```
C           SUBROUTINE FOR DEVICE TYPE 9-DUMMY
            EFF=0
C           STORE EFF IN MAT DD
            DD(J,20)=EFF
          ELSE
C           DEVICES >9 OR <1 (OR ANY DEVICE NOT IN ABOVE CASES)
            DD(J,20)=0
          ENDIF
  080 CONTINUE
  100 CONTINUE
          RETURN
C        **********  END OF SUBROUTINE CNRSUB  ***************************
          END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

NASA
Technical
Memorandum

NASA TM - 108409

COMPUTERIZED ATMOSPHERIC TRACE CONTAMINANT CONTROL SIMULATION FOR MANNED SPACECRAFT

By J.L. Perry

Structures and Dynamics Laboratory
Science and Engineering Directorate

June 1993

NASA

National Aeronautics and
Space Administration

George C. Marshall Space Flight Center

MSFC - Form 3190 (Rev. May 1983)

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1993	Technical Memorandum

4. TITLE AND SUBTITLE
Computerized Atmospheric Trace Contaminant Control Simulation for Manned Spacecraft

5. FUNDING NUMBERS

6. AUTHOR(S)
J.L. Perry

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TM - 108409

11. SUPPLEMENTARY NOTES
Prepared by Structures and Dynamics Laboratory, Science and Engineering Directorate.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified—Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
Buildup of atmospheric trace contaminants in enclosed volumes such as a spacecraft may lead to potentially serious health problems for the crewmembers. For this reason, active control methods must be implemented to minimize the concentration of atmospheric contaminants to levels that are considered safe for prolonged, continuous exposure. Designing hardware to accomplish this has traditionally required extensive testing to characterize and select appropriate control technologies. Data collected since the Apollo project can now be used in a computerized performance simulation to predict the performance and life of contamination control hardware to allow for initial technology screening, performance prediction, and operations and contingency studies to determine the most suitable hardware approach before specific design and testing activities begin. The program, written in FORTRAN 77, provides contaminant removal rate, total mass removed, and per pass efficiency for each control device for discrete time intervals. In addition, projected cabin concentration is provided. Input and output data are manipulated using commercial spreadsheet and data graphing software. These results can then be used in analyzing hardware design parameters such as sizing and flow rate, overall process performance and program economics. Test performance may also be predicted to aid test design.

14. SUBJECT TERMS atmospheric contamination, trace contaminant, control, software, analysis, analytical tool, spacecraft, systems design, systems analysis, ECLSS, atmospheric revitalization, computer program

15. NUMBER OF PAGES 173

16. PRICE CODE NTIS

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

ACKNOWLEDGMENTS

A complex computer model requires the expertise of many individuals to produce a useful engineering design tool. The trace contaminant control simulation computer program has benefited from the scientific, engineering, and computer programming expertise of many individuals since 1965. In particular, the granular activated charcoal performance and capacity characterization conducted by Dr. A.J. Robell and his colleagues, C.R. Arnold, A. Wheeler, G.J. Kersels, and R.P. Merrill, at Lockheed Missiles and Space Company in Sunnyvale, CA, formed the basis for the contaminant removal device routines. The work by Thomas Olcott, also of Lockheed, which built on the techniques developed by Dr. Robell, provided the first stepping stone for developing today's version of the computer program. The first generalized version was developed by R.B. Jagow, R.J. Jaffe, and R.A. Lamparter of Lockheed for analyzing Spacelab atmospheric contamination control designs. This work was expanded into an even more general model by Lamparter and R.Q. Yee to become the predecessor of the current program design. The work of Lamparter is further acknowledged for developing a more user-friendly version of the program which was the version that I began working on in 1989. These accomplishments were made under the direction of C.D. Ray of NASA's Marshall Space Flight Center. Without the basis set by these individuals, the current version would not exist. Development of version 8.0 also involved Wes Coleman of Sverdrup Technology in Huntsville, AL, who helped make the plot data output capabilities possible and provided a very nice solution to a convergence instability problem. He also provided a fresh viewpoint which helped find several bugs which I had overlooked. I am extremely grateful to Bob DaLee of McDonnell Douglas Space Systems Company in Huntsville, AL, who painstakingly translated the computer program from the IBM/compatible personal computer version to a version which is compatible with Apple Macintosh computers.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

TECHNICAL MEMORANDUM

COMPUTERIZED ATMOSPHERIC TRACE CONTAMINANT CONTROL SIMULATION FOR MANNED SPACECRAFT

INTRODUCTION

Buildup of atmospheric trace contaminants in enclosed volumes such as a manned spacecraft may lead to potentially serious health problems for the crewmembers. For this reason, control methods are implemented to minimize the concentration of atmospheric contaminants to levels that are considered safe for prolonged, continuous exposure. Methods employed to achieve these levels are classified into two major categories—passive and active. Passive control refers to the material selection and control process to which all bulk materials and assembled articles must be subjected to qualify for flight onboard a manned spacecraft. Criteria included in this process are flammability, odor, and offgassing characteristics. Through this screening, the uncontrolled production via offgassing mechanisms of atmospheric contaminants from materials used in the habitable areas of the spacecraft are limited, but not eliminated. Active control methods must be used to provide the second level of contaminant control. Activities such as food preparation, housekeeping, and personal hygiene contribute to the base contamination load found in the spacecraft atmosphere. Contaminants are also produced by crewmember metabolism, experiment hardware and operations, and spacecraft hardware operations. The activities onboard the spacecraft combined with material offgassing produce contaminants in sufficient quantities that the atmospheric concentration may build up to levels which the crewmembers may find irritating or possibly intolerable. Therefore, the total contamination control approach requires both the passive and active approaches.

Active contamination control can be achieved through various means. Typical technologies used onboard manned spacecraft include physical adsorption on granular activated charcoal or other adsorbent material, chemical adsorption using impregnated granular activated charcoals and granular lithium hydroxide (LiOH), ambient temperature catalytic oxidation, and high temperature catalytic oxidation. Other means which are significant include removal by absorption into humidity condensate and spacecraft atmospheric leakage. These latter methods are not specifically designed to remove atmospheric contaminants, but have been shown to be significant when the total contaminant material balance in a spacecraft is considered.

Designs for active contamination control hardware have evolved from single granular-activated charcoal canisters used onboard earlier manned spacecraft, to a regenerable molecular sieve adsorbent used onboard the *Skylab* orbital workshop. Space shuttle and Spacelab programs use layered adsorbent canisters containing nonregenerable adsorbent and ambient temperature catalytic oxidation materials. Future contamination control hardware will build on these technologies to include regenerable adsorbent materials and high temperature oxidation catalysts.

Designing these contamination control devices is difficult since they must be designed without thorough knowledge of the overall load that must be controlled. Historical data from past manned missions are important for understanding the loads from offgassing, crew metabolism, and housekeeping sources. This information forms the basis for a generic spacecraft load model which can be used for hardware design purposes. As the spacecraft design matures, the contamination control hardware is refined by replacing the load model data with data for the actual flight hardware.

Ultimately, the spacecraft and contamination control designs converge. The final result is a robust contamination control hardware design with sufficient margin which allows a wide range of hardware and activities to be accommodated onboard the spacecraft with no adverse effects on the crew-members' health.

COMPUTER SIMULATION BACKGROUND

In the past, designing contamination control hardware required extensive testing to characterize and select the appropriate control technologies. Capacities for granular-activated charcoal and other adsorbent materials had to be determined along with development of theoretical approaches to predicting the performance of future contamination control hardware. Early efforts to select technology and design hardware for spacecraft contamination control hardware began during the *Apollo* project. Studies conducted before 1965 by Lockheed Missiles and Space Company in Sunnyvale, CA, developed early spacecraft contaminant load models and selected candidate technologies.[1 2 3] These technologies have been the basis for designing contamination control hardware for manned spacecraft ever since. More detailed studies on adsorption and catalytic oxidation followed. High temperature catalytic oxidation was studied by Olcott[4] and contaminant adsorption by Robell, et al.[5] by late 1970. In his work, Robell developed an extremely useful correlation for predicting the loading capacity of granular-activated charcoal by using the potential theory developed by Polanyi.[6] This work became the basis for computer simulation of adsorption processes to be used onboard manned spacecraft for atmospheric contamination control. The first generation simulation program was developed by Olcott before May 1972 using the technique developed by Robell.[7] By 1975, a prototype contamination control system had been fabricated by Lockheed, and the first generation computer simulation program for adsorption had been used under NASA contract NAS 1-11526 to analyze the hardware performance.[8]

Work continued in characterizing contamination control technologies, and low temperature oxidation catalysts were studied in 1977. That same year, the first integrated computer simulation model for spacecraft atmospheric contamination control was developed by Jagow, et al. at Lockheed.[9] This program was specifically tailored for analyzing the performance of the Spacelab contamination control canister which consisted of two layers of adsorbent material followed by a layer of low-temperature, carbon monoxide oxidation catalyst. This program was modified and released later in 1977 as a general computer simulation program capable of simulating any spacecraft contamination control system given the appropriate input data.[10] This basic computer program was used for design and preflight performance analyses for Spacelab missions.

Little work was conducted on the basic computer program until NASA contract NAS8-36406 produced a new release of the computer program in 1986 which featured input data manipulation using commercially available spreadsheet software and operation on a personal computer. This program version, designated version 5.0, was used as the basis for Space Station *Freedom* contamination control hardware design.[11] Improvements since 1986 have included further enhancements for input data manipulation, a routine to calculate the toxic hazard index, improved simulation of humidity condensate absorption, and improved output data manipulation flexibility. Written in FORTRAN 77, the program can be run on IBM-compatible personal computers (version 8.0 Alpha), Digital Equipment Corporation VAX mainframe computers using the VMS operation system (version 8.0 Beta), and Apple Computer, Inc., Macintosh® computers (version 8.0 Gamma). Table 1 summarizes the progression of the computer program from its initial version in 1970 to the present version which was completed in April 1992.

Table 1. Atmospheric TCCS program development summary.

Version	Release Date	Remarks
1.0	May 1972	Initial program version developed by Lockheed for assessing charcoal bed performance using empirical charcoal loading data. Documented in NASA contract report NAS CR-2027.
2.0	November 28, 1975	Improved program version used by Lockheed for prototype contamination control hardware performance predictions.
3.0	January 15, 1977	A generalized program developed by Lockheed based on versions 1.0 and 2.0 to predict Spacelab contamination control system performance.
4.0	September 1, 1977	A revised version of the Spacelab computer program which simulated more general contamination control systems designs. This version was developed by Lockheed.
5.0	September 5, 1986	The first version developed by Lockheed specifically for personal computer use. This version greatly enhanced data input capabilities.
6.0	June 1, 1989	Version 6.0 added the calculation of the toxic hazard index. This version was developed by the Boeing Company and MSFC personnel.
7.0 to 7.9	August 2, 1980 through April 1, 1992	Versions 7.0 through 7.9 were developed at MSFC. This progression improved the ability of the program to simulate transient contamination cases and an improved humidity condensate absorption routine based on Henry's law. Charcoal bed breakthrough calculation instability was corrected using a Newton-Raphson method and user interaction was improved.
8.0	April 10, 1992	Output routines were modified to allow data output for concentration, efficiency, and toxic hazard index to separate files which can be accessed by spreadsheet programs for data analysis and plotting. This version was developed at MSFC.

SIMULATION PROGRAM OVERVIEW

Computer Hardware Requirements

The trace contamination control simulation (TCCS) computer program is written in FORTRAN 77 and can be run on IBM AT-compatible personal computers running the disk operating system (DOS) version 3.0 or higher. This version requires at least an IBM AT or compatible machine with an 80287 floating point processor and a minimum of 512 kilobytes of memory. Upgraded personal computers using a 386 or 486 processor are preferred since they provide more rapid execution. The program is compiled and linked on the personal computer using Ryan-McFarland FORTRAN version 2.42.

A similar version of the program, version 8.0 Beta, allows operating on a Digital Equipment Corporation VAX mainframe computer which uses the VMS operating system. A 50,000 block disk allocation is recommended to accommodate files produced by a large simulation. The program files occupy up to 5,000 blocks of disk space. The mainframe version may offer increased speed and disk storage for very large simulations; however, recent improvements in personal computing speed and disk storage capacity have made this version less desirable especially since input data files are more easily created using a personal computer and uploaded to the mainframe computer. This makes a personal computer with file transfer capabilities an important hardware requirement for using the mainframe version.

Version 8.0 Gamma, which runs on Apple Computer, Inc., Macintosh machines, operates similarly to the Alpha and Beta versions. The Macintosh user interface, which uses a mouse and pop-up menus for input file management, makes the program even more user-friendly by requiring fewer commands to be entered from the keyboard. It is recommended that this version be run on a Macintosh II family machine with 8.0 megabytes of RAM and Macintosh System 7 operating system version 7.0 or later.

Input/Output Data Manipulation Recommendations

Input data is manipulated using commercially available spreadsheet software. The program can read input data using both space- and comma-delimited formats, while output data is produced in a space-delimited ASCII format. Because of these program input and output data formats, it is recommended that the Lotus Development Corporation (Lotus) software, Lotus 123® with Printgraph®, be used for input data generation and output data manipulation for the personal computer version. This software can print to a file in the space-delimited ASCII format required by the program and import program output data directly without any additional processing by the user. Other spreadsheet programs may be used, such as Microsoft Corporation's Excel®, for input file generation, but the user must save the input files in the comma separated variable (CSV) format. The ASCII-formatted output data produced by the program requires additional manipulation within the spreadsheet program, such as using the parse function, to allow it to be analyzed if Excel is used for output data analysis.

The mainframe program version input and output data manipulation is more difficult. It is recommended that input data files be prepared on a personal computer using Lotus 123® and uploaded to the mainframe. Text files may be created using the VMS editor; however, managing these files is more difficult than using a spreadsheet program. Some output data, such as data to be used to generate plots, should be downloaded and manipulated using the personal computer spreadsheet software or other commercially available data plotting software.

The Macintosh version of the program should use the Microsoft Corporation Excel spreadsheet software for input file preparation and output data analysis. Input files prepared using Excel must be saved as a CSV file for the program to read the data. Output data produced by the program may be imported into the Excel spreadsheet and then the parse function run to place each number in a spreadsheet cell. Data plots may be prepared using Excel or the data may also be loaded into other popular Macintosh plotting software such as Cricket Graph® or Kaleidograph®.

Calculation Overview

The TCCS program calculates the cabin concentration of specified contaminants given a set of contamination control devices as a function of time. Depending on the output data selections made, data for each basic time increment may be output detailing the current contaminant cabin concentrations, their comparison to the spacecraft maximum allowable concentrations (SMAC's), the removal rate for each contaminant in each device, the cumulative total contaminant mass removed by each device, the increment device single pass removal efficiency for each contaminant, and the toxic hazard index.

Input Data

Input information consists of the names and properties of each contaminant, the order and design characteristics of the removal devices, and time dependent data such as changes in contaminant generation rate or removal device flow rate as a function of time. These input data are read into the program using three separate data files. The first file contains contaminant data, the second contains removal device definition data, and the third contains time dependent data. Files are used for input to minimize repetitive input by the user.

Contaminant Removal Devices

Removal device configuration is provided by a routine which allows up to 15 total devices with parallel and series arrangement. Up to three devices in parallel can be placed upstream of any removal device. The program simulates nine removal device types. Two device types, the cabin and leakage, must always be included in the device definition file. Modular programming allows for quick revision or addition of device types if necessary; however, any internal modifications require recompiling and relinking the program.

Calculation Convergence

A discrete time increment approach is used by the program which uses contaminant concentration from the previous increment as the initial concentration. The final concentration is calculated based on the device configuration, contaminant generation rate, contaminant removal rate, removal device efficiency, and other specific information relevant to that time increment. This approach allows flexibility which simulates transient situations such as inadvertent contaminant releases or other time dependent contamination events. The calculation data are stored in matrices which increases the program execution speed.

Since contaminant properties can vary with respect to ease of removal and cabin concentration, small time increments would be the most accurate for any given simulation. However, the program uses a log-mean direct solution approach combined with a Newton-Raphson convergence routine to allow much larger time increments which significantly reduce the computer runtime required. This approach produces an exact solution during steady-state conditions with constant removal device efficiencies. This is normally the case during a simulation, but efficiency may vary in some cases, such as for a contaminant that is breaking through an adsorbent bed. In this case, the Newton-Raphson convergence technique combined with decreasing the time increment to one-twentieth the size of the basic increment for that contaminant only provides the desired calculation accuracy without affecting the overall run time.

The prediction and convergence routines use the Newton-Raphson technique to ensure that the removal device calculations are made with the proper contaminant inlet concentration. Contaminant concentrations are predicted using removal efficiencies of the previous increment and the generation rates for the present increment. Actual device efficiencies and contaminant concentrations are calculated by the convergence routine and compared to the predicted values. This process continues until the predicted and calculated values agree within a specified error.

If convergence is not obtained for a contaminant, warning statements built into the program alert the user. In addition, other warning statements concerning input and output are provided to aid simulation diagnostics. Print options are provided to allow printing the contents of calculation matrices and convergence values. These options allow quick identification of problems with the input device configuration or contaminant data.

Output Data

Program output data consist of standard formatted output with column headings and page numbers and interactive formatted output to files for data plotting. Standard formatted data output is specified using print switches in the device definition input file while plot data output is specified interactively. Standard output consists of the contaminant name, cabin concentration, SMAC, and a designation of whether the cabin concentration exceeds the SMAC. Also, the standard print switches allow the contaminant removal rate, cumulative mass removed, and removal efficiencies for each contaminant in each device to be output. Standard output may be directed to the computer terminal screen, a specific data file, or a printer. These data include page numbers, time increment identification, and other information relating to the time increment. Interactive output decisions are made by the user to produce separate files containing concentration, device efficiency, and toxic hazard index data as a function of time. These files are in a spreadsheet format and can be imported to a spreadsheet to allow manipulation and preparation of data plots. The toxic hazard index is also interactively sent to the standard formatted output in addition to a plot file if desired.

PROGRAM OPERATION

The computerized atmospheric trace contaminant control simulation program requires 33 FORTRAN 77 routines. Figure 1 shows a block flow diagram of a simulation of an atmospheric contamination control case. An operation description for using this program includes instructions for preparing input data files and operating the program.

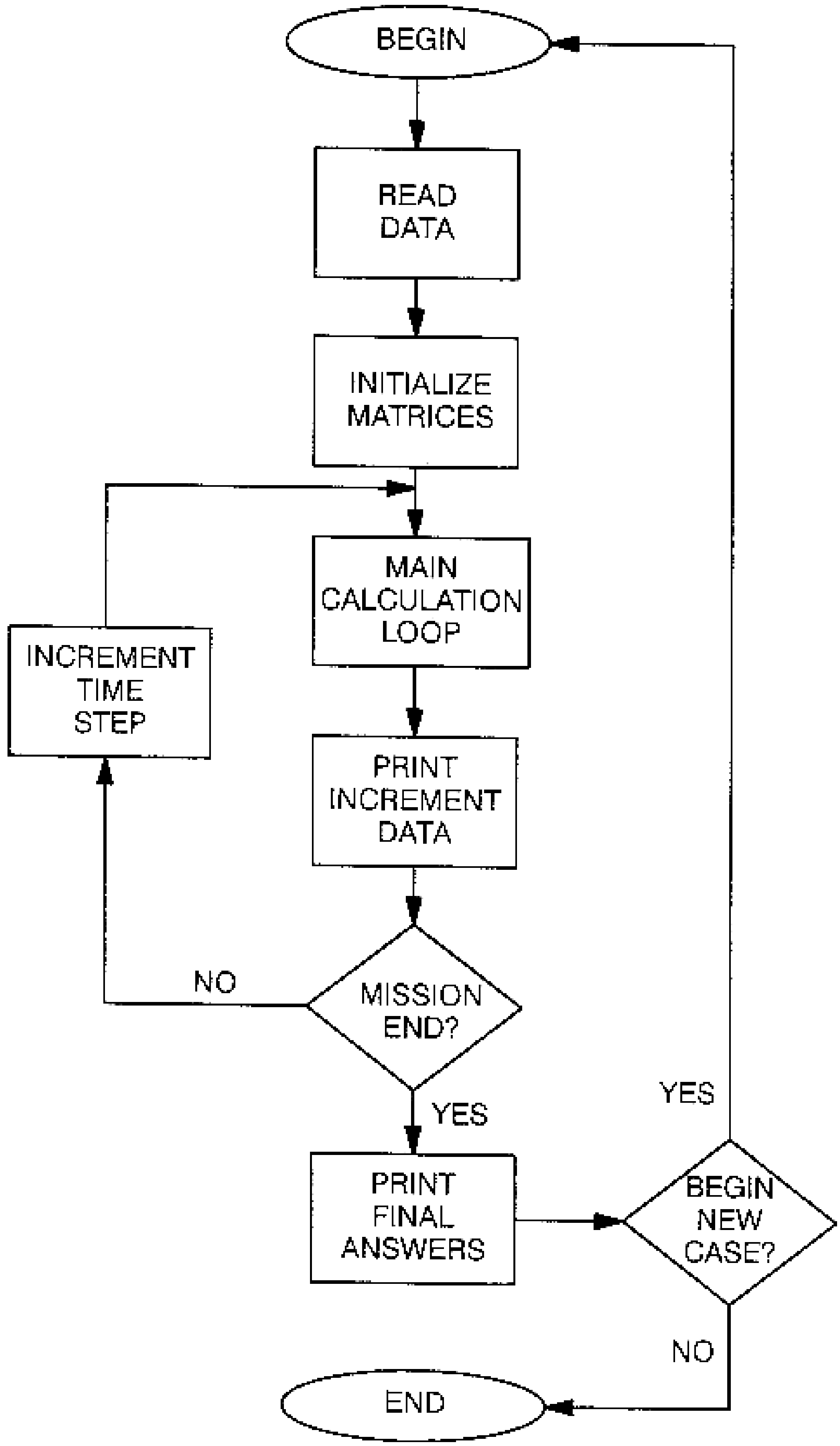


Figure 1. Simplified computer program flow chart.

Input Data File Preparation

Before program operation, three input files must be prepared to define the contaminant properties and generation rates, removal device configuration, and time dependent inputs. These files are created using commercially available spreadsheet software which is capable of producing space delimited ASCII files. Lotus 123 is the preferred spreadsheet program for creating these input files. The print-to-file option in Lotus is used to send the appropriate data range to each file. These files are read by the program, and the data is placed into calculation matrices NN, CDI, DD, and TT. Matrix NN is a 1 by 300 matrix containing the contaminant names, CDI is a 23 by 300 matrix containing contaminant physical properties defined in table 2, matrix DD is a 15 by 22 matrix containing removal device design and configuration data defined in tables 3 and 4, and matrix TT is a 7 by 300 matrix containing time dependent contaminant generation rate and device flow rate data. The matrix sizes are adjustable; however, the personal computer version is limited to 300 contaminants so that the contaminant matrix does not exceed 64 kilobytes to provide a more rapid run time.

Table 2. Required contaminant physical properties and generation data.

Contaminant Property	Units and Remarks
Contaminant name	30 charcter maximum
Cabin generation rate	mg/h
Liquid density	gm/cm^3 (cabin conditions)
Molar volume	cm^3/gram-mole (cabin conditions) (0 = no charchoal removal)
Molecular weight	gm/gram-mole
Vapor pressure	mg/m^3 (cabin conditions)
Henry's Law constant	atm/mole-fraction
Lithium hydroxide removal	lb_m LiOH/lb_m contaminant removed (0 = no removal)
Chemical category	According to NHB 8060.1B Appendix D
Maximum allowable concentration	mg/m^3
Generation rate in devices 3 to 15	mg/h
Degree of oxidation in a high-temperature oxidizer	0 = none; 1 = fully

Table 3. Example device definition table.

USE RANGE 'LIST' DATA TABLE TO PRINT TO PRINTER
USE RANGE 'DATA' TO PRINT DATA TO FILE

TCCS PROGRAM INPUT DATA
DEVICE DEFINITION TABLE

MATRIX COLUMN NO.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DEVICE NO.	DEVICE FLOW (m3/h)	DEVICE TYPE NO.	UPSTREAM DEVICES BY NUMBER NO.1	NO.2	NO.3	NOT FOR INPUT	COL 8 TO 16 USED FOR DEVICE SPECIFIC INPUTS SEE TABLE BELOW								
1	0.00	1	0	0	0	0	0	227.86	293	24	1.00E-03	1.2	0	0	0
2	0.0460	2	1	0	0	0	1	0	0	0	0	0	1	1	0
3	519.90	8	1	0	0	0	1	1.451	0	0	0	0	0	0	0
4	15.12	3	1	0	0	0	1	0.580	0.320	0	490	2	0	0	0
5	4.08	7	4	0	0	0	1	0.000	0.000	0	0	0	0	0	0
6	4.08	5	5	0	0	0	1	0.234	0.129	0	442	0	0	0	0

Table 4. Device definition data by device type for data table columns 8 through 16.

Device Number	Device Type	Device Definition Table Column Number 8	9	10	11	12	13	14	15	16
1	Cabin	0	Volume (m^3)	Temperature (K)	Time Increment (h)	Converge Error (decimal)	Coexist Factor (1.20)	0	0	0
2	Leakage	1	Print Switch No.1	Print Switch No.2	Print Switch No. 3	Print Switch No. 4	Print Switch No. 5	Print Switch No. 6	Print Switch No. 7	Print (3) Switch No. 8
								Regeneration Data		
3	Axial Charcoal	Maximum Efficiency (decimal)	Bed Length (m)	Bed Diameter (m)	0	Matl. Bulk Density (kg/m^3)	Bed Treatment Type (4)	First (h)	Interval (h)	Duration (h)
								Regeneration Data		
4	Radial Charcoal	Maximum Efficiency (decimal)	Bed Length (m)	Bed Outside Diameter (m)	Bed Inside Diameter (m)	Matl. Bulk Density (kg/m^3)	Bed Treatment Type (4)	First (h)	Interval (h)	Duration (h)
							Changeout Data			
5	LiOH Bed	Maximum Efficiency (decimal)	Bed Length	Bed Diameter (m)	0	Matl. Bulk Density (kg/m^3)	First (h)	Interval (h)	0 (1)	0 (2)
6	CO Oxidizer	Maximum Efficiency (deciman)	Bed Length (m)	Bed Diameter (m)	0	0	0	0	0	0
7	Catalytic Oxidizer	1	0	0	0	0	0	0	0	0
8	Condensing Heat Exchanger	1	Condensate Flow (kg/h)	0	0	0	0	0	0	0
9	Dummy Device	0	0	0	0	0	0	0	0	0

NOTES:

1. Used to store total weight of LiOH used in the calculation in kg/h
2. Used to store total weight of LiOH used during the calculation in kg
3. Print switches are set to 1 for printing and 0 for no printing. No. 1 to 5 for diagnostics, 6 to 8 for printout.
4. 0 = no treatment; 1 = chromate treatment for formaldehyde; 2 = phosphoric acid treatment for ammonia

Contaminant Data Input File Preparation

The contaminant data input file contains the contaminant names, physical properties, generation rates in the cabin and each device, and the degree of oxidation in a high temperature catalytic oxidation process. These data are read into program calculation matrices NN and CDI. Sample input is shown in figure 2. Data required for each column of the contaminant data input file are the following:

1. The contaminant sequential number, 1 through 300
2. An apostrophe to enclose the contaminant name as required by FORTRAN 77
3. The contaminant name up to 30 characters
4. An apostrophe to enclose the contaminant name
5. The contaminant cabin generation rate in milligrams per hour (mg/h)
6. The contaminant liquid phase density in grams per cubic centimeter (g/cm^3)
7. The contaminant molar volume in cubic centimeters per gram-mole ($cm^3/gmole$)
8. The contaminant molecular weight in grams per gram-mole (g/gmole)
9. The contaminant vapor pressure at the cabin temperature expressed in milligrams per cubic meter (mg/m^3)
10. The water solubility for the contaminant expressed as the Henry's Law coefficient in atmospheres per mole fraction (atm/mole fraction)—enter a zero if the compound is insoluble or no Henry's Law coefficient is available
11. The LiOH removal expressed as the weight of LiOH consumed per weight of contaminant removed in lb_m LiOH/lb_m contaminant
12. The chemical group classification for the contaminant specified in appendix D of NHB 8060.1B
13. The SMAC or other threshold value expressed in milligrams per cubic meter (mg/m^3)
14. The contaminant generation rate at the inlet to device number 3 in mg/h
15. The contaminant generation rate at the inlet to device number 4 in mg/h
16. The contaminant generation rate at the inlet to device number 5 in mg/h
17. The contaminant generation rate at the inlet to device number 6 in mg/h
18. The contaminant generation rate at the inlet to device number 7 in mg/h

CONTAMINANT INPUT DATA TABLE (Spacelab Missions 1 and 3 basis using maximum generation rates)

MAKE SURE ALL CELLS HAVE A VALUE OR 0 IN CELL

CONT NO.	NAME(30 CHR MAX)	CABIN RATE mg/h	LIQUID DENSITY g/cc	MOLAR VOL cc/mol	MOLAR MASS g/mol	VAPOR CONC mg/m3	H2O SOL HENRY atm,0=i	LIOH lb LiOH/ lb cont	NHB 8060.1B GROUP	SMAC mg/m3	GENERATION IN DEVICES 3 THROUGH 15 mg/h													OXIDIZER EFF decimal
1	1,1,2-triCl-1,2,2-triFlethane	278.99	1.564	120.0	187.40	3280000	2.69E+04	0	7	383.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0.33
2	1,3-Butadiene	0.06	0.678	82.0	54.09	7047000	2.42E+03	0	9	221.20	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
3	Methane	30.81	0.425	37.7	16.04	225400000	3.54E+04	0	9	1771.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0.95
4	2-butanone	45.81	0.805	96.7	72.11	405400	1.63E-01	0	11	59.00	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
5	1-benzo[b]pyrrole	1.04	1.220	139.8	117.15	74	2.48E-02	0	15	0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
6	methyl hydrazine	0.03	0.875	53.0	46.07	1233000	2.70E-02	0	15	0.08	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
7	Ammonia	53.49	0.689	25.0	17.00	7522000	9.37E-01	0	16	17.40	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
8	Carbon monoxide	24.04	0.803	22.1	28.01	974700000	6.34E+04	0	16	28.60	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
9	Hydrogen	4.65	0.071	3.0	2.02	2033000	6.52E+04	0	16	247.30	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00

CONTAMINANT INPUT DATA FILE EXAMPLE

' 1,1,2-triCl-1,2,2-triFlethane	278.99	1.564	120.0	187.40	3280000	2.69E+04	0	7	383.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0.33
' 1,3-Butadiene	0.06	0.678	82.0	54.09	7047000	2.42E+03	0	9	221.20	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
' Methane	30.81	0.425	37.7	16.04	225400000	3.54E+04	0	9	1771.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0.95
' 2-butanone	45.81	0.805	96.7	72.11	405400	1.63E-01	0	11	59.00	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
' 1-benzo[b]pyrrole	1.04	1.220	139.8	117.15	74	2.48E-02	0	15	0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
' methyl hydrazine	0.03	0.875	53.0	46.07	1233000	2.70E-02	0	15	0.08	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
' Ammonia	53.49	0.689	25.0	17.00	7522000	9.37E-01	0	16	17.40	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
' Carbon monoxide	24.04	0.803	22.1	28.01	974700000	6.34E+04	0	16	28.60	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
' Hydrogen	4.65	0.071	3.0	2.02	2033000	6.52E+04	0	16	247.30	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00

NOTE: THE CONTAMINANT INPUT DATA FILE IS CREATED BY WRITING THE SELECTED RANGE TO A FILE FROM THE SPREADSHEET. THIS FILE IS IN ASCII FORMAT. THE CONTAMINNAT NUMBER IS NOT SELECTED FOR THE INPUT DATA FILE.

Figure 2. Sample contaminant input data table and input file.

19. The contaminant generation rate at the inlet to device number 8 in mg/h

20. The contaminant generation rate at the inlet to device number 9 in mg/h

21. The contaminant generation rate at the inlet to device number 10 in mg/h

22. The contaminant generation rate at the inlet to device number 11 in mg/h

23. The contaminant generation rate at the inlet to device number 12 in mg/h

24. The contaminant generation rate at the inlet to device number 13 in mg/h

25. The contaminant generation rate at the inlet to device number 14 in mg/h

26. The contaminant generation rate at the inlet to device number 15 in mg/h

27. The decimal oxidation efficiency per pass for the contaminant in a high temperature catalytic oxidizer where 1.0 is fully oxidized and 0.0 is no oxidation.

Some general considerations should be noted when preparing the contaminant data input file. They are the following:

1. The molecular weights for ammonia, formaldehyde, carbon monoxide, and hydrogen must be exactly 17.0, 30.03, 28.10, and 2.02 since they are used as flags in the program

2. The entire set of information, columns 1 through 27, must be input for each individual contaminant

3. Each cell in all matrices (contaminant, device, and time) must contain a value or a zero

4. The column width in all matrices (contaminant, device, and time) should be sufficient to permit at least one blank space between the column entries

5. Printing to a file should be unformatted and the range should exclude the first column (contaminant number) and any column headings

6. The file must be created in one piece without any line feeds (except at the end of a line) or carriage returns and the left margin must be set to 0 and the right margin to 240 when using Lotus 123

7. The contaminant generation rates at the inlets of devices (columns 14 through 26) are for cases where contaminants are generated in a closed or confined space and are passed through a removal device(s) before entering the cabin (for example, a contaminant is produced in a experimental working volume and is passed through a charcoal bed before entering the cabin)

8. Some contaminant physical property data may not be available in the literature and must be estimated using correlations.

Removal Device Definition Input File Preparation

The device definition matrix, DD, contains the information which defines the removal device configuration. This information includes the type of devices, their arrangement, flow rates, and physical size or dimensions. Basic system data; such as the cabin volume, temperature, basic time increment size, convergence error specification, and the multicomponent adsorption coexistence factor; are also contained in this matrix. In addition, program data output for standard formatted data is controlled from this matrix. A sample device definition file is shown by figure 3. Data required for each of the first seven columns are the following:

1. The device number ranging from 1 through 15
2. The standard volumetric atmospheric flow through the device in cubic meters per hour (m^3/h)
3. Device type number as defined in table 5
4. Upstream device number 1 by device number (column 1)
5. Upstream device number 2 by device number
6. Upstream device number 3 by device number
7. A zero is entered in this column for all devices since this column is used to store calculated values during the program run.

Columns 8 through 16 are specific for each device type. Data required for these columns for each device type are the following:

Device Type Number 1: The Cabin

8. Device decimal removal efficiency equal to zero
9. Cabin atmospheric volume in cubic meters (m^3)
10. Cabin absolute temperature in kelvin (K)
11. Basic time increment in hours (h)
12. Convergence error as decimal percent (recommended values of 0.01 or 0.001)
13. Coexistence factor, approximate value of 1.20, to account for interaction or blockage of contaminants in granular activated charcoal beds
14. Not used for input therefore zero is input
15. Not used for input therefore zero is input
16. Not used for input therefore zero is input.

Example Device Definition Data Table

TCCS PROGRAM INPUT DATA
DEVICE DEFINITION TABLE

USE RANGE 'LIST' DATA TABLE TO PRINT TO PRINTER
USE RANGE 'DATA' TO PRINT DATA TO FILE

MATRIX COLUMN NO.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DEVICE NO.	DEVICE FLOW (m3/h)	DEVICE TYPE NO.	UPSTREAM DEVICES BY NUMBER NO. 1	NO. 2	NO. 3	NOT FOR INPUT	COL 8 TO 16 USED FOR DEVICE SPECIFIC INPUTS SEE TABLE BELOW								
1	0.00	1	0	0	0	0	0	227.86	293	24	1.00E-01	1.2	0	0	0
2	0.0460	2	1	0	0	0	1	0	0	0	0	0	1	1	0
3	519.90	8	1	0	0	0	1	1.451	0	0	0	0	0	0	0
4	15.12	3	1	0	0	0	1	0.580	0.320	0	490	2	0	0	0
5	4.08	7	4	0	0	0	1	0.000	0.000	0	0	0	0	0	0
6	4.08	5	5	0	0	0	1	0.234	0.129	0	442	0	0	0	0

Example Device Definition Input File

1	0.00	1	0	0	0	0	0	227.86	293	24	1.00E-01	1.2	0	0	0
2	0.0460	2	1	0	0	0	1	0	0	0	0	0	1	1	0
3	519.90	8	1	0	0	0	1	1.451	0	0	0	0	0	0	0
4	15.12	3	1	0	0	0	1	0.580	0.320	0	490	2	0	0	0
5	4.08	7	4	0	0	0	1	0.000	0.000	0	0	0	0	0	0
6	4.08	5	5	0	0	0	1	0.234	0.129	0	442	0	0	0	0

Figure 3. Sample device definition data.

Table 5. Device type number and description

Type Number	Description
1	Cabin
2	Cabin leakage
3	Axial flow charcoal bed
4	Radial flow charcoal bed
5	Lithium hydroxide bed
6	Carbon monoxide oxidizer
7	High temperature catalytic oxidizer
8	Condensing heat exchanger
9	Dummy device

Device Type Number 2: Cabin Leakage

8. Device decimal removal efficiency equal to 1.0.

Columns 9 through 16 for the leakage device are used to set standard formatted data output print switches. These columns contain either zero (0) to tell the program not to print or one (1) to tell the program to print. Print switches 1 through 5 are for diagnostic data output and print switches 6 through 8 are for final data output. Typically, print switches 1 through 5 are set to zero during a normal simulation run.

9. Set print switch 1 to zero or one to regulate printing the contents of matrices CC and DD at the end of the precalculation setup routine, PCSET

10. Set print switch 2 to zero or one to regulate printing of the one-tenth increment convergence routine for one time increment for one contaminant (these results will be printed only if the one-tenth increment routine is used)

11. Set print switch 3 to zero or one to regulate printing of the convergence values of predicted and calculated contaminant concentration for the convergence routine CONVRG for the standard and one-twentieth time increments (useful if convergence problems occur)

12. Set print switch 4 to zero or one to regulate printing of the results for one contaminant from the main calculation routine, MCALC, after the average concentration is calculated (matrix DD and part of matrix CC for one contaminant)

13. Set print switch 5 to zero or one to regulate printing the contents of matrices DD and CC at the end of each time increment

14. Set print switch 6 to zero to print only the cabin concentration for each contaminant for each time increment or to one to print the contaminant removal rate, mass removed, and device efficiencies in addition to the cabin concentration for each time increment

15. Set print switch 7 to zero to allow form feeds between output pages or to one to suppress form feeds between output pages

16. Set print switch 8 to zero to print calculation results specified by print switch 6 at the end of the run only or to one to print calculation results at the end of each time increment.

<u>Device Type Number 3: Axial Flow Charcoal Bed</u>

8. Maximum device decimal removal efficiency equal to 1.0

9. Charcoal bed length in meters (m)

10. Charcoal bed diameter in meters (m)

11. Not used for input and set to 0

12. Charcoal bulk density in kilograms per cubic meter (kg/m^3)

13. Charcoal bed treatment type (1 for chromate treated charcoal for formaldehyde removal, 2 for phosphoric acid treated charcoal for ammonia removal, or 0 for untreated charcoal)

14. Time of first regeneration in hours (h)

15. Time interval between each regeneration in hours (h)

16. Regeneration duration in hours (h).

<u>Device Type Number 4: Radial Flow Charcoal Bed</u>

8. Maximum device decimal removal efficiency equal to 1.0

9. Radial flow bed length in meters (m)

10. Radial flow bed outside diameter in meters (m)

11. Radial flow bed inside diameter in meters (m)

12. Charcoal bulk density in kilograms per cubic meter (kg/m^3)

13. Charcoal bed treatment type (1 for chromate treated charcoal for formaldehyde removal, 2 for phosphoric acid treated charcoal for ammonia removal, or 0 for untreated charcoal)

14. Time of first regeneration in hours (h)

15. Time interval between each regeneration in hours (h)

16. Regeneration duration in hours (h).

Device Type Number 5: LiOH Bed

8. Maximum device decimal removal efficiency equal to 1.0
9. LiOH bed length in meters (m)
10. LiOH bed diameter in meters (m)
11. Not used for input and set to 0
12. LiOH bulk density in kilograms per cubic meter (kg/m^3)
13. Time of first changeout in hours (h)
14. Time between changeouts in hours (h)
15. Not used for input and set to 0
16. Not used for input and set to 0.

Device Type Number 6: Ambient-Temperature CO Oxidizer

8. Maximum device decimal removal efficiency equal to 1.0
9. Oxidizer bed length in meters (m)
10. Oxidizer bed diameter in meters (m)
11. Not used for input and set to 0
12. Not used for input and set to 0
13. Not used for input and set to 0
14. Not used for input and set to 0
15. Not used for input and set to 0
16. Not used for input and set to 0.

Device Type Number 7: High-Temperature Catalytic Oxidizer

8. Maximum device decimal removal efficiency equal to 1.0
9. Not used for input and set to 0
10. Not used for input and set to 0
11. Not used for input and set to 0

12. Not used for input and set to 0

13. Not used for input and set to 0

14. Not used for input and set to 0

15. Not used for input and set to 0

16. Not used for input and set to 0.

Device Type Number 8: Condensing Heat Exchanger

8. Maximum device decimal removal efficiency equal to 1.0

9. Condensate flow rate in kilograms per hour (kg/h)

10. Not used for input and set to 0

11. Not used for input and set to 0

12. Not used for input and set to 0

13. Not used for input and set to 0

14. Not used for input and set to 0

15. Not used for input and set to 0

16. Not used for input and set to 0.

Device Type Number 9: The Dummy Device

The dummy device serves to set the flow rate through a line when combining or splitting flows between upstream and downstream devices. Legal and illegal dummy device use is illustrated at the bottom of figure 4. When combining or splitting streams, dummy devices are required to set the upstream flow rates being combined or split.

8. Maximum device decimal removal efficiency equal to 0

9. Not used for input and set to 0

10. Not used for input and set to 0

11. Not used for input and set to 0

12. Not used for input and set to 0

13. Not used for input and set to 0

14. Not used for input and set to 0

15. Not used for input and set to 0

16. Not used for input and set to 0

Removal Device Arrangement Rules

Some simple rules apply to arranging removal devices in the removal device definition file. These rules are the following:

1. The first device must always be the cabin device type 1

2. The second device must always be the leakage device type 2

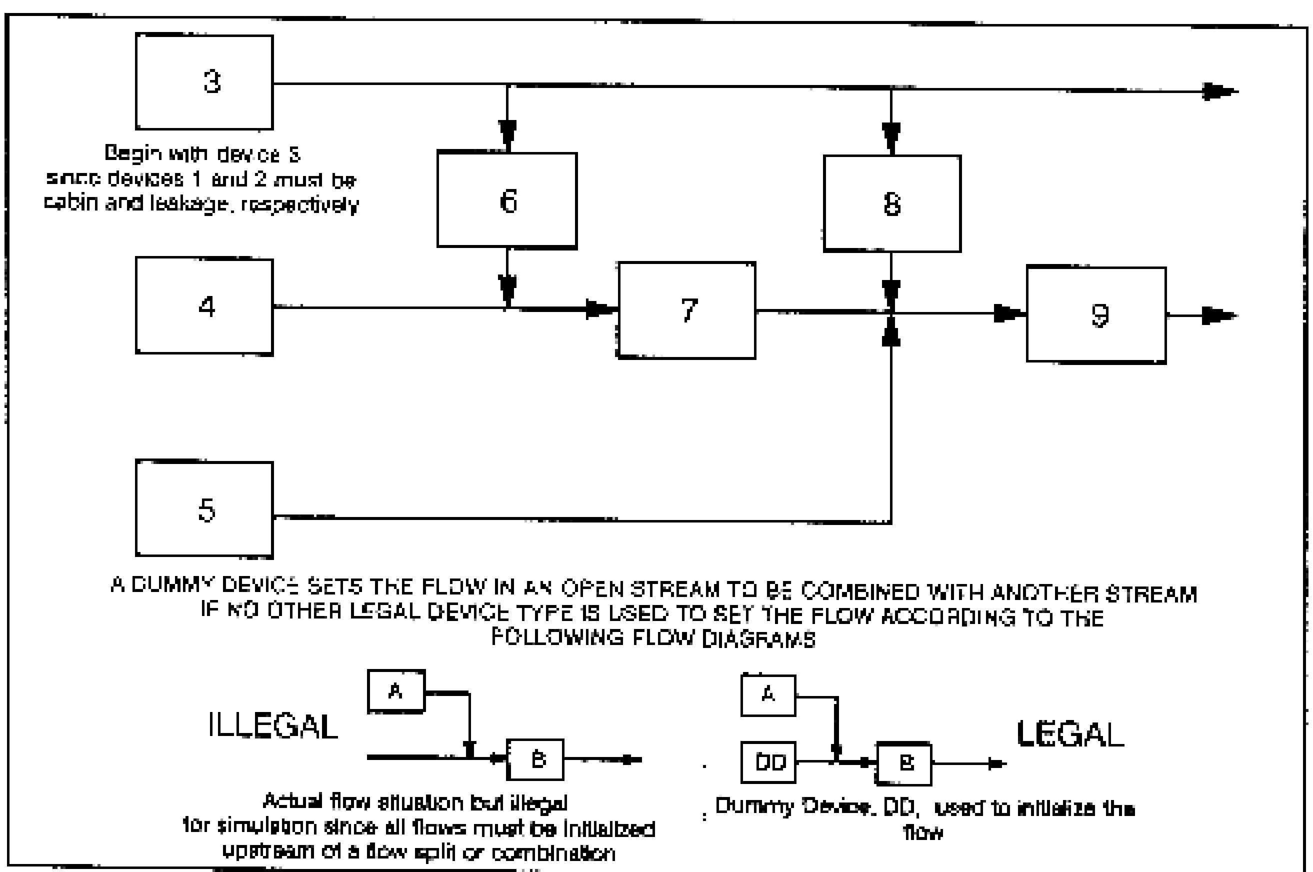
3. Device numbers 3 through 15 may be any legal device type

4. Devices must be numbered so that calculations for upstream devices occur before calculations for downstream devices since calculations are sequential by the device number specified in column 1

5. Inlet flow rates must be specified for each device to allow proper inlet concentration calculation for split flows

6. If the cabin is upstream of any device, column 4 should indicate this by entering a 1.

Figure 4 shows a sample device configuration along with some legal and illegal device configurations.



NOTE: NUMBERS INDICATE THE DEVICE NUMBER, NOT THE DEVICE TYPE AND BEGIN WITH 3 SINCE CABIN AND LEAKAGE ARE DEVICE NUMBERS 1 AND 2 FOR ANY SIMULATION.

Figure 4. Sample device configurations and dummy device use.

Removal Device Regeneration

Regeneration of the axial and radial flow charcoal beds and the LiOH bed are accomplished by entering the appropriate regeneration information into matrix DD. Columns 14, 15, and 16 are used for regenerating the axial and radial charcoal beds and should contain the first time in hours from the beginning of the run that regeneration occurs (note that the run begins at zero hours and the run time increments itself according to the basic time increment), the time in hours between each regeneration, and the time required to regenerate the bed in hours. Columns 13 and 14 are used for LiOH bed regeneration and should contain the first changeout time in hours and the time between each change out in hours, respectively. No regeneration duration is required for the LiOH bed. It is important that the time between each regeneration or changeout and the regeneration time must be equal to or an even multiple of the basic time increment for the calculation. If they are not, the program will truncate them. Rules to be followed when simulating a regenerable bed are the following:

1. The initial regeneration time, changeout or regeneration interval, and time to regenerate must be multiples of the basic time increment specified in the device definition matrix row 1 column 11

2. If the values are not even multiples of the time increment, they will be truncated to the next smallest multiple which could be zero

3. If the regeneration duration desired is less than one time increment, the input flow rate can be modified manually in the program input according to the following:

$$Q_{\text{mod}} = Q_{\text{actual}} \times \{1 - (\tau_{\text{reg duration}}/\tau_{\text{reg interval}})\} \quad (1)$$

   The modified flow rate should be input in the device definition matrix column 2 for that device. the regeneration duration should then be set to zero. If the flow rate is changed during the simulation, it may be necessary to add a dummy device in parallel with the regenerating device so that the original flow rate may be supplied to any possible downstream devices

4. For no regeneration, set the regeneration interval equal to zero.

Time-Dependent Data Input File Preparation

During a simulation run, it is possible to change the generation rate of any contaminant both in the cabin or for any device which is operating. Device flow rates and other device information such as the cabin volume, the cabin temperature, and basic time increment may also be changed. Time-dependent changes are accomplished by using the time dependent data matrix, TT. This matrix contains seven columns for each row. Each line in matrix TT does not allow simultaneous contaminant generation rate and device definition data changes. For example, the flow rate and contamination generation rate cannot be changed using a single line. Each change must have its own unique change line. However, two simultaneous device definition changes may be made for a single removal device using one time change file line. Table 6 shows the heading for the time-dependent data table and six common cases for time-dependent data changes. An example time-dependent data file is shown by the seven columns under the data heading of table 7. The time-dependent data file read by the program is created by selecting only the time change numbers when writing the data to the file. The

Table 6. Several common time-dependent changes.

Type of Change	Change Items						
Time-Dependent Data Table Sample Heading	FILE:TCTT02 28-May 1992 TIME DEPENDENT CONTAMINANT INPUT DATA FILE (MAT TT) MAKE SURE ALL CELLS HAVE VALUE OR 0 IN CELL USE RANGE 'LIST' TO PRINT TO PRINTER USE RANGE 'DATA' TO PRINT TO FILE ==	CHANGE CONT GEN IN MAT CDI <----------------------->		CHANGE DEVICE FLOW RATE IN MAT DD <------------------>		CHANGE OTHER MAT DD INFO <-------------->	
	CHANGE TIME (h)	CONT NO.	CONT GEN RT (mg/h)	DEVICE NO.	NEW Q DEVICE (m^3/h)	MAT DD COLUMN NO.	NEW VALUE
1. No changes	0	0	0	0	0	0	0
2. Change contaminant generation rate	24	1	800	1	-1	-1	-1
3. Change device flow rate	24	-1	-1	3	15	-1	-1
4. Change cabin volume	24	-1	-1	1	-1	9	150
5. Change device flow rate and bed length	24	-1	-1	3	15	9	1.5
6. Change time increment	24	-1	-1	1	-1	11	1

Table 7. A sample time-dependent data input file.

Change Type	Data to be Read by the Computer Program						
Change increment	720	-1	-1	1	-1	11	1
Change flow	720	-1	-1	3	4.8	-1	-1
Change flow	720	-1	-1	4	4.8	-1	-1
Change flow	720	-1	-1	5	4.8	-1	-1
Change increment	732	-1	-1	1	-1	11	6
Change increment	768	-1	-1	1	-1	11	5
Change increment	773	-1	-1	1	-1	11	23
Change flow	773	-1	-1	3	0	-1	-1
Change flow	773	-1	-1	4	0	-1	-1
Change flow	773	-1	-1	5	0	-1	-1
Change increment	796	-1	-1	1	-1	11	24
Change increment	844	-1	-1	1	-1	11	2
Change flow	846	-1	-1	3	4.8	-1	-1
Change flow	846	-1	-1	4	4.8	-1	-1
Change flow	846	-1	-1	5	4.8	-1	-1
Change flow	846	-1	-1	6	281	-1	-1
Change generation	846	5	0.1274	1	-1	-1	-1
Change generation	846	8	3.4825	1	-1	-1	-1
Change generation	846	13	3.5398	1	-1	-1	-1
Change generation	846	14	2.8109	1	-1	-1	-1
Change generation	846	22	0.3228	1	-1	-1	-1
Change generation	846	27	0.6651	1	-1	-1	-1
Change generation	846	121	14.9589	1	-1	-1	-1
Change generation	846	137	4.2861	1	-1	-1	-1
Change generation	846	149	0.0075	1	-1	-1	-1
Change generation	846	153	39.6206	1	-1	-1	-1
Change generation	846	154	6.5515	1	-1	-1	-1
Change generation	846	155	2.1837	1	-1	-1	-1
Change increment	892	-1	-1	1	-1	11	24

heading should not be included in the data file. If it is, the program will write an error message to the screen telling the user that the data file cannot be read. Each time-dependent file may contain up to 300 lines of changes. Rules for using this data file are the following:

General Time-Dependent Change Rule

1. Column 1 contains the change time in hours for all changes. Any update can occur only at the beginning of a time increment. If a change time is not an exact multiple of the basic time increment, the time will be truncated to the next lowest multiple of the increment.

Changing Contaminant Generation Rates

1. Column 1 contains the change time in hours (h)

2. Column 2 contains the contaminant sequential number to be changed. The contaminant number must be greater than or equal to one, and less than or equal to the maximum number of contaminants in the contaminant input file

3. Column 3 contains the new or changed generation rate

4. Column 4 contains the removal device number whose contaminant generation rate is to be changed. The device number must be greater than or equal to 1, and less than or equal to the maximum number of devices in the simulation. (Device 2 may be used, but it does not effect the calculation since any change would be canceled out by the 100-percent removal efficiency provided by cabin leakage.)

5. Column 5 is ignored and is set equal to -1

6. Column 6 is ignored and is set equal to -1

7. Column 7 is ignored and is set equal to -1.

Changing Device Definition Matrix Information

1. Column 1 contains the change time in hours (h)

2. Column 2 is ignored and is set equal to -1

3. Column 3 is ignored and is set equal to -1

4. Column 4 contains the device number and must be greater than or equal to 1 and less than or equal to the maximum number of removal devices

5. Column 5 contains the new device definition matrix information. This information must be greater than zero if a change is desired. If no input is desired, use -1 since this will prevent the program from reading the new information in column 7

6. Column 6 contains the device definition matrix column number that the change is to occur and must be greater than or equal to zero and less than or equal to 16. Any other value in this column causes the program to ignore the value in column 7

7. Column 7 contains the new device definition matrix value. Both the flow rate and any other device change can be made simultaneously for any device.

A time-dependent data file is necessary for program execution even when no time dependent changes will be made. In this case, a single row with all the columns set equal to zero is required for the time-dependent data file.

IBM and Compatible Program Version Execution

Once the three input files have been prepared, the TCCS computer program is ready to be executed. Execution is initiated by typing TCCS at the DOS prompt. If a very large contaminant data file will be used for the simulation and contaminant plot data will be obtained during the run, the program should be executed by typing TCCS /R 10000 to allow for larger output records during the run. The number used with the "/R" option can range between 512 and 65,527 and corresponds to record lengths of 512 bytes to 65,527 bytes. Normal execution without the "/R" option results in the default record length of 1,024 bytes.

It is important that the terminal keyboard is in the CAPS LOCK mode since the program only reads capitalized alphanumeric during interactive input. A carriage return is entered from the keyboard after responding to each prompt. If inappropriate information is entered, or a file with the incorrect format is specified, the program will indicate that an error reading the file has occurred and will prompt the user for the desired information again.

Reading Input Data

The program begins by prompting the user for the names of the input files to be used for the simulation. A prompt is written to the screen asking for each file name beginning with the contaminant data input file. The file name is entered by typing the file path, the filename, and its extension. This allows the program to read the files if they are stored on the computer disk in a separate directory or allows the program to access the files from a floppy diskette. If the files are in the same directory as the program executable file, then no path designation is necessary. The removal device definition file is read next, followed by the time-dependent data file.

After each file has been read successfully, the program writes a message to the screen indicating that the data file has been read. Also, after each file has been read, the program prompts the user for a decision concerning input file data output by writing a prompt to the screen requesting whether the data file that has just been read should be printed. The user should respond to this prompt with a "Y" or an "N." The "Y" response results in a second prompt for the user to designate how the data file should be printed. The choices are the printer (LPT1), the computer screen (CON), or neither (END). This option is available for the user to view the contents of each data file to make sure the data are correct before the program calculation routine begins. Usually, this option is not necessary if the input files have been created carefully. The "N" response causes the program to move to the next input data prompt.

Once data file input is completed, the program prompts the user for the mission duration in hours. This is the length of time that the particular mission being simulated will last. For example, a 7-day Spacelab mission duration is 168 h and the number 168.0 should be entered at this prompt.

Data Output Decisions

After the data files and mission duration have been read by the program, the user is prompted for a decision on calculation data output. The TCCS computer program is capable of producing output data in several formats and writing this data to multiple output devices. The principle output data consist of standard formatted output data controlled by the print switches defined in the device definition data file and plot data output which is specified interactively by the user for each program execution. The program prompts the user for decisions concerning plot data output and output devices in the following order.

Concentration and Device Efficiency Plot Data Output. Concentration data and removal device efficiency data for each basic time increment can be output to two separate data files. The user is prompted by the program for a decision to output only concentration data, only efficiency data, both concentration and efficiency data, or neither. The user selects these options by entering "C," "E," "B," or "N," respectively, at the prompt. Examples of the data output for both concentration and efficiency plot data are shown by tables 8 and 9.

Table 8. Example concentration plot data output.

Initial Time (h)	Final Time (h)	Concentration (mg/m^3)				
		Cont. 1	Cont. 2	Cont. 3	Cont. 4	Cont. n
0.000	1.000	1.189	0.2432E–03	0.1339	0.1899....	0.2022E–1
1.000	12.000	10.16	0.2077E–02	1.464	1.308	0.2201
12.000	24.000	14.72	0.3012E–02	2.655	1.637	0.3972

Toxic Hazard Index Output. The toxic hazard index is a technique used by toxicologists to determine the toxicological acceptability for a mixture of chemical compounds in the atmosphere. It is analogous to an SMAC in some respects. The overall toxic hazard index is the summation of the ratios of each contaminant's atmospheric concentration to its SMAC. Ideally, this summation should be less than one. A more detailed definition of the toxic hazard index is found in appendix B and reference 12.

The user is prompted by the program for a decision to write the toxic hazard index to the standard formatted output device, to the standard output device and a plot data file, or neither. Each option is selected by entering "Y," "P," or "N" at this prompt.

Standard Formatted Output. Standard formatted calculation data produced by the program can be output to a datafile, the printer, the screen, or none of these options. The datafile option requires the user to input the path, filename, and extension of a file that the program will open and write calculation data to during the execution. Again, the path is necessary only if the datafile is not in the same directory as the program executable file. The program checks the file name to determine whether the file already exists. If the file exists, the program indicates this on the screen and asks the user whether it should be overwritten. The user responds to this prompt by entering "Y" or "N." The "Y" response allows overwriting the present contents of the file, and the "N" response causes the program to prompt the user for another file name. The printer selection is made by entering LPT1 at this prompt and results in formatted calculation data output to the print device for the computer. This option is recommended only if output is desired at the end of the program execution since large quantities of paper are produced. The screen option is selected by entering CON and results in the formatted data to be written to the screen. The data scrolls by rapidly when the CON option is selected, and scrolling may be stopped by holding down the control key and pressing the S key (CTRL S). Scrolling is restarted by typing CTRL S again. The END option terminates program execution. The program is designed to ask the user if any other cases will be

Table 9. Example removal device efficiency plot data output.

Initial Time	Final Time	Device	Efficiency for Each Contaminant								
			1	2	3	4	5	6	7	8	n
0.00	1.00	2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.00	1.00	3	0.000	0.000	0.000	0.021	0.125	0.116	1.000	0.000	0.000
0.00	1.00	4	1.000	1.000	0.056	1.000	1.000	1.000	1.000	0.000	0.000
0.00	1.00	5	0.330	1.000	0.950	1.000	1.000	1.000	1.000	1.000	1.000
0.00	1.00	6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.00	12.00	3	0.000	0.000	0.000	0.021	0.125	0.116	1.000	0.000	0.000
1.00	12.00	4	1.000	1.000	0.003	1.000	1.000	1.000	1.000	0.000	0.000
1.00	12.00	5	0.330	1.000	0.950	1.000	1.000	1.000	1.000	1.000	1.000
1.00	12.00	6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
12.00	24.00	3	0.000	0.000	0.000	0.021	0.125	0.116	1.000	0.000	0.000
12.00	24.00	4	1.000	1.000	0.002	1.000	1.000	1.000	1.000	0.000	0.000
12.00	24.00	5	0.330	1.000	0.950	1.000	1.000	1.000	1.000	1.000	1.000
12.00	24.00	6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

run before terminating execution. The END option causes this prompt to be written to the screen, and the user responds by entering "Y" or "N." The "Y" response reinitializes the program, and the user is prompted for new input data. The "N" response terminates execution completely.

WARNING: Be careful to specify LPT1, CON, or END when making data output decisions to send data to the printer, the screen, or to reset the program. Any other response will cause the data to be written to a file.

Plot Data File Designation. After the output selections have been made, the program prompts the user for file names for each of the plot data files. A separate file is created for concentration plot data, device removal efficiency plot data, and toxic hazard index plot data. Prompts for each file name are written to the screen only if the user has chosen the options specifying this data for the particular program execution run. As with the standard output device, the user responds to this prompt by entering the path, filename, and extension. The path is not necessary if the file is in the same directory as the program executable file. The program checks to see if the file already exists. If this file exists, the program writes a prompt to the screen asking the user if the file should be overwritten. The user must respond to this prompt by entering a "Y" or "N." The "Y" response opens the file and permits overwriting, while the "N" response causes the program to prompt the user for a new file name.

Calculation Routine Execution

After completing the input and output designations, the program begins the calculation routine. During this routine, information will be written to the computer screen depending on the output selections made. Standard formatted output for each time increment will be written to the screen if print switch 8 in the device definition data file is set to 1 and the CON option for output has been selected. However, if print switch 8 is set to 1 and the LPT1 or file option has been selected, only the increment number and the increment starting and ending times will be written to the screen. Similarly, if print switch 8 is set to 0 and any standard output device selection has been made, the program will write only the increment number and the increment starting and ending times to the screen during the calculation routine. At the end of the calculation routine, the CON output option instructs the program to write the final standard formatted output data to the screen, the LPT1 option sends this data to the printer, and the file option sends the data to the designated file.

During execution, the program may also write error messages to the screen. Most error messages relate to output devices and cause the data to not be written to the specified data file. The program indicates the location of the error in the execution and will continue to execute unless the disk capacity has been exceeded. In this case, the execution terminates.

A second type of warning is written to the screen when the calculation for a contaminant does not converge. This warning indicates a serious problem where the concentration and efficiency calculations for a contaminant did not converge to within the specified convergence error within 20 iterations using the basic time increment and 20 iterations using one-tenth the basic time increment. The program will continue to execute; however, the results for that contaminant will be incorrect. This problem is corrected by either using a smaller basic time increment or increasing the convergence error specified in the device definition data file.

After the calculation routine is complete, the program writes a prompt to the screen asking the user if more cases will be run. The user responds to this prompt by entering "Y" or "N." The "Y" response causes the program to loop to the data entry routine to allow the user to specify input files for the next case. The "N" response terminates the program execution, and the DOS prompt will appear on the screen.

Output Data File Manipulation

The program is capable of producing up to four output data files during a single execution. These files contain the standard formatted output, concentration plot data, device removal efficiency plot data, and toxic hazard index plot data. All of these files are in ASCII format and can be loaded into commercially available word processing and spreadsheet programs for editing and analysis. Specific guidelines for manipulating these files are described.

Standard Formatted Output

The data file containing the standard formatted output may be loaded into a word processing program to be included in a report or memorandum. The format may be edited in the program to suit the user's needs. The font size may have to be reduced to fit on 8.5 by 11 inch paper in the portrait orientation. The landscape orientation is recommended. An example of the standard formatted output is shown by figure 5.

Concentration Plot Data

Concentration plot data consists of the beginning and ending time for each time increment and the concentration for each contaminant in the simulation at the end of the time increment. An example of concentration plot data output is shown by table 8. The data are arranged in columns according to the following:

1. The first column contains the increment beginning time in hours (h)

2. The second column contains the increment ending time in hours (h)

3. All columns beyond the first two contain the sequential (contaminant 1 through n) contaminant concentration for each contaminant in milligrams per cubic meter (mg/m^3).

These data can be loaded into a spreadsheet program (Lotus 123 is recommended) and the appropriate ranges selected for preparing plots of concentration as a function of time.

Efficiency Plot Data

Removal device efficiency data consist of the beginning and ending time for each increment, the removal device number (2 through 15), and the decimal of each device for each contaminant for that time increment. This data file is arranged as a square matrix in which each time increment has 14 rows as shown by table 9. Each row corresponds to a removal device and contains columns which include the following:

1. The first column contains the increment beginning time in hours (h)

1

PROGRAM VERSION 8.0 Alpha 04-10-92

5/27/1992 14:51 SAMPLE.DAT PAGE 1
TIME INCR 1 INITIAL TIME (HRS)= 0.00 FINAL TIME (HRS)= 1.00

CONT NO.	NAME	FINAL CABIN CONC (MG/M3)	MAC	EXCEEDS MAC
1	1,1,2-triCl-1,2,2-triFlethane	1.189	383.0	N
2	1,3-Butadiene	0.2432E-03	221.2	N
3	Methane	0.1339	1771.	N
4	2-butanone	0.1899	59.00	N
5	1-benzo[b]pyrrole	0.3853E-02	0.4800	N
6	methyl hydrazine	0.1093E-03	0.8000E-01	N
7	Ammonia	0.9042E-01	17.40	N
8	Carbon monoxide	0.1046	28.60	N
9	Hydrogen	0.2022E-01	247.3	N

GROUP T-VALUES AS SPECIFIED IN NHB 8060.1B APPENDIX D

-01-	-02-	-03-	-04-	-05-	-06-	-07-	-08-	-09-	-10-	-11-	-12-	-13-	-14-	-15-	-16-
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01

OVERALL T-VALUE
0.01

1

5/27/1992 14:51 SAMPLE.DAT PAGE 2
TIME INCR 1 INITIAL TIME (HRS)= 0.00 FINAL TIME (HRS)= 1.00
RATE OF CONTAMINANT REMOVAL-EACH DEVICE (MG/HR) SHEET 1

NO.	NAME	CABIN	LEAK	DEV3	DEV4	DEV5	DEV6	DEV7	DEV8
1	1,1,2-triCl-1,2,2-triFlethane	270.9	0.2765E-01	0.4108E-04	9.087	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
2	1,3-Butadiene	0.5541E-01	0.5655E-05	0.9341E-07	0.1859E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
3	Methane	30.50	0.3089E-02	0.3489E-05	0.5648E-01	0.2458	0.0000E+00	0.0000E+00	0.0000E+00
4	2-butanone	43.27	0.4452E-02	1.069	1.463	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
5	1-benzo[b]pyrrole	0.8779	0.9381E-04	0.1328	0.3084E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
6	methyl hydrazine	0.2490E-01	0.2652E-05	0.3463E-02	0.8718E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
7	Ammonia	20.60	0.2827E-02	31.96	0.9293	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
8	Carbon monoxide	23.83	0.2413E-02	0.1521E-05	0.7547E-08	0.2140	0.0000E+00	0.0000E+00	0.0000E+00
9	Hydrogen	4.608	0.4666E-03	0.2861E-08	0.1359E-13	0.4138E-01	0.0000E+00	0.0000E+00	0.0000E+00

Figure 5. Example standard formatted data output.

1

5/27/1992 14:51 SAMPLE.DAT PAGE 3
TIME INCR 1 INITIAL TIME (HRS)= 0.00 FINAL TIME (HRS)= 1.00
TOTAL CONTAMINANT MASS REMOVED BY EACH DEVICE (MG) SHEET 1

NO.	NAME	CABIN	LEAK	DEV3	DEV4	DEV5	DEV6	DEV7	DEV8
1	1,1,2-triCl-1,2,2-triFlethane	270.9	0.2765E-01	0.4108E-04	9.087	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
2	1,3-Butadiene	0.5541E-01	0.5655E-05	0.9341E-07	0.1859E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
3	Methane	30.50	0.3089E-02	0.3489E-05	0.5648E-01	0.2458	0.0000E+00	0.0000E+00	0.0000E+00
4	2-butanone	43.27	0.4452E-02	1.069	1.463	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
5	1-benzo[b]pyrrole	0.8779	0.9381E-04	0.1328	0.3084E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
6	methyl hydrazine	0.2490E-01	0.2652E-05	0.3483E-02	0.8718E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
7	Ammonia	20.60	0.2827E-02	31.96	0.9293	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
8	Carbon monoxide	23.83	0.2413E-02	0.1521E-05	0.7547E-08	0.2140	0.0000E+00	0.0000E+00	0.0000E+00
9	Hydrogen	4.606	0.4666E-03	0.2861E-06	0.1359E-13	0.4138E-01	0.0000E+00	0.0000E+00	0.0000E+00

1

5/27/1992 14:51 SAMPLE.DAT PAGE 4
TIME INCR 1 INITIAL TIME (HRS)= 0.00 FINAL TIME (HRS)= 1.00
DEVICE REMOVAL EFFICIENCY AT END OF TIME INCREMENT (DEC)

NO.	NAME	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
1	1,1,2-triCl-1,2,2-triFlethane	1.000	0.000	1.000	0.330	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	1,3-Butadiene	1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	Methane	1.000	0.000	0.056	0.950	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	2-butanone	1.000	0.021	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	1-benzo[b]pyrrole	1.000	0.125	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	methyl hydrazine	1.000	0.116	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7	Ammonia	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	Carbon monoxide	1.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
9	Hydrogen	1.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Figure 5. Example standard formatted data output (continued).

2. The second column contains the increment ending time in hours (h)

3. The third column contains the removal device number (dimensionless)

4. All columns beyond the first three contain the sequential contaminant removal efficiency for that device for each individual contaminant (contaminant 1 through n).

It must be noted that the cabin, designated as device number 1, is not included since cabin removal efficiency is not relevant. These data can be loaded into a spreadsheet program (Lotus 123 is recommended) and then sorted by the removal device number column and the time increment ending time column to obtain a time-dependent removal device efficiency profile for each removal device. The appropriate data ranges may then be selected for producing data plots.

Toxic Hazard Index Data

Toxic hazard index data consists of the beginning time, ending time, and overall toxic hazard index calculation for each time increment. This data as shown by table 10 is arranged in three columns which include the following:

1. The first column contains the increment beginning time in hours (h)

2. The second column contains the increment ending time in hours (h)

3. The third column contains the overall toxic hazard index (dimensionless).

These data can be loaded into a spreadsheet program (Lotus 123 is recommended) and the second and third columns selected as the plot data ranges.

Table 10. Example toxic hazard index plot data output.

Initial Time	Final Time	Hazard Index
0.00	1.00	0.01
1.00	12.00	0.07
12.00	24.00	0.11

Macintosh Program Version Execution

The Macintosh program version execution is much more simple than the IBM and compatible program version. Execution is started by positioning the cursor over the TCCS computer program application icon, shown in figure 6, and double clicking with the mouse. The program displays the program banner which contains the program name, version number, and version date and then prompts for the input data. At this prompt, a standard Macintosh file location window is displayed, as shown in figure 7, and the appropriate file is selected by highlighting the file name with the cursor and double clicking with the mouse on the open bar to the right. Similar windows appear for selecting the device definition and time-dependent data files.

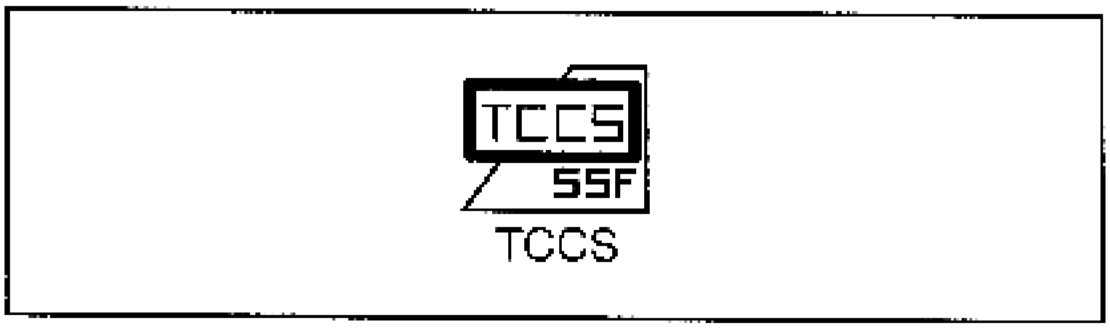


Figure 6. Macintosh program version application icon.

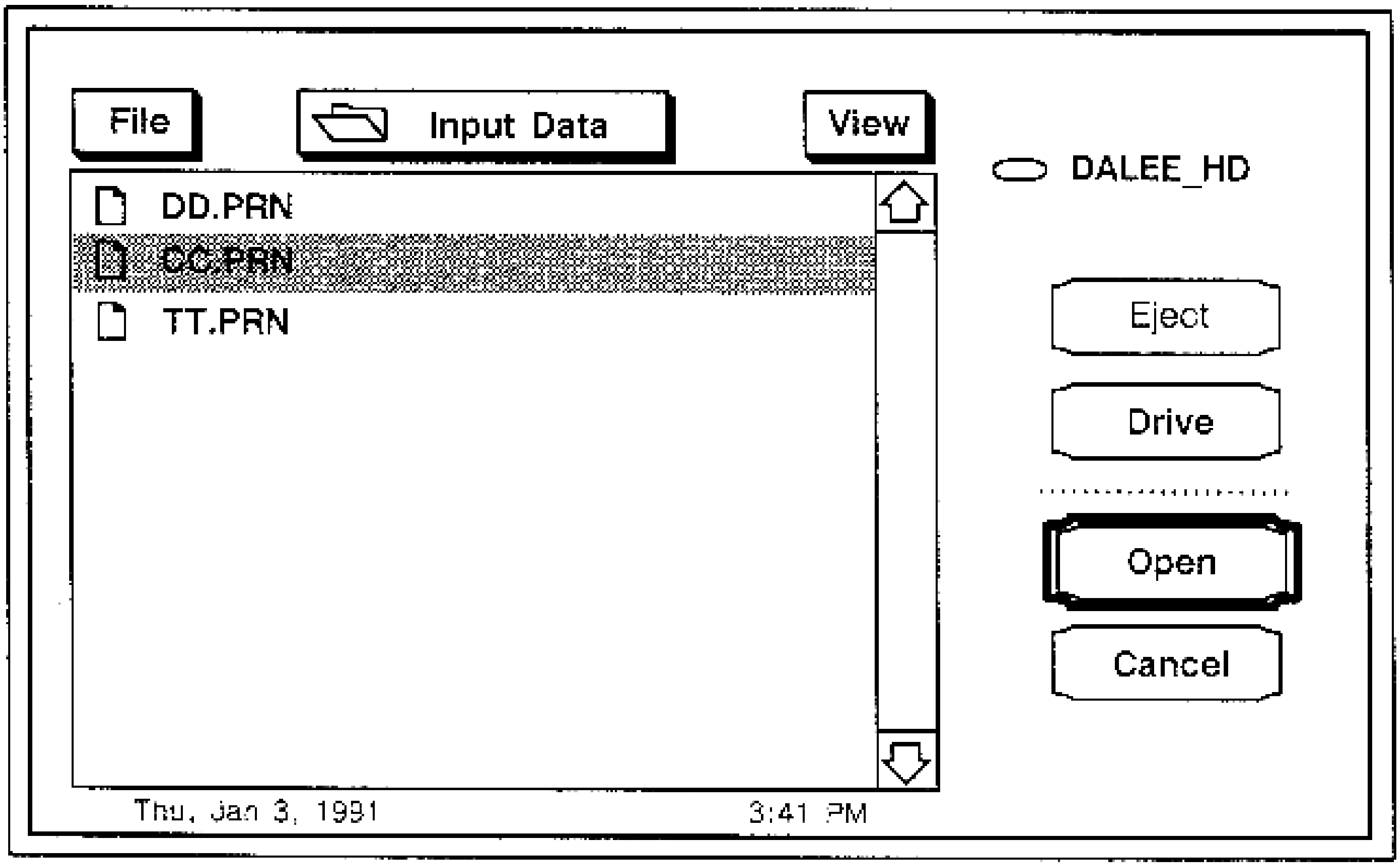


Figure 7. Example Macintosh file selection window.

After the three input files have been located, the main user input screen appears as shown in figure 8. The user enters the mission duration time and selects options to write the runtime data to the screen, files, or by both positioning the cursor in the appropriate box and clicking with the mouse. When the user is finished with this screen, the cursor is positioned over the "OK" box and clicked with the mouse. If any of the data are to be sent to a file, the program prompts for the file name with a standard Macintosh pop-up window similar to the illustration in figure 9.

The final step before the program begins the main calculation routine is to name the plot data output files for concentration and device removal efficiency data. This is accomplished with the standard Macintosh pop-up window shown in figure 9.

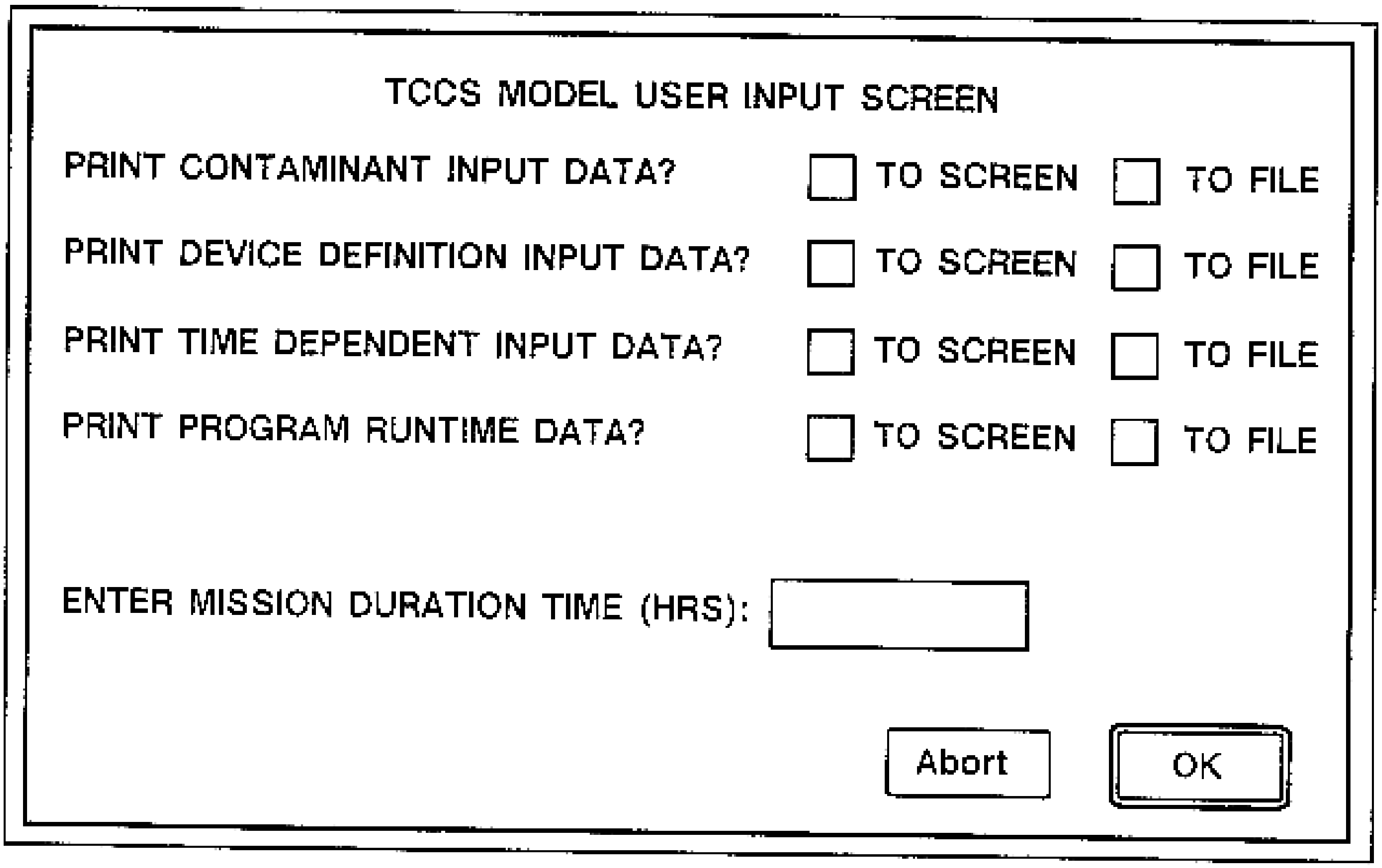


Figure 8. TCCS computer program user input screen.

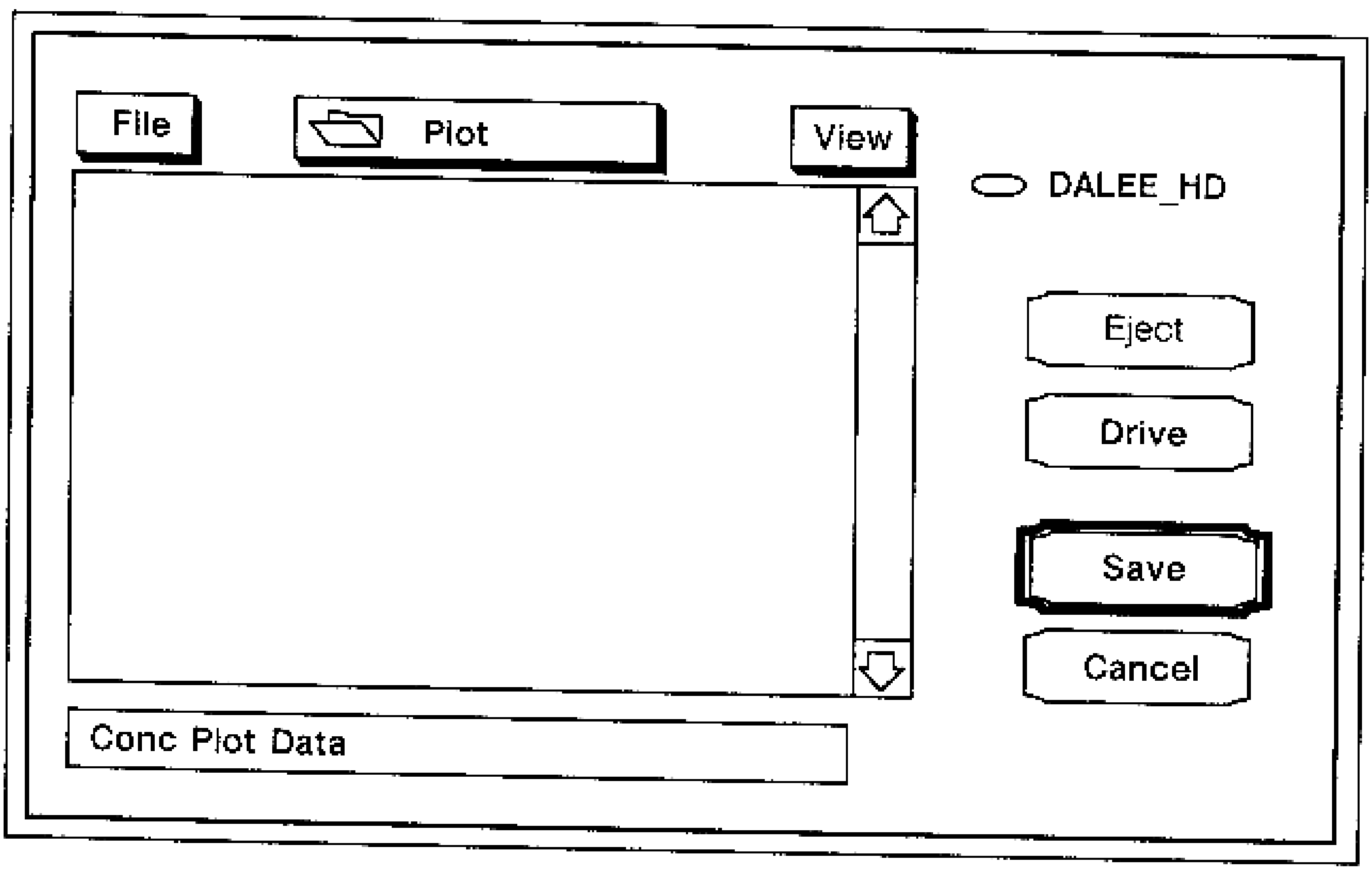


Figure 9. Window for naming plot data output files.

Program solution is initiated after designating the plot files and, upon completion, the user is asked by the program if another run is desired. The user responds to this prompt by typing "yes" or "no." A "yes" response reinitializes the program and prompts the user for new input and output files. A "no" response terminates the program execution and the user may then quit the application or return to the Macintosh Finder. From the Finder, data plotting applications may be selected to analyze the program's concentration and device efficiency data.

DETAILED COMPUTER PROGRAM DESCRIPTION

A detailed description of the TCCS computer program source files is provided to acquaint the user with the main program and each significant subroutine. Flow charts of these routines are provided, and discussion of the theoretical basis for some routines is provided where appropriate. A listing of the program source files is provided in appendix A. This description is paraphrased from a description produced by Lockheed Missiles and Space Company, Inc., under contract NAS8-36406. This work served as the primary reference for this section, and all block flow diagrams were adapted from this document.[13]

Program Editing, Compiling, and Linking

This program was edited, compiled, and linked using the Ryan-McFarland RM/FORTRAN® version 2.42 which includes the RM/FORTE® project manager. This FORTRAN compiler is recommended for making changes to the source files.

Main Program

The main program, MAIN, is a simple program with no branching and two loops. A flow diagram is shown in figure 10. Each subroutine required for the particular program run is called during each pass of the main calculation loop until the end of the simulation.

Subroutines CAFILL and RAFILL, which write zeros into all the calculation matrices are called initially to initialize each calculation matrix. Next, CRIN and PRIN are called to read the contaminant, device definition, and time-dependent input data into matrices NN, CDI, DD, and TT. The input data is printed line by line, if desired, by calling subroutines CROUT2 and RROUT2. All initial variables such as time-increment beginning time, time-increment ending time, and the increment counter are zeroed.

The precalculation setup routine, PCSET, is called next. This routine calculates the initial removal efficiency for each removal device, the equilibrium cabin concentration, and the final cabin concentration for a cabin concentration of 1×10^{-20} mg/m^3 for all contaminants. Intermediate and final calculation results are stored in matrices CC and DD.

The calculation loop is now entered, and the iterative process of determining the cabin concentration for each contaminant at the end of a time increment is begun. Since cabin concentration is a function of the contaminant mass removed and the contaminant mass removed is a function of the cabin concentration, it is important that the same value be used in the mass balance and removal

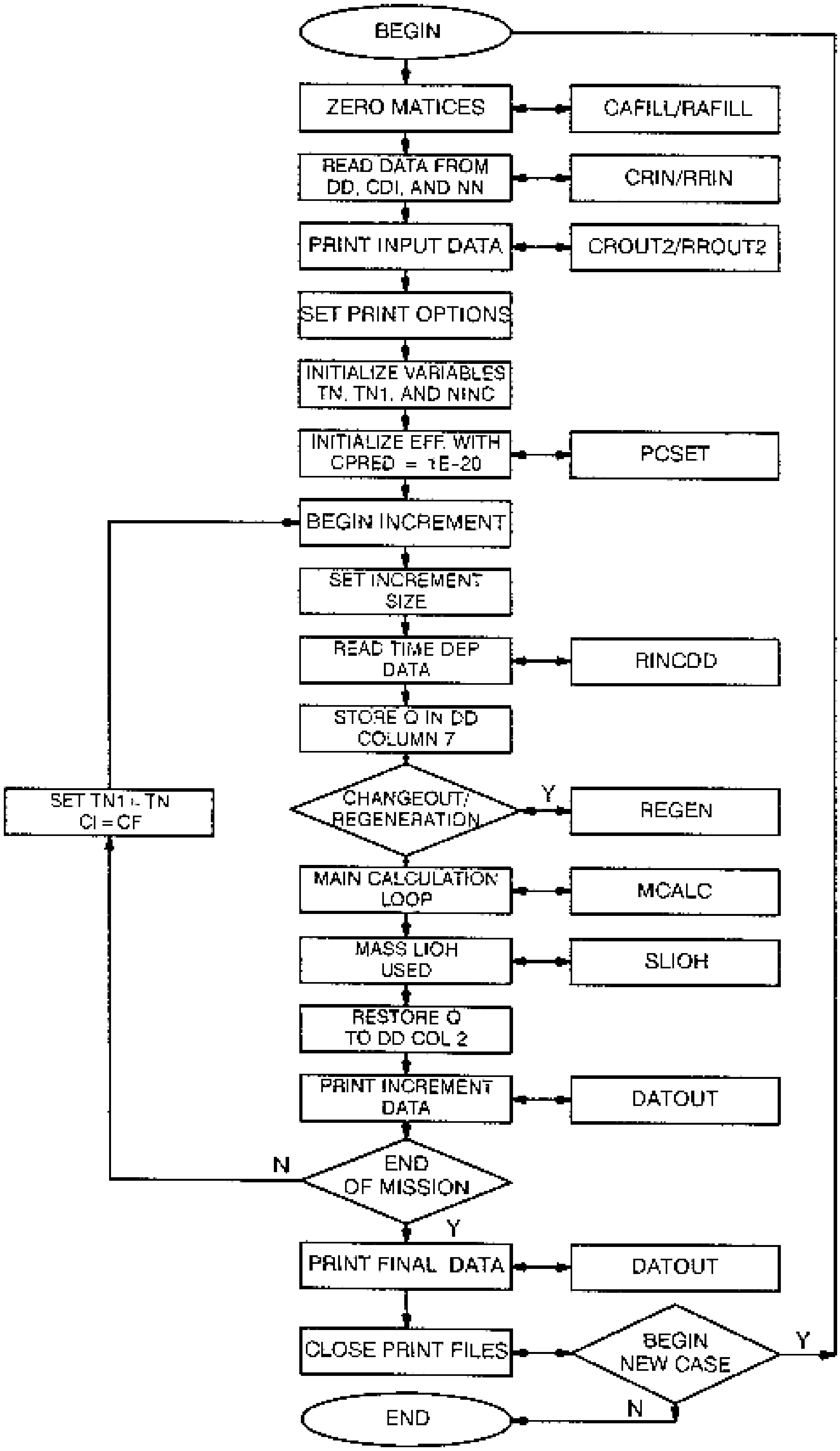


Figure 10. Main program block flow diagram.

efficiency calculation routines. A solution is achieved by assuming an increment concentration, calculating an average contaminant concentration, and comparing the two concentrations. This procedure continues until the assumed and calculated concentrations are equal.

Calculation for each time increment is initiated by increasing the increment counter, setting the increment size, and listing the increment number, starting time, and ending time to the computer screen. Subroutine RTNCDD is called to read time-dependent data from matrix TT at this time, and subroutine REGEN is called to check for regeneration of any devices during the time increment and set the adsorbed contaminant masses and device flow rates to zero if necessary. The main calculation loop subroutine, MCALC, is called next to calculate the removal efficiencies, average calculated concentration, and final concentration for each contaminant based on the sum of the mass removed during the previous time increment. Subroutine SLIOH then calculates the amount of LiOH used during the time increment if a LiOH bed is specified in the device definition file. Next, the original device flow rate is restored for any device that was being regenerated during the increment. Subroutine DATOUT is called to print the calculated data at the end of a time increment, if necessary, to both the standard formatted and plot data output devices. The simulation and mission duration times are then compared to determine whether the mission simulation has ended. If the mission simulation has not ended, another pass through the calculation loop begins by setting the new increment beginning time and initial cabin concentration equal to the previous increment ending time and concentration. If the mission simulation has ended, subroutine DATOUT is called to write the final answers to the appropriate output devices as specified by the user. The output files are closed, and the program loops to the beginning to begin another run if the user wishes. If the user has no other runs to make, the program execution is terminated, otherwise, the calculation matrices are zeroed and new input data is supplied to the program for the next run.

Brief descriptions for each major TCCS computer program subroutine are provided in the order that they are called by MAIN. Figure 11 lists the subroutines as they are called and provides a brief description of each subroutine's purpose. Block flow diagrams are provided for the most significant subroutines.

Calculation Loop Subroutines

The following subroutines comprise the principal calculation framework for the TCCS computer program.

Subroutine CAFILL

The subroutine CAFILL is called by MAIN and fills the matrix NN with blanks. Matrix NN contains the contaminant names during the simulation run.

Subroutine RAFILL

The subroutine RAFILL is called by MAIN and fills the matrices CC, TT, CDI, and DD with zeros. Matrices CDI and CC contains contaminant input and calculation data, matrix DD contains device calculation data, and matrix TT contains time-dependent data. This routine is used at the beginning of a computer simulation to initialize these matrices in the event a previous run has been made.

SUBROUTINE LEVEL					DESCRIPTION
1	2	3	4	5	
MAIN					MAIN PROGRAM
	CAFILL				ZERO MATRIX NN
	RAFILL				ZERO MATRICES CDI, CC, DD, TT
	CRIN				INPUT DATA INTO NN AND CDI
	RRIN				INPUT FROM FILE TO DD AND TT
	CROUT2				PRINT DATA FROM NN AND CDI
	RROUT2				PRINT DATA FROM DD AND TT
	PCSET				PRECALCULATION SETUP FOR ALL CONT
		PRAFIL			ZERO MATRIX DD COLUMNS 17–21
		CNRSUB			CALCULATE REMOVAL EFFICIENCY
			ACHBD		AXIAL CHARCOAL BED EFFICIENCY
			RCHBD		RADIAL CHARCOAL BED EFFICIENCY
			ALIOH		AXIAL LiOH BED EFFICIENCY
			COOXID		CO OXIDIZER EFFICIENCY
			CATBNR		CATALYTIC OXIDIZER EFFICIENCY
			CONDHX		CONDENSATE EFFICIENCY
		MASBAL			CONTAMINANT MATERIAL BALANCE
			CALCM		SUM OF MASS REMOVED BY DEVICES
			LDIGEN		LOAD GENERATION INTO DD COL 19
			PCAVCF		CALCULATE FINAL AND AVERAGE CONC
	RINCDD				READ INCREMENT DEPENDENT DATA
	REGEN				CALCULATE REGERATION/CHANGEOUT
	MCALC				MAIN CALCULATION ROUTINE
		PREDCT			CALCULATE PREDICTED AVERAGE CONC
			PRAFIL		ZERO MATRIX DD COLUMNS 17–21
			LODEFF		PUT LAST INCREMENT EFFICIENCY IN DD
			MASBAL		MASS BALANCE ROUTINE
				CALCM	SUM OF MASS REMOVED BY DEVICES
				LDIGEN	LOAD GENERATION INTO DD COL 19
				PCAVCF	CALCULATE FINAL AND AVERAGE CONC
		CONVRG			SOLVE FOR NEW REMOVAL EFFICIENCY
			PRAFIL		ZERO MATRIX DD COLUMNS 17–21
			CNRSUB		CALCULATE REMOVAL EFFICIENCIES
				ACHBD	AXIAL CHARCOAL BED EFFICIENCY
				RCHBD	RADIAL CHARCOAL BED EFFICIENCY
				ALIOH	AXIAL LIOH BED EFFICIENCY
				COOXID	CO OXIDIZER EFFICIENCY
				CATBNR	CATALYTIC OXIDIZER EFFICIENCY
				CONDHX	CONDENSATE EFFICIENCY
			MASBAL		CONTAMINANT MATERIAL BALANCE
				CALCM	SUM OF MASS REMOVED BY DEVICES
				LDIGEN	LOAD GENERATION INTO DD COL 19
				PCAVCF	CALCULATE FINAL AND AVERAGE CONC
	SLIOH				CALCULATE LiOH USED IN INCREMENT
	DATOUT				PRINT DATA TO THE SPECIFIED DEVICE(S)
		PRFANS			PRINT CONCENTRATION DATA ANSWERS
			HEADGS		PRINT DATA HEADINGS
		GROUP			PRINT TOXIC HAZARD INDEX ANSWERS

Figure 11. TCCS computer program subroutine listing and description.

Subroutine PCSET

The subroutine PCSET is the precalculation setup routine. PCSET gets calculations started by assuming an initial cabin concentration before the program enters the time calculation loop. Figure 12 shows a flow diagram of PCSET. PCSET sets the initial time increment ending time to 1/240 of the basic time increment specified in the device definition input file. Subroutine PRAFIL is then called, and columns 17 to 21 are zeroed. These columns are used to store the results of subsequent calculations. CNRSUB is called to calculate each device removal efficiency for an assumed initial contaminant concentration of 1×10^{-20} mg/m^3. Contaminant removal rates and predicted, equilibrium, and final cabin concentrations are calculated by subroutine MASBAL. These calculated values are copied from matrix DD to the calculation matrix, CC, and printed out by subroutines CROUT and RROUT if required.

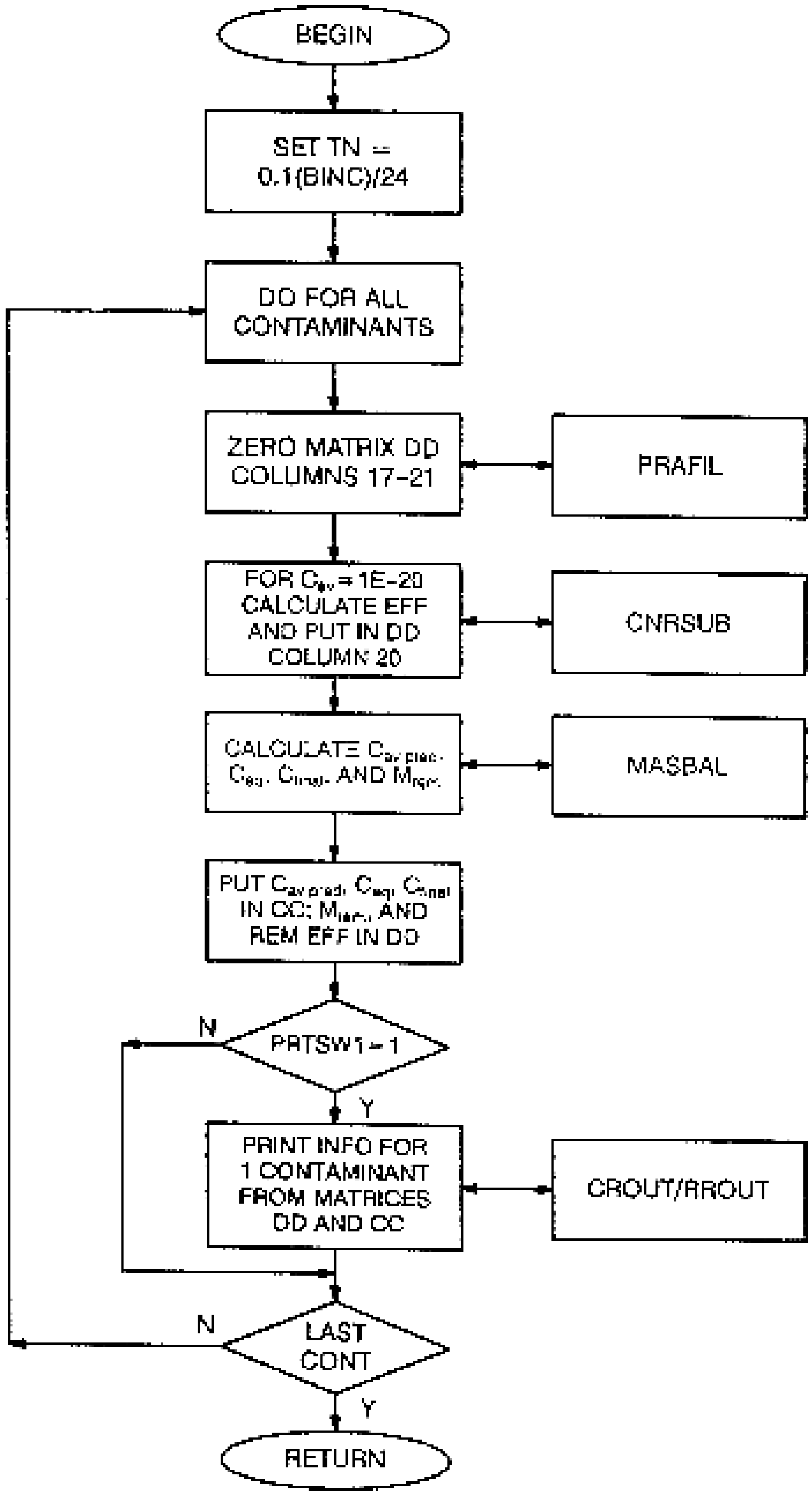


Figure 12. Subroutine PCSET block flow diagram.

Subroutine PRAFIL

Subroutine PRAFIL is called by PCSET and places zeros in matrix DD columns 17 through 21.

Subroutine CNRSUB

The subroutine, CNRSUB, calculates the removal efficiency of each device for each contaminant in the simulation during every time increment. This calculation is based on the average calculated cabin concentration. Figure 13 shows a block flow diagram for CNRSUB. These calculations are conducted by device type rather than the relative positions of each device with respect to each other.

This routine sets the cabin removal efficiency to zero and the leakage efficiency to the maximum of 1.0. Removal efficiencies for all devices with no flow are also set to zero. The remaining device efficiencies are calculated by calling the subroutines ACHBD, RCHBD, ALIOH, COOXID, CATBNR, and CONDHX. These calculated efficiencies are stored in matrix DD.

Subroutine MASBAL

Using the device efficiencies calculated by CNRSUB, MASBAL determines the mass removed, the calculated cabin concentration, the equilibrium cabin concentration, and final cabin concentration for each contaminant during a time increment. This calculation is conducted for all removal devices in parallel and in series. Figure 14 shows a block flow diagram for MASBAL.

MASBAL uses the mass of contaminant removed and the net mass to the cabin to determine the final cabin concentration for each contaminant. The mass of the contaminant removed is defined as the product of the removal device flow rate, contaminant concentration, and device removal efficiency. The net mass of contaminant to the cabin is defined as the difference between the mass generated and mass removed. At steady state or equilibrium, the mass removed equals the mass generated. The mass generated is the sum of all generation sources which include the cabin generation rate and the generation rate in each device. The steady-state of equilibrium concentration is defined according to the following equation:

$$C_{SS} = (m_{\text{net-to-cabin}})/(\eta_r \times Q) \ , \tag{2}$$

where $m_{\text{net-to-cabin}}$ is the mass of contaminant, η_r is the overall removal efficiency for all devices, and Q is the atmospheric flow rate through the removal devices.

MASBAL is composed of two parts to determine the steady-state concentration. The first part of MASBAL determines the product of the overall efficiency and flow rate by setting the device generation rates to zero, assuming an arbitrary value for average cabin concentration (100 mg/m^3) and cabin generation rate (50 mg/h), and calling CALCM to determine the sum of mass removed for all the removal devices. The second part of MASBAL evaluates the net mass to the cabin by setting the average cabin concentration equal to zero, restoring the contaminant and device generation rates to the values specified in the contaminant data matrix, and calculating the mass removed using CALCM. The $m_{\text{net-to-cabin}}$ equals the difference between the masses generated in the cabin and removal devices and the mass removed. From these values, C_{ss} is calculated according to equation (2). After calculating the steady state concentration, the final and average cabin concentrations are calculated by calling PCAVCF and CALCM is called to calculate the mass removed by the cabin and each device using the average calculated cabin concentration.

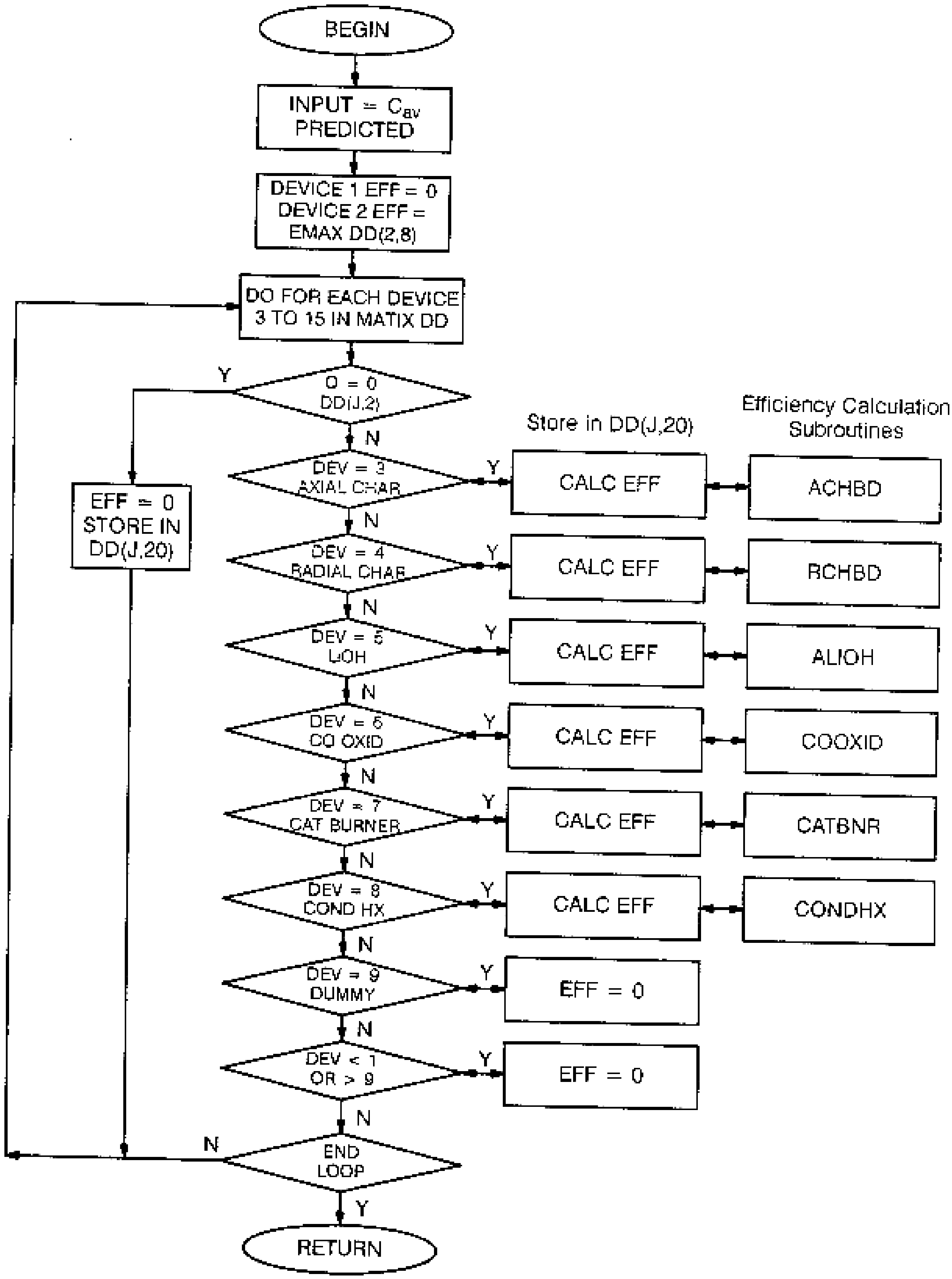


Figure 13. Subroutine CNRSUB block flow diagram.

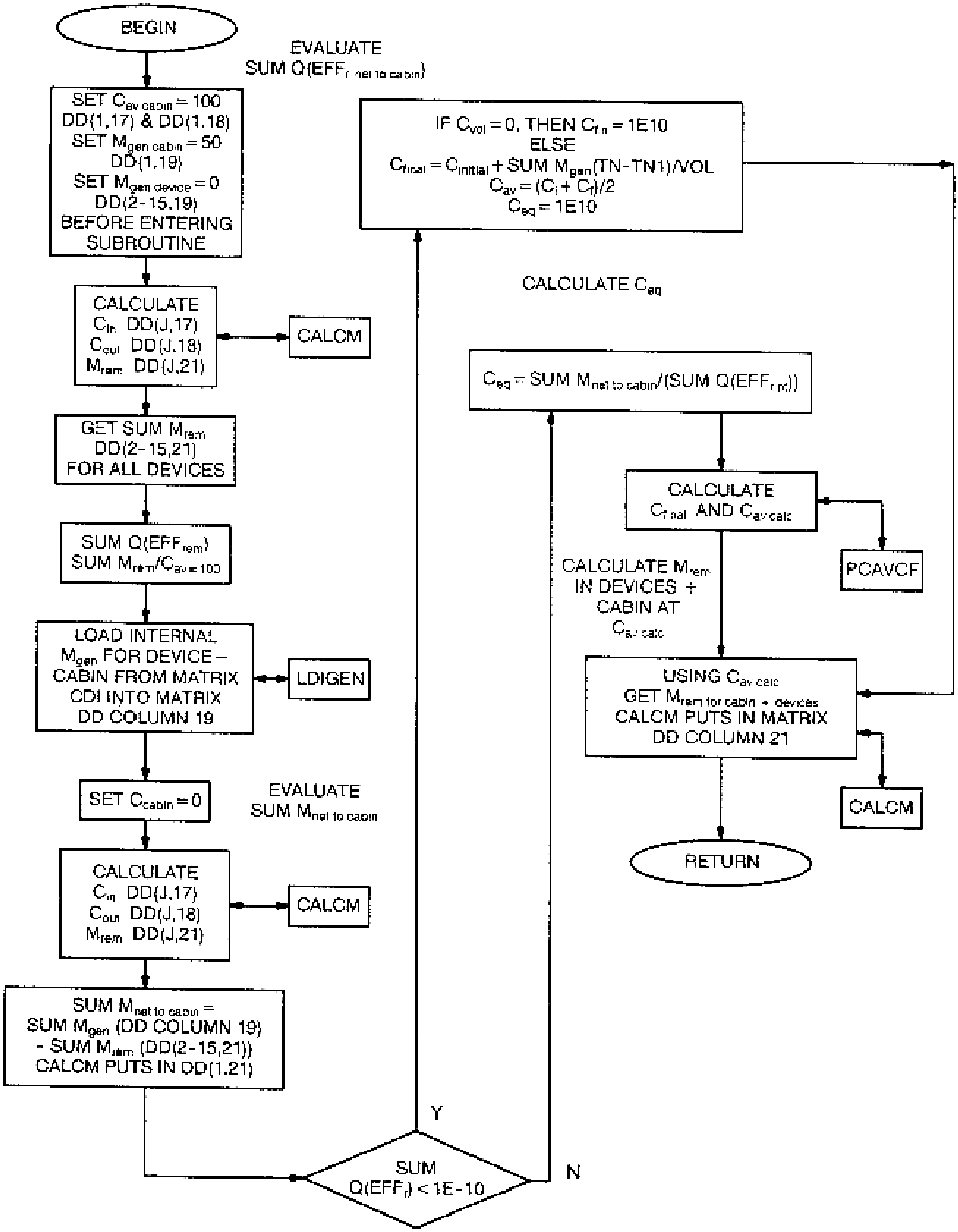


Figure 14. Subroutine MASBAL block flow diagram.

Subroutine CALCM

The removal device inlet and outlet concentrations and the total mass removed by the cabin and the specified removal devices is calculated by CALCM by using the removal efficiencies, generation rates, and average cabin concentration. This calculation is sequential from one device to another and uses the outlet concentration of an upstream device as the inlet concentration for a downstream device. This calculation requires the device definition input data to be arranged to allow calculations for all upstream devices to be completed before calculations for the downstream devices. Figure 15 shows a block flow diagram for CALCM.

The subroutine sets the cabin and leakage device inlet and outlet concentrations equal to the average cabin concentration. All other devices are tested for zero flow. Devices with zero flow have their inlet concentration, outlet concentration, and mass removed set equal to zero. Upstream devices for each removal device are identified. If the upstream device type is 1 or 2, the inlet concentration is set equal to the average cabin concentration plus any internal device generation rate divided by the device flow rate. Upstream device types other than 1 or 2 cause the device inlet concentration to be based on the flow rates and outlet concentrations of all the upstream devices.

Outlet concentration and the mass removed by the devices are calculated according to the following equations:

$$C_{\text{out}} = C_{\text{in}}(1-\eta_r) \tag{3}$$

$$m_{\text{rem}} = C_{\text{in}}(Q)(\eta_r) \ . \tag{4}$$

The inlet concentration for a device with an upstream device is set equal to the outlet concentration for the upstream device. A device with multiple upstream devices requires the mixing of streams with varying concentrations to be considered. For example, the inlet concentration for a device with three upstream devices must be calculated according to the following equation:

$$C_4 = (C_1 Q_1 + C_2 Q_2 + C_3 Q_3)/Q_4 \ . \tag{5}$$

The sum of the mass removed and mass generated is calculated by adding the masses removed and masses generated by all the devices. The difference between the sum of the mass generated and the sum of the mass removed gives the mass removed by the cabin.

Subroutine LDIGEN

Subroutine LDIGEN is called by MASBAL to load the generation rates from matrix CDI column 1 and columns 10 through 22 into matrix DD column 19.

Subroutine PCAVCF

Subroutine PCAVCF is called by MASBAL to calculate the increment final and average cabin concentrations for each contaminant.

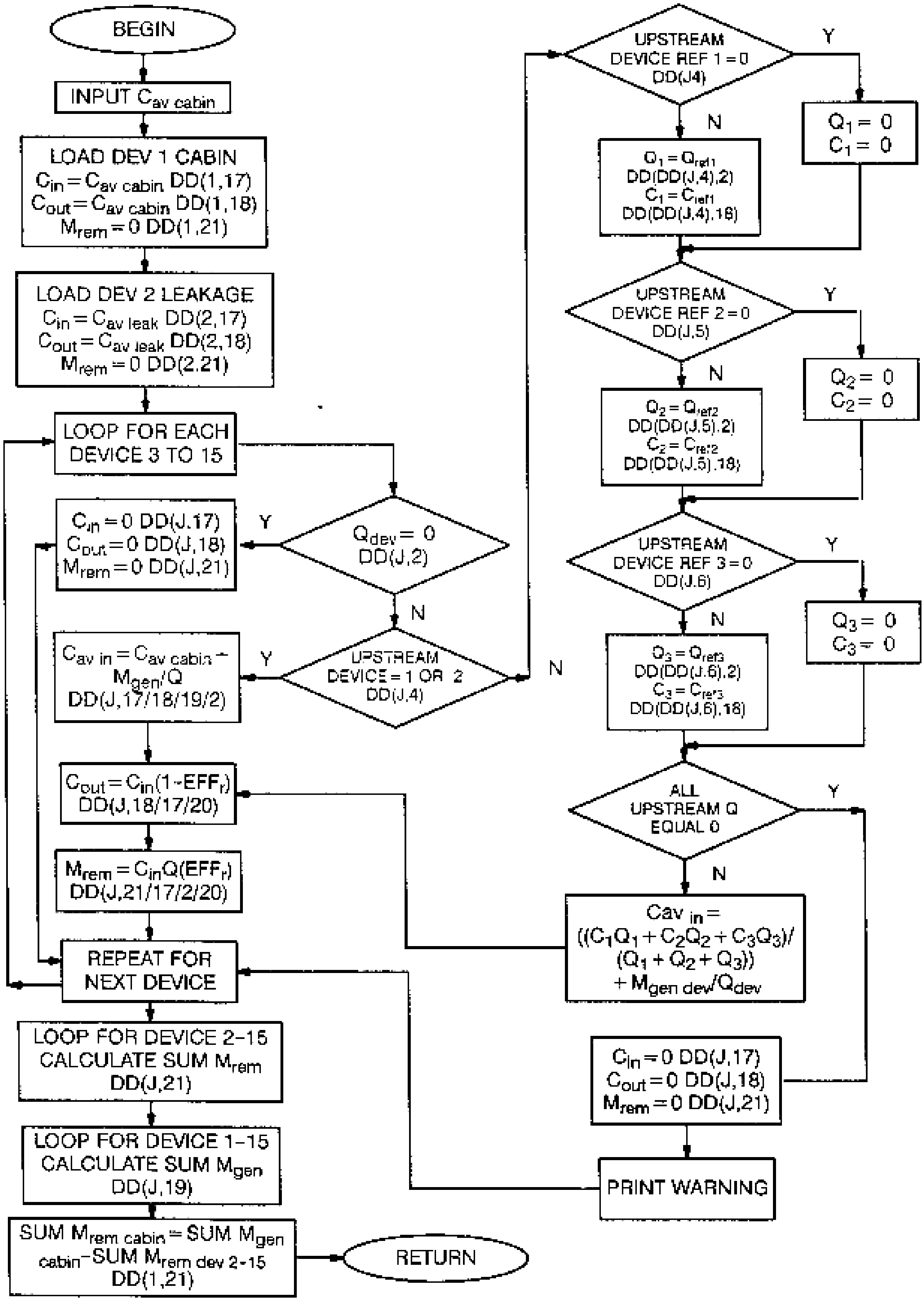


Figure 15. Subroutine CALCM block flow diagram.

Subroutine RINCDD

Subroutine RINCDD is used at the beginning of each time increment to input and operate on the time-dependent data. A flow diagram of RINCDD is shown by figure 16. This subroutine checks the time-dependent data to determine whether any changes occur during the current time increment. Variables in matrix TT are identified. If a contaminant generation rate is indicated, the new rate is placed in the calculation matrix, CDI. Likewise, if a change in removal device flow rate or any other device change is indicated, the new information is placed in the appropriate device definition matrix, DD, location.

Subroutine REGEN

Figure 17 shows a block flow diagram for subroutine REGEN. This subroutine determines whether any charcoal or LiOH beds will be regenerated during the current time increment. If regeneration occurs, the mass of contaminants stored in the beds is set equal to zero. Similarly, if the regeneration duration lasts for the entire time increment, the device flow rate is set equal to zero.

The first check conducted by the subroutine is for device type. Only charcoal and LiOH beds may be regenerated. Only regeneration cases which begin at the time increment beginning or regeneration cases which last for one or more complete time increments are treated. Data concerning the regeneration interval, duration, and first regeneration time are obtained from matrix DD. For the LiOH bed regeneration, the duration is set equal to zero since bed changeout is assumed to occur quickly. The initial time, regeneration time, and regeneration duration are then checked to determine whether they are exact multiples of the basic time increment. If they are not, they are rounded to the next lowest multiple of the time increment and a warning is written to the screen.

The next checks conducted by the routine determine whether regeneration occurs at the beginning of a time increment and whether the regeneration lasts for the entire increment. Regeneration for the entire increment causes the program to deactivate this device for that increment by setting the device flow rate equal to zero. Regeneration at the beginning of an increment causes the sum of the mass removed by that device to be set equal to zero. For an LiOH bed, the total mass of LiOH used is also set equal to zero.

Subroutine MCALC

Calculation of the removal efficiency, mass removed, and calculated, equilibrium, and final cabin concentration for each contaminant and each removal device is controlled by MCALC. These calculations are based on the cumulative mass removed for each contaminant during the previous time increment. Figure 18 shows a block flow diagram for MCALC.

MCALC calls the subroutine PREDCT to calculate the average predicted cabin concentration based on the removal efficiency and the cumulative mass of contaminant removed during the previous time increment and the generation rate during the present time increment. The average predicted concentration is used by the subroutine CONVRG to calculate a new removal efficiency; mass removed; and average calculated, equilibrium, and final cabin concentrations. The predicted and calculated concentrations are compared in CONVRG and recalculated until the difference between them is less than the convergence error specified in the device definition data file, matrix DD. This recalculation and comparison continues for 20 iterations with the full time increment or until the difference is less than the convergence error. If the convergence error is still exceeded after 20 iterations,

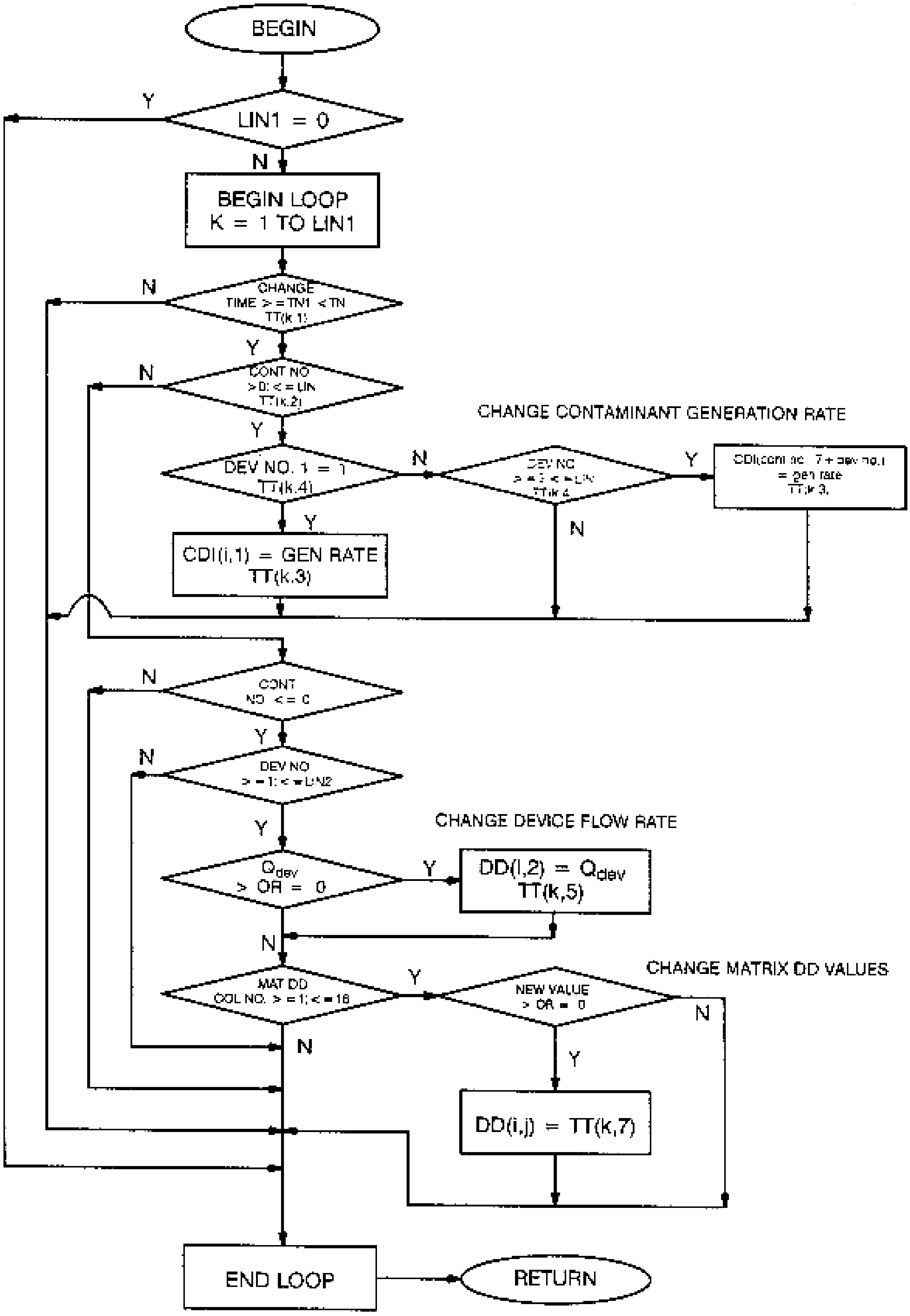


Figure 16. Subroutine RINCDD block flow diagram.

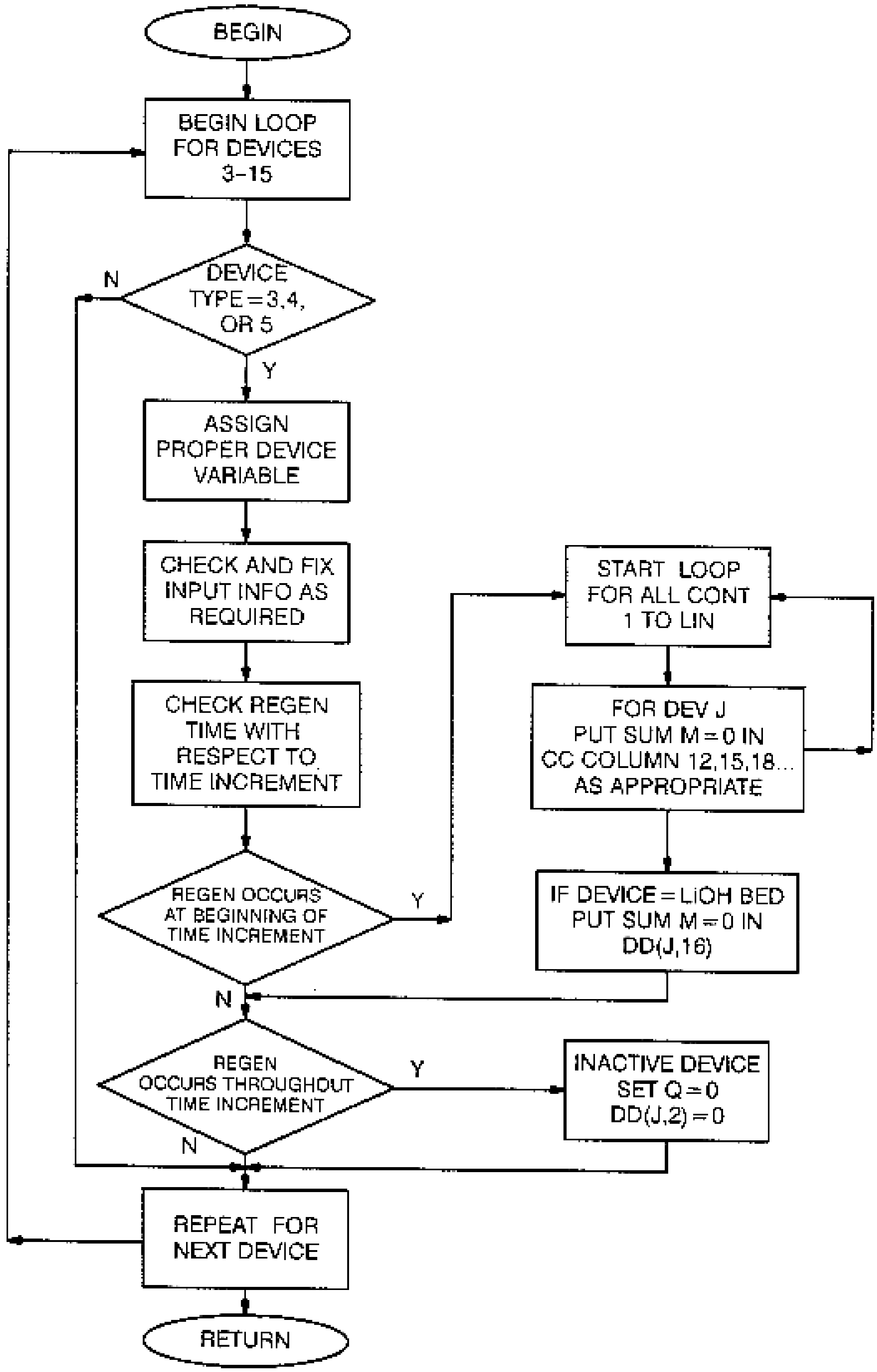


Figure 17. Subroutine REGEN block flow diagram.

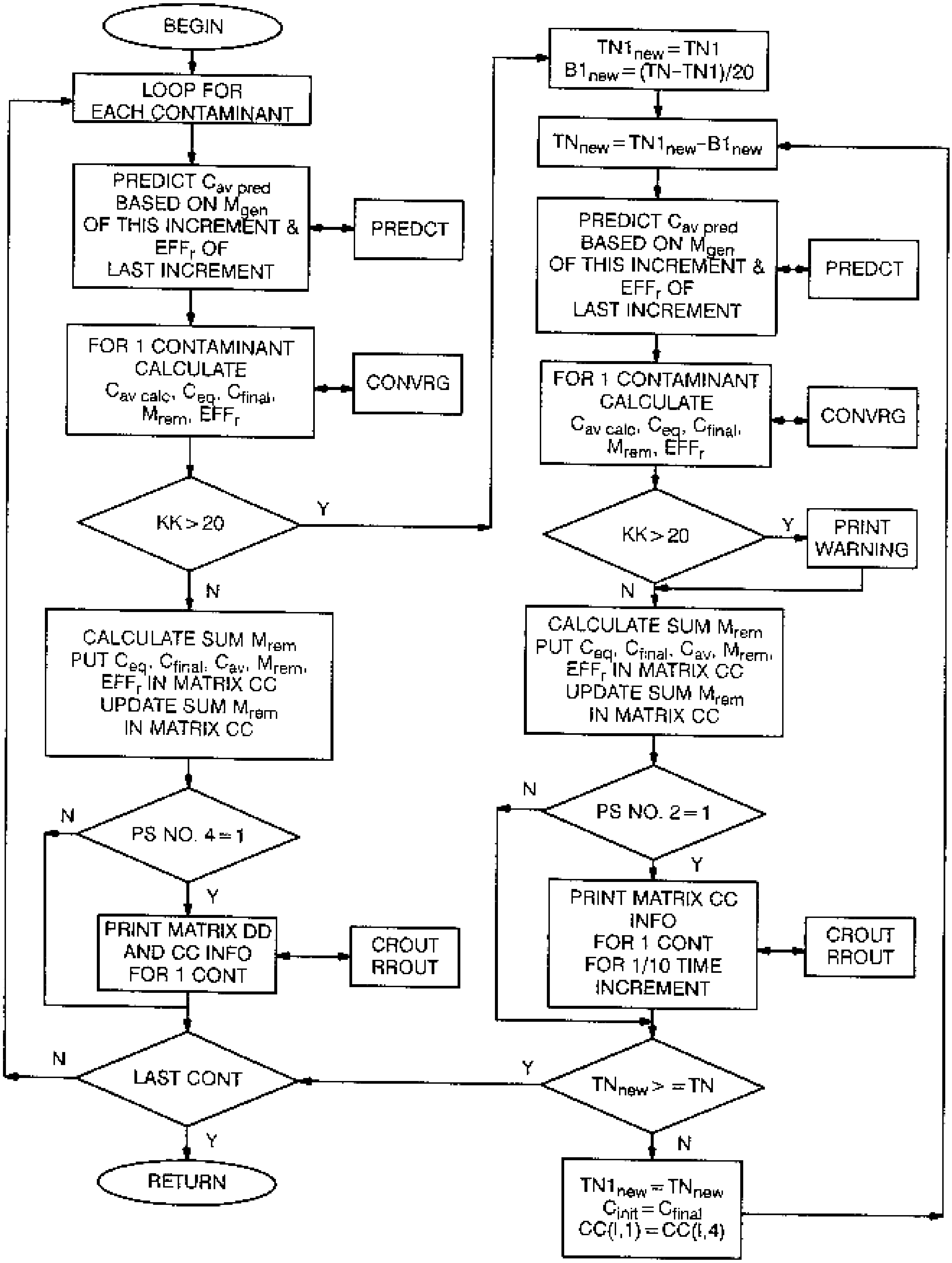


Figure 18. Subroutine MCALC block flow diagram.

another loop with a maximum of 20 iterations is entered which uses one-twentieth the basic time increment for the calculation. This loop ends as soon as the difference between the predicted and calculated values is less than the convergence error or 20 iterations have been completed. If convergence is not attained after this loop, the program writes a warning to the screen indicating that the calculation for the contaminant did not converge. The loop using the one-twentieth time increment is used only for a contaminant that does not converge during the first 20 iterations. This is more efficient than reducing the time increment for all the contaminant calculations.

Subroutine PREDCT

MCALC calls the subroutine PREDCT to calculate the average predicted cabin concentration for each contaminant during each time increment. This calculation is based on the removal efficiency and sum of contaminant mass removed in the previous increment and the generation rate during the present increment. Figure 19 shows a block flow diagram for PREDCT.

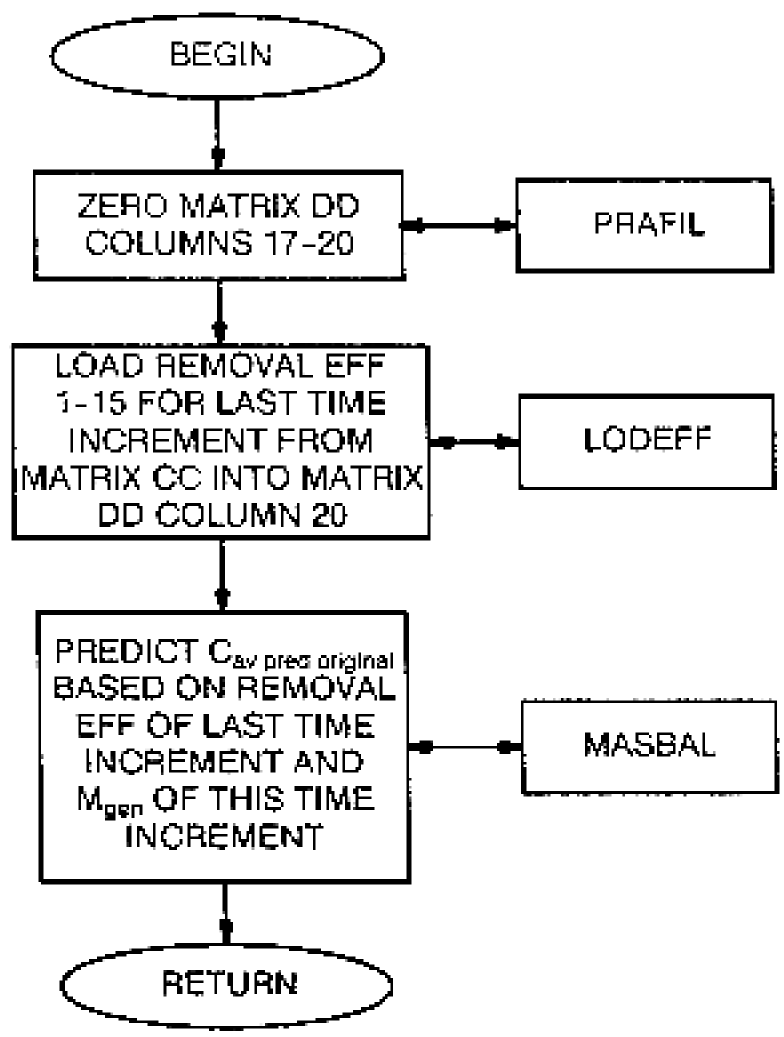


Figure 19. Subroutine PREDCT block flow diagram.

PREDCT calls the subroutine PRAFIL to zero the part of matrix DD required for storing the calculation results. Data from the previous time increment is obtained by LODEFF and MASBAL is called to calculate the efficiency and concentration.

Subroutine LODEFF

Subroutine LODEFF loads the efficiency calculated in the preceding increment from matrix CC to matrix DD.

Subroutine CONVRG

CONVRG is the main convergence loop subroutine. This subroutine calculates the average cabin concentration and compares it with the predicted cabin concentration for each contaminant during every time increment. Figure 20 shows a block flow diagram for CONVRG.

CNRSUB is called by CONVRG to calculate the removal efficiency for each device using the predicted cabin concentration. Based on this removal efficiency, MASBAL calculates the average, final, and equilibrium cabin concentrations and the mass of contaminant removed by each device. The predicted and calculated cabin concentrations are compared by calculating the absolute value of the difference of the predicted and calculated concentration divided by the predicted concentration and comparing that value to the convergence error. If the absolute value of the comparison is less than the convergence error, convergence has been achieved and the iteration stops for that contaminant. If convergence has not been achieve, a new cabin concentration is calculated using a bisection technique after the first iteration and a Newton-Raphson technique for each additional increment. The loop counter value passed back to MCALC which determines whether convergence has been reached within 20 iterations.

Subroutine SLIOH

SLIOH is the subroutine which calculates the cumulative mass of LiOH used during the simulation run. This calculation is cumulative since the mass of LiOH consumed during the present increment is added to the mass consumed in all the previous increments.

Contaminant Removal Device Calculation Subroutines

The contaminant removal device subroutines are supported by a substantial amount of theoretical and experimental data. A brief description of each subroutine is provided in addition to a discussion of the supporting theory and experimental data.

Subroutine ACHBD

Subroutine ACHBD calculates the removal device efficiency for an axial flow charcoal bed. This routine simulates the physical adsorption of contaminants onto the surface of the charcoal. Specially treated charcoals are also considered which include chemical reaction between the surface treatment and the contaminant in addition to adsorption.

A charcoal bed is composed of two zones during the adsorption process. These zones are designated as the saturated zone and the adsorption zone. All contaminant removal takes place in the adsorption zone. The saturated zone provides no net removal since it is in equilibrium with the vapor phase contaminant composition. Figure 21 illustrates the zones simulated by the program graphically. Physical adsorption is an equilibrium process which depends on variables such as the contaminant vapor pressure, inlet concentration, molar volume, and the cabin temperature. Studies conducted by Robell investigated the thermodynamics of adsorption dynamics and developed a correlation between the physical properties of a contaminant and the charcoal saturation capacity. This correlation is based on the Polanyi potential theory and the Gibbs equation. From this study, a correlation factor, called the adsorption potential factor was developed. This factor is defined according to the following equation:

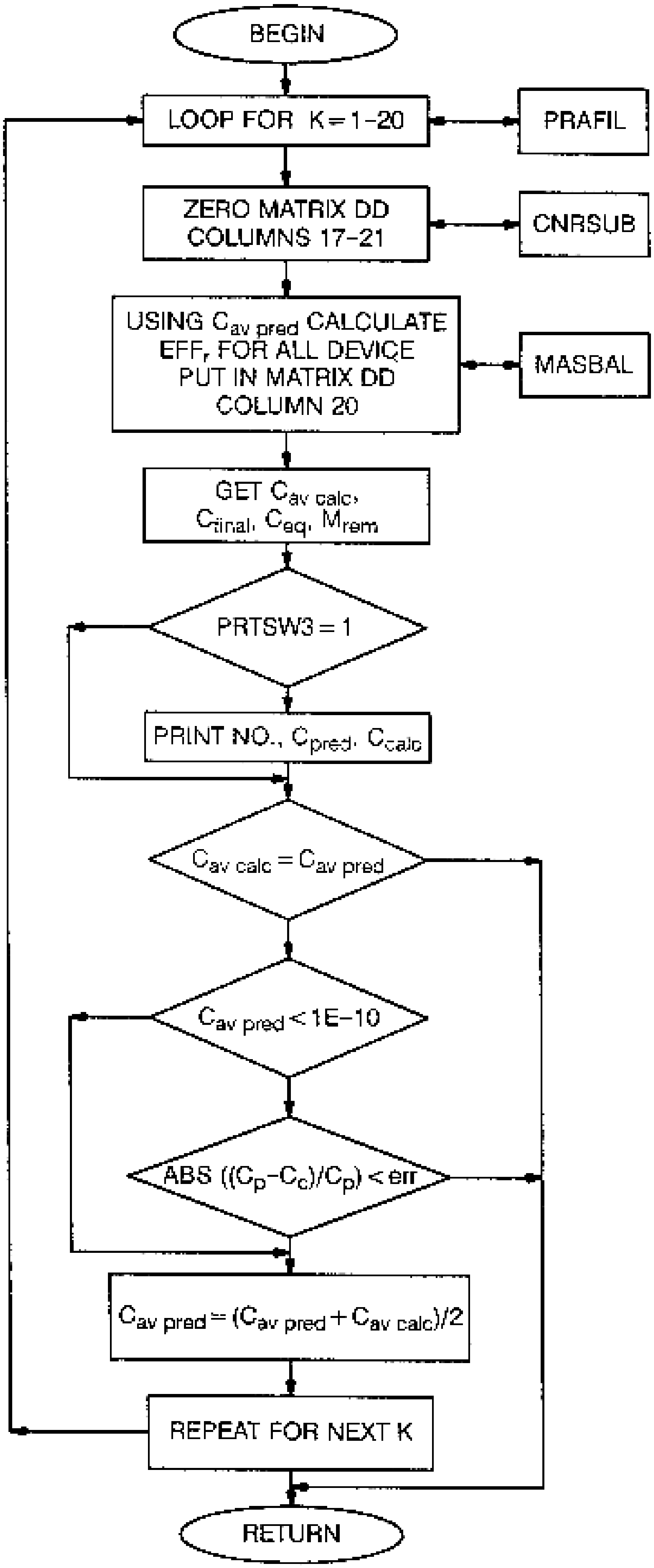
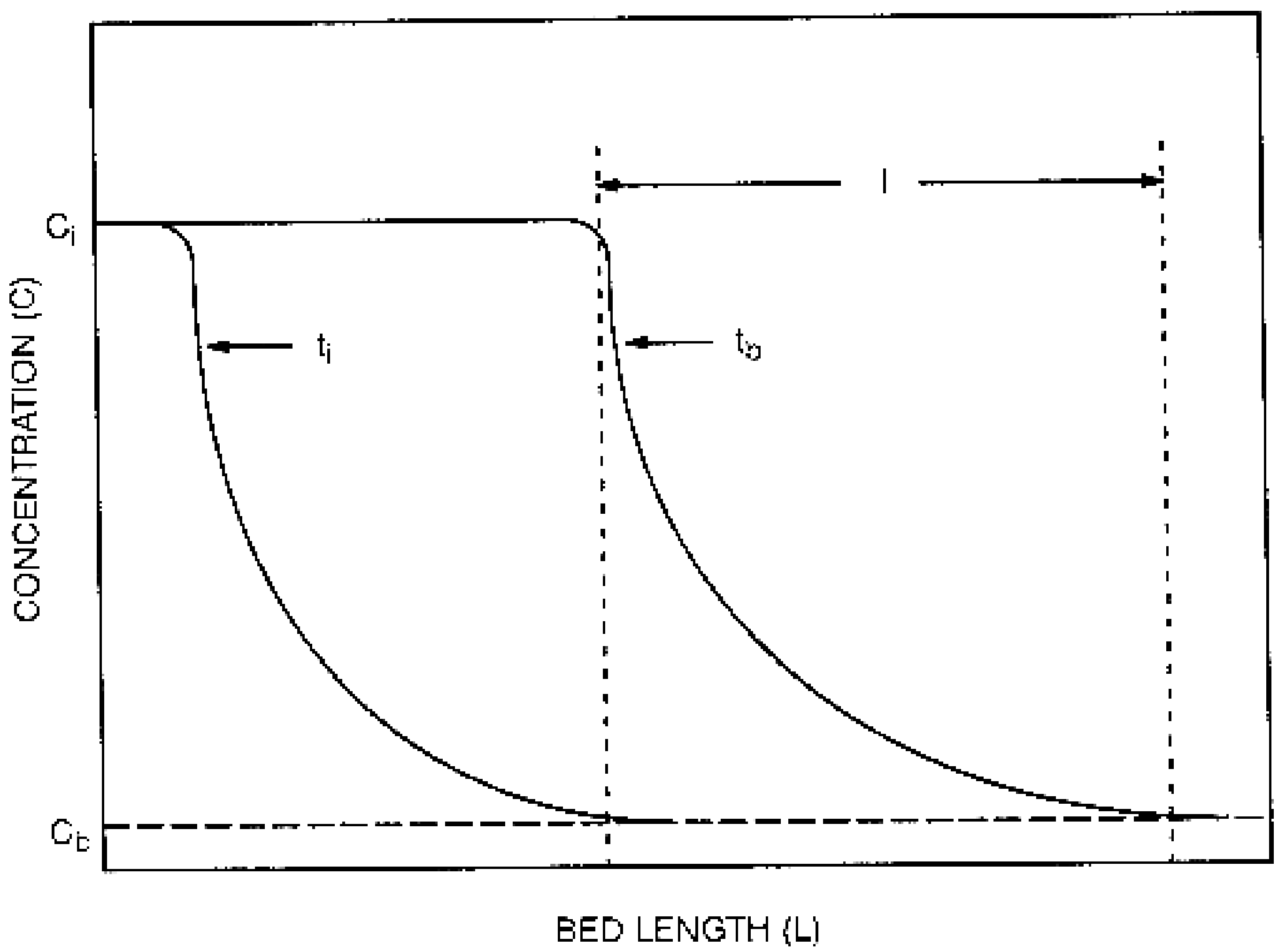


Figure 20. Subroutine CONVRG block flow diagram.



LEGEND

t_i = Bed profile at the time when a bed segment reaches steady state
t_b = Bed profile at the service time when the bed outlet concentratio equals C_b
l = Active adsorption zone length
C_i = Bed inlet concentration
C_b = Bed penetration concentration

Figure 21. Charcoal saturation and adsorption zone distribution.

$$A = (T/V_m)\log_{10}(p_v/p_c) \ , \qquad (6)$$

where T is the cabin temperature in kelvin, V_m is the contaminant liquid molar volume in cm^3/gram-mole, p_v is the contaminant vapor pressure at the cabin temperature expressed in concentration units of mg/m^3, and p_c is the cabin contaminant partial pressure expressed in concentration units of mg/m^3. This factor was plotted as a function of experimentally determined charcoal saturation capacities to obtain the plot shown by figure 22.[14] The plot in this figure was constructed for Barnebey-Sutcliffe type BD granular activated charcoal. This correlation is not only sensitive to charcoal impregnation and contaminant solubility, but also to relative humidity as shown by figure 23.[15] Additional information may be obtained on charcoal capacity and performance from references 16 and 17. Based on potential plots, empirical equations are obtained which relate the potential factor to the charcoal saturation capacity. The equations used in this program are the following:

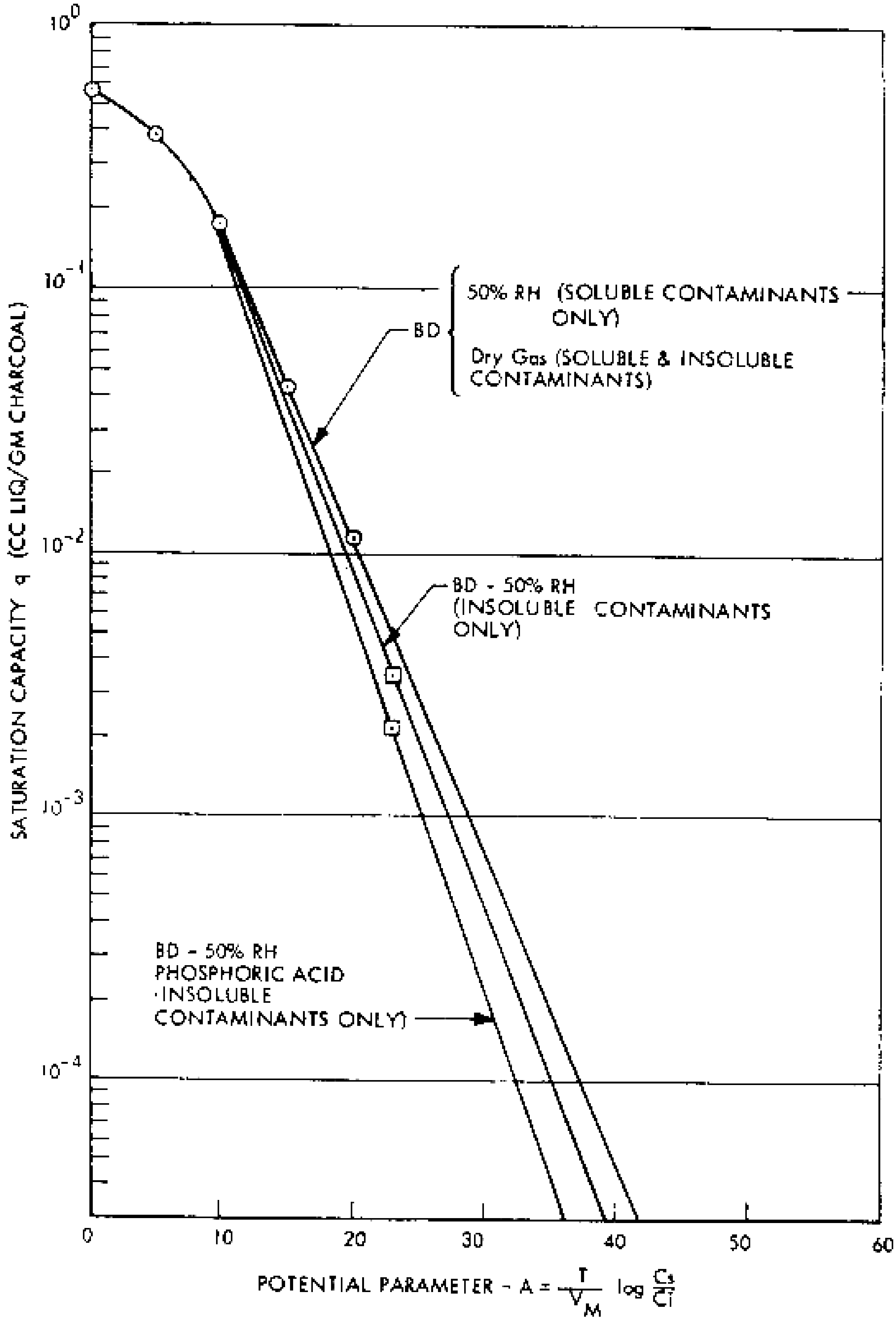


Figure 22. Potential plot for type BD granular activated charcoal.

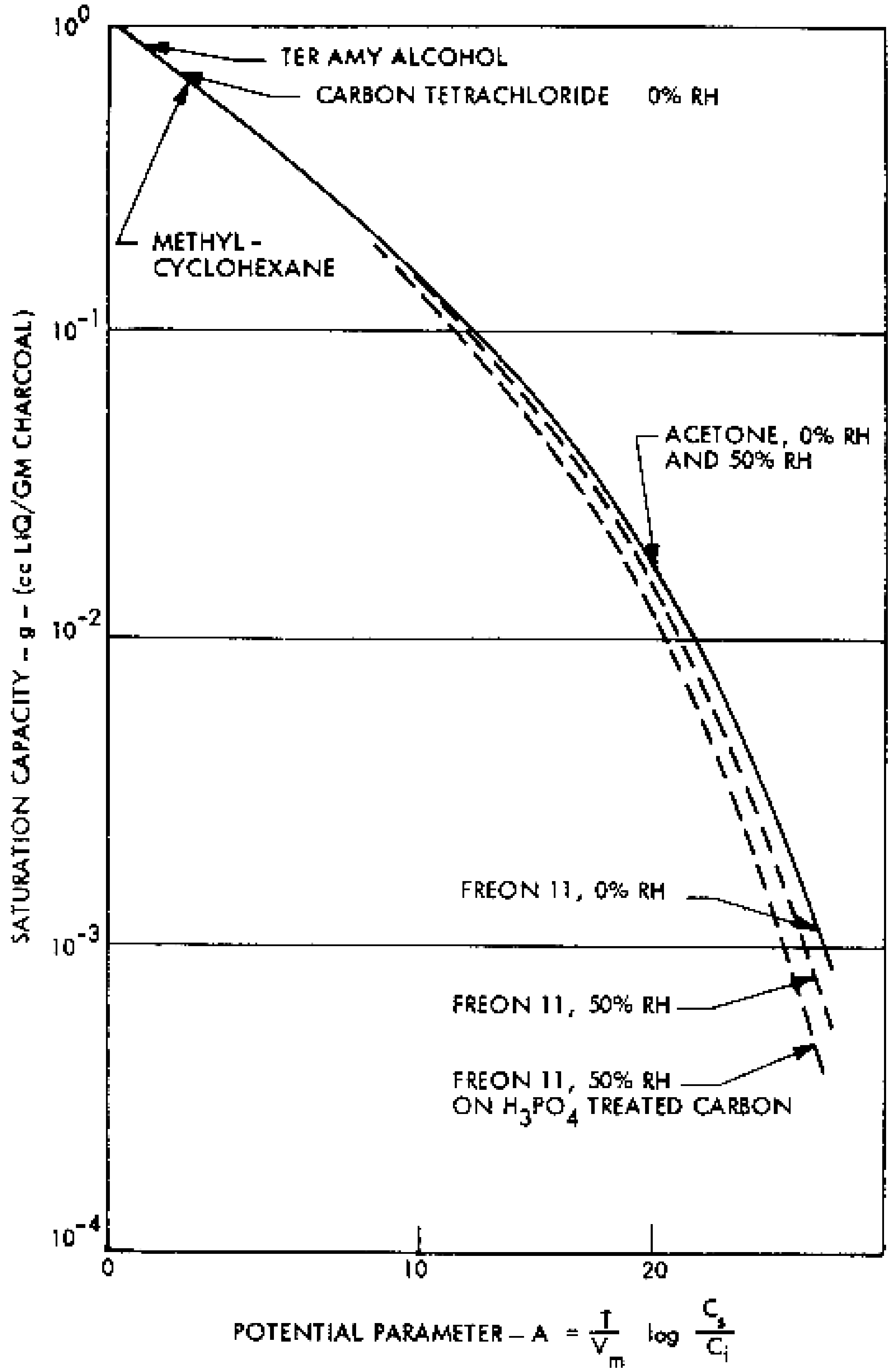


Figure 23. Potential plot showing relative humidity effects.

$$q = 2.2947e^{-0.26492A} \tag{7}$$

$$q = 2.80289e^{-0.28842A} \tag{8}$$

$$q = 3.8636e^{-0.32307A} \tag{9}$$

where q is the charcoal saturation capacity in cm^3 of liquid contaminant per gram of charcoal and A is the adsorption potential factor in kelvin-gmole/cm^3. As new information concerning adsorption capacity is obtained, these equations can be modified accordingly. In addition, this technique can be applied to other adsorbent materials to simulate other packing materials besides charcoal.

The adsorption zone length for 90-percent removal is determined from experimental data obtained by Olcott at a 0.0066 m/s (1.3 ft/min) flow rate.[18] These data are plotted in figure 24, and the computer program uses the following equation to calculate the adsorption zone length:

$$L_{ads} = (L_{ads\ \mathrm{at\ 1.3\ ft/min}})(V/1.3)^{0.8} \tag{10}$$

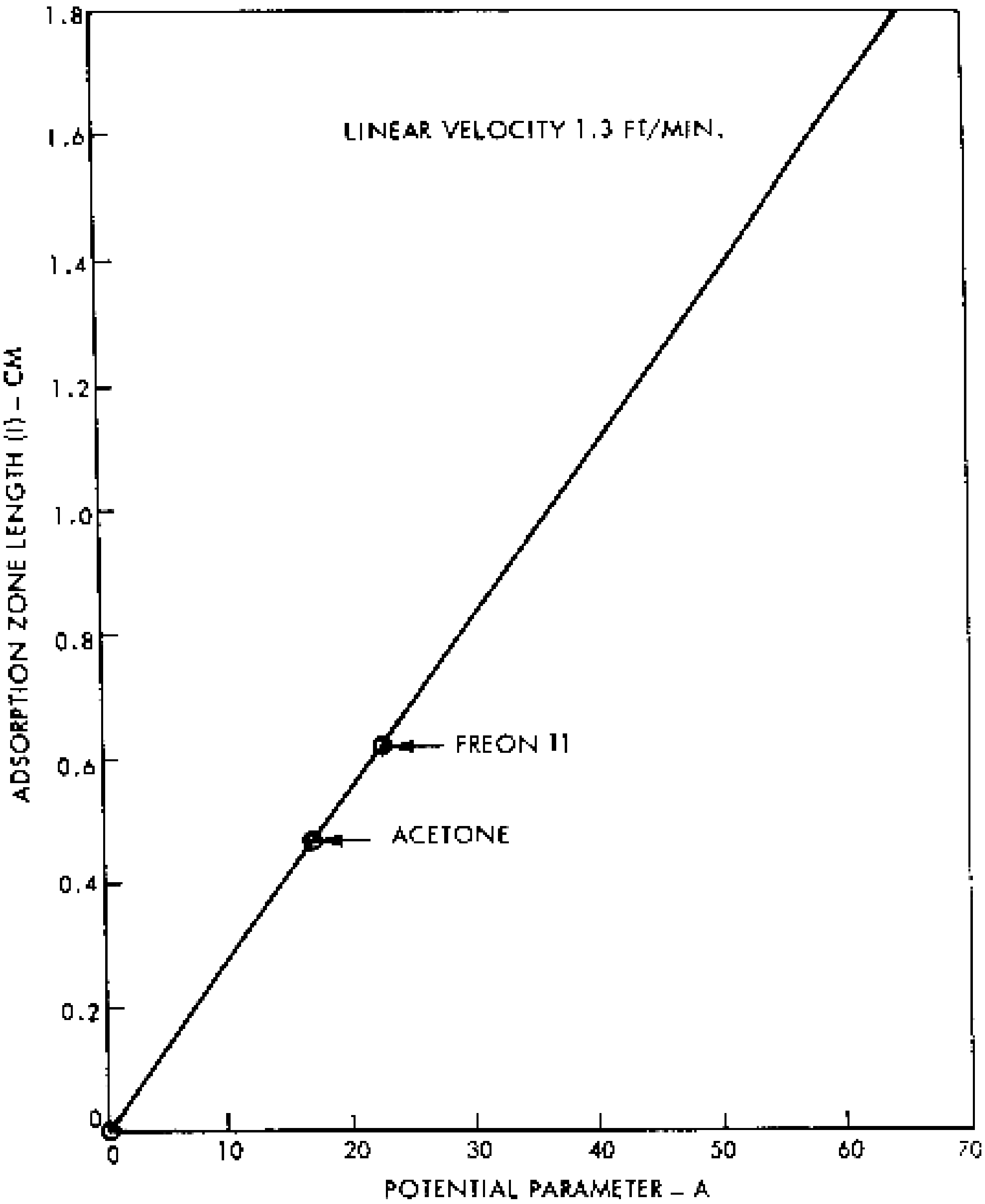


Figure 24. Adsorption zone length as a function of the potential factor.

The adsorption zone length study conducted by Olcott shows the adsorption zone length increases with velocity to the 0.8 power as indicated in the equation. The saturated zone length is based on the charcoal capacity at the prevailing cabin conditions and the amount of contaminant already adsorbed. For a given contaminant mass retained in the bed, the saturation zone length equals the mass of contaminant adsorbed divided by the saturation capacity, q. The total bed length minus the saturated zone length equals the adsorption zone length. The adsorption zone length is the length of the bed actually available for contaminant removal. The program calculates the saturation zone length which is calculated based on the bed geometry, the amount of contaminant removed by the bed, and the saturation capacity, q.

Adsorption of multiple contaminants by charcoal involves some interaction between the contaminants. This interaction, called blockage or coexistence, reduces the capacity of the charcoal to hold other contaminants. Although the theory for coexistence is complex, experimental data indicate that an additional 20 percent can be added to the saturation zone when the calculation for the adsorption zone is conducted.

Reaction with Specially Treated Charcoals. Some contaminants are not readily removed by granular activated charcoal but can be removed by charcoal which has been specially treated with chemicals that react with the contaminant after adsorption onto the charcoal surface. Two commonly used treated charcoals target ammonia and formaldehyde.

Ammonia Removal. Ammonia is removed by treating granular activated charcoal with phosphoric acid. Usually, phosphoric acid loading is two millimoles per gram of charcoal. This results in a requirement of 9.61 grams of charcoal per gram of ammonia to be removed if the reaction goes to completion. The bed removal efficiency is typically 90 to 99 percent per pass for a fresh bed. As the phosphoric acid is depleted, the efficiency drops, eventually reaching zero. This routine assumes the removal efficiency is 100 percent if the bed is less than 80 percent utilized. The efficiency for the last 20 percent of the bed is calculated using the following sine relationship:

$$\eta_r = \sin(m_{\text{charcoal}} - m_{\text{treated charcoal used}})/(0.2)m_{\text{charcoal}} \tag{11}$$

where m_{charcoal} is the mass of charcoal in the bed and $m_{\text{treated charcoal used}}$ is the mass of treated charcoal used.

Formaldehyde Removal. Formaldehyde is removed most efficiently by chromate impregnated charcoal. Manufacturer's data indicate that this charcoal can chemisorb a total amount of formaldehyde equivalent to 5 percent of its weight.[19] Testing at Lockheed Missiles and Space Company, Inc., showed this material's efficiency to drop linearly from 100 percent to 90 percent for an amount of formaldehyde chemisorbed from zero to 0.12 percent of the bed weight. Also, if the bed residence time is less than 0.25 s, the removal efficiency drops linearly.

Subroutine RCHBD

The subroutine, RCHBD, uses the same logic as ACHBD for simulating charcoal adsorption. However, this routine accommodates the geometry of a radial flow charcoal bed.

Subroutine ALIOH

ALIOH simulates removal of acidic contaminants by granular lithium hydroxide and lithium carbonate. The amount of lithium hydroxide consumed per weight of contaminant is calculated from the reaction stoichiometry. This number is input with the contaminant data. Reaction of lithium hydroxide with carbon dioxide to produce lithium carbonate has no noticeable effect on the contaminant removal. The removal efficiency for this device is 100 percent unless the bed is less than 1.905 cm thick or more than 80 percent utilized. The drop in efficiency as the bed is utilized is approximated by the following sine relationship:

$$\eta_r = \sin(m_{\mathrm{LiOH}} - m_{\mathrm{LiOH\ used}})/(0.2)m_{\mathrm{LiOH}} \quad (12)$$

where m_{LiOH} is the mass of the lithium hydroxide bed and m_{LiOH} used is the mass of lithium hydroxide utilized. Efficiency for a bed less than 1.905 cm thick decreases linearly with thickness.

Subroutine COOXID

Ambient temperature catalytic oxidation of carbon monoxide and hydrogen is simulated by the COOXID subroutine. This routine simulates ambient temperature catalytic oxidation using a granular activated charcoal with two weight percent platinum loading. This simulation is effective only for carbon monoxide and hydrogen. Efficiency remains constant at 100 percent per pass unless the residence time falls below 0.2 s. For residence times below 0.2 s, the efficiency decreases linearly according to figure 25.[20]

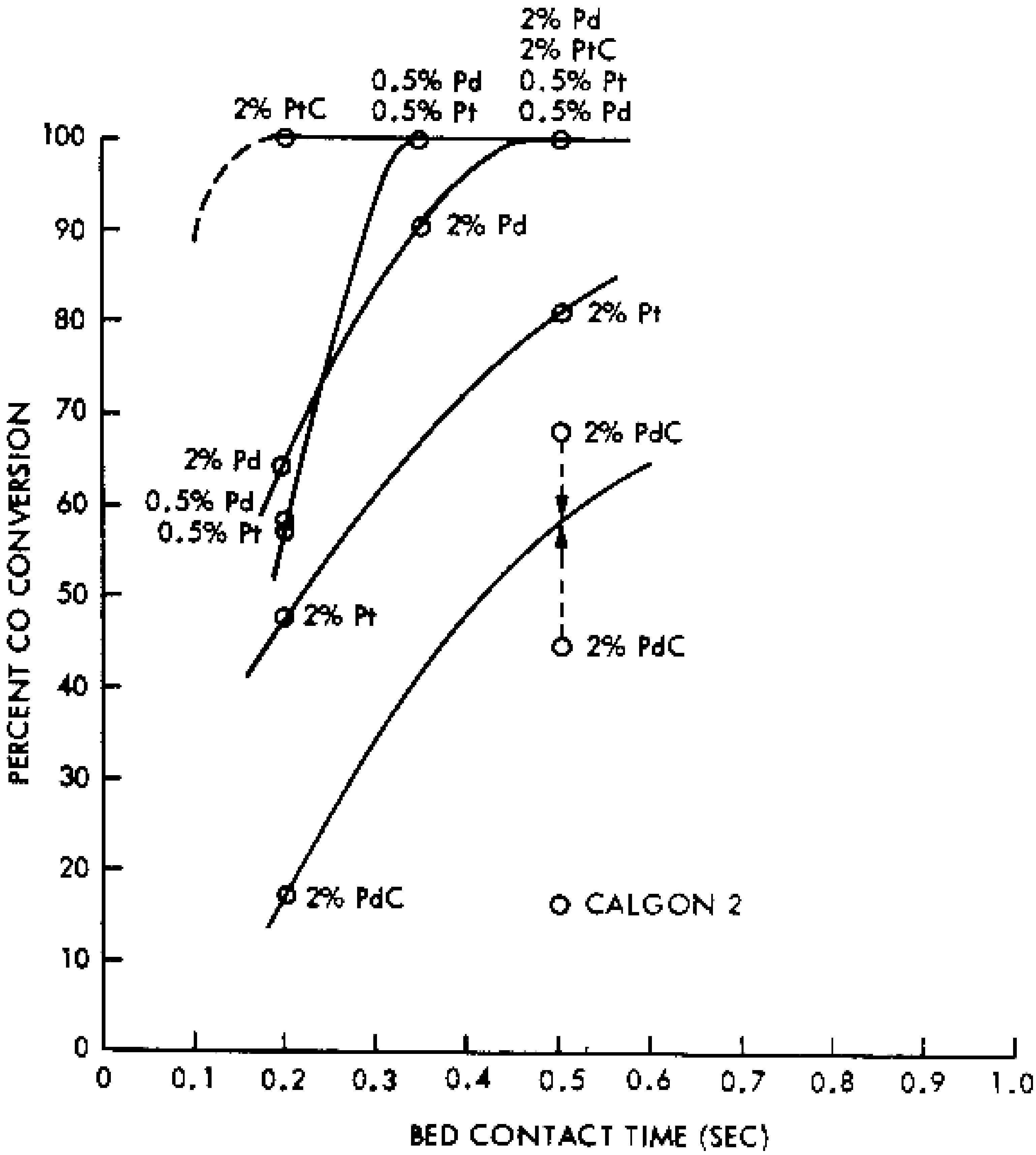
Subroutine CATBNR

The CATBNR subroutine simulates the destruction of hydrogen, carbon monoxide, methane, and other low molecular weight organic contaminants into carbon dioxide and water vapor using high temperature catalytic oxidation. The degree of oxidation in the oxidizer must be input by the user in the contaminant data input file. Typically, oxidation efficiency is based on experimental oxidation performance testing. On average, operating the oxidizer at 400 °C (750 °F) provides removal efficiency of 100 percent for most contaminants.

Subroutine CONDHX

The subroutine, CONDHX, simulates the removal of contaminants by absorption into humidity condensate in a condensing heat exchanger. Some contaminants are removed by this route by not only absorption but also chemical reaction in the condensate. Ammonia is treated in this manner since it dissociates in water and reacts with dissolved carbon dioxide.[21] All other contaminant removal is simulated using Henry's Law. Using Henry's Law is justified for trace contaminants since their concentrations in the atmosphere approach infinite dilution. Henry's Law correlates the concentration of a contaminant in the atmosphere to its concentration in the liquid phase. The correlation coefficient is the Henry's Law Constant, H, which has units of atmospheres/mole fraction. Equation (13) shows the Henry's Law relationship in which p_c is the contaminant partial pressure in atmospheres, H is the Henry's Law constant in atmospheres per mole fraction, and x is the liquid phase mole fraction.

$$p_c = Hx \quad (13)$$



NOTES:

1. ALL NOBLE METALS ON ALUMINA SUBSTRATE UNLESS OTHERWISE STATED
2. 2% PtC = 2% PLATINUM ON CARBON
3. 2% PdC = 2% PALLADIUM ON CARBON

Figure 25. Noble metal CO catalyst performance.

The simulation assumes that the absorption process is cocurrent and that equilibrium is very closely approached. A material balance on this process provides a relationship for the condensate mole fraction shown by:

$$x = y/(C/A+H/P) \ . \tag{14}$$

In this equation, x is the liquid phase mole fraction, y is the vapor phase mole fraction, C is the condensate mass molar flow rate in moles/h, A is the atmospheric molar flow rate in moles/h, H is the Henry's Law constant in atmospheres, and P is the total pressure in atmospheres. Figure 26 illustrates the absorption process. Based on the cabin concentration, the program calculates the inlet mole fraction based on a 1 atmosphere total pressure. The condensate flow rate and atmosphere flow rate are entered in the device definition data and converted to molar flow rates based on 1 atmosphere pressure and 294 K absolute temperature. The mole fraction of contaminant leaving in the condensate is used to determine the mass of contaminant removed. The removal efficiency is calculated from the ratio of the difference in mass of contaminant entering and mass of contaminant removed to the mass of contaminant entering.

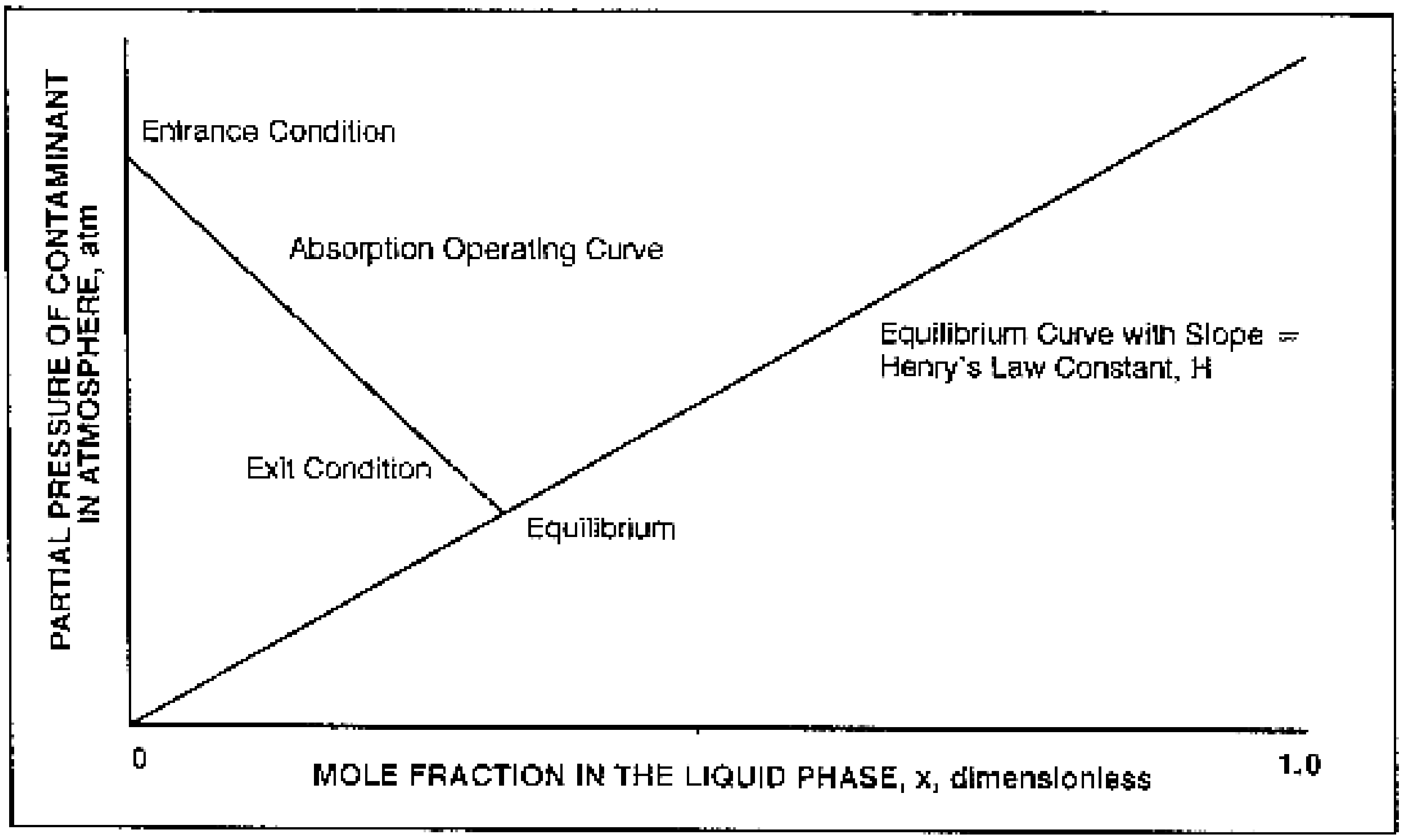


Figure 26. Absorption of contaminants by humidity condensate.

Ammonia removal is treated separately since it reacts chemically with dissolved carbon dioxide in the humidity condensate. According to reference 22, data correlating ammonia partial pressure to liquid phase ammonia concentration for several carbon dioxide atmospheric partial pressures were used to obtain an equation relating liquid and gas phase ammonia composition. These data were obtained by sparging and gas mixture through a volume of water. Figure 27 shows a plot of the result.[22] The carbon dioxide curve corresponding to 666.6 Pa (5 mm Hg) was used to obtain equation (15) which relates ammonia mass per kilogram of condensate to the entering ammonia concentration.

$$m_a = 189.6 C_C^{0.535} \ . \tag{15}$$

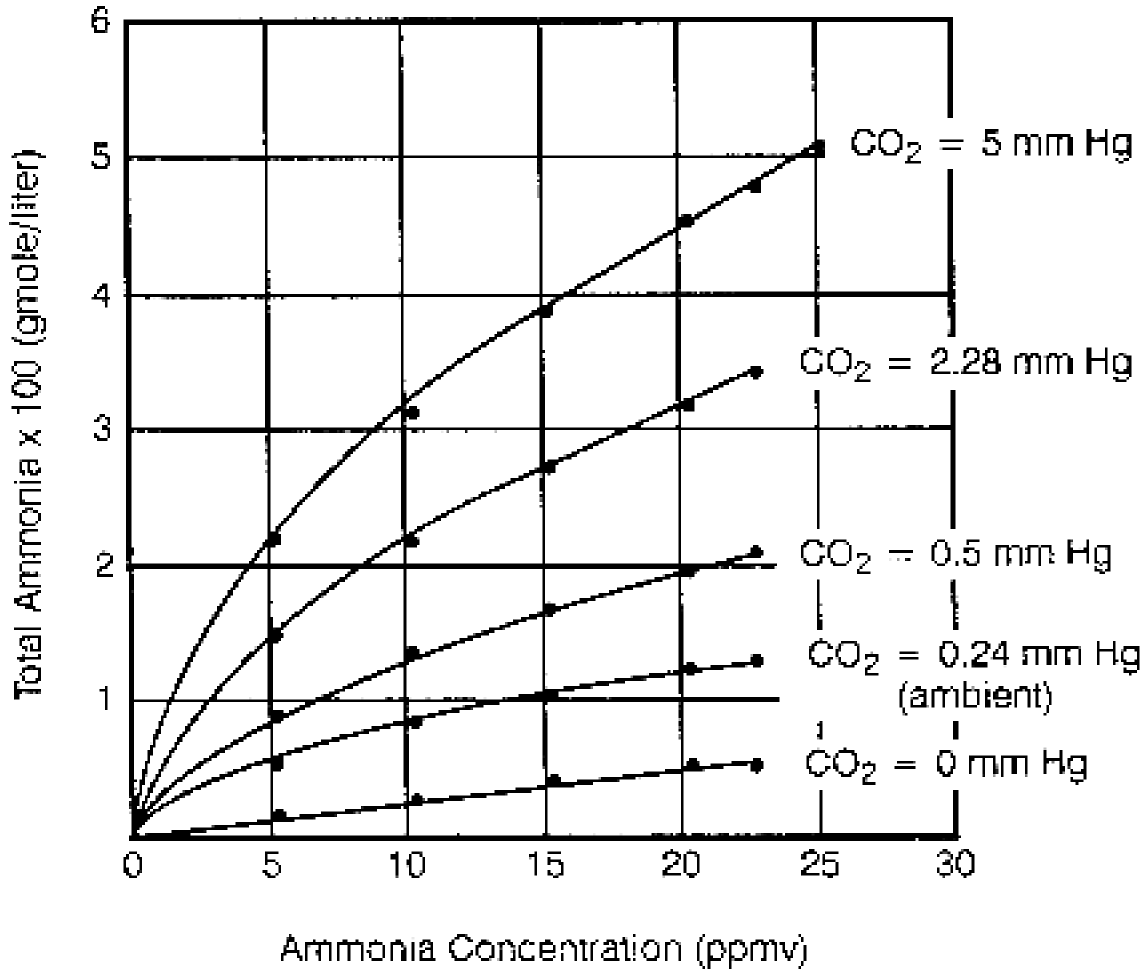


Figure 27. Ammonia solubility in water at varying carbon dioxide concentration.

In this equation, m_a is the mass of ammonia in milligrams per kilogram of condensate and C_c is heat exchanger inlet ammonia concentration in mg/m^3. This equation is used to determine the ammonia removal efficiency from the mass of ammonia entering and leaving the condensing heat exchanger assembly.

Data Input and Output Subroutines

The following subroutines regulate the data input and output for each computer simulation run. These subroutines have been designed to allow maximum flexibility for calculated data output to facilitate data analysis are reporting.

Subroutine CRIN

Subroutine CRIN is called by MAIN and reads contaminant input data into the contaminant name matrix, NN, and the main calculation matrix, CDI.

Subroutine RRIN

Subroutine RRIN is called by MAIN and reads device definition data and time-dependent data into matrices DD and TT, respectively.

Subroutine CROUT2

CROUT2 is called by MAIN and controls output of the contaminant input data to the printer or computer terminal screen. One row at a time without headings is written to these output devices for the user to review before entering the calculation loop.

Subroutine RROUT2

Subroutine RROUT2 is called by MAIN and controls output of the device definition data and time-dependent data to the printer or computer terminal screen. One row at a time without headings is written to these output devices for the user to review before entering the calculation loop.

Subroutine CROUT

CROUT is called by MAIN and regulates output of matrix CC data during each time increment for diagnostic purposes. This subroutine is called only when print switch 5 is set equal to 1.

Subroutine RROUT

Subroutine RROUT is called by MAIN and regulates output of matrix DD for diagnostic purposes. This subroutine is called only when print switch 5 is set equal to 1.

Subroutine DATOUT

DATOUT is called by MAIN and serves as the master output regulation routine. Routines contained within PRFANS are called from DATOUT according to the print switch designations made by the user.

Subroutine GROUP

Subroutine GROUP is called by DATOUT and calculates the toxic hazard index. This subroutine also regulates the output for the toxic hazard index for both the standard formatted output and the plot data output.

Subroutine PRFANS

PRFANS contains several subroutines that are called by DATOUT which regulate the output for contaminant concentration data, sum of contaminant masses removed data, and removal device efficiency data. This subroutine regulates output for both the standard formatted output and the plot data output.

Subroutine HEADGS

Subroutines within HEADGS are called by PRFANS subroutines to regulate standard formatted data output headings. Headings are provided for contaminant concentration data, contaminant removal rate data, sum of contaminant masses removed data, and removal device efficiency data.

APPENDIX A

TRACE CONTAMINANT CONTROL SIMULATION (TCCS) COMPUTER PROGRAM FORTRAN CODE LISTING

This appendix contains listings for each major subroutine and the main TCCS computer program. The main program is listed first followed by listings of each subroutine. The subroutines listings are arranged in alphabetical order by name to provide easy reference.

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\MAIN.FO Options: /C 80 /L /BY     05/21/92 12:57:09

C FILE:MAIN08.FOR
        PROGRAM TCCS08
C       ****************************************************************
C       ****************************************************************
C       *            PROGRAM TO MODEL REMOVAL OF SPACECRAFT            *
C       *                    GASEOUS CONTAMINANTS                      *
C       *                     VERSION  8.0 Alpha                       *
C       *                      APRIL  10,  1992                        *
C       ****************************************************************
C       ****************************************************************
C       SUBROUTINES REQUIRED:
C         CAFILL-FILL MATRIX WITH ZEROS
C         RAFILL-FILL MATRIX WITH ZEROS
C         CRIN-READ IN INPUT DATA
C         RRIN-READ IN INPUT DATA
C         CROUT-PRINT OUT INPUT DATA
C         RROUT-PRINT OUT INPUT DATA
C         PCSET-PRECALCULATION SET UP ROUTINE
C         MCALC-MAIN CALCULATION ROUTINE
C         DATOUT-DATA PRINTOUT ROUTINE
C         XXXXX-TIME DEPENDENT DATA ROUTINE
C         REGEN-REGENERATION OF DEVICES ROUTINE
C         SLIOH-SUM LIOH USED IN TIME INCREMENT

C       NOTE:SUBROUTINES USE ADJUSTABLE SIZE ARRAYS
C            WATCH COMPILER OPTIONS/DIMENSIONING IF
C            ANY ARRAY IS LARGER THAN 64K BYTES
C
C       ******************DIMENSION MAIN PROGRAM MATRICES***************
C       NN=CONTAMINANT NAME MATRIX
C       CDI=CONTAMINANT INPUT DATA MATRIX
C       CC=CALCULATON MATRIX
C       DD=DEVICE DEFINITION MATRIX
C       TT=TIME DEPENDENT DATA MATRIX
C
C       PRESENTLY SET TO HANDLE MAXIMUM OF 150 CONTAMINANTS IN MATRICES
C       THIS VALUE =NROW AND IS USED IN ADDRESSING ADJUSTABLE SIZE
C            ARRAYS IN SUBROUTINES
C       NTTROW IS USED FOR MATRIX TT MAXIMUM LENGTH
C
        CHARACTER NN(300)*30
        REAL CDI(300,23)
        REAL CC(300,48)
C       REAL DD(15,21)
        REAL DD(15,23)
        REAL TT(300,7)

