TRACE CONTAMINANT CONTROL SIMULATION COMPUTER PROGRAM—VERSION 8.1

By J.L. Perry

Structures and Dynamics Laboratory
Science and Engineering Directorate

May 1994
The Trace Contaminant Control Simulation computer program is a tool for assessing the performance of various process technologies for removing trace chemical contamination from a spacecraft cabin atmosphere. Included in the simulation are chemical and physical adsorption by activated charcoal, chemical adsorption by lithium hydroxide, absorption by humidity condensate, and low- and high-temperature catalytic oxidation. Means are provided for simulating regenerable as well as nonregenerable systems. The program provides an overall mass balance of chemical contaminants in a spacecraft cabin given specified generation rates. Removal rates are based on device flow rates specified by the user and calculated removal efficiencies based on cabin concentration and removal technology experimental data. Versions 1.0 through 8.0 are documented in NASA TM-108409. TM-108409 also contains a source file listing for version 8.0. Changes to version 8.0 are documented in this technical memorandum and a source file listing for the modified version, version 8.1, is provided. Detailed descriptions for the computer program subprograms are extracted from TM-108409 and modified as necessary to reflect version 8.1. Version 8.1 supersedes version 8.0. Information on a separate user’s guide is available from the author.
ACKNOWLEDGMENTS

Version 8.1 of the Trace Contaminant Control Simulation computer program was made possible through the efforts of Mark Leban of Lockheed Missiles and Space Co., who provided the latest information on activated charcoal loading characteristics and modified the code accordingly. Also, he added additional code to account for poisoning of high temperature oxidation catalysts by selected trace contaminants. His expertise and professionalism in making these modifications have been greatly appreciated.
# TABLE OF CONTENTS

**INTRODUCTION** ........................................................................................................................................... 1

**VERSION 8.0 MODIFICATIONS** ...................................................................................................................... 1

**DETAILED COMPUTER PROGRAM DESCRIPTION** .......................................................................................... 3

- Program Editing, Compiling, and Linking ........................................................................................................... 3
- Main Program ....................................................................................................................................................... 3
- Calculation Loop Subroutines
  - Subroutine CAFILL ........................................................................................................................................... 5
  - Subroutine RAFILL .......................................................................................................................................... 5
  - Subroutine PCSET ......................................................................................................................................... 7
  - Subroutine PRAFIL ....................................................................................................................................... 8
  - Subroutine CNRSUB ................................................................................................................................... 8
  - Subroutine MASBAL ................................................................................................................................... 9
  - Subroutine CALCM ................................................................................................................................... 9
  - Subroutine LDIGEN .................................................................................................................................... 12
  - Subroutine PCAVCF ................................................................................................................................... 12
  - Subroutine RINCDD ................................................................................................................................... 12
  - Subroutine REGEN .................................................................................................................................... 15
  - Subroutine PREDCT ................................................................................................................................... 15
  - Subroutine LODEFF ................................................................................................................................... 17
  - Subroutine CONVRG ................................................................................................................................ 17
  - Subroutine SLIOH ....................................................................................................................................... 19

- Contaminant Removal Device Calculation Subroutines ...................................................................................... 19
  - Subroutine ACHBD ....................................................................................................................................... 19
  - Subroutine RCHBD ....................................................................................................................................... 24
  - Subroutine ALIOH ....................................................................................................................................... 24
  - Subroutine COOXID ................................................................................................................................... 24
  - Subroutine CATBNR ................................................................................................................................... 24
  - Subroutine CONDHX .................................................................................................................................. 26

- Data Input and Output Subroutines .................................................................................................................... 28
  - Subroutine CRIN .......................................................................................................................................... 28
  - Subroutine RRIN .......................................................................................................................................... 28
  - Subroutine CROUT2 .................................................................................................................................... 28
  - Subroutine RROUT2 .................................................................................................................................... 28
  - Subroutine CROUT .................................................................................................................................... 28
  - Subroutine RROUT .................................................................................................................................... 28
  - Subroutine DATOUT ................................................................................................................................... 28
  - Subroutine GROUP .................................................................................................................................... 28
  - Subroutine PRFANS .................................................................................................................................... 29
  - Subroutine HEADGS ................................................................................................................................... 29
TABLE OF CONTENTS (Continued)

APPENDIX A – TCCS COMPUTER PROGRAM, VERSION 8.1, FORTRAN CODE LISTING ................................................................. 31

APPENDIX B – TOXIC HAZARD INDEX DESCRIPTION ......................................................................................... 118

REFERENCES ................................................................................................................................. 122
<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Main program block flow diagram</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>Subroutine PCSET block flow diagram</td>
<td>7</td>
</tr>
<tr>
<td>3.</td>
<td>Subroutine CNRSUB block flow diagram</td>
<td>8</td>
</tr>
<tr>
<td>4.</td>
<td>Subroutine MASBAL block flow diagram</td>
<td>10</td>
</tr>
<tr>
<td>5.</td>
<td>Subroutine CALCM block flow diagram</td>
<td>11</td>
</tr>
<tr>
<td>6.</td>
<td>Subroutine RINCD block flow diagram</td>
<td>13</td>
</tr>
<tr>
<td>7.</td>
<td>Subroutine REGEN block flow diagram</td>
<td>14</td>
</tr>
<tr>
<td>8.</td>
<td>Subroutine MCALC block flow diagram</td>
<td>16</td>
</tr>
<tr>
<td>9.</td>
<td>Subroutine PREDCT block flow diagram</td>
<td>17</td>
</tr>
<tr>
<td>10.</td>
<td>Subroutine CONVRG block flow diagram</td>
<td>18</td>
</tr>
<tr>
<td>11.</td>
<td>Charcoal saturation and adsorption zone distribution</td>
<td>20</td>
</tr>
<tr>
<td>12.</td>
<td>Potential plot for type BD granular activated charcoal</td>
<td>21</td>
</tr>
<tr>
<td>13.</td>
<td>Potential plot showing relative humidity effects</td>
<td>22</td>
</tr>
<tr>
<td>14.</td>
<td>Adsorption zone length as a function of the potential factor</td>
<td>23</td>
</tr>
<tr>
<td>15.</td>
<td>Noble metal CO catalyst performance</td>
<td>25</td>
</tr>
<tr>
<td>16.</td>
<td>Absorption of contaminants by humidity condensate</td>
<td>27</td>
</tr>
<tr>
<td>17.</td>
<td>Ammonia solubility in water at varying carbon dioxide concentration</td>
<td>27</td>
</tr>
</tbody>
</table>
INTRODUCTION

