chapter 3 describes the technical bases for the EXTRAN code, and Chapter 4 describes the code.

The user's guide discusses the code options and the required input. Using the code options, users may send the results to a printer or to disk files for retention. Following selection of the output options, input occurs in three stages. The first stage establishes the physical relationship between the tank and air intake. The second stage sets the environmental conditions, and the third stage establishes the physical characteristics of the material to be released. Each stage ends by listing the input and providing an opportunity to change it.

Chapter 3 presents the equations EXTRAN uses to estimate the transfer of material from the tank to the atmosphere. This transfer may occur in one step, as in release of a gas. Alternatively, it may involve the formation and evaporation of a liquid pool. EXTRAN then calculates transport and diffusion of the material in the atmosphere using a puff dispersion model. The puff model includes a new algorithm (Ramsdell 1990a,b) for estimating diffusion in building wakes.

EXTRAN is a modular computer code. Chapter 4 discusses the organization of the code and describes the code modules. The code is written in FORTRAN-77. The compiled version of the code uses a math coprocessor, if present, but it doesn't require one. Code output may be directed to a printer or disk files.

EXTRAN has several features that allow users to check their work. The code checks for input errors. These checks include type and range checks where appropriate. EXTRAN echoes entries that appear to be correct. As users complete each input stage, EXTRAN lists the entries for review and possible modification. All computer output contains a title entered by the user and the program execution date and time. The output includes a plot of the concentration at the intake and an output file that contains mass and energy balances for the source.

The "Good Practices Standard" of the PNL Quality Assurance program governed development of the EXTRAN computer code. The NRC staff tested and commented on early versions of the code. This version of the code incorporates revisions based on their comments and suggestions.

2.0 USER'S GUIDE

The physical setting assumed in the EXTRAN code is simple. It is a tank containing a hazardous substance that is directly upwind of a control room air intake. The hazardous substance may be gas, a liquefied gas, a volatile liquid, or fine particulate material suspended in a gas. As the simulation begins, the tank fails, releasing the toxic material to the environment. When you run EXTRAN, you will be asked to describe the tank failure, the physical relationship between the tank and the intake, the environmental conditions, and the physical characteristics of the material in the tank.

The work sheet shown in Figure 2.1 will help you in using the EXTRAN code. It lists the model options and input. You can record the information that will be needed by the program in the space provided on the work sheet. If you do, you will have organized the information in the order in which the program requests it.

When you start EXTRAN, the first thing that you will see is a screen containing information about the program. This information includes the program title and version, the NRC organization sponsoring the work, the date of the version, and names of contacts for questions about the code. The program pauses at this point to allow you time to read the information.

To continue program execution, press the space bar or the ENTER (RETURN) key. EXTRAN will request a program title. This is your opportunity to enter a short description of the problem that you are modeling. The description is a single line of text that may contain up to 70 characters and blanks. Whatever you enter will appear on all EXTRAN output. If you choose, the title may be blank.

When you press the ENTER key after entering the title, EXTRAN asks you to select from among the options available for output of the results. You will then select a release type and enter input data corresponding to the release type. The four columns on the left side of the work sheet show the status of each input item--required (R), optional (O), or not used (N). Required items must be entered. A zero may be entered or the ENTER key may be pressed for optional items, if no better information is available. You will not see prompts for the information marked as not used.

The remainder of the User's Guide describes the output options and the data to be entered.

2.1 OUTPUT OPTIONS

There are three types of output from the EXTRAN computer code. The first type is a listing of the input data followed by a summary of the predicted concentrations. This is the primary output of the program. The other two types of output provide more detailed information. The second type is a chronology of the predicted concentrations at the control room air intake. It may be used as input to codes that estimate concentrations within the control

Run Title: _____

Output Options: Concentration Units ____ g/m^3 ____ ppm ____ mCi/m^3

Primary Output	____ Printer	____ File
Plot	____ Printer	
Conc. Chronology	____ Printer	____ File
Mass & Energy Bal.		____ File

Release Type:

(1) Liquid Tank Burst	____
(2) Liquid Tank Leak	____
(3) Gas Tank Burst	____
(4) Gas Tank Leak	____

Release Type 1	2	3	4		
R	R	R	R	Initial Mass (kg or Ci)	_____
N	R	N	R	Release Rate (kg/sec or Ci/sec)	_____
O	O	O	O	Release Height (m)	_____
R	R	R	R	Storage Temp. (°C)	_____
O	O	N	N	Max. Pool Radius (m)	_____
R	R	R	R	Distance to Intake (m)	_____
O	O	O	O	Intake Height (m)	_____
O	O	O	O	Building Area (m^2)	_____
N	N	N	O	Vent Flow (m^3/sec)	_____

R	R	R	R	Wind Speed (m/s)	____	____	____
R	R	R	R	Atmospheric Stability	____	____	____
R	R	R	R	Air Temperature (°C)	____	____	____
R	R	R	R	Atm. Pressure (mm Hg)	____	____	____
R	R	N	N	Solar Radiation (w/m^2)	____	____	____
R	R	N	N	Cloud Cover (tenths)	____	____	____
R	R	N	N	Ground Temperature (°C)	____	____	____

O	O	O	O	Chemical Name	_____
R	R	R	R	Molecular Wt. (g/mole)	_____
R	R	N	N	Boiling Point (°C)	_____
R	R	N	N	Liq. Heat Cap. (j/g-°C)	_____
R	R	N	N	Heat of Vap. (j/g)	_____
R	R	N	N	Spec. Gravity	_____
O	O	N	N	Mol. Diff. Coef. (cm^2/sec)	_____

Output File Names:

Primary Output	EXPR_____.___	EXPR_____.___	EXPR_____.___
Conc. Chronology	EXCR_____.___	EXCR_____.___	EXCR_____.___
Mass & Energy Bal.	EXMB_____.___	EXMB_____.___	EXMB_____.___

Date: _____ Time: _____

FIGURE 2.1. EXTRAN Version 1.2 Work Sheet

room. The third type of output contains mass balances for the source and the pool and an energy budget for the pool. It provides details that may be used to check EXTRAN computations for liquid releases.

2.1.1 Concentration Units

Prior to selecting from among the output types, you will be asked to select concentration units for the output. EXTRAN may be used to estimate concentrations of either toxic or radioactive materials. It includes three sets of units for expressing concentrations. The default set of units is grams per cubic meter. It may not be convenient to compare results expressed in these units with standards or to use the results as input to dose models. Therefore, you have the option of two alternate methods of expressing concentrations. EXTRAN can compute concentrations of toxic materials in parts per million, and it can compute concentrations of radioactive materials in millicuries per cubic meter.

You will be asked if you want the concentrations expressed as parts per million. If you respond with a Y or y, output will be in parts per million. If you respond with an N or n, you will be asked if you want concentrations in millicuries per cubic meter. If you respond with a Y or y, output will be in millicuries per cubic meter. Otherwise, the output will be in grams per cubic meter. In any case, the remaining program prompts will reflect your choice of output type.

2.1.2 Primary Output

EXTRAN's primary output is a summary that begins with a header. The header contains the title given to the model run as a part of the input and the date and time of the EXTRAN run. A complete listing of the input data follows the heading. Computed model parameters follow the heading. Then in the first 2 min come the average concentration and exposure (time-integrated concentration). Finally, EXTRAN lists the time from plume arrival to the maximum concentration and the maximum concentration. The primary output ends with a listing of the names of files containing the model results.

The input data lists are in sections that correspond to the types of information required by the model. The first block of data describes the SCENARIO, the second the ENVIRONMENTAL CONDITIONS, and the third the EFFLUENT CHARACTERISTICS. Data in these lists should be identical with the data you enter. The next section discusses these data in detail.

The MODEL PARAMETERS section of the summary lists information computed from the input data. Parameters in the list include the interval between puffs, the time step used in the concentration and exposure computations, the time delay between the release and the arrival of the substance at the air intake, and the concentration used as an indicator of the arrival of the plume at the intake. It also contains a concentration conversion factor if you chose concentration units of grams per cubic meter or parts per million.

EXTRAN simulates plumes with a series of puffs. The numerical accuracy of the simulation is a function of the spacing between puffs. If the puffs

are too far apart or the time step is too large, there can be large differences between concentrations estimated with the puff model and those estimated with plume models. EXTRAN uses the criteria discussed in Ramsdell et al. (1983) for determining appropriate spacing between puffs and sampling intervals. The minimum interval between puffs is 1 sec. This interval increases with increasing distance between the release point and the air intake. The maximum interval is 10 sec if the distance to the intake is less than 1000 m and 30 sec if the distance is 1000 m or greater. The first item in the list of model parameters is the puff release interval selected by the model.

The next item in the list is the concentration computation interval. It is called the time step. EXTRAN generally computes concentrations at twice the puff release frequency. But, if it releases puffs at 1-sec intervals, it will use 1-sec intervals in concentration calculations.

Except in unusual circumstances there will be a delay between the accidental release of toxic substances and the arrival of the substance at the air intake. The time delay listed in the summary output is an estimate of this interval. After the data input is complete, EXTRAN estimates the magnitude of the maximum concentration in the first puff to reach the intake. The intake concentration threshold is four orders of magnitude lower than this maximum. The time delay is the elapsed time between the release and the time the concentration at the intake exceeds the threshold. It also serves as time zero for the concentration chronologies.

The threshold concentration is just below the time delay in the listing in the primary output. It is generally lower than the first concentration listed in the chronology because there is a large concentration gradient on the leading edge of the first puff. In addition, overlapping puffs overlap give concentrations that are higher than the concentration computed for a single puff.

The last section of the summary output gives results of the computations for the 2 min immediately following the arrival of the plume. These 2 min are an estimated response time for taking protective actions in the control room. The first item in the list is the average concentration at the air intake during this period. The concentration may be in grams per cubic meter, parts per million, or millicuries per cubic meter. The exposure, which follows the average concentration, is the product of the average concentration and time. In this case, the time is 2 min. The last two lines give the time of occurrence of the maximum concentration and the maximum concentration in the first 2-min period. The time of occurrence of the maximum is seconds after the arrival of the plume at the intake. The scenario determines the time of the maximum concentration. If the scenario involves a catastrophic failure of a liquefied gas tank and a short distance, the maximum will occur within a few seconds after the arrival of the plume. If the scenario is a tank leak or the distance between the release point and intake is large, the maximum is likely to occur after the 2-min period.

You may choose the disposition of the primary output, but you can't choose to bypass it. The primary output may be written to a printer as the

code executes. It may be written to a file, or it may be printed and written to a file. Figure 2.2 shows an example of output written to a printer.

If you choose to have EXTRAN create a file with the primary output, the file will be written to the active disk drive. If you are running the program from a floppy disk, the file will be written to that disk. EXTRAN generates file names as it runs. Names for the primary output have the form EXPRhhmm.nnn. The hh and mm give the hour and minute of the day of the execution of EXTRAN. The nnn is a three digit number that shows the set of environmental conditions for the program execution beginning at hhmm. For example, consider the file name EXPR1118.001. The EXPR shows that the file contains the primary output from EXTRAN. The 1118 shows that the EXTRAN run started at 11:18 am, and the .001 shows that the output is for the first set of environmental conditions submitted during the run. If you choose to rerun EXTRAN with different environmental conditions without quitting, the primary output for the second set of conditions would have a file name EXPR1118.002.

Program Title: EXTRAN VERSION 1.2

Developed For: U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Division of Reactor Accident Analysis

Date: October 1990

NRC Contact(s): C. Ferrell Phone: (FTS) 492 3944
Code Developer: J. V. Ramsdell Phone: (509) 376-8626
(FTS) 444-8626

Code Documentation:
EXTRAN: A Computer Code For Estimating
Concentrations Of Toxic Substances At
Control Room Air Intakes
NUREG/CR-5656

FIGURE 2.2. EXTRAN Primary Output

```
Example 1  --  EXTRAN Version 1.2

RUN DATE = 10/11/1990  RUN TIME = 15:34:24

SCENARIO:
Release Type                =  Liquid Tank Burst
Initial Mass          (kg)  =       1000.
Release Height         (m)  =          .0
Storage Temperature    (C)  =         20.0
Maximum Pool Radius    (m)  =         10.0
Intake Distance        (m)  =        250.
Intake Height          (m)  =         25.0
Building Area       (m**2)  =        100.

ENVIRONMENTAL CONDITIONS:
Wind Speed                (m/sec) =        3.0
Atmospheric Stability Class       =        4
Air Temperature                (C) =       20.0
Atmospheric Pressure       (mm Hg) =      760.0
Solar Radiation       (watts/m**2) =      650.0
Cloud Cover              (tenths) =        2
Ground Temperature             (C) =       25.0

EFFLUENT CHARACTERISTICS:
Material Released                 =      CHLORINE
Molecular Weight        (gm/mole) =       70.9
Initial Boiling Point          (C) =      -34.1
Heat Capacity           (j/gm-C) =         .946
Heat of Vapor.            (j/gm) =      288.0
Specific Gravity                  =        1.570
Diffusion Coef.       (cm**2/sec) =         .079

MODEL PARAMETERS:
Puff Release Interval              (sec) =   10
Time Step                          (sec) =    5
Delay Between Release and Intake   (sec) =   50
Threshold Concentration       (g/m**3) =    7.08E-05
To convert g/m**3 to ppm, multiply by      3.39E+02

RESULTS:
Average Concentration During First Two Minutes
     After Arrival of Plume                (g/m**3) =    1.93E-01
Exposure Two Minutes After Arrival    (g-sec/m**3) =    2.41E+01
Time From Plume Arrival to Max. Conc.        (sec) =   30.
Max. Conc. in Two Minutes After Arrival   (g/m**3) =    6.12E-01

ADDITIONAL OUTPUT FILES:
     EXCR1534.001
     EXMB1534.001
```

FIGURE 2.2. (contd)

2.1.3 Concentration Chronology

The primary output lists the time required to travel from the tank to the intake, the threshold concentration that marks the arrival of the plume, and the average and maximum concentrations during the first 2 min after plume arrival. It doesn't provide the details of the concentration variation following plume arrival at the air intake. EXTRAN generates a concentration chronology that provides this information. The chronology is the second type of output generated by EXTRAN.

EXTRAN has three ways to present the concentration chronology. The chronology may be printed; it may be written to a file, and it may be plotted by the printer. The plot provides a quick indication of the concentrations, while the other two forms provide numerical values.

Figure 2.3 shows a typical concentration plot. Elapsed time, on the horizontal axis, starts at zero with the arrival of the plume at the intake. The left axis gives logarithms (base 10) of the instantaneous and average concentrations. The symbol used for the instantaneous concentration is a "C," and an "A" is used for the average concentration. The right axis gives logarithms of the exposure (time-integrated concentration). An "E" is the symbol for exposures. Occasionally, when the curves cross, more than one symbol would occupy the same location on the graph. Only one symbol is plotted in these cases. The order of precedence for plotting symbols is instantaneous concentration, exposure, then average concentration.

The numerical values along the left and right axes are logarithms, but they are also powers of 10. For example, a logarithm of 3 is equivalent to 10^3, and a logarithm of 3.5 is equivalent to a concentration of 3.2×10^3. In contrast, a logarithm of -3.5 is equivalent to a concentration of 3.2×10^{-4}.

The printer listing and file output provide the details of the chronology. They contain the instantaneous concentration, exposure, and average (mean) concentration at the intake each time step after the concentration exceeds the threshold. They also list the total number of number of puffs since the beginning of the release. The times and concentrations in these listings are more precise than those estimated from the graph. The length of the chronology is a function of the distance between release point and intake, wind speed and atmospheric stability. If the release occurs in a building wake, the length is also a function of the building area. The listing may extend for as many as three pages. Figure 2.4 shows part of a listing as an example of the printer and file output.

If you choose to have EXTRAN create a chronology file, the file will be written to the active disk drive. If you are running the program from a floppy disk, the file will be written to that disk. Names for chronology files have the form EXCRhhmm.nnn, where hh and mm are the hour and minute of the day that the program started, and nnn is a three digit number that shows the set of environmental conditions. The numerical portions of the chronology files have the same values as the corresponding parts of the primary output file. For example, if the primary output file has the name EXPR1435.001, the corresponding chronology file will have the name EXCR1435.001.

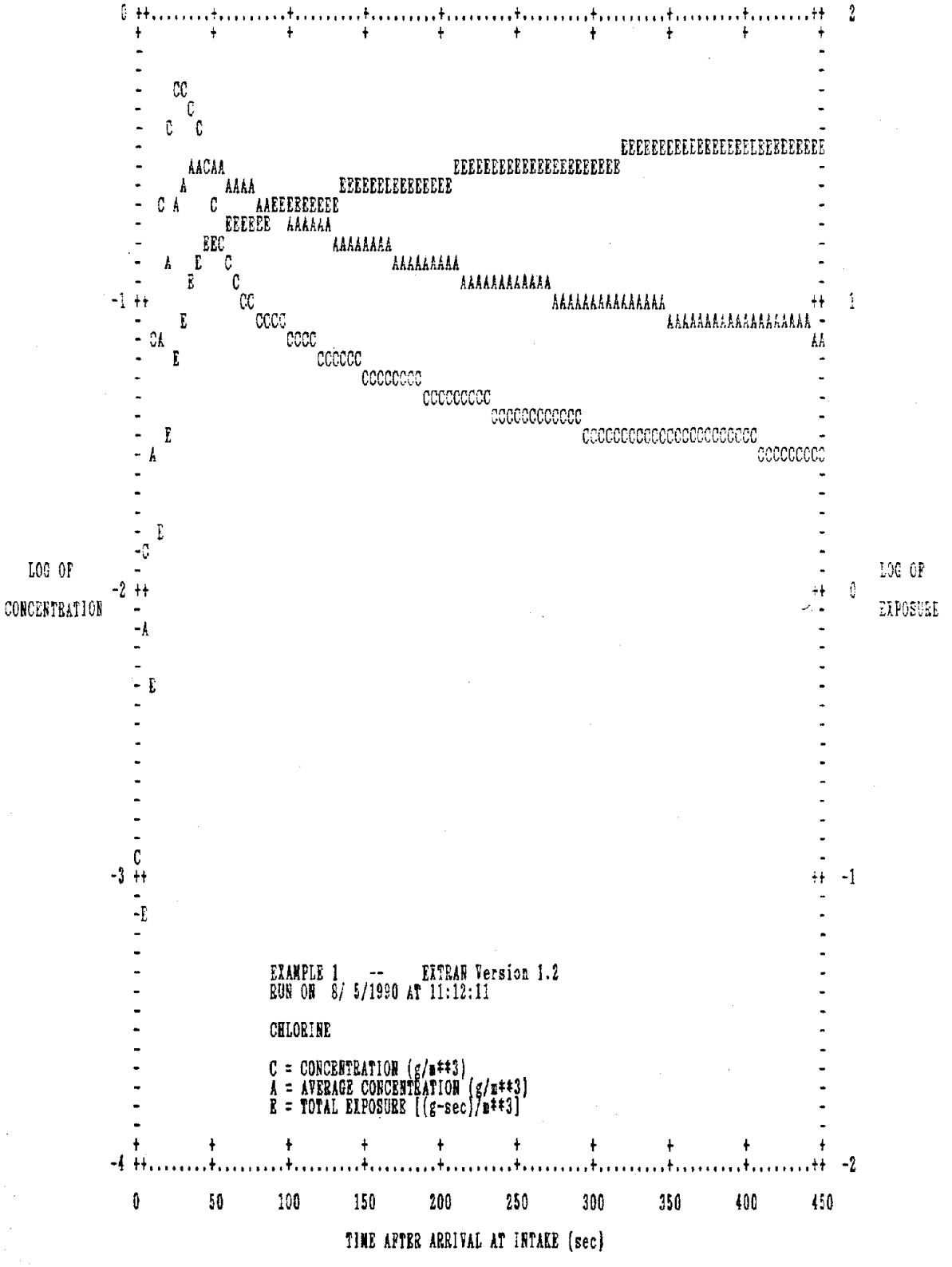

FIGURE 2.3. EXTRAN Concentration Chronology Plot

Example 1 -- EXTRAN Version 1.2
Run on 10/11/1990 at 15:34:24

TIME (sec)	CONCENTRATION (g/m**3)	EXPOSURE (g-sec/m**3)	MEAN CONC. (g/m**3)	NUM OF PUFFS
0	1.27E-03	6.35E-03	1.27E-03	7
5	1.50E-02	8.13E-02	8.13E-03	7
10	7.92E-02	4.77E-01	3.18E-02	8
15	2.32E-01	1.64E+00	8.18E-02	8
20	4.35E-01	3.81E+00	1.52E-01	9
25	5.85E-01	6.74E+00	2.25E-01	9
30	6.12E-01	9.80E+00	2.80E-01	10
35	5.32E-01	1.25E+01	3.11E-01	10
40	4.11E-01	1.45E+01	3.22E-01	11
45	2.99E-01	1.60E+01	3.20E-01	11
50	2.19E-01	1.71E+01	3.11E-01	12
55	1.69E-01	1.80E+01	2.99E-01	12

FIGURE 2.4. EXTRAN Concentration Chronology File Listing

2.1.4 Mass and Energy Balance

The third type of output generated by EXTRAN contains a mass and energy balance for the source at each puff release time. This information is useful in understanding the changes in the physical scenario that take place during releases of liquids. It isn't particularly useful for gaseous releases.

If you choose to have EXTRAN create a mass and energy balance file, the file will be written to the active disk drive. If you are running the program from a floppy disk, the file will be written to that disk. Names for mass and energy balance files have the form EXMBhhmm.nnn, where hh and mm are the hour and minute of the day that the program started, and nnn is a three digit number that shows the set of environmental conditions. The numerical values in the file name are the same as those in the primary output file name. For example, if the primary output file has the name EXPR1435.001, the corresponding mass and energy balance file will have the name EXMB1435.001.

Figures 2.5 and 2.6 show portions of mass and energy balance files. The example in Figure 2.5 is for a catastrophic failure of a 1000-kg liquid chlorine tank, and the example in Figure 2.6 is for a leak from a tank with 3000 kg of carbon tetrachloride. Each mass and energy balance file starts with a heading. The heading includes the run title, the date and time EXTRAN started, and a data format description. Each time EXTRAN releases a puff, it writes an entry to the file. After each 10 entries, it repeats the heading.

There are four lines in each entry in the mass and energy balance file. The first line gives the time after the beginning of the release and the total number of puffs released. The second line contains a mass balance for the

```
Example 1  --  EXTRAN Version 1.2
Run on 10/11/1990 at 15:34:24

DATA FORMAT:
MASS BALANCE:  TANK, CURRENT RELEASE, POOL, FLASHED, EVAPORATED
POOL STATUS:   VOLUME, RADIUS, AREA, DEPTH, TEMPERATURE
ENERGY BUDGET: NET SW, NET LW, ATM CONV, GRND COND, NET FLUX

TIME SINCE RELEASE =    0     TOTAL NUMBER OF PUFFS RELEASED =    2
MASS BALANCE         .00   1000.00    785.73    177.70     36.57
POOL STATUS          .52      4.08     52.38       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67  18689.06  20107.62
TIME SINCE RELEASE =   10     TOTAL NUMBER OF PUFFS RELEASED =    3
MASS BALANCE         .00       .00    760.30       .00     25.43
POOL STATUS          .50      3.99     50.05       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67  13215.16  14633.72
TIME SINCE RELEASE =   20     TOTAL NUMBER OF PUFFS RELEASED =    4
MASS BALANCE         .00       .00    739.77       .00     20.53
POOL STATUS          .48      3.93     48.43       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67  10790.13  12208.69
TIME SINCE RELEASE =   30     TOTAL NUMBER OF PUFFS RELEASED =    5
MASS BALANCE         .00       .00    722.16       .00     17.61
POOL STATUS          .47      3.87     47.12       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67   9344.53  10763.09
```

FIGURE 2.5. EXTRAN Mass and Energy Balance File Listing (Catastrophic Tank Failure)

source and puffs. The third line describes the pool status, and the last line gives the energy budget for the pool.

There are five elements listed in the mass balance line. All entries are masses in kilograms. The first element is the mass remaining in the tank. If the scenario involves a catastrophic tank failure, the mass remaining in the tank should be zero. The second element is the mass released from the tank in the current interval. The third element is the mass remaining in the pool after evaporation. Finally, the last two elements give the mass assigned to the new puffs. The first of these elements (fourth element on the line) is the mass of a liquefied gas that flashes on release from the tank. If the material released is a volatile liquid, the mass in this puff will remain zero. Finally, the last element in the mass balance line is the mass evaporating from the pool. EXTRAN computes this mass from the area of the pool, the pool energy budget, and the puff release interval. If both the fourth and fifth elements are greater than zero, two puffs will be released.

The sum of the mass in the tank and the mass released in each entry should equal the mass in the tank in the previous entry. Similarly, the mass in the pool in an entry should equal the mass in the pool in the previous

```
Example 2  --  EXTRAN Version 1.2
Run on 10/11/1990 at 15:22:11

DATA FORMAT:
MASS BALANCE:  TANK, CURRENT RELEASE, POOL, FLASHED, EVAPORATED
POOL STATUS:   VOLUME, RADIUS, AREA, DEPTH, TEMPERATURE
ENERGY BUDGET: NET SW, NET LW, ATM CONV, GRND COND, NET FLUX

TIME SINCE RELEASE =    0     TOTAL NUMBER OF PUFFS RELEASED =    1
MASS BALANCE    2940.00     60.00     59.90       .00       .10
POOL STATUS         .04      1.09      3.75       .01     14.60
ENERGY BUDGET    225.00   -111.24    -87.86       .00     25.90
TIME SINCE RELEASE =    6     TOTAL NUMBER OF PUFFS RELEASED =    2
MASS BALANCE    2880.00     60.00    119.70       .00       .19
POOL STATUS         .07      1.54      7.49       .01     14.46
ENERGY BUDGET    225.00   -110.16    -84.36     57.53     88.02
TIME SINCE RELEASE =   12     TOTAL NUMBER OF PUFFS RELEASED =    3
MASS BALANCE    2820.00     60.00    179.43       .00       .28
POOL STATUS         .11      1.89     11.23       .01     14.33
ENERGY BUDGET    225.00   -109.29    -81.54     84.75    118.91
TIME SINCE RELEASE =   18     TOTAL NUMBER OF PUFFS RELEASED =    4
MASS BALANCE    2760.00     60.00    239.07       .00       .36
POOL STATUS         .15      2.18     14.96       .01     14.21
ENERGY BUDGET    225.00   -108.53    -79.05    102.28    139.70
```

FIGURE 2.6. EXTRAN Mass and Energy Balance File Listing (Tank Leak)

entry plus the mass released from the tank minus the mass in the puffs. Occasionally, round-off and truncation will keep the equalities from being satisfied. When differences occur they should be small.

The pool status line also includes five elements. The first four elements are the pool volume, radius, area, and thickness. Volume is in cubic meters, area is in square meters, and radius and thickness are in meters. The last element in the pool status line is the pool temperature. It is in degrees Celsius. All values represent the conditions at the end of the period, after evaporation has taken place.

The last line of each entry is the energy budget of the pool. All entries in this line are fluxes in watts per square meter. Positive values indicate energy transfer to the pool, and negative values indicate energy transfer from the pool.

The first element in the pool energy budget is the net solar radiation flux. It is the solar radiation adjusted for the albedo of the pool. EXTRAN assumes an albedo of 0.1. This assumption, based on the albedo of water, should be conservative. Long-wave radiation flux to the pool from the atmosphere is the second element in the energy budget. It is a function of the air temperature. Long-wave radiation from the pool computed from the pool temperature is the third element. The difference between the incoming and

outgoing long-wave radiation (i.e., the net long-wave radiation) is the fourth element in the line. In general, the net long-wave radiation should be less than the net solar radiation during the day. The last two elements in the energy budget line are the heat flux to the pool from the air and from the ground, respectively. Typically the heat flux from the ground is the dominant term in energy budget.

The energy budget line doesn't include one important term. That term is the latent heat flux associated with evaporation from the pool. EXTRAN computes the latent heat flux from the mass evaporating from the pool and the rate of evaporation from the vapor pressure of the liquid in the pool. It then adjusts the temperature of the pool to account for the difference between the latent heat loss and the net energy gained from radiation, conduction, and convection. If there is a net energy loss from the pool, the temperature of the pool will decrease. If there is a net gain of energy by the pool, the temperature will increase until it reaches the boiling point. After the pool temperature reaches the boiling point, all additional energy increases evaporation.

2.2 DATA ENTRY

EXTRAN uses a series of prompts to obtain the information needed for program execution. The prompts requesting data are arranged in three groups. The first of these groups defines the physical characteristics of the scenario. The second group defines environmental conditions, and the last group defines the characteristics of the material. When appropriate, the prompts show the required units for numerical input. If the prompt doesn't specify units, the input is dimensionless.

In general, you will enter data on the same line as the prompt so the input appears to follow the prompt. Pressing the ENTER key signals the end of an entry. Input may be changed at any time prior to pressing the ENTER key by using the BACKSPACE or DEL key and retyping the input. EXTRAN performs range checks on data entries. If an entry is within the acceptable range, the entry will be rewritten below the prompt, and the next prompt will appear. Otherwise, the prompt will reappear, or an error message will appear. If the prompt reappears, the program had a problem reading the input. For example, the letter o may have been input in a numeric response in stead of a zero. If the input is out of range, an error message will state the nature of problem.

2.2.1 Release Scenario

The input used to define the physical scenario associated with a potential accident is discussed below. This input deals with the type of accident, amount of material, where the release occurs in relation to the intake, and how the material enters the atmosphere.

Failure Mode

EXTRAN has four options for release of hazardous material. The first two deal with materials stored as liquids, and the other two deal with materials

stored as gases. In both cases there is one option in which the tank fails catastrophically and another option in which the tank develops a leak. In the first of these options, the tank empties instantaneously. In the second, it doesn't. Generally, the catastrophic failure results in a greater maximum concentration than the tank leak mode. But, the tank leak may be the more realistic failure mode. It is possible to approximate a catastrophic failure in the leak failure mode by increasing the magnitude of the leak rate.

The four combinations of material and tank failure are a part of the prompt that requests a release type. The only acceptable responses are 1, 2, 3, or 4. If you make any other response, the prompt will be repeated.

Initial Mass

After you select a tank failure mode, you will be asked to enter the amount of material in the tank. The specific wording of the prompt depends on the concentration units that you chose as an output option. If you requested concentrations in millicuries per cubic meter, you will be asked to enter the amount of material in curies. Otherwise, you will be asked to enter the amount of material in kilograms. In either case, enter the amount in tank at the time of the failure, not the amount of material that the tank could contain.

If you are entering curies, any positive number greater than zero and less than 1.0×10^6 will be accepted. If you are entering kilograms, numbers between 1 and 1×10^6 will be accepted. As a point of reference, the 1 m^3 of water has a mass of about 1000 kg.

Release Rate

If you select a leak failure mode, you will be asked to enter a release rate. Acceptable responses to the release rate prompt range from near zero (but not zero) to a rate one half of the total mass per second. The release rate will remain constant at the entered value until the tank is empty. When the tank is empty, the release rate will be set to zero. If you enter a release rate that is too small or too large, the prompt will be repeated until you make an acceptable response.

Release Height

Following the request for the tank temperature, you will be asked to enter a release height. For a liquid, this is the height of the pool above the ground and not the height of the hole in the tank. For a gas, the release height may be the height of the hole. Release heights greater than zero may result from failure of a tank on a flat roof or loss of the top of a large tank. In the first of these situations the pool might be formed on the roof, and in the second, the pool may be within the original tank. The range of acceptable responses to the request for release height is 0.0 to 100.0 m. You may enter a zero or press RETURN if the release height is zero. Use ground level at the control room air intake as a reference point.

Tank Temperature

You must enter a temperature for the storage tank for all release types. This temperature determines the expansion of gases and the rate of evaporation of liquids. The prompt and output refer to this temperature as the tank temperature. The tank temperature should be near the air temperature unless the tank is cooled or heated. The range of acceptable tank temperatures is -40 to 50°C.

Consider factors such as the size of the tank, the exposure of the tank to solar radiation, the insulation of tank, and perhaps the time of day and time of year in estimating the tank temperature. A temperature near the day time high temperature for the season may be reasonable choice for the tank temperature.

Maximum Pool Radius

Berms frequently surround large tanks to contain spills. If you specify a liquid tank burst or leak, you will be asked to specify a maximum pool radius. The acceptable responses are from 0 through 50. If you enter a zero or press RETURN, EXTRAN will assume that a berm does not exist. The liquid will spread without horizontal restriction until the pool depth decreases to 0.01 m.

Intake Distance

Concentrations in plumes decrease as the distance from the release point increases. When asked for the intake distance, enter the distance between the tank and the control room air intake. If there are structures between the tank and the intake, enter a distance corresponding to the shortest possible path. The range of acceptable distances is 5 to 1999 m.

Intake Height

The reference plane for heights in EXTRAN is ground level near the control room air intake. When asked to enter the intake height, enter the height of the control room air intake above to this reference plane. The range of acceptable values for the intake heights is 0.0 to 100.0 m.

Concentration estimates are sensitive to the difference between the release height and the intake height when distance between the tank and intake is small. As the distance increases, this sensitivity decreases. Maximum concentration estimates can be obtained by setting both the release height and intake height to zero. If the release and intake heights are the same, but are not equal to zero, the concentration will be up to a factor of two lower than when the heights are equal to zero.

Building Area

EXTRAN includes a new model to estimate diffusion in the wakes of structures. To use this model, you must enter an estimate of the cross-sectional

area of the structure projected on a plane perpendicular to the wind direction. The acceptable range for building areas is 0 to 3000 m^2. Note that the area requested is the area of the tank or a structure near the tank. It is not necessarily the area of the reactor containment building or of the control room air intake structure. If the area is zero, the enhanced diffusion due to the wake of the structure will be neglected.

Vent Flow

If the release type you selected is a gas tank leak, you will be asked to enter a vent flow. This prompt allows you to model releases through building vents. If the release is through a vent enter the vent flow. Otherwise enter a zero or press RETURN. Acceptable flows range from zero through 50 m^3/sec.

The vent flow is particularly important when the distance between the release point and air intake is small. It prevents the concentration at the intake from becoming unreasonably large. When the vent flow is greater than zero, concentrations in the environment can't become larger than the concentration in the vent.

2.2.2 Environmental Conditions

After you define the physical scenario, the next group of prompts requests information on the environmental conditions at the time of the release. These conditions remain constant during the model run. The environmental variables used in EXTRAN enter the calculations in various ways. As a result, no single set of variables can be given the label "worst case." Worst case conditions must be determined by trial and error considering the physical scenario and the material. For example, sometimes worst case conditions involve low wind speeds. However, if the release is in a building wake and involves a volatile liquid, worst case conditions involve high wind speeds.

In selecting a set of conditions for EXTRAN, maintain consistency within the set. For example, a combination of high temperature, high solar radiation, and strongly stable atmospheric conditions leads to larger concentrations than high temperature, high solar radiation, and unstable atmospheric conditions. But, the latter combination is realistic, while the former is not. Thus, the former set ought not to be used in control room habitability assessments.

Wind Speed

The first environmental variable entered is the wind speed. When requested, enter the wind speed at a height of 10 m above ground level. The range of acceptable wind speeds is from 0.5 to 20.0 m/s.