```

```
C       NOTE:MUST COMPILE SUBROUTINES PROPERLY FOR ADJUSTABLE SIZE
C       ARRAYS IF A MAIN MATRIX EXCEEDS 65536 BYTES (REALS=4 BYTES)
C
C       ************* DECLARE OTHER TERMS USED IN MAIN PROGRAM  ***********
        CHARACTER FNAME*24,DES*1,FCPLOT*24,FTPLOT*24,FEPLOT*24
        LOGICAL EX
        INTEGER PRTSW1,PRTSW2,PRTSW3,PRTSW4,PRTSW5,PRTSW6,PRTSW7,PRTSW8,
     +  TVAL,IDEVNO,IDEVN1,IDEVN2,IDEVN3,PRTSW9

C       ******* PRINT WELCOME AND PROGRAM VERSION NUMBER ****************
        WRITE (*,9)
  009     FORMAT  (1X,'****************************************************'/
     +            1X,'*              WELCOME TO THE WORLD                  *'/
     +            1X,'*                    OF THE                          *'/
     +            1X,'*   SPACECRAFT ATMOSPHERIC TRACE CONTAMINATION       *'/
     +            1X,'*          CONTROL SIMULATION PROGRAM                *'/
     +            1X,'*              -VERSION 8.0  Alpha-                  *'/
     +            1X,'*                April 10, 1992                      *'/
     +            1X,'****************************************************')
C
C       ******* DEFINE PROGRAM VARIABLES **********************************
C       LIN=NO. OF LINES OF DATA IN MAT NN & MAT CDI & MAT CC
C       LIN1=NO. LINES OF DATA IN MAT TT
C       LIN2=NO. LINES OF DATA IN MAT DD
C       TN=INCREMENT END TIME (HRS)
C       TN1=INCREMENT BEGINNING TIME (HRS)
C       TMIS=TOTAL MISSION TIME (HRS)
C       NINC=NUMBER OF TIME INCREMENTS ELAPSED
C
C       MAT NN,CC, AND CDI MUST HAVE SAME NO. OF ROWS
C       DIMENSIONS OF MAT DD
        NROW=15
C       NCOL=21
        NCOL=23
C       DIMENSIONS OF MAT CC & ROWS IN MAT NN
        NROW1=300
        NCOL1=48
C       DIMENSIONS OF MAT CDI
        NROW2=NROW1
        NCOL2=23
C       DIMENSIONS OF MAT TT
        NTTROW=300
        NTTCOL=7
C       DEVICE NUMBER FOR OUTPUT DATA (SET TO 6 FOR FORM FEED ON OUTPUT)
        IDEVNO=6
C       DEVICE NUMBER FOR MESSAGE OUTPUT
        IMSGDN=2
C       DEVICE NUMBER FOR CONTAMINANT PLOT DATA
        IDEVN1=10
```

```
C       DEVICE NUMBER FOR T-VALUE PLOT DATA
        IDEVN3=11
C       DEVICE NUMBER FOR EFFICIENCY PLOT DATA
        IDEVN2=12
C       ***********   END OF DEFINITION SECTION *************************
C
C       **************** ZERO MATRICES************************************
C       PUT BLANKS IN NAME MATRIX
    011 CALL CAFILL(NN,1,NROW1)
C       PUT ZEROS IN OTHER MATRICES
        CALL RAFILL(CDI,NROW2,NCOL2)
        CALL RAFILL(CC,NROW1,NCOL1)
        CALL RAFILL(DD,NROW,NCOL)
        CALL RAFILL(TT,NTTROW,NTTCOL)
C
C       ******* READ IN DATA FROM FILES AND PRINT IT IF DESIRED ********
C
    010 WRITE(*,*)'INPUT CONTAMINANT DATA FILE NAME:  '
        CALL CRIN(NN,CDI,NROW2,NCOL2,LIN)
    012 WRITE(*,*)'PRINT CONTAMINANT INPUT DATA? (Y/N) '
        READ(*,'(A)')DES
        IF((DES.NE.'Y') .AND. (DES.NE.'N')) GOTO 12
        IF (DES.EQ.'N') GOTO 20
        CALL CROUT2(NN,CDI,NROW2,NCOL2,1,NCOL2,LIN,1,LIN,IMSGDN)
C
    020 WRITE(*,*)'INPUT DEVICE DEFINITION TABLE FILE NAME:  '
C       NOTE: ONLY 16 COLUMNS ARE IN THE INPUT FILE
        CALL RRIN(DD,NROW,NCOL,16,LIN2)
    022 WRITE(*,*)'PRINT DEVICE DEFINITION TABLE? (Y/N) '
        READ(*,'(A)')DES
        IF((DES.NE.'Y') .AND. (DES.NE.'N'))  GOTO 22
        IF(DES.EQ.'N') GOTO 30
        CALL RROUT2(DD,NROW,NCOL,1,16,LIN2,IMSGDN)
C
    030 WRITE(*,*)'INPUT TIME DEPENDENT DATA FILE NAME:  '
        CALL RRIN(TT,NTTROW,NTTCOL,NTTCOL,LIN1)
    032 WRITE(*,*)'PRINT TIME DEPENDENT DATA? (Y/N) '
        READ(*,'(A)')DES
        IF((DES.NE.'Y') .AND. (DES.NE.'N'))  GOTO 32
        IF (DES.EQ.'N') GOTO 40
        CALL RROUT2(TT,NTTROW,NTTCOL,1,NTTCOL,LIN1,IMSGDN)
C
C       *************READ IN MISSION DATA VARIABLES ********************
C       READ IN MISSION TOTAL TIME (HRS)
    040 WRITE(*,*)'INPUT TOTAL MISSION TIME IN HOURS:  '
        READ(*,*) TMIS
C
C        ********** PRINT SWITCH DEFINITION ****************************
C      1=RESULTS FOR ONE CONTAMINANT IN PCSET
C      2=RESULTS FOR 1 CONT & INCR IN 1/20 INCR CONV ROUTINE (IN MCALC)
```

```
C      3=CONVERGENCE VALUES IN CONVRG
C      4=RESULTS FOR 1 CONT IN MCALC AFTER CAV CALC
C      5=MAT CC AND MAT DD AT END OF TIME INCREMENT
C      6=PRINT CONC+M.REM+SUM MASS REM+REM EFF(OTHERWISE ONLY CONC DATA)
C      7=PRINT OUTPUT WITH NO FORM FEEDS
C      8=PRINT ANSWERS DURING EACH ITERATION (IN MAIN PROGRAM)
C      9=CONTROL PLOT FILE OUTPUT
C      TVAL=CONTROL OUTPUT OF GROUP CONTRIBUTION T-VALUE DATA
        PRTSW1=NINT(DD(2,9))
        PRTSW2=NINT(DD(2,10))
        PRTSW3=NINT(DD(2,11))
        PRTSW4=NINT(DD(2,12))
        PRTSW5=NINT(DD(2,13))
        PRTSW6=NINT(DD(2,14))
        PRTSW7=NINT(DD(2,15))
        PRTSW8=NINT(DD(2,16))
C
C       **** MAKE DECISION ON CONCENTRATION AND EFFICIENCY PLOT DATA ****
        WRITE (*,*) 'DO YOU WISH TO WRITE INCREMENT DATA TO A PLOT FILE?'
        WRITE (*,*) '      1. Concentration Data (C)'
        WRITE (*,*) '      2. Efficiency Data (E)'
        WRITE (*,*) '      3. Both Concentration and Efficiency Data (B)'
        WRITE (*,*) '      4. Neither (N)'
        WRITE (*,*) 'ENTER YOUR SELECTION:  '
         READ (*,'(A)') DES
        IF (DES.EQ.'C') THEN
          PRTSW9=1
        ELSEIF (DES.EQ.'E') THEN
          PRTSW9=2
        ELSEIF (DES.EQ.'B') THEN
          PRTSW9=3
        ELSE
          PRTSW9=0
        ENDIF
C
C      *************** MAKE DECISION ON T-VALUE OUTPUT ****************
        WRITE (*,*) 'PRINT GROUP CONTRIBUTION T-VALUE DATA?'
        WRITE (*,*) '      1. Print to Normal Output (Y)'
        WRITE (*,*) '      2. Print to Normal Output and Plot File (P)'
        WRITE (*,*) '      3. Do Not Print (N)'
       WRITE (*,*) 'ENTER YOUR SELECTION:  '
          READ (*,'(A)')DES
        IF (DES.EQ.'Y') THEN
          TVAL=1
        ELSEIF (DES.EQ.'P') THEN
          TVAL=2
        ELSE
          TVAL=3
        ENDIF
C
```

```
C      CHANGE TO NO FORM FEED IF PRTSW7=1
         IF (PRTSW7.EQ.1) THEN
            IDEVNO=7
         ENDIF

C      ***************  MAKE DECISIONS ON DATA OUTPUT *****************
C        THIS IS WHERE ALL PROGRAM OUTPUT DATA FILES ARE OPENED
C        THEY MUST BE CLOSED AT THE END OF THE PROGRAM

C      SECTION WHICH CHECKS FOR EXISTANCE OF OUTPUT FILE & OPENS IT
  050 WRITE(*,*) ' WRITE OUTPUT TO FILE, PRINTER, SCREEN, OR END?'
      WRITE (*,*)'      (FILE NAME/LPT1/CON/END) '
C     *********** NOTE:  LPT1 OUTPUT REQUIRES 132 COLUMNS ***********
      READ(*,'(A)') FNAME
C     QUIT IF FNAME=END
      IF(FNAME.EQ.'END') GOTO 999
      IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) THEN
          INQUIRE(FILE=FNAME,EXIST=EX)
          IF (EX) THEN
             WRITE(*,*)'FILE EXISTS - OVERWRITE? (Y/N) '
             READ(*,'(A)') DES
             IF (DES.NE.'Y') THEN
                GOTO 50
             ELSE
                OPEN(IDEVNO,FILE=FNAME,STATUS='OLD',IOSTAT=IOVAL)
             ENDIF
          ELSE
           OPEN (IDEVNO,FILE=FNAME,STATUS='NEW',IOSTAT=IOVAL)
          ENDIF
      ENDIF
      IF ((FNAME.EQ.'LPT1').OR.(FNAME.EQ.'CON')) THEN
          OPEN(IDEVNO,FILE=FNAME,IOSTAT=IOVAL)
      ENDIF
      IF(IOVAL.NE.0) THEN
          CLOSE (IDEVNO)
          WRITE(*,*)'IOERROR= ',IOVAL
          GOTO 50
      ENDIF
C
C     ****** OPEN FILE FOR CONCENTRATION PLOT DATA IF PRTSW9=1 OR 3 ******
C
      IF ((PRTSW9.EQ.1).OR.(PRTSW9.EQ.3)) THEN
  052   WRITE (*,*) 'FILE NAME FOR CONCENTRATION PLOT DATA OUTPUT? '
             READ (*,'(A)') FCPLOT
          INQUIRE (FILE=FCPLOT,EXIST=EX)
          IF (EX) THEN
             WRITE (*,*) 'PLOT FILE EXISTS - OVERWRITE? (Y/N) '
                READ (*,'(A)') DES
             IF (DES.NE.'Y') THEN
                GOTO 52
```

```
                ELSE
                  OPEN (UNIT=10,FILE=FCPLOT,STATUS='OLD',IOSTAT=IOVAL)
                ENDIF
              ENDIF
              OPEN (UNIT=10,FILE=FCPLOT,STATUS='NEW',IOSTAT=IOVAL)
            ENDIF
            IF (IOVAL.NE.0) THEN
              CLOSE (UNIT=10)
              WRITE (*,*) 'IOERROR= ',IOVAL
              GOTO 52
            ENDIF
C
C           ******* OPEN FILE FOR EFFICIENCY PLOT DATA IF PRTSW9=2 OR 3 ******
C
            IF ((PRTSW9.EQ.2).OR.(PRTSW9.EQ.3)) THEN
   054       WRITE (*,*) 'FILE NAME FOR EFFICIENCY PLOT DATA OUTPUT? '
                READ (*,'(A)') FEPLOT
              INQUIRE (FILE=FEPLOT,EXIST=EX)
              IF (EX) THEN
                WRITE (*,*) 'PLOT FILE EXISTS - OVERWRITE? (Y/N) '
                  READ (*,'(A)') DES
                IF (DES.NE.'Y') THEN
                  GOTO 54
                ELSE
                  OPEN (UNIT=12,FILE=FEPLOT,STATUS='OLD',IOSTAT=IOVAL)
                ENDIF
              ENDIF
              OPEN (UNIT=12,FILE=FEPLOT,STATUS='NEW',IOSTAT=IOVAL)
            ENDIF
            IF (IOVAL.NE.0) THEN
              CLOSE (UNIT=12)
              WRITE (*,*) 'IOERROR= ',IOVAL
              GOTO 54
            ENDIF
C
C           ********** OPEN FILE FOR T-VALUE PLOT DATA IF TVAL=2 ***********
C
            IF (TVAL.EQ.2) THEN
   056       WRITE (*,*) 'FILE NAME FOR T-VALUE PLOT DATA OUTPUT? '
               READ (*,'(A)') FTPLOT
              INQUIRE (FILE=FTPLOT,EXIST=EX)
              IF (EX) THEN
               WRITE (*,*) 'PLOT FILE EXISTS - OVERWRITE? (Y/N) '
                 READ (*,'(A)') DES
               IF (DES.NE.'Y') THEN
                 GOTO 56
               ELSE
                 OPEN (UNIT=11,FILE=FTPLOT,STATUS='OLD',IOSTAT=IOVAL)
               ENDIF
              ENDIF
```

```
            OPEN (UNIT=11,FILE=FTPLOT,STATUS='NEW',IOSTAT=IOVAL)
          ENDIF
          IF (IOVAL.NE.0) THEN
            CLOSE (UNIT=11)
            WRITE (*,*) 'IOERROR= ',IOVAL
            GOTO 56
          ENDIF
C
C
C         CALL SYSTEM TIME AND DATE
C         THIS MUST BE CALLED ONLY ONCE SO THAT THE TIME AND DATE WILL
C           BE THE SAME ON ALL OUTPUT INFORMATION FOR ONE RUN
          CALL DATTM(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,ISECOND)

C         SET IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUTS
          IPGCTR=0

C         ********* CHECK BASIC TIME INCREMENT **************************
C         ********* BASIC TIME INCREMENT (HRS)-DD(1,11) *****************
          BINC=DD(1,11)
C         ********* TEST FOR BINC=0 (CAUSES ENDLESS TIME LOOP) **********
          IF (BINC.EQ.0) THEN
            CLOSE (IDEVNO)
            WRITE(*,*)' ERROR-BASIC TIME INCREMENT=0'
            GOTO 999
          ENDIF

C         **************  ZERO INITIAL VARIABLES *************************
          TN=0
          TN1=0
          NINC=0
C
C         *******************************************************************
C         *           PRECALCULATION SET UP ROUTINE                         *
C         *******************************************************************
C         FOR ALL CONTAMINANTS ONE AT A TIME AT CAV PRED=1E-20, CALC INIT
C         DEV EFF AND LOAD IT INTO MAT CC-ALSO CALC CAVPRD(CAV PREDICTED)
C     OUTPUTS TO PRECALC SET UP ROUTINE:
C      TN1=INCREMENT INITIAL TIME (HRS)
C      BINC=BASIC INCREMENT SIZE (HRS) (REF.=DD(1,11)) PASS IN????
C      LIN=NO. OF CONT IN MAT CC AND NN
C      DD,NROW,NCOL=NAME & SIZE OF MAT DD
C      CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
C      CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
C      LIN2=NO. DEVICES IN MAT DD
C      NN=NAME OF MAT NN
C     INPUTS FROM PRECALC SETUP ROUTINE-SUBROUTINE PCSET:
C      PUT IN MAT CC
C         CAVPRD=PRED CABIN AV CONC (MG/CU M): =CC(I,2)
C         CEQLIB=EQUILIBRIUM CABIN CONT CONC (MG/CU M):=CC(I,3)
```

```
C       CFINAL=FINAL CABIN CONT CONC (MG/CU M):=CC(I,4)
C       PUTS REM EFF FROM DD COL 20 IN CC(I,7-10-13-16 ETC)
C       PUTS M.REM IN CC(I,5-8-11-14..)

        CALL PCSET(TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +CDI,NROW2,NCOL2,LIN2,NN,PRTSW1,IMSGDN)

C       ************** END OF PRECALCULATION SETUP ROUTINE ***************

C       ******************************************************************
C       *         BEGINNING OF CALCULATION FOR EACH TIME INCREMENT       *
C       ******************************************************************
  100 CONTINUE
C
C       *********  INCREASE INCREMENT COUNTER  ***************************
        NINC=NINC+1
C       *********  CHECK FOR CHANGES IN BASIC TIME INCREMENT  ***********
        DO 105 K=1,LIN1
          IF (TT(K,1).EQ.TN1) THEN
            IF ((TT(K,4).EQ.1).AND.(TT(K,6).EQ.11)) THEN
              BINC=TT(K,7)
            ENDIF
          ENDIF
  105 CONTINUE
C       *********  SET UP TIME INCREMENT SIZE FOR INCREMENT *************
        IF (NINC.EQ.1) TN=BINC/24
        IF (NINC.EQ.2) TN=BINC/2
        IF (NINC.EQ.3) TN=BINC
        IF (NINC.GT.3) TN=TN+BINC
C       CHECK FOR INCREMENT FINAL TIME > MISSION TIME
        IF(TN.GT.TMIS) TN=TMIS

C       ****** LIST INCREMENT NO. AND TIMES TO CONSOLE ******************
        IF((FNAME.NE.'CON').OR.(PRTSW8.NE.1))THEN
          OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
           WRITE(IMSGDN,65)NINC,TN1,TN
  065      FORMAT (1X,'INCR NO.= ',I5,' BEGIN & END TIMES (hours)=',
     +     F8.3,2X,F8.3)
          CLOSE (IMSGDN)
        ENDIF

C       ***********  READ TIME DEPENDENT DATA ***************************
C       CALL TIME DEPENDENT DATA SUBROUTINE-RINCDD
        CALL RINCDD(I,TN,TN1,DD,NROW,NCOL,LIN2,
     +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,LIN,TT,NTTROW,NTTCOL,LIN1)

C       ****STORE ORIGINAL Q DEVICE IN DD COL 7 (TAKEN FROM DD COL 2)****
        DO 110 J=1,LIN2
          DD(J,7)=DD(J,2)
  110 CONTINUE
```

```
C       ************ CHECK FOR REGENERATION IN TIME INCREMENT ***********
C       CALL REGENERATION SUBROUTINE REGEN
        CALL REGEN(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
       +CDI,NROW2,NCOL2,LIN,LIN2,IMSGDN)

C       *********** CALL MAIN CALCULATION SUBROUTINE  *******************

C  OUTPUTS TO MAIN CALC SUBROUTINE-MCALC:
C    I=CONTAMINANT NO.
C    TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
C    DD,NROW,NCOL=NAME & DIM OF MAT DD
C    CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C    CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C    NN=NAME OF MAT NN
C    LIN=NUMBER OF CONTAMINANTS IN MAT NN & CDI
C    LIN2=NO. DEVICES IN MAT DD
C  INPUTS FROM MAIN CALC ROUTINE-MCALC:
C  TO MAT CC
C    PUTS CAVCLC,CEQLIV,&CFINAL IN CC(I,2-3 &4)
C    PUTS REM EFF FROM DD COL20 IN CC(I,7-10-13 ETC)
C    PUTS M.REM FOR EACH DEV FROM DD COL21 IN CC(I,6-9-12 ETC)
C    PUTS SUM MASS REM FOR EACH DEV IN CC(I,8-11-14 ETC)


        CALL MCALC(I,TN,TN1,DD,NROW,NCOL,
       +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,LIN,LIN2,
       +PRTSW2,PRTSW3,PRTSW4,IMSGDN)

C       ************* CALCULATE LIOH USED IN INCREMENT ******************
C       CALL LIOH REMOVAL SUBROUTINE SLIOH
        CALL SLIOH(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
       +CDI,NROW2,NCOL2,LIN,LIN2)
C
C       ************* RESTORE DEVICE FLOW ********************************
C       RESTORE ORIGINAL DEVICE FLOW RATE FROM DD COL 7 TO DD COL 2
        DO 120 J=1,LIN2
          DD(J,2)=DD(J,7)
  120 CONTINUE

C       ********* PRINTOUT OF DATA FOR EACH TIME INCREMENT **************
C       IF PRTSW5=1 THEN PRINT MAT DD+MAT CC INFO FOR THIS CONTAMINANT
        IF (PRTSW5.EQ.1)  THEN
C         OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
C           WRITE(IMSGDN,*)'PRINTOUT FOR MAT CC & DD AT END OF TIME INCR'
C           WRITE(IMSGDN,*)'INFO FROM MAT CC'
C         CLOSE(IMSGDN)
C         CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,1,LIN,IMSGDN)
          CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,1,LIN,IMSGDN,NINC,
       +   FNAME,IDEVNO,IOVAL)
```

```
C           OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
C              WRITE(IMSGDN,*)'INFO FROM MAT DD'
C           CLOSE(IMSGDN)
C           CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN)
            CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN,FNAME,IDEVNO,IOVAL)
          ENDIF

C         ************ REGULAR PRINTOUT OF DATA FOR EACH INCREMENT ********
          IF (PRTSW8.EQ.1) THEN
            CALL DATOUT(TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +      CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
     +      IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
     +      TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)

            IF (IOVAL.NE.0) THEN
              CLOSE (IDEVNO)
              WRITE(*,*)'PROGRAM DATA OUTPUT ERROR IN INCREMENT = ',NINC
              GOTO 999
            ENDIF
          ENDIF
C         ********* CONTROLS PLOT DATA OUTPUT IF PRTSW8=0 ****************
          IF (PRTSW8.EQ.0) THEN
            IF ((PRTSW9.GT.0).OR.(TVAL.EQ.2)) THEN
              CALL DATOUT (TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +      CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
     +       IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
     +      IPGCTR,TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)
            ENDIF
          ENDIF
C
C        ********************* CHECK FOR END OF MISSION *****************
          IF(TN.GE.TMIS) THEN
C           END TIME LOOP
            CONTINUE
          ELSE
C        ********* UPDATE FOR NEXT TIME INCREMENT AND REPEAT ************
C           SET TFINAL FOR THIS INCR = TINIT FOR NEXT INCR
              TN1=TN
C          SET CFINAL FOR INCR=CINIT FOR NEXT INCR-ALL CONTAMINANTS
              DO 130 I=1,LIN
                CC(I,1)=CC(I,4)
  130         CONTINUE
            GOTO 100
          ENDIF
C
C        ****************************************************************
C        *          END OF CALCULATION FOR EACH TIME INTERVAL           *
C        ****************************************************************

C        ****** PRINT FINAL ANSWERS AT END OF MISSION IF DESIRED ********
```

```
  900 CONTINUE
        IF (PRTSW8.EQ.0) THEN
        CALL DATOUT(TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +    CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
     +     IDEVNO,-1,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
     +    TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)

          IF (IOVAL.NE.0) THEN
            CLOSE (IDEVNO)
            WRITE(*,*)'PROGRAM DATA OUTPUT ERROR - FINAL PRINTOUT'
            GOTO 999
          ENDIF
        ENDIF
C      ************ CLOSE ALL PROGRAM OUTPUT FILES *******************
        CLOSE (IDEVNO)
        CLOSE (IDEVN1)
        CLOSE (IDEVN3)
        CLOSE (IDEVN2)
C
  999 CONTINUE
      WRITE(*,*)'DO YOU WISH TO RUN ANOTHER CASE? (Y/N)'
          READ (*,'(A)')DES
        IF (DES.EQ.'Y') THEN
          GO TO 011
        ENDIF
C       ******************************************************************
C       *                          END OF MAIN PROGRAM                   *
C       ******************************************************************
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\ACHBD.F Options: /C 80 /L /BY   05/21/92 12:54:30

```
C       ******************************************************
C       * FILE:ACHBD.FOR                                     *
C       * SUBROUTINE FOR REM EFF-AXIAL FLOW CHARCOAL BED     *
C       * DOESNT ALLOW FOR DESORPTION                        *
C       ******************************************************
C
        SUBROUTINE ACHBD (TN,TN1,CIIN,TCABIN,COEXIS,BEDQ,EMAX,BEDL,
        +BEDDIA,DENCH,TRTTYP,DCONT,VMOL,MW,VCONC,SOL,SMR,EFF)
C       OUTPUT:
C         EFF=BED REMOVAL EFF(DEC)
C       INPUTS:
C         TN,TNI=INCREMENT INITIAL AND FINAL TIMES(HR)
C         CIIN=BED INLET CONT CONC (MG/CU M)
C         TCABIN=CABIN TEMP (DEG K)
C         COEXIS=COEXISTANCE FACTOR
C         BEDQ=BED FLOW RATE(CU M/HR)
C         EMAX=MAXIMUM BED EFF (DEC)
C         BEDL=BED LENGTH (M)
C         BEDDIA=BED DIAMETER (M)
C         DENCH=DENSITY OF CHARCOAL IN BED (KG/CU M)
C         TRTTYP=BED TREATMENT TYPE(1=CI CHAR,2=PHOS ACID, OTHER #=NONE)
C         DCONT=CONT LIQUID DENSITY (GM/CC)
C         VMOL=CONT MOLAR VOL(GM/CC)
C         MW=CONT MOLECULAR WGT
C         VCONC=CONT VAPOR CONCENTRATION AT TCABIN (MG/CU M)
C         SOL=HENRY'S LAW CONSTANT FOR WATER SOLUBILITY (ATM/MOLE FRACTION)
C         SMR=SUM OF CONT MASS STORED IN BED(MG)-FROM LAST INCR
C
        REAL LPREV, LAVN1, LUTIL, LIMM, LAVAV, LADS,MW
        INTEGER FACID,FCI
C
C       SET CIN=CIIN (THIS PREVENTS CIN FROM BEING PASSED BACK UP
C               TO OTHER SUBROUTINES IF IT IS SET TO 1E-20)
        CIN=CIIN
C       BED TREATMENT LOGIC
C         FACID=FLAG IF BED IS TREATED WITH PHOSPHORIC ACID (Y=1 N=0)
C         FCI=FLAG FOR CI CHAR IN BED (REMOVES FORMALDAHYDE)
        IF (NINT(TRTTYP).EQ.2) THEN
          FACID=1
          FCI=0
        ELSEIF (NINT(TRTTYP).EQ.1) THEN
          FACID=0
          FCI=1
        ELSE
          FACID=0
          FCI=0
        ENDIF
```