The wind speed is used in computing evaporation rates, diffusion, and transport. Depending on the specific scenario, increasing the wind speed may increase or decrease the predicted maximum concentration. If the scenario involves evaporation or wake diffusion, increasing the wind speed is likely to result in an increase in the maximum concentration. Otherwise, an increase in wind speed will decrease the concentration.

Atmospheric Stability Class

Atmospheric stability classes defined in Regulatory Guide 1.23 are used in EXTRAN. Enter the class designation as a number rather than as a letter. The prompt requesting stability class gives a key for translating the letter designation in Regulatory Guide 1.23 to numerical values. When requested enter the number for the appropriate stability class. The only acceptable entries are the integers 1 through 7. Any other response will result in an error message and will cause the prompt to be repeated.

Atmospheric stability is a factor in determining the dilution of the plume between the pool and the air intake. If there isn't a wake involved, the stability has a significant effect on the maximum concentration. Otherwise, the effect of stability on concentration is small.

Air Temperature

After you enter the atmospheric stability, you will be asked for the ambient air temperature. The ambient air temperature determines if a liquid will flash or not. It is a factor in the energy balance that determines the rate of evaporation from pools, and it determines the volume occupied by gases. In general, increasing the air temperature increases the maximum concentration at the air intake following liquid releases and decreases the concentrations following gas releases.

The acceptable range of air temperature entries is -40°C to +50°C. EXTRAN converts the air temperature to degrees Kelvin as needed within the code.

Atmospheric Pressure

Atmospheric pressure is last of the environmental variables used in all scenarios. It is used in adjusting the air density, the boiling point of liquids, and the saturation vapor pressure. When requested, enter the pressure in millimeters of mercury. Pressures in inches, millibars, pounds per square inch, or other units must be converted to millimeters of mercury.

The range of acceptable pressures is from 600 to 800 mm Hg. This range is larger than would be reasonable for a single location because pressure is a function of elevation. The upper end of the range (720 to 800 mm Hg) is appropriate for sea-level locations. The lower end of the range (600 to 670 mm Hg) is appropriate for elevations of 5000 to 6000 ft above sea level.

Solar Radiation

If you have selected a liquid release scenario, you will be asked to enter information for three additional environmental variables. Solar radiation is the first of the three. During the day, solar radiation adds energy to the liquid pool, thereby increasing the evaporation rate.

Enter solar radiation in units of watts/square meter. The range of acceptable entries is 0 to 1200 w/m^2. Enter a zero or press RETURN if you want to simulate a night-time release.

During the day, solar radiation is a function of latitude, season, and time of day. The data in the following two tables provide some guidance for selection of reasonable solar radiation values. Table 2.1 presents noon-time maximum solar radiation fluxes for the first day of each month based on Weather Bureau Records (Langhaar 1953). The values are for clear days and are rarely exceeded. Table 2.2 presents the average solar radiation for the Hanford Meteorological Station (46°34'N, 119°36'W) for 9:00 a.m., 12:00 p.m., and 3:00 p.m. for each month. The Hanford Meteorological Station is in a semi-arid region in southcentral Washington State.

Cloud Cover

The second additional variable you will be asked to enter if you have selected a liquid release scenario is cloud cover. Cloud cover is the fraction of the sky covered by clouds. Enter it in tenths. For example, if clouds cover half sky, the cloud cover would be entered as 5. EXTRAN converts the 5 to 0.5. The range of acceptable cloud cover entries is 0 to 10.

Cloud cover is used in calculating the long-wave radiation from the atmosphere. The program makes an initial estimate of the long-wave radiation from the air temperature. It then refines this estimate using the cloud cover. Cloud cover increases the long-wave radiation flux to the liquid pool. This increase ranges from 0 to 60 w/m^2.

Table 2.2 presents the average cloud cover, in tenths, for each month at the Hanford Meteorology Station. These values provide some guidance in observed values. You will also want to run the model with zero and 10/10 cloud cover.

TABLE 2.1. Maximum Noon Solar Radiation Flux (watts/m^2)

Month	Latitude			
	30	35	40	45
January	760	650	540	430
February	850	760	660	550
March	960	900	800	730
April	1070	1010	950	880
May	1140	1100	1060	1010
June	1150	1140	1090	1060
July	1150	1140	1100	1070
August	1140	1100	1070	1030
September	1100	1060	990	950
October	990	930	850	770
November	850	770	680	580
December	760	660	550	440

TABLE 2.2. Average Solar Radiation (SR) in watts/m^2 and Cloud Cover (CC) in Tenths at the Hanford Meteorology Station. Based on data from Ramsdell (1978).

Month	9:00		12:00		3:00	
	SR	CC	SR	CC	SR	CC
January	136	8.2	255	8.1	79	8.0
February	230	7.6	385	7.4	181	7.4
March	410	6.7	560	6.9	307	6.9
April	574	6.3	671	6.7	396	6.9
May	674	5.5	741	6.0	489	6.3
June	714	4.8	804	5.2	537	5.4
July	738	2.4	858	2.8	589	3.1
August	645	3.0	768	3.3	491	3.6
September	531	4.2	646	4.3	342	4.4
October	367	5.8	463	6.0	172	6.4
November	198	7.6	267	7.6	65	7.5
December	113	8.3	201	8.3	40	8.1

Ground Temperature

Ground temperature is the last of the additional environmental variables needed for liquid releases. Along with the pool temperature, it controls the energy flux from the ground to the pool. Increasing the ground temperature increases evaporation.

During the day, the ground temperature in the shade is about the same as the air temperature, while the ground temperature in the open is several degrees warmer than the air temperature. At night the ground temperature is generally lower than the air temperature.

Enter the ground temperature in degrees Celsius. The range of acceptable ground temperatures is -40 to +60°C.

2.2.3 Effluent Characteristics

The last group of data entries defines the physical characteristics of the effluent. These characteristics include the boiling point, heat capacity, heat of vaporization, and specific gravity (or density) of the liquid phase, and the density of the vapor phase. The initial prompt for the group lists 19 chemicals and an option to enter data manually. If you select one of the 19 chemicals, the properties of the chemical will be drawn from the chemical library that accompanies the EXTRAN code. You won't need to enter any further data.

Table 2.3 shows the chemical library supplied with EXTRAN. The first line of the library gives the number of entries (20) and headings for the data columns. The library file is named CHEMICAL.DAT. It can be edited with any ASCII text editor.

TABLE 2.3. The EXTRAN Chemical Library

NCHEM = 20	MW	BP	CP	HV	SG	DC
CHLORINE	70.9	-34.1	0.946	288.	1.57	0.0792
AMMONIA	17.0	-33.4	4.60	1370.	0.674	0.139
SULFUR DIOXIDE	64.1	-10.0	1.32	397.	1.46	0.0906
ACETONE	58.1	56.2	2.21	536.	0.791	0.109
BENZENE	78.1	80.1	1.75	434.	0.880	0.077
BROMINE	159.8	58.7	0.448	188.	3.12	0.0755
CARBON DISULFIDE	76.1	46.5	1.01	352.	1.29	0.0892
CARBON TETRACHLORIDE	153.8	76.8	0.841	198.	1.60	0.0579
DIETHYLAMINE	73.1	55.5	2.36	403.	0.685	0.0884
ETHYL ACETATE	88.1	77.2	1.92	427.	0.895	0.0715
ETHYL BENZENE	106.2	136.2	1.71	398.	0.867	0.0658
ETHYL ETHER	74.1	36.4	2.29	351.	0.708	0.0666
HYDROGEN	2.02	-252.7	8.4	452.	0.07	0.0047
HYDROGEN CYANIDE	27.0	25.7	2.62	1034.	0.687	0.173
ISOPROPYL ALCOHOL	60.1	80.3	3.26	667.	0.785	0.0818
METHANOL	32.0	64.7	2.51	1100.	0.792	0.132
PHOSGENE	98.9	8.2	1.02	247.	1.42	0.095
PROPANE	44.1	-44.5	2.13	429.	0.585	0.0622
TOLUENE	92.1	110.	1.76	412.	0.866	0.076
OTHER						

BP = BOILING POINT, DEGREES C
CP = HEAT CAPACITY OF LIQUID, JOULES/(GRAM DEGREE K)
HV = HEAT OF VAPORIZATION OF LIQUID, JOULES/GRAM
MW = MOLECULAR WEIGHT, GRAMS/MOLE
SG = SPECIFIC GRAVITY OF LIQUID
DC = DIFFUSION COEFFICIENT, CM**2/SEC, AT 0 C OR BP IF BP < 0 C

If you select "OTHER," you will need to enter additional information. The information required depends on the type of release selected. For gaseous releases, the only other information required is the molecular weight. For liquid releases, you will be asked to enter the boiling point, heat capacity, heat of vaporization, specific gravity (density), and molecular diffusion coefficient in addition to the molecular weight. Descriptions of these data entries follow.

Name

The first entry is the chemical name. Entry of a name is optional. Names only identify the chemical in the code output. The name may consist of as many as 20 characters. The character count includes blank spaces. A string of 20 blanks will be used if you don't enter a name.

Molecular Weight

EXTRAN uses the molecular weight in thermodynamic computations for volatile liquids and in estimating the initial volume of gases. It must be greater than 1.0.

Boiling Point

Enter boiling point temperatures in degrees Celsius. Absolute boiling point temperatures are computed within the program as needed. The acceptable range of boiling point temperatures is -100 to +200°C. Boiling point temperatures are a function of pressure. The value entered should be the boiling point at 1 atmosphere pressure (760 mm Hg). It will be adjusted to the boiling point at the actual pressure within the program.

Heat Capacity

The heat capacity determines the fraction of a liquefied gas that flashes on release and the temperature change in pools. It must be in units of joules per gram per degree Celsius. The heat capacity must be greater than zero. Heat capacities given in calories per gram per degree Celsius may be converted to the required units by multiplying by 4.184. Within EXTRAN, the heat capacity is a constant, independent of temperature and pressure.

Heat of Vaporization

The heat of vaporization is the energy used in calculation of the evaporation of liquids. It must be in units of joules per gram and must have a value greater than zero. Within EXTRAN, the heat capacity is constant.

Specific Gravity

EXTRAN uses density in calculation of the volume and area of the pool formed by a release of liquids. You may enter either the density, in grams per milliliter, or specific gravity of the liquid when requested. EXTRAN assumes that the density of water is 1 g/ml. Thus, the density and specific gravity are numerically identical.

Enter the specific gravity of the liquid or the density in grams per milliliter. The range of acceptable entries is from 0.1 to 4.0.

Diffusion Coefficient

The rate of evaporation of the pool depends in part on diffusion of the vapor near the pool surface. This diffusion is characterized by a molecular diffusion coefficient. If the substance is a volatile liquid, a diffusion coefficient in units of square centimeters per second must be entered when requested. Molecular diffusion coefficients typically have values near 0.1 cm^2/sec. The range of acceptable values is 0.0 to 0.3.

If you enter a zero, EXTRAN will estimate the molecular diffusion coefficient from kinetic theory.

2.3 RERUN OPTION

When EXTRAN has completed computations for a scenario, it closes all the output files and requests directions. At this point you will see the prompt:

EXTRAN RUN COMPLETE. DO YOU WISH TO REVISE THE ENVIRONMENTAL CONDITIONS AND RUN ANOTHER CASE? Y OR N?

At this point, you have the option of changing the environmental conditions and running the code again. If you choose to rerun the model, you won't need to re-enter the data on the physical scenario or the effluent.

This option exists because you may want to run several sets of environmental conditions to determine the range of potential concentrations at the intake. There is no single set of worst case environmental conditions for EXTRAN. In some scenarios, high concentrations will be associated with low wind speeds, and in others, high concentrations will be associated with high wind speeds. For gases released outside a wake, high concentrations will be associated with low wind speeds. However, if a release occurs in a wake or involves a volatile liquid, the highest concentrations are likely to occur with high wind speeds. In other conditions, you will need to search for the environmental conditions that give the highest concentrations.

If you respond with an N when you are given the opportunity to modify the environmental data, execution of EXTRAN will end, and you should see the prompt from your computer operating system or shell program. If you respond with a Y, EXTRAN will restore data to their initial values. When this is complete, you will see a list of the environmental data. The list will be the same as the list you saw during data entry, and the data will be the data for the case just run. At this time you may change any or all the environmental data by following the instructions on the screen.

When you complete revision of the environmental data, the code renames the output files and opens the new files. The names of the new files will be the same as the names as the previous files except they will have a different extension. For example, if the first set of files had the names EXPRhhmm.001 and EXCRhhmm.001, the new file names will be EXPRhhmm.002 and EXCRhhmm.002. The hhmm portion of the name remain the same for both sets of environmental conditions in the session.

2.4 OUTPUT INTERPRETATION

EXTRAN is based on a generally conservative set of assumptions that tend to maximize estimates of concentrations at the control room air intake. The most important of these assumptions is that the wind is blowing directly from the tank to the control room air intake when a release occurs. Even if the tank or release point were in the worst possible position with respect to the intake, the likelihood that the wind would be blowing directly toward the intake at the time of an accidental release is only a few percent. The likelihood that wind direction would remain constant for more than a few seconds is also small. The conservative nature of the model is enhanced by

not including several physical processes that would tend to reduce concentrations. Processes not in EXTRAN include plume rise, deposition on surfaces, washout, and chemical transformations. In addition, the model neglects radioactive decay.

Concentrations estimated with EXTRAN should be considered typical concentrations that might be observed given the conditions you specify in the model input. However, it is highly unlikely that concentrations observed following an actual release would be the same as the concentration predicted by the model. There are many reasons not to expect the predicted and observed concentrations to be identical. These reasons include the random nature of the atmospheric dispersion process and the very simple nature of the mathematical representations of the process in the EXTRAN code. This section discusses interpretation of the EXTRAN output in light of the uncertainty of the concentration estimates. It also discusses likely errors associated with the use of EXTRAN to predict concentrations downwind of large releases of liquefied gases.

2.4.1 Model Uncertainty

The method of estimating diffusion coefficients was developed from data obtained in experiments with releases that were generally 15 min to 1 hr in duration. As a result, the coefficients are more appropriate for estimating 15- to 60-min average concentrations than they are for estimating 2-min average concentrations. Experimental data (e.g., Hinds 1969; Ramsdell and Hinds 1971) show the variation of short-term concentrations within a longer averaging period. The uncertainty in average concentrations estimated by the wake diffusion model in EXTRAN has also been determined experimentally (Ramsdell 1990a). Assuming that these sources of uncertainty are independent, it is possible to make a rough estimate of the probability that a 2-min average concentration at an air intake following an accidental release would exceed the concentration predicted by EXTRAN.

Figure 2.7 shows an estimate of the percentage of the time 2-min average concentrations at an air intake might exceed EXTRAN concentration predictions. The probability that the ratio between an average concentration during a specific 2-min period and the predicted concentration exceeds 1 (i.e., that the 2-min average is greater the predicted concentration) is less than 40%. You will note that the probability decreases as the concentration ratio increases. The probability that the ratio will exceed 2 is about 25%, and the probability it will exceed 5 is less than 15%. The following examples illustrate two uses of Figure 2.7.

Suppose that EXTRAN predicts a chlorine concentration of 15 ppm at a control room air intake. The figure shows that there is about a 50% chance that the maximum observed 2-min concentration at the intake following the release would be less than 7.5 ppm. It also shows that there is about a 25% chance that the concentration might exceed 30 ppm. To get these numbers, select the probability of exceeding the predicted concentration, and then use that probability to find the approximate ratio between actual and predicted concentrations that corresponds to the probability. Finally, you multiply the predicted concentration by the ratio.

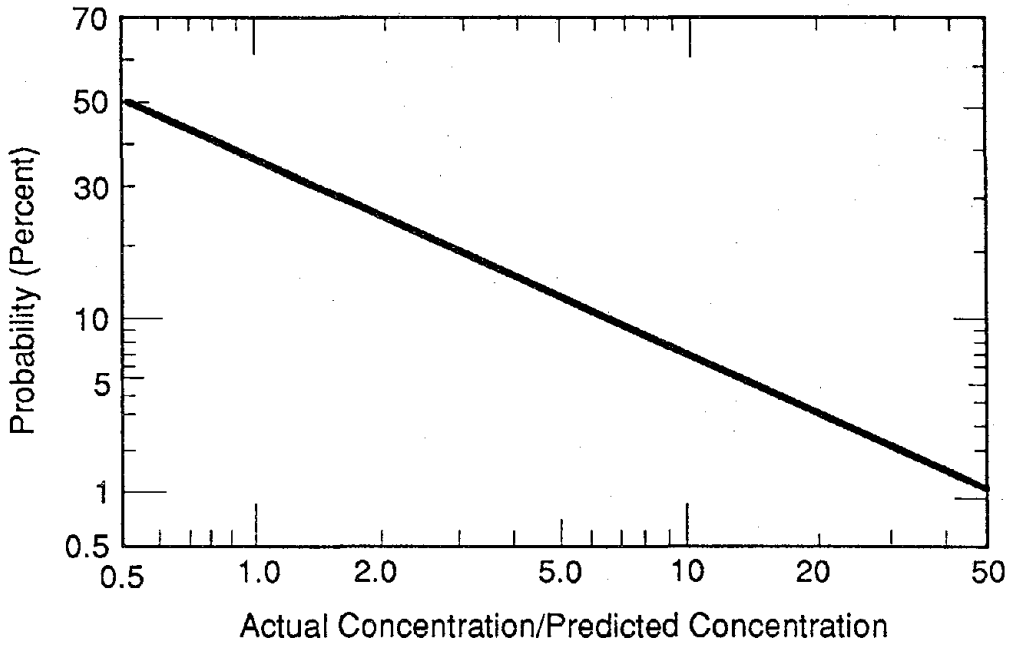

FIGURE 2.7. Estimated Probability of Exceeding the Predicted 2-min Concentration

You can also use the figure to estimate the probability of exceeding a concentration threshold given a predicted concentration. Suppose that the short-term (2 min) exposure limit for ammonia is 300 ppm and that EXTRAN predicts a 2-min average concentration of 10 ppm. The ratio between the threshold and the predicted concentration is 30. Entering Figure 2.7 with this ratio as if it were an actual/predicted concentration ratio, you will find that the ratio has a probability of about 2%. Thus, given the scenario that you modeled, the chance of exceeding the 2-min ammonia exposure limit at the air intake would be about 2%.

2.4.2 Dense Gases

EXTRAN treats the movement and diffusion of airborne material as if the material is a trace gas. The gas temperature and density do not affect dispersion in the EXTRAN code. However, temperature and density do affect atmospheric dispersion of large quantities of liquefied gases. You should consider the probable effects of the temperature and density on dispersion if you are attempting to model the release of a large quantity of a liquefied gas.

Dickerson and Ermak (1988) list three ways in which dispersion of dense gases differs from dispersion of trace gases. Dense gas clouds are lower and wider than trace gas clouds. Dense gas clouds move slower than trace gas clouds, and gravity affects dense gas cloud movement in addition to the wind.

The first of these differences may result in a significant difference between predicted and actual concentrations at an air intake. Assume a ground-level release and an elevated intake. If the distance between the

release point and intake is small (perhaps up to a few hundred meters), actual concentrations at the intake are likely to be less than the predicted concentration. At larger distances, the actual concentrations may be consistently larger than the predicted concentrations.

The second difference affects the time required for the cloud to move from the release point to the intake. EXTRAN is likely to underestimate the travel time. Similarly, it will underestimate the time required for the cloud to pass the intake. Further, if the intake structure is in a depression, the cloud may stagnate near the intake.

Finally, the third difference in dispersion is important if the terrain is uneven. During low wind speeds, the movement of the cloud will have a downhill component. If the intake structure is in a depression, and the tank is on higher ground, the probability of a cloud reaching the intake following an accident may be significantly higher than would be predicted from the wind direction frequency. On the other hand, if the intake structure is on high terrain, the probability of a cloud reaching the intake may be significantly lower than estimated from the wind direction.

2.5 EXAMPLES

This report includes five examples to allow you to practice running the EXTRAN code and check code operation. They cover each of release types for toxic chemicals and the use of the OTHER option in the chemical list for modeling dispersion of radionuclides. They also show the use of the rerun option.

Appendix A contains the input and output for each example. If your results aren't the same as those given in the appendix, you should look for the cause of the difference by comparing the data in your input summary with the data in the input summaries in the appendix.

As an exercise, you might vary the input for one variable at a time and observe the sensitivity of the model concentration prediction to the variation in the input. Compare the model response to changes in wind speed for gas and liquid releases in the absence of a building wake. Repeat the comparison for releases in a building wake.

3.0 EXTRAN TECHNICAL BASIS

The EXTRAN computer code is tool to estimate short-term concentrations at a single location downwind of a release of toxic or radioactive material. The code estimates the amount of material entering the atmosphere and then moves the material downwind as it diffuses. This part of the report describes the technical basis for the EXTRAN code. It includes descriptions of the equations and continuous mass and energy balances used to estimate the source term for the dispersion model. It also includes a description of the dispersion model.

3.1 SOURCE TERM MODELS

EXTRAN deals with releases of two types of material, gases and liquids. You specify the source term for releases of material stored as a gas when you enter the mass of material in the tank and the leak rate. The mass balance associated with gaseous releases is simple. All of the material leaving the tank enters the atmosphere immediately. EXTRAN does not maintain an energy balance for gaseous releases.

When you model the release of material stored as a liquid, specifying the amount of liquid in the tank and the release rate doesn't completely determine the source term for atmospheric dispersion. EXTRAN computes the atmospheric source term from information you supply about the material, its storage conditions, and the environment. The material released from the tank must be partitioned between the atmosphere and the pool, material in the pool evaporates in response to the vapor pressure of the liquid, and the temperature of the pool is adjusted to maintain an energy balance between the pool and the environment.

If you model a catastrophic failure of a tank containing a gas, all of the gas enters the atmosphere at the time of the failure. Otherwise, you must supply a leak rate as part of the scenario. This leak rate is used to determine the amount of the gas in each puff and the amount remaining in the tank. This information is shown in the first line of each entry in the mass balance file. The remaining lines of the entries in the mass balance file are zero because there is no pool.

EXTRAN assumes that when a tank containing liquefied gases or volatile liquids fails, the material is distributed between the tank, a pool in the vicinity of the tank, and in a plume in the atmosphere downwind of the tank. A reasonable estimate of the concentrations at a receptor downwind of the tank can only be made if material can be realistically distributed between these locations as a function of time following the release. Mass must not be gained or lost between the release point and the receptor. The next section describes the mass balance for materials that are stored as liquids. Before any of the material can enter a plume it must change to the vapor phase. This change involves energy transfers and transformations. Thus the mass balance depends on a scenario dependent energy budget. The energy budget will be described in Section 3.3.

3.2 MASS BALANCE FOR LIQUEFIED GASES AND VOLATILE LIQUIDS

As an initial condition, a known mass of the material is assumed to be in a tank. When the tank fails, the material is released to the environment. If the material is a liquefied gas, part of the material leaving the tank flashes at the time of release; the remaining material forms a pool at the base of the tank. The amount of material that flashes can be determined from thermodynamic considerations. If the material is a volatile liquid, all of the material leaving the tank is assumed to gather in a pool at the base of the tank. Material entering the pool is assumed to enter the plume by evaporation. The equations used in EXTRAN to compute the mass of a liquefied gas that flashes and the evaporation of the pool were presented by Wing (1979).

3.2.1 Flashing Mass

The mass of liquefied gas that flashes when the material is released can be computed from the mass of the liquid released, its heat capacity and heat of vaporization, and the difference between the boiling point of the liquid and the temperature of the air. The relationship is

$$m_f = m_r c_p (T_a - T_b)/h_v \qquad (1)$$

where m_f = mass that flashes (kg)

m_r = mass released (kg)

c_p = heat capacity (j/g°C)

T_a = air temperature (°C)

T_b = boiling point (°C)

h_v = heat of vaporization (j/g).

Any liquid that does not flash forms a pool at the base of the tank.

3.2.2 Liquid Pool

The liquid pool is assumed to be cylindrical with a volume that is determined by the mass and density of the liquid in the pool. The area of the pool is estimated in two ways. It is estimated as a function of time using

$$A = \pi\{r_o^2 + [2t(gV/\pi)^{1/2}]\} \qquad (2)$$

where A = area of the pool at time t (m^2)

r_o = an initial radius estimate (m)

t = time since tank failure (sec)

g = gravitational constant (m^2/s)

V = pool volume (m^3).

The relationship for pool area given by Wing has an additional term that involves the densities of the liquid and air. The value of that term does not deviate significantly from 1. Consequently, the term has been dropped from Equation (2). Pool area is also estimated from the pool volume (mass divided by density) assuming a thickness of 0.01 m. These two area estimates are compared with the maximum pool area specified in the input; the smallest value is used as the pool area.

An initial estimate of the pool temperature is made by averaging the temperature of the material being released (boiling point if it flashes) and the temperature of the material already in pool. This temperature is used to estimate the saturation vapor pressure of the material for use in evaporation calculations.

3.2.3 Evaporation

Evaporation of the pool is driven by the difference between the saturation vapor pressure of the liquid in the pool and the vapor pressure in the atmosphere. For the materials of concern, the vapor pressure in the atmosphere may be assumed to be zero. Therefore, the relationship used to compute the mass of volatile liquid evaporating in an interval Δt is

$$m_v = (h_d \, A \, \Delta t \, P_s)/(R_g T_p) \qquad (3)$$

where h_d = mass transfer coefficient (m/sec)

P_s = saturation vapor pressure (atm)

R_g = specific gas constant ($atm\text{-}m^3kg^{-1}K^{-1}$)

T_p = pool temperature ($^\circ K$).

The mass transfer coefficient is a function of the radius of the pool, a diffusion coefficient, and the dimensionless Reynolds and Schmidt numbers associated with the air flow across the pool. In EXTRAN, the mass transfer coefficient is computed using the relationship given by Wing for turbulent flow. That relationship is

$$h_d = 0.037 \, (D/2r) \, Re^{4/5} \, Sc^{1/3} \qquad (4)$$

where D = molecular diffusion coefficient (m^2/s)

r = pool radius (m)

Re = Reynolds Number

Sc = Schmidt Number.

Molecular diffusion coefficients for many substances have been determined experimentally. They are listed in reference works (e.g., Reid et al. 1987). However, there are many substances for which molecular diffusion coefficients are not readily available. If you don't enter a diffusion coefficient as part of the effluent characteristics, EXTRAN will compute one from kinetic theory (Bird et al. 1960). The equation used in computing the molecular diffusion coefficient is

$$D = \frac{0.001853[T_r^3(1/mw_a + 1/mw_g)]^{1/2}}{P\,\sigma_{ag}\,\Omega_{ag}} \tag{5}$$

where T_r = temperature (°K)

mw_a, mw_g = molecular weights of air and the material in the pool

P = pressure (atmospheres)

σ_{ag} = effective collision diameter for molecules of air and the gas (Å)

Ω_{ag} = dimensionless function of temperature and intermolecular potential field.

An effective collision diameter for the combination of air and a second gas can be estimated by averaging the collision diameters for air and the gas. The collision diameter for air is 3.617 Å. For a liquid, the collision diameter may be estimated from the normal boiling point. EXTRAN estimates the collision diameter for the material in the tank from

$$\sigma_g = 1.166\ V^{1/3} \tag{6}$$

where V is the molecular volume in cubic centimeters per gram-mole (Bird et al. 1960). The molecular volume is computed from the molecular weight and the liquid density.

The dimensionless function Ω_{ag} in Equation (5) is estimated using interpolation equations derived from data presented by Bird et al. (1960, Table B-2). Data in the table indicate that Ω_{ag} can be estimated adequately as

$$\Omega_{ag} = \alpha(kT/\epsilon_{ag})^{\beta} \tag{7}$$

where α,β = constants defined for three ranges of kT/ϵ

k = Boltzmann's constant

T = temperature (°K)

ϵ_{ag} = a characteristic energy of interaction between molecules of air and the gas.

Finally, ϵ_{ag} is the geometric mean of the characteristic energies for air and the gas. The ratio ϵ_g/k can be estimated from

$$\epsilon_g/k = 1.15\ T_b \tag{8}$$

where T_b is the boiling point of the liquid (°K). For air ϵ/k is approximately 97.0.

The boiling point of liquids depends on atmospheric pressure. Tabled boiling points are generally for a pressure of one atmosphere (760 mm Hg). EXTRAN adjusts the boiling point to account for departures from the standard pressure using an integrated form of the Clausius-Clapeyron equation. The adjusted boiling point is given by

$$T_b^* = T_b/[1 + R_g T_b\ \ln(760/P)/h_v] \tag{9}$$

where T_b^* = adjusted boiling point (°K)

T_b = boiling point at 760 mm Hg (°K)

P = atmospheric pressure (mm Hg).

Similarly, the saturation vapor pressure in Equation (3) is computed for pool temperature from the boiling point and atmospheric pressure using another form of the integrated Clausius-Clapeyron equation. The equation for saturation vapor pressure is

$$P_v = P\ \exp[h_v(1.0 - T_b^*/T_p)/R_g T_b^*] \tag{10}$$

where all terms are as previously defined.

3.3 ENERGY BUDGET

Several terms in the mass balance just discussed are functions of energy transfer in the environment. This section discusses the components of energy budget of the pool. These components include solar and long-wave radiation, sensible and latent heat exchange with the atmosphere, and heat transfer between the pool and the ground.

3.3.1 Solar Radiation

The solar radiation balance associated with the liquid pool has two terms, incoming solar radiation and reflection from the surface. The net solar radiation, R_{sn}, is estimated as

$$R_{sn} = (1 - a)R_s \tag{11}$$

where R_s is the incoming solar radiation, and a is the albedo of the pool. An estimate of the incoming solar radiation is required as part of the model

input. Tables 2.1 and 2.2 provide some guidance on the values of solar radiation observed in the latitude band occupied by the contiguous United States.

The albedo of a pool of any of the materials of concern is unknown. Consequently, a default albedo has been assumed. The default albedo is 0.1. The assumption of this value is based on albedos reported for sea surfaces (Rosenberg et al. 1983, p. 44). This justification is relatively weak, but the net energy budget is not particularly sensitive to the value assumed. The assumption of a relatively low albedo should be conservative. That is, a low albedo is associated with high absorption of solar radiation, which will tend to maximize evaporation.

3.3.2 Long-Wave Radiation

The long-wave radiation balance of the pool is more complicated than the short-wave balance. The long-wave radiation that the pool receives from the atmosphere is a function of the temperature and humidity near the ground and the amount and type of clouds. This is countered by long-wave radiation from the pool, which is a function of the pool temperature.

A number of empirical expressions have been developed to estimate the long-wave radiation from the atmosphere. Swinbank (1963) proposed an equation relating the long-wave radiation only to the air temperature based on an analysis of data collected at night. This relationship is reasonable because the humidity at night is also a function of temperature. However, the relationship has been found to systematically overestimate the long-wave radiation during the day. Paltridge (1970) suggests reducing day-time atmospheric long-wave radiation estimates made using Swinbank's relationship by 30 w/m^2. The same correction to Swinbank's relationship is suggested by Idso (1972). A further correction to Swinbank's relationship to account for the effects of clouds is discussed by Paltridge and Platt (1976). In its simplest form, this correction increases the atmospheric long-wave radiation by 6 w/m^2 for each tenth of the sky covered by clouds. Combining these relationships, the long-wave radiation flux to the pool from the atmosphere is estimated using

$$R_{la} = 5.31 \times 10^{-13} T_a^6 + 60\ cc \qquad (12)$$

where R_{la} is the long-wave radiation flux from the atmosphere (w/m^2), and CC is the cloud cover in tenths. If the solar radiation is greater than 100 w/m^2, the long-wave radiation flux estimated by Equation (12) is reduced by 30 w/m^2.

The long-wave radiation from the pool is a function only of the pool temperature. It is given by

$$R_{lp} = \sigma T_p^4 \qquad (13)$$

where σ is the Stephan-Boltzmann constant, which has a value of 5.67×10^{-8} $w/(m^2\ {}^oK^{-4})$.

The net long-wave radiation flux to the pool, R_{ln}, is then

$$R_{ln} = R_{la} - R_{lp}. \tag{14}$$

Estimating this flux requires both the air and pool temperatures. If it is estimated using only the air temperature, the result will be an under estimate of the true flux because the pool temperature will be lower than the air temperature. This will tend to decrease the rate of evaporation and is therefore not conservative.

3.3.3 Air Convection

Radiation is only one of the energy transfer mechanisms in the EXTRAN model. The model also includes sensible heat transfer between the pond and the atmosphere due to convection, heat transfer between the pond and the ground due to conduction, and the transfer of latent heat from the pond to the atmosphere due to evaporation.

The heat flux between the pool and the atmosphere is computed as the product of a heat transfer coefficient and the temperature difference between the pool and the atmosphere. The equation for this flux is

$$Q_a = h_c(T_a - T_p) \tag{15}$$

where Q_a is the heat flux due to convection and h_c is the heat transfer coefficient. The heat transfer coefficient is a function of the thermal conductivity of the air, the air density and viscosity, the heat capacity of the air, a characteristic length, and the wind speed. Assuming that all of these factors are constant except the wind speed, the heat transfer coefficient is computed by

$$h_c = 6.69U^{0.6} \tag{16}$$

where U is the wind speed. This relationship is based on an extrapolation by Wing (1979) to a wind speed of 1 m/sec of values published by Bolz and Tuve (1973).

3.3.4 Ground Conduction

The heat flux between the ground and the pool is estimated in a manner that is similar to the method used to estimate the heat flux from the air. The flux is proportional to a thermal conductivity and the temperature difference between the ground and the pool. However, unlike the air that passes over the pool, the ground beneath the pool cools as a function of time, which reduces the flux. The equation used to estimate the heat flux from the ground is

$$Q_g = k(T_g - T_p)t^{-1/2} \tag{17}$$

where Q_g is the flux, and k is a constant that incorporates the thermal conductivity, heat capacity, and density of the ground. The derivation of Equation (17) and k is discussed by Bird et al. (1960). Following Wing,

when values of the thermal conductivity, heat capacity, and density that are typical of the earth's surface are used, the value of k is approximately 824 j-sec$^{1/2}$/(°K). However, k is a strong function of surface type. When the characteristics of specific surface types reported by Oke (1978) are used to compute k, the values obtained range from 190 for dry peat to 2560 for wet sandy soil. The value for dense concrete is 1780, and bricks and stone have values near 1000. As a result of this diversity, a mid-range value of 1000 has been assumed for k rather than 824.

3.3.5 Latent Heat Transfer

As liquid evaporates, it carries energy from the pool to the atmosphere. The energy lost is equal to the product of the heat of vaporization and the mass evaporated. Thus,

$$Q_h = h_v m_v \tag{18}$$

where Q_h is the latent heat lost. When the pool consists of boiling liquefied gas, the evaporation is determined by the energy entering the pool. As a result, the latent heat loss exactly balances the net energy gained from solar and long-wave radiation, convection, and conduction. This balance may be written as

$$Q_h = (R_{sn} + R_{ln} + Q_a + Q_g)A\Delta t. \tag{19}$$

We have assumed that evaporation is determined by vapor pressure. Consequently, the latent heat loss is not necessarily equal to the input of energy by other mechanisms. If the latent heat loss does not balance the energy gained from other mechanisms, then the temperature of the pool must change as a result of the imbalance. The energy stored or lost by the pool is given by

$$m_p c_p \Delta T_p = (R_{sn} + R_{ln} + Q_a + Q_g)A\Delta t - Q_h. \tag{20}$$

Solving for the temperature change yields

$$\Delta T_p = [(R_{sn} + R_{ln} + Q_a + Q_g)A\Delta t - Q_h]/m_p c_p. \tag{21}$$

Common experience indicates that evaporation leads to cooling. Therefore ΔT_p may be expected to be negative. However, when the pool is extremely cool, the energy balance may be positive.