```
C
C       TEST FOR NO BED FLOW(BEDQ=<0) OR TN-TN1<=0;BEDL,BEDDIA,DENCH=0
        IF((BEDQ.LE.0).OR.(TN-TN1.LE.0).OR.(BEDL.LE.0).OR.(BEDDIA.LE.0)
     +.OR.(DENCH.LE.0)) THEN
          EFF=0
          GOTO 199
        ENDIF
C       TEST FOR CI CHARCOAL AND FORMALDEHYDE(FCI=1 AND MW=30.03)
        IF((MW.EQ.30.03).AND.(FCI.EQ.1)) THEN
          CALL CICH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR,BEDQ)
          GOTO 199
        ENDIF
C
C       TEST FOR AMMONIA AND PHOS ACID ON CHAR(FACID=1 AND MW=17.0 )
        IF ((MW.EQ.17.0).AND.(FACID.EQ.1)) THEN
          CALL ACIDCH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR)
          GOTO 199
        ENDIF
C
C       TEST FOR MOL VOL=0 (NO CHAR REMOVAL)
        IF (VMOL.EQ.0) THEN
          EFF=0
          GOTO 199
        ENDIF
C
C       CHARCOAL REMOVAL EFFICIENCY CALCULATION
C       SUPERFICIAL BED VEL(FT/MIN)
        BEDVEL=BEDQ*.06960/BEDDIA**2
C       TEST FOR CIN TOO SMALL IN AVAL CALC
        IF (CIN.LT.1E-20) CIN=1E-20
        AVAL=(TCABIN/VMOL)*LOG10(VCONC/CIN)
C       ADS ZONE LENGTH FOR 90% REMOVAL (M)
        LADS=AVAL*.000275*(BEDVEL/1.3)**.8
C       GET QI(CC LIQ CONT/GM CHAR)
        CALL FQI(AVAL,QI,FACID,SOL)
C       LENGTH OF BED PREVIOUSLY USED BY CONT AT THIS C INLET (M)
        LPREV=SMR*1.273E-6*COEXIS/(DCONT*DENCH*BEDDIA**2*QI)
C       RATE OF BED USAGE (M BED/ MG CONT)
        LIMM=1.273E-6*COEXIS/(DCONT*DENCH*BEDDIA**2*QI)
C       LENGTH OF BED AVAILABLE FOR ADS ZONE AT BEGINNING OF INCR (M)
        LAVN1=BEDL-LPREV
        IF (LAVN1.LT.0) LAVN1=0
C       FIX HERE IF DESORPTION IS DESIRED
        IF (LAVN1/LADS.GT.20) THEN
          EFFAV=EMAX
        ELSE
C         INIT INCR EFF BASED ON C IN AND BED L AVAIL AT BEG OF INCR(DEC)
          EFAVN1=EMAX*(1-EXP(-2.3025851*LAVN1/LADS))
C         LOOP FOR EFFICIENCY
          EFFAV=EFAVN1
```

```
           DO  399 J=1,10,1
C             LENGTH OF BED UTILIZ IN INCR (M)
              LUTIL=CIN*BEDQ*EFFAV*(TN-TN1)*LIMM
              IF (LUTIL.GT.LAVN1) THEN
                 GOTO 299
              ELSE
C                AVERAGE BED LENGTH AVAIL (M)
                 LAVAV=LAVN1-LUTIL/2
                 IF ((LAVAV/LADS).GE.20) THEN
                   EFFAV=EMAX
                   GOTO 299
                 ELSE
C                   AV EFF BASED ON AV BED L AVAIL (DEC)
                    EFFAV=EMAX*(1-EXP(-2.3025851*LAVAV/LADS))
                 ENDIF
              ENDIF
  399   CONTINUE
  299 ENDIF
C     MAX EFF BASED ON C IN AND RATE OF BED USAGE (DEC)
        EFFMAX=LAVN1/(CIN*BEDQ* (TN-TN1)*LIMM)
        IF (EFFAV.GT.EFFMAX) EFFAV=EFFMAX
        IF (EFFAV.LT.0) EFFAV=0
        IF (EFFAV.GT.EMAX) EFFAV=EMAX
C       EFF=ACTUAL EFF OUTPUT FROM SUBROUTINE
        EFF=EFFAV
C       REMOVE THIS CHECK IF DESORPTION IS ADDED
  199 IF (EFF.LT.0)  EFF=0
        IF (EFF.GT.EMAX) EFF=EMAX
        RETURN
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS    IN PROGRAM UNIT: 0

C         *****************************************************
C         * SUBROUTINE ACIDCH   -  CALCULATES REMOVAL EFF     *
C         * BED WITH NH3 AND .2 MILLIMOLE H3PO4 ON CHAR       *
C         *****************************************************
          SUBROUTINE ACIDCH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR)
C       OUTPUTS
C         EFF=OUTPUT REMOVAL EFF (DEC)
C       INPUTS
C         EMAX=MAXIMUM BED REMOVAL EFF (DEC)
C         BEDL=BED LENGTH (M)
C         BEDDIA=BED DIAMETER (M)
C         DENCH=CHARCOAL DENSITY(KG/CU M)
C        SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
C
C       CHAR USED (KG)
          CHRUSD=9.61E-6*SMR
```

```
C       CHAR BED WGT(KG)
        BEDWGT=BEDL*BEDDIA**2*.785*DENCH
        IF (CHRUSD.LT.0.8*BEDWGT) THEN
          EFF=EMAX
        ELSE
         EFF=EMAX*SIN((BEDWGT-CHRUSD)*1.57/(BEDWGT*0.2))
        ENDIF
C       PREVENTS NEGATIVE EFF FOR REACTION
C       IF (EFF.LT.0) EFF=0
        IF(EFF.GT.EMAX) EFF=EMAX
        RETURN
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

C       ****************************************************
C       * SUBROUTINE CICH    -  CALCULATES REMOVAL EFF     *
C       *   FOR FORMALDELYDE AND CI CHAR BED               *
C       ****************************************************
        SUBROUTINE CICH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR,BEDQ)
C       OUTPUTS
C         EFF=OUTPUT REMOVAL EFF (DEC)
C       INPUTS
C         EMAX=MAXIMUM BED REMOVAL EFF (DEC)
C         BEDL=BED LENGTH (M)
C         BEDDIA=BED DIAMETER (M)
C         DENCH=CHARCOAL DENSITY(KG/CU M)
C        SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
C         BEDQ=BED FLOW RATE (CU M/HR)
C
        BEDWGT=BEDL*BEDDIA**2*.785*DENCH
C       PERCENT OF BED WEIGHT CONSUMED (DEC)
        PBWGT=SMR/(BEDWGT*1E6)
        IF(PBWGT.LT..0012) THEN
          EFF=1-PBWGT*83.3
        ELSE
          EFF=.9*COS(PBWGT*1.57/.05)
        ENDIF
C       BED RESIDENCE TIME (SEC)
        BREST=BEDL*BEDDIA**2*3600/(BEDQ*1.273)
        IF(BREST.LT.0.25)THEN
          EFF=EFF*BREST/.25
        ENDIF
C       PREVENTS NEGATIVE EFF FOR REACTION
C       IF (EFF.LT.0) EFF=0
        IF(EFF.GT.EMAX) EFF=EMAX
        RETURN
        END
NUMBER OF WARNINGS IN PROGRAM UNIT: 0
```

```
NUMBER OF ERRORS    IN PROGRAM UNIT: 0
C         ******************************************************
C          * SUBROUTINE FQI    -    FINDS QI, THE CHARCOAL        *
C         * CAPACITY FOR A CONT  (CC LIQ CONT/GM CHAR)            *
C         ******************************************************
          SUBROUTINE FQI(AVAL,QI,FACID,SOL)
          INTEGER FACID
C         OUTPUTS
C           QI=CHARCOAL CAPACITY (CC LIQ CONT/GM CHAR)
C         INPUTS
C           AVAL= A VALUE OF CONTAMINANT
C          FACID= FLAG FOR ACID TREATED CHAR IN BED (Y=1 N=0)
C          SOL= CONTAMINANT HENRY'S LAW CONSTANT (ATM/MOL FRACTION)
C
          IF (AVAL.LT.0)  AVAL=0
C         A VALUE .GT. 10 AND .LT. 200
          IF ((AVAL .GT. 10) .AND. (AVAL.LT.200)) THEN
C          SOLUBLE CONTAMINANT - ACID OR NON-ACID TREATED CHAR
            IF (SOL.GT.0) THEN
              QI=2.2947*EXP(-0.26492*AVAL)
C           INSOLUBLE CONTAMINANT + NON ACID TREATED CHARCOAL
            ELSEIF (FACID.LE.0.1) THEN
              QI=2.80289*EXP(-0.28842*AVAL)
C           INSOLUBLE CONT + ACID TREATED CHARCOAL
            ELSE
              QI=3.8636*EXP(-0.32307*AVAL)
            ENDIF
C         A VALUE .LE. 10
          ELSEIF (AVAL.LE.10) THEN
            QI=.56-AVAL*.04
C         AVAL .GE. 200
          ELSE
            QI=1E-20
          ENDIF
          RETURN
          END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS    IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS    IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\ALIOH.F Options: /C 80 /L /BY     05/21/92 12:54:48

C        *****************************************************************
C        *               SUBROUTINE  ALIOH - AXIAL FLOW BED               *
C        *           CALCULATES REMOVAL EFFICIENCY FOR LIOH               *
C        *****************************************************************
C
         SUBROUTINE ALIOH(TN,TN1,EMAX,BEDL,DENLI,BEDDIA,RWUTLI,SWUTLI,
        +REMFCT,EFF)
C
C     INPUTS:
C      TN=FINAL INCREMENT TIME (HRS)
C      TN1=INITIAL INCREMENT TIME (HRS)
C      EMAX=MAXIMUM POSSIBLE REMOVAL EFFICIENCY (DEC)
C      BEDL=BED LENGTH (M)
C      DENLI=LIOH DENSITY (KG/CU M)
C      BEDDIA=BED DIAMETER (M)
C      RWUTLI=RATE OF LIOH USAGE FOR ALL CONTAMINANTS FROM LAST INCR(KG/HR)
C      SWUTLI=SUM OF WEIGHT OF LIOH UTILIZED FROM LAST INCR(KG)
C      REMFCT=LIOH REMOVAL FACTOR (LB LIOH/LB CONTAMINANT)
C     OUTPUTS:
C      EFF=REMOVAL EFFICIENCY (DEC)
C
C     IF CONT DOESNT REACT WITH LIOH OR BEDL<=0 OR BED DIA <=0 OR
C      DENLI <=0 THEN REM EFF =0
         IF((REMFCT.LE.0).OR.(BEDL.LE.0).OR.(BEDDIA.LE.0).OR.(DENLI.LE.0))
         + THEN
           EFF=0
         ELSE
C          BED WEIGHT (KG)
           BEDWGT=BEDL*(BEDDIA)**2*.785*DENLI
C          TOTAL WEIGHT OF LIOH UTILIZED AT AVERAGE TIME IN INCREMENT (KG)
           TWUTLI=SWUTLI+RWUTLI*(TN-TN1)/2
           IF (TWUTLI/BEDWGT.LE.0.8) THEN
             EFF=EMAX
           ELSE
             EFF=EMAX*SIN((BEDWGT-TWUTLI)*1.57/(BEDWGT*0.2))
           ENDIF
           IF (BEDL.LT.0.0191) THEN
             EFF=EFF*BEDL/0.0191
           ENDIF
         ENDIF
         IF(EFF.LT.0) EFF=0
         IF(EFF.GT.EMAX) EFF=EMAX
         RETURN
         END
C        ***********  END OF SUBROUTINE ALIOH ***************************

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CAFILL. Options: /C 80 /L /BY     05/21/92 12:54:53

C       ****************************************************************
C       *          SUBROUTINE CAFILL                                   *
C       *  SUBROUTINE TO FILL ADJUSTABLE SIZE CHAR ARRAY WITH BLANKS   *
C       ****************************************************************
        SUBROUTINE CAFILL(NN,NROW,NCOL)
        INTEGER NROW,NCOL
        CHARACTER NN(NROW,NCOL)*30

C     NN=ARRAY NAME-ARRAY HAS 30 CHARACTERS
C     NROW=NUMBER OF ROWS IN ARRAY (INTEGER)
C     NCOL=NUMBER OF COLUMNS IN ARRAY(INTEGER)

        DO 110 I=1,NROW
        DO 100 J=1,NCOL
         NN(I,J)=' '
  100    CONTINUE
  110    CONTINUE
        RETURN
C       ************* END OF SUBROUTINE CAFILL  **************************
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CALCM.F Options: /C 80 /L /BY 05/21/92 12:54:56

```
C       ****************************************************************
C       *                     SUBROUTINE    CALCM                      *
C       * SUBROUTINE TO CALCULATE SUM OF MASSES REMOVED BY ALL DEVICES *
C       * USES CAV CABIN, REM EFF DD(J,20), & M.GEN DD(J,19) TO CALC   *
C       * DEVICE CIN & COUT DD(J,17)& DD(J,18),M.REM CABIN+DEV DD(J,21)*
C       ****************************************************************

        SUBROUTINE CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
        INTEGER NROW,NCOL,LIN2
        REAL DD(NROW,NCOL)
C
C SUBROUTINES REQUIRED:
C  NONE

C INPUTS:
C  CAV=CABIN CONT AVERAGE CONCENTRATION (MG/CU M)
C  DD,NROW,NCOL=NAME AND SIZE OF MAT DD
C  TN=INCREMENT FINAL TIME (HRS)
C  LIN2=NO. OF DEVICES IN MAT DD
C  REM EFF AND DEVICE+CABIN M.GEN MUST BE LOADED INTO MAT DD
C    BEFORE USING THIS SUBROUTINE
C OUTPUTS:
C  M.REM (MG/HR) FOR ALL DEVICES + CABIN CALCULATED AT CAV,
C    ARE STORED IN MAT DD COL 21
C  SMREM=SUM OF MASS OF CONT REM IN DEVICES (MG/HR)
C  SMGEN=SUM OF MASS GENERATED BY ALL DEVICES + CABIN (MG/HR)
C
C  LOAD DEVICES 1 AND 2 WITH CIN AND COUT+M.REMOVED FOR DEV 2
C    DEVICE 1=CABIN
C       LOAD MAT DD WITH CAV CABIN
        DD(1,17)=CAV
        DD(1,18)=CAV
        DD(1,21)=0
C   DEVICE 2
        DD(2,17)=CAV
        DD(2,18)=CAV
        DD(2,21)=DD(2,2)*DD(2,18)
C   FOR DEV 3-15 CALC CIN COUT AND M.REMOVED BY DEVICE
C       IF DEVICE FLOW RATE=0 THEN SET CIN,COUT,& M.GEN=0
        DO 100 J=3,LIN2
        IF (DD(J,2).EQ.0) THEN
          DD(J,17)=0
          DD(J,18)=0
          DD(J,21)=0
          GOTO 100
        ENDIF
C  IF UPSTREAM DEVICE=1 OR 2 THEN SET INLET=CABIN CONC+DEV M.GEN/Q
```

```
          IF((DD(J,4).EQ.1).OR.(DD(J,4).EQ.2)) THEN
            DD(J,17)=DD(1,18)+DD(J,19)/DD(J,2)
          ELSE
C         DETERMINE FLOWS,CIN AND COUT FOR DEVICES WITH RELATIVE ADDRESSES
            IF (DD(J,4).EQ.0) THEN
              QNO1=0
              CNO1=0
            ELSE
              QNO1=DD(NINT(DD(J,4)),2)
              CNO1=DD(NINT(DD(J,4)),18)
            ENDIF
            IF (DD(J,5).EQ.0) THEN
              QNO2=0
              CNO2=0
            ELSE
              QNO2=DD(NINT(DD(J,5)),2)
              CNO2=DD(NINT(DD(J,5)),18)
            ENDIF
            IF (DD(J,6).EQ.0) THEN
              QNO3=0
              CNO3=0
            ELSE
              QNO3=DD(NINT(DD(J,6)),2)
              CNO3=DD(NINT(DD(J,6)),18)
            ENDIF
C
C           IF ALL UPSTREAM DEVICE FLOWS=0
            IF (QNO1+QNO2+QNO3.EQ.0) THEN
              DD(J,17)=0
              DD(J,18)=0
              DD(J,21)=0
              OPEN(2,FILE='CON',IOSTAT=IOVAL)
                WRITE(*,*)' FLOW HALTED-UPSTREAM DEV TURNED OFF-
     + INC END TIME;DEV=',TN,DD(J,1)
              CLOSE (2)
              GO TO 100
            ELSE
C             CALCULATE CIN
              DD(J,17)=(QNO1*CNO1+QNO2*CNO2+QNO3*CNO3)/
     +  (QNO1+QNO2+QNO3)+DD(J,19)/DD(J,2)
            ENDIF
C         END OF DETERMINE FLOWS,CIN,COUT OF DEV WITH REL ADDR.
          ENDIF
C       CALCULATE COUT
           DD(J,18)=DD(J,17)*(1-DD(J,20))
C          CALCULATE SUM OF MASS REMOVED (CIN*Q*REM EFF)
           DD(J,21)=DD(J,17)*DD(J,2)*DD(J,20)
  100 CONTINUE
C  END OF  LOADING OF MAT DD WITH DATA AND CALCULATING CIN COUT,M.REM
C  SUM TOTAL MASS OF CONT REMOVED BY ALL DEVICES (2-15) (MG/HR)
```

```
      SMREM=0
      DO 101 J=2,LIN2
      SMREM=SMREM+DD(J,21)
  101 CONTINUE
C  SUM MASS OF CONT GENERATED IN ALL DEVICES+CABIN (1-15)  (MG/HR)
      SMGEN=0
      DO 102 J=1,LIN2
      SMGEN=SMGEN+DD(J,19)
  102 CONTINUE
C  CALC M.REM CABIN AND PUT IN DD(1,21)
      DD(1,21)=SMGEN-SMREM
      RETURN
C      ************ END OF SUBROUTINE CALCM ***************************
      END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CATBNR. Options: /C 80 /L /BY     05/21/92 12:55:12

C       *******************************************************************
C       *                       SUBROUTINE  CATBNR                        *
C       *               CALCULATES EFFICIENCY OF CAT BURNER               *
C       *******************************************************************
C
        SUBROUTINE CATBNR(EMAX,OXID,EFF)
C
C  INPUTS:
C   EMAX=MAXIMUM BED EFFICIENCY (DEC)
C   OXID=DEGREE OF OXIDIZATION OF CHEMICAL (1=FULLY, 0=NONE)
C  OUTPUTS:
C   EFF=REMOVAL EFF (DEC)
C
        IF(OXID.LT.0) OXID=0
        IF(OXID.GT.1) OXID=1
        EFF=EMAX*OXID
        IF(EFF.LE.0) EFF=0
        IF(EFF.GT.EMAX) EFF=EMAX
        RETURN
        END
C       ***************** END OF SUBROUTINE CATBNR **********************

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CNRSUB. Options: /C 80 /L /BY   05/21/92 12:55:15