In EXTRAN, the initial pool temperature is assumed to be equal to the boiling point or temperature of the liquid in the tank, whichever is lower. The temperature of the pool is adjusted as a function of time to maintain the energy balance. Equation (21) is used to estimate the change in temperature due to evaporation. If the temperature change is negative, the pool temperature is adjusted accordingly. EXTRAN does not permit the pool to freeze.

If the temperature change is positive, EXTRAN makes an initial estimate of the new pool temperature and then compares the estimate with the boiling point of the liquid. Should the estimated temperature exceed the boiling point, the pool temperature is set at the boiling point. The energy not used

in raising the temperature of the pool to the boiling point is treated as latent heat of evaporation, and mass associated with evaporation is increased.

If the accident scenario involves a leak, the temperature of the pool is also adjusted to account for the difference in temperature between the tank and the pool. In making this adjustment, it is assumed that the liquid entering the pool mixes instantaneously with the liquid already in the pool.

3.4 TRANSPORT AND DIFFUSION

EXTRAN includes a Gaussian puff dispersion model. This approach was selected because puff models permit more realistic treatment of temporal variations in release terms and concentrations. It is consistent with the Gaussian plume models used by the NRC for other licensing applications and the puff models used for emergency response applications.

3.4.1 Puff Model

The derivation of the Gaussian plume model starts with a specific solution to the one-dimensional diffusion equation. A three-dimensional puff diffusion model is then produced by superposition of solutions to the one-dimensional equation. If it is assumed that diffusion proceeds independently in the longitudinal, lateral, and vertical directions, and that the center of the puff is at position x_o, y_o, z_o; then, in the absence of boundaries, the concentration at position x,y,z is given by

$$C(x,y,z) = \frac{Q}{(2\pi)^{3/2}\sigma_x\sigma_y\sigma_z} \exp\left[-\frac{1}{2}\left(\frac{x - x_o}{\sigma_x}\right)^2\right] \exp\left[-\frac{1}{2}\left(\frac{y - y_o}{\sigma_y}\right)^2\right] \exp\left[-\frac{1}{2}\left(\frac{z - z_o}{\sigma_z}\right)^2\right] \quad (22)$$

where $C(x,y,z)$ = the concentration at x,y,z

Q = the mass of material in the puff

$\sigma_x,\sigma_y,\sigma_z$ = diffusion coefficients in the longitudinal, lateral, and vertical directions.

The diffusion coefficients are characteristic dimensions of the puff. They are functions of the distance (or time) from the release point, the atmospheric stability, and the surface roughness.

Next, a Cartesian coordinate system is defined that has its origin at the ground directly below the release point with the x-axis parallel to the wind vector, the y-axis directed cross wind, and the z-axis vertical. With this definition, the center of the puff can now be allowed to move with the wind. At any moment t following the release, the coordinates of the center of the puff are $x_o = Ut$, $y_o = 0$, $z_o = h$ where U is the wind speed and h is the height of release. This results in

$$C(x,y,z) = \frac{Q}{(2\pi)^{3/2}\sigma_x\sigma_y\sigma_z} \exp\left[-\frac{1}{2}\left(\frac{x - Ut}{\sigma_x}\right)^2\right] \exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_y}\right)^2\right] \exp\left[-\frac{1}{2}\left(\frac{z - h}{\sigma_z}\right)^2\right] \quad (23)$$

The final step in derivation of the plume model is integration of Equation (23) from t = 0 to t = ∞. This step makes the plume model a steady-state model. The time delay between the source and the receptor does not appear explicitly in the model, and temporal variations in the source can only be modeled by assuming a sequence of steady-state releases.

The puff model alternative chosen for EXTRAN stops at Equation (23). Using the puff model, a plume is approximated by releasing a sequence of puffs at small time intervals. The concentration at a point in the plume is then calculated by summation of the concentrations at the point resulting from all puffs. In essence, the integration that leads to the plume model is replaced by

$$C(x,y,z) = \sum_i C_i \quad (24)$$

where $C(x,y,z)$ is the concentration at x,y,z, and the C_i are the contributions to the total concentration of the individual puffs given by Equation (23). It is common to assume that σ_x and σ_y are equal and to substitute σ_y for σ_x. Equations (23) and (24) retain the ability to model the temporal variation in concentrations at an air intake realistically because the concentration will not increase until a puff approaches receptor, and the values of Q may be varied as function of time.

The accuracy of the puff approximation can be checked by modeling a steady-state release. It is a function of the distance between puffs. The approximation can be made as accurate as desired by reducing this distance. Ramsdell et al. (1983) show that if the distance between adjacent puffs is less than σ_y, concentrations estimated by the puff model are within 1 or 2% of those estimated by a plume model. Puff release rates in EXTRAN are adjusted to maintain this accuracy.

Equation (23) assumes that the diffusion takes place without the interference of boundaries. That assumption in not tenable for releases at or near ground level. It is common to assume that the ground acts as a reflecting surface. This assumption is incorporated into puff and plume models by assuming an imaginary second source of equal strength located at or below ground level. Concentrations are then computed by adding the contributions from the real and imaginary sources. Mathematically this is accomplished by replacing the term

$$\exp\left[-\frac{1}{2}\left(\frac{z - h}{\sigma_z}\right)^2\right]$$

in Equation (23) with

$$\left\{\exp\left[-\frac{1}{2}\left(\frac{z - h}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z + h}{\sigma_z}\right)^2\right]\right\}$$

3.4.2 Source Term

Concentrations in the puffs are directly proportional to the mass that is included in the puff. In the EXTRAN code, puffs are released at regular intervals of length Δt where Δt is determined by the distance between the release point and the air intake, the wind speed, and the atmospheric stability. The mass in a puff released at time t is the mass entering the atmosphere in the period between t and $t + \Delta t$.

If the toxic substance is a liquefied gas and both flashing and evaporation are occurring, two puffs will be released simultaneously. One of these puffs will have the mass of the liquid that has flashed [Equation (1)], and the other will have the mass that has evaporated [Equation (3)]. Otherwise only one puff will be released, and the mass in the puff will be determined using Equation (3).

3.4.3 Diffusion Coefficients

Equation (23) shows that the decrease in concentrations in puffs as they move downwind is due only to increases in the magnitudes of the diffusion coefficients. Relationships describing the increase in these coefficients in flat terrain under normal atmospheric conditions are readily available in the literature. The coefficients increase with increasing distance and generally decrease as the atmosphere becomes more stable.

These standard relationships do not adequately describe the growth of diffusion coefficients in the wakes of structures. The effect of wakes is to increase the rate of diffusion, but the effect is limited to the vicinity of the structure. As result, composite diffusion coefficients that include both normal diffusion and wake effects are used in EXTRAN. These coefficients are computed using

$$\sigma_c^2 = (\sigma_n^2 + \sigma_w^2)^{1/2} \tag{25}$$

where σ_c = the composite diffusion coefficient

σ_n = a normal diffusion coefficient

σ_w = a wake diffusion coefficient.

Normal diffusion coefficients are computed with the Eimutis and Konicek (1972) relationships used in the NRC PAVAN (Bander 1982) and XOQDOQ (Sagendorf et al. 1982) codes. In these relationships the diffusion coefficients are functions of distance and atmospheric stability. The wake diffusion coefficients are computed using equations derived by Ramsdell following an analysis

of data from building-wake diffusion experiments (Ramsdell 1988). Ramsdell (1990a,b) present derivations of the equations for the wake diffusion coefficients.

The puff diffusion equation was derived for point-source releases. The point source equations are reasonable as long as the distance between the release point and the receptor is large. In the current application, the point source assumption may not be appropriate. Consequently, an adjustment is made to the diffusion coefficients to account for the size of the source. The diffusion coefficients are given initial values that result in concentrations at the center of the puff that are no greater than the concentration the pure vapor would have been at the atmospheric conditions. These dimensions are related to the density of the vapor and the area of the pool. If a wake is a factor, the adjustment is made to the wake diffusion coefficients. Otherwise, the adjustment is made to the normal coefficients.

3.4.4 Transport

The transport of material is completely defined during model input. Puffs are assumed to move with the wind directly from the release point to the air intake. The time required for material to arrive at the intake is determined by the wind speed and the growth of the puffs. It is somewhat less than the time estimated by x/U where x is the distance to the intake, and U is the wind speed.

4.0 THE EXTRAN COMPUTER CODE

EXTRAN is an interactive computer code to model the EXternal TRANsport of toxic materials from a release point to a control room air intake. It is written in the FORTRAN-77 programming language. The code generally conforms to the ANSI standard (ANSI 1978). However, it does use several common extensions, including long names and INCLUDE statements. These extensions should not cause any problems because the compiled code is transferable among personal computers using the MSDOS operating system. EXTRAN has been compiled to run on computers with and without math coprocessors. A math coprocessor will significantly reduce the time required to complete a simulation.

4.1 PROGRAM ELEMENTS

The EXTRAN code consists of a main program, 16 subroutines, and one function. The main program primarily serves as a control program and generates much of the output. The subroutines perform specific functions. For example, seven subroutines obtain input from the user, and three subroutines compute different types of diffusion coefficients, etc. The function UC converts user input from lower case to upper case letters. In general, the program passes data among the program elements using NAMED COMMON BLOCKS. There are seven named common blocks. The common blocks are in the INCLUDE files that are incorporated in the appropriate program elements when the code is compiled.

The calling sequence of the EXTRAN program elements is shown in Table 4.1. Program element names are in upper case letters, and indentation shows the nesting of the elements. Brief descriptions of the use of the elements follow the colons. The table also lists the INCLUDE files and their contents. Table 4.2 shows the interrelationships between the program elements and INCLUDE files.

The list of program elements in these tables does not include the function UC. The input subroutines that request letter responses to prompts use the function UC.

Subroutines GETDAT and GETTIM called by EXTRAN are systems subroutines supplied with the Microsoft® FORTRAN, Version 5.0. FORTRAN compilers from other vendors may have subroutines that perform the same function but have different names.

®Microsoft is a registered trademark of Microsoft Corporation.

TABLE 4.1. The EXTRAN Program Element Calling Sequence

```
EXTRAN    : Main program, control model, primary output of model results
     GETDAT    : Get the date of model execution from system clock
     GETTIM    : Get the time from the system clock
     INPUT     : Select scenario type, print input data
          INPUTO    : Establish output options
          INPUTS    : Establish source - receptor geometry, release rate,
                        storage tank temperature
          INPUTE    : Enter required environmental data
          INPUTEF   : Select effluent, enter effluent physical properties
                        if necessary
               DIFCOEF   : Compute molecular diffusion coefficient if
                             required by scenario and not included in
                             physical properties
     MODELPAR  : Determine the diffusion model parameters (puff release
                   interval, time step, etc.)
          NSIG      : Compute normal diffusion coefficient at intake
                        distance for use in determining model time step
          WSIG      : Compute wake diffusion coefficient at intake distance
                        for use in determining model time step
     PUFFMASS  : Compute mass in each puff from effluent characteristics,
                   release rate, pool conditions, and energy balance
     PUFFINIT  : Compute initial puff dimensions and virtual travel time
     RELPUFF   : Assign characteristics to puff at release time
     NSIG      : Compute normal diffusion coefficients for puff
     WSIG      : Compute building wake diffusion coefficients for puff
     CHIT      : Compute concentrations at intake from all nearby puffs
     CEPLOT    : Plot chronology of concentration, exposure and average
                   concentration on printer... ASCII codes used
     RINPUT    : Reset variables, open new files if model is to be rerun
                   with different environmental conditions
         RINPUTE   : Revise environmental conditions


INCLUDE BLOCKS

     EFFLUENT  : Physical properties of the effluent
     ENVIRON   : Environmental information
     OPTIONS   : Control variables for model output, output file names
     POOL      : Pool dimensions, status, energy flux data
     PRINT     : Data to be plotted
     PUFF      : Data on individual puffs
     SCENARIO  : Scenario description, geometry, model parameters
```

TABLE 4.2. Interrelationship Between EXTRAN Program Elements and INCLUDE Files

Program Element	INCLUDE FILE						
	EFFLUENT	ENVIRON	OPTIONS	POOL	PRINT	PUFF	SCENARIO
EXTRAN	XXX	XXX	XXX	XXX	XXX	XXX	XXX
CEPLOT			XXX		XXX		
CHIT		XXX			XXX	XXX	XXX
DIFCOEF	XXX	XXX					
INPUT	XXX	XXX	XXX				XXX
INPUTE		XXX					
INPUTEF	XXX	XXX					
INPUTO			XXX				XXX
INPUTS			XXX				XXX
MODELPAR		XXX	XXX				XXX
NSIG		XXX					
PUFFINIT	XXX	XXX		XXX		XXX	XXX
PUFFMASS	XXX	XXX		XXX			XXX
RELPUFF						XXX	
RINPUT	XXX	XXX	XXX				XXX
RINPUTE	XXX	XXX					
WSIG		XXX					XXX

4.2 PROGRAM ORGANIZATION

EXTRAN is a simple puff dispersion model with a built-in module for estimating source terms of liquid releases from limited data supplied by the user. The model begins with user input and computation of model control variables based on the input. EXTRAN then enters a loop. In the loop it estimates the mass entering the atmosphere in the current time step, releases new puffs, moves the new and old puffs, and computes the concentration at the control room air intake from the concentrations in the puffs. This loop repeats until the model predicts that the concentration at the air intake has been above background level for at least three minutes. When this computation loop is complete, EXTRAN generates the output requested by the user. It then asks if the user wishes to revise the environmental input and rerun the model. Figure 4.1 shows a general flow chart for EXTRAN.

With one exception, interactions between EXTRAN and the user take place in the seven input subroutines. The first five of these subroutines obtain the input for the initial model run. The other two input subroutines are used to reset the model and revise the environmental data when the model is to be rerun with different environmental data. The only user interaction that takes place outside the input subroutines is the selection of the rerun option; that selection is made in the main program. The input subroutines perform initial adjustments of the air density, the liquid vapor pressure, and the boiling point for the ambient atmospheric pressure and temperature. When all input and adjustments are complete, they write the input data to the primary output file.

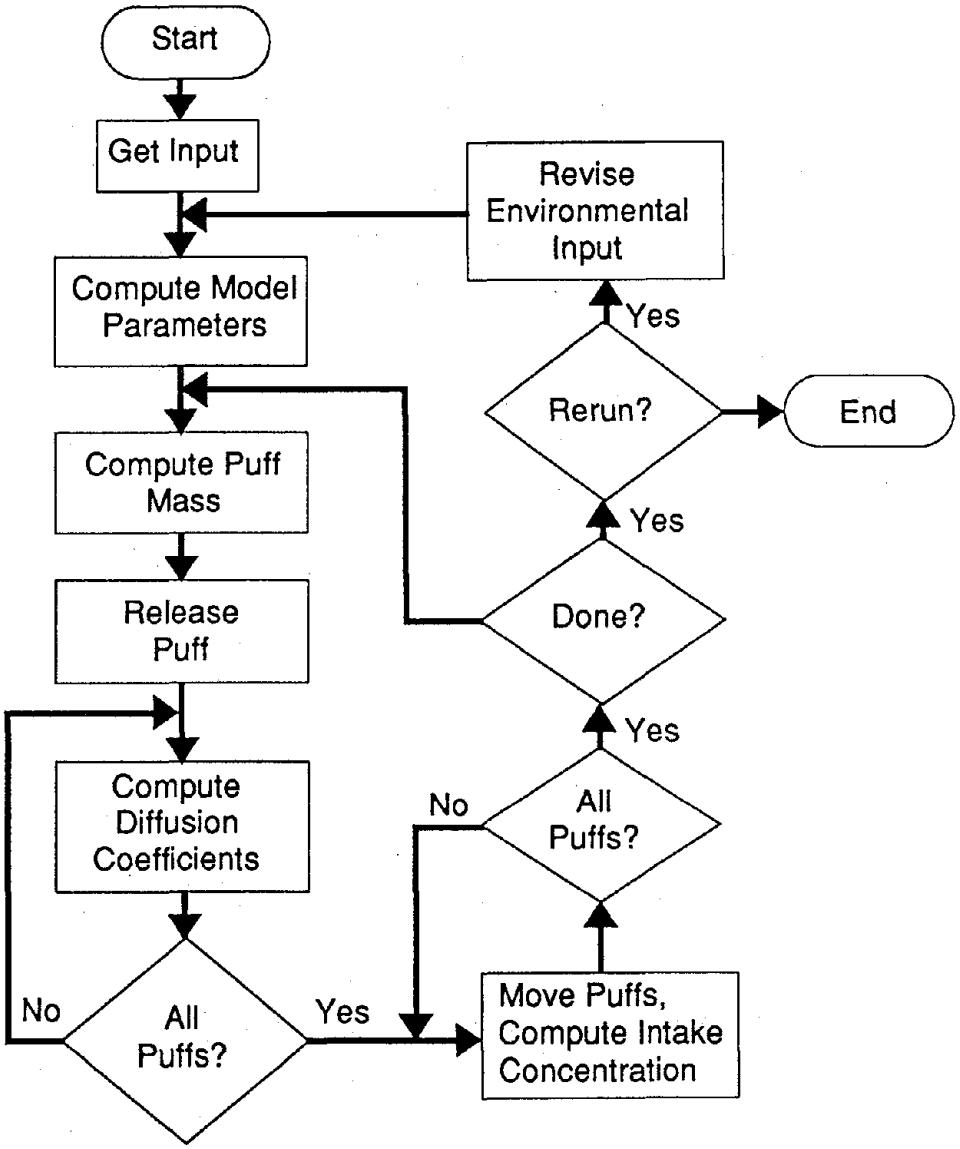

FIGURE 4.1 EXTRAN Flow Chart

When data entry is complete, EXTRAN calls subroutine MODELPAR. MODELPAR computes the intervals for releasing puffs and computing concentrations. The distance between the release point and the intake, the wind speed, and the atmospheric stability determine these intervals. The minimum interval for both puff releases and concentration computations is 1 sec. If the distance between release point and intake is less than 1000 m, the maximum interval between puffs is 10 sec. Otherwise it is 30 sec. The interval between concentrations is one half the puff release interval unless the puff release interval is 1 sec.

After setting the model parameters, EXTRAN enters the first of a pair of nested DO loops. The outer loop deals with puff releases, while the inner

loop deals with puff movement and concentration computations. Three subroutines deal specifically with releasing puffs. They are subroutines PUFFMASS, PUFFINIT, and RELPUFF. Similarly, there are three subroutines associated with the concentration computations. These subroutines are NSIG, WSIG, and CHIT.

The outer loop establishes and maintains the time of the simulation. It starts at t = 0 and increments in time steps that are equal to the interval between concentration computations. Puffs releases occur when the ratio between the time and the puff release interval has an integer value. At intermediate times the puff release portion of the program is skipped.

When puff releases occur, subroutine PUFFMASS determines the mass to be included in each puff. It is also the subroutine that makes the mass balance and energy budget computations. Subroutine PUFFINIT determines the initial dimensions for each puff. The mass to be included in the puff and the dimensions of the liquid pool determine the initial puff dimensions. Finally, subroutine RELPUFF creates the puffs by incrementing the puff counter and assigning the initial characteristics to each puff. Prior to entering the inner loop, EXTRAN writes a record to the mass and energy balance file, if the user has requested creation of the file.

The inner DO loop is the loop that makes the transport and diffusion coefficient computations. The loop index is puff number. Starting with the first puff released, the position of each puff is updated for movement in the current interval, and new diffusion coefficients are computed. Subroutine NSIG computes the normal diffusion coefficients for the puffs from the coefficient values in the previous time step and the distance moved in the current step. Wake diffusion coefficients are computed in subroutine WSIG from the current puff age. EXTRAN leaves the inner loop when all puffs have been moved and their diffusion coefficients have been updated.

At this time, EXTRAN calls subroutine CHIT to compute the concentration at the air intake. It only computes the contributions to the concentration from those puffs that are sufficiently close to the intake to make significant contributions to the total concentration. A puff is sufficiently close if its center is within 4.24 σ_y of the intake. If the puff center is farther from the intake, the concentration at the intake will be less than 1/10000 of the concentration at the center of the puff.

Computation of the intake concentration completes the outer loop. If sufficient time has elapsed since the arrival of the plume at the intake EXTRAN will leave the loop and output the results. Otherwise, it will return to the beginning of the loop and continue with the next time step.

After EXTRAN leaves the transport and diffusion computation loops, the main program will complete printing the summary page of the primary output. If a plot has been requested, EXTRAN calls subroutine CEPLOT, which will plot the chronology of the instantaneous and average concentrations and the exposure at the intake. Finally, the program will create a chronology file

and print the numerical chronology of the concentrations and exposure as requested. When model output is complete, the program closes all output files.

EXTRAN then requests directions from the user. If the user wishes to revise the environmental data and rerun the program, EXTRAN resets variables to their initial values and calls the subroutines used to revise the environmental data. After data revision is complete, the program returns to computation of model parameters.

Appendix B contains listings of the EXTRAN program, the subroutines, and the INCLUDE files.

5.0 REFERENCES

American National Standards Institute (ANSI). 1978. American National Standard FORTRAN, X3.9-1978 (FORTRAN 77). American National Standards Institute, New York.

Bander, T. J. 1982. PAVAN: An Atmospheric Dispersion Program for Evaluating Design Basis Accidental Releases of Radioactive Materials from Nuclear Power Stations. NUREG/CR-2858, PNL-4413. Prepared by Pacific Northwest Laboratory for the U.S. Nuclear Regulatory Commission, Washington, D.C.

Bird, R. B., W. E. Stewart, and E. N. Lightfoot. 1960. Transport Phenomena. John Wiley & Sons, New York.

Bolz, R. E., and G. L. Tuve. 1973. Handbook of Tables for Applied Engineering Science. CRC Press, Cleveland, Ohio.

Dickerson, M. H., and D. L. Ermak. 1988. "The Evaluation of Emergency Response Trace Gas and Dense Gas Dispersion Models." UCRL-99348, Lawrence Livermore National Laboratory, Livermore, California. Presented at the American Meteorological Society Sponsored Advanced Studies Institute Short Course on the Meteorological Aspects of Emergency Response, September 28-29, 1988.

Eimutis, E. C., and M. G. Konicek. 1972. "Derivations of Continuous Functions for the Lateral and Vertical Atmospheric Dispersion Coefficients." Atmos. Environ. 6:859-863.

Hinds, W. T. 1969. "Peak-to-Mean Concentration Ratios from Ground Level Sources in Building Wakes." Atmos. Environ. 3:145-156.

Idso, S. B. 1972. "Systematic Deviations of Clear Sky Atmospheric Thermal Radiation from Predictions of Empirical Formulae." Q. J. Roy. Meteorol. Soc. 98:399-401.

Langhaar, J. W. 1953. "Cooling Pond May Answer Your Water Cooling Problem." Chem. Eng. 60(8):194-198.

Murphy, K. G., and K. M. Campe. 1974. "Nuclear Power Plant Control Room Ventilation System Design for Meeting General Design Criterion 19." In Proceedings of the 13th AEC Air Cleaning Conference, August 12-15, 1974, San Francisco, California, CONF-740807, U.S. Atomic Energy Commission, Washington, D.C.

Oke, T. R. 1978. Boundary Layer Climates. Methuen & Co., New York.

Owczarski. P. C., K. W. Burk, J. V. Ramsdell, and D. D. Yasuda. 1990. FPFP 2: A Code for Following Airborne Fission Products in Generic Nuclear Plant Flow Paths. NUREG/CR-5658, PNL-7513. Prepared by Pacific Northwest Laboratory for the U.S. Nuclear Regulatory Commission, Washington, D.C.

Paltridge, G. W. 1970. "Daytime Long-Wave Radiation from the Sky." Q. J. Roy. Meteorol. Soc. 96:645-653.

Paltridge, G. W., and C. M. R. Platt. 1976. Radiative Processes in Meteorology and Climatology. Elsevier Scientific Publishing Company, New York.

Ramsdell, J. V. 1978. Impact of a Hanford Nuclear Energy Center on Cloudiness and Insolation. PNL-2638, Pacific Northwest Laboratory, Richland, Washington.

Ramsdell, J. V. 1988. Atmospheric Diffusion for Control Room Habitability Assessments. NUREG/CR-5055, PNL-6391. Prepared by Pacific Northwest Laboratory for the U.S. Nuclear Regulatory Commission, Washington, D.C.

Ramsdell, J. V. 1990a. "Alternatives to Current Procedures Used to Estimate Concentrations in Building Wakes." PNL-SA-17941, Pacific Northwest Laboratory, Richland, Washington. Presented at the 21st DOE/NRC Nuclear Air Cleaning Conference, August 13-16, 1990, San Diego, California.

Ramsdell, J. V. 1990b. "Diffusion in Building Wakes for Ground-Level Releases." Atmos. Environ. 24B(3):377-388.

Ramsdell, J. V., and W. T. Hinds. 1971. "Concentration Fluctuations and Peak-to-Mean Concentration Ratios in Plumes from a Ground-Level Continuous Point Source." Atmos. Environ. 5:483-495.

Ramsdell, J. V., G. F. Athey, and C. S. Glantz. 1983. MESOI Version 2.0: An Interactive Mesoscale Lagrangian Puff Dispersion Model With Deposition and Decay. NUREG/CR-3344, PNL-4753. Prepared by Pacific Northwest Laboratory for the U.S. Nuclear Regulatory Commission, Washington, D.C.

Reid, R. C., J. M. Prausnitz, and B. E. Poling. 1987. The Properties of Gases & Liquids. McGraw-Hill Book Company, New York, 741 pp.

Rosenberg, N. J., B. L. Blad, and S. B. Verma. 1983. Microclimate The Biological Environment. John Wiley & Sons, New York.

Sagendorf, J. F., J. T. Goll, and W. F. Sandusky. 1982. XOQDOQ: Computer Program for the Meteorological Evaluation of Routine Effluent Releases at Nuclear Power Stations. PNL-4380. Prepared by Pacific Northwest Laboratory for the U.S. Nuclear Regulatory Commission, Washington, D.C.

Swinbank, W. C. 1963. "Long-Wave Radiation from Clear Skies." Q. J. Roy. Meteorol. Soc. 89:339-348.

Wing, J. 1979. Toxic Vapor Concentrations in the Control Room Following a Postulated Accidental Release. NUREG/CR-0570, U.S. Nuclear Regulatory Commission, Washington, D.C.

APPENDIX A

EXAMPLE INPUT AND OUTPUT

APPENDIX A

EXAMPLE INPUT AND OUTPUT

This appendix contains EXTRAN output for five example problems. The output for each example includes the summary page that lists the input data and 2-min average concentration, a plot, the first part of the chronology file, and the first part of the mass and energy balance file. Only the first example includes the output title page that contains program information.

Note that there are two sets of output for Example 5. The environmental conditions were changed for the second set. You might also note the variations in the intake concentration in the plot and chronology in the first part of this example. This variation is an artifact of puff models that occurs occasionally. It happens when the separation between puffs approaches the upper limit in the model.

Program Title: EXTRAN VERSION 1.2

Developed For: U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Division of Reactor Accident Analysis

Date: October 1990

NRC Contact(s): C. Ferrell Phone: (FTS) 492 3944
Code Developer: J. V. Ramsdell Phone: (509) 376-8626
(FTS) 444-8626

Code Documentation:
EXTRAN: A Computer Code For Estimating Concentrations Of Toxic Substances At Control Room Air Intakes
NUREG/CR-5656

Example 1 -- EXTRAN Version 1.2

RUN DATE = 10/11/1990 RUN TIME = 15:34:24

SCENARIO:

Release Type		=	Liquid Tank Burst
Initial Mass	(kg)	=	1000.
Release Height	(m)	=	.0
Storage Temperature	(C)	=	20.0
Maximum Pool Radius	(m)	=	10.0
Intake Distance	(m)	=	250.
Intake Height	(m)	=	25.0
Building Area	(m**2)	=	100.

ENVIRONMENTAL CONDITIONS:

Wind Speed	(m/sec)	=	3.0
Atmospheric Stability Class		=	4
Air Temperature	(C)	=	20.0
Atmospheric Pressure	(mm Hg)	=	760.0
Solar Radiation	(watts/m**2)	=	650.0
Cloud Cover	(tenths)	=	2
Ground Temperature	(C)	=	25.0

EFFLUENT CHARACTERISTICS:

Material Released		=	CHLORINE
Molecular Weight	(gm/mole)	=	70.9
Initial Boiling Point	(C)	=	-34.1
Heat Capacity	(j/gm-C)	=	.946
Heat of Vapor.	(j/gm)	=	288.0
Specific Gravity		=	1.570
Diffusion Coef.	(cm**2/sec)	=	.079

MODEL PARAMETERS:

Puff Release Interval	(sec)	=	10
Time Step	(sec)	=	5
Delay Between Release and Intake	(sec)	=	50
Threshold Concentration	(g/m**3)	=	7.08E-05
To convert g/m**3 to ppm, multiply by			3.39E+02

RESULTS:

Average Concentration During First Two Minutes After Arrival of Plume	(g/m**3)	=	1.93E-01
Exposure Two Minutes After Arrival	(g-sec/m**3)	=	2.41E+01
Time From Plume Arrival to Max. Conc.	(sec)	=	30.
Max. Conc. in Two Minutes After Arrival	(g/m**3)	=	6.12E-01

ADDITIONAL OUTPUT FILES:

EXCR1534.001
EXMB1534.001

Example 1 -- EXTRAN Version 1.2

Run on 10/11/1990 at 15:34:24

TIME (sec)	CONCENTRATION (g/m**3)	EXPOSURE (g-sec/m**3)	MEAN CONC. (g/m**3)	NUM OF PUFFS
0	1.27E-03	6.35E-03	1.27E-03	7
5	1.50E-02	8.13E-02	8.13E-03	7
10	7.92E-02	4.77E-01	3.18E-02	8
15	2.32E-01	1.64E+00	8.18E-02	8
20	4.35E-01	3.81E+00	1.52E-01	9
25	5.85E-01	6.74E+00	2.25E-01	9
30	6.12E-01	9.80E+00	2.80E-01	10
35	5.32E-01	1.25E+01	3.11E-01	10
40	4.11E-01	1.45E+01	3.22E-01	11
45	2.99E-01	1.60E+01	3.20E-01	11
50	2.19E-01	1.71E+01	3.11E-01	12
55	1.69E-01	1.80E+01	2.99E-01	12
60	1.40E-01	1.87E+01	2.87E-01	13
65	1.23E-01	1.93E+01	2.75E-01	13
70	1.12E-01	1.98E+01	2.64E-01	14
75	1.04E-01	2.03E+01	2.54E-01	14
80	9.87E-02	2.08E+01	2.45E-01	15
85	9.41E-02	2.13E+01	2.37E-01	15
90	9.01E-02	2.18E+01	2.29E-01	16
95	8.66E-02	2.22E+01	2.22E-01	16
100	8.34E-02	2.26E+01	2.15E-01	17
105	8.06E-02	2.30E+01	2.09E-01	17
110	7.80E-02	2.34E+01	2.03E-01	18
115	7.56E-02	2.38E+01	1.98E-01	18
120	7.34E-02	2.41E+01	1.93E-01	19
125	7.13E-02	2.45E+01	1.88E-01	19
130	6.94E-02	2.48E+01	1.84E-01	20
135	6.77E-02	2.52E+01	1.80E-01	20
140	6.60E-02	2.55E+01	1.76E-01	21
145	6.44E-02	2.58E+01	1.72E-01	21
150	6.30E-02	2.62E+01	1.69E-01	22
155	6.16E-02	2.65E+01	1.65E-01	22
160	6.03E-02	2.68E+01	1.62E-01	23
165	5.90E-02	2.71E+01	1.59E-01	23
170	5.78E-02	2.73E+01	1.56E-01	24
175	5.67E-02	2.76E+01	1.54E-01	24
180	5.56E-02	2.79E+01	1.51E-01	25
185	5.46E-02	2.82E+01	1.48E-01	25
190	5.36E-02	2.85E+01	1.46E-01	26
195	5.27E-02	2.87E+01	1.44E-01	26
200	5.18E-02	2.90E+01	1.41E-01	27
205	5.09E-02	2.92E+01	1.39E-01	27
210	5.01E-02	2.95E+01	1.37E-01	28
215	4.93E-02	2.97E+01	1.35E-01	28
220	4.85E-02	3.00E+01	1.33E-01	29
225	4.77E-02	3.02E+01	1.31E-01	29
230	4.70E-02	3.04E+01	1.30E-01	30
235	4.63E-02	3.07E+01	1.28E-01	30

```
Example 1  --  EXTRAN Version 1.2
Run on 10/11/1990 at 15:34:24

DATA FORMAT:
MASS BALANCE:  TANK, CURRENT RELEASE, POOL, FLASHED, EVAPORATED
POOL STATUS:   VOLUME, RADIUS, AREA, DEPTH, TEMPERATURE
ENERGY BUDGET: NET SW, NET LW, ATM CONV, GRND COND, NET FLUX

TIME SINCE RELEASE =    0     TOTAL NUMBER OF PUFFS RELEASED =    2
MASS BALANCE         .00   1000.00    785.73    177.70     36.57
POOL STATUS          .52      4.08     52.38       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67  18689.06  20107.62
TIME SINCE RELEASE =   10     TOTAL NUMBER OF PUFFS RELEASED =    3
MASS BALANCE         .00       .00    760.30       .00     25.43
POOL STATUS          .50      3.99     50.05       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67  13215.16  14633.72
TIME SINCE RELEASE =   20     TOTAL NUMBER OF PUFFS RELEASED =    4
MASS BALANCE         .00       .00    739.77       .00     20.53
POOL STATUS          .48      3.93     48.43       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67  10790.13  12208.69
TIME SINCE RELEASE =   30     TOTAL NUMBER OF PUFFS RELEASED =    5
MASS BALANCE         .00       .00    722.16       .00     17.61
POOL STATUS          .47      3.87     47.12       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67   9344.53  10763.09
TIME SINCE RELEASE =   40     TOTAL NUMBER OF PUFFS RELEASED =    6
MASS BALANCE         .00       .00    706.55       .00     15.61
POOL STATUS          .46      3.83     46.00       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67   8358.00   9776.56
TIME SINCE RELEASE =   50     TOTAL NUMBER OF PUFFS RELEASED =    7
MASS BALANCE         .00       .00    692.41       .00     14.14
POOL STATUS          .45      3.78     45.00       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67   7629.78   9048.34
TIME SINCE RELEASE =   60     TOTAL NUMBER OF PUFFS RELEASED =    8
MASS BALANCE         .00       .00    679.42       .00     12.99
POOL STATUS          .44      3.75     44.10       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67   7063.80   8482.36
TIME SINCE RELEASE =   70     TOTAL NUMBER OF PUFFS RELEASED =    9
MASS BALANCE         .00       .00    667.36       .00     12.06
POOL STATUS          .43      3.71     43.28       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67   6607.58   8026.14
TIME SINCE RELEASE =   80     TOTAL NUMBER OF PUFFS RELEASED =   10
MASS BALANCE         .00       .00    656.07       .00     11.29
POOL STATUS          .43      3.68     42.51       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67   6229.69   7648.25
TIME SINCE RELEASE =   90     TOTAL NUMBER OF PUFFS RELEASED =   11
MASS BALANCE         .00       .00    645.44       .00     10.63
POOL STATUS          .42      3.65     41.79       .01    -34.10
ENERGY BUDGET     585.00    133.89    699.67   5910.00   7328.56
```

Example 2 -- EXTRAN Version 1.2

RUN DATE = 10/11/1990 RUN TIME = 15:22:11

SCENARIO:

Release Type		=	Liquid Tank Leak
Initial Mass	(kg)	=	3000.
Release Rate	(kg/sec)	=	1.00E+01
Release Height	(m)	=	.0
Storage Temperature	(C)	=	15.0
Maximum Pool Radius	(m)	=	10.0
Intake Distance	(m)	=	400.
Intake Height	(m)	=	25.0
Building Area	(m**2)	=	10.