```
C       ******************************************************************
C       *                   SUBROUTINE   CNRSUB                          *
C       *       FOR 1 CONT AT A TIME AT CAV PRED, CALL EFF SUBROUTINES   *
C       *       FOR DEVICES AND PUT EFFICIENCY IN MAT DD COL 20          *
C       ******************************************************************

C       NOTE:DEVICE NUMBERS IN THIS SUBROUTINE REFER TO DEVICE TYPES,
C       NOT THEIR RELATIVE POSITION IN MAT DD

C       SUBROUTINE CNRSUB(I,TN,TN1,CAVPRD,DD,NROW,NCOL,CC,NROW1,NCOL1,
        SUBROUTINE CNRSUB(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
       +CDI,NROW2,NCOL2,LIN2)
        INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
        REAL DD(NROW,NCOL)
        REAL CC(NROW1,NCOL1)
        REAL CDI(NROW2,NCOL2)
C
C SUBROUTINES REQUIRED:
C  ACHBD-REMOVAL EFF OF AXIAL CHARCOAL BED
C  RCHBD-REMOVAL EFF OF RADIAL CHARCOAL BED
C  ALIOH-REMOVAL EFF OF AXIAL LIOH BED
C  COOXID-REMOVAL EFF OF CO OXIDIZER
C  CATBNR-REMOVAL EFF OF CAT BURNER
C  CONDHX-REMOVAL EFF OF CONDENSING HX
C
C INPUTS:
C  FROM PCSET PREDCT,AND CONVRG
C    I=CONT NO.
C    TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
C    CAVPRD=CABIN AVERAGE CONT CONC (MG/CU M)
C    DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
C    CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C    CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C    LIN2=NUMBER OF DEVICES IN MAT DD
C  FROM EFFICIENCY SUBROUTINES
C    EFF=REMOVAL EFF (DEC)
C OUTPUTS:
C  TO EFFICIENCY SUBROUTINES
C    DEVICE AND CONTAMINANT INFORMATION AS REQUIRED
C         DD(J,22)=DEVICE INLET CONCENTRATION
C  TO PCSET, PREDCT, & CONVRG
C    PUTS REMOVAL EFF FOR EACH DEVICE IN MAT DD COL 20
C
C  EFFICIENCY FOR DEVICES 1 AND 2
        DD(1,20)=0
        DD(2,20)=DD(2,8)

```

```
C   SET COUNTER FOR READING CHAR SMR IN MAT CC FOR DEVICE TYPE 3
          K=9
C
C   BEGIN LOOP FOR DEVICES 3 TO 15 OF MAT DD
          DO 100 J=3,LIN2
C         INDEX COUNTER
          K=K+3
C         SET REM EFF=0 AND GO TO END OF J LOOP IF DEVICE FLOW = 0
          IF(DD(J,2).LE.1E-10) THEN
            DD(J,20)=0
            GOTO 80
          ENDIF
C   DECISIONS FOR VARIOUS DEVICES
          IF (NINT(DD(J,3)).EQ.3) THEN
C           GO TO SUBROUTINE FOR DEVICE TYPE 3-AXIAL CHARCOAL BED
           CALL ACHBD(TN,TN1,DD(J,22),DD(1,10),DD(1,13),DD(J,2),DD(J,8),
     +     DD(J,9),DD(J,10),DD(J,12),DD(J,13),CDI(I,2),CDI(I,3),
     +     CDI(I,4),CDI(I,5),CDI(I,6),CC(I,K),EFF)
C           STORE EFF IN MAT DD
            DD(J,20)=EFF
          ELSEIF (NINT(DD(J,3)).EQ.4) THEN
C           GO TO SUBROUTINE FOR DEVICE TYPE 4-RADIAL CHARCOAL BED
           CALL RCHBD(TN,TN1,DD(J,22),DD(1,10),DD(1,13),DD(J,2),DD(J,8),
     +     DD(J,9),DD(J,10),DD(J,11),DD(J,12),DD(J,13),CDI(I,2),CDI(I,3),
     +     CDI(I,4),CDI(I,5),CDI(I,6),CC(I,K),EFF)
C           STORE EFF IN MAT DD
            DD(J,20)=EFF
          ELSEIF (NINT(DD(J,3)).EQ.5) THEN
C           GO TO SUBROUTINE FOR DEVICE TYPE 5-LIOH BED
            CALL ALIOH(TN,TN1,DD(J,8),DD(J,9),DD(J,10),DD(J,12),DD(J,15),
     +     DD(J,16),CDI(I,7),EFF)
C           STORE EFF IN MAT DD
            DD(J,20)=EFF
          ELSEIF (NINT(DD(J,3)).EQ.6) THEN
C           GO TO SUBROUTINE FOR DEVICE TYPE 6-CO OXIDIZER
            CALL COOXID(DD(J,2),DD(J,8),DD(J,9),DD(J,10),CDI(I,4),EFF)
C           STORE EFF IN MAT DD
            DD(J,20)=EFF
          ELSEIF (NINT(DD(J,3)).EQ.7) THEN
C           GO TO SUBROUTINE FOR DEVICE TYPE 7-CAT BURNER
            CALL CATBNR(DD(J,8),CDI(I,23),EFF)
C           STORE EFF IN MAT DD
            DD(J,20)=EFF
          ELSEIF (NINT(DD(J,3)).EQ.8) THEN
C           GO TO SUBROUTINE FOR DEVICE TYPE 8-CONDENSING HX
            CALL CONDHX(DD(J,2),DD(J,8),DD(J,9),CDI(I,4),CDI(I,5),CDI(I,6),
     +     DD(J,22),EFF)
C           STORE EFF IN MAT DD
            DD(J,20)=EFF
          ELSEIF (NINT(DD(J,3)).EQ.9) THEN
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CONDHX. Options: /C 80 /L /BY     05/21/92 12:55:36

C        ****************************************************************
C        *                     SUBROUTINE  CONDHX                        *
C        *              CALCULATES EFFICIENCY OF CONDENSING HX           *
C        ****************************************************************
C
         SUBROUTINE CONDHX(BEDQ,EMAX,MLIQ,MW,VCONC,SOL,CAVPRD,EFF)
         REAL MW,MLIQ,H,PA,XA,NOUT
C
C  INPUTS:
C    BEDQ=BED FLOW RATE (M3/HR)
C    EMAX=MAXIMUM POSSIBLE REMOVAL EFFICIENCY (DEC)
C    MLIQ=WATER FLOW RATE IN HX DUE TO CONDENSING (KG/HR)
C    MW=MOLECULAR WEIGHT OF CONTAMINANT
C    VCONT=VAPOR CONCENTRATION OF CONTAMINANT (MG/M3)
C    SOL=HENRY'S LAW COEFFICIENT (ATM/MOL FRACTION)
C OUTPUTS:
C    EFF=REMOVAL EFF (DEC)
C
         CAIN=CAVPRD
         IF (CAIN.LE.1E-10) THEN
           CAIN=0.1E-10
         ENDIF
         IF (SOL.LE.1E-10) THEN
           EFF=0
         ELSE
C    IF CONTAMINANT IS AMMONIA - USES EXPERIMENTAL DATA FROM JSC-08797
C     FOR AMMONIA REMOVAL AS A FUNCTION OF CO2 CONCENTRATION (8/23/76)
           IF (MW.EQ.17.0) THEN
             CAOUT=((CAIN*BEDQ)-(MLIQ*189.5847418*CAIN**0.534915256))/BEDQ
             EFF=((CAIN-CAOUT)/CAIN)*EMAX
           ELSE
C    CONTAMINANT IS NOT AMMONIA
C    CALCULATE CONTAMINANT PARTIAL PRESSURE AND WATER MOLE FRACTION
             PA=CAIN*1.0E-9*82.06*278/MW
             XA=(PA/1)/((MLIQ*(1000/18))/(BEDQ*(1000/22.4))+SOL/1)
             NOUT=MLIQ*XA*1000/18
             CAOUT=((CAIN*BEDQ)-(NOUT*MW*1000))/BEDQ
             EFF=((CAIN-CAOUT)/CAIN)*EMAX
           ENDIF
         ENDIF
         IF(EFF.LE.0) EFF=0
         IF(EFF.GT.EMAX) EFF=EMAX
         RETURN
         END
C        ***************** END OF SUBROUTINE CONDHX **********************

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CONVRG. Options: /C 80 /L /BY   05/21/92 12:55:41

```
C        *****************************************************************
C        *                    SUBROUTINE CONVRG                          *
C        *            MAIN CONVERGENCE LOOP SUBROUTINE                   *
C        *  USING CAV PRED & BASED ON SUM MASS REM OF LAST INCR, CALC    *
C        *  NEW REM EFF, CAV CALC, CEQULIB, CFINAL, & M. REMOVED         *
C        *  PUT THEM IN MAT DD - WORKS FOR ONE CONT AT A TIME            *
C        *****************************************************************

         SUBROUTINE CONVRG(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
        +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,KK,LIN,LIN2,
        +NN,PRTSW3,IMSGDN)
         INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,PRTSW3,KK
         CHARACTER NN(NROW1)*30
         REAL DD(NROW,NCOL)
         REAL CC(NROW1,NCOL1)
         REAL CDI(NROW2,NCOL2)
         REAL X1,X2,Y1,Y2,SLOPE

C SUBROUTINES REQUIRED:
C  PRAFIL-ZERO MAT DD COL 17-21
C  CNRSUB-USING CAV PRED CALC REM EFF FOR ALL DEVICES-PUT IN DD COL 20
C  MASBAL-CALC CAV CALC,CFINAL,CEQULIB,M.REM
C INPUTS:
C  FROM MCALC
C    I=CONTAMINANT NO.
C    TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
C    DD,NROW,NCOL=NAME & DIM OF MAT DD
C    CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C    CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C    LIN=NO. OF CONTAMINANTS IN MAT CDI
C    LIN2=NO. DEVICES IN MAT DD
C    IMSGDN=DEVICE NO. FOR MESSAGE OUTPUT
C    CAVPRD=PREDICTED CABIN CONC FOR INCREMENT (MG/CU M)
C  FROM PRAFIL
C    ZEROS IN MAT DD COL 17-21
C  FROM CNRSUB
C    CALL REM EFF SUBROUTINE & PUTS REM EFF FOR EACH DEV IN DD COL 20
C  FROM MASBAL
C    CAVCLC=CALC CABIN CONT CONC (MG/CU M)
C    CFINAL=FINAL INCR CABIN CONT CONC(MG/CUM)
C    CEQLIB=EQUILIBRIUM CABIN CONT CONC(MG/CU M)
C OUTPUTS:
C  TO MCALC
C    CAVCLC=CALC CABIN CONT CONC (MG/CU M)
C    CFINAL=FINAL INCR CABIN CONT CONC(MG/CUM)
C    KK=COUNTER FOR CONVERGENCE
C    CEQLIB=EQUILIBRIUM CABIN CONT CONC(MG/CU M)
```

```
C    M.REM IS IN MAT DD COL 21
C  TO PRAFIL
C    NAME & SIZE OF MATRIX + FIRST AND LAST COL TO BE ZEROED
C  TO CNRSUB
C    I=CONT NO.
C    TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
C    DEVICE AVERAGE CONT CONC (MG/M3) = DD(J,22)
C    DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
C    CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C    CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C    LIN2=NUMBER OF DEVICES IN MAT DD
C  TO MASBAL
C    I
C    TN,TN1
C    CVOL=CABIN VOL (CU M)=DD(1,9)
C    CINIT=INCR INIT CABIN CONT CONC (MG/CU M)=CC(I,1)
C
C    CONVERGENCE ERROR (DEC)
       CNVERR=DD(1,12)
       IF(CNVERR.LT.1E-10) THEN
          OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
             WRITE(IMSGDN,*) 'CONV ERROR<1E-10:PROGRAM TERMINATED'
          CLOSE(IMSGDN)
          STOP
       ENDIF
       KK=1
       DO 100 KK=1,20
C         ZERO MAT DD COL 17-21
             CALL PRAFIL(DD,NROW,NCOL,17,21)
C         USING CAVPRD FIND REM EFF OF EACH DEV & PUT IN DD COL 20
          CALL CNRSUB(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +          CDI,NROW2,NCOL2,LIN2)
C         FIND CAVCLC FOR THESE REMOVAL EFFICIENCIES
             CALL MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +    CAVCLC,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
C         IF PRTSW3=1 THEN PRINT NAME & NO + CONV VALUES
             IF (PRTSW3.EQ.1)  THEN
                OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
                   WRITE(IMSGDN,*)'PRINTOUT FOR CONVERGENCE VALUES IN CONVRG'
                   WRITE(IMSGDN,50) I,NN(I)
  050              FORMAT(1X,'CONT NO.= ',I4,2X,A)
                   WRITE(IMSGDN,*)'CAVPRD,CAVCLC= ',CAVPRD,CAVCLC
                CLOSE(IMSGDN)
             ENDIF
C
C         IF CAVCLC=CAVPRD THEN EXIT THE KK LOOP
             IF(CAVCLC.EQ.CAVPRD) GOTO 101
C         IF CAVPRD<1E-10 THEN SKIP CONVERGENCE STEP
             IF(CAVPRD.LT.1E-10) GOTO 80
C         IF CONVERGENCE IS REACHED EXIT THE KK LOOP
```

```
                IF(ABS((CAVPRD-CAVCLC)/CAVPRD).LT.CNVERR) THEN
                  GOTO 101
                ENDIF
C         CONVERGENCE CALCULATION ROUTINE
C         USE THE BISECTION METHOD FOR THE ITERATION WHERE KK=1
  080      IF (KK.EQ.1) THEN
C         INITIALIZE X2 AND Y2 FOR THE NEXT ITERATION
                X2=CAVPRD
                Y2=CAVCLC-CAVPRD
                CAVPRD=(CAVPRD+CAVCLC)/2
              ELSE
C         USE THE NEWTON-RAPHSON METHOD FOR ITERATIONS WHERE KK>1
                X1=X2
                Y1=Y2
                X2=CAVPRD
                Y2=CAVCLC-CAVPRD
                SLOPE=(Y2-Y1)/(X2-X1)
                CAVPRD=X2-0.95*Y2/SLOPE
              ENDIF
C
C         SET CAV IN PRED DD(I,22) = CAV IN CALC DD(I,17)
          DO 90 J=1,LIN2
            DD(J,22)=DD(J,17)
  090 CONTINUE
C
  100 CONTINUE
C         END OF KK LOOP
  101 CONTINUE
C

          RETURN
C        ************ END OF SUBROUTINE CONVRG   ***************************
          END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\COOXID. Options: /C 80 /L /BY     05/21/92 12:55:52

C       *****************************************************************
C       *                    SUBROUTINE  COOXID                          *
C       *   CALCULATES EFFICIENCY OF CO OXIDIZER (Pt on charcoal)        *
C       *****************************************************************
C
        SUBROUTINE COOXID(BEDQ,EMAX,BEDL,BEDDIA,MW,EFF)
        REAL MW
C
C  INPUTS:
C    BEDQ=BED FLOW RATE (CU M/HR)
C    EMAX=MAXIMUM POSSIBLE REMOVAL EFFICIENCY (DEC)
C    BEDL=BED LENGTH (M)
C    BEDDIA=BED DIAMETER (M)
C    MW=MOLECULAR WEIGHT OF CONTAMINANT
C  OUTPUTS:
C    EFF=REMOVAL EFF (DEC)
C
C       WORKS ONLY FOR C0 MW=28.01 OR H2=2.02; OTHERWISE REM EFF=0
        IF ((MW.EQ.28.01).OR.(MW.EQ.2.02)) THEN
          EFF=EMAX
C       IF RESIDENCE TIME <0.2 SEC THEN REM EFF DROPS LINEARLY
C       BREST = BED RESIDENCE TIME (SEC)
          BREST=(3.141592654/4)*BEDL*BEDDIA**2*3600/BEDQ
          IF (BREST.LT.0.2) THEN
            EFF=EMAX*BREST/0.2
          ENDIF
        ELSE
C         REM EFF FOR OTHER THAN CO OR H2 = ZERO
          EFF=0
        ENDIF
        IF(EFF.LE.0) EFF=0
        IF(EFF.GT.EMAX) EFF=EMAX
        RETURN
        END
C       ***************** END OF SUBROUTINE COOXID **********************

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CRIN.FO Options: /C 80 /L /BY     05/21/92 12:55:55

C        ********************************************************************
C        *            SUBROUTINE   CRIN                                      *
C        *       SUBROUTINE TO READ STRING OF LENGTH 30 INTO MAT NN          *
C        *        AND READ REAL DATA INTO MAT XX(ROW,COL)                    *
C        *       RETURNS NUMBER OF LINES OF DATA READ FROM FILE              *
C        ********************************************************************
C     NOTE: INPUT STRING MUST HAVE SINGLE QUOTES AROUND IT
C     NOTE: INPUT NUMBERS MUST BE SEPARATED BY BLANKS
        SUBROUTINE CRIN(NN,XX,NROW,NCOL,LIN)
        INTEGER NROW,NCOL,IOVAL,LIN
        CHARACTER NN(NROW)*30,FNAME*24
        REAL XX(NROW,NCOL)
    010 READ(*,'(A)') FNAME
        OPEN(1,FILE=FNAME,STATUS='OLD',IOSTAT=IOVAL)
        IF(IOVAL.NE.0) GOTO 900
        LIN=0
        DO 100 I=1,NROW
        READ(1,*,IOSTAT=IOVAL,END=500,ERR=900 ) NN(I),(XX(I,J),J=1,NCOL)
        LIN=LIN+1
    100 CONTINUE
    500 WRITE(*,'(A)') ' DONE WITH FILE INPUT'
        WRITE (*,*) ' '
        CLOSE (1)
        GOTO 990
    900 WRITE(*,*)'IOERROR= ',IOVAL
        CLOSE (1)
        WRITE(*,*) 'WHAT IS THE INPUT FILE NAME? '
        GOTO 10
    990 RETURN
C       *************** END OF SUBROUTINE CRIN  ************************
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CROUT.F Options: /C 80 /L /BY     05/21/92 12:56:01

C       ********************************************************************
C       *         SUBROUTINE  CROUT                                        *
C       *    SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER               *
C       *  WRITES STRG OF LENGTH 30 FROM MAT NN & REAL DATA FROM MAT       *
C       *  XX(ROW,COL) STARTING WITH COL FSTCOL, AND ENDING WITH LSTCOL    *
C       *   AND FROM LINE FSTLIN TO LINE LSTLIN                            *
C       ********************************************************************
        SUBROUTINE CROUT(NN,XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN,
        +IMSGDN,NINC,FNAME,IDEVNO,IOVAL)
        INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN,NINC,IDEVNO,
        +  IOVAL
        CHARACTER FNAME*24,DES*1
        CHARACTER NN(NROW)*30
        REAL XX(NROW,NCOL)
        IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
        IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
        IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
        IF (FSTLIN.GT.LIN) FSTLIN=LIN
        IF (LSTLIN.GT.LIN) LSTLIN=LIN
        IF (FSTLIN.GT.LSTLIN) FSTLIN=LSTLIN

C 010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
C       WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END
C       CLOSE (IMSGDN)
C       READ(*,'(A)') FNAME
C       QUIT IF FNAME=END
C       IF (FNAME.EQ.'END') GO TO 990
C       IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
C       OPEN(1,FILE=FNAME, IOSTAT=IOVAL)
        IF(IOVAL.NE.0) GOTO 900
          WRITE (IDEVNO,55,IOSTAT=IOVAL,ERR=900) NINC
  055      FORMAT ('INCREMENT NO. = ',I7)
        DO 110 I=FSTLIN,LSTLIN
C         WRITE(1,60,IOSTAT=IOVAL,ERR=900) I,NN(I)
          WRITE(IDEVNO,60,IOSTAT=IOVAL,ERR=900) I,NN(I)
  060    FORMAT(1X,'CONT NO.=  ',I4,2X,A)
C         WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
          WRITE(IDEVNO,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
  070    FORMAT(1X,7G11.4)
  110 CONTINUE
C       CLOSE (1)
        GOTO 990
  900 OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
        WRITE(IMSGDN,*)'IOERROR= ',IOVAL
        CLOSE(IMSGDN)
C       CLOSE (1)
        CLOSE (IDEVNO)
```

```
C       GOTO 10
  990 RETURN
C       ********** END OF SUBROUTINE CROUT  ******************************
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CROUT2. Options: /C 80 /L /BY     05/21/92 12:56:09

C        ****************************************************************
C        *          SUBROUTINE  CROUT2                                  *
C        *       SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER        *
C        *   WRITES STRG OF LENGTH 30 FROM MAT NN & REAL DATA FROM MAT  *
C        *   XX(ROW,COL) STARTING WITH COL FSTCOL, AND ENDING WITH LSTCOL *
C        *    AND FROM LINE FSTLIN TO LINE LSTLIN                       *
C        ****************************************************************
         SUBROUTINE CROUT2(NN,XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN,
         +IMSGDN)
         INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN
         CHARACTER FNAME*24,DES*1
         CHARACTER NN(NROW)*30
         REAL XX(NROW,NCOL)
         IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
         IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
         IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
         IF (FSTLIN.GT.LIN) FSTLIN=LIN
         IF (LSTLIN.GT.LIN) LSTLIN=LIN
         IF (FSTLIN.GT.LSTLIN) FSTLIN=LSTLIN

     010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
         WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END  '
         CLOSE (IMSGDN)
         READ(*,'(A)') FNAME
         QUIT IF FNAME=END
         IF (FNAME.EQ.'END') GO TO 990
         IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
         OPEN(1,FILE=FNAME, IOSTAT=IOVAL)
         IF(IOVAL.NE.0) GOTO 900
         DO 110 I=FSTLIN,LSTLIN
           WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
     070   FORMAT(1X,7G11.4)
     110 CONTINUE
         CLOSE (1)
         GOTO 990
     900 OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
         WRITE(IMSGDN,*)'IOERROR= ',IOVAL
         CLOSE(IMSGDN)
         CLOSE (1)
         GOTO 10
     990 RETURN
C        ********** END OF SUBROUTINE CROUT  ******************************
         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\DATOUT. Options: /C 80 /L /BY   05/21/92 12:56:16

```
C        ****************************************************************
C        *          SUBROUTINE DATOUT                                   *
C        * SUBROUTINE TO PRINT HEADINGS AND DATA TO PRINTER,CON,OR FILE *
C        ****************************************************************
C        NOTES:(1)FILE MUST BE OPEN BEFORE CALLING THIS SUBROUTINE
C              (2) IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED

         SUBROUTINE DATOUT(TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
     .  +CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
        +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
        +TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)

         INTEGER PRTSW6,PRTSW8,PRTSW9,TVAL,NINC

C    SUBROUTINES REQUIRED:
C        PRCDA=PRINT OUT OF CONTAMINANT DATA
C        PRREM1=PRINTOUT OF RATE OF CONTAMINANT REMOVAL (MG/HR)-SHEET1
C        PRREM2=PRINTOUT OF RATE OF CONTAMINANT REMOVAL (MG/HR)-SHEET2
C        PRMAS1=PRINTOUT OF SUM MASSES REMOVED BY DEVICES (MG)-SHEET1
C        PRMAS2=PRINTOUT OF SUM MASSES REMOVED BY DEVICES (MG)-SHEET2
C        PREFF=PRINTOUT OF INCREMENT END REMOVAL EFFICIENCIES

C    INPUTS FROM MAIN PROGRAM:
C        TN=INCREMENT FINAL TIME (HRS)
C        TN1=INCREMENT INITIAL TIME (HRS)
C        LIN=NO. OF CONT IN MAT CC A D NN
C        DD,NROW,NCOL=NAME & SIZE OF MAT DD
C        CC,NROW1,NCOL1=NAME & SIZE  F MAT
CC
C        CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
C        LIN2=NO. DEVICES IN MAT DD
C        NN=NAME OF MAT NN
C        NINC=TIME INCREMENT NUMBER
C           =0 THEN PRINT HEADINGS & DATA FOR PRECALCULATION SET UP ROUTINE
C           =-1 THEN PRINT HEADINGS & DATA FOR FINAL ANSWERS
C           ELSE PRINT WITH PROPER INCREMENT NUMBER
C        IDEVNO=OUTPUT DEVICE NUMBER (SHOULD BE 6)
C        IMONTH..IHOUR=DATE AND TIME VARIABLES
C        IPGCTR=PAGE COUNTER FOR SEQUENTIAL PAGE NO.'S ON ALL PAGES

C    OUTPUT TO MAIN PROG:
C        IOVAL=STATUS OF IOERROR IN SUBROUTINES

*   TEST CASE **********
*        LIN=120

C        PRINT OUT CONCENTRATION DATA
         CALL PRCDA(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
```

```
     +  IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
     +  PRTSW8,PRTSW9,FCPLOT,IDEVN1)
C      PRINT OUT NHB 8060.1 GROUP CONTRIBUTION VALUES (T-VALUES)
       IF ((TVAL.EQ.1).OR.(TVAL.EQ.2)) THEN
       CALL GROUP(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
     +  IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
     +  IPGCTR,TVAL,IDEVN3,PRTSW8)
       ENDIF
       IF (PRTSW6.EQ.1) THEN
        IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
C         PRINTOUT OF RATE OF CONTAMINANT REMOVAL BY DEVICES-SHEET1
          CALL PRREM1(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
     +  IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
C         DON'T PRINT SHEET 2 UNLESS NUMBER OF DEVICES IN MAT DD > 8
          IF (LIN2.GT.8) THEN
C          PRINTOUT OF RATE OF CONTAMINANT REMOVAL BY DEVICES-SHEET2
           CALL PRREM2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
     +   IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
          ENDIF
C         PRINTOUT OF SUM OF MASS REMOVED BY DEVICES-SHEET1
          CALL PRMAS1(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
     +  IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
C         DON'T PRINT SHEET 2 UNLESS NUMBER OF DEVICES IN MAT DD > 8
          IF(LIN2.GT.8) THEN
C          PRINTOUT OF SUM OF MASS REMOVED BY DEVICES-SHEET2
           CALL PRMAS2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
     +   IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
          ENDIF
        ENDIF
C         PRINTOUT OF INCREMENT END REMOVAL EFFICIENCIES
          CALL PREFF(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
     +  IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
     +  IPGCTR,PRTSW8,PRTSW9,IDEVN2)
       ENDIF

       RETURN
       END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\GROUP.F Options: /C 80 /L /BY   05/21/92 12:56:54

```
C   FILE GROUP
C       ****************************************************************
C       *       SUBROUTINE GROUP                                       *
C       * PROGRAM TO PRINT THE GROUP TOXICITY LEVELS AND T LEVEL       *
C       ****************************************************************
C
C
        SUBROUTINE GROUP(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
       +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
       +IPGCTR,TVAL,IDEVNT,PRTSW8)
C
C
        DIMENSION GL(16)
        REAL CC(NROW1,NCOL1)
        REAL CDI(NROW2,NCOL2)
        REAL TLEVL
        INTEGER TVAL,PRTSW8,NINC,IDEVNO,IDEVNT
C       DETERMINE THE SUMS FOR EACH GROUP LEVEL
        DO 25 J=1,16
   25  GL(J)=0.
        DO 30 I=1,LIN
          FRACT = CC(I,4)/CDI(I,9)
          TSTR = CDI(I,8)
          NHB = IFIX(TSTR)
          GL(NHB) = GL(NHB) + FRACT
   30  CONTINUE
C       CALCULATE THE TLEVEL OF THE ASSOCIATED GROUP LEVELS
        TLEVL = GL(1)+GL(2)+GL(3)+GL(4)+GL(5)+GL(9)+GL(10)+GL(11)+
       +       GL(13)+GL(14)+GL(16)
        IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
        WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' '
        WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' '
        WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900)
       +' GROUP T-VALUES AS SPECIFIED IN NHB 8060.1B APPENDIX D'
        WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' '
        WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900)
       +'  -01-   -02-   -03-   -04-   -05-   -06-   -07-   -08-   -09-
       +-10-   -11-   -12-   -13-   -14-   -15-   -16-'
        WRITE (IDEVNO,'(/1X,16(F6.2,1X)/)',IOSTAT=IOVAL,ERR=900) GL(1),
       +  GL(2),GL(3),GL(4),GL(5),GL(6),GL(7),GL(8),GL(9),GL(10),
       +  GL(11),GL(12),GL(13),GL(14),GL(15),GL(16)
        WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' OVERALL T-VALUE'
        OALLT = GL(1)+GL(2)+GL(3)+GL(4)+GL(5)+GL(9)+GL(10)+GL(11)+
       +       GL(13)+GL(14)+GL(16)
        WRITE (IDEVNO,'(T4,F7.2)',IOSTAT=IOVAL,ERR=900) OALLT
        ENDIF
        IF (NINC.NE.-1) THEN
```

```
C      ************** WRITE T-VALUE DATA TO A PLOT FILE ***************
         IF (TVAL.EQ.2) THEN
           WRITE (IDEVNT,50,IOSTAT=IOVAL,ERR=900) TN1,TN,TLEVL
   50       FORMAT (T2,2(F8.2,1X),F7.2)
         ENDIF
       ENDIF
       GOTO 999
C      ********************** END OF SUBROUTINE **********************
  900 WRITE(*,*)'IO ERROR IN GROUP= ',IOVAL
  999 RETURN
       END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\HEADGS. Options: /C 80 /L /BY     05/21/92 12:56:28

C  FILE:HEADGS.FOR
C        ******************************************************************
C        *          SUBROUTINE HDG1                                       *
C        * PROGRAM TO PRINT HEADING-DATE, TIME, FILE NAME, & PAGE NO.    *
C        ******************************************************************

C        NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

         SUBROUTINE HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,
        +FNAME,IPGNO,IDEVNO)
C         IMONTH...ISECOND=TIME AND DATE NAMES
C         FNAME=FILE NAME
C         IPGNO=PAGE NUMBER
C         IDEVNO=DEVICE NUMBER FOR OUTPUT

         CHARACTER FNAME*24


C        WRITE HEADING
         IF (IPGNO .EQ. 1) THEN
         WRITE (IDEVNO,5,IOSTAT=IOVAL,ERR=900)
   005     FORMAT (2X,'PROGRAM VERSION 8.0 Alpha',5X,'04-10-92',/)
         ENDIF
         WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) IMONTH,IDAY,IYEAR,
        +IHOUR,IMINUTE,FNAME,IPGNO
   010 FORMAT(2X,I2,'/',I2,'/',I4,5X,I2,':',I2,5X,A,2X,'PAGE ',I4)

         GO TO 999
   900 WRITE(*,*)'IO ERROR IN HDG1= ',IOVAL
   999 RETURN
         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
```

```
C        ****************************************************************
C        *        SUBROUTINE DATTM                                      *
C        *  SUBROUTINE TO READ SYSTEM DATE AND TIME FOR IBM PC OR AT    *
C        ****************************************************************

         SUBROUTINE DATTM(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,ISECOND)

C        REQUIRED FOR IBM PROF FORTRAN
         INTEGER*2  IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,ISECOND,IHUNSEC

         CALL GETDAT(IYEAR,IMONTH,IDAY)
         CALL GETTIM(IHOUR,IMINUTE,ISECOND,IHUNSEC)

         RETURN
         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

C        ****************************************************************
C        *        SUBROUTINE HDG2                                       *
C        * PROGRAM TO PRINT HEADING-TIME INCR+INCR INIT AND FINAL TIME  *
C        * PRINTS TIME INCREMENT NUMBER                                 *
C        ****************************************************************

C        NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

         SUBROUTINE HDG2(INCRNO,TN1,TN,IDEVNO)

C     INPUTS:
C        INCRNO=TIME INCREMENT NUMBER
C        TN1=INCREMENT INITIAL TIME (HRS)
C        TN=INCREMENT FINAL TIME (HRS)
C        IDEVNO=DEVICE NUMBER FOR OUTPUT

C        WRITE HEADING
         WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)INCRNO,TN1,TN
   010 FORMAT(1X,'TIME INCR ',I5,2X,'INITIAL TIME (HRS)= ',F8.2,2X,
     +'FINAL TIME (HRS)= ',F8.2)

         GO TO 999
   900 WRITE(*,*)'IO ERROR IN HDG2= ',IOVAL
   999 RETURN
         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
```

```
C       ***************************************************************
C       *        SUBROUTINE HDG3                                      *
C       * PROGRAM TO PRINT HEADING-TIME INCR+INCR INIT AND FINAL TIME *
C       * PRINTS PCALC OR FINAL INSTEAD OF TIME INCREMENT NUMBER      *
C       ***************************************************************

C       NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

        SUBROUTINE HDG3(IFLAG,TN1,TN,IDEVNO)

C    INPUTS:
C       IFLAG=FLAG FOR TIME INCREMENT (1=PCALC, 2=FINAL)
C       TN1=INCREMENT INITIAL TIME (HRS)
C       TN=INCREMENT FINAL TIME (HRS)
C       IDEVNO=DEVICE NUMBER FOR OUTPUT

        CHARACTER INAME*5

        IF(IFLAG.EQ.1) THEN
          INAME='PCALC'
        ELSEIF(IFLAG.EQ.2) THEN
          INAME='FINAL'
        ELSE
          INAME='ERROR'
        ENDIF

C       WRITE HEADING
        WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)INAME,TN1,TN
  010 FORMAT(1X,'TIME INCR ',A,2X,'INITIAL TIME (HRS)= ',F8.2,2X,
     +'FINAL TIME (HRS)= ',F8.2)

        GO TO 999
  900 WRITE(*,*)'IO ERROR IN HDG3= ',IOVAL
  999 RETURN
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
```

```
C        ******************************************************************
C        *          SUBROUTINE HDG4                                       *
C        * PROGRAM TO PRINT HEADING-CONT NO., NAME, FINAL CABIN CONC      *
C        * MAC, EXCEEDS MAC                                               *
C        ******************************************************************

C        NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

         SUBROUTINE HDG4(IDEVNO)

C    INPUTS:
C        IDEVNO=DEVICE NUMBER FOR OUTPUT

C        WRITE HEADING
         WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
 010 FORMAT(1X,'CONT',14X,'NAME',14X,'FINAL CABIN',5X,'MAC',5X,
     +'EXCEEDS')
         WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
  020 FORMAT(1X,' NO.',32X,'CONC (MG/M3)',12X,'  MAC  ')

         GO TO 999
  900 WRITE(*,*)'IO ERROR IN HDG4= ',IOVAL
  999 RETURN
         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

C        ******************************************************************
C         *          SUBROUTINE HDG5                                      *
C        * PROGRAM TO PRINT HEADING-TOTAL CONT REMOVED BY EACH DEV (MG) *
C        * PRINTS SHEET 1-NO,NAME,CABIN,LEAK,&DEV3..DEV8                  *
C        ******************************************************************

C        NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

         SUBROUTINE HDG5(IDEVNO)

C    INPUTS:
C        IDEVNO=DEVICE NUMBER FOR OUTPUT

C        WRITE HEADING
         WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
 010 FORMAT(1X,24X,'TOTAL CONTAMINANT MASS REMOVED BY EACH DEVICE (MG)
     +       SHEET 1')
         WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
 020 FORMAT(1X,' NO.',14X,'NAME',16X,'CABIN',9X,'LEAK',8X,'DEV3',
     +8X,'DEV4',8X,'DEV5',8X,'DEV6',8X,'DEV7',8X,'DEV8')
```

```

      GO TO 999
  900 WRITE(*,*)'IO ERROR IN HDG5= ',IOVAL
  999 RETURN
      END
```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```

C     ****************************************************************
C     *        SUBROUTINE HDG6                                        *
C     * PROGRAM TO PRINT HEADING-TOTAL CONT REMOVED BY EACH DEV (MG) *
C     * PRINTS SHEET 2-NO,NAME,& DEV9..DEV15                          *
C     ****************************************************************

C     NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

      SUBROUTINE HDG6(IDEVNO)

C    INPUTS:
C     IDEVNO=DEVICE NUMBER FOR OUTPUT

C     WRITE HEADING
      WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
  010 FORMAT(1X,24X,'TOTAL CONTAMINANT MASS REMOVED BY EACH DEVICE (MG)
     +     SHEET 2')
      WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
  020 FORMAT(1X,' NO.',14X,'NAME',16X,' DEV9',8X,'DEV10',7X,'DEV11',
     +7X,'DEV12',7X,'DEV13',7X,'DEV14',7X,'DEV15')

      GO TO 999
  900 WRITE(*,*)'IO ERROR IN HDG6= ',IOVAL
  999 RETURN
      END
```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```

C     ****************************************************************
C     *        SUBROUTINE HDG7                                        *
C     * PROGRAM TO PRINT HEADING-DEVICE REM EFF AT END OF TIME INCR  *
C     * PRINTS NO.,NAME,#2..#12                                       *
C     ****************************************************************

C     NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

      SUBROUTINE HDG7(IDEVNO)

```

```
C    INPUTS:
C        IDEVNO=DEVICE NUMBER FOR OUTPUT

C        WRITE HEADING
         WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
     010 FORMAT(1X,24X,'DEVICE REMOVAL EFFICIENCY AT END OF TIME INCREMENT
        + (DEC)')
         WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
      020 FORMAT(1X,' NO.',14X,'NAME',15X,'#2',4X,'#3',
        +4X,'#4',4X,'#5',4X,'#6',4X,'#7',4X,'#8',4X,'#9',3X,'#10',
        +3X,'#11',3X,'#12',3X,'#13',3X,'#14',3X,'#15')

         GO TO 999
      900 WRITE(*,*)'IO ERROR IN HDG7= ',IOVAL
      999 RETURN
         END
```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```

C        ****************************************************************
C        *        SUBROUTINE HDG8                                       *
C        * PROGRAM TO PRINT HEADING-RATE OF CONT REMOVAL-EACH DEV .(MG) *
C        * PRINTS SHEET 1-NO,NAME,CABIN,LEAK,&DEV3..DEV8                *
C        ****************************************************************

C        NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

         SUBROUTINE HDG8(IDEVNO)

C    INPUTS:
C        IDEVNO=DEVICE NUMBER FOR OUTPUT

C        WRITE HEADING
         WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
      010 FORMAT(1X,24X,'RATE OF CONTAMINANT REMOVAL-EACH DEVICE (MG/HR)
        +        SHEET 1')
         WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
      020 FORMAT(1X,' NO.',14X,'NAME',16X,'CABIN',9X,'LEAK',8X,'DEV3',
         +8X,'DEV4',8X,'DEV5',8X,'DEV6',8X,'DEV7',8X,'DEV8')

         GO TO 999
      900 WRITE(*,*)'IO ERROR IN HDG8= ',IOVAL
      999 RETURN
         END
```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```
C       *****************************************************************
C       *        SUBROUTINE HDG9                                        *
C       * PROGRAM TO PRINT HEADING-RATE OF CONT REMOVAL-EACH DEV(MG/HR)*
C       * PRINTS SHEET 2-NO,NAME,& DEV9..DEV15                          *
C       *****************************************************************

C       NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

        SUBROUTINE HDG9(IDEVNO)

C    INPUTS:
C       IDEVNO=DEVICE NUMBER FOR OUTPUT

C       WRITE HEADING
        WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
    010 FORMAT(1X,24X,'RATE OF CONTAMINANT REMOVAL-EACH DEVICE (MG/HR)
     +        SHEET 2')
        WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
    020 FORMAT(1X,' NO.',14X,'NAME',16X,' DEV9',8X,'DEV10',7X,'DEV11',
     +7X,'DEV12',7X,'DEV13',7X,'DEV14',7X,'DEV15')

        GO TO 999
    900 WRITE(*,*)'IO ERROR IN HDG9= ',IOVAL
    999 RETURN
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\LDIGEN. Options: /C 80 /L /BY     05/21/92 12:56:50

C        *****************************************************************
C        *          SUBROUTINE LDIGEN                                    *
C        *   SUBROUTINE TO LOAD INTERNAL GENERATION FROM MAT CDI COL 1 & *
C        *   COL 10-22 INTO MAT  DD COL 19                               *
C        *****************************************************************
         SUBROUTINE LDIGEN(I,DD,NROW,NCOL,CDI,NROW2,NCOL2,LIN2)
         INTEGER NROW,NCOL,NROW2,NCOL2,LIN2
         REAL DD(NROW,NCOL)
         REAL CDI(NROW2,NCOL2)

C  INPUTS:
C       I=CONTAMINANT NUMBER
C       DD,NROW,NCOL=NAME AND DIMENSIONS OF MAT DD
C       CDI,NROW2,NCOL2=NAME AND DIMENSIONS OF MAT CDI
C       LIN2=NUMBER OF DEVICES IN MAT DD
C  OUTPUT
C       LOADS INTERNAL GENERATION FROM MAT CDI INTO MAT DD COL 19
C
         DD(1,19)=CDI(I,1)
         DD(2,19)=0
         DO 10 J=3,LIN2
           DD(J,19)=CDI(I,J+7)
 010  CONTINUE
         RETURN
C        ************ END OF SUBROUTINE LDIGEN  *************************
         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\LODEFF. Options: /C 80 /L /BY     05/21/92 12:57:05

C        *******************************************************************
C        *                    SUBROUTINE LODEFF                            *
C        * SUBROUTINE TO LOAD LAST INCR EFF FROM MAT CC INTO MAT DD COL 20*
C        *      USES ADJUSTABLE SIZE ARRAYS                                *
C        *******************************************************************
         SUBROUTINE LODEFF(I,DD,NROW,NCOL,CC,NROW1,NCOL1,LIN2)
         INTEGER NROW,NCOL,NROW1,NCOL1
         REAL DD(NROW,NCOL)
         REAL CC(NROW1,NCOL1)

C      INPUTS:
C        I=CONTAMINANT LINE NUMBER IN MAT CC
C        DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
C        CC,NROW1,NCOL1=NAME & DIMENSIONS OF MAT CC
C        LIN2=NO. OF DEVICES IN MAT DD

         DD(1,20)=0
         K=7
         DO 100 J=2,LIN2
           DD(J,20)=CC(I,K)
           K=K+3
     100 CONTINUE
         RETURN
C        ************* END OF SUBROUTINE LODEFF  *************************
         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\MASBAL. Options: /C 80 /L /BY     05/21/92 12:57:51

C       ****************************************************************
C       *                 MASS BALANCE SUBROUTINE-MASBAL               *
C       * FOR 1 CONT AT A TIME AT A GIVEN DEVICE EFFICIENCY CALCULATES *
C       * CAV,CFINAL,CEQ,M.REMOVED (ALL DEV+CABIN)-DATA PUT IN MAT DD  *
C       ****************************************************************
C NOTE: BEFORE RUNNING THIS SUBROUTINE MUST ZERO MAT DD COL 17-21
C         (DONE BY PRAFIL) &  LOAD REM EFF FOR EACH DEVICE INTO
C         MAT DD COL 20 (DONE BY LODEFF OR CNRSUB)

      SUBROUTINE MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +CAVCLC,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
      INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
      REAL DD(NROW,NCOL)
      REAL CC(NROW1,NCOL1)
      REAL CDI(NROW2,NCOL2)
C
C SUBROUTINES REQUIRED:
C  CALCM-CALCULATE CIN,COUT,M.REM,SUM MASS REM-IN MAT DD
C  LDIGEN-LOAD INTERNAL M.GEN FOR DEVICE + CABIN FROM CDI INTO DD COL 19
C  PCAVCF-USING CEQ & CINIT, CALC CFINAL & CAVERAGE

C INPUTS:
C  FROM PCSET, PREDCT, AND CONVRG
C    I=CONTAMINANT NO.
C    TN=INCREMENT END TIME(HRS); TNI=INCR BEGINNING TIME HRS
C    DD,NROW,NCOL=NAME AND SIZE OF MAT DD
C    CC,NROW1,NCOL1=NAME AND SIZE OF MAT CC
C    CDI,NROW2,NCOL2=NAME AND SIZE OF MAT CDI
C    LIN=NO. OF CONT IN MAT CDI
C    LIN2=NO. OF DEVICES IN MAT DD
C  FROM CALCM
C    SMREM=SUM OF MASS REM FOR ALL DEVICES (MG/HR)-TOTAL OF DD COL 21
C    SMGEN=SUM OF MASS GEN IN ALL DEVICES INCL CABIN(MG/HR)-DD COL 19
C  FROM LDIGEN
C    IT LOADS CABIN M.GEN (MG/HR) FROM MAT CDI INTO DD(1,19)
C    IT LOADS M.GEN DEVICES FROM MAT CDI COL 2-15,19 INTO DD COL 19
C  FROM PCAVCF
C    CAVCLC=CALC INCR CABIN CONT CONC (MG/CU M)
C    CFINAL=FINAL INCR CABIN CONT CONC (MG/CU M)
C OUTPUTS:
C  TO PCSET, PREDCT, AND CONVRG
C    CAVCLC=CALCULATED CABIN AVERAGE CONC(MG/CU M)
C    CEQLIB=CABIN EQUILIBRIUM CONCENTRATION (MG/CU M)
C    CFINAL=INCREMENT FINAL CABIN CONCENTRATION (MG/CU M)
C    PUTS M.REM FOR CABIN + DEVICES IN MAT DD COL 21
C  TO CALCM
C    CAV=CABIN CONT CONC (MG/CU M)
```

```
C     DD(1,19)=50  (CABIN M.GEN)
C     OTHER DEVICES DD(2-15,19) MUST =0 AT THIS POINT (SEE PRAFIL)
C   TO LDIGEN
C      I=CONTAMINANT NUMBER
C      DD,NROW,NCOL=NAME AND DIMENSIONS OF MAT DD
C      CDI,NROW2,NCOL2=NAME AND DIMENSIONS OF MAT CDI
C      LIN2=NUMBER OF DEVICES IN MAT DD
C   TO PCAVCF
C     TN,TN1
C     CINIT=INITIAL INCR CABIN CONT CONC  (MG/CU M)=CC(I,1)
C     CEQLIB=CABIN EQULIB CONC  (MG/CU M)
C     SQEFFN=SUM OF Q*REM EFF NET FOR ALL DEVICES  (CU M/HR)
C     CVOL=CABIN VOL  (CU M)=DD(1,9)
C     SMNTC=SUM OF MASS NET TO CABIN(MG/HR)

C   CABIN VOL  (CU M)
        CVOL=DD(1,9)
C   CINITIAL  (MG/CU M)
        CINIT=CC(I,1)

C   EVALUATE SUM Q*REM EFF NET USING M.GEN IN DEVICES=0  (DD COL 19)
C     SET CABIN AVERAGE CONCENTRATION = TO ARBITRARY VALUE OF 100
C     AND INTERNAL GENERATION IN DEVICES =0  (NOT YET LOADED)
C        SET CABIN M.GEN=ARBITRARY VALUE OF 50  (DD(1,19))
C        SMGEN=SUM M.GEN IN ALL DEVICES +CABIN  (MG/HR)
C        SMREM=SUM M. REMOVED BY ALL DEVICES  (MG/HR)
C        CAV=CABIN AVERAGE CONCENTRATION  (MG/CU M)
         CAV=100
         DD(1,19)=50

         CALL CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
C        SQEFFN=SUM OF Q*REMOVAL EFF NET  (MG/HR)
         SQEFFN=SMREM/CAV

C
C   LOAD INTERNAL GENERATION FOR ALL DEV+CABIN FROM CDI INTO DD COL 19
         CALL LDIGEN(I,DD,NROW,NCOL,CDI,NROW2,NCOL2,LIN2)
C
C   EVALUATE SUM OF M.NET TO CABIN=M.GEN CABIN+SUM M.GEN ALL DEVICES -
C        SUM M.REM ALL DEVICES
C        SMNTC=SUM M.NET TO CABIN=AMT GEN WHICH GETS TO THE CABIN DIRECTLY
C        SET C CABIN AV=0
         CAV=0
C        GET SUM MASS GEN CABIN+ INTERNAL DEVICES AND SUM MASS REMOVED ALL
C        DEVICES FROM SUBROUTINE-SINCE CABIN C=0 NO CABIN CONT WILL BE REM
         CALL CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
         SMNTC=SMGEN-SMREM
C         NOTE:SMNTC IS ALSO PUT IN DD(1,21) BY CALCM
C
C   GET CALCULATED CABIN EQUILIBRIUM CONCENTRATION (CAVCLC)  (MG/CU M)
```

```
C
      IF(SQEFFN.LT.1E-10) THEN
        IF(CVOL.EQ.0) THEN
          CFINAL=1E10
        ELSE
          CFINAL=CINIT+(TN-TN1)*SMGEN/CVOL
        ENDIF
        CAVCLC=(CINIT+CFINAL)/2
        CEQLIB=1E10
      ELSE
        CEQLIB=SMNTC/SQEFFN
C       CALCULATE CAVCLC AND CFINAL FROM SUBROUTINE
        CALL PCAVCF(TN,TN1,CEQLIB,SQEFFN,CVOL,SMNTC,CINIT,
     +  CAVCLC,CFINAL)
      ENDIF
C USING CAV CALC EVALUATE M.REM FOR CABIN + DEVICES AND PUT
C     IN MAT DD COL 21
      CAV=CAVCLC
      CALL CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
C
      RETURN
C END OF SUBROUTINE MASBAL
C ************** END OF SUBROUTINE  MASBAL **************************
      END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\MCALC.F Options: /C 80 /L /BY   05/21/92 12:57:59