ENVIRONMENTAL CONDITIONS:

Wind Speed	(m/sec)	=	5.0
Atmospheric Stability Class		=	6
Air Temperature	(C)	=	10.0
Atmospheric Pressure	(mm Hg)	=	775.0
Solar Radiation	(watts/m**2)	=	250.0
Cloud Cover	(tenths)	=	6
Ground Temperature	(C)	=	15.0

EFFLUENT CHARACTERISTICS:

Material Released		=	CARBON TETRACHLORIDE
Molecular Weight	(gm/mole)	=	153.8
Initial Boiling Point	(C)	=	77.5
Heat Capacity	(j/gm-C)	=	.841
Heat of Vapor.	(j/gm)	=	198.0
Specific Gravity		=	1.600
Diffusion Coef.	(cm**2/sec)	=	.058

MODEL PARAMETERS:

Puff Release Interval	(sec)	=	6
Time Step	(sec)	=	3
Delay Between Release and Intake	(sec)	=	69
Threshold Concentration	(ppm)	=	4.69E-05
To convert ppm to g/m**3, multiply by			6.75E-03

RESULTS:

Average Concentration During First Two Minutes After Arrival of Plume	(ppm)	=	2.69E-02
Exposure Two Minutes After Arrival	(g-sec/m**3)	=	2.23E-02
Time From Plume Arrival to Max. Conc.	(sec)	=	120.
Max. Conc. in Two Minutes After Arrival	(ppm)	=	5.11E-02

ADDITIONAL OUTPUT FILES:

EXCR1522.001
EXMB1522.001

Example 2 -- EXTRAN Version 1.2

Run on 10/11/1990 at 15:22:11

TIME (sec)	CONCENTRATION (ppm)	EXPOSURE (g-sec/m**3)	MEAN CONC. (ppm)	NUM OF PUFFS
0	4.90E-05	9.91E-07	4.90E-05	12
3	5.27E-04	1.17E-05	2.88E-04	13
6	1.97E-03	5.15E-05	8.47E-04	13
9	3.67E-03	1.26E-04	1.55E-03	14
12	5.41E-03	2.35E-04	2.32E-03	14
15	7.02E-03	3.77E-04	3.11E-03	15
18	8.61E-03	5.52E-04	3.89E-03	15
21	1.01E-02	7.56E-04	4.67E-03	16
24	1.16E-02	9.92E-04	5.44E-03	16
27	1.31E-02	1.26E-03	6.20E-03	17
30	1.45E-02	1.55E-03	6.96E-03	17
33	1.59E-02	1.87E-03	7.71E-03	18
36	1.73E-02	2.22E-03	8.44E-03	18
39	1.87E-02	2.60E-03	9.18E-03	19
42	2.00E-02	3.01E-03	9.90E-03	19
45	2.14E-02	3.44E-03	1.06E-02	20
48	2.27E-02	3.90E-03	1.13E-02	20
51	2.40E-02	4.38E-03	1.20E-02	21
54	2.52E-02	4.89E-03	1.27E-02	21
57	2.66E-02	5.43E-03	1.34E-02	22
60	2.78E-02	5.99E-03	1.41E-02	22
63	2.91E-02	6.58E-03	1.48E-02	23
66	3.03E-02	7.20E-03	1.55E-02	23
69	3.15E-02	7.84E-03	1.61E-02	24
72	3.27E-02	8.50E-03	1.68E-02	24
75	3.40E-02	9.19E-03	1.74E-02	25
78	3.51E-02	9.90E-03	1.81E-02	25
81	3.64E-02	1.06E-02	1.88E-02	26
84	3.75E-02	1.14E-02	1.94E-02	26
87	3.87E-02	1.22E-02	2.00E-02	27
90	3.98E-02	1.30E-02	2.07E-02	27
93	4.11E-02	1.38E-02	2.13E-02	28
96	4.21E-02	1.47E-02	2.19E-02	28
99	4.34E-02	1.55E-02	2.26E-02	29
102	4.44E-02	1.64E-02	2.32E-02	29
105	4.56E-02	1.74E-02	2.38E-02	30
108	4.66E-02	1.83E-02	2.44E-02	30
111	4.79E-02	1.93E-02	2.51E-02	31
114	4.89E-02	2.03E-02	2.57E-02	31
117	5.01E-02	2.13E-02	2.63E-02	32
120	5.11E-02	2.23E-02	2.69E-02	32
123	5.23E-02	2.34E-02	2.75E-02	33
126	5.32E-02	2.45E-02	2.81E-02	33
129	5.44E-02	2.56E-02	2.87E-02	34
132	5.54E-02	2.67E-02	2.93E-02	34
135	5.66E-02	2.78E-02	2.99E-02	35
138	5.75E-02	2.90E-02	3.05E-02	35

```
Example 2  --  EXTRAN Version 1.2
Run on 10/11/1990 at 15:22:11

DATA FORMAT:
MASS BALANCE:  TANK, CURRENT RELEASE, POOL, FLASHED, EVAPORATED
POOL STATUS:   VOLUME, RADIUS, AREA, DEPTH, TEMPERATURE
ENERGY BUDGET: NET SW, NET LW, ATM CONV, GRND COND, NET FLUX

TIME SINCE RELEASE =    0     TOTAL NUMBER OF PUFFS RELEASED =    1
MASS BALANCE    2940.00     60.00     59.90       .00       .10
POOL STATUS         .04      1.09      3.75       .01     14.60
ENERGY BUDGET    225.00   -111.24    -87.86       .00     25.90
TIME SINCE RELEASE =    6     TOTAL NUMBER OF PUFFS RELEASED =    2
MASS BALANCE    2880.00     60.00    119.70       .00       .19
POOL STATUS         .07      1.54      7.49       .01     14.46
ENERGY BUDGET    225.00   -110.16    -84.36     57.53     88.02
TIME SINCE RELEASE =   12     TOTAL NUMBER OF PUFFS RELEASED =    3
MASS BALANCE    2820.00     60.00    179.43       .00       .28
POOL STATUS         .11      1.89     11.23       .01     14.33
ENERGY BUDGET    225.00   -109.29    -81.54     84.75    118.91
TIME SINCE RELEASE =   18     TOTAL NUMBER OF PUFFS RELEASED =    4
MASS BALANCE    2760.00     60.00    239.07       .00       .36
POOL STATUS         .15      2.18     14.96       .01     14.21
ENERGY BUDGET    225.00   -108.53    -79.05    102.28    139.70
TIME SINCE RELEASE =   24     TOTAL NUMBER OF PUFFS RELEASED =    5
MASS BALANCE    2700.00     60.00    298.64       .00       .43
POOL STATUS         .19      2.44     18.69       .01     14.10
ENERGY BUDGET    225.00   -107.83    -76.78    115.08    155.47
TIME SINCE RELEASE =   30     TOTAL NUMBER OF PUFFS RELEASED =    6
MASS BALANCE    2640.00     60.00    358.13       .00       .51
POOL STATUS         .22      2.67     22.41       .01     13.99
ENERGY BUDGET    225.00   -107.18    -74.67    125.07    168.21
TIME SINCE RELEASE =   36     TOTAL NUMBER OF PUFFS RELEASED =    7
MASS BALANCE    2580.00     60.00    417.56       .00       .58
POOL STATUS         .26      2.88     26.13       .01     13.89
ENERGY BUDGET    225.00   -106.58    -72.69    133.18    178.91
TIME SINCE RELEASE =   42     TOTAL NUMBER OF PUFFS RELEASED =    8
MASS BALANCE    2520.00     60.00    476.91       .00       .65
POOL STATUS         .30      3.08     29.85       .01     13.79
ENERGY BUDGET    225.00   -106.00    -70.82    139.96    188.13
TIME SINCE RELEASE =   48     TOTAL NUMBER OF PUFFS RELEASED =    9
MASS BALANCE    2460.00     60.00    536.19       .00       .72
POOL STATUS         .34      3.27     33.56       .01     13.70
ENERGY BUDGET    225.00   -105.46    -69.04    145.73    196.23
TIME SINCE RELEASE =   54     TOTAL NUMBER OF PUFFS RELEASED =   10
MASS BALANCE    2400.00     60.00    595.41       .00       .78
POOL STATUS         .37      3.44     37.26       .01     13.61
ENERGY BUDGET    225.00   -104.94    -67.34    150.73    203.45
```

```
Example 3  --  EXTRAN Version 1.2

RUN DATE = 10/11/1990  RUN TIME = 15:28:54

SCENARIO:
Release Type                =  Gas Tank Burst
Initial Mass          (kg) =        500.
Release Height         (m) =          3.0
Storage Temperature    (C) =         20.0
Intake Distance        (m) =        200.
Intake Height          (m) =         20.0
Building Area       (m**2) =         10.

ENVIRONMENTAL CONDITIONS:
Wind Speed                (m/sec) =        2.0
Atmospheric Stability Class       =        3
Air Temperature                (C) =       20.0
Atmospheric Pressure       (mm Hg) =      760.0

EFFLUENT CHARACTERISTICS:
Material Released                 =      HYDROGEN
Molecular Weight         (gm/mole) =        2.0

MODEL PARAMETERS:
Puff Release Interval              (sec) =   10
Time Step                          (sec) =    5
Delay Between Release and Intake   (sec) =   55
Threshold Concentration            (ppm) =    2.99E+00
To convert ppm to g/m**3, multiply by       8.40E-05

RESULTS:
Average Concentration During First Two Minutes
     After Arrival of Plume                    (ppm) =    7.74E+03
Exposure Two Minutes After Arrival    (g-sec/m**3) =    8.12E+01
Time From Plume Arrival to Max. Conc.         (sec) =   40.
Max. Conc. in Two Minutes After Arrival       (ppm) =    2.77E+04

ADDITIONAL OUTPUT FILES:
     EXCR1528.001
     EXMB1528.001
```

Example 3 -- EXTRAN Version 1.2

Run on 10/11/1990 at 15:28:54

TIME (sec)	CONCENTRATION (ppm)	EXPOSURE (g-sec/m**3)	MEAN CONC. (ppm)	NUM OF PUFFS
0	1.42E+01	5.98E-03	1.42E+01	1
5	1.55E+02	7.10E-02	8.45E+01	1
10	9.21E+02	4.58E-01	3.63E+02	1
15	3.40E+03	1.88E+00	1.12E+03	1
20	8.55E+03	5.47E+00	2.61E+03	1
25	1.59E+04	1.21E+01	4.82E+03	1
30	2.30E+04	2.18E+01	7.41E+03	1
35	2.74E+04	3.33E+01	9.91E+03	1
40	2.77E+04	4.49E+01	1.19E+04	1
45	2.47E+04	5.53E+01	1.32E+04	1
50	1.99E+04	6.36E+01	1.38E+04	1
55	1.47E+04	6.98E+01	1.39E+04	1
60	1.02E+04	7.41E+01	1.36E+04	1
65	6.68E+03	7.69E+01	1.31E+04	1
70	4.21E+03	7.87E+01	1.25E+04	1
75	2.56E+03	7.97E+01	1.19E+04	1
80	1.52E+03	8.04E+01	1.13E+04	1
85	8.90E+02	8.08E+01	1.07E+04	1
90	5.12E+02	8.10E+01	1.01E+04	1
95	2.92E+02	8.11E+01	9.66E+03	1
100	1.65E+02	8.12E+01	9.20E+03	1
105	9.34E+01	8.12E+01	8.79E+03	1
110	5.27E+01	8.12E+01	8.41E+03	1
115	2.97E+01	8.12E+01	8.06E+03	1
120	1.68E+01	8.12E+01	7.74E+03	1
125	9.57E+00	8.12E+01	7.44E+03	1
130	5.46E+00	8.12E+01	7.17E+03	1
135	3.14E+00	8.12E+01	6.91E+03	1
140	1.81E+00	8.12E+01	6.67E+03	1
145	1.05E+00	8.12E+01	6.45E+03	1
150	6.17E-01	8.12E+01	6.24E+03	1

Example 3 -- EXTRAN Version 1.2

Run on 10/11/1990 at 15:28:54

DATA FORMAT:
MASS BALANCE: TANK, CURRENT RELEASE, POOL, FLASHED, EVAPORATED
POOL STATUS: VOLUME, RADIUS, AREA, DEPTH, TEMPERATURE
ENERGY BUDGET: NET SW, NET LW, ATM CONV, GRND COND, NET FLUX

TIME SINCE RELEASE = 0 TOTAL NUMBER OF PUFFS RELEASED = 1

MASS BALANCE	.00	500.00	.00	500.00	.00
POOL STATUS	.00	.00	.00	.00	.00
ENERGY BUDGET	.00	.00	.00	.00	.00

```
Example 4  --  EXTRAN Version 1.2

RUN DATE = 10/11/1990  RUN TIME = 15:39:30

SCENARIO:
Release Type                =  Gas Tank Leak
Initial Mass          (kg)  =      2000.
Release Rate      (kg/sec)  =         1.00E+01
Release Height        (m)   =         5.0
Storage Temperature   (C)   =        10.0
Intake Distance       (m)   =       300.
Intake Height         (m)   =        20.0
Building Area       (m**2)  =        50.
Vent Flow         (m**3/s)  =          .00

ENVIRONMENTAL CONDITIONS:
Wind Speed                  (m/sec) =        2.0
Atmospheric Stability Class         =        6
Air Temperature                 (C) =        5.0
Atmospheric Pressure        (mm Hg) =      760.0

EFFLUENT CHARACTERISTICS:
Material Released                   =      AMMONIA
Molecular Weight          (gm/mole) =       17.0

MODEL PARAMETERS:
Puff Release Interval               (sec) =   10
Time Step                           (sec) =    5
Delay Between Release and Intake    (sec) =   95
Threshold Concentration             (ppm) =    6.78E-02
To convert ppm to g/m**3, multiply by     =    7.45E-04

RESULTS:
Average Concentration During First Two Minutes
     After Arrival of Plume                    (ppm) =    1.21E+03
Exposure Two Minutes After Arrival  (g-sec/m**3) =    1.13E+02
Time From Plume Arrival to Max. Conc.          (sec) =  120.
Max. Conc. in Two Minutes After Arrival        (ppm) =    1.97E+03

ADDITIONAL OUTPUT FILES:
     EXCR1539.001
     EXMB1539.001
```

Example 4 -- EXTRAN Version 1.2
Run on 10/11/1990 at 15:39:30

TIME (sec)	CONCENTRATION (ppm)	EXPOSURE (g-sec/m**3)	MEAN CONC. (ppm)	NUM OF PUFFS
0	4.10E-01	1.53E-03	4.10E-01	10
5	2.03E+00	9.08E-03	1.22E+00	11
10	8.14E+00	3.94E-02	3.52E+00	11
15	2.55E+01	1.34E-01	9.02E+00	12
20	6.65E+01	3.82E-01	2.05E+01	12
25	1.47E+02	9.28E-01	4.15E+01	13
30	2.80E+02	1.97E+00	7.57E+01	13
35	4.72E+02	3.73E+00	1.25E+02	14
40	7.13E+02	6.39E+00	1.91E+02	14
45	9.79E+02	1.00E+01	2.69E+02	15
50	1.24E+03	1.46E+01	3.58E+02	15
55	1.47E+03	2.01E+01	4.50E+02	16
60	1.65E+03	2.63E+01	5.42E+02	16
65	1.78E+03	3.29E+01	6.31E+02	17
70	1.87E+03	3.98E+01	7.13E+02	17
75	1.92E+03	4.70E+01	7.88E+02	18
80	1.94E+03	5.42E+01	8.56E+02	18
85	1.96E+03	6.15E+01	9.17E+02	19
90	1.96E+03	6.88E+01	9.73E+02	19
95	1.97E+03	7.61E+01	1.02E+03	20
100	1.97E+03	8.35E+01	1.07E+03	20
105	1.97E+03	9.08E+01	1.11E+03	20
110	1.97E+03	9.81E+01	1.15E+03	20
115	1.97E+03	1.05E+02	1.18E+03	20
120	1.97E+03	1.13E+02	1.21E+03	20
125	1.97E+03	1.20E+02	1.24E+03	20
130	1.97E+03	1.27E+02	1.27E+03	20
135	1.97E+03	1.35E+02	1.29E+03	20
140	1.97E+03	1.42E+02	1.32E+03	20
145	1.97E+03	1.49E+02	1.34E+03	20
150	1.97E+03	1.57E+02	1.36E+03	20

Example 4 -- EXTRAN Version 1.2
Run on 10/11/1990 at 15:39:30

DATA FORMAT:
MASS BALANCE: TANK, CURRENT RELEASE, POOL, FLASHED, EVAPORATED
POOL STATUS: VOLUME, RADIUS, AREA, DEPTH, TEMPERATURE
ENERGY BUDGET: NET SW, NET LW, ATM CONV, GRND COND, NET FLUX

TIME SINCE RELEASE = 0 TOTAL NUMBER OF PUFFS RELEASED = 1

MASS BALANCE	1900.00	100.00	.00	100.00	.00
POOL STATUS	.00	.00	.00	.00	.00
ENERGY BUDGET	.00	.00	.00	.00	.00

TIME SINCE RELEASE = 10 TOTAL NUMBER OF PUFFS RELEASED = 2

MASS BALANCE	1800.00	100.00	.00	100.00	.00
POOL STATUS	.00	.00	.00	.00	.00
ENERGY BUDGET	.00	.00	.00	.00	.00

```
Example 5  --  EXTRAN Version 1.2

RUN DATE = 10/11/1990  RUN TIME = 16:07:49

SCENARIO:
Release Type                =  Gas Tank Leak
Initial Activity     (Ci)   =       1000.
Release Rate     (Ci/sec)   =          5.00E-01
Release Height       (m)    =         25.0
Storage Temperature  (C)    =         20.0
Intake Distance      (m)    =         50.
Intake Height        (m)    =         25.0
Building Area     (m**2)    =       2000.
Vent Flow       (m**3/s)    =         10.00

ENVIRONMENTAL CONDITIONS:
Wind Speed                (m/sec) =        2.0
Atmospheric Stability Class       =        3
Air Temperature              (C)  =       25.0
Atmospheric Pressure      (mm Hg) =      730.0

EFFLUENT CHARACTERISTICS:
Material Released                 =      KRYPTON-85
Molecular Weight        (gm/mole) =       84.9

MODEL PARAMETERS:
Puff Release Interval           (sec) =   10
Time Step                       (sec) =    5
Delay Between Release and Intake (sec) =   10
Threshold Concentration     (mCi/m**3) =    1.28E-05

RESULTS:
Average Concentration During First Two Minutes
     After Arrival of Plume              (mCi/m**3) =    1.93E-01
Exposure Two Minutes After Arrival  (mCi-sec/m**3) =    2.42E+01
Time From Plume Arrival to Max. Conc.        (sec) =  115.
Max. Conc. in Two Minutes After Arrival (mCi/m**3) =    2.23E-01

ADDITIONAL OUTPUT FILES:
     EXCR1607.001
     EXMB1607.001
```

Example 5 -- EXTRAN Version 1.2
Run on 10/11/1990 at 16:07:49

TIME (sec)	CONCENTRATION (mCi/m**3)	EXPOSURE (mCi-sec/m**3)	MEAN CONC. (mCi/m**3)	NUM OF PUFFS
0	3.87E-02	1.93E-01	3.87E-02	2
5	1.45E-01	9.18E-01	9.18E-02	2
10	1.52E-01	1.68E+00	1.12E-01	3
15	2.04E-01	2.70E+00	1.35E-01	3
20	1.80E-01	3.60E+00	1.44E-01	4
25	2.17E-01	4.68E+00	1.56E-01	4
30	1.87E-01	5.62E+00	1.60E-01	5
35	2.21E-01	6.72E+00	1.68E-01	5
40	1.89E-01	7.66E+00	1.70E-01	6
45	2.22E-01	8.77E+00	1.75E-01	6
50	1.90E-01	9.72E+00	1.77E-01	7
55	2.22E-01	1.08E+01	1.81E-01	7
60	1.90E-01	1.18E+01	1.81E-01	8
65	2.23E-01	1.29E+01	1.84E-01	8
70	1.90E-01	1.38E+01	1.85E-01	9
75	2.23E-01	1.50E+01	1.87E-01	9
80	1.90E-01	1.59E+01	1.87E-01	10
85	2.23E-01	1.70E+01	1.89E-01	10
90	1.90E-01	1.80E+01	1.89E-01	11
95	2.23E-01	1.91E+01	1.91E-01	11
100	1.90E-01	2.00E+01	1.91E-01	12
105	2.23E-01	2.12E+01	1.92E-01	12
110	1.90E-01	2.21E+01	1.92E-01	13
115	2.23E-01	2.32E+01	1.94E-01	13
120	1.90E-01	2.42E+01	1.93E-01	14
125	2.23E-01	2.53E+01	1.95E-01	14
130	1.90E-01	2.62E+01	1.94E-01	15
135	2.23E-01	2.74E+01	1.95E-01	15
140	1.90E-01	2.83E+01	1.95E-01	16
145	2.23E-01	2.94E+01	1.96E-01	16
150	1.90E-01	3.04E+01	1.96E-01	17

Example 5 -- EXTRAN Version 1.2
Run on 10/11/1990 at 16:07:49

DATA FORMAT:
MASS BALANCE: TANK, CURRENT RELEASE, POOL, FLASHED, EVAPORATED
POOL STATUS: VOLUME, RADIUS, AREA, DEPTH, TEMPERATURE
ENERGY BUDGET: NET SW, NET LW, ATM CONV, GRND COND, NET FLUX

TIME SINCE RELEASE =	0	TOTAL NUMBER OF PUFFS RELEASED =			1
MASS BALANCE	995.00	5.00	.00	5.00	.00
POOL STATUS	.00	.00	.00	.00	.00
ENERGY BUDGET	.00	.00	.00	.00	.00
TIME SINCE RELEASE =	10	TOTAL NUMBER OF PUFFS RELEASED =			2
MASS BALANCE	990.00	5.00	.00	5.00	.00
POOL STATUS	.00	.00	.00	.00	.00
ENERGY BUDGET	.00	.00	.00	.00	.00

```
Example 5  --  EXTRAN Version 1.2

RUN DATE = 10/11/1990  RUN TIME = 16:07:49

SCENARIO:
Release Type                =  Gas Tank Leak
Initial Mass          (Ci)  =       1000.
Release Rate      (Ci/sec)  =          5.00E-01
Release Height         (m)  =         25.0
Storage Temperature    (C)  =         20.0
Intake Distance        (m)  =         50.
Intake Height          (m)  =         25.0
Building Area       (m**2)  =       2000.
Vent Flow         (m**3/s)  =         10.00

ENVIRONMENTAL CONDITIONS:
Wind Speed                  (m/sec) =        4.0
Atmospheric Stability Class         =        5
Air Temperature                 (C) =       15.0
Atmospheric Pressure        (mm Hg) =      730.0

EFFLUENT CHARACTERISTICS:
Material Released                   =      KRYPTON-85
Molecular Weight          (gm/mole) =       84.9

MODEL PARAMETERS:
Puff Release Interval               (sec) =    2
Time Step                           (sec) =    1
Delay Between Release and Intake    (sec) =    7
Threshold Concentration      (mCi/m**3) =    2.09E-05

RESULTS:
Average Concentration During First Two Minutes
     After Arrival of Plume                 (mCi/m**3) =    3.80E-01
Exposure Two Minutes After Arrival  (mCi-sec/m**3) =    4.60E+01
Time From Plume Arrival to Max. Conc.          (sec) =   23.
Max. Conc. in Two Minutes After Arrival (mCi/m**3) =    3.93E-01

ADDITIONAL OUTPUT FILES:
     EXCR1607.002
     EXMB1607.002
```

Example 5 -- EXTRAN Version 1.2
Run on 10/11/1990 at 16:07:49

TIME (sec)	CONCENTRATION (mCi/m**3)	EXPOSURE (mCi-sec/m**3)	MEAN CONC. (mCi/m**3)	NUM OF PUFFS
0	4.53E-04	4.53E-04	4.53E-04	4
1	1.29E-02	1.34E-02	6.68E-03	5
2	7.37E-02	8.71E-02	2.90E-02	5
3	1.71E-01	2.58E-01	6.44E-02	6
4	2.60E-01	5.18E-01	1.04E-01	6
5	3.22E-01	8.40E-01	1.40E-01	7
6	3.58E-01	1.20E+00	1.71E-01	7
7	3.76E-01	1.57E+00	1.97E-01	8
8	3.85E-01	1.96E+00	2.18E-01	8
9	3.90E-01	2.35E+00	2.35E-01	9
10	3.91E-01	2.74E+00	2.49E-01	9
11	3.93E-01	3.13E+00	2.61E-01	10
12	3.93E-01	3.53E+00	2.71E-01	10
13	3.93E-01	3.92E+00	2.80E-01	11
14	3.93E-01	4.31E+00	2.87E-01	11
15	3.93E-01	4.71E+00	2.94E-01	12
16	3.93E-01	5.10E+00	3.00E-01	12
17	3.93E-01	5.49E+00	3.05E-01	13
18	3.93E-01	5.88E+00	3.10E-01	13
19	3.93E-01	6.28E+00	3.14E-01	14
20	3.93E-01	6.67E+00	3.18E-01	14
21	3.93E-01	7.06E+00	3.21E-01	15
22	3.93E-01	7.46E+00	3.24E-01	15
23	3.93E-01	7.85E+00	3.27E-01	16
24	3.93E-01	8.24E+00	3.30E-01	16
25	3.93E-01	8.64E+00	3.32E-01	17
26	3.93E-01	9.03E+00	3.34E-01	17
27	3.93E-01	9.42E+00	3.37E-01	18
28	3.93E-01	9.82E+00	3.39E-01	18
29	3.93E-01	1.02E+01	3.40E-01	19
30	3.93E-01	1.06E+01	3.42E-01	19

Example 5 -- EXTRAN Version 1.2
Run on 10/11/1990 at 16:07:49

DATA FORMAT:
MASS BALANCE: TANK, CURRENT RELEASE, POOL, FLASHED, EVAPORATED
POOL STATUS: VOLUME, RADIUS, AREA, DEPTH, TEMPERATURE
ENERGY BUDGET: NET SW, NET LW, ATM CONV, GRND COND, NET FLUX

TIME SINCE RELEASE = 0 TOTAL NUMBER OF PUFFS RELEASED = 1

MASS BALANCE	999.00	1.00	.00	1.00	.00
POOL STATUS	.00	.00	.00	.00	.00
ENERGY BUDGET	.00	.00	.00	.00	.00

TIME SINCE RELEASE = 2 TOTAL NUMBER OF PUFFS RELEASED = 2

MASS BALANCE	998.00	1.00	.00	1.00	.00
POOL STATUS	.00	.00	.00	.00	.00
ENERGY BUDGET	.00	.00	.00	.00	.00

APPENDIX B

CODE LISTINGS

APPENDIX B

CODE LISTINGS

```
      PROGRAM EXTRAN

C*************************************************************************
C
C     EXTRAN                                                  VERSION  1.2
C     External Transport Dispersion Model For Conrol Room
C     Habitability Assessments
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 11/88
C     Updated: 10/90
C
C     Description:  Controlling program for estimating concentration
C                   time histories following release of a gas or
C                   volatile liquid.  Based on NUREG/CR-5055 and
C                   NUREG-0570.
C
C     Relationship to other modules:
C
C         Makes calls to: CEPLOT, CHIT, INPUT, MODELPAR, NSIG, PUFFINIT,
C                         PUFFMASS, RELPUFF, RINPUT, WSIG
C
C         Called from:     NONE
C
C*************************************************************************

      INCLUDE 'EFFLUENT.INC'
      INCLUDE 'ENVIRON.INC'
      INCLUDE 'OPTIONS.INC'
      INCLUDE 'POOL.INC'
      INCLUDE 'PRINT.INC'
      INCLUDE 'PUFF.INC'
      INCLUDE 'SCENARIO.INC'

      INTEGER*2 time, time_one, puffid, i, iflg, ITER,
     .          YY, MM, DD, HH, MIN, SS, HS

      REAL*4 wsigy, wsigz, delta_dist, maxconc, maxtime, cur_rel,
     .       tty, ttz, chimin
```

```
      CHARACTER formfeed, linefeed, yn, UC
      CHARACTER*8 RTIME
      CHARACTER*10 RDATE
      CHARACTER*20 gname
      CHARACTER*70 title

      formfeed = CHAR(12)
      linefeed = CHAR(10)

      PRTFILE  = 'EXPR0000.000'
      CRONFILE = 'EXCR0000.000'
      MBFILE   = 'EXMB0000.000'

      CALL GETDAT( YY,MM,DD )
      CALL GETTIM( HH,MIN,SS,HS )

      RDATE = '  /  /    '
      WRITE( RDATE(1:2), '(I2)' ) MM
      WRITE( RDATE(4:5), '(I2)' ) DD
      WRITE( RDATE(7:10), '(I4)' ) YY

      RTIME = '  :  :  '
      WRITE( RTIME(1:2), '(I2.2)' ) HH
      WRITE( RTIME(4:5), '(I2.2)' ) MIN
      WRITE( RTIME(7:8), '(I2.2)' ) SS

      WRITE(*,11) RDATE, RTIME
   11 FORMAT(10X,'PROGRAM RUN ', A10, '   AT ', A8)

      run_count = 1

C     NAME AND OPEN OUTPUT FILES

      WRITE( PRTFILE(5:6), '(I2.2)' ) HH
      WRITE( CRONFILE(5:6), '(I2.2)' ) HH
      WRITE( MBFILE(5:6), '(I2.2)' ) HH

      WRITE( PRTFILE(7:8), '(I2.2)' ) MIN
      WRITE( CRONFILE(7:8), '(I2.2)' ) MIN
      WRITE( MBFILE(7:8), '(I2.2)' ) MIN

      WRITE( PRTFILE(10:12), '(I3.3)' ) run_count
      WRITE( CRONFILE(10:12), '(I3.3)' ) run_count
      WRITE( MBFILE(10:12), '(I3.3)' ) run_count

      CALL input(title,RDATE,RTIME,gname)

C     RE-ENTRY POINT FOR MULTIPLE MODEL RUNS

 1000 CONTINUE

      iflg = 0
      time_one = 0
```

```
C     COMPUTE MODEL PARAMETERS

      WRITE(*,'(/10X,A)') 'COMPUTING MODEL RUN PARAMETERS '

      CALL modelpar(chimin)

      IF( statflg ) WRITE(12,12) TITLE, RDATE, RTIME
   12   FORMAT( /7X, A70, /7X, 'Run on ', A10, ' at ', A8,
     .      //7X, 'DATA FORMAT:', /7X,
     .   'MASS BALANCE:  TANK, CURRENT RELEASE, POOL,',
     .   ' FLASHED, EVAPORATED', /7X,'POOL STATUS:   VOLUME, RADIUS, ',
     .   'AREA, DEPTH, TEMPERATURE',/7X,'ENERGY BUDGET: NET SW, NET LW,',
     .   ' ATM CONV, GRND COND, NET FLUX' / )

C     ENTERING TIME STEP LOOP

      ITER = 0
      count    = 0
      numpuffs = 0
      maxconc  = 0.0
      pool_mass = 0.0

      WRITE(*,'(/10X,A)') 'ENTERING DIFFUSION COMPUTATION LOOP '

      DO time = 0, 1800, delta_time

C     DETERMINE IF IT IS TIME TO RELEASE ANOTHER PUFF

        IF( (MOD(time,pri) .EQ. 0) .AND.
     .       ( (tank_mass .GT. 0.0) .OR. (pool_mass .GT. 0.0) ) )  THEN
          pmass1 = 0.0
          pmass2 = 0.0
          CALL puffmass( pmass1, pmass2, time )
          IF( numpuffs .EQ. 0 ) THEN
            chimin = chimin * AMAX1( pmass1, pmass2 )
          ENDIF
          CALL puffinit
          CALL relpuff

C     OUTPUT MASS AND ENERGY BALANCE

          IF( rel_type .LE. 2 ) THEN
            cur_rel = rel_mass + pmass1
          ELSE
            cur_rel = pmass1
          ENDIF

          IF( statflg ) THEN
            ITER = ITER + 1
            IF( MOD(ITER,11) .EQ. 0 ) THEN
              WRITE(12,12) TITLE, RDATE, RTIME
              ITER = 0
            ENDIF
```

```
            WRITE(12,15) time, numpuffs,
     .         tank_mass, cur_rel, pool_mass, pmass1, pmass2,
     .         pool_vol, pool_radius, pool_area, pool_thick, pool_temp,
     .         net_swrad, net_lwrad, air_conv, grnd_cond, net_flux
  15         FORMAT(7x,'TIME SINCE RELEASE = ',I4,5X,
     .             'TOTAL NUMBER OF PUFFS RELEASED = ',
     .             I4, /7X, 'MASS BALANCE ', 5F10.2, /7X,'POOL STATUS  ',
     .             5F10.2, /7X, 'ENERGY BUDGET', 5F10.2)
           ENDIF

         ENDIF

C        MOVE AND DIFFUSE ALL PUFFS USING NRC SIGMA CURVES AND
C        BUILDING WAKE IF TV >= 0

         DO puffid=1, numpuffs
           puffage(puffid) = puffage(puffid) + FLOAT(delta_time)
           delta_dist  = delta_time * ubar

           CALL nsig(delta_dist, puffnsigy(puffid), puffnsigz(puffid))

           IF( (pufftvy(puffid) .GE. 0.0) .AND.
     .         (pufftvz(puffid) .GE. 0.0) ) THEN
             tty = puffage(puffid) + pufftvy(puffid)
             ttz = puffage(puffid) + pufftvz(puffid)
             CALL wsig(tty, ttz, wsigy, wsigz)
           ELSE
             wsigy = 0.0
             wsigz = 0.0
           ENDIF

           pufftsigy(puffid) = SQRT(puffnsigy(puffid)**2 + wsigy**2)
           pufftsigz(puffid) = SQRT(puffnsigz(puffid)**2 + wsigz**2)

         ENDDO

C        COMPUTE CONCENTRATION, EXPOSURE, AND MEAN CONCENTRATION

           CALL chit( time, time_one, chimin )

           IF( (iflg .EQ. 0) .AND. (time_one .GT. 0) ) THEN
             iflg = 1
             IF( prtflg1 ) THEN

C        OUTPUT TO PRINTER

               WRITE(3,'(10X,A,I4)')
     .           ' Delay Between Release and Intake (sec) = ', time_one
               IF( ppmflg ) THEN
                 WRITE(3,'(10X,A,1PE11.2)')
     .             ' Threshold Concentration          (ppm) = ',
     .                          chimin * 1.0E3 * ppmconv
```

```
            WRITE(3,'(10X,A,1PE11.2)')
     .        ' To convert ppm to g/m**3, multiply by    ', 1./ppmconv

          ELSE IF( ciflg ) THEN
            WRITE(3,'(10X,A,1PE11.2)')
     .        ' Threshold Concentration     (mCi/m**3) = ',
     .                  chimin * 1.0E3
          ELSE
            WRITE(3,'(10X,A,1PE11.2)')
     .        ' Threshold Concentration       (g/m**3) = ',
     .                  chimin * 1.0E3
            WRITE(3,'(10X,A,1PE11.2)')
     .        ' To convert g/m**3 to ppm, multiply by    ', ppmconv
          ENDIF
        ENDIF

        IF( prtflg2 ) THEN

C       OUTPUT TO FILE

          WRITE(10,'(10X,A,I4)')
     .        ' Delay Between Release and Intake (sec) = ', time_one
          IF( ppmflg ) THEN
            WRITE(10,'(10X,A,1PE11.2)')
     .        ' Threshold Concentration          (ppm) = ',
     .                  chimin * 1.0E3 * ppmconv
            WRITE(10,'(10X,A,1PE11.2)')
     .        ' To convert ppm to g/m**3, multiply by    ', 1./ppmconv

          ELSE IF( ciflg ) THEN
            WRITE(10,'(10X,A,1PE11.2)')
     .        ' Threshold Concentration     (mCi/m**3) = ',
     .                  chimin * 1.0E3
          ELSE
            WRITE(10,'(10X,A,1PE11.2)')
     .        ' Threshold Concentration       (g/m**3) = ',
     .                  chimin * 1.0E3
            WRITE(10,'(10X,A,1PE11.2)')
     .        ' To convert g/m**3 to ppm, multiply by    ', ppmconv
          ENDIF
        ENDIF

      ENDIF

      IF(count .LE. 90) THEN
        CYCLE
      ELSE IF(count .EQ. 181) THEN
        GOTO 2001
      ELSE IF( (count .EQ. 91) .AND. (ttime(count) .GT. 270) ) THEN
        GOTO 2001
      ENDIF
    ENDDO
```

```
 2001 CONTINUE

      IF( ppmflg ) THEN

C     CONVERT g/m**3 TO ppm

        DO i = 1,count
          conc(i) = conc(i) * ppmconv
          avconc(i) = avconc(i) * ppmconv
        ENDDO
      ENDIF

      WRITE(*,'(/10X,A)') 'WRITING OUT RESULTS'

      IF( prtflg1 ) WRITE(3,'(//10X,A)') ' RESULTS:'
      IF( prtflg2 ) WRITE(10,'(//10X,A)') ' RESULTS:'

      DO i=1, count
        IF( (ttime(i) .LE. 120) .AND. (conc(i) .GT. maxconc) ) THEN
          maxconc = conc(i)
          maxtime = ttime(i)
        ENDIF
        IF( ( (i .LE. 180) .AND.
     .       (ttime(i) .LE. 120) .AND. (ttime(i+1) .GT. 120) )
     .      .OR. ( (i .EQ. 181) .AND. (ttime(i-1) .LT. 120) ) ) THEN

          IF( prtflg1 ) THEN
            IF( ppmflg ) THEN
              WRITE(3,40)
     .    ' Average Concentration During First Two Minutes',
     .    '      After Arrival of Plume                  (ppm) = ',
     .                                                    avconc(i)
   40         FORMAT(10x,A,/10x,A,1PE11.2)
            ELSE IF( ciflg ) THEN
              WRITE(3,40)
     .    ' Average Concentration During First Two Minutes',
     .    '      After Arrival of Plume            (mCi/m**3) = ',
     .                                                    avconc(i)
            ELSE
              WRITE(3,40)
     .    ' Average Concentration During First Two Minutes',
     .    '      After Arrival of Plume              (g/m**3) = ',
     .                                                    avconc(i)
            ENDIF

            IF( ciflg ) THEN
              WRITE(3,60)
     .    ' Exposure Two Minutes After Arrival  (mCi-sec/m**3) = ',
     .                                                     expos(i)
            ELSE
              WRITE(3,60)
     .    ' Exposure Two Minutes After Arrival    (g-sec/m**3) = ',
     .                                                     expos(i)
```

```
        ENDIF
      ENDIF

      IF( prtflg2 ) THEN
        IF( ppmflg ) THEN
          WRITE(10,40)
     .    ' Average Concentration During First Two Minutes',
     .    '      After Arrival of Plume                  (ppm) = ',
     .                                                    avconc(i)
        ELSE IF( ciflg ) THEN
          WRITE(10,40)
     .    ' Average Concentration During First Two Minutes',
     .    '      After Arrival of Plume            (mCi/m**3) = ',
     .                                                    avconc(i)
        ELSE
          WRITE(10,40)
     .    ' Average Concentration During First Two Minutes',
     .    '      After Arrival of Plume              (g/m**3) = ',
     .                                                    avconc(i)
        ENDIF

        IF( ciflg ) THEN
          WRITE(10,60)
     .    ' Exposure Two Minutes After Arrival  (mCi-sec/m**3) = ',
     .                                                     expos(i)
        ELSE
          WRITE(10,60)
     .    ' Exposure Two Minutes After Arrival    (g-sec/m**3) = ',
     .                                                     expos(i)
        ENDIF
      ENDIF

      GOTO 3001

    ENDIF
  ENDDO

3001 CONTINUE

  IF( prtflg1 ) THEN
    WRITE(3,50)
     .    ' Time From Plume Arrival to Max. Conc.        (sec) = ',
     .                                                      maxtime
 50   FORMAT(10X,A,F5.0)

    IF( ppmflg ) THEN
      WRITE(3,60)
     .    ' Max. Conc. in Two Minutes After Arrival      (ppm) = ',
     .                                                      maxconc
 60     FORMAT(10x,A,1PE11.2)
    ELSE IF( ciflg ) THEN
      WRITE(3,60)
     .    ' Max. Conc. in Two Minutes After Arrival (mCi/m**3) = ',
```

```
     .                                                                maxconc
        ELSE
          WRITE(3,60)
     .     ' Max. Conc. in Two Minutes After Arrival   (g/m**3) = ',
     .                                                                maxconc
        ENDIF
       ENDIF

       IF( prtflg2 ) THEN
         WRITE(10,50)
     .     ' Time From Plume Arrival to Max. Conc.        (sec) = ',
     .                                                                maxtime

         IF( ppmflg ) THEN
           WRITE(10,60)
     .     ' Max. Conc. in Two Minutes After Arrival     (ppm)  = ',
     .                                                                maxconc
         ELSE IF( ciflg ) THEN
           WRITE(10,60)
     .     ' Max. Conc. in Two Minutes After Arrival (mCi/m**3) = ',
     .                                                                maxconc
         ELSE
           WRITE(10,60)
     .     ' Max. Conc. in Two Minutes After Arrival   (g/m**3) = ',
     .                                                                maxconc
         ENDIF
       ENDIF

       IF( prtflg1 .AND. (prtflg2 .OR. histflg2 .OR. statflg) ) THEN
         WRITE(3, '(/10X,A)' ) 'OUTPUT FILES: '
         IF( prtflg2 ) WRITE(3,'(15X,A)') PRTFILE
         IF( histflg2 ) WRITE(3,'(15X,A)') CRONFILE
         IF( statflg ) WRITE(3,'(15X,A)') MBFILE
       ENDIF

       IF( prtflg2 .AND. (histflg2 .OR. statflg) ) THEN
         WRITE(10, '(/10X,A)' ) 'ADDITIONAL OUTPUT FILES: '
         IF( histflg2 ) WRITE(10,'(15X,A)') CRONFILE
         IF( statflg ) WRITE(10,'(15X,A)') MBFILE
       ENDIF

C      PLOT DATA TO PRINTER

       IF( pltflg ) CALL ceplot( title,RDATE,RTIME,gname )

C      WRITE INFORMATION TO THE PRINTER IF histflg1 IS TRUE

       IF( histflg1 ) THEN

         WRITE(3,*) formfeed
         WRITE(3,19) TITLE, RDATE, RTIME
    19   FORMAT( 7X, A70, /7X, 'Run on ', A10, ' at ', A8 )
         WRITE(3,*) linefeed
```

```
      WRITE(3,20)
20    FORMAT(7X,' TIME',3X,'CONCENTRATION',4X,'    EXPOSURE',
     .                     4X,'  MEAN CONC.',3X, 'NUM OF PUFFS')

      IF( ppmflg ) THEN
        WRITE(3,30)
30      FORMAT(7X,'(sec)',11X,'(ppm)',4X,'(g-sec/m**3)',11X,'(ppm)')
        WRITE(3,*) linefeed
      ELSE IF( ciflg ) THEN
        WRITE(3,37)
37      FORMAT(7X,'(sec)',6X,'(mCi/m**3)',2X,'(mCi-sec/m**3)',5X,
     .           '(mCi/m**3)')
        WRITE(3,*) linefeed
      ELSE
        WRITE(3,31)
31      FORMAT(7X,'(sec)',8X,'(g/m**3)',4X,'(g-sec/m**3)',8X,
     .           '(g/m**3)')
        WRITE(3,*) linefeed
      ENDIF

      DO i=1, 55
        WRITE(3,32) ttime(i), conc(i), expos(i), avconc(i), npuffs(i)
32      FORMAT( 8X, I4, 3(4X, 1PE12.2), 6X, I4 )
        IF(i .GE. count) GOTO 6001
      ENDDO

      WRITE(3,*) formfeed
      WRITE(3,19) TITLE, RDATE, RTIME
      WRITE(3,20)
      IF( ppmflg ) THEN
        WRITE(3,30)
      ELSE IF( ciflg ) THEN
        WRITE(3,37)
      ELSE
        WRITE(3,31)
      ENDIF

      DO i=56, 111
        WRITE(3,32) ttime(i), conc(i), expos(i), avconc(i), npuffs(i)
        IF(i .GE. count) GOTO 6001
      ENDDO

      WRITE(3,*) formfeed
      WRITE(3,19) TITLE, RDATE, RTIME
      WRITE(3,20)
      IF( ppmflg ) THEN
        WRITE(3,30)
      ELSE IF( ciflg ) THEN
        WRITE(3,37)
      ELSE
        WRITE(3,31)
      ENDIF
```

```
        DO i=112, 167
          WRITE(3,32) ttime(i), conc(i), expos(i), avconc(i), npuffs(i)
          IF(i .GE. count) GOTO 6001
        ENDDO

        WRITE(3,*) formfeed
        WRITE(3,19) TITLE, RDATE, RTIME
        WRITE(3,20)
        IF( ppmflg ) THEN
          WRITE(3,30)
        ELSE IF( ciflg ) THEN
          WRITE(3,37)
        ELSE
          WRITE(3,31)
        ENDIF

        DO i=168, 181
          WRITE(3,32) ttime(i), conc(i), expos(i), avconc(i), npuffs(i)
          IF(i .GE. count) GOTO 6001
        ENDDO
 6001   CONTINUE
        WRITE(3,*) formfeed
      ENDIF

C     WRITE RESULTS TO A FILE IF histflg2 IS TRUE

      IF( histflg2 ) THEN
        WRITE(11,19) TITLE, RDATE, RTIME
        WRITE(11,20)
        IF( ppmflg ) THEN
          WRITE(11,30)
        ELSE IF( ciflg ) THEN
          WRITE(11,37)
        ELSE
          WRITE(11,31)
        ENDIF
        DO i = 1,count
          WRITE(11,32) ttime(i), conc(i), expos(i), avconc(i), npuffs(i)
        ENDDO
      ENDIF

 9000 CONTINUE

      IF( prtflg1 )  CLOSE(3)
      IF( prtflg2 )  CLOSE(10)
      IF( histflg2 ) CLOSE(11)
      IF( statflg )  CLOSE(12)

      WRITE(*,'(//10X,A,/10X,A\)')
     .  'EXTRAN RUN COMPLETE. DO YOU WISH TO REVISE THE ENVIRONMENTAL',
     .  'CONDITIONS AND RUN ANOTHER CASE? Y OR N? '
```

```
      READ(*,'(A)') yn
      IF( UC( yn ) .EQ. 'Y' ) THEN

        run_count = run_count + 1

C..... RESET ARRAYS TO ZERO

        WRITE(*,'(/10X,A\)') 'RESETTING ARRAYS'

        DO i = 1,181
          conc(i) = 0.0
          avconc(i) = 0.0
          expos(i) = 0.0
          ttime(i) = 0
          npuffs(i) = 0
        ENDDO

        DO i = 1,500
          puffm(i) = 0.0
          puffage(i) = 0.0
          pufftvy(i) = 0.0
          pufftvz(i) = 0.0
          puffnsigy(i) = 0.0
          puffnsigz(i) = 0.0
          pufftsigy(i) = 0.0
          pufftsigz(i) = 0.0
        ENDDO

        exposure = 0.0
        start = 0
        tank_mass = init_tank_mass
        rel_rate = init_rel_rate
        rel_mass = 0.0
        pool_mass = 0.0
        pool_temp = 0.0
        pool_vol = 0.0
        pool_radius = 0.0
        pool_area = 0.0
        pool_thick = 0.0
        net_swrad = 0.0
        net_lwrad = 0.0
        lw_in = 0.0
        lw_out = 0.0
        air_conv = 0.0
        grnd_cond = 0.0
        net_flux = 0.0

C..... START DATA REVISION

        CALL RINPUT( title, RDATE, RTIME, gname )
        GOTO 1000
```

```
ELSE IF( UC( yn ) .NE. 'N' ) THEN
  GOTO 9000
ENDIF

END
```

```
      SUBROUTINE CEPLOT( title, RDATE, RTIME, gname )

C*********************************************************************
C
C     CEPLOT                                  EXTRAN  Version 1.2
C     Plotting Subroutine
C
C     J. V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created:  11/88 from EVPLOT written 8/85 for Extreme Wind Analysis
C     Updated:  10/90
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     EXTRAN
C
C*********************************************************************

      INCLUDE 'OPTIONS.INC'
      INCLUDE 'PRINT.INC'

      INTEGER CMAG, EMAG,IN,IP

      CHARACTER*1 LPI8(2), LPI6(2), GRAPH(132,66),
     +            LFEED, FFEED, PRCOND, PRNORM, GNAME1(20),
     +            GTITLE(70)
      CHARACTER*1 LEGEND1(40), LEGEND2(40), LEGEND3(40)
      CHARACTER*2 L8,L6
      CHARACTER*3 TEMP3
      CHARACTER*4 TEMP4
      CHARACTER*6 YAXIS1
      CHARACTER*8 RTIME, YAXIS3
      CHARACTER*10 RDATE
      CHARACTER*13 YAXIS2
      CHARACTER*20 GNAME
      CHARACTER*29 DATTIM
      CHARACTER*34 XAXIS
      CHARACTER*40 LGND1, LGND2, LGND3
      CHARACTER*70 TITLE

      EQUIVALENCE (L8,LPI8(1)),(L6,LPI6(1))
      EQUIVALENCE (GRAPH(40,51),GTITLE(1)), (GRAPH(40,54),GNAME1(1))
      EQUIVALENCE (GRAPH(40,56),LEGEND1(1)), (GRAPH(40,57),LEGEND2(1)),
     +            (GRAPH(40,58),LEGEND3(1))
```

```
      DO J = 1,66
        DO I = 1,132
          GRAPH(I,J) = ' '
        ENDDO
      ENDDO

      DO I = 1,20
        GNAME1(I) = ' '
      ENDDO

      IF( ppmflg ) THEN
        LGND1 = 'C = CONCENTRATION (ppm)                '
        LGND2 = 'A = AVERAGE CONCENTRATION (ppm)        '
      ELSE IF( ciflg ) THEN
        LGND1 = 'C = CONCENTRATION (mCi/m**3)           '
        LGND2 = 'A = AVERAGE CONCENTRATION (mCi/m**3)   '
      ELSE
        LGND1 = 'C = CONCENTRATION (g/m**3)             '
        LGND2 = 'A = AVERAGE CONCENTRATION (g/m**3)     '
      ENDIF
      IF( ciflg ) THEN
        LGND3 = 'E = TOTAL EXPOSURE [(mCi-sec)/m**3]    '
      ELSE
        LGND3 = 'E = TOTAL EXPOSURE [(g-sec)/m**3]      '
      ENDIF

C     SET PRINTER CONTROLS

      LFEED = CHAR(10)
      FFEED = CHAR(12)

C     CONDENSED PRINT

      PRCOND = CHAR(15)

C     NORMAL PRINT

      PRNORM = CHAR(18)

C     6 LINES/INCH

      LPI6(1) = CHAR(27)
      LPI6(2) = '2'

C     8 LINES /INCH

      LPI8(1) = CHAR(27)
      LPI8(2) = '0'

C     SET BORDERS FOR PLOT
```

```
C     HORIZONTAL BORDERS
      DO 100  N = 22,112
        K = MOD( N-22,10 )
        IF( K .NE. 0 ) THEN
          GRAPH(N,1)  = '.'
          GRAPH(N,61) = '.'
        ELSE
          GRAPH(N,1)  = '+'
          GRAPH(N,2)  = '+'
          GRAPH(N,60) = '+'
          GRAPH(N,61) = '+'
        ENDIF
  100 CONTINUE

C     VERTICAL BORDERS
      DO 101  N = 1,61
        K = MOD( N-1,15 )
        IF( K .NE. 0 ) THEN
          GRAPH(22,N) =  '-'
          GRAPH(112,N) = '-'
        ELSE
          GRAPH(22,N) =  '+'
          GRAPH(23,N) =  '+'
          GRAPH(111,N) = '+'
          GRAPH(112,N) = '+'
        ENDIF
  101 CONTINUE
      GRAPH(22,2)  = '+'
      GRAPH(22,60) = '+'
      GRAPH(112,2)  = '+'
      GRAPH(112,60) = '+'

C     ADD TIME SCALE

      IF( count .GT. 91 ) THEN
        ISTEP = 2
      ELSE
        ISTEP = 1
      ENDIF
      K1 = 10 * ISTEP + 1
      K2 = 90 * ISTEP + 1
      KSTEP = 10 * ISTEP

      GRAPH(22,63) = '0'
      DO 105  K = K1,K2,KSTEP
        WRITE( TEMP4(1:4), '(I4)') TTIME(K)
        If( ISTEP .EQ. 1) THEN
          I1 = K + 18
        ELSE
          I1 = K / ISTEP + 19
        ENDIF
        DO 104  II = 1,4
          GRAPH(I1+II,63) = TEMP4(II:II)
```

```
  104   CONTINUE
  105 CONTINUE

C     DETERMINE RANGES FOR CONCENTRATION AND EXPOSURE SCALES

      CMAX = -99.0
      EMAX = -99.0
      DO 110   I = 1,count
        CMAX = AMAX1(conc(I),CMAX)
        EMAX = AMAX1(expos(I),EMAX)
  110 CONTINUE

      IF( CMAX .LT. 1. ) THEN
        ICMIN = IFIX( ALOG10(CMAX) ) - 4
      ELSE
        ICMIN = IFIX( ALOG10(CMAX) + 1.) - 4
      ENDIF
      IF( EMAX .LT. 1. ) THEN
        IEMIN = IFIX( ALOG10(EMAX) ) - 4
      ELSE
        IEMIN = IFIX( ALOG10(EMAX) + 1.) - 4
      ENDIF

C     PLACE SCALE VALUES IN CHARACTER STRINGS

      K = 4
      DO 120   J = 1,61,15
        CMAG = ICMIN + K
        EMAG = IEMIN + K
        WRITE( TEMP3(1:3), '(I3)' ) CMAG
        DO 118  II = 1,3
          GRAPH(17+II,J) = TEMP3(II:II)
  118   CONTINUE
        WRITE( TEMP3(1:3), '(I3)' ) EMAG
        DO 119 II = 1,3
          GRAPH(113+II,J) = TEMP3(II:II)
  119   CONTINUE
        K = K - 1
  120 CONTINUE

C     COMPUTE NORMALIZED PLOTTING POSITIONS AND PLACE SYMBOLS IN GRAPH

      DO 200  I= 1,count,ISTEP

        IF( ISTEP .EQ. 1 ) THEN
          IP = 21 + I
        ELSE
          IP = 21 + (I+1)/2
        ENDIF

        IF( avconc(I) .GT. 0.0 ) THEN
          ALGAC = ALOG10( avconc(I) )
          JA = 61 - INT( 15.0 * ( ALGAC - ICMIN ) )
```

```
          IF( JA .GE. 1 .AND. JA .LE. 61 ) GRAPH(IP,JA) = 'A'
        ENDIF

        IF( expos(I) .GT. 0.0 ) THEN
          ALOGE = ALOG10( expos(I) )
          JE = 61 - INT( 15.0 * ( ALOGE - IEMIN ) )
          IF( JE .GE. 1 .AND. JE .LE. 61 ) GRAPH(IP,JE) = 'E'
        ENDIF

        IF( conc(I) .GT. 0.0 ) THEN
          ALOGC = ALOG10( conc(I) )
          JC = 61 - INT( 15.0 * (ALOGC - ICMIN ) )
          IF( JC .GE. 1 .AND. JC .LE. 61 ) GRAPH(IP,JC) = 'C'
        ENDIF

  200 CONTINUE

C     ADD AXIS LABELS, TITLE AND LEGEND

      DATTIM = 'RUN ON            AT        '
      WRITE( DATTIM(8:17),'(A10)') RDATE
      WRITE( DATTIM(22:29),'(A8)') RTIME

      XAXIS = 'TIME AFTER ARRIVAL AT INTAKE (sec)'
      YAXIS1 = 'LOG OF'
      YAXIS2 = 'CONCENTRATION'
      YAXIS3 = 'EXPOSURE'

      IN = 1
      DO 205 II = 1,70
        GTITLE(II) = TITLE(II:II)
        IF( II .LE. 6 ) THEN
          GRAPH(7+II,30) = YAXIS1(II:II)
          GRAPH(119+II,30) = YAXIS1(II:II)
        ENDIF
        IF( II .LE. 8 ) GRAPH(119+II,32) = YAXIS3(II:II)
        IF( II .LE. 13 ) GRAPH(4+II,32) = YAXIS2(II:II)
        IF( II .LE. 20 ) THEN
            GNAME1(II) = GNAME(II:II)
        ENDIF
        IF( II .LE. 29 ) GRAPH(39+II,52) = DATTIM(II:II)
        IF( II .LE. 34 ) GRAPH(49+II,65) = XAXIS(II:II)
        IF( II .LE. 40 ) THEN
          LEGEND1(II) = LGND1(II:II)
          LEGEND2(II) = LGND2(II:II)
          LEGEND3(II) = LGND3(II:II)
        ENDIF
  205 CONTINUE

C     PREPARE FOR PLOTTING DATA

      WRITE(3,*) FFEED,L8,PRCOND
      WRITE(3,*) LFEED,LFEED,LFEED,LFEED,LFEED,LFEED,LFEED,LFEED
```

```
C     PLOT GRAPH

      DO 301  J = 1,66
        WRITE(3,300) (GRAPH(I,J), I=1,132)
  300   FORMAT(15X,132A1)
  301 CONTINUE

C     RESTORE PRINTER

      WRITE(3,*) L6, PRNORM

      RETURN
      END
```

```
      SUBROUTINE CHIT(time, time_one, chimin)

C*********************************************************************
C
C     CHIT                                                EXTRAN  Version 1.2
C     Computes Puff Concentration
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 10/88
C     Updated: 10/90
C
C     Description: Computes concentration within the puffs.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     EXTRAN
C
C*********************************************************************

      INCLUDE 'ENVIRON.INC'
      INCLUDE 'PRINT.INC'
      INCLUDE 'PUFF.INC'
      INCLUDE 'SCENARIO.INC'

      REAL*4   chi_cc, f_of_y, g_of_h, radius, chimin,
     .         meanchi, chi, leadedge, twopi, chiprint,
     .         meanchiprint, exposprint, center, trailedge

      INTEGER*2 i, time, time_one, pstart

C....... COMPUTE CONCENTRATION, EXPOSURE, AND MEAN CONCENTRATION

      twopi = 2.0 * 3.14159
      chi   = 0.0
      pstart = MAX0(1,start)
      DO 1000 i=pstart, numpuffs
        center = puffage(i) * ubar
        radius = 4.24 * pufftsigy(i)
        leadedge =  center + radius
        trailedge = center - radius

        IF(leadedge .GE. intake_dist)  THEN

          chi_cc  = puffm(i) / (twopi**1.5
     .              * pufftsigy(i)**2 * pufftsigz(i) + pri * vent_flow )
          f_of_y  = EXP(-.5 * ((intake_dist -
     .              (puffage(i) * ubar)) / pufftsigy(i))**2)
```

```
          g_of_h = EXP(-.5 * ((rel_height - intake_height)/
     .                          pufftsigz(i))**2) +
     .                          EXP(-.5 * ((rel_height + intake_height)/
     .                          pufftsigz(i))**2)

          chi = chi + (chi_cc * f_of_y * g_of_h)

          IF( time_one .EQ. 0 ) THEN
            IF( chi .GE. chimin ) THEN
              time_one = time
            ELSE
              RETURN
            ENDIF
          ENDIF

        ENDIF

        IF( trailedge .GT. intake_dist ) start = start + 1

 1000 CONTINUE

      exposure   = exposure + (chi * delta_time)
      meanchi = exposure / (time - time_one + delta_time)

C..... CONVERT FROM KG/M**3 TO GM/M**3

      chiprint     = chi * 1000.0
      exposprint   = exposure * 1000.0
      meanchiprint = meanchi * 1000.0

      IF((time_one .GT. 0) .AND. (count.LE.181))THEN
        count          = count + 1
        npuffs(count)  = numpuffs
        ttime(count)   = time - time_one
        conc(count)    = chiprint
        expos(count)   = exposprint
        avconc(count)  = meanchiprint
      ENDIF

      RETURN
      END
```

```
      SUBROUTINE DIFCOEF

C*********************************************************************
C
C     DIFCOEF                                          EXTRAN  Version 1.2
C     Computes Molecular Diffusion Coefficient
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 10/89
C     Updated: 10/90
C
C     Description:  DIFCOEF computes molecular diffusion coefficient
C             in estimating evaporation rate when a diffusion
C                   coefficient is not provided in model input.  The
C                   Diffusion coefficient is computed at the liquid
C                   boiling point for liquids with boiling points
C                   below 0C and at 0C for other liquids.  The method
C                   is based on kinetic theory as described in Bird,
C                   Stewart and Lightfoot 1960.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     INPUTEF
C
C*********************************************************************

      INCLUDE 'EFFLUENT.INC'
      INCLUDE 'ENVIRON.INC'

      REAL   mwair, ekair, r0air, ekgas, r0gas, r012, tref3,
     .         ek12, kte, colint

      DATA mwair / 29.87 /, ekair / 97.0 /, r0air / 3.617 /

C     COMPUTE THE COLLISION DIAMETER
C     BS&L Equations 1.4-15 and 16.4-15

      r0gas = 1.166 * ( molec_wt / src_density )**(1./3.)

      r012 = (r0air + r0gas) / 2.

C     COMPUTE THE epsilon/k RATIO FOR THE MATERIAL
C     BS&L Equations 1.4-14 and 16.4-16

      ekgas = 1.15 * (boil_point + 273.16)

      ek12 = SQRT( ekair * ekgas )
```

```
C     ESTIMATE THE COLLISION INTEGRAL FROM THE epsilon/k RATIO AND
C     REFERENCE TEMPERATURE.  THE VALUES IN B S AND L'S TABLE B-2
C     ARE REPRESENTED BY POWER FUNCTIONS FOR THREE RANGES OF
C     kT/epsilon.

      tref3 = tref**3

      kte =  tref/ek12

      IF( kte .LE. 1.6 ) THEN
        colint = 1.471 * kte**(-0.4926)
      ELSE IF( kte .LE. 5.0 ) THEN
        colint = 1.3351 * kte**(-0.2863)
      ELSE
        colint = 1.0946 * kte**(-0.1629)
      ENDIF

C     COMPUTE DIFFUSION COEFFICIENT
C     B S & L Equation 16.4-13

      diff_coef = 0.0018583 * SQRT( tref3 * (1.0/mwair + 1.0/molec_wt) )
     .            / ( (air_press/760) * r012**2 * colint)

      RETURN
      END
```

```
      SUBROUTINE INPUT(title,RDATE,RTIME,gname)

C*********************************************************************
C
C     INPUT                                              EXTRAN  Version 1.2
C     Prompts User For Input Data
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 11/88
C     Updated: 10/90
C
C     Description: Prompts user for input data into the WAKE model.
C                  There are four types of input data:
C                  1)   output control options
C                  2)   release and receptor geometry
C                  3)   source and effluent characteristics
C                  4)   environmental data.
C
C     Relationship to other modules:
C
C         Makes calls to:  INPUTE, INPUTEF, INPUTO, INPUTS
C
C         Called from:     EXTRAN
C
C*********************************************************************

      INCLUDE 'EFFLUENT.INC'
      INCLUDE 'ENVIRON.INC'
      INCLUDE 'OPTIONS.INC'
      INCLUDE 'SCENARIO.INC'

      REAL*4 air_dens_stp, std_temp, std_press, log_sat_press, ugcnst,
     .       pi, mol_vol

      CHARACTER formfeed, UC, yn
      CHARACTER*8 RTIME
      CHARACTER*10 RDATE
      CHARACTER*20 gname
      CHARACTER*70 title

      formfeed = CHAR(12)
      air_dens_stp = 1.29
      std_temp = 273.16
      std_press = 760.
      ugcnst = 8.3144
      pi = 3.14159
```

```
      WRITE(*,1)
    1 FORMAT(/' Program Title:   EXTRAN VERSION 1.2'//
     +  ' Developed For:   U.S. Nuclear Regulatory Commission'/
     +  '                 Office of Nuclear Regulatory Research'/
     +  '                 Division of Reactor Accident Analysis'//
     +  ' Date:           October 1990'//
     +  ' NRC Contact(s): C. Ferrell       Phone: (FTS) 492 3944'/
     +  ' Code Developer: J. V. Ramsdell   Phone: (509) 376-8626'/
     +  '                                         (FTS) 444-8626'//
     +  ' Code Documentation: '/
     +  '                 EXTRAN: A Computer Code For Estimating'
     +  '                 Concentrations Of Toxic Substances At'
     +  '                 Control Room Air Intakes'
     +  '                 NUREG/CR-5656 '////
     +  ' The program was prepared for an agency of the United States',
     +  ' Government. Neither'/' the United States Government nor any',
     +  ' agency thereof, nor any of their'/' employees, makes any',
     +  ' warranty, expressed or implied, or assumes any legal'/
     +  ' liability or responsibilities for any third party''s use,',
     +  ' or the results of such'/' use, of any portion of this',
     +  ' program or represents that its use by such third'/' party',
     +  ' would not infringe privately owned rights. '// )

      PAUSE ' PRESS ENTER TO CONTINUE '

C     GET TITLE FOR RUN

      WRITE(*,'(/A)') ' ENTER TITLE FOR OUTPUT -- 70 CHARACTER MAXIMUM'
      READ(*,'(A)') title

C     GET INPUT OPTIONS

      CALL INPUTO

C     PROMPT USER FOR GEOMETRY OF INTAKE AND RELEASE HEIGHT

      WRITE(*,'(//A)') ' SCENARIO:'

  100 CONTINUE
  105 WRITE(*,'(/A,/A,/A,/A,/A,/A\)')
     .                 ' Available Release Scenarios ',
     .                 '     1 = Liquid Tank Burst ',
     .                 '     2 = Liquid Tank Leak ',
     .                 '     3 = Gas Tank Burst',
     .                 '     4 = Gas Tank Leak',
     .                 ' Enter Scenario Number; 1 Through 4 : '
      READ(*,'(I2)',ERR=105) rel_type
      IF( (rel_type .LT. 1 ) .OR. (rel_type .GT. 4) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Enter 1 Through 4 '
        GOTO 105
      ELSE
        WRITE(*,'(BN,I4)')        rel_type
      ENDIF
```

```
  106 WRITE(*,'(///A\)' )  '   The Scenario Selected is: '
      IF( rel_type .EQ. 1) THEN
        WRITE(*,'(30X,A)') ' Liquid Tank Burst '
      ELSE IF( rel_type .EQ. 2) THEN
        WRITE(*,'(30X,A)') ' Liquid Tank Leak'
      ELSE IF( rel_type .EQ. 3) THEN
        WRITE(*,'(30X,A)') ' Gas Tank Burst '
      ELSE IF( rel_type .EQ. 4) THEN
        WRITE(*,'(30X,A)') ' Gas Tank Leak'
      ELSE
        WRITE(*,'(//A,/A,/A//)') ' ************************************',
     .                           '  Scenario Type Not Properly Defined',
     .                           ' ************************************'

        GOTO 105
      ENDIF

      IF( ciflg .AND. (rel_type .LE. 2) ) WRITE(*,'(//A/A/A/A//)')
     .    ' ************************************************** ',
     .    '                      ... WARNING ...               ',
     .    ' A Liquid Release in Ci will not give correct results ',
     .    ' ************************************************** '

      WRITE(*,'(A\)' ) '   IS SCENARIO CORRECT, Y or N? '
      READ (*,'(A)',ERR=106)  yn
      IF( (UC(yn) .NE. 'Y') ) GOTO 100

      CALL INPUTS

C..... PROMPT USER FOR ENVIRONMENTAL CONDITIONS

      CALL INPUTE( rel_type )

C..... PROMPT USER FOR EFFLUENT CHARACTERISTICS

      CALL INPUTEF( rel_type,gname)

C     OPEN OUTPUT FILES

      IF( prtflg1 .OR. pltflg .OR. histflg1 )
     .     OPEN(UNIT=3, FILE='LPT1',STATUS='UNKNOWN')
      IF( prtflg2 ) OPEN( UNIT=10,FILE=PRTFILE,STATUS='NEW' )
      IF( histflg2 ) OPEN( UNIT=11,FILE=CRONFILE,STATUS='NEW' )
      IF( statflg ) OPEN( UNIT=12,FILE=MBFILE,STATUS='NEW' )

C..... LIST INPUT DATA IN PRIMARY OUTPUT

      IF( prtflg1 .OR. pltflg .OR. histflg1 ) THEN
        WRITE(3,2)
    2   FORMAT(/10X,'Program Title:   EXTRAN VERSION 1.2'//
     +  10X,'Developed For:   U.S. Nuclear Regulatory Commission'/
     +  10X,'                 Office of Nuclear Regulatory Research'/
     +  10X,'                 Division of Reactor Accident Analysis'//
```

```
     +  10X,'Date:            October 1990'//
     +  10X,'NRC Contact(s):  C. Ferrell        Phone: (FTS) 492 3944'/
     +  10X,'Code Developer:  J. V. Ramsdell    Phone: (509) 376-8626'/
     +  10X,'                                         (FTS) 444-8626'//
     +  10X,'Code Documentation: ' /
     +  10X,'                  EXTRAN: A Computer Code For Estimating'
     +  10X,'                  Concentrations Of Toxic Substances At'
     +  10X,'                  Control Room Air Intakes'
     +  10X,'                  NUREG/CR-5656'////
     +  10X,'The program was prepared for an agency of the United ',
     +  'States',/10X,'Government.  Neither the United States ',
     +  'Government nor any',/10x,'agency thereof, nor any of their ',
     +  'employees, makes any',/10x,'warranty, expressed or implied, ',
     +  'or assumes any legal'/10x,'liability or responsibilities for ',
     +  'any third party''s use,'/10x,'or the results of such use, of ',
     +  'any portion of this program',/10x,'or represents that its use',
     +  ' by such third party would not',/10x,'infringe privately ',
     +  'owned rights. '// )

        WRITE(3,3) title
    3   FORMAT(/11X,A70)
        WRITE(3,4) RDATE, RTIME
    4   FORMAT(/10X,' RUN DATE = ',A10,'  RUN TIME = ',A8)
      ENDIF

      IF( prtflg1 ) THEN
        WRITE(3,*) formfeed
        WRITE(3,3) title
        WRITE(3,4) RDATE, RTIME

        WRITE(3,5)
    5   FORMAT(/10X,' SCENARIO:')
        IF( rel_type .EQ. 1 ) THEN
          WRITE(3,6)
    6     FORMAT(10X,' Release Type            =  Liquid Tank Burst')
        ELSE IF( rel_type .EQ. 2 ) THEN
          WRITE(3,7)
    7     FORMAT(10X,' Release Type            =  Liquid Tank Leak')
        ELSE IF( rel_type .EQ. 3 ) THEN
          WRITE(3,8)
    8     FORMAT(10X,' Release Type            =  Gas Tank Burst')
        ELSE IF( rel_type .EQ. 4 ) THEN
          WRITE(3,9)
    9     FORMAT(10X,' Release Type            =  Gas Tank Leak')
        ELSE
          STOP ' !!!!!!  RELEASE TYPE ILL DEFINED !!!!!!! '
        ENDIF

        IF( ciflg ) THEN
          WRITE(3,10) tank_mass
   10       FORMAT(10X,' Initial Activity    (Ci) = ',F10.0)
          IF( (rel_type .EQ. 2) .OR. (rel_type .EQ. 4) ) THEN
            WRITE(3,11) rel_rate
```

```
11        FORMAT(10X,' Release Rate    (Ci/sec) = ',1PE16.2)
        ENDIF
      ELSE
        WRITE(3,35) tank_mass
35      FORMAT(10X,' Initial Mass        (kg) = ',F10.0)
        IF( (rel_type .EQ. 2) .OR. (rel_type .EQ. 4) ) THEN
          WRITE(3,36) rel_rate
36        FORMAT(10X,' Release Rate    (kg/sec) = ',1PE16.2)
        ENDIF
      ENDIF

      WRITE(3,12) rel_height
12      FORMAT(10X,' Release Height       (m) = ',F11.1)
      WRITE(3,13) tank_temp
13      FORMAT(10X,' Storage Temperature  (C) = ',F11.1)

      IF( rel_type .LT. 3 ) THEN
        WRITE(3,14) max_pool_rad
14      FORMAT(10X,' Maximum Pool Radius  (m) = ',F11.1)
      ENDIF

      WRITE(3,15) intake_dist
15      FORMAT(10X,' Intake Distance      (m) = ',F10.0)

      WRITE(3,16) intake_height
16      FORMAT(10X,' Intake Height        (m) = ',F11.1)

      WRITE(3,17) area
17      FORMAT(10X,' Building Area     (m**2) = ',F10.0)

      IF( rel_type .EQ. 4 ) THEN
        WRITE(3,18) vent_flow
18      FORMAT(10X,' Vent Flow       (m**3/s) = ',F12.2)
      ENDIF

      WRITE(3,19)
19    FORMAT(//10X,' ENVIRONMENTAL CONDITIONS:')
      WRITE(3,20) ubar
20      FORMAT(10X,' Wind Speed                (m/sec) = ',F10.1)
      WRITE(3,21) stab
21      FORMAT(10X,' Atmospheric Stability Class       = ',I8)
      WRITE(3,22) air_temp
22      FORMAT(10X,' Air Temperature               (C) = ',F10.1)
      WRITE(3,23) air_press
23      FORMAT(10X,' Atmospheric Pressure      (mm Hg) = ',F10.1)
      IF( rel_type .LT. 3 ) THEN
        WRITE(3,24) sol_rad
24      FORMAT(10X,' Solar Radiation      (watts/m**2) = ',F10.1)
        iccover = ccover * 10
        WRITE(3,25) iccover
25      FORMAT(10X,' Cloud Cover              (tenths) = ',I8)
        WRITE(3,26) earth_temp
26      FORMAT(10X,' Ground Temperature            (C) = ',F10.1)
```

```
      ENDIF

      WRITE(3,27)