```
C        ********************************************************************
C        *                   SUBROUTINE MCALC                               *
C        *     MAIN CALCULATION LOOP SUBROUTINE FOR 1 TIME INCREMENT        *
C        *     FOR ALL CONTAMINANTS ONE AT A TIME                           *
C        *     BASED ON SUM MASS REM LAST INCR, FOR EACH CONT               *
C        *     CALCULATE NEW REMOVAL EFF, CAV CALC CABIN,                   *
C        *     CEQUILIB, CFINAL, & M.REMOVED ALL DEVICES-PUT IN MAT CC      *
C        ********************************************************************
         SUBROUTINE MCALC(I,TN,TN1,DD,NROW,NCOL,
        +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,LIN,LIN2,
        +PRTSW2,PRTSW3,PRTSW4,IMSGDN)
         INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,PRTSW2,PRTSW3,PRTSW4,KK
         CHARACTER NN(NROW1)*30
         REAL DD(NROW,NCOL)
         REAL CC(NROW1,NCOL1)
         REAL CDI(NROW2,NCOL2)
C SUBROUTINES REQUIRED:
C     PREDCT=PREDICT CAV BASED ON M.GEN OF THIS INCR & REM EFF OF LST INC
C     CONVRG=CALC CAV CALC,CEQ,CFINAL,M.REM,REM EFF
C     CROUT=PRINT TEST VALUES OF MAT CC
C     RROUT=PRINT TEST VALUES OF MAT DD
C
C INPUTS:
C  FROM MAIN PROG
C     I=CONTAMINANT NO.
C     TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
C     DD,NROW,NCOL=NAME & DIM OF MAT DD
C     CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C     CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C     NN=NAME OF MAT NN
C     LIN=NUMBER OF CONTAMINANTS IN MAT NN & CDI
C     LIN2=NO. DEVICES IN MAT DD
C  FROM PREDCT
C     CAVPRD=PRED CABIN CONT CONC (MG/CU M)
C  FROM CONVRG
C     CAVCLC=CALC CABIN CONT CONC (MG/CU M)
C     CFINAL=FINAL INCR CABIN CONT CONC(MG/CUM)
C     KK=COUNTER FOR CONVERGENCE
C     CEQLIB=EQUILIBRIUM CABIN CONT CONC(MG/CU M)
C     IMSGDN=DEVICE NO FOR MESSAGE AND TEXT PRINTOUT OUTPUT
C OUTPUTS:
C  TO PREDCT
C     I=CONTAMINANT NO.
C     TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
C     DD,NROW,NCOL=NAME & DIM OF MAT DD
C     CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C     CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
```

```
C     LIN=NO. OF CONTAMINANTS IN MAT CDI
C     LIN2=NO. DEVICES IN MAT DD
C   TO CONVRG
C     I=CONTAMINANT NO.
C     TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
C     DD,NROW,NCOL=NAME & DIM OF MAT DD
C     CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C     CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C     LIN=NO. OF CONTAMINANTS IN MAT CDI
C     LIN2=NO. DEVICES IN MAT DD
C     CAVPRD=PREDICTED CABIN CONC FOR INCREMENT (MG/CU M)
C   TO MAT CC
C     PUTS CAVCLC,CEQLIV,&CFINAL IN CC(I,2-3 &4)
C     PUTS REM EFF FROM DD COL20 IN CC(I,7-10-13 ETC)
C     PUTS M.REM FOR EACH DEV FROM DD COL21 IN CC(I,6-9-12 ETC)
C     PUTS SUM MASS REM FOR EACH DEV IN CC(I,8-11-14 ETC)
C
C
C     BEGIN LOOP FOR EACH CONTAMINANT FOR EACH TIME INCREMENT
        DO 100 I=1,LIN
C       CALC CAV PRED CABIN FOR CONT BASED ON REM EFF OF LAST INCREMENT
C       AND GENERATION RATES OF THIS INCREMENT
        CALL PREDCT(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
       +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,LIN,LIN2,NN)

C
C       CONVERGE UNTIL CCALC=CPRED
        CALL CONVRG(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
       +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,KK,LIN,
       +LIN2,NN,PRTSW3,IMSGDN)
C
C       IF KK>20 THEN BEGIN 1/20 TIME INCREMENT CONVERGENCE ROUTINE
        IF (KK.GT.20) THEN
           KK=1
C          BEGIN 1/10 INCREMENT CONVERGENCE ROUTINE
C          NEW INCREMENT INITIAL TIME (HRS)
            TN1NEW=TN1
C          NEW TIME INCREMENT (HRS)
            BINEW=(TN-TN1)/20
C          BEGIN LOOP FOR 1/20 INCREMENT SIZE TIME INCREMENT
C           NEW INCREMENT FINAL TIME (HRS)
  200      TNNEW=TN1NEW+BINEW

C          ZERO MAT DD COL 17-21
            CALL PRAFIL(DD,NROW,NCOL,17,21)
C          LOAD EFFICIENCY FROM LAST INCREMENT INTO MAT DD COL 20
            CALL LODEFF(I,DD,NROW,NCOL,CC,NROW1,NCOL1,LIN2)
C          CALC CAV PRED CABIN FOR CONT BASED ON REM EFF OF LAST INCREMENT
C          AND GENERATION RATES OF THIS INCREMENT
           CALL PREDCT(I,TNNEW,TN1NEW,CAVPRD,DD,NROW,NCOL,
```

```
      +    CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,LIN,
      +    LIN2,NN)
C
C          CONVERGE UNTIL CCALC=CPRED
           CALL CONVRG(I,TNNEW,TN1NEW,CAVPRD,DD,NROW,NCOL,
      +    CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,KK,LIN,
      +    LIN2,NN,PRTSW3,IMSGDN)
C
C          IF KK>20 THEN PRINT CONVERGENCE WARNING
           IF (KK.GT.20) THEN
             OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
             WRITE(IMSGDN,*)'WARNING: CALCULATION DID NOT CONVERGE FOR'
             WRITE(IMSGDN,*)'           FULL AND 1/20 INCREMENT ROUTINES'
             WRITE(IMSGDN,50) I,NN(I),TN1NEW,TNNEW
   050         FORMAT (1X,'CONT NO.= ',I4,2X,A,/,1X,
     +          'FOR INCREMENT INIT & FINAL TIMES= ',F8.2,F8.2)
             CLOSE(IMSGDN)
           ENDIF

C          FILL MAT CC WITH RESULTS
C          PUT CAVCLC,CEQLIB, AND CFINAL IN CC
             CC(I,2)=CAVCLC
             CC(I,3)=CEQLIB
             CC(I,4)=CFINAL
C          PUT REM EFF FROM LAST ITER DD COL 20 IN CC(I,7-10-13ETC)
             K=7
             DO 102 J=2,LIN2
               CC(I,K)=DD(J,20)
               K=K+3
   102       CONTINUE

C          TAKE CABIN M.REM(MG/HR) FROM DD(1,21) & PUT IN MAT CC(I,5)
            CC(I,5)=DD(1,21)
C          TAKE M.REM FROM DD COL 21 & PUT IN CC(I,8-11-14ETC)
             K=8
             DO 103 J=2,LIN2
               CC(I,K)=DD(J,21)
               K=K+3
   103       CONTINUE

C          CALCULATE SUM OF MASS REMOVED IN DEVICES + CABIN TO DATE AND
C          PUT IN CC(I,6-9-12ETC)
           K=5
           DO 104 J=1,LIN2
             CC(I,K+1)=CC(I,K+1)+CC(I,K)*(TNNEW-TN1NEW)
             K=K+3
   104     CONTINUE
C
C          IF PRTSW2=1 THEN PRINT MAT CC INFO FOR THIS CONTAMINANT
             IF (PRTSW2.EQ.1)  THEN
```

```
                 OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
                   WRITE(IMSGDN,*)'NINC,TN,TN1 ',NINC,TN,TN1
                   WRITE(IMSGDN,*)'PRINTOUT FOR ONE CONT INSIDE 1/20 INCR
     + LOOP OF MCALC'
                   WRITE(IMSGDN,*)'INFO FROM MAT CC'
                 CLOSE(IMSGDN)
                 CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,I,I,IMSGDN)
             ENDIF
C

C            REPEAT LOOP FOR 1/20 INCREMENT IF END OF 1/20 INCREMENT TIME
C            (TNNEW) IS < THAN END OF LARGER TIME INCR (TN)
C            ELSE IF TNNEW>=TN, END 1/20 ENCR CONV & PRINT ANSWERS+REPEAT
C             FOR ANOTHER CONTAMINANT

             IF (TNNEW.LT.TN) THEN
C               RESET FOR ANOTHER 1/20 TIME INCREMENT
                TN1NEW=TNNEW
                CC(I,1)=CC(I,4)
                GO TO 200
             ELSE
C               END 1/10 INCR CONV ROUTINE-REPEAT FOR ANOTHER CONT
                GOTO 100
             ENDIF

C
C          END OF CONVERGENCE ROUTINE
           ENDIF

C        CALC SUM MASS REMOVED & FILL MAT CC WITH RESULTS
C        PUT CAVCLC,CEQLIB, AND CFINAL IN CC
           CC(I,2)=CAVCLC
           CC(I,3)=CEQLIB
           CC(I,4)=CFINAL
C        PUT REM EFF FROM LAST ITER DD COL 20 IN CC(I,7-10-13ETC)
           K=7
           DO 302 J=2,LIN2
              CC(I,K)=DD(J,20)
              K=K+3
  302     CONTINUE

C        TAKE CABIN M.REM(MG/HR) FROM DD(1,21) & PUT IN MAT CC(I,5)
          CC(I,5)=DD(1,21)
C        TAKE M.REM FROM DD COL 21 & PUT IN CC(I,8-11-14ETC)
           K=8
           DO 303 J=2,LIN2
              CC(I,K)=DD(J,21)
              K=K+3
  303     CONTINUE

```

```
C      CALCULATE SUM OF MASS REMOVED IN DEVICES + CABIN TO DATE AND PUT
C        IN CC(I,6-9-12ETC)
         K=5
         DO 304 J=1,LIN2
           CC(I,K+1)=CC(I,K+1)+CC(I,K)*(TN-TN1)
           K=K+3
  304  CONTINUE
C

C      IF PRTSW4=1 THEN PRINT MAT DD+MAT CC INFO FOR THIS CONTAMINANT
           IF (PRTSW4.EQ.1)  THEN
             OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
               WRITE(IMSGDN,*)'PRINTOUT FOR ONE CONT AT END OF MCALC'
               WRITE(IMSGDN,*)'INFO FROM MAT CC'
             CLOSE(IMSGDN)
             CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,I,I,IMSGDN)
             OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
               WRITE(IMSGDN,*)'INFO FROM MAT DD'
             CLOSE(IMSGDN)
            CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN)
           ENDIF
C
C        END OF I LOOP FOR EACH CONTAMINANT
C
  100  CONTINUE
         RETURN
C      ****** END OF SUBROUTINE MCALC **********************************
         END
```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\PCAVCF. Options: /C 80 /L /BY   05/21/92 12:57:51

```
C       ****************************************************************
C       *                   SUBROUTINE PCAVCF                          *
C       * SUBROUTINE TO PREDICT INCREMENT CALCULATED AVERAGE EFF       *
C       * (C AV CALC), FINAL EFF (CFINAL) & CABIN CONTAMINANT          *
C       * CONCENTRATION                                                *
C       ****************************************************************

        SUBROUTINE PCAVCF(TN,TN1,SCEQLIB,SQEFFN,CVOL,SMNTC,CINIT,
       + CAVCLC,SCFINAL)
        DOUBLE PRECISION EXPON,CEQLIB,CFINAL
        CEQLIB=DBLE(SCEQLIB)

C SUBROUTINES REQUIRED: NONE
C
C INPUTS:
C  TN, TN1=INITIAL & FINAL INCREMENT TIME (HRS)
C  SCEQLIB(CEQLIB)=EQUILIBRIUM CABIN CONC (MG/CUM)
C  SQEFFN=SUM Q*REMOVAL EFF NET (MG/HR)
C  CVOL=CABIN VOLUME (CU M)
C  SMNTC=SUM MASS CONT NET TO CABIN (MG/HR)
C  CINIT=INITIAL INCREMENT CONT CONC (MG/CU M)
C OUTPUTS:
C  SCAVCLC(CAVCLC)=CALC AVERAGE CABIN CONC (MG/CU M)
C  SCFINAL(CFINAL)=FINAL INCREMENT CONC (MG/CU M)
C
        IF(CVOL.LE.0) THEN
          CAVCLC=CEQLIB
          CFINAL=CEQLIB
          GOTO 99
        ENDIF
C       CALCULATION FOR CFINAL
        EXPON=(TN-TN1)*SQEFFN/CVOL
        IF(ABS(EXPON).GT.50) THEN
          CAVCLC=CEQLIB
          CFINAL=CEQLIB
          GOTO 99
        ENDIF
        IF(ABS(EXPON).LT.1E-6) THEN
          CFINAL=CINIT+SMNTC*(TN-TN1)/CVOL
          CAVCLC=(CINIT+CFINAL)/2
          CEQLIB=1E10
          GOTO 99
        ELSE
          CFINAL=CINIT+(SMNTC/SQEFFN-CINIT)*(1-EXP(-EXPON))
        ENDIF
C       CALCULATION FOR C AVERAGE CALC
        IF ((CINIT.EQ.CFINAL).OR.(CFINAL.EQ.CEQLIB)) THEN
```

```
        CAVCLC=CFINAL
        GOTO 99
       ENDIF
       IF ((CEQLIB-CINIT)/(CEQLIB-CFINAL).LT.1E-6) THEN
         CAVCLC=(CINIT+CFINAL)/2
       ELSE
         CAVCLC=CEQLIB-(CFINAL-CINIT)/LOG((CEQLIB-CINIT)/
     +  (CEQLIB -CFINAL))
       ENDIF
 099 CONTINUE
       SCFINAL=REAL(CFINAL)
       RETURN
C      *********** END OF SUBROUTINE PCAVCF  **************************
       END
C

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\PCSET.F Options: /C 80 /L /BY   05/21/92 12:58:31

```
C       ******************************************************************
C       *                  SUBROUTINE   PCSET                            *
C       *       SUBROUTINE  FOR PRECALCULATION SETUP ROUTINE             *
C       *       FOR ALL CONTAMINANTS ONE AT A TIME                       *
C       *       CALL EFF SUBROUTINES FOR DEVICES; GET CAV CABIN PRED,    *
C       *       CEQUILIB, CFINAL, & M.REMOVED ALL DEVICES-PUT IN MAT CC  *
C       ******************************************************************
        SUBROUTINE PCSET(TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
       +CDI,NROW2,NCOL2,LIN2,NN,PRTSW1,IMSGDN)
        INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,PRTSW1
        CHARACTER NN(NROW1)*30
        REAL DD(NROW,NCOL)
        REAL CC(NROW1,NCOL1)
        REAL CDI(NROW2,NCOL2)

C SUBROUTINES REQUIRED:
C    PRAFIL-ZERO MAT DD COL 17-21
C    CNRSUB-USING CAV=1E-20, FIND REMOVAL EFF AND PUT IN MAT DD COL 20
C    MASBAL-CALC CAV CALC PRED, CEQ, CFINAL, M.REM
C    CROUT-TEST PRINTOUT OF CONT INFO
C
C INPUTS:
C  FROM MAIN CALC LOOP
C    TN1=INCREMENT INITIAL TIME (HRS)
C    LIN=NO. OF CONT IN MAT CC AND NN
C    DD,NROW,NCOL=NAME & SIZE OF MAT DD
C    CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
C    CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
C    LIN2=NO. DEVICES IN MAT DD
C    NN=NAME OF MAT NN
C    PRTSW1=PRINTSWITCH WHICH CONTROLS TEST PRINTOUT
C    IMSGDN=DEVICE NUMBER FOR MESSAGES AND TEST PRINTOUT
C  FROM PRAFIL
C    PUTS ZEROS IN MAT DD COL 17-21
C  FROM CNRSUB
C    CNRSUB PUTS REM EFF(DEC) FOR EACH DEVICE IN MAT DD COL 20
C  FROM MASBAL (PREDICTED VALUES)
C    CAVCLC=AVERAGE CABIN CONC  (MG/CU M)
C    CFINAL=FINAL INCREMENT CONT CONC (MG/CU M)
C    CEQLIB=EQUILIBRIUM CONT CONC (MG/CU M)
C    M.REM FOR ALL DEVICES PLACED IN COL 21 OF MAT DD
C OUTPUTS
C  TO MAIN PROGRAM
C    PUT IN MAT CC
C       CAVPRD=PRED CABIN AV CONC (MG/CU M): =CC(I,2)
C       CEQLIB=EQUILIBRIUM CABIN CONT CONC (MG/CU M):=CC(I,3)
C       CFINAL=FINAL CABIN CONT CONC (MG/CU M):=CC(I,4)
```

```
C       PUTS REM EFF FROM DD COL 20 IN CC(I,7-10-13-16 ETC)
C       PUTS M.REM IN CC(I,5-8-11-14...)
C  TO PRAFIL
C     NAME AND SIZE OF MAT DD+FIRST & LAST COLUMN TO ZERO
C  TO CNRSUB
C    I=CONT NO.
C    TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
C    CAVPRD=CABIN AVERAGE CONT CONC (MG/CU M)
C    DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
C    CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C    CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C    LIN2=NO. ACTIVE DEVICES IN MAT DD
C  TO MASBAL
C    I=CONT NO.
C    TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
C    DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
C    CALCLC=CALC CABIN AV CONC (MG/CU M)
C    CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C    CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C    CFINAL=CABIN FINAL CONCENTRATION (MG/CU M)
C   CEQLIB=CABIN EQUILIBRIUM CONCENTRATION (MG/CU M)
C   LIN=NO. OF CONTAMINANTS IN MAT CDI
C   LIN2=NO. ACTIVE DEVICES IN MAT DD
C  TO RROUT
C    MATRIX NAME,#ROWS,#COLS,FIRST & LAST COL TO PRINT,#LINES TO PRINT

C   BASIC TIME INCREMENT (HRS)
        BINC=DD(1,11)
C   SET FINAL INCREMENT TIME (HRS)
        TN=0.1*BINC/24
C   BEGIN LOOP FOR EACH CONTAMINANT - ONE AT A TIME
C       CALCULATE REM EFF FOR EACH DEVICE, GET M.REM, CAV CABIN CALC
C       CEQULIB, CFINAL-PUT IN MAT CC
        DO 100 I=1,LIN
C          ZERO MAT DD COL 17 TO 21
             CALL PRAFIL(DD,NROW,NCOL,17,21)
C          SET CAVPRD = MINIMUM VALUE TO ALLOW COMPUTATION
             CAVPRD=1E-20
             DD(J,22)=CAVPRD
C          CALC REMOVAL EFFICIENCYS (THROUGH EFF CALLING SUBROUTINE)
C            THIS STORES REM EFF IN MAT DD COL 20 FOR EACH DEVICE
           CALL CNRSUB(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +     CDI,NROW2,NCOL2,LIN2)
C          CALL MASS BALANCE-GET CAVPRD(=CAVCLC IN MASBAL),CEQ,CFINAL,M.REM
             CALL MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +     CAVCLC,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
           CAVPRD=CAVCLC
C          PUT CAVPRD,CEQLIB,& CFINAL IN MAT CC
             CC(I,2)=CAVPRD
             CC(I,3)=CEQLIB
```

```
                CC(I,4)=CFINAL
C             GET REM EFF FROM DD COL 20 AND PUT IN CC(I,7-10-13 ETC)
                K=7
                DO 101 J=2,LIN2
                  CC(I,K)=DD(J,20)
                  K=K+3
  101           CONTINUE
C           TAKE M.REMOVED FROM MAT DD COL 21 AND PUT IN MAT CC(I,5-8-...)
C               CABIN REMOVAL RATE
                CC(I,5)=DD(1,21)
C               DEVICE 2-15 REMOVAL RATE
                K=8
                DO 102 J=2,LIN2
                  CC(I,K)=DD(J,21)
                  K=K+3
  102           CONTINUE
C           IF PRTSW1=1 THEN PRINT MAT DD+MAT CC INFO FOR THIS CONTAMINANT
                IF (PRTSW1.EQ.1)  THEN
                  OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
                WRITE(IMSGDN,*)'PRINTOUT FOR ONE CONT AT END OF PCSET'
                  WRITE(IMSGDN,*)'INFO FROM MAT CC'
                  CLOSE (IMSGDN)
                  CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,I,I,IMSGDN)
                  OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
                  WRITE(IMSGDN,*)'INFO FROM MAT DD'
                  CLOSE (IMSGDN)
                  CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN)
                ENDIF
C
  100 CONTINUE
C
            RETURN
C           **********  END OF SUBROUTINE PCSET  ****************************
            END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\PRAFIL. Options: /C 80 /L /BY     05/21/92 12:58:44

C       *****************************************************************
C       *                  SUBROUTINE PRAFIL                            *
C       *   SUBROUTINE TO FILL ADJUSTABLE SIZE REAL ARRAY WITH ZEROS    *
C       *   PARTIAL FILL-FROM COL FSTCOL TO COL LSTCOL                  *
C       *****************************************************************
        SUBROUTINE PRAFIL(X,NROW,NCOL,FSTCOL,LSTCOL)
        INTEGER NCOL,NROW,FSTCOL,LSTCOL
        REAL X(NROW,NCOL)

C     INPUTS:
C       X,NROW,NCOL=NAME AND DIMENSIONS OF MATRIX X
C       FSTCOL,LSTCOL=FIRST AND LAST COLUMN TO FILL WITH ZEROS

        IF(FSTCOL.GT.NCOL) FSTCOL=NCOL
        IF(LSTCOL.GT.NCOL) LSTCOL=NCOL
        IF(FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
        DO 110 I=1,NROW
        DO 100 J=FSTCOL,LSTCOL
          X(I,J)=0.0
  100  CONTINUE
  110  CONTINUE
        RETURN
C       *************   END OF SUBROUTINE PRAFIL  ***********************
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\PRFANS. Options: /C 80 /L /BY     05/21/92 12:58:58

C   FILE PRFANS
C        ******************************************************************
C        *          SUBROUTINE PRCDA                                       *
C        * PROGRAM TO PRINT ANSWERS FOR CONCENTRATION DATA                 *
C        ******************************************************************

C        NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
C                 (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED

         SUBROUTINE PRCDA(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
        +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
        +PRTSW8,PRTSW9,FCPLOT,IDEVN1)
C    SUBROUTINES REQUIRED:
C        HDG1,HDG2,HDG3,HDG4

C        TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
C        LIN=TOTAL NUMBER OF CONTAMINANTS
C        CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
C        CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
C        NN=NAME OF MAT NN
C        IDEVNO=DEVICE NUMBER FOR OUTPUT
C        NINC=TIME INCREMENT NUMBER
C           =0 THEN PRINT HDG3 WITH PCALC
C           =-1 THEN PRINT HDG3 WITH FINAL
C           ELSE PRINT HDG2 WITH INCREMENT NUMBER
C        IMONTH..IMINUTE=TIME AND DATE INFO
C        FNAME=FILE NAME OUTPUT DATA IS STORED ON
C        IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT

         REAL CC(NROW1,NCOL1)
         REAL CDI(NROW2,NCOL2)
         CHARACTER CNAME*30,FNAME*24,ECHR*1,FCPLOT*24
         CHARACTER NN(NROW1)*30
         INTEGER PRTSW8,PRTSW9,NINC
C        ECHR=EXCEEDS MAC CHARACTER (Y OR N)
         IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
C        INCREMENT PAGE COUNTER BY ONE
         IPGCTR=IPGCTR+1

C        START FIRST PAGE
C        PRINT FORM FEED
          WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
   020   FORMAT('1')
C        PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &4
           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
   040    FORMAT(1X,'')
           CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
```

```
         IF(NINC.EQ.0) THEN
            CALL HDG3(1,TN1,TN,IDEVNO)
         ELSEIF(NINC.EQ.-1) THEN
            CALL HDG3(2,TN1,TN,IDEVNO)
         ELSE
            CALL HDG2(NINC,TN1,TN,IDEVNO)
         ENDIF
         CALL HDG4(IDEVNO)
C       PRINT ANOTHER BLANK LINE
         WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)

C       BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
        DO 100 I=1,LIN

C          CNAME=CONTAMINANT NAME
           CNAME=NN(I)
C          FCONC=FINAL CONT CONCENTRATION (MG CU M)
           FCONC=CC(I,4)
C          RMAC=MAXIMUM ALLOWABLE CONCENTRATIION (MG/CU M)
           RMAC=CDI(I,9)

C          IF CABIN CONC>MAC PRINT 'Y' OTHERWISE PRINT 'N'
           IF(FCONC.GT.RMAC) THEN
              ECHR='Y'
           ELSE
              ECHR='N'
           ENDIF
C
C          PRINT 56 LINES OF DATA AND THEN START NEW PAGE
           WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,CNAME,FCONC,RMAC,ECHR
  010     FORMAT(1X,I4,1X,A,1X,G11.4,1X,G11.4,5X,A)

C          CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
           IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
              IPGCTR=IPGCTR+1
C             START SUBSEQUENT PAGES
C             PRINT FORM FEED
                WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
  050           FORMAT('1')
C             PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &4
              WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
  030         FORMAT(1X,'')
              CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
              IF(NINC.EQ.0) THEN
                CALL HDG3(1,TN1,TN,IDEVNO)
              ELSEIF(NINC.EQ.-1) THEN
                CALL HDG3(2,TN1,TN,IDEVNO)
              ELSE
                CALL HDG2(NINC,TN1,TN,IDEVNO)
```

```
                 ENDIF
                 CALL HDG4(IDEVNO)
C                PRINT ANOTHER BLANK LINE
               WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)

             ENDIF
  100 CONTINUE
           ENDIF
C
C       ********* WRITE CONCENTRATION DATA TO A PLOT FILE *************
           IF (NINC.NE.-1) THEN
           IF ((PRTSW9.EQ.1).OR.(PRTSW9.EQ.3)) THEN
             DO 120 I=1,LIN,300
                IS=I
                IE=I+299
                IF (IE.GT.LIN) IE=LIN
                WRITE (IDEVN1,110,IOSTAT=IOVAL,ERR=900) TN1,TN,
     +            (CC(J,4),J=IS,IE)
  110              FORMAT (T2,2(F8.3,1X),300(G11.4,:,1X))
  120       CONTINUE
           ENDIF
           ENDIF
           GO TO 999
  900 WRITE(*,*)'IO ERROR IN PRCDA= ',IOVAL
  999 RETURN
           END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0


C          ***************************************************************
C          *          SUBROUTINE PRREM1                                  *
C          * PROGRAM TO PRINT ANSWERS-RATE OF CONTAMINAMT REMOVAL (MG/HR) *
C          * SHEET 1                                                     *
C          ***************************************************************

C        NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
C                (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED

           SUBROUTINE PRREM1(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
         +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)

C     SUBROUTINES REQUIRED:
C          HDG1,HDG2,HDG3,HDG8

C        TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
C          LIN=TOTAL NUMBER OF CONTAMINANTS
C          CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
C        CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
C          NN=NAME OF MAT NN
```

```
C        IDEVNO=DEVICE NUMBER FOR OUTPUT
C        NINC=TIME INCREMENT NUMBER
C          =0 THEN PRINT HDG3 WITH PCALC
C          =-1 THEN PRINT HDG3 WITH FINAL
C          ELSE PRINT HDG2 WITH INCREMENT NUMBER
C        IMONTH..IMINUTE=TIME AND DATE INFO
C        FNAME=FILE NAME OUTPUT DATA IS STORED ON
C       IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT

         REAL CC(NROW1,NCOL1)
         REAL CDI(NROW2,NCOL2)
        CHARACTER CNAME*30,FNAME*24
         CHARACTER NN(NROW1)*30

C       INCREMENT PAGE COUNTER BY ONE
         IPGCTR=IPGCTR+1

C        START FIRST PAGE
C        PRINT FORM FEED
          WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
  020    FORMAT('1')
C        PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
  040     FORMAT(1X,'')
          CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
           IF(NINC.EQ.0) THEN
              CALL HDG3(1,TN1,TN,IDEVNO)
           ELSEIF(NINC.EQ.-1) THEN
              CALL HDG3(2,TN1,TN,IDEVNO)
           ELSE
              CALL HDG2(NINC,TN1,TN,IDEVNO)
           ENDIF
           CALL HDG8(IDEVNO)
C        PRINT ANOTHER BLANK LINE
           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)

C        BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
         DO 100 I=1,LIN

C          PRINT 56 LINES OF DATA AND THEN START NEW PAGE
           WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
     +CC(I,5),CC(I,8),CC(I,11),CC(I,14),CC(I,17),CC(I,20),CC(I,23),
     +CC(I,26)
  010     FORMAT(1X,I4,1X,A,8(1X,G11.4))

C         CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
           IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
              IPGCTR=IPGCTR+1
C             START SUBSEQUENT PAGES
C             PRINT FORM FEED
```

```
                   WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
     050           FORMAT('1')
C                PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
                 WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
     030         FORMAT(1X,'')
               CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
                 IF(NINC.EQ.0) THEN
                   CALL HDG3(1,TN1,TN,IDEVNO)
                 ELSEIF(NINC.EQ.-1) THEN
                   CALL HDG3(2,TN1,TN,IDEVNO)
                 ELSE
                   CALL HDG2(NINC,TN1,TN,IDEVNO)
                 ENDIF
                 CALL HDG8(IDEVNO)
C                PRINT ANOTHER BLANK LINE
                 WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)

              ENDIF
     100 CONTINUE

         GO TO 999
     900 WRITE(*,*)'IO ERROR IN PRREM1= ',IOVAL
     999 RETURN
         END
```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```

C        ******************************************************************
C        *        SUBROUTINE PRREM2                                       *
C        * PROGRAM TO PRINT ANSWERS-RATE OF CONTAMINAMT REMOVAL (MG/HR) *
C        * SHEET 2                                                        *
C        ******************************************************************

C        NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
C              (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED

         SUBROUTINE PRREM2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
        +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
C     SUBROUTINES REQUIRED:
C        HDG1,HDG2,HDG3,HDG9

C        TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
C        LIN=TOTAL NUMBER OF CONTAMINANTS
C        CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
C        CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
C        NN=NAME OF MAT NN
C        IDEVNO=DEVICE NUMBER FOR OUTPUT
```

```
C       NINC=TIME INCREMENT NUMBER
C         =0 THEN PRINT HDG3 WITH PCALC
C         =-1 THEN PRINT HDG3 WITH FINAL
C         ELSE PRINT HDG2 WITH INCREMENT NUMBER
C       IMONTH..IMINUTE=TIME AND DATE INFO
C       FNAME=FILE NAME OUTPUT DATA IS STORED ON
C       IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT

        REAL CC(NROW1,NCOL1)
        REAL CDI(NROW2,NCOL2)
        CHARACTER CNAME*30,FNAME*24
        CHARACTER NN(NROW1)*30

C       INCREMENT PAGE COUNTER BY ONE
        IPGCTR=IPGCTR+1

C       START FIRST PAGE
C       DON'T PRINT FORM FEED UNLESS NO. CONT > 20
        IF(LIN.GT.20) THEN
C         PRINT FORM FEED
          WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
  020    FORMAT('1')
        ENDIF
C       PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &9
          WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
  040    FORMAT(1X,'')
          CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
          IF(NINC.EQ.0) THEN
            CALL HDG3(1,TN1,TN,IDEVNO)
          ELSEIF(NINC.EQ.-1) THEN
            CALL HDG3(2,TN1,TN,IDEVNO)
          ELSE
            CALL HDG2(NINC,TN1,TN,IDEVNO)
          ENDIF
          CALL HDG9(IDEVNO)
C       PRINT ANOTHER BLANK LINE
          WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)

C       BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
        DO 100 I=1,LIN

C         PRINT 56 LINES OF DATA AND THEN START NEW PAGE
          WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
      +CC(I,29),CC(I,32),CC(I,35),CC(I,38),CC(I,41),CC(I,44),CC(I,47)
  010    FORMAT(1X,I4,1X,A,7(1X,G11.4))

C        CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
          IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
             IPGCTR=IPGCTR+1
C            START SUBSEQUENT PAGES
```

```
C             PRINT FORM FEED
                WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
   050          FORMAT('1')
C             PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
              WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
   030        FORMAT(1X,'')
            CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
              IF(NINC.EQ.0) THEN
                CALL HDG3(1,TN1,TN,IDEVNO)
              ELSEIF(NINC.EQ.-1) THEN
                CALL HDG3(2,TN1,TN,IDEVNO)
              ELSE
                CALL HDG2(NINC,TN1,TN,IDEVNO)
              ENDIF
              CALL HDG9(IDEVNO)
C             PRINT ANOTHER BLANK LINE
              WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)

            ENDIF
   100 CONTINUE

        GO TO 999
   900 WRITE(*,*)'IO ERROR IN PRREM2= ',IOVAL
   999 RETURN
        END
```

```
NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
```

```

C       *****************************************************************
C       *       SUBROUTINE PRMAS1                                       *
C       * PROGRAM TO PRINT ANSWERS-SUM OF CONT REMOVED BY DEVICE (MG)   *
C       * SHEET 1                                                       *
C       *****************************************************************

C     NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
C             (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED

        SUBROUTINE PRMAS1(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
      +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)

C   SUBROUTINES REQUIRED:
C       HDG1,HDG2,HDG3,HDG5

C     TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
C       LIN=TOTAL NUMBER OF CONTAMINANTS
C       CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
C     CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
C       NN=NAME OF MAT NN
```

```
C       IDEVNO=DEVICE NUMBER FOR OUTPUT
C       NINC=TIME INCREMENT NUMBER
C         =0 THEN PRINT HDG3 WITH PCALC
C         =-1 THEN PRINT HDG3 WITH FINAL
C         ELSE PRINT HDG2 WITH INCREMENT NUMBER
C       IMONTH..IMINUTE=TIME AND DATE INFO
C       FNAME=FILE NAME OUTPUT DATA IS STORED ON
C      IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT

        REAL CC(NROW1,NCOL1)
        REAL CDI(NROW2,NCOL2)
        CHARACTER CNAME*30,FNAME*24
        CHARACTER NN(NROW1)*30

C       INCREMENT PAGE COUNTER BY ONE
        IPGCTR=IPGCTR+1

C       START FIRST PAGE
C       PRINT FORM FEED
         WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
  020   FORMAT('1')
C       PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &5
          WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
  040    FORMAT(1X,'')
          CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
          IF(NINC.EQ.0) THEN
             CALL HDG3(1,TN1,TN,IDEVNO)
          ELSEIF(NINC.EQ.-1) THEN
             CALL HDG3(2,TN1,TN,IDEVNO)
          ELSE
             CALL HDG2(NINC,TN1,TN,IDEVNO)
          ENDIF
          CALL HDG5(IDEVNO)
C       PRINT ANOTHER BLANK LINE
          WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)

C       BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
        DO 100 I=1,LIN

C         PRINT 56 LINES OF DATA AND THEN START NEW PAGE
          WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
     +CC(I,6),CC(I,9),CC(I,12),CC(I,15),CC(I,18),CC(I,21),CC(I,24),
     +CC(I,27)
  010    FORMAT(1X,I4,1X,A,8(1X,G11.4))

C         CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
          IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
             IPGCTR=IPGCTR+1
C            START SUBSEQUENT PAGES
C            PRINT FORM FEED
```

```
            WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
 050        FORMAT('1')
C         PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
          WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
 030      FORMAT(1X,'')
          CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
          IF(NINC.EQ.0) THEN
            CALL HDG3(1,TN1,TN,IDEVNO)
          ELSEIF(NINC.EQ.-1) THEN
            CALL HDG3(2,TN1,TN,IDEVNO)
          ELSE
            CALL HDG2(NINC,TN1,TN,IDEVNO)
          ENDIF
          CALL HDG5(IDEVNO)
C         PRINT ANOTHER BLANK LINE
          WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)

        ENDIF
  100 CONTINUE

      GO TO 999
  900 WRITE(*,*)'IO ERROR IN PRMAS1= ',IOVAL
  999 RETURN
      END
```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```

C     ******************************************************************
C     *          SUBROUTINE PRMAS2                                     *
C     * PROGRAM TO PRINT ANSWERS-SUM OF CONT REMOVED BY DEVICE (MG)    *
C     * SHEET 2                                                        *
C     ******************************************************************

C    NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
C          (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED

      SUBROUTINE PRMAS2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
     +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)

C   SUBROUTINES REQUIRED:
C      HDG1,HDG2,HDG3,HDG6

C    TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
C     LIN=TOTAL NUMBER OF CONTAMINANTS
C     CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
C    CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
C     NN=NAME OF MAT NN
C     IDEVNO=DEVICE NUMBER FOR OUTPUT
```

```
C        NINC=TIME INCREMENT NUMBER
C           =0 THEN PRINT HDG3 WITH PCALC
C           =-1 THEN PRINT HDG3 WITH FINAL
C           ELSE PRINT HDG2 WITH INCREMENT NUMBER
C        IMONTH..IMINUTE=TIME AND DATE INFO
C        FNAME=FILE NAME OUTPUT DATA IS STORED ON
C      IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT

         REAL CC(NROW1,NCOL1)
         REAL CDI(NROW2,NCOL2)
         CHARACTER CNAME*30,FNAME*24
         CHARACTER NN(NROW1)*30

C        INCREMENT PAGE COUNTER BY ONE
         IPGCTR=IPGCTR+1

C        START FIRST PAGE
C      DON'T PRINT FORM FEED UNLESS NO. CONT > 20
         IF(LIN.GT.20) THEN
C           PRINT FORM FEED
            WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
  020     FORMAT('1')
         ENDIF
C        PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &9
            WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
  040     FORMAT(1X,'')
          CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
            IF(NINC.EQ.0) THEN
               CALL HDG3(1,TN1,TN,IDEVNO)
            ELSEIF(NINC.EQ.-1) THEN
               CALL HDG3(2,TN1,TN,IDEVNO)
            ELSE
               CALL HDG2(NINC,TN1,TN,IDEVNO)
            ENDIF
            CALL HDG6(IDEVNO)
C        PRINT ANOTHER BLANK LINE
            WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)

C        BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
         DO 100 I=1,LIN

C           PRINT 56 LINES OF DATA AND THEN START NEW PAGE
            WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
     +CC(I,30),CC(I,33),CC(I,36),CC(I,39),CC(I,42),CC(I,45),CC(I,48)
  010     FORMAT(1X,I4,1X,A,7(1X,G11.4))

C         CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
            IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
               IPGCTR=IPGCTR+1
C              START SUBSEQUENT PAGES
```

```
C               PRINT FORM FEED
                  WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
    050           FORMAT('1')
C               PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
                WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
    030         FORMAT(1X,'')
              CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
                IF(NINC.EQ.0) THEN
                  CALL HDG3(1,TN1,TN,IDEVNO)
                ELSEIF(NINC.EQ.-1) THEN
                  CALL HDG3(2,TN1,TN,IDEVNO)
                ELSE
                  CALL HDG2(NINC,TN1,TN,IDEVNO)
                ENDIF
                CALL HDG6(IDEVNO)
C               PRINT ANOTHER BLANK LINE
                WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)

             ENDIF
    100 CONTINUE

         GO TO 999
    900 WRITE(*,*)'IO ERROR IN PRMAS2= ',IOVAL
    999 RETURN
         END
```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```

C        ****************************************************************
C        *        SUBROUTINE PREFF                                      *
C        * PROGRAM TO PRINT ANSWERS-END OF INCREMENT REMOVALL EFF (DEC) *
C        *                                                              *
C        ****************************************************************

C         NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
C               (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED

         SUBROUTINE PREFF(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
        +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
        +PRTSW8,PRTSW9,IDEVN2)

C    SUBROUTINES REQUIRED:
C        HDG1,HDG2,HDG3,HDG7

C      TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
C        LIN=TOTAL NUMBER OF CONTAMINANTS
C        CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
C      CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
```

```
C        NN=NAME OF MAT NN
C        IDEVNO=DEVICE NUMBER FOR OUTPUT
C        NINC=TIME INCREMENT NUMBER
C          =0 THEN PRINT HDG3 WITH PCALC
C          =-1 THEN PRINT HDG3 WITH FINAL
C         ELSE PRINT HDG2 WITH INCREMENT NUMBER
C       IMONTH..IMINUTE=TIME AND DATE INFO
C        FNAME=FILE NAME OUTPUT DATA IS STORED ON
C        IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT

         REAL CC(NROW1,NCOL1)
         REAL CDI(NROW2,NCOL2)
         CHARACTER CNAME*30,FNAME*24
         CHARACTER NN(NROW1)*30
         INTEGER PRTSW8,PRTSW9,IDEVN2,IDEVNO,I,J,K,H,NINC
         IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
C        INCREMENT PAGE COUNTER BY ONE
         IPGCTR=IPGCTR+1