27    FORMAT(//10X,' EFFLUENT CHARACTERISTICS:')
      WRITE(3,28) gname
28      FORMAT(10X,' Material Released                  = ',5X,A)
      WRITE(3,29) molec_wt
29      FORMAT(10X,' Molecular Weight        (gm/mole) = ',F10.1)
      IF( rel_type .LT. 3 ) THEN
        WRITE(3,30) boil_point
30      FORMAT(10X,' Initial Boiling Point         (C) = ',F10.1)
        WRITE(3,31) cp
31      FORMAT(10X,' Heat Capacity           (j/gm-C) = ',F12.3)
        WRITE(3,32) hv
32      FORMAT(10X,' Heat of Vapor.            (j/gm) = ',F10.1)
        WRITE(3,33) src_density
33      FORMAT(10X,' Specific Gravity                 = ',F12.3)
        WRITE(3,34) diff_coef
34      FORMAT(10X,' Diffusion Coef.       (cm**2/sec) = ',F12.3)
      ENDIF
    ENDIF

    IF( prtflg2 ) THEN
      WRITE(10,2)

      WRITE(10,3) title
      WRITE(10,4) RDATE, RTIME

      WRITE(10,5)
      IF( rel_type .EQ. 1 ) THEN
        WRITE(10,6)
      ELSE IF( rel_type .EQ. 2 ) THEN
        WRITE(10,7)
      ELSE IF( rel_type .EQ. 3 ) THEN
        WRITE(10,8)
      ELSE IF( rel_type .EQ. 4 ) THEN
        WRITE(10,9)
      ELSE
        STOP ' !!!!!!  RELEASE TYPE ILL DEFINED !!!!!!! '
      ENDIF

      IF( ciflg ) THEN
        WRITE(10,10) tank_mass
        IF( (rel_type .EQ. 2) .OR. (rel_type .EQ. 4) ) THEN
          WRITE(10,11) rel_rate
        ENDIF
      ELSE
        WRITE(10,35) tank_mass
        IF( (rel_type .EQ. 2) .OR. (rel_type .EQ. 4) ) THEN
          WRITE(10,36) rel_rate
        ENDIF
      ENDIF
```

```
        WRITE(10,12) rel_height
        WRITE(10,13) tank_temp
        IF( rel_type .LT. 3 ) THEN
          WRITE(10,14) max_pool_rad
        ENDIF
        WRITE(10,15) intake_dist
        WRITE(10,16) intake_height
        WRITE(10,17) area
        IF( rel_type .EQ. 4 ) THEN
          WRITE(10,18) vent_flow
        ENDIF

        WRITE(10,19)
        WRITE(10,20) ubar
        WRITE(10,21) stab
        WRITE(10,22) air_temp
        WRITE(10,23) air_press
        IF( rel_type .LT. 3 ) THEN
          WRITE(10,24) sol_rad
          iccover = ccover * 10
          WRITE(10,25) iccover
          WRITE(10,26) earth_temp
        ENDIF

        WRITE(10,27)
        WRITE(10,28) gname
        WRITE(10,29) molec_wt
        IF( rel_type .LT. 3 ) THEN
          WRITE(10,30) boil_point
          WRITE(10,31) cp
          WRITE(10,32) hv
          WRITE(10,33) src_density
          WRITE(10,34) diff_coef
        ENDIF
      ENDIF

C     CORRECT FOR DEPARTURE FROM STANDARD CONDITIONS

C     Compute molar volume for conversion to ppm

      mol_vol =  22.414E-3  * ( std_press/air_press ) *
     .                   ( (air_temp + 273.16)/std_temp)

      IF( rel_type .LE. 2 ) THEN

C     Compute air_density at air_temp and air_press

        air_density = air_dens_stp * (std_temp/(air_temp+273.16))
     .                             * (air_press/std_press)

C     Compute initial saturation vapor pressure

        IF( boil_point .GT. tank_temp ) THEN
```

```
            pconst = ( molec_wt * hv )
     .           / ( ugcnst * (boil_point+273.16) )
            log_sat_press = pconst
     .           * ( 1.0 - (boil_point+273.16) / (tank_temp+273.16) )
            sat_press = air_press * EXP( log_sat_press )
          ELSE
            sat_press = air_press
          ENDIF
        ENDIF

C       COMPUTE FACTOR TO CONVERT g/m**3 TO ppm

        ppmconv = mol_vol * 1.0E6 / molec_wt

C       CONVERT FROM CGS TO MKS

        cp = cp * 1.0E3
        hv = hv * 1.0E3
        sat_press = sat_press / 1.0E3
        src_density = src_density * 1.0E3
        diff_coef = diff_coef / 1.0E4
        air_press = air_press / 1.0E3

C       SET MIXING LAYER

        mix_depth = 1000.

C       SET MAXIMUM POOL RADIUS TO DEFAULT IF RADIUS NOT SUPPLIED BY USER
C       AND COMPUTE MAXIMUM POOL AREA

        IF( max_pool_rad .EQ. 0.0 ) max_pool_rad = 9999.9
        max_pool_area =  pi * max_pool_rad**2

        RETURN
        END
```

```
      SUBROUTINE INPUTE( rel_type )

C*************************************************************************
C
C     INPUTE                                                EXTRAN  Version 1.2
C     Prompts User For Environmental Data
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 10/89 from INPUT
C     Updated: 10/90
C
C     Description: Prompts user for environmental data input depending
C                  on release type.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     INPUT
C
C*************************************************************************

      INCLUDE 'ENVIRON.INC'

      INTEGER*2 NN, NOPT, rel_type, iccover

      CHARACTER UC, yn

      NOPT = 0
      NN = 4

      WRITE(*,'(///A)') ' ENVIRONMENTAL CONDITIONS:'

  100 CONTINUE
  105 WRITE(*,'(/A\)') ' Enter Wind Speed (m/s) (e.g., 5.0) : '
      READ(*,'(F15.0)',ERR=105) ubar
      IF( (ubar .LT. 0.5) .OR. (ubar .GT. 20.0) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is (0.5 - 20.0)'
        GOTO 105
      ELSE
        WRITE(*,'(F15.1)')        ubar
      ENDIF
      IF( NOPT .NE. 0 ) GOTO 800

  200 CONTINUE
  205 WRITE(*,'(/A)') ' Atmospheric Stability Classes '
      WRITE(*,'(A)') ' 1 = A, 2 = B, 3 = C, 4 = D, 5 = E, 6 = F, 7 = G'
      WRITE(*,'(A\)') ' Enter Stability Class Number : '
      READ(*,'(I1)',ERR=205) stab
```

```
      IF( (stab .LT. 1) .OR. (stab .GT. 7) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is (1 - 7)'
        GOTO 205
      ELSE
        WRITE(*,'(11x,i4)')     stab
      ENDIF
      IF( NOPT .NE. 0 ) GOTO 800

  300 CONTINUE
  305 WRITE(*,'(/A,A\)') ' Enter Ambient Air Temperature (C)',
     .                   ' (e.g., 10.0) : '
      READ(*,'(F15.0)',ERR=305) air_temp
      IF( (air_temp .LT. -40.0) .OR. (air_temp .GT. 50.0) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is (-40.0 TO +50.0)'
        GOTO 305
      ELSE
        WRITE(*,'(F15.0)')        air_temp
      ENDIF
      IF( NOPT .NE. 0 ) GOTO 800

  400 CONTINUE
  405 WRITE(*,'(/A,A\)') ' Enter Atmospheric Pressure (mm Hg)',
     .                   ' (e.g., 760.0) : '
      READ(*,'(F15.0)',ERR=405) air_press
      IF( (air_press .LT. 600.0) .OR. (air_press .GT. 800.0) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is (600.0 - 800.0)'
        GOTO 405
      ELSE
        WRITE(*,'(F15.1)')        air_press
      ENDIF
      IF( NOPT .NE. 0 ) GOTO 800

      IF( rel_type .GT. 2 )  GOTO 800
      NN = 7

  500 CONTINUE
  505 WRITE(*,'(/A,A\)')
     .  ' Enter Value for Solar Radiation ',
     .  '(watts/m**2) (e.g. 500.0) : '
      READ(*,'(F15.0)',ERR=505) sol_rad
      IF( (sol_rad .LT. 0.0) .OR. (sol_rad .GT. 1200.) ) THEN
        WRITE(*,'(A,A)') ' Solar Radiation Out of Range, Range is ',
     .                   '0 to 1200. '
        GOTO 505
      ELSE
        WRITE(*,'(F15.0)')        sol_rad
      ENDIF
      IF( NOPT .NE. 0 ) GOTO 800

  600 CONTINUE
  605 WRITE(*,'(/A\)')
     .  ' Enter Cloud Cover in tenths (e.g. 4 ) '
      READ(*,'(I4)',ERR=605) iccover
```

```
      IF( (iccover .LT. 0 ) .OR. ( iccover .GT. 10 ) ) THEN
        WRITE(*,'(A,A)') ' Cloud Cover Out of Range, Range is ',
     .                   '0 to 10 '
        GOTO 605
      ELSE
        WRITE(*,'(10x,I4)')        iccover
      ENDIF
      ccover = FLOAT(iccover) / 10.
      IF( NOPT .NE. 0 ) GOTO 800

  700 CONTINUE
  705 WRITE(*,'(/A,A\)') ' Enter Temperature (C) of the Ground',
     .                   ' (e.g., Air Temp. + 10.0) : '
      READ(*,'(F15.0)',ERR=705) earth_temp
      IF( (earth_temp .LT. -40.0) .OR. (earth_temp .GT. 60.0) ) THEN
          WRITE(*,'(/A)') ' Out of Range; Range is (-40.0 to +60.0)'
          GOTO 705
      ELSE
        WRITE(*,'(F15.0)')        earth_temp
      ENDIF

      NOPT = 1

C..... REVIEW INPUT DATA AND REVISE AS NECESSARY

  800 WRITE(*,'(//A)') ' ENVIRONMENTAL CONDITIONS REVIEW:'
      WRITE(*,'(A,F10.1)')  '  1   Wind Speed                (m/sec) = ',
     .                                                          ubar
      WRITE(*,'(A,I8)')     '  2   Atmosperic Stability Class       = ',
     .                                                          stab
      WRITE(*,'(A,F10.1)')  '  3   Air Temperature              (C) = ',
     .                                                       air_temp
      WRITE(*,'(A,F10.1)')  '  4   Atmospheric Pressure     (mm Hg) = ',
     .                                                       air_press
      IF( rel_type .LT. 3 ) THEN
        WRITE(*,'(A,F10.1)')'  5   Solar Radiation     (watts/m**2) = ',
     .                                                        sol_rad
        WRITE(*,'(A,I8)')   '  6   Cloud Cover             (tenths) = ',
     .                                                        iccover
        WRITE(*,'(A,F10.1)')'  7   Ground Temperature          (C) = ',
     .                                                     earth_temp
      ENDIF

      WRITE(*,'(1X,A)') ' C   CONTINUE WITH DATA ENTRY '
      WRITE(*,'(1X,A)') ' X   EXIT PROGRAM '

  900 CONTINUE
      WRITE(*,901)
  901 FORMAT(/'  ENTER  C  TO CONTINUE IF DATA ARE CORRECT,',
     .  /'  IF DATA ARE NOT CORRECT ENTER NUMBER OF ITEM TO BE CHANGED',
     .  /'  OR, ENTER  X  TO STOP PROGRAM : '\ )

      READ(*,'(A)',ERR=800) yn
```

```
 IF( UC(yn) .EQ. 'X' )
.   STOP '  PROGRAM STOPPED AT ENVIRONMENTAL DATA ENTRY '
 IF( UC(yn) .EQ. 'C' ) RETURN
 IF( yn .EQ. '1' )  THEN
   GOTO 100
 ELSE IF( yn .EQ. '2' )  THEN
   GOTO 200
 ELSE IF( yn .EQ. '3' )  THEN
   GOTO 300
 ELSE IF( yn .EQ. '4' )  THEN
   GOTO 400
 ENDIF
 IF( rel_type .LE. 2 ) THEN
   IF( yn .EQ. '5' )  THEN
     GOTO 500
   ELSE IF( yn .EQ. '6' )  THEN
     GOTO 600
   ELSE IF( yn .EQ. '7' )  THEN
     GOTO 700
   ENDIF
 ENDIF
 WRITE(*,'(A,I2,A)' )
.    '  ENTER A NUMBER FROM 1 THRU ', NN, ', A C, OR AN X. '
 GOTO 800

 RETURN
 END
```

```
      SUBROUTINE INPUTEF( rel_type,gname )

C*********************************************************************
C
C     INPUTEF                                              EXTRAN  Version 1.2
C     Prompts User For Effluent Data
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 10/89 from INPUT
C     Updated: 10/90
C
C     Description: Prompts user for effluent characteristics.
C
C     Relationship to other modules:
C
C         Makes calls to:  DIFCOEF
C
C         Called from:     INPUT
C
C*********************************************************************

      INCLUDE 'EFFLUENT.INC'
      INCLUDE 'ENVIRON.INC'

      INTEGER*2 sel, I, NOPT, nchem, nchemlim,
     .          nn, np, nend, indx, rel_type

      REAL*4 phys_prop(6,30)

      CHARACTER UC, yn
      CHARACTER*20 gas(30),gname

      nchemlim = 30

      ugcnst = 8.3144

C     READ IN THE CHEMICAL DATA FILE

      OPEN(UNIT=10,FILE='CHEMICAL.DAT',STATUS='OLD')
      READ(10,'(15X,I5)') nchem

      IF( nchem .GT. nchemlim ) THEN
        WRITE(*,'(//A)') '   ******* CHEMICAL FILE TO LARGE **********'
        STOP
      ENDIF

      DO 10  nn = 1,nchem
        READ(10,9) gas(nn), (phys_prop(np,nn),np=1,6)
    9   FORMAT(1X, A20, 6F8.0)
```

```
 10 CONTINUE

    WRITE(*,'(/A)') ' CHEMICAL DATA FILE READ '
    CLOSE(10)

    WRITE(*,'(//A)') ' EFFLUENT CHARACTERISTICS:'

    NOPT = 0
    NN = 2
 50 CONTINUE
 55 WRITE(*,'(/A)') ' *******   MATERIAL RELEASED  ******* '
    nend = nchem/2
    DO 60  I = 1,nend
      indx = 2 * I - 1
      WRITE(*,'(2x,A,A,I2,10x,A,A,I2)')
     .          gas(indx), ' ==> ', indx, gas(indx+1), ' ==> ', indx+1
 60 CONTINUE
    IF( MOD(nchem,2) .EQ. 1 )
     .   WRITE(*,'(2x,A,A,I2)') gas(nchem), ' ==> ', nchem

    WRITE(*,'(/A\)') ' ENTER SELECTION NUMBER : '
    READ(*,'(I4)',ERR=55) sel
    IF( sel .LT. 1 .OR. sel .GT. nchem ) THEN
      WRITE(*,'(A,I2)') ' SELECTION MUST BE IN THE RANGE 1 TO ',nchem
      GOTO 55
    ELSE
      WRITE(*,'(11X,I4)')        sel
    ENDIF

100 CONTINUE
    IF( sel .NE. nchem ) THEN
      gname = gas(sel)
      molec_wt = phys_prop(1,sel)
      std_boil_pt = phys_prop(2,sel)
      cp = phys_prop(3,sel)
      hv = phys_prop(4,sel)
      src_density = phys_prop(5,sel)
      idiff_coef = phys_prop(6,sel)
      diff_coef = idiff_coef
C     Correct boiling point for atmospheric pressure

      boil_point = ( std_boil_pt+273.16 )
     .   / ( 1 + ugcnst * (std_boil_pt+273.16)
     .   * ( ALOG(760.)-ALOG(air_press) )
     .   / ( hv * molec_wt )  ) - 273.16

      IF( boil_point .LT. 0.0 ) THEN
        tref = boil_point + 273.16
      ELSE
        tref = 273.16
      ENDIF
```

```
      IF( (diff_coef .LE. 0.0) .AND. (rel_type .LT. 3) ) THEN
        WRITE(*,'(A,A,/A)') ' Diffusion coefficient in data file ',
     .                      ' is zero.  One will be computed',
     .            '  from molecular weight, boiling point and density.'
        CALL DIFCOEF
        WRITE(*, '(/A,F8.3)') ' Computed diffusion coefficient = ',
     .                         diff_coef
      ENDIF
      GOTO 800

    ELSE

      WRITE(*,'(/A,A,/A)')' See Perry and Chilton, or Handbook of ',
     .    'Chemistry and Physics',' for Physical Properties'
      WRITE(*,'(/A,/A,/A)') ' ****** WARNING ***********',
     .    ' PROGRAM WILL NOT GIVE CORRECT RESULTS IF DATA ARE NOT ',
     .    ' ENTERED IN THE PROPER UNITS !!!!!!!!!!!!! '
105   WRITE(*,'(/A,A\)') ' Enter Name of Substance, ',
     .                   '20 Character Maximum : '
      READ(*,'(A)',ERR=105) gname
      IF( NOPT .EQ. 1) GOTO 800
    ENDIF

200 CONTINUE
205 WRITE(*,'(/A,A\)')
     .       ' Enter Molecular Weight (grams/mole) of Substance ',
     .       ' (e.g., 70.9) : '
    READ(*,'(F15.0)',ERR=205) molec_wt
    IF( molec_wt .LE. 1.0 ) THEN
      WRITE(*,'(A)') ' Molecular Weight Must Be > 1.0 '
      GOTO 205
    ELSE
      WRITE(*,'(F15.1)')        molec_wt
    ENDIF
    IF( NOPT .EQ. 1 ) GOTO 800

    IF( rel_type .GT. 2 ) GOTO 800
    NN = 7

300 CONTINUE
305 WRITE(*,'(/A,A\)') ' Enter Boiling Point (C) of Liquid ',
     .       ' at 760 mm Hg (e.g., -34.1) '
    READ(*,'(F15.0)',ERR=305) std_boil_pt
    IF( (std_boil_pt .LT. -270.) .OR. ( std_boil_pt .GT. 200.) ) THEN
      WRITE(*,'(A)') ' Boiling Point Range is -270C to +200C '
      GOTO 305
    ELSE
      WRITE(*,'(F15.1)')        std_boil_pt
    ENDIF

C   Correct boiling point for atmospheric pressure

    boil_point = ( std_boil_pt+273.16 )
```

```
     .   / ( 1 + ugcnst * (std_boil_pt+273.16)
     .   * ( ALOG(760.)-ALOG(air_press) )
     .   / ( hv * molec_wt )  ) - 273.16

      IF( boil_point .LT. 0.0 ) THEN
        tref = boil_point + 273.16
      ELSE
        tref = 273.16
      ENDIF

      IF( NOPT .EQ. 1 ) GOTO 800

  400 CONTINUE
  405 WRITE(*,'(/A,A\)')
     .      ' Enter Heat Capacity (j/gm-C) of Liquid Phase',
     .      ' (e.g., 0.946) : '
      READ(*,'(F15.0)',ERR=405) cp
      IF( cp .LE. 0.0 ) THEN
        WRITE(*,'(A)') ' Heat Capacity Must be > 0.0 '
        GOTO 405
      ELSE
        WRITE(*,'(1PE15.2)')        cp
      ENDIF
      IF( NOPT .EQ. 1 ) GOTO 800

  500 CONTINUE
  505 WRITE(*,'(/A,A\)')
     .    ' Enter Heat of Vaporization (j/gm) of Liquid Phase ',
     .    '(e.g., 288.) : '
      READ(*,'(F15.0)',ERR=505) hv
      IF( hv .LE. 0.0 ) THEN
        WRITE(*,'(A)') ' Heat of Vaporization Must be > 0.0 '
        GOTO 505
      ELSE
        WRITE(*,'(1PE15.2)')        hv
      ENDIF
      IF( NOPT .EQ. 1 ) GOTO 800

  600 CONTINUE
  605 WRITE(*,'(/A,A\)')
     .      ' Enter Density or Specific Gravity of Liquid Phase ',
     .      '(e.g., 1.57) : '
      READ(*,'(F15.0)',ERR=605) src_density
      IF( (src_density .LT. 0.1) .OR. (src_density .GT. 4.0) ) THEN
        WRITE(*,'(A)') ' Specific Gravity Range is 0.1 to 4.0 '
        GOTO 605
      ELSE
        WRITE(*,'(1PE15.2)')        src_density
      ENDIF
      IF( NOPT .EQ. 1 ) GOTO 800

  700 CONTINUE
  705 WRITE(*,'(/A,A\)')
```

```
     .        ' Enter Diffusion Coefficient (cm**2/sec) of Source',
     .        ' (e.g., .0792) : '
      READ(*,'(F15.0)',ERR=705) idiff_coef
      diff_coef = idiff_coef
      IF( diff_coef .GT. 0.4) THEN
        WRITE(*,'(A)')  ' Diffusion coefficient too large, ',
     .                  'maximum is 0.4'
        GOTO 705
      ELSE IF( diff_coef .LE. 0.0 ) THEN
        WRITE(*,'(A)') ' Diffusion coefficient will be computed '

        CALL DIFCOEF
        WRITE(*, '(F15.3)') diff_coef
      ELSE
        WRITE(*,'(F15.3)')  diff_coef
      ENDIF

      NOPT = 1

C..... REVIEW INPUT DATA AND REVISE AS NECESSARY

  800 CONTINUE
      WRITE(*,'(///A)') ' EFFLUENT CHARACTERISTICS REVIEW:'

      WRITE(*,'(A,A)')       '  1   Material Released              =   ',
     .                                                              gname
      WRITE(*,'(A,1PE10.2)') '  2   Molecular Weight (gm/mole)     = ',
     .                                                             molec_wt
      IF(rel_type .GT. 2) GOTO 850

      WRITE(*,'(A,1PE10.2)') '  3   Initial Boiling Point  (C)     = ',
     .                                                          std_boil_pt
      WRITE(*,'(A,1PE10.2)') '  4   Heat Capacity     (j/gm-C)    = ',  cp
      WRITE(*,'(A,1PE10.2)') '  5   Heat of Vapor.      (j/gm)    = ',  hv
      WRITE(*,'(A,1PE10.2)') '  6   Specific Gravity              = ',
     .                                                         src_density
      WRITE(*,'(A,1PE10.2)') '  7   Diffusion Coef. (cm**2/sec)   = ',
     .                                                           diff_coef
  850 CONTINUE
      WRITE(*,'(1X,A)')  ' C   BEGIN CALCULATIONS '
      WRITE(*,'(1X,A)')  ' X   EXIT PROGRAM '

  855 WRITE(*,856)
  856 FORMAT(/'  ENTER  C  TO CONTINUE IF DATA ARE CORRECT,',
     .  /'  IF DATA ARE NOT CORRECT ENTER NUMBER OF ITEM TO BE CHANGED',
     .  /'  OR, ENTER  X  TO STOP PROGRAM : '\ )
      READ(*,'(a)',ERR=855) yn

      IF( UC(yn) .EQ. 'X' )
     .  STOP '  PROGRAM STOPPED AT EFFLUENT DATA ENTRY '
      IF( UC(yn) .EQ. 'C' ) RETURN
```

```
IF( yn .EQ. '1' ) THEN
  GOTO 50
ELSE IF( yn .EQ. '2' ) THEN
      GOTO 200
ENDIF
IF( rel_type .LE. 2 ) THEN
  IF( yn .EQ. '3' ) THEN
    GOTO 300
  ELSE IF( yn .EQ. '4' ) THEN
    GOTO 400
  ELSE IF( yn .EQ. '5' ) THEN
    GOTO 500
  ELSE IF( yn .EQ. '6' ) THEN
    GOTO 600
  ELSE IF( yn .EQ. '7' ) THEN
    GOTO 700
  ENDIF
ENDIF

WRITE(*,'(A,I2,A)' )
.     '   ENTER NUMBER FROM 1 THRU ', NN, ', A C, OR AN X. '
GOTO 800

RETURN
END
```

```
      SUBROUTINE INPUTO

C*********************************************************************
C
C     INPUTO                                              EXTRAN  Version 1.2
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 7/90
C     Updated: 10/90
C
C     Description: Allows user control of model output options
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     INPUT
C
C*********************************************************************

      INCLUDE 'OPTIONS.INC'
      INCLUDE 'SCENARIO.INC'

      CHARACTER UC, yn, yn1

      WRITE(*,'(//A)')  ' OUTPUT OPTION SELECTION '

      NOPT = 0

  100 CONTINUE
  105 WRITE(*,'(/A,/A\)')
     .    ' Output concentrations may be expressed as g/m**3, ppm, or ',
     .    ' mCi/m**3.  Do you want concentrations in ppm? Y or N '
      READ(*,'(A)',ERR=105) yn

      IF( UC(yn) .EQ. 'Y' ) THEN
        ppmflg = .true.
      ELSE IF( UC(yn) .EQ. 'N' ) THEN
        ppmflg = .false.
  106   WRITE( *, '(/A\)' )
     .    ' Do you want concentrations in mCi/m**3?  Y or N '
        READ(*,'(A)' ) yn1
        IF( UC(yn1) .EQ. 'Y' ) THEN
          ciflg = .true.
        ELSE IF( UC(yn1) .EQ. 'N' ) THEN
          ciflg = .false.
        ELSE
          GOTO 106
        ENDIF
```

```
      ELSE
        GOTO 105
      ENDIF
      IF( NOPT .NE. 0 ) GOTO 900

  200 CONTINUE
  205 WRITE(*,'(/A\)')
     .   ' Do you want primary output to be sent to a printer? Y or N '
      READ(*,'(A)',ERR=205) yn

      IF( UC(yn) .EQ. 'Y' ) THEN
        prtflg1 = .true.

  255   CONTINUE
        WRITE(*,'(/A\)')
     .    ' Do you want a copy of the output sent to a file?  Y or N '
        READ(*,'(A)',ERR=255) yn1
        IF( UC( yn1 ) .EQ. 'Y' ) THEN
          prtflg2 = .true.
        ELSE IF( UC( yn1 ) .EQ. 'N' ) THEN
          prtflg2 = .false.
        ELSE
          GOTO 255
        ENDIF
      ELSE IF( UC(yn) .EQ. 'N' ) THEN
        prtflg1 = .false.
        prtflg2 = .true.
      ELSE
        GOTO 205
      ENDIF
      IF( NOPT .NE. 0 ) GOTO 900

  300 CONTINUE
  305 WRITE(*,'(/A,/A\)')
     .   ' Concentrations may be plotted.  Do you want a plot of',
     .   ' the output?  Y or N '
      READ(*,'(A)',ERR=305) yn

      IF( UC(yn) .EQ. 'Y' ) THEN
        pltflg = .true.
      ELSE IF( UC(yn) .EQ. 'N' ) THEN
        pltflg = .false.
      ELSE
        GOTO 305
      ENDIF
      IF( NOPT .NE. 0 ) GOTO 900

  400 CONTINUE
  405 WRITE(*,'(/A/A\)')
     .  ' A concentration chronology may be printed.  Do you want a',
     .  ' printer output of the concentration chronology?  Y or N '
      READ(*,'(A)',ERR=405) yn
```

```
      IF( UC(yn) .EQ. 'Y' ) THEN
        histflg1 = .true.
      ELSE IF( UC(yn) .EQ. 'N' ) THEN
        histflg1 = .false.
      ELSE
        GOTO 405
      ENDIF
      IF( NOPT .NE. 0 ) GOTO 900

  500 CONTINUE
  505 WRITE(*,'(/A,A\)')
     .    ' Do you want a file with the concentration chronology?',
     .    '  Y or N '
      READ(*,'(A)',ERR=505) yn

      IF( UC(yn) .EQ. 'Y' ) THEN
        histflg2 = .true.
      ELSE IF( UC(yn) .EQ. 'N' ) THEN
        histflg2 = .false.
      ELSE
        GOTO 505
      ENDIF
      IF( NOPT .NE. 0 ) GOTO 900

  600 CONTINUE
  605 WRITE(*,'(/A/A\)')
     .    ' Do you want a file with details of the release and pool',
     .    ' status and the energy balance of the pool?  Y or N '
      READ(*,'(A)',ERR=605) yn

      IF( UC(yn) .EQ. 'Y' ) THEN
        statflg = .true.
      ELSE IF( UC(yn) .EQ. 'N' ) THEN
        statflg = .false.
      ELSE
        GOTO 605
      ENDIF

C..... REVIEW INPUT DATA AND REVISE AS NECESSARY

      NOPT = 1
  900 CONTINUE

      WRITE(*,'(//A)') ' OUTPUT OPTION REVIEW:'
      IF( ppmflg ) THEN
        WRITE(*,'(1X,A)') ' 1   Concentration output in ppm '
      ELSE IF( ciflg ) THEN
        WRITE(*,'(1X,A)') ' 1   Concentration output in mCi/m**3'
      ELSE
        WRITE(*,'(1X,A)') ' 1   Concentration output in g/m**3 '
      ENDIF
```

```
      IF( prtflg1 ) THEN
        WRITE(*,'(1X,A)') ' 2   Print primary results '
        IF( prtflg2 ) THEN
          WRITE(*,'(6X,A)')  'Create file with primary results '
        ELSE
          WRITE(*,'(6X,A)')  'Don''t create file with primary results '
        ENDIF
      ELSE
        WRITE(*,'(1X,A)') ' 2   Don''t print primary results '
        WRITE(*,'(6X,A)')  'Create file with primary results '
      ENDIF

      IF( pltflg ) THEN
        WRITE(*,'(1X,A)') ' 3   Plot results '
      ELSE
        WRITE(*,'(1X,A)') ' 3   Don''t plot results '
      ENDIF

      IF( histflg1 ) THEN
        WRITE(*,'(1X,A)')
     .     ' 4   Write concentration chronology to printer '
      ELSE
        WRITE(*,'(1X,A)')
     .     ' 4   Don''t write concentration chronology to printer '
      ENDIF

      IF( histflg2 ) THEN
        WRITE(*,'(1X,A)')
     .     ' 5   Create file with concentration chronology '
      ELSE
        WRITE(*,'(1X,A)')
     .     ' 5   Don''t create file with concentration chronology '
      ENDIF

      IF( statflg ) THEN
        WRITE(*,'(1X,A)')
     .    ' 6   Create file with release, pool and energy balance data'
      ELSE
        WRITE(*,'(1X,A,A)')
     .    ' 6   Don''t create file with release, pool and energy balance'
     .    ,' data'
      ENDIF
      WRITE(*,'(1X,A)') ' C   CONTINUE WITH DATA ENTRY '
      WRITE(*,'(1X,A)') ' X   EXIT PROGRAM '

      WRITE(*,906)
  906 FORMAT(/'  ENTER  C  TO CONTINUE IF DATA ARE CORRECT,',
     .  /'  IF DATA ARE NOT CORRECT ENTER NUMBER OF ITEM TO BE CHANGED',
     .  /'  OR, ENTER  X  TO STOP PROGRAM : '\ )
      READ(*,'(A)',ERR=900) yn

      IF(  UC(yn) .EQ. 'X'  )
     .      STOP '  PROGRAM STOPPED AT OUTPUT OPTION ENTRY '
```

```
IF( UC(yn) .EQ. 'C' )   RETURN

IF( yn .EQ. '1' ) THEN
  GOTO 100
ELSE IF( yn .EQ. '2' ) THEN
  GOTO 200
ELSE IF( yn .EQ. '3' ) THEN
  GOTO 300
ELSE IF( yn .EQ. '4' ) THEN
  GOTO 400
ELSE IF( yn .EQ. '5' ) THEN
  GOTO 500
ELSE IF( yn .EQ. '6' ) THEN
  GOTO 600
ELSE
  WRITE(*,'(/A )')
.    ' ENTER A NUMBER FROM 1 THRU 6, A C, OR AN X.'
  GOTO 900
ENDIF

RETURN
END
```

```
      SUBROUTINE INPUTS

C*********************************************************************
C
C     INPUTS                                              EXTRAN  Version 1.2
C     Prompts User For Source Description
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 10/89 from INPUT
C     Updated: 10/90
C
C
C     Description: Prompts user for release and receptor geometry
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     INPUT
C
C*********************************************************************

      INCLUDE 'OPTIONS.INC'
      INCLUDE 'SCENARIO.INC'

      INTEGER*2 NOPT, NN
      CHARACTER UC, yn

      WRITE(*,'(//A)')  ' RELEASE AND RECEPTOR GEOMETRY: '

      NOPT = 0
      NN = 1

  100 CONTINUE

  105 IF( ciflg ) THEN
        WRITE(*,'(/A,A\)') ' Enter Total Activity (Ci) in Tank ',
     .                     '(e.g., 1000.0) : '
        READ(*,'(F15.0)',ERR=105) init_tank_mass
        tank_mass = init_tank_mass
        IF( ( tank_mass .LE. 0.0 ) .OR. ( tank_mass .GT. 1.0E6) ) THEN
          WRITE(*,'(/A)') ' Out of Range; Range is 0.0 to 1,000,000 Ci '
          GOTO 105
        ELSE
          WRITE(*,'(G15.3)')        tank_mass
        ENDIF
      ELSE
        WRITE(*,'(/A,A\)') ' Enter Total Mass (kg) in Tank ',
     .                     '(e.g., 1000.0) : '
```

```
      READ(*,'(F15.0)',ERR=105) init_tank_mass
      tank_mass = init_tank_mass
      IF( ( tank_mass .LE. 1.0 ) .OR. ( tank_mass .GT. 1.0E6) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is 1.0 to 1,000,000 kg '
        GOTO 105
      ELSE
        WRITE(*,'(F15.1)')        tank_mass
      ENDIF
    ENDIF

    IF( NOPT .NE. 0 ) GOTO 900

200 CONTINUE
    IF( NOPT .EQ. 0 ) NN = NN + 1
    IF( rel_type .EQ. 2 .OR. rel_type .EQ. 4 ) THEN

205   IF( ciflg ) THEN
        WRITE(*,'(/A,A\)') ' Enter Release Rate in Ci/sec ',
     .                     '(e.g., 10.0) : '
        READ(*,'(F15.0)',ERR=205) init_rel_rate
        rel_rate = init_rel_rate
        IF( rel_rate .GE. tank_mass/2 ) THEN
          WRITE(*,'(/A,A)') ' Release Rate Too Large -- ',
     .                      'Catastrophic Failure '
          GOTO 205
        ELSE IF( rel_rate .LE. 0.0 ) THEN
          WRITE(*,'(/A\)') ' Release Rate must be greater than 0.0 '
          GOTO 205
        ELSE
          WRITE(*,'(G15.3)')          rel_rate
        ENDIF
      ELSE
        WRITE(*,'(/A,A\)') ' Enter Release Rate in kg/sec ',
     .                     '(e.g., 100.0) : '
        READ(*,'(F15.0)',ERR=205) init_rel_rate
        rel_rate = init_rel_rate
        IF( rel_rate .GE. tank_mass/2 ) THEN
          WRITE(*,'(/A,A)') ' Release Rate Too Large -- ',
     .                      'Catastrophic Failure '
          GOTO 205
        ELSE IF( rel_rate .LE. 0.0 ) THEN
          WRITE(*,'(/A\)') ' Release Rate must be greater than 0.0 '
          GOTO 205
        ELSE
          WRITE(*,'(F15.3)')          rel_rate
        ENDIF
      ENDIF

    ENDIF
    IF( NOPT .NE. 0) GOTO 900

300 CONTINUE
    IF( NOPT .EQ. 0 ) NN = NN+1
```

```
305 WRITE(*,'(/A,A\)') ' Enter Height (meters) of Release ',
   .                   '(e.g., 0.0) : '
    READ(*,'(F15.0)',ERR=305) rel_height
    IF((rel_height .LT. 0.0) .OR. (rel_height .GT. 100.0)) THEN
      WRITE(*,'(/A)') ' Out of Range; Range is (0.0 - 100.0)'
      GOTO 305
    ELSE
      WRITE(*,'(F15.1)')        rel_height
    ENDIF
    IF( NOPT .NE. 0) GOTO 900

400 CONTINUE
    IF( NOPT .EQ. 0 ) NN = NN + 1
405 WRITE(*,'(/A,A\)')' Enter Storage Temperature in Tank in Degrees',
   .                  ' C  (e.g., 15.0) : '
    READ(*,'(F15.0)',ERR=405) tank_temp
    IF( ( tank_temp .LT. -40.0 ) .OR. ( tank_temp .GT. 50.0 ) ) THEN
      WRITE(*,'(/A)') ' Out of Range; Range is -40.0 to 50.0 C'
      GOTO 405
    ELSE
      WRITE(*,'(F15.1)')        tank_temp
    ENDIF
    IF( NOPT .NE. 0) GOTO 900

500 CONTINUE
    IF( rel_type .LT. 3 ) THEN
      IF( NOPT .EQ.0 ) NN= NN + 1
505   WRITE(*,'(/A,A\)') ' Enter Maximum Radius of Pool',
   .        ' in Meters, if appropriate (e.g., 15.0) : '
      READ(*,'(F15.0)',ERR=505) max_pool_rad
      IF( (max_pool_rad .LT. 0.0) .OR. (max_pool_rad .GT. 50.0) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is 0 to 50 m'
        GOTO 505
      ELSE
        WRITE(*,'(F15.1)')        max_pool_rad
      ENDIF
      IF( NOPT .NE. 0) GOTO 900
    ENDIF

600 CONTINUE
    IF( NOPT .EQ. 0 ) NN = NN + 1
605 WRITE(*,'(/A,A\)') ' Enter Distance (meters) to Intake ',
   .                   '(e.g., 100.0) : '
    READ(*,'(F15.0)',ERR=605) intake_dist
    IF((intake_dist .LT. 5.0) .OR. (intake_dist .GT. 1999.0) ) THEN
      WRITE(*,'(/A)') ' Out of Range; Range is (5.0 - 1999.)'
      GOTO 605
    ELSE
      WRITE(*,'(F15.0)')        intake_dist
    ENDIF
    IF( NOPT .NE. 0) GOTO 900
```

```
  700 CONTINUE
      IF( NOPT .EQ. 0 ) NN = NN + 1
  705 WRITE(*,'(/A,A\)') ' Enter Height (meters) of Intake ',
     .                   '(e.g., 25.0) : '
      READ(*,'(F15.0)',ERR=705) intake_height
      IF((intake_height.LT.0.0) .OR. (intake_height.GT.100.0)) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is (0.0 - 100.0)'
        GOTO 705
      ELSE
        WRITE(*,'(F15.1)')        intake_height
      ENDIF
      IF( NOPT .NE. 0) GOTO 900

  800 CONTINUE
      IF( NOPT .EQ. 0 ) NN = NN + 1
  805 WRITE(*,'(/A,A\)') ' Enter Area (m**2) of Building, ',
     .             'if release is in wake (e.g., 100.0) : '
      READ(*,'(F15.0)',ERR=805) area
      IF( (area .LT. 0.0) .OR. (area .GT. 4000.0) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is (0.0 - 4000.0)'
        GOTO 805
      ELSE
        WRITE(*,'(F15.0)')        area
      ENDIF
      IF( NOPT .NE. 0) GOTO 900

  850 CONTINUE
      vent_flow = 0.0
      IF( rel_type .EQ. 4 ) THEN
        IF( NOPT .EQ. 0 ) NN = NN + 1
  855   WRITE(*,'(/A,A,/A\)')
     .   ' If Release is thru a short stack or vent enter flow ',
     .   '(m**3/sec).  ',' Enter zero if unknown or zero. (e.g. 10.0) '
        READ(*,'(F15.0)',ERR=855) vent_flow
        IF( (vent_flow .LT. 0.0) .OR. (vent_flow .GT. 50.0) ) THEN
          WRITE(*,'(/A)') ' Out of Range; Range is (0.0 - 50.0)'
          GOTO 855
        ELSE
          WRITE(*,'(F15.2)')        vent_flow
        ENDIF
      ENDIF

C..... REVIEW INPUT DATA AND REVISE AS NECESSARY

      NOPT = 1

  900 CONTINUE

      WRITE(*,'(//A)') ' SCENARIO DATA REVIEW:'
      IF( rel_type .EQ. 1 ) THEN
        WRITE(*,'(/A,A)')     '      Release Type              = ',
     .                        '  Liquid Tank Burst'
        IF( ciflg ) THEN
```

```
    WRITE(*,'(A,G16.2)')' 1   Initial Activity    (Ci) = ',
.                        tank_mass
   ELSE
    WRITE(*,'(A,F11.1)')' 1   Initial Mass        (kg) = ',
.                        tank_mass
   ENDIF
   NN = 2

 ELSE IF( rel_type .EQ. 2 ) THEN
   WRITE(*,'(7A,A)')     '     Release Type             = ',
.                        '    Liquid Leak'
   IF( ciflg ) THEN
    WRITE(*,'(A,G16.2)')' 1   Initial Activity    (Ci) =      ',
.                        tank_mass
    WRITE(*,'(A,G15.2)')' 2   Release Rate    (Ci/sec) = ',
.                        rel_rate
   ELSE
    WRITE(*,'(A,F11.1)')' 1   Initial Mass        (kg) = ',
.                        tank_mass
    WRITE(*,'(A,F12.2)')' 2   Release Rate    (kg/sec) = ',
.                        rel_rate
   ENDIF
   NN = 3

 ELSE IF( rel_type .EQ. 3 ) THEN
   WRITE(*,'(7A,A)')     '     Release Type             = ',
.                        ' Gas Tank Burst'
   IF( ciflg ) THEN
    WRITE(*,'(A,G16.2)')' 1   Initial Activity    (Ci) = ',
.                        tank_mass
   ELSE
    WRITE(*,'(A,F11.1)')' 1   Initial Mass        (kg) = ',
.                        tank_mass
   ENDIF
   NN = 2

 ELSE IF( rel_type .EQ. 4 ) THEN
   WRITE(*,'(7A,A)')     '     Release Type             = ',
.                        '    Gas Leak'
   IF( ciflg ) THEN
    WRITE(*,'(A,G16.2)')' 1   Initial Activity    (Ci) = ',
.                        tank_mass
    WRITE(*,'(A,G15.2)')' 2   Release Rate    (Ci/sec) = ',
.                        rel_rate
   ELSE
    WRITE(*,'(A,F11.1)')' 1   Initial Mass        (kg) = ',
.                        tank_mass
    WRITE(*,'(A,F11.1)')' 2   Release Rate             = ',
.                        rel_rate
   ENDIF
   NN = 3
```

```
      ENDIF

      WRITE(*,'(1X,I2,A,F10.0)') NN,'   Release Height       (m) = ',
     .                           rel_height
      NN = NN + 1
      WRITE(*,'(1X,I2,A,F10.0)') NN,'   Storage Temperature  (C) = ',
     .                           tank_temp
      NN = NN + 1

      IF( rel_type .LT. 3 ) THEN
        WRITE(*,'(1X,I2,A,F10.0)') NN,'   Maximum Pool Radius  (m) = ',
     .                           max_pool_rad
        NN = NN + 1
      ENDIF

      WRITE(*,'(1X,I2,A,F10.0)') NN,'   Intake Distance      (m) = ',
     .                           intake_dist
      NN = NN + 1
      WRITE(*,'(1X,I2,A,F10.0)') NN,'   Intake Height        (m) = ',
     .                           intake_height
      NN = NN + 1
      WRITE(*,'(1X,I2,A,F10.0)') NN,'   Building Area     (m**2) = ',
     .                            area

      IF( rel_type .EQ. 4) THEN
        NN = NN + 1
        WRITE(*,'(1X,I2,A,F12.2)')NN,'   Vent Flow       (m**3/s) = ',
     .                            vent_flow
      ENDIF

      WRITE(*,'(1X,A)')  ' C   CONTINUE WITH DATA ENTRY '
      WRITE(*,'(1X,A)')  ' X   EXIT PROGRAM '

      WRITE(*,906)
  906 FORMAT(/'  ENTER  C  TO CONTINUE IF DATA ARE CORRECT,',
     .  /'  IF DATA ARE NOT CORRECT ENTER NUMBER OF ITEM TO BE CHANGED',
     .  /'  OR, ENTER  X  TO STOP PROGRAM : '\ )
      READ(*,'(A)',ERR=900) yn

      IF( UC(yn) .EQ. 'X' )
     .   STOP '  PROGRAM STOPPED AT RELEASE AND RECEPTOR DATA ENTRY '
      IF( UC(yn) .EQ. 'C' ) RETURN

      IF( rel_type .EQ. 1) THEN
        IF( yn .EQ. '1' ) THEN
          GOTO 100
        ELSE IF( yn .EQ. '2' ) THEN
          GOTO 300
        ELSE IF( yn .EQ. '3' ) THEN
          GOTO 400
        ELSE IF( yn .EQ. '4' ) THEN
          GOTO 500
```

```
  ELSE IF( yn .EQ. '5' ) THEN
    GOTO 600
  ELSE IF( yn .EQ. '6' ) THEN
    GOTO 700
  ELSE IF( yn .EQ. '7' ) THEN
    GOTO 800
  ENDIF
ELSE IF( rel_type .EQ. 2) THEN
  IF( yn .EQ. '1' ) THEN
    GOTO 100
  ELSE IF( yn .EQ. '2' ) THEN
    GOTO 200
  ELSE IF( yn .EQ. '3' ) THEN
    GOTO 300
  ELSE IF( yn .EQ. '4' ) THEN
    GOTO 400
  ELSE IF( yn .EQ. '5' ) THEN
    GOTO 500
  ELSE IF( yn .EQ. '6' ) THEN
    GOTO 600
  ELSE IF( yn .EQ. '7' ) THEN
    GOTO 700
  ELSE IF( yn .EQ. '8' ) THEN
    GOTO 800
  ENDIF
ELSE IF( rel_type .EQ. 3 ) THEN
  IF( yn .EQ. '1' ) THEN
    GOTO 100
  ELSE IF( yn .EQ. '2' ) THEN
    GOTO 300
  ELSE IF( yn .EQ. '3' ) THEN
    GOTO 400
  ELSE IF( yn .EQ. '4' ) THEN
    GOTO 600
  ELSE IF( yn .EQ. '5' ) THEN
    GOTO 700
  ELSE IF( yn .EQ. '6' ) THEN
    GOTO 800
  ENDIF
ELSE IF( rel_type .EQ. 4 ) THEN
  IF( yn .EQ. '1' ) THEN
    GOTO 100
  ELSE IF( yn .EQ. '2' ) THEN
    GOTO 200
  ELSE IF( yn .EQ. '3' ) THEN
    GOTO 300
  ELSE IF( yn .EQ. '4' ) THEN
    GOTO 400
  ELSE IF( yn .EQ. '5' ) THEN
    GOTO 600
  ELSE IF( yn .EQ. '6' ) THEN
    GOTO 700
```

```
  ELSE IF( yn .EQ. '7' ) THEN
    GOTO 800
  ELSE IF( yn .EQ. '8' ) THEN
    GOTO 850
  ENDIF
ENDIF

WRITE(*,'(A,I2,A)' )
.   '   ENTER A NUMBER FROM 1 THRU ', NN, ', A C, OR AN X.'
GOTO 900

RETURN
END
```

```
      SUBROUTINE MODELPAR( chimin )

C*************************************************************************
C
C     MODELPAR                                              EXTRAN  Version 1.2
C     Computes Model Parameters
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 11/88
C     Updated: 10/90
C
C     Description: Computes model parameters such as puff release interval,
C                  delta time, time one
C
C     Relationship to other modules:
C
C         Makes calls to:  NSIG, WSIG
C
C         Called from:     EXTRAN
C
C*************************************************************************

      INCLUDE 'ENVIRON.INC'
      INCLUDE 'OPTIONS.INC'
      INCLUDE 'SCENARIO.INC'

      REAL*4  sigmay, sigmaz, tty, wsigy, wsigz, tsigy, chimin, pi,
     .        tsigz

C....... COMPUTE PUFF RELEASE INTERVAL

      pi = 3.14159
      sigmay = 0.0
      sigmaz = 0.0

      CALL nsig(intake_dist, sigmay, sigmaz)
      tty = intake_dist / ubar
      wsigy = 0.0
      wsigz = 0.0
      IF( area .gt. 0.0 ) CALL wsig(tty,tty,wsigy,wsigz)
      tsigy = SQRT( sigmay**2 + wsigy**2 )
      tsigz = SQRT( sigmaz**2 + wsigz**2 )

      chimin = 1.0E-4 * 1.0 / ( (2 * pi)** (1.5) * tsigy**2 * tsigz )

      pri = INT(2.0 * tsigy / ubar)

        IF(pri .LE. 1) THEN
          pri = 1
```

```
         ELSE IF( pri .GT. 10 ) THEN
           IF( intake_dist .GE. 1000.0 ) THEN
             pri = MINO(30,pri)
           ELSE
             pri = 10
           ENDIF
      ELSE
           pri = pri - MOD(pri,2)
      ENDIF

C..... COMPUTE DELTA TIME

      delta_time = MAXO(pri/2,1)
      IF(delta_time .EQ. 0) delta_time = 1

C..... WRITE MODEL PARAMETER INFORMATION

         IF( prtflg1 ) THEN
        WRITE(3,'(//10X,A)')  ' MODEL PARAMETERS:'
        WRITE(3,'(10X,A,I4)')
     .       ' Puff Release Interval           (sec) = ', pri
        WRITE(3,'(10X,A,I4)')
     .       ' Time Step                       (sec) = ', delta_time
         ENDIF
         IF( prtflg2 ) THEN
        WRITE(10,'(//10X,A)')  ' MODEL PARAMETERS:'
        WRITE(10,'(10X,A,I4)')
     .       ' Puff Release Interval           (sec) = ', pri
        WRITE(10,'(10X,A,I4)')
     .       ' Time Step                       (sec) = ', delta_time
         ENDIF

      RETURN
      END
```

```
      SUBROUTINE NSIG( DSMTRI, SIGMAY, SIGMAZ )

C*********************************************************************
C
C     NSIG                                                EXTRAN  Version 1.2
C     Diffusion Curves As Used In XOQDOQ, PAVAN, MESOI, and MESORAD
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/83
C     Updated: 10/90
C
C     Description: Computes new diffusion coefficients given the
C                  last values, atmospheric stability, mixing layer
C                  thickness and distance moved.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     EXTRAN, MODELPAR
C
C*********************************************************************

      INCLUDE 'ENVIRON.INC'

      REAL*4 AY(7), AZ(7,3), BZ(7,3), CZ(7,3), XVY, XEY, SZLIM,
     .       XVZ, XEZ

      DATA AY/ 0.3658, 0.2751,0.2089,0.1471,0.1046,0.0722,0.0481/
      DATA AZ/ 0.192,  0.156, 0.116, 0.079, 0.063, 0.053, 0.032,
     +         0.00066,0.0382,0.113, 0.222, 0.211, 0.086, 0.052,
     +         0.00024,0.055, 0.113, 1.26,  6.73, 18.05, 10.83 /
      DATA BZ/ 0.936, 0.922, 0.905, 0.881, 0.871, 0.814, 0.814,
     +         1.941,  1.149, 0.911, 0.725, 0.678, 0.74,  0.74,
     +         2.094,  1.098, 0.911, 0.516, 0.305, 0.18,  0.18 /
      DATA CZ/ 0.0,    0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
     +         9.27,   3.3,   0.0,  -1.7,  -1.3,  -0.35, -0.21,
     +        -9.6,    2.0,   0.0, -13.,  -34.0, -48.6, -29.2  /

      XVY = ( SIGMAY/AY(STAB) )**(1.0/0.9031)
      XEY = XVY + DSMTRI
      SIGMAY = AY(STAB) * XEY**0.9031

C  **  SIGMA Z COMPUTATIONS
C  **  CHECK INITIAL SIGMA Z SIZE AGAINST MAXIMUM

      SZLIM = 0.8 * mix_depth
      IF ( SIGMAZ .GE. SZLIM)  RETURN
```

```
C  ** COMPUTE VIRTUAL DISTANCE

       XVZ = ( SIGMAZ/AZ(STAB,1) )**(1.0 / BZ(STAB,1))
      IF (SIGMAZ.GT.1.0) THEN
       IF ((XVZ+DSMTRI).GT.100.0.AND.SIGMAZ.GT.CZ(STAB,2)) THEN
         XVZ= ( ( SIGMAZ - CZ(STAB,2) ) / AZ(STAB,2) )**
     +          ( 1.0/ BZ(STAB,2) )
         IF ((XVZ+DSMTRI).GE.1000.0.AND.SIGMAZ.GT.CZ(STAB,3)) THEN
           XVZ= ( ( SIGMAZ - CZ(STAB,3) ) / AZ(STAB,3) )**
     +          ( 1.0  / BZ(STAB,3) )
         ENDIF
       ENDIF
      ENDIF
      XEZ = XVZ + DSMTRI
       IF ( XEZ .LE. 100.0 ) THEN
         SIGMAZ = AZ(STAB,1) * XEZ ** BZ(STAB,1)
       ELSEIF ( XEZ .LE. 1000.0 ) THEN
         SIGMAZ = AZ(STAB,2) * XEZ ** BZ(STAB,2) + CZ(STAB,2)
       ELSEIF ( XEZ .GT. 1000.0 ) THEN
         SIGMAZ = AZ(STAB,3) * XEZ ** BZ(STAB,3) + CZ(STAB,3)
       ENDIF
       IF (SIGMAZ .GT. SZLIM)  SIGMAZ = SZLIM

      RETURN
      END
```

```
      SUBROUTINE PUFFINIT

C*************************************************************************
C
C     PUFFINIT                                              EXTRAN  Version 1.2
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 11/88
C     Updated: 10/90
C
C     Description: Computes initial puff dimensions and virtual times
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     EXTRAN
C
C*************************************************************************

      INCLUDE 'EFFLUENT.INC'
      INCLUDE 'ENVIRON.INC'
      INCLUDE 'POOL.INC'
      INCLUDE 'PUFF.INC'
      INCLUDE 'SCENARIO.INC'

      REAL*4  constant, ftv, ftvprime, differ, pi, ustar, tsh, tsv,
     .        c1, c2, zol(7), puff_vol, ugcnst

      INTEGER*2 kount

      DATA zol/ -1.00, -0.60, -0.33, -0.07, 0.09, 0.20, 0.50/

C     Constant in Wake Diffusion Equation

      constant = 0.5

C     Gas constant in (m^3 atm)/(kg-mole K)

      ugcnst = 8.2057E-2

      pi = 3.14159

C     COMPUTE INITIAL SIGMA Y AND Z FOR PUFF FROM FLASHING

      isigy1 = 0.0
      isigz1 = 0.0
      IF( pmass1 .GT. 0.0 ) THEN
        puff_vol = ugcnst * (pmass1/molec_wt) * (tank_temp+273.16)
```

```
     .              / (air_press / .76 )
            vapor_density = pmass1 / puff_vol
            IF( vapor_density .LT. 2.0 ) THEN
              isigy1 = puff_vol**(1./3.) / SQRT( 2 * pi )
              isigz1 = isigy1
            ELSE
              isigz1 = 1.0
              isigy1 = SQRT(puff_vol / ( 2 * pi * isigz1 ) )
            ENDIF
          ENDIF

C         COMPUTE INITIAL SIGMA Y AND SIGMA Z FOR EVAPORATION OF
C         SPILLED LIQUID

          isigy2 = 0.0
          isigz2 = 0.0
          IF( pmass2 .GT. 0.0 ) THEN
            isigy2 = SQRT( pool_area / ( 4.0 * pi ) )
            isigz2 = 0.1
          ENDIF

          z = 10.0
          znot = .1
          ustar = (.4 * ubar) / ALOG( z/znot )

C         COMPUTE TIME SCALES FOR WAKE DIFFUSION MODEL
          tsh = SQRT(area / ustar**2)
          tsv = SQRT(area) / ((2.0+zol(stab)) * ustar)

C         COMPUTE VIRTUAL TIMES FOR WAKE MODEL FOR PUFF FROM FLASHING

          IF( (isigy1**2) .GE. (constant*tsh**2) .OR.
         .    (isigz1**2) .GE. (constant*tsv**2) ) THEN
            tvy1 = -1
            tvz1 = -1
          ELSE

C         USE NEWTON-RAPHSON METHOD TO ESTIMATE VIRTUAL TIMES FOR
C         WAKE MODEL

C         INITIAL GUESSES
            tvy1 = 1
            tvz1 = 1

C         CONSTANT TERMS

            c1 = tsh * ALOG(1.0 - isigy1**2 / (constant * tsh**2) )
            c2 = tsv * ALOG(1.0 - isigz1**2 / (constant * tsv**2) )

C         ESTIMATE VIRTUAL TIME FOR HORIZONTAL DIFFUSION
            kount = 0
  110       CONTINUE
           · kount = kount + 1
```

```
        ftv = tvy1 - tsh * ALOG(1. + tvy1/tsh) + c1
        ftvprime = tvy1 / ( tsh + tvy1 )
        differ = - ftv / ftvprime
        tvy1 = tvy1 + differ

        IF( ABS( differ ) .LT. 0.01 ) GOTO 111
        IF( kount .GT. 20 ) STOP 'NO CONVERGENCE IN NEWTON'
        GOTO 110
  111   CONTINUE

C     ESTIMATE VIRTUAL TIME FOR VERTICAL DIFFUSION

        kount = 0
  120   CONTINUE
        kount = kount + 1
        ftv =  tvz1 - tsv * ALOG(1. + tvz1/tsv) + c2
        ftvprime = tvz1 / ( tsv + tvz1 )
        differ = - ftv / ftvprime
        tvz1 = tvz1 + differ
        IF( ABS( differ ) .LT. 0.01 ) GOTO 121
        IF( kount .GT. 20 ) STOP 'NO CONVERGENCE IN NEWTON'
        GOTO 120
  121   CONTINUE

C     INITIAL PUFF SIZE INCLUDED IN WAKE MODEL DELETE FROM NORMAL
C     DIFFUSION

        isigy1 = 0.0
        isigz1 = 0.0

      ENDIF

C     COMPUTE VIRTUAL TIMES FOR WAKE MODEL FOR PUFF FROM EVAPORATION

      IF( (isigy2**2) .GE. (constant*tsh**2) .OR.
     .    (isigz2**2) .GE. (constant*tsv**2) ) THEN
        tvy2 = -1
        tvz2 = -1
      ELSE

C     USE NEWTON-RAPHSON METHOD TO ESTIMATE VIRTUAL TIMES FOR
C     WAKE MODEL

C     INITIAL GUESSES
        tvy2 = 1
        tvz2 = 1

C     CONSTANT TERMS

        c1 = tsh * ALOG(1.0 - isigy2**2 / (constant * tsh**2) )
        c2 = tsv * ALOG(1.0 - isigz2**2 / (constant * tsv**2) )
```

```
C     ESTIMATE VIRTUAL TIME FOR HORIZONTAL DIFFUSION
        kount = 0
  130   CONTINUE
        kount = kount + 1
        ftv = tvy2 - tsh * ALOG(1. + tvy2/tsh) + c1
        ftvprime = tvy2 / ( tsh + tvy2 )
        differ = - ftv / ftvprime
        tvy2 = tvy2 + differ

        IF( ABS( differ ) .LT. 0.01 ) GOTO 131
        IF( kount .GT. 20 ) STOP 'NO CONVERGENCE IN NEWTON'
        GOTO 130
  131   CONTINUE

C     ESTIMATE VIRTUAL TIME FOR VERTICAL DIFFUSION

        kount = 0
  140   CONTINUE
        kount = kount + 1
        ftv =  tvz2 - tsv * ALOG(1. + tvz2/tsv) + c2
        ftvprime = tvz2 / ( tsv + tvz2 )
        differ = - ftv / ftvprime
        tvz2 = tvz2 + differ
        IF( ABS( differ ) .LT. 0.01 ) GOTO 141
        IF( kount .GT. 20 ) STOP 'NO CONVERGENCE IN NEWTON'
        GOTO 140
  141   CONTINUE

C     INITIAL PUFF SIZE INCLUDED IN WAKE MODEL DELETE FROM NORMAL
C     DIFFUSION

        isigy2 = 0.0
        isigz2 = 0.0

      ENDIF

      RETURN
      END
```

```
      SUBROUTINE PUFFMASS( pmass1,pmass2,time )

C*********************************************************************
C
C     PUFFMASS                                          EXTRAN  Version 1.2
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 10/88
C     Updated: 10/90
C
C     Description: Computes the initial mass of material in a puff
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     EXTRAN
C
C*********************************************************************

      INCLUDE 'EFFLUENT.INC'
      INCLUDE 'ENVIRON.INC'
      INCLUDE 'POOL.INC'
      INCLUDE 'SCENARIO.INC'

      REAL*4 pi, g, hc, gas_const, hd, re_num, sc_num, init_radius,
     .       char_len, visc_air, sb_const, albedo, log_sat_press,
     .       temp_chg, est_p_temp, xs_energy, add_evap,
     .       dif_temp, adj_diff_coef, pool_area_1, pool_area_2

      INTEGER*2 time

C..... INITIALIZE CONSTANTS -- ALL VALUES ARE IN MKS

      sb_const = 5.67E-8
      pi = 3.14159
      g = 9.8
      gas_const = 0.0624
      visc_air = 2.0E-5

      IF( (rel_type .EQ. 1) .OR. (rel_type .EQ. 3) ) THEN
        rel_mass = tank_mass
        tank_mass = 0.0
        rel_rate = 0.0
      ELSE IF( (rel_type .EQ. 2) .OR. (rel_type .EQ. 4) ) THEN
        rel_mass = AMINO( rel_rate * pri, tank_mass )
        tank_mass = tank_mass - rel_mass
        IF( (tank_mass .LE. 0.0) .AND. (rel_rate .GT. 0.0) ) THEN
          WRITE(*,'(A)')  '  Last Puff, Tank Empty '
```

```
          tank_mass = 0.0
          rel_rate = 0.0
        ENDIF
      ENDIF

      IF( ( rel_mass .LE. 0.0 ) .AND. (pool_mass .LE. 0.0 ) ) THEN
        pmass1 = 0.0
        pmass2 = 0.0
        RETURN
      ENDIF

      IF( rel_type .GT. 2 ) THEN
        pmass1 = rel_mass
        pmass2 = 0.0
        RETURN
      ENDIF

      IF( boil_point .LE. air_temp ) THEN

C       COMPUTE PORTION OF RELEASE THAT FLASHES, ADD REMAINDER TO
C       POOL AND COMPUTE POOL TEMPERATURE

        IF( rel_mass .GT. 0.0 ) THEN
          pmass1 = cp * rel_mass * (air_temp - boil_point) / hv
          rel_mass = rel_mass - pmass1
        ENDIF

      ENDIF

C     EVAPORATION FORCED BY VAPOR PRESSURE
C     POOL TEMPERATURE CHANGES CONTROLLED BY ENERGY BALANCE

C     COMPUTE POOL DIMENSIONS AND TEMPERATURE ASSUMING
C     UNIFORM MIXING IN POOL

      IF( boil_point .LE. air_temp ) THEN
        pool_temp = pool_temp * pool_mass + rel_mass * boil_point
      ELSE
        pool_temp = pool_temp * pool_mass + rel_mass * tank_temp
      ENDIF
      pool_mass = pool_mass + rel_mass

      IF( pool_mass .GT. 0.0 ) THEN
        pool_temp = pool_temp / pool_mass
        pool_vol = pool_mass / src_density

C       ESTIMATE POOL AREA BY 2 METHODS AND COMPARE WITH MAXIMUM
C       AREA SUPPLIED BY USER -- CHOOSE SMALLEST VALUE

        init_radius = (pool_vol / pi)**(1.0/3.0)
        pool_area_1 = pi * (init_radius**2 + ( 2.0 * (time + pri) )
     .                * SQRT( (g * pool_vol / pi) ) )
        pool_area_2 = pool_vol / 0.01
```

```
      pool_area = AMIN0( max_pool_area, pool_area_1, pool_area_2 )

      pool_radius = SQRT( pool_area / pi )
      pool_thick = pool_vol / pool_area

C     COMPUTE SATURATION VAPOR PRESSURE AT POOL TEMPERATURE

      log_sat_press = pconst *
     .      ( 1.0 - (boil_point + 273.16) / (pool_temp + 273.16) )
      sat_press = air_press * EXP( log_sat_press )

C     ADJUST DIFFUSION COEFFICIENT FOR TEMPERATURE

      dif_temp = (air_temp + pool_temp)/2 + 273.16
      adj_diff_coef = diff_coef * (dif_temp / tref)**1.5

C     COMPUTE REYNOLDS AND SCHMIDT NUMBER

      char_len = 2.0 * pool_radius
      re_num = (char_len * ubar * air_density) / visc_air
      sc_num = visc_air / (adj_diff_coef * air_density)

C     COMPUTE hd BASED FOR TURBULENT FLOW

      hd = .037 * (adj_diff_coef/char_len)
     .     * re_num**.8 * sc_num**(1.0/3.0)

C     COMPUTE MASS EVAPORATED DURING PERIOD

      pmass2 = hd * molec_wt * pool_area * pri * sat_press
     .             / (gas_const * (pool_temp + 273.16) )
      IF( pmass2 .GE. pool_mass ) THEN
        pmass2 = pool_mass
        pool_mass = 0.0
        WRITE(*,'(A)' ) '  POOL DRIED UP '
      ELSE
        pool_mass = pool_mass - pmass2

C       NET SHORT-WAVE RADIATION (watts/m**2)

        albedo = 0.1
        net_swrad = (1.0 - albedo) * sol_rad

C       NET LONG-WAVE RADIATION (watts/m**2)

        lw_in = 5.31E-13 * (air_temp + 273.16)**6 + 60 * ccover
        IF( sol_rad .GT. 100.0 ) lw_in = lw_in - 30
        lw_out = sb_const * (pool_temp + 273.16)**4
        net_lwrad = lw_in - lw_out
```

```
C       FORCED CONVECTION OF AIR OVER THE SPILL

        hc = 6.69 * ubar**0.6
        air_conv = hc * (air_temp - pool_temp)

C       CONDUCTION FROM GROUND

        grnd_cond = (1000. * (earth_temp - pool_temp))
                          / SQRT( FLOAT(time + pri) )

C       NET ENERGY FLUX TO SPILL

        net_flux =  net_swrad + net_lwrad + air_conv + grnd_cond

C       CORRECT POOL TEMPERATURE

        temp_chg = pool_area * net_flux * pri / ( cp * pool_mass )
           -  ( pmass2 / pool_mass ) * ( hv / cp )

        IF( temp_chg .LE. 0.0 ) THEN

C         POOL COOLS FROM EVAPORATION

          pool_temp = pool_temp + temp_chg

        ELSE

C         POOL WARMS FROM ENERGY INPUT

          est_p_temp = pool_temp + temp_chg

C         CHECK TO SEE IF BOILING POINT EXCEEDED AND CORRECT
C         EVAPORATION IF IT IS

          IF( est_p_temp .LE. boil_point ) THEN
            pool_temp = est_p_temp
          ELSE
            pool_temp = boil_point
            xs_energy = pool_mass * (est_p_temp - boil_point) * cp

            add_evap = xs_energy / hv

            IF( pool_mass .GT. add_evap ) THEN
              pmass2 = pmass2 + add_evap
              pool_mass = pool_mass - add_evap
            ELSE
              pmass2 = pmass2 + pool_mass
              pool_mass = 0.0
            ENDIF

          ENDIF

        ENDIF
```

```
  ENDIF
ENDIF

RETURN
END
```

```
      SUBROUTINE RELPUFF

C*************************************************************************
C
C     RELPUFF                                                EXTRAN  Version 1.2
C     Puff Release
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 10/88
C     Updated: 10/90
C
C     Description: Assigns initial characteristics to a puff at release
C                  time.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     EXTRAN
C
C*************************************************************************

      INCLUDE 'PUFF.INC'

      IF( pmass1 .GT. 0.0 ) THEN
        numpuffs = numpuffs + 1
        puffm(numpuffs)   = pmass1
        puffage(numpuffs) = 0.0
        pufftvy(numpuffs) = tvy1
        pufftvz(numpuffs) = tvz1
        puffnsigy(numpuffs) = isigy1
        puffnsigz(numpuffs) = isigz1
        pufftsigy(numpuffs) = 0.0
        pufftsigz(numpuffs) = 0.0
      ENDIF

      IF( pmass2 .GT. 0.0 ) THEN
        numpuffs = numpuffs + 1
        puffm(numpuffs)   = pmass2
        puffage(numpuffs) = 0.0
        pufftvy(numpuffs) = tvy2
        pufftvz(numpuffs) = tvz2
        puffnsigy(numpuffs) = isigy2
        puffnsigz(numpuffs) = isigz2
        pufftsigy(numpuffs) = 0.0
        pufftsigz(numpuffs) = 0.0
      ENDIF
```

```
RETURN
END
```

```
      SUBROUTINE RINPUT( title,RDATE,RTIME,gname )

C*********************************************************************
C
C     RINPUT                                              EXTRAN  Version 1.2
C     Prompts User For Input Data
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 7/90
C     Updated: 10/90
C
C     Description: Prompts user for revisions to the environmental
C                  data for multiple runs.
C
C     Relationship to other modules:
C
C         Makes calls to:  RINPUTE
C
C         Called from:     EXTRAN
C
C*********************************************************************

      INCLUDE 'EFFLUENT.INC'
      INCLUDE 'ENVIRON.INC'
      INCLUDE 'OPTIONS.INC'
      INCLUDE 'SCENARIO.INC'

      REAL*4 air_dens_stp, std_temp, std_press, log_sat_press, ugcnst,
     .       pi, mol_vol

      CHARACTER formfeed
      CHARACTER*8 RTIME
      CHARACTER*10 RDATE
      CHARACTER*20 gname
      CHARACTER*70 title

      formfeed = CHAR(12)
      air_dens_stp = 1.29
      std_temp = 273.16
      std_press = 760.
      ugcnst = 8.3144
      pi = 3.14159

C....... RESET ENVIRONMENTAL VARIABLES TO INITIAL cgs VALUES

      cp = cp * 1.0E-3
      hv = hv * 1.0E-3
      src_density = src_density * 1.0E-3
      diff_coef = diff_coef * 1.0E4
```

```
      air_press = air_press * 1.0E3

C..... PROMPT USER FOR ENVIRONMENTAL CONDITIONS

      CALL RINPUTE( rel_type )

C     RENAME OUTPUT FILES

      WRITE( PRTFILE(10:12), '(I3.3)' ) run_count
      WRITE( CRONFILE(10:12), '(I3.3)' ) run_count
      WRITE( MBFILE(10:12), '(I3.3)' ) run_count

C     OPEN OUTPUT FILES

      IF( prtflg1 .OR. pltflg .OR. histflg1 )
     .  OPEN(UNIT=3, FILE='LPT1',STATUS='UNKNOWN')
      IF( prtflg2 ) OPEN( UNIT=10,FILE=PRTFILE,STATUS='NEW' )
      IF( histflg2 ) OPEN( UNIT=11,FILE=CRONFILE,STATUS='NEW' )
      IF( statflg ) OPEN( UNIT=12,FILE=MBFILE,STATUS='NEW' )

C..... LIST INPUT DATA IN PRIMARY OUTPUT

      IF( prtflg1 .OR. pltflg .OR. histflg1 ) THEN
        WRITE(3,*) formfeed
        WRITE(3,2)
    2   FORMAT(/10X,'Program Title:   EXTRAN VERSION 1.2'//
     +  10X,'Developed For:   U.S. Nuclear Regulatory Commission'/
     +  10X,'                 Office of Nuclear Regulatory Research'/
     +  10X,'                 Division of Reactor Accident Analysis'//
     +  10X,'Date:            October 1990'//
     +  10X,'NRC Contact(s):  C. Ferrell        Phone: (FTS) 492 3944'/
     +  10X,'Code Developer:  J. V. Ramsdell    Phone: (509) 376-8626'/
     +  10X,'                                          (FTS) 444-8626'//
     +  10X,'Code Documentation: ' /
     +  10X,'                 EXTRAN: A Computer Code For Estimating'
     +  10X,'                 Concentrations Of Toxic Substances At'
     +  10X,'                 Control Room Air Intakes'
     +  10X,'                 NUREG/CR-5656'////
     +  10X,'The program was prepared for an agency of the United ',
     +  'States',/10X,'Government.  Neither the United States ',
     +  'Government nor any',/10x,'agency thereof, nor any of their ',
     +  'employees, makes any',/10x,'warranty, expressed or implied, ',
     +  'or assumes any legal'/10x,'liability or responsibilities for ',
     +  'any third party''s use,'/10x,'or the results of such use, of ',
     +  'any portion of this program',/10x,'or represents that its use',
     +  ' by such third party would not',/10x,'infringe privately ',
     +  'owned rights. '// )

        WRITE(3,3) title
    3   FORMAT(/11X,A70)
        WRITE(3,4) RDATE, RTIME
    4   FORMAT(/10X,' RUN DATE = ',A10,'  RUN TIME = ',A8)
```

```
      WRITE(3,*)  formfeed
    ENDIF
    IF( prtflg1 ) THEN
      WRITE(3,3) title
      WRITE(3,4) RDATE, RTIME

      WRITE(3,5)
 5    FORMAT(/10X,' SCENARIO:')
      IF( rel_type .EQ. 1 ) THEN
        WRITE(3,6)
 6      FORMAT(10X,' Release Type            =  Liquid Tank Burst')
      ELSE IF( rel_type .EQ. 2 ) THEN
        WRITE(3,7)
 7      FORMAT(10X,' Release Type            =  Liquid Tank Leak')
      ELSE IF( rel_type .EQ. 3 ) THEN
        WRITE(3,8)
 8      FORMAT(10X,' Release Type            =  Gas Tank Burst')
      ELSE IF( rel_type .EQ. 4 ) THEN
        WRITE(3,9)
 9      FORMAT(10X,' Release Type            =  Gas Tank Leak')
      ELSE
        STOP ' !!!!!!  RELEASE TYPE ILL DEFINED !!!!!!! '
      ENDIF

      IF( ciflg ) THEN
        WRITE(3,10) tank_mass
10      FORMAT(10X,' Initial Mass        (Ci) = ',F10.0)
        IF( (rel_type .EQ. 2) .OR. (rel_type .EQ. 4) ) THEN
          WRITE(3,11) rel_rate
11        FORMAT(10X,' Release Rate    (Ci/sec) = ',1PE16.2)
        ENDIF
      ELSE
        WRITE(3,35) tank_mass
35      FORMAT(10X,' Initial Mass        (kg) = ',F10.0)
        IF( (rel_type .EQ. 2) .OR. (rel_type .EQ. 4) ) THEN
          WRITE(3,36) rel_rate
36        FORMAT(10X,' Release Rate    (kg/sec) = ',1PE16.2)
        ENDIF
      ENDIF

      WRITE(3,12) rel_height
12      FORMAT(10X,' Release Height       (m) = ',F11.1)
      WRITE(3,13) tank_temp
13      FORMAT(10X,' Storage Temperature  (C) = ',F11.1)

      IF( rel_type .LT. 3 ) THEN
        WRITE(3,14) max_pool_rad
14      FORMAT(10X,' Maximum Pool Radius  (m) = ',F11.1)
      ENDIF

      WRITE(3,15) intake_dist
15      FORMAT(10X,' Intake Distance      (m) = ',F10.0)
```

```
      WRITE(3,16) intake_height
16      FORMAT(10X,' Intake Height        (m) = ',F11.1)

      WRITE(3,17) area
17      FORMAT(10X,' Building Area     (m**2) = ',F10.0)

      IF( rel_type .EQ. 4 ) THEN
        WRITE(3,18) vent_flow
18      FORMAT(10X,' Vent Flow       (m**3/s) = ',F12.2)
      ENDIF

      WRITE(3,19)
19    FORMAT(//10X,' ENVIRONMENTAL CONDITIONS:')
      WRITE(3,20) ubar
20      FORMAT(10X,' Wind Speed                (m/sec) = ',F10.1)
      WRITE(3,21) stab
21      FORMAT(10X,' Atmospheric Stability Class       = ',I8)
      WRITE(3,22) air_temp
22      FORMAT(10X,' Air Temperature               (C) = ',F10.1)
      WRITE(3,23) air_press
23      FORMAT(10X,' Atmospheric Pressure      (mm Hg) = ',F10.1)
      IF( rel_type .LT. 3 ) THEN
        WRITE(3,24) sol_rad
24      FORMAT(10X,' Solar Radiation      (watts/m**2) = ',F10.1)
        iccover = ccover * 10
        WRITE(3,25) iccover
25      FORMAT(10X,' Cloud Cover             (tenths) = ',I8)
        WRITE(3,26) earth_temp
26      FORMAT(10X,' Ground Temperature            (C) = ',F10.1)
      ENDIF

      WRITE(3,27)
27    FORMAT(//10X,' EFFLUENT CHARACTERISTICS:')
      WRITE(3,28) gname
28      FORMAT(10X,' Material Released                 = ',5X,A)
      WRITE(3,29) molec_wt
29      FORMAT(10X,' Molecular Weight        (gm/mole) = ',F10.1)
      IF( rel_type .LT. 3 ) THEN
        WRITE(3,30) boil_point
30      FORMAT(10X,' Initial Boiling Point         (C) = ',F10.1)
        WRITE(3,31) cp
31      FORMAT(10X,' Heat Capacity           (j/gm-C) = ',F12.3)
        WRITE(3,32) hv
32      FORMAT(10X,' Heat of Vapor.            (j/gm) = ',F10.1)
        WRITE(3,33) src_density
33      FORMAT(10X,' Specific Gravity                 = ',F12.3)
        WRITE(3,34) diff_coef
34      FORMAT(10X,' Diffusion Coef.       (cm**2/sec) = ',F12.3)
      ENDIF
    ENDIF
```

```
IF( prtflg2 ) THEN
  WRITE(10,2)

  WRITE(10,3) title
  WRITE(10,4) RDATE, RTIME

  WRITE(10,5)
  IF( rel_type .EQ. 1 ) THEN
    WRITE(10,6)
  ELSE IF( rel_type .EQ. 2 ) THEN
    WRITE(10,7)
  ELSE IF( rel_type .EQ. 3 ) THEN
    WRITE(10,8)
  ELSE IF( rel_type .EQ. 4 ) THEN
    WRITE(10,9)
  ELSE
    STOP ' !!!!!!  RELEASE TYPE ILL DEFINED !!!!!!! '
  ENDIF

  IF( ciflg ) THEN
    WRITE(10,10) tank_mass
    IF( (rel_type .EQ. 2) .OR. (rel_type .EQ. 4) ) THEN
      WRITE(10,11) rel_rate
    ENDIF
  ELSE
    WRITE(10,35) tank_mass
    IF( (rel_type .EQ. 2) .OR. (rel_type .EQ. 4) ) THEN
      WRITE(10,36) rel_rate
    ENDIF
  ENDIF

  WRITE(10,12) rel_height
  WRITE(10,13) tank_temp
  IF( rel_type .LT. 3 ) WRITE(10,14) max_pool_rad
  WRITE(10,15) intake_dist
  WRITE(10,16) intake_height
  WRITE(10,17) area
  IF( rel_type .EQ. 4 ) THEN
    WRITE(10,18) vent_flow
  ENDIF

  WRITE(10,19)
  WRITE(10,20) ubar
  WRITE(10,21) stab
  WRITE(10,22) air_temp
  WRITE(10,23) air_press
  IF( rel_type .LT. 3 ) THEN
    WRITE(10,24) sol_rad
    iccover = ccover * 10
    WRITE(10,25) iccover
    WRITE(10,26) earth_temp
  ENDIF
```

```
          WRITE(10,27)
          WRITE(10,28) gname
          WRITE(10,29) molec_wt
          IF( rel_type .LT. 3 ) THEN
            WRITE(10,30) boil_point
            WRITE(10,31) cp
            WRITE(10,32) hv
            WRITE(10,33) src_density
            WRITE(10,34) diff_coef
          ENDIF
        ENDIF

C       CORRECT FOR DEPARTURE FROM STANDARD CONDITIONS

C       Compute molar volume for conversion to ppm

        mol_vol =  22.414E-3  * ( std_press/air_press ) *
       .                   ( (air_temp + 273.16)/std_temp)

        IF( rel_type .LE. 2 ) THEN

C       Compute air_density at air_temp and air_press

          air_density = air_dens_stp * (std_temp/(air_temp+273.16))
       .                             * (air_press/std_press)

C       Correct boiling point for atmospheric pressure

          boil_point = ( std_boil_pt + 273.16 )
       .   / ( 1 + ugcnst * ( std_boil_pt + 273.16 )
       .   * ( ALOG(760.)-ALOG(air_press) )
       .   / ( hv * molec_wt )  ) - 273.16

C       Compute initial saturation vapor pressure

          IF( boil_point .GT. tank_temp ) THEN
            pconst = ( molec_wt * hv )
       .         / ( ugcnst * (boil_point+273.16) )
            log_sat_press = pconst
       .         * ( 1.0 - (boil_point+273.16) / (tank_temp+273.16) )
            sat_press = air_press * EXP( log_sat_press )
          ELSE
            sat_press = air_press
          ENDIF
        ENDIF

C       COMPUTE FACTOR TO CONVERT g/m**3 TO ppm

        ppmconv = mol_vol * 1.0E6 / molec_wt
```

```
C       CONVERT FROM CGS TO MKS

        cp = cp * 1.0E3
        hv = hv * 1.0E3
        sat_press = sat_press / 1.0E3
        src_density = src_density * 1.0E3
        diff_coef = diff_coef / 1.0E4
        air_press = air_press / 1.0E3

        RETURN
        END
```

```
      SUBROUTINE RINPUTE( rel_type )

C*************************************************************************
C
C     RINPUTE                                          EXTRAN  Version 1.2
C     Allows User to Revise Environmental Data
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 7/90 from INPUT
C     Updated: 10/90
C
C     Description: Prompts user for environmental data input depending
C                  on release type.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     RINPUT
C
C*************************************************************************

      INCLUDE 'EFFLUENT.INC'
      INCLUDE 'ENVIRON.INC'

      REAL ugcnst

      INTEGER*2 NN, rel_type, iccover

      CHARACTER UC, yn

      ugcnst = 8.3144

      WRITE(*,'(//A)') ' REVISE ENVIRONMENTAL CONDITIONS:'

      IF( rel_type .LE. 2 ) THEN
        NN = 7
      ELSE
        NN = 4
      ENDIF

   10 CONTINUE

C..... REVIEW INPUT DATA AND REVISE AS NECESSARY

      WRITE(*,'(//A)') ' ENVIRONMENTAL CONDITIONS REVIEW:'
      WRITE(*,'(A,F10.1)')  '  1   Wind Speed                (m/sec) = ',
                                                                    ubar
```

```
      WRITE(*,'(A,I8)')     ' 2   Atmospheric Stability Class      = ',
     .                                                              stab

      WRITE(*,'(A,F10.1)')  ' 3   Air Temperature              (C) = ',
     .                                                           air_temp
      WRITE(*,'(A,F10.1)')  ' 4   Atmospheric Pressure     (mm Hg) = ',
     .                                                         air_press
      IF( rel_type .LT. 3 ) THEN
        WRITE(*,'(A,F10.1)')' 5   Solar Radiation     (watts/m**2) = ',
     .                                                           sol_rad
        iccover = 10*ccover
        WRITE(*,'(A,I8)')   ' 6   Cloud Cover             (tenths) = ',
     .                                                           iccover
        WRITE(*,'(A,F10.1)')' 7   Ground Temperature          (C) = ',
     .                                                        earth_temp
      ENDIF

      WRITE(*,'(1X,A)') ' C   START CALCULATIONS '
      WRITE(*,'(1X,A)') ' X   EXIT PROGRAM '

   20 CONTINUE
      WRITE(*,21)
   21 FORMAT(/'  ENTER  C  TO CONTINUE IF DATA ARE CORRECT,',
     .  /'  IF DATA ARE NOT CORRECT ENTER NUMBER OF ITEM TO BE CHANGED',
     .  /'  OR, ENTER  X  TO STOP PROGRAM : '\ )

      READ(*,'(A)',ERR=10) yn

      IF( UC(yn) .EQ. 'X' )
     .   STOP '  PROGRAM STOPPED AT ENVIRONMENTAL DATA REVISION '
      IF( UC(yn) .EQ. 'C' ) RETURN

C.....  DETERMINE ITEM TO BE REVISED

      IF( yn .EQ. '1' )  THEN
        GOTO 100
      ELSE IF( yn .EQ. '2' )  THEN
        GOTO 200
      ELSE IF( yn .EQ. '3' )  THEN
        GOTO 300
      ELSE IF( yn .EQ. '4' )  THEN
        GOTO 400
      ELSE IF( yn .EQ. '5' )  THEN
        GOTO 500
      ELSE IF( yn .EQ. '6' )  THEN
        GOTO 600
      ELSE IF( yn .EQ. '7' )  THEN
        GOTO 700
      ENDIF

      WRITE(*,'(A,I2,A)' )
     .    '  ENTER NUMBER FROM 1 THRU ', NN, ', A C, OR AN X.'
      GOTO 10
```

```
C......  DATA REVISION SECTION

  100 CONTINUE
  105 WRITE(*,'(/A\)') ' Enter Wind Speed (m/s) (e.g., 5.0) : '
      READ(*,'(F15.0)',ERR=105) ubar
      IF( (ubar .LT. 0.5) .OR. (ubar .GT. 20.0) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is (0.5 - 20.0)'
        GOTO 105
      ELSE
        WRITE(*,'(F15.1)')        ubar
      ENDIF
      GOTO 10

  200 CONTINUE
  205 WRITE(*,'(/A)') ' Atmospheric Stability Classes '
      WRITE(*,'(A)') ' 1 = A, 2 = B, 3 = C, 4 = D, 5 = E, 6 = F, 7 = G'
      WRITE(*,'(A\)') ' Enter Stability Class Number : '
      READ(*,'(I1)',ERR=205) stab
      IF( (stab .LT. 1) .OR. (stab .GT. 7) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is (1 - 7)'
        GOTO 205
      ELSE
        WRITE(*,'(11x,i4)')     stab
      ENDIF
      GOTO 10

  300 CONTINUE
  305 WRITE(*,'(/A,A\)') ' Enter Ambient Air Temperature (C)',
     .                   ' (e.g., 10.0) : '
      READ(*,'(F15.0)',ERR=305) air_temp
      IF( (air_temp .LT. -40.0) .OR. (air_temp .GT. 50.0) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is (-40.0 TO +50.0)'
        GOTO 305
      ELSE
        WRITE(*,'(F15.0)')        air_temp
      ENDIF
      GOTO 10

  400 CONTINUE
  405 WRITE(*,'(/A,A\)') ' Enter Atmospheric Pressure (mm Hg)',
     .                   ' (e.g., 760.0) : '
      READ(*,'(F15.0)',ERR=405) air_press
      IF( (air_press .LT. 600.0) .OR. (air_press .GT. 800.0) ) THEN
        WRITE(*,'(/A)') ' Out of Range; Range is (600.0 - 800.0)'
        GOTO 405
      ELSE
        WRITE(*,'(F15.1)')        air_press
      ENDIF

C     Correct boiling point for atmospheric pressure

      boil_point = ( std_boil_pt+273.16 )
     .   / ( 1 + ugcnst * (std_boil_pt+273.16)
```

```
     .    * ( ALOG(760.)-ALOG(air_press) )
     .    / ( hv * molec_wt )  ) - 273.16

      IF( boil_point .LT. 0.0 ) THEN
        tref = boil_point + 273.16
      ELSE
        tref = 273.16
      ENDIF

C     Reset diffusion coefficient

      diff_coef = idiff_coef
      IF( (diff_coef .LE. 0.0) .AND. (rel_type .LT. 3) ) THEN
        WRITE(*,'(A)') ' Computing revised diffusion coefficient '
        CALL DIFCOEF
      ENDIF
      GOTO 10

  500 CONTINUE
  505 WRITE(*,'(/A,A\)')
     .  ' Enter Value for Solar Radiation ',
     .  '(watts/m**2) (e.g. 500.0) : '
      READ(*,'(F15.0)',ERR=505) sol_rad
      IF( (sol_rad .LT. 0.0) .OR. (sol_rad .GT. 1200.) ) THEN
        WRITE(*,'(A,A)') ' Solar Radiation Out of Range, Range is ',
     .                   '0 to 1200. '
        GOTO 505
      ELSE
        WRITE(*,'(F15.0)')        sol_rad
      ENDIF
      GOTO 10

  600 CONTINUE
  605 WRITE(*,'(/A\)')
     .  ' Enter Cloud Cover in tenths (e.g. 4 ) '
      READ(*,'(I4)',ERR=605) iccover
      IF( (iccover .LT. 0 ) .OR. ( iccover .GT. 10 ) ) THEN
        WRITE(*,'(A,A)') ' Cloud Cover Out of Range, Range is ',
     .                   '0 to 10 '
        GOTO 605
      ELSE
        WRITE(*,'(I4)')        iccover
      ENDIF
      ccover = FLOAT(iccover) / 10.
      GOTO 10

  700 CONTINUE
  705 WRITE(*,'(/A,A\)') ' Enter Temperature (C) of the Ground',
     .                   ' (e.g., Air Temp. + 10.0) : '
      READ(*,'(F15.0)',ERR=705) earth_temp
      IF( (earth_temp .LT. -40.0) .OR. (earth_temp .GT. 60.0) ) THEN
          WRITE(*,'(/A)') ' Out of Range; Range is (-40.0 to +60.0)'
          GOTO 705
```

```
ELSE
  WRITE(*,'(F15.0)')        earth_temp
ENDIF
GOTO 10

END
```

```
      FUNCTION UC(A)

C*********************************************************************
C
C     UC                                                EXTRAN  Version 1.2
C
C     Return an upper case letter if input is lower case letter
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created:   7/90
C     Updated:  10/90
C
C     Description:  The input character is examined.  If has an ASCII
C                   character in the range from 97 through 122, it is a
C                   lower case letter.  If it is a lower case letter
C                   NN is set equal to the value minus 32.  This is the
C                   upper case equivalent to the letter.  UC is set equal
C                   to CHAR(NN)
C
C*********************************************************************

      CHARACTER UC, A

      INTEGER*2 NN

      NN = ICHAR( a )

      IF( (NN .GE. 97) .AND. (NN .LE. 122) ) THEN
        NN = NN - 32
        UC = CHAR( NN )
      ELSE
        UC = A
      ENDIF

      RETURN
      END
```

```
      SUBROUTINE WSIG( TTY, TTZ, wsigy, wsigz )

C*********************************************************************
C
C     WSIG                                                EXTRAN  Version 1.2
C     Computes Building-Wake Diffusion Coefficients
C
C     J.V. Ramsdell
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 11/88
C     Updated: 10/90
C
C     Description: Computes Sigma YW and ZW for building wake
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     EXTRAN, MODELPAR
C
C*********************************************************************

      INCLUDE 'ENVIRON.INC'
      INCLUDE 'SCENARIO.INC'

      REAL*4    ZOL(7), c, ustar, hts, vts, tty, ttz

      DATA ZOL/ -1.0, -0.6, -0.33, -0.07, 0.09, 0.2, 0.5 /

C     Constant ==>  includes sigv, sigw, lagrangian
C     autocorrelations and calibration constant

      C = 0.5

C     Horizontal Time Scale for Diffusion  ==>  HTS

      USTAR = 0.4 * UBAR / ALOG(10.0/0.1)
      HTS = SQRT(AREA) / USTAR

C     Vertical Time Scale for Diffusion ==>  VTS
C     First find approximate z/L at 10 m given zo = 0.1 m and stability
C     class

      VTS = HTS / ( 2 + ZOL(STAB) )

C     Compute Wake diffusion Coefficients

      WSIGY =  C * HTS**2 * ( EXP(-TTY/HTS) * (-TTY/HTS - 1) + 1 )
      WSIGZ =  C * VTS**2 * ( EXP(-TTZ/VTS) * (-TTZ/VTS - 1) + 1 )
```

```
WSIGY = SQRT(WSIGY)
WSIGZ = SQRT(WSIGZ)

RETURN
END
```

```
C*********************************************************************
C
C     EFFLUENT.INC                                        EXTRAN  Version 1.2
C
C     Created: 11/88
C     Updated: 10/90
C
C     Description:  Contains effluent information and constants used
C                   throughout the EXTRAN code.
C
C     Included in modules: DIFCOEF, EXTRAN, INPUT, INPUTEF, PUFFINIT,
C                          PUFFMASS, RINPUT, RINPUTE
C
C*********************************************************************

      REAL*4  src_density, cp, hv, std_boil_pt, vapor_density, tref,
     .        vapor_press, diff_coef, sat_press, molec_wt, pconst,
     .        ppmconv, boil_point, idiff_coef

      COMMON /effluent/ src_density, cp, hv, std_boil_pt, vapor_density,
     .                  vapor_press, diff_coef, sat_press, molec_wt,
     .                  pconst, tref, ppmconv, boil_point, idiff_coef


C*********************************************************************
C
C     ENVIRON.INC                                         EXTRAN  Version 1.2
C
C     Created: 11/88
C     Updated: 10/90
C
C     Description:  Contains environmental information and constants
C                   used throughout the EXTRAN code.
C
C     Included in modules: CHIT, DIFCOEF, EXTRAN, INPUT, INPUTE,
C                          INPUTEF, MODELPAR, NSIG, PUFFINIT, PUFFMASS,
C                          RINPUT, RINPUTE, WSIG
C
C*********************************************************************

      REAL*4  ubar, z, znot, air_temp, air_press, mix_depth, earth_temp,
     .        sol_rad, ccover, air_density

      INTEGER*2 stab

      COMMON /environ/ ubar, z, znot, air_temp, air_press, mix_depth,
     .                 earth_temp, sol_rad, ccover, air_density, stab
```

```
C*********************************************************************
C
C     OPTIONS.INC                                          EXTRAN  Version 1.2
C
C     Created: 7/90
C     Updated: 10/90
C
C     Description: Contains flags that control model output.
C
C     Included in modules: CEPLOT, EXTRAN, INPUT, INPUTO, INPUTS,
C                          MODELPAR, RINPUT
C
C*********************************************************************

      LOGICAL ppmflg, prtflg1, prtflg2, pltflg, histflg1, histflg2,
     .        statflg, ciflg

      INTEGER*2 run_count

      CHARACTER*12 PRTFILE, CRONFILE, MBFILE

      COMMON /options/ ppmflg, prtflg1, prtflg2, pltflg, histflg1,
     .                 histflg2, statflg, ciflg, run_count

      COMMON /filenames/ PRTFILE, CRONFILE, MBFILE


C*********************************************************************
C
C     POOL.INC                                             EXTRAN  Version 1.2
C
C     Created: 11/88
C     Updated: 10/90
C
C     Description:  Contains information on the pool of material
C
C     Included in modules: EXTRAN, PUFFINIT, PUFFMASS
C
C*********************************************************************

      REAL*4 rel_mass, pool_mass, pool_temp, pool_vol, pool_radius,
     .       pool_area, pool_thick, net_swrad, lw_in, lw_out,
     .       net_lwrad, air_conv, grnd_cond, net_flux

      COMMON /pool/ rel_mass, pool_mass, pool_temp, pool_vol,
     .       pool_radius, pool_area, pool_thick, net_swrad, lw_in,
     .       lw_out, net_lwrad, air_conv, grnd_cond, net_flux
```

```
C***********************************************************************
C
C     PRINT.INC                                          EXTRAN  Version 1.2
C
C     Created: 11/88
C     Updated: 10/90
C
C     Description: Contains information for plotting.
C
C     Included in modules: CEPLOT, CHIT, EXTRAN
C
C***********************************************************************

      REAL*4  conc(181), expos(181), avconc(181), exposure

      INTEGER*2 ttime(181), count, npuffs(181), start

      COMMON /print/ conc, expos, avconc, ttime, count, npuffs, start,
     +               exposure


C***********************************************************************
C
C     PUFF.INC                                           EXTRAN  Version 1.2
C
C     Created: 11/88
C     Updated: 10/90
C
C     Description:  Contains puff parameters
C
C     Included in modules: CHIT, EXTRAN, PUFFINIT, RELPUFF
C
C***********************************************************************

      REAL*4 puffm(500), puffage(500), pufftvy(500), pufftvz(500),
     .       puffnsigy(500), puffnsigz(500), pufftsigy(500),
     .       pufftsigz(500), pmass1, pmass2, isigy1, isigz1, isigy2,
     .       isigz2, tvy1, tvz1, tvy2, tvz2

      INTEGER*2 numpuffs

      COMMON /puff/ puffm, puffage, pufftvy, pufftvz, puffnsigy,
     .              puffnsigz, pufftsigy, pufftsigz, pmass1, pmass2,
     .              isigy1, isigz1, isigy2, isigz2, tvy1, tvz1, tvy2,
     .              tvz2, numpuffs
```

```
C*********************************************************************
C
C     SCENARIO.INC                                        EXTRAN  Version 1.2
C
C     Created: 11/88
C     Updated: 10/90
C
C     Description:  Contains scenario information and constants used
C                   throughout the WAKE model.
C
C     Included in modules: CHIT, EXTRAN, INPUT, INPUTO, INPUTS,
C                          MODELPAR, PUFFINIT, PUFFMASS, RINPUT, WSIG
C
C*********************************************************************

      REAL*4   intake_dist, intake_height, area, rel_height, tank_mass,
     .         tank_temp, rel_rate, max_pool_rad, max_pool_area,
     .         init_tank_mass, init_rel_rate, vent_flow

      INTEGER*2  rel_type, pri, delta_time

      COMMON /scenario/ intake_dist, intake_height, area, rel_height,
     .                  tank_mass, tank_temp, rel_rate, max_pool_rad,
     .                  max_pool_area, rel_type, pri, delta_time,
     .                  init_tank_mass, init_rel_rate, vent_flow
```