C        START FIRST PAGE
C        PRINT FORM FEED
          WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
   020  FORMAT('1')
C        PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &9
           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
   040   FORMAT(1X,'')
           CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
           IF(NINC.EQ.0) THEN
              CALL HDG3(1,TN1,TN,IDEVNO)
           ELSEIF(NINC.EQ.-1) THEN
              CALL HDG3(2,TN1,TN,IDEVNO)
           ELSE
              CALL HDG2(NINC,TN1,TN,IDEVNO)
           ENDIF
           CALL HDG7(IDEVNO)
C        PRINT ANOTHER BLANK LINE
           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)

C        BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
         DO 100 I=1,LIN

C           PRINT 56 LINES OF DATA AND THEN START NEW PAGE
            WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
     +CC(I,7),CC(I,10),CC(I,13),CC(I,16),CC(I,19),CC(I,22),CC(I,25),
     +CC(I,28),CC(I,31),CC(I,34),CC(I,37),CC(I,40),CC(I,43),CC(I,46)
   010   FORMAT(1X,I4,1X,A,14(1X,F5.3))

C           CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
            IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
               IPGCTR=IPGCTR+1
```

```
C               START SUBSEQUENT PAGES
C               PRINT FORM FEED
                  WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
  050             FORMAT('1')
C               PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
                WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
  030           FORMAT(1X,'')
                CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
                IF(NINC.EQ.0) THEN
                  CALL HDG3(1,TN1,TN,IDEVNO)
                ELSEIF(NINC.EQ.-1) THEN
                  CALL HDG3(2,TN1,TN,IDEVNO)
                ELSE
                  CALL HDG2(NINC,TN1,TN,IDEVNO)
                ENDIF
                CALL HDG7(IDEVNO)
C               PRINT ANOTHER BLANK LINE
                WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)

              ENDIF
  100 CONTINUE
          ENDIF
C         *************** WRITE DATA TO A FILE FOR PLOTTING *************
          IF (NINC.NE.-1) THEN
          IF ((PRTSW9.EQ.2).OR.(PRTSW9.EQ.3)) THEN
            DO 70 I=1,LIN,300
              IS=I
              IE=I+299
              IF (IE.GT.LIN) IE=LIN
              K=7
              DO 60 H=2,15
               WRITE (IDEVN2,55,IOSTAT=IOVAL,ERR=900) TN1,TN,H,
     +                  (CC(J,K),J=IS,IE)
  055            FORMAT (T2,2(F8.2,1X),I2,1X,300(F5.3,:,1X))
                K=K+3
  060       CONTINUE
  070     CONTINUE
          ENDIF
          ENDIF
          GO TO 999
  900 WRITE(*,*)'IO ERROR IN PREFF= ',IOVAL
  999 RETURN
          END
```

```
NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\PREDCT. Options: /C 80 /L /BY   05/21/92 12:58:48

```
C       *****************************************************************
C       *                   SUBROUTINE PREDCT                           *
C       *  BASED ON REMOVAL EFF & SUM MASS REMOVED OF LAST INCREMENT,   *
C       *  AND M.GEN OF THIS INCREMENT, PREDICT CAV PRED                *
C       *     (CEQUILIB, CFINAL, M.REM ARE ALSO CALC, BUT NOT NEEDED)   *
C       *           WORKS FOR ONE CONT AT A TIME                        *
C       *****************************************************************

        SUBROUTINE PREDCT(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
       +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,LIN,LIN2,NN)

        INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
        CHARACTER NN(NROW1)*30
        REAL DD(NROW,NCOL)
        REAL CC(NROW1,NCOL1)
        REAL CDI(NROW2,NCOL2)
C
C SUBROUTINES REQUIRED:
C  PRAFIL-ZERO MAT DD COL 17-21
C  LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
C  MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
C
C INPUTS:
C  FROM MCALC
C    I=CONTAMINANT NO.
C    TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
C    DD,NROW,NCOL=NAME & DIM OF MAT DD
C    CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C    CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C    LIN=NO. OF CONTAMINANTS IN MAT CDI
C    LIN2=NO. DEVICES IN MAT DD
C    NN=NAME OF MAT NN
C  FROM LODEFF
C    TAKES REM EFF FOR LAST INCR (IN MAT CC) AND PUTS IT IN
C       MAT DD COL 20 (FOR ALL DEVICES)
C  FROM MASBAL
C    CAVCLC=CALC CABIN CONT CONC (MG/CU M)
C OUTPUTS:
C  TO LODEFF
C    I=CONTAMINANT LINE NUMBER IN MAT CC
C    DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
C    CC,NROW1,NCOL1=NAME & DIMENSIONS OF MAT CC
C    LIN2=NO. OF DEVICES IN MAT DD
C  TO MASBAL
C    TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
C    CNVERR=CONVERGENCE ERROR
C    DD,NROW,NCOL=NAME & DIM OF MAT DD
```

```
C      CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C      CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C      LIN=NO. OF CONTAMINANTS IN MAT CDI
C      LIN2=NO. DEVICES IN MAT DD
C      CVOL=CABIN VOLUME (CU M)
C      TCABIN=CABIN TEMP (DEG K)
C      CINIT=INCR INIT CABIN CONT CONC (MG/CU M)=CC(I,1)
C    TO MCALC
C      CAVCLC
C
C         ZERO MAT DD COL 17-21
           CALL PRAFIL(DD,NROW,NCOL,17,21)

C         LOAD REM EFF FROM LAST TIME INCR FROM MAT CC INTO MAT DD COL 20
            CALL LODEFF(I,DD,NROW,NCOL,CC,NROW1,NCOL1,LIN2)

C         FIND CAV PRED FOR THESE REMOVAL EFFICIENCIES
               CALL MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
         +     CAVPRD,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
C
C         SET CAV IN PRED DD(I,22)=CAV IN CALC DD(I,17)
          DO 100 J=1,LIN2
            DD(J,22)=DD(J,17)
    100 CONTINUE
C
          RETURN
C         ************ END OF SUBROUTINE PREDCT    **************************
          END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\RAFILL. Options: /C 80 /L /BY     05/21/92 12:58:55

C         *****************************************************************
C         *                  SUBROUTINE RAFILL                            *
C         *  SUBROUTINE TO FILL ADJUSTABLE SIZE REAL ARRAY WITH ZEROS     *
C         *****************************************************************
          SUBROUTINE RAFILL(XX,NROW,NCOL)
          INTEGER NROW,NCOL
          REAL XX(NROW,NCOL)

C        XX=ARRAY NAME
C        NCOL= COLUMNS IN MATRIX
C        NROW= ROWS IN MATRIX

          DO 110 I=1,NROW
          DO 100 J=1,NCOL
            XX(I,J)=0.0
  100    CONTINUE
  110    CONTINUE
          RETURN
C         ************  END OF SUBROUTINE RAFILL  **************************
          END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\RCHBD.F Options: /C 80 /L /BY   05/21/92 13:00:11

```
C       ****************************************************
C       * FILE:RCHBD.FOR                                   *
C       * SUBROUTINE FOR REM EFF-RADIAL FLOW CHARCOAL BED  *
C       * DOESNT ALLOW FOR DESORPTION                      *
C       * ASSUMES RELATIVELY THIN BED (OD CLOSE TO ID)     *
C       ****************************************************
C
        SUBROUTINE RCHBD (TN,TN1,CIIN,TCABIN,COEXIS,BEDQ,EMAX,CARTL,
       +BEDOD,BEDID,DENCH,TRTTYP,DCONT,VMOL,MW,VCONC,SOL,SMR,EFF)
C       OUTPUT:
C         EFF=BED REMOVAL EFF(DEC)
C       INPUTS:
C         TN,TN1=INCREMENT INITIAL AND FINAL TIMES(HR)
C         CIIN=BED INLET CONT CONC (MG/CU M)
C         TCABIN=CABIN TEMP (DEG K)
C         COEXIS=COEXISTANCE FACTOR
C         BEDQ=BED FLOW RATE(CU M/HR)
C         EMAX=MAXIMUM BED EFF (DEC)
C         CARTL=CARTRIDGE LENGTH (M)
C         BEDOD=BED OUTSIDE DIAMETER (M)
C         BEDID=BED INSIDE DIAMETER (M)
C         DENCH=DENSITY OF CHARCOAL IN BED (KG/CU M)
C         TRTTYP=BED TREATMENT TYPE(1=CI CHAR,2=PHOS ACID, OTHER #=NONE)
C         DCONT=CONT LIQUID DENSITY (GM/CC)
C         VMOL=CONT MOLAR VOL(GM/CC)
C         MW=CONT MOLECULAR WGT
C         VCONC=CONT VAPOR CONCENTRATION AT TCABIN (MG/CU M)
C         SOL=CONTAMINANT HENRY'S LAW CONSTANT (ATM/MOL FRACTION)
C         SMR=SUM OF CONT MASS STORED IN BED(MG)-FROM LAST INCR
C
        REAL LPREV, LAVN1, LUTIL, LIMM, LAVAV, LADS,MW
        INTEGER FACID,FCI
C
C       SET CIN=CIIN (THIS PREVENTS CIN FROM BEING PASSED BACK UP
C                TO OTHER SUBROUTINES IF IT IS SET TO 1E-20)
        CIN=CIIN

C       BED TREATMENT LOGIC
C         FACID=FLAG IF BED IS TREATED WITH PHOSPHORIC ACID (Y=1 N=0)
C         FCI=FLAG FOR CI CHAR IN BED (REMOVES FORMALDAHYDE)
        IF (NINT(TRTTYP).EQ.2) THEN
          FACID=1
          FCI=0
        ELSEIF (NINT(TRTTYP).EQ.1) THEN
          FACID=0
          FCI=1
        ELSE
```

```
          FACID=0
          FCI=0
        ENDIF
C
C       TEST FOR NO BED FLOW(BEDQ=<0) OR TN-TN1<=0;BEDL,BEDDIA,DENCH=0
        IF((BEDQ.LE.0).OR.(TN-TN1.LE.0).OR.(CARTL.LE.0).OR.(BEDOD.LE.0)
     +.OR.(DENCH.LE.0)) THEN
          EFF=0
          GOTO 199
        ENDIF
C       TEST FOR CI CHARCOAL AND FORMALDEHYDE(FCI=1 AND MW=30.03)
        IF((MW.EQ.30.03).AND.(FCI.EQ.1)) THEN
          CALL RCICH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR,BEDQ)
          GOTO 199
        ENDIF
C
C       TEST FOR AMMONIA AND PHOS ACID ON CHAR(FACID=1 AND MW=17.0 )
        IF ((MW.EQ.17.0).AND.(FACID.EQ.1)) THEN
          CALL RACCH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR)
          GOTO 199
        ENDIF
C
C       TEST FOR MOL VOL=0 (NO CHAR REMOVAL)
        IF (VMOL.EQ.0) THEN
          EFF=0
          GOTO 199
        ENDIF
C
C       CHARCOAL REMOVAL EFFICIENCY CALCULATION
C       BED LENGTH (M)-ASSUMES THIN BED
        BEDL=(BEDOD-BEDID)/2
        IF (BEDL.LT.0) BEDL=0
C       BED WGT (KG)
        BEDWGT=DENCH*.785*(BEDOD**2-BEDID**2)*CARTL
C       SUPERFICIAL BED VEL(FT/MIN)
        BEDVEL=BEDQ*.0348/((BEDOD+BEDID)*CARTL)
C       TEST FOR CIN TOO SMALL IN AVAL CALC
        IF (CIN.LT.1E-20) CIN=1E-20
        AVAL=(TCABIN/VMOL)*LOG10(VCONC/CIN)
C       ADS ZONE LENGTH FOR 90% REMOVAL (M)
        LADS=AVAL*.000275*(BEDVEL/1.3)**.8
C       GET QI(CC LIQ CONT/GM CHAR)
        CALL FQI(AVAL,QI,FACID,SOL)
C       LENGTH OF BED PREVIOUSLY USED BY CONT AT THIS C INLET (M)
        LPREV=SMR*1.0E-6*COEXIS*BEDL/(DCONT*BEDWGT*QI)
C       RATE OF BED USAGE (M BED/ MG CONT)
        LIMM=1.0E-6*COEXIS*BEDL/(DCONT*BEDWGT*QI)
C       LENGTH OF BED AVAILABLE FOR ADS ZONE AT BEGINNING OF INCR (M)
        LAVN1=BEDL-LPREV
        IF (LAVN1.LT.0) LAVN1=0
```

```
C       FIX HERE IF DESORPTION IS DESIRED
        IF (LAVN1/LADS.GT.20) THEN
          EFFAV=EMAX
        ELSE
C         INIT INCR EFF BASED ON C IN AND BED L AVAIL AT BEG OF INCR(DEC)
          EFAVN1=EMAX*(1-EXP(-2.3025851*LAVN1/LADS))
C         LOOP FOR EFFICIENCY
          EFFAV=EFAVN1
          DO  399 J=1,10,1
C           LENGTH OF BED UTILIZ IN INCR (M)
            LUTIL=CIN*BEDQ*EFFAV*(TN-TN1)*LIMM
            IF (LUTIL.GT.LAVN1) THEN
              GOTO 299
            ELSE
C             AVERAGE BED LENGTH AVAIL (M)
              LAVAV=LAVN1-LUTIL/2
              IF ((LAVAV/LADS).GE.20) THEN
                EFFAV=EMAX
                GOTO 299
              ELSE
C               AV EFF BASED ON AV BED L AVAIL (DEC)
                EFFAV=EMAX*(1-EXP(-2.3025851*LAVAV/LADS))
              ENDIF
            ENDIF
  399     CONTINUE
  299 ENDIF
C     MAX EFF BASED ON C IN AND RATE OF BED USAGE (DEC)
        EFFMAX=LAVN1/(CIN*BEDQ* (TN-TN1)*LIMM)
        IF (EFFAV.GT.EFFMAX) EFFAV=EFFMAX
        IF (EFFAV.LT.0) EFFAV=0
        IF (EFFAV.GT.EMAX) EFFAV=EMAX
C       EFF=ACTUAL EFF OUTPUT FROM SUBROUTINE
        EFF=EFFAV
C       REMOVE THIS CHECK IF DESORPTION IS ADDED
  199 IF (EFF.LT.0)  EFF=0
        IF (EFF.GT.EMAX) EFF=EMAX
        RETURN
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

C       ****************************************************
C       * SUBROUTINE RACCH    -  CALCULATES REMOVAL EFF    *
C       * BED WITH NH3 AND .2 MILLIMOLE H3PO4 ON CHAR      *
C       ****************************************************
        SUBROUTINE  RACCH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR)
C       OUTPUTS
C         EFF=OUTPUT REMOVAL EFF (DEC)
C       INPUTS
```

```
C           EMAX=MAXIMUM BED REMOVAL EFF (DEC)
C           CARTL=CARTRIDGE LENGTH (M)
C           BEDOD=BED OUTSIDE DIAMETER (M)
C          BEDID=BED INSIDE DIAMETER (M)
C           DENCH=CHARCOAL DENSITY(KG/CU M)
C          SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
C
C         CHAR USED (KG)
            CHRUSD=9.61E-6*SMR
C         CHAR BED WGT(KG)
          BEDWGT=CARTL*(BEDOD**2-BEDID**2)*.785*DENCH
          IF (CHRUSD.LT.0.8*BEDWGT) THEN
            EFF=EMAX
          ELSE
           EFF=EMAX*SIN((BEDWGT-CHRUSD)*1.57/(BEDWGT*0.2))
          ENDIF
C         PREVENTS NEGATIVE EFF FOR REACTION
C         IF (EFF.LT.0) EFF=0
          IF(EFF.GT.EMAX) EFF=EMAX
          RETURN
          END
```

```
NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
```

```
C         ******************************************************
C         * SUBROUTINE RCICH   -  CALCULATES REMOVAL EFF        *
C         *    FOR FORMALDELYDE AND CI CHAR BED                 *
C         ******************************************************
        SUBROUTINE RCICH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR,BEDQ)
C         OUTPUTS
C           EFF=OUTPUT REMOVAL EFF (DEC)
C         INPUTS
C           EMAX=MAXIMUM BED REMOVAL EFF (DEC)
C           CARTL=CARTRIDGE LENGTH (M)
C           BEDOD=BED OUTSIDE DIAMETER (M)
C          BEDID=BED INSIDE DIAMETER (M)
C           DENCH=CHARCOAL DENSITY(KG/CU M)
C          SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
C           BEDQ=BED FLOW RATE (CU M/HR)
C
          BEDWGT=CARTL*(BEDOD**2-BEDID**2)*.785*DENCH
C         PERCENT OF BED WEIGHT CONSUMED (DEC)
          PBWGT=SMR/(BEDWGT*1E6)
          IF(PBWGT.LT..0012) THEN
            EFF=1-PBWGT*83.3
          ELSE
            EFF=.9*COS(PBWGT*1.57/.05)
          ENDIF
C         BED RESIDENCE TIME (SEC)
```

```
      BREST=(BEDOD-BEDID)*CARTL*(BEDOD+BEDID)*3600/(BEDQ*1.273)
        IF(BREST.LT.0.25)THEN
          EFF=EFF*BREST/.25
        ENDIF
C       PREVENTS NEGATIVE EFF FOR REACTION
C       IF (EFF.LT.0) EFF=0
        IF(EFF.GT.EMAX) EFF=EMAX
        RETURN
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\REGEN.F Options: /C 80 /L /BY   05/21/92 13:00:29

```
C       ****************************************************************
C       *                REGENERATION SUBROUTINE-REGEN                 *
C       * FOR ALL BEDS (3-15) DETERMINES IF BED IS TO BE REGENERATED   *
C       * AT BEGINNING OF TIME INCREMENT, AND IF REGENERATION IS TO    *
C       * DURING THE ENTIRE TIME INCREMENT- IF THE BED IS TO BE        *
C       * REGENERATED THE MASSES STORED ARE SET TO ZERO, AND IF        *
C       * REGENERATION IS TO OCCUR THROUGHOUT THE TIME INCREMENT THE   *
C       * BED FLOW RATE IS SET TO ZERO; OTHERWISE IT IS SET TO THE     *
C       * ORIGINAL VALUE                                               *
C       ****************************************************************
        SUBROUTINE REGEN(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
       +CDI,NROW2,NCOL2,LIN,LIN2,IMSGDN)
        INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
        REAL DD(NROW,NCOL)
        REAL CC(NROW1,NCOL1)
        REAL CDI(NROW2,NCOL2)

C NOTE: BEFORE RUNNING THIS SUBROUTINE THE ORIGINAL FLOW RATES FROM
C       MAT DD COL 2 MUST BE SAVED IN MAT DD COL 7- AT THE END OF THE
C       TIME INCREMENT THEY MUST BE RESTORED TO COL 2
C DIRECT INPUTS:
C    TN=INCREMENT END TIME(HRS); TN1=INCR BEGINNING TIME HRS
C    DD,NROW,NCOL=NAME AND SIZE OF MAT DD
C    CC,NROW1,NCOL1=NAME AND SIZE OF MAT CC
C    CDI,NROW2,NCOL2=NAME AND SIZE OF MAT CDI
C    LIN=NO. OF CONT IN MAT CDI
C    LIN2=NO. OF DEVICES IN MAT DD

C OTHER INPUTS FROM MAT DD
C    TIR=INITIAL (FIRST) REGENERATION TIME (HRS)
C    TRCI=REGEN/CHANGEOUT INTERVAL (HRS)
C    TRD=REGENERATION DURATION (HRS)
C    DEVICE NO., TYPE, FLOW RATE, ETC

C OUTPUTS:
C    A)IF REGENERATION OCCURS AT THE BEGINNING OF ANY TIME INCREMENT
C       1)FOR ANY DEVICE WHICH IS A CHARCOAL BED
C          FOR ALL CONT 1 TO LIN IT PUTS SUM MASS REM=0 IN MAT CC
C          COL 12,15,18....48 AS APPROPRIATE FOR THAT DEVICE
C       2)FOR ANY DEVICE WHICH IS A LIOH BED
C          IT DOES 1) ABOVE, AND IN ADDITION PUTS SUM MASS REM=0 IN
C          MAT DD COL 16 FOR THAT DEVICE
C   B)IF REGENERATION IS OCCURRING THROUGHOUT THE WHOLE INTERVAL
C          IT SETS Q OF DEVICE=0;  IF REGENERATION IS NOT OCCURRING, IT
C          SETS Q=THE ORIGINAL VALUE
C
C     SUBROUTINES REQUIRED:
```

```
C          REGCHG
C
C    START LOOP FOR ALL DEVICES 3 TO 15
       DO 100 J=3,LIN2
C          IF DEVICE DOES NOT EQUAL CHARCOAL OR LIOH THEN GO TO END OF LOOP
           IF(DD(J,3).NE.3.AND.DD(J,3).NE.4.AND.DD(J,3).NE.5) GOTO 100

C      ASSIGN PROPER VARIABLES FOR DEVICE
C          DEVICE = CHARCOAL
           IF(DD(J,3).EQ.3.OR.DD(J,3).EQ.4) THEN
C            REGENERATION/CHANGEOUT INTERVAL (HRS)
             TRCI=DD(J,15)
C            REGENERATION DURATION (HRS)
             TRD=DD(J,16)
C            INITIAL(FIRST) REGENERATION
             TIR=DD(J,14)
           ENDIF

C          DEVICE = LIOH
           IF(DD(J,3).EQ.5) THEN
C            REGENERATION/CHANGEOUT INTERVAL (HRS)
             TRCI=DD(J,14)
C            REGENERATION DURATION (HRS)
             TRD=0
C            INITIAL(FIRST) REGENERATION
             TIR=DD(J,13)
           ENDIF

C      CHECK AND FIX INPUT AS REQ + PRINT WARNINGS
C          TIME INCREMENT (HRS)
             TINC=DD(1,11)
C          INITIAL TIME NOT EQUAL TO MULTIPLE OF TIME INCREMENT
             IF(AINT(TIR/TINC).NE.(TIR/TINC)) THEN
               TIR=AINT(TIR/TINC)*TINC
               OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
                 WRITE(IMSGDN,*)'INCREMENT BEGINNING TIME ',TN1,
     +'DEV NO.',(J)
                 WRITE(IMSGDN,*)'INITIAL TIME NOT = MULTIPLE OF TIME
     + INCREMENT'
                 WRITE(IMSGDN,*)'TRUNCATED TO ',TIR
               CLOSE(IMSGDN)
             ENDIF

C          REGEN/CHGOUT INTERVAL < TIME INCR OR NOT= MULTIPLE OF TIME INCR
             IF(AINT(TRCI/TINC).NE.(TRCI/TINC)) THEN
               TRCI=AINT(TRCI/TINC)*TINC
               OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
                 WRITE(IMSGDN,*)'INCREMENT BEGINNING TIME ',TN1,
     +'DEV NO.',(J)
                 WRITE(IMSGDN,*)'REGEN/CHG TIME NOT = MULTIPLE OF TIME
```

```
     + INCREMENT'
                WRITE(IMSGDN,*)'TRUNCATED TO ',TRCI
              CLOSE(IMSGDN)
            ENDIF

C         IF REGEN/CHGOUT INTERVAL <= 0 THEN GOTO END OF LOOP FOR CONT
          IF (TRCI.LE.0) THEN
            GOTO 100
        ENDIF

C        REGEN DURATION < OR NOT = MULTIPLE OF TIME INCREMENT
            IF(AINT(TRD/TINC).NE.(TRD/TINC)) THEN
              TRD=AINT(TRD/TINC)*TINC
              OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
                WRITE(IMSGDN,*)'INCREMENT BEGINNING TIME ',TN1,
     +'DEV NO.',(J)
              WRITE(IMSGDN,*)'REG DURATION NOT = MULTIPLE OF TIME
     + INCREMENT'
                WRITE(IMSGDN,*)'TRUNCATED TO ',TRD
               CLOSE(IMSGDN)
            ENDIF

C        REGENERATION DURATION > REGEN/CHG INTERVAL
            IF(TRD.GT.TRCI) THEN
              TRD=TRCI
              OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
                WRITE(IMSGDN,*)'INCREMENT BEGINNING TIME ',TN1,
     +'DEV NO.',(J)
              WRITE(IMSGDN,*)'REGEN DURATION > REGEN/CHG INTERVAL'
                WRITE(IMSGDN,*)'TRUNCATED TO ',TRD
              CLOSE(IMSGDN)
            ENDIF

C      CHECK TO SEE IF REGENERATION OCCURS AT BEGINNING OF TIME INCR,
C        AND IF REGEN OCCURS THROUGHOUT WHOLE TIME INCREMENT

       CALL REGCHG(TN1,TRCI,TRD,TIR,TINC,IRBFLG,IRTFLG)
C        REGENERATION OCCURS AT BEGINNING OF INCREMENT
            IF (IRBFLG.EQ.1) THEN
C           PUT SUM MASS REM =0 IN MAT CC FOR THIS DEVICE
C             START LOOP FOR ALL CONT FOR THIS DEVICE
              K=J*3+3
              DO 101 I=1,LIN
                CC(I,K)=0
  101         CONTINUE

C           IF DEVICE = LIOH BED PUT SUM MASS=0 IN DD(J,16)
              IF (DD(J,3).EQ.5) THEN
                DD(J,16)=0
              ENDIF
```

```
            ENDIF

C        REGENERATION OCCURS THROUGHOUT ENTIRE INCREMENT
            IF (IRTFLG.EQ.1) THEN
C             SET DEVICE Q=0
              DD(J,2)=0
            ELSE
C             SET DEVICE Q= ORIGINAL VALUE
              DD(J,2)=DD(J,7)
            ENDIF

C   END OF J LOOP FOR EACH DEVICE
  100 CONTINUE

      RETURN
      END
```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```

C       *****************************************************************
C       *         AUXILIARY REGENERATION SUBROUTINE-REGCHG              *
C       *  DETERMINES IF REGEN/CHANGEOUT IS TO OCCUR AT BEGINNING OF    *
C       *  TIME INCREMENT-ALSO DETERMINES IF REGENERATION IS OCCURRING  *
C       *  THROUGHOUT THE TIME INCREMENT                                *
C       *****************************************************************
         SUBROUTINE REGCHG(TN1,TRCI,TRD,TIR,TINC,IRBFLG,IRTFLG)
C
C INPUTS:
C    TN1=INCREMENT INITIAL TIME (HRS)
C    TRCI=CHANGEOUT/REGENERATION INTERVAL (HRS)
C    TRD=REGENERATION DURATION (HRS)
C    TIR=INITIAL (FIRST) REGENERATION TIME (HRS)
C    TINC=TIME INCREMENT (HRS)
C
C OUTPUTS:
C    REGENERATION OCCURS AT BEGINNING OF TIME INCREMENT (Y OR N)
C       (IRBFLG=1 FOR Y & 0 FOR N)
C   REGENERATION IS OCCURRING THROUGHOUT THE WHOLE INCREMENT (Y OR N)
C       (IRTFLG=1 FOR Y & 0 FOR N)
C

C       REGENERATION OCCURS AT BEGINNING OF TIME INCREMENT

        IF (TN1.EQ.0) GOTO 10
        IF(TN1.LT.TIR) GOTO 10
        IF(TRCI.LE.0) GOTO 10
       IF(AINT((TN1-TIR)/TRCI).EQ.((TN1-TIR)/TRCI)) THEN
```

```
C           REGENERATION OCCURS
            IRBFLG=1
            GO TO 20
          ENDIF
C         NO REGENERATION OCCURS
   010 IRBFLG=0
   020 CONTINUE

C      REGENERATION OCCURRING THROUGHOUT ENTIRE TIME INCREMENT

          IF(TRCI.LE.0) GOTO 30
          IF((TRD.LE.0).OR.(TN1.LT.TIR)) GOTO 30
          IF(TN1.GE.AINT((TN1-TIR)/TRCI)*TRCI+TIR+TRD) THEN
            GOTO 30
          ELSE
C           REGENERATION OCCURS
            IRTFLG=1
            GOTO 40
          ENDIF

C         REGEN DOESN'T OCCUR
   030 IRTFLG=0

   040 CONTINUE
          RETURN
          END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\RINCDD. Options: /C 80 /L /BY 05/21/92 13:00:48

```
C       ****************************************************************
C       *                   SUBROUTINE RINCDD                          *
C       *        SUBROUTINE TO OPERATE ON INCREMENT DEPENDENT DATA     *
C       *  READS DATA FROM MAT TT AND PUT IT IN THE PROPER PLACES IN   *
C       *  MAT CDI OR MAT DD - USED AT THE BEGINNING OF EACH TIME INCR *
C       ****************************************************************

        SUBROUTINE RINCDD(I,TN,TN1,DD,NROW,NCOL,LIN2,
     +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,LIN,TT,NTTROW,NTTCOL,LIN1)
        INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,NTTROW,NTTCOL
        REAL DD(NROW,NCOL)
        REAL CC(NROW1,NCOL1)
        REAL CDI(NROW2,NCOL2)
        REAL TT(NTTROW,NTTCOL)

C SUBROUTINES REQUIRED:
C       NONE
C       DD,NROW,NCOL,LIN2=NAME,DIM & NO DEV IN MAT DD
C       CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C       CDI,NROW2,NCOL,LIN=NAME,DIM & NO CONT IN MAT CDI
C       TT,NTTROW,NTTCOL,LIN1=NAME,DIM & NO ITEMS IN MAT TT

        IF (LIN1.EQ.0) GOTO 999
C       BEGIN LOOP FOR ALL LINES IN MAT TT
        DO 100 K=1,LIN1

C       CHECK FOR TIME >= TN1 AND < TN
        IF((TT(K,1).LT.TN1).OR.(TT(K,1).GE.TN)) THEN
          GO TO 100
        ENDIF

C       IDENTIFY VARIABLES
        ICONTN=NINT(TT(K,2))
        GENRT=TT(K,3)
        IDEVNO=NINT(TT(K,4))
        DEVQ=TT(K,5)
        ICOLNO=TT(K,6)
        VAL=TT(K,7)

C       ICONTN=INTEGER CONTAMINANT NO. - TT(K,2)
C       GENRT=CONT GENERATION RATE (MG/HR) - TT(K,3)
C       IDEVNO=INTEGER DEVICE NUMBER - TT(K,4)
C       DEVQ=DEVICE FLOW RATE (CU M/HR) - TT(K,5)
C       ICOLNO=INTEGER COLUMN NUMBER IN MAT DD - TT(K,6)
C       VAL=NEW VALUE IN MAT DD - TT(K,7)

C       CASE NO. 1 - CHANGE CONTAMINANT GENERATION RATE
```

```

      IF((ICONTN.GT.0).AND.(ICONTN.LE.LIN)) THEN
        IF(IDEVNO.EQ.1) THEN
          CDI(ICONTN,1)=GENRT
        ELSEIF ((IDEVNO.GE.3).AND.(IDEVNO.LE.LIN2)) THEN
          CDI(ICONTN,(7+IDEVNO))=GENRT
        ENDIF
        GOTO 100
      ENDIF

C     CASE 2 - CHANGE DEVICE FLOW OR OTHER DD DATA
C     THIS CASE WORKS ONLY IF ANY CONT NO. <=0
C     MUST USE -1 FOR ANY Q OR NEW VALUE NOT TO BE CHANGED
C     MAT DD COL NO. <0 ALSO STOPS NEW VALUE FROM BEING CHANGED

      IF(ICONTN.LE.0) THEN
        IF((IDEVNO.GE.1).AND.(IDEVNO.LE.LIN2)) THEN
          IF(DEVQ.GE.0) THEN
C           CHANGE DEVICE FLOW IN MAT DD
            DD(IDEVNO,2)=DEVQ
          ENDIF
          IF((ICOLNO.GE.1).AND.(ICOLNO.LE.16)) THEN
            IF (VAL.GE.0) THEN
C             CHANGE VALUE IN MAT DD
             DD(IDEVNO,ICOLNO)=VAL
            ENDIF
          ENDIF
        ENDIF
      ENDIF
  100 CONTINUE
  999 RETURN
      END
```

```
NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\RRIN.FO Options: /C 80 /L /BY     05/21/92 13:00:57

C       *******************************************************************
C       *           SUBROUTINE   RRIN                                      *
C       *     SUBROUTINE TO READ REAL DATA INTO MAT XX(ROW,COL)            *
C       *     RETURNS NUMBER OF LINES OF DATA READ FROM FILE               *
C       *     READS FROM COL 1 TO COL LSTCOL                               *
C       *******************************************************************
C   NOTE: INPUT NUMBERS MUST BE SEPARATED BY BLANKS
        SUBROUTINE RRIN(XX,NROW,NCOL,LSTCOL,LIN)
        INTEGER NROW,NCOL,IOVAL,LSTCOL,LIN
        CHARACTER FNAME*24
        REAL XX(NROW,NCOL)
        IF(LSTCOL.GT.NCOL) LSTCOL=NCOL
  010 READ(*,'(A)') FNAME
        OPEN(1,FILE=FNAME,STATUS='OLD',IOSTAT=IOVAL)
        IF(IOVAL.NE.0) GOTO 900
        LIN=0
        DO 100 I=1,NROW
        READ(1,*,IOSTAT=IOVAL,END=500,ERR=900 ) (XX(I,J),J=1,LSTCOL)
        LIN=LIN+1
  100 CONTINUE
  500 WRITE(*,'(A)') ' DONE WITH FILE INPUT'
        WRITE (*,*) ' '
        CLOSE (1)
        GOTO 990
  900 WRITE(*,*)'IOERROR= ',IOVAL
        CLOSE (1)
        WRITE(*,*) 'WHAT IS THE INPUT FILE NAME? '
        GOTO 10
  990 RETURN
C       ************ END OF SUBROUTINE RRIN ******************************
        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\RROUT.F Options: /C 80 /L /BY     05/21/92 13:01:03

C         *****************************************************************
C         *          SUBROUTINE  RROUT                                    *
C         *     SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER           *
C         *     WRITES REAL DATA FROM MAT XX(ROW,COL)                     *
C         *     WRITES FROM FSTCOL TO LSTCOL                              *
C         *****************************************************************
          SUBROUTINE RROUT(XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,IMSGDN,FNAME,
        +  IDEVNO,IOVAL)
          INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,IDEVNO
          CHARACTER FNAME*24,DES*1
          REAL XX(NROW,NCOL)
          IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
          IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
          IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
C 010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
C         WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END  '
C         CLOSE(IMSGDN)
C         READ(*,'(A)') FNAME
C         QUIT IF FNAME=END
C         IF(FNAME.EQ.'END') GOTO 990
C         IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
C         OPEN(1,FILE=FNAME, IOSTAT=IOVAL)
          IF(IOVAL.NE.0) GOTO 900
          DO 110 I=1,LIN
C           WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
           WRITE(IDEVNO,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
    070    FORMAT(1X,7G11.4)
C           WRITE(1,*,IOSTAT=IOVAL,ERR=900)
            WRITE(IDEVNO,*,IOSTAT=IOVAL,ERR=900)
    110 CONTINUE
C         CLOSE (1)
          GOTO 990
    900 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
          WRITE(IMSGDN,*)'IOERROR= ',IOVAL
          CLOSE(IMSGDN)
C         CLOSE (1)
          CLOSE (IDEVNO)
C         GOTO 10
    990 RETURN
C         ***************  END OF SUBROUTINE RROUT  ***********************
          END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\RROUT2. Options: /C 80 /L /BY     05/21/92 13:01:08

  C       ********************************************************************
  C       *         SUBROUTINE  RROUT2                                       *
  C       *     SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER              *
  C       *     WRITES REAL DATA FROM MAT XX(ROW,COL)                        *
  C       *     WRITES FROM FSTCOL TO LSTCOL                                 *
  C       ********************************************************************
          SUBROUTINE RROUT2 (XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,IMSGDN)
          INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,IDEVNO
          CHARACTER FNAME*24,DES*1
          REAL XX(NROW,NCOL)
          IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
          IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
          IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
      010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
          WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END  '
          CLOSE(IMSGDN)
          READ(*,'(A)') FNAME
          QUIT IF FNAME=END
          IF(FNAME.EQ.'END') GOTO 990
          IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
          OPEN(1,FILE=FNAME, IOSTAT=IOVAL)
          IF(IOVAL.NE.0) GOTO 900
          DO 110 I=1,LIN
           WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
      070  FORMAT(1X,7G11.4)
            WRITE(1,*,IOSTAT=IOVAL,ERR=900)
      110 CONTINUE
          CLOSE (1)
          GOTO 990
      900 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
          WRITE(IMSGDN,*)'IOERROR= ',IOVAL
          CLOSE(IMSGDN)
          CLOSE (1)
          GOTO 10
      990 RETURN
  C       ***************  END OF SUBROUTINE RROUT  ************************
          END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\SLIOH.F Options: /C 80 /L /BY   05/21/92 13:01:15

```
C       ****************************************************************
C       *                      SUBROUTINE-SLIOH                        *
C       * SUM LIOH USED IN TIME INCREMENT FOR EACH BED ONE AT A TIME   *
C       * AND FOR ALL CONTAMINANTS FOR EACH BED                        *
C       ****************************************************************

        SUBROUTINE SLIOH(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +CDI,NROW2,NCOL2,LIN,LIN2)
        INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
        REAL DD(NROW,NCOL)
        REAL CC(NROW1,NCOL1)
        REAL CDI(NROW2,NCOL2)
C
C SUBROUTINES REQUIRED: NONE
C DIRECT INPUTS:
C    TN=INCREMENT END TIME(HRS); TN1=INCR BEGINNING TIME HRS
C    DD,NROW,NCOL=NAME AND SIZE OF MAT DD
C    CC,NROW1,NCOL1=NAME AND SIZE OF MAT CC
C    CDI,NROW2,NCOL2=NAME AND SIZE OF MAT CDI
C    LIN=NO. OF CONT IN MAT CDI
C    LIN2=NO. OF DEVICES IN MAT DD

C OTHER INPUTS FROM MAT DD
C    DD(J,3)=DEVICE NUMBER
C    DD(J,16)=AMT OF LIOH PREVIOUSLY USED BY DEVICE
C    CDI(I,17)=LB LIOH UTIL/LB CONT ADSORBED IN BED (FOR ONE CONT)

C OUTPUTS (STORED IN MAT DD):
C    DD(J,16)=AMOUNT OF LIOH UTILIZED BY DEVICE THROUGH THE END OF
C    THIS TIME INCREMENT
C    DD(J,15)=RATE OF LIOH USAGE FOR DEVICE
C
        K=11

C       START LOOP FOR ALL DEVICES 3 TO 15
        DO 100 J=3,LIN2
C         CHECK FOR DEVICE = LIOH BED
          IF (DD(J,3).EQ.5) THEN
C           RATE OF LIOH UTILIZATION (KG/HR)
            RWUTLI=0
C           BEGIN LOOP FOR ALL CONTAMINANTS
            DO 110 I=1,LIN
              RWUTLI=RWUTLI+CC(I,K)*CDI(I,7)*1E-6
  110      CONTINUE
C           STORE RATE OF LIOH UTILIZATION IN MAT DD FOR THIS DEVICE
             DD(J,15)=RWUTLI
C          UPDATE AMOUNT OF LIOH UTIL THROUGH THE END OF TIME INCR(KG)
```

```
                  DD(J,16)=DD(J,16)+RWUTLI*(TN-TN1)
                ENDIF
                K=K+3
C           END J LOOP
    100 CONTINUE
            RETURN
            END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS    IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS    IN COMPILATION : 0
```

APPENDIX B

TOXIC HAZARD INDEX DESCRIPTION

The toxic hazard index, or **T**-value, is the method used by toxicologists to assess the acceptability of an atmosphere containing a mixture of contaminants. This approach is derived from the American Conference of Governmental Industrial Hygienists guidelines for setting threshold limit values for contaminant mixtures. Since the effects on humans of many atmospheric contaminants are considered to be additive this mixture approach is applied to sixteen contaminant groups. The groups considered in the T-value calculation used in the TCCS Computer Program are the following:

1. Alcohols
2. Aldehydes
3. Aromatic hydrocarbons
4. Esters
5. Ethers
6. Chlorocarbons
7. Chlorofluorocarbons
8. Fluorocarbons
9. Hydrocarbons
10. Inorganic acids
11. Ketones
12. Mercaptans and sulfides
13. Nitrogen oxides
14. Organic acids
15. Organic nitrogens
16. Miscellaneous

The group numbers used in the computer program output correspond to the above group listing.

The **T**-value is calculated for each group by calculating the sum of the ratios of the contaminants' concentrations to their maximum allowable concentration while the overall **T**-value is the sum of the group T-values for the alcohols, aldehydes, aromatic hydrocarbons, esters, ethers, hydrocarbons, inorganic acids, ketones, nitrogen oxides, organic acids, and miscellaneous groups. These calculations are conducted according to the following equations:

$$T_{group} = \sum C_c/C_m \quad \text{(B1)}$$

$$T_{overall} = \sum T_{group} \quad , \quad \text{(B2)}$$

where C_c is the contaminant concentration in the atmosphere in mg/m^3 and C_m is the maximum allowable concentration in the atmosphere in mg/m^3.

The criteria for acceptability are the following:

1. The T-value for each group must be less than one
2. The overall T-value must be less than one

If either of these criteria are exceeded, the atmosphere is considered unacceptable.

REFERENCES

1. "Design and Fabrication of a Trace Contaminant Removal System for Apollo," Phase I Report, NASA Contract NAS9-3415. Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, March 15, 1965.

2. Ibid., Phase II Report. November 23, 1965.

3. Ibid., Phase III Report. April 14, 1966.

4. Olcott. T.M.: "Study and Preliminary Design of an Isotope-Heated Catalytic Oxidizer System." NASA CR-66346, NASA Contract NAS 1-6256. Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, 1969.

5. Robell, A.J., Arnold, C.R., Wheeler, A., Kersels, G.J., and Merrill, R.P.: "Trace Contaminant Adsorption and Sorbent Regeneration." NASA CR-1582. Lockheed Missiles and Space Co., Palo Alto, CA, September 1970.

6. Ibid., pp. 32–70.

7. Olcott, T.M.: "Development of a Sorber Trace Contaminant Control System Including Pre- and Post-Sorbers for a Catalytic Oxidizer." NASA CR-2027. Lockheed Missiles and Space Co., Sunnyvale, CA, May 1972, pp. D1–D18.

8. Olcott, T.M.: "Design, Fabrication, and Test of a Trace Contaminant Control System." LMSC-D462467. Biotechnology Organization Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, November 28, 1975.

9. Jagow, R.B., Jaffe, R.J., and Lamparter, R.A.: "Development of a Computer Program for Spacelab Contaminant Control Analysis." LMSC-D556710, NASA Contract NAS 8-31551. Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, January 15, 1977.

10. Lamparter, R.A., and Yee, R.Q.: "A Generalized Computer Program for Trace Contaminant Control." LMSC-D564045, NASA Contract NAS 8-32400. Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, September 1, 1977.

11. "Preliminary Documentation for Space Station Gaseous Contaminant Control Program." NASA Contract NAS8-36406. Bioastronautics Division, Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, September 5, 1986.

12. "Flammability, Odor, and Offgassing Requirements and Test Procedures for Material in Environments that Support Combustion." NASA Handbook NHB 8060.1B. National Aeronautics and Space Administration Office of Space Transportation Systems, Washington, DC, September 1981, pp. D10–D11.

13. "Preliminary Documentation for Space Station Gaseous Contaminant Control Program." NASA Contract NAS8-36406. Bioastronautics Division, Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, September 5, 1986, pp. 33–61.

14. Olcott, T.M.: "Development of a Sorber Trace Contaminant Control System Including Pre- and Post-Sorbers for a Catalytic Oxidizer." NASA CR-2027, Lockheed Missiles and Space Co., Sunnyvale, CA, May 1972, p. 26.

15. Ibid., p. 24

16. Robell, A.J., Arnold, C.R., Wheeler, A., Kersels, G.J., and Merrill, R.P.: "Trace Contaminant Adsorption and Sorbent Regeneration." NASA CR-1582. Lockheed Missiles and Space Co., Palo Alto, CA, September 1970.

17. Olcott, T.M.: "Development of a Sorber Trace Contaminant Control System Including Pre- and Post-Sorbers for a Catalytic Oxidizer." NASA CR-2027, Lockheed Missiles and Space Co., Sunnyvale, CA, May 1972.

18. Ibid., p. 31.

19. "Preliminary Documentation for Space Station Gaseous Contaminant Control Program." NASA Contract NAS8-36406, Bioastronautics Division, Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, September 5, 1986, p. 59.

20. Jagow, R.B., Katan, T., Lamparter, R.A., Dibble, K.P., Masi, O.H., Richardi, C.L., and Kawasaki, E.H.: "Investigation and Testing of Low Temperature Carbon Monoxide Catalysts." LMSC-D556669, NASA Contract NAS 8-31960. Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, January 20, 1977, p. 29.

21. Kissinger, L.D., and Verostko, C.E.: "Inherent Ammonia Removal Potential of the Space Shuttle Orbiter Condensing Heat Exchanger." JSC-08797, NASA Lyndon B. Johnson Space Center Crew Systems Division, Houston, TX, August 23, 1976, p. 10.

22. Ibid., p.14.

APPROVAL

COMPUTERIZED ATMOSPHERIC TRACE CONTAMINANT CONTROL SIMULATION FOR MANNED SPACECRAFT

By J.L. Perry

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

J.C. BLAIR
Director, Structures and Dynamics Laboratory

☆ U.S. GOVERNMENT PRINTING OFFICE 1993—733-050/80086