DISTRIBUTION

No. of
Copies

OFFSITE

10 C. Ferrell
Division of Reactor Accident
Analysis
Office of Nuclear Regulatory
Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555

ONSITE

50 Pacific Northwest Laboratory

K. J. Allwine
C. M. Berkowitz
J. C. Doran
J. G. Droppo
C. E. Elderkin
J. W. Falco
C. S. Glantz
J. M. Hales
P. C. Hayes
D. J. Hoitink
L. A. Mahoney
P. C. Owczarski
W. T. Pennell
J. V. Ramsdell (27)
R. L. Skaggs
E. D. Skyllingstad
L. L. Wendell
Publishing Coordination (2)
Technical Report Files (5)

NRC FORM 335 (2-89) NRCM 1102, 3201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

1. REPORT NUMBER (Assigned by NRC, Add Vol., Supp., Rev., and Addendum Numbers, if any.)
NUREG/CR-5656
PNL-7510

2. TITLE AND SUBTITLE
EXTRAN: A Computer Code for Estimating Concentrations of Toxic Substances at Control Room Air Intakes

3. DATE REPORT PUBLISHED
MONTH: March
YEAR: 1991

4. FIN OR GRANT NUMBER
B2929

5. AUTHOR(S)
J. V. Ramsdell

6. TYPE OF REPORT
Technical

7. PERIOD COVERED *(Inclusive Dates)*

8. PERFORMING ORGANIZATION – NAME AND ADDRESS *(If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)*
Pacific Northwest Laboratory
Richland, WA 99352

9. SPONSORING ORGANIZATION – NAME AND ADDRESS *(If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)*
Division of Safety Issue Resolution
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555

10. SUPPLEMENTARY NOTES

11. ABSTRACT *(200 words or less)*

This report presents the NRC staff with a tool for assessing the potential effects of accidental releases of radioactive materials and toxic substances on habitability of nuclear facility control rooms. The tool is a computer code that estimates concentrations at nuclear facility control room air intakes given information about the release and the environmental conditions.

The name of the computer code is EXTRAN. EXTRAN combines procedures for estimating the amount of airborne material, a Gaussian puff dispersion model, and the most recent algorithms for estimating diffusion coefficients in building wakes. It is a modular computer code, written in FORTRAN-77, that runs on personal computers. It uses a math coprocessor, if present, but does not require one. Code output may be directed to a printer or disk files.

12. KEY WORDS/DESCRIPTORS *(List words or phrases that will assist researchers in locating the report.)*
Control Room Habitability
Building Wakes
Atmospheric Diffusion
Toxic Substances

13. AVAILABILITY STATEMENT
Unlimited

14. SECURITY CLASSIFICATION
(This Page) Unclassified
(This Report) Unclassified

15. NUMBER OF PAGES

16. PRICE