The Trace Contaminant Control Simulation (TCCS) computer program development began with the efforts of Olcott which were documented in 1972. Since then, improvements in the user interface and more up-to-date information on activated charcoal loading characteristics and high temperature oxidation catalysts have become available. The progression of the program from version 1.0 through 8.0 and acknowledgments for its development are documented by reference 1. The descriptive material on the computer program subroutines has been extracted in its entirety from reference 1; however, some modifications have been made where necessary.

VERSION 8.0 MODIFICATIONS

The TCCS computer program version 8.0 which is documented in reference 1 has been modified as part of the International Space Station (ISS) design process. These modifications were made initially by personnel at Lockheed Missiles and Space Co. in Sunnyvale, CA, as part of the design activities for the ISS trace contaminant control subassembly. During these activities, new information on activated charcoal loading capacity, influence of relative humidity on charcoal loading capacity, and preliminary information on poisoning characteristics of high temperature oxidation catalysts was obtained. This information was assessed by both Lockheed and NASA Marshall Space Flight Center personnel and integrated into the TCCS computer program as version 8.1. These modifications were officially accepted by NASA in March 1994. Specific details on the modifications can be found in reference 2. A listing of the source files for version 8.1 is contained in appendix A. Version 8.1 supersedes all other versions of the program.

A brief summary of changes to version 8.0 which have resulted in the new version 8.1 are the following:

1. The main program, MAIN.FOR, was modified in the following ways:
   a. Matrix TT was increased from 300 by 7 to 750 by 7 to accommodate larger time-dependent data files
   b. Code which sets the time increment size was moved to precede the code which checks for changes in the basic time increment
   c. The code was modified to accept changes in the basic time increment size as long as the change occurs between the beginning and end of the current time step rather than at the beginning of the time step only
(d) Changes to the cabin volume now result in a recalculation of all contaminant concentrations.

(e) Mission duration output formats have been changed to accommodate number sizes up to five digits.

2. Testing of Barnebey-Sutcliffe types AC and 3032 activated carbon with and without 10 weight percent phosphoric acid impregnation has resulted in new charcoal capacity equations. The programs ACHBD.FOR and RCHBD.FOR were modified in the following ways:

(a) Capacity for water soluble contaminants, those with a Henry's Law constant between 0 and 5, was determined to be a function of the adsorption potential factor, A, only and not a function of relative humidity. Activated carbon capacity for this case is expressed by the following equations:

\[ q = 2.1e^{-0.31A} \quad \text{for } A > 8 \]  
\[ q = 0.5 - 0.0405A \quad \text{for } A \leq 8 \]  

(b) Insoluble contaminants were found to be a function of adsorption potential factor, A, and relative humidity, \( H_R \). Activated carbon capacity for this case is expressed by the following equations:

\[ q = (\text{9.6} \times 10^{-5})H_R^2 - (1.88 \times 10^{-7})H_R - 2.11 \times e^{-0.31A} \quad \text{for } A > 8; H_R \leq 50 \text{ percent} \]  
\[ q = (\text{9.6} \times 10^{-5})H_R^2 - (1.88 \times 10^{-7})H_R - 2.11 \times e^{-0.31A} \quad \text{for } A > 8; H_R > 50 \text{ percent} \]

(c) The programs were modified to read cabin percent relative humidity from the device definition matrix, DD, row 1 column 14.

(d) Carbon chemisorption capacity for ammonia at its spacecraft maximum allowable concentration (SMAC) was changed to 0.0061 grams of ammonia per gram of carbon to reflect the latest phosphoric acid impregnated charcoal performance data. The previous number was based on theoretical estimates rather than experimental results.

3. Subprogram CATBNR.FOR was modified to reflect poisoning of 0.5-percent palladium on alumina catalyst by halocarbons and sulfide compounds. The result of this poisoning is a decrease in the removal efficiency, \( \eta \), for methane. The efficiency calculation was modified to account for the total mass of halocarbon and sulfide compounds, \( P \), in milligrams by using the following equations:

\[ \eta = 97.506 \times 10^{-0.00010507}P \quad \text{for } P \leq 5,500 \text{ mg} \]  
\[ \eta = 31.453 - 1.151 \times 10^{-3}P + 1.9045 \times 10^{-8}P^2 - 1.0389 \times 10^{-13}P^3 \quad \text{for } P > 5,500 \text{ mg} \]
4. The following changes were made to subprogram CNRSUB.FOR:

(a) Modifications were made to allow transferring the relative humidity value to ACHBD.FOR and RCHBD.FOR

(b) Partial catalytic oxidizer efficiency restoration for methane removal is set to coincide with axial and radial charcoal bed regeneration

(c) Cumulative masses of halocarbons and sulfides removed by the catalytic oxidizer are transferred to CATBNR.FOR. The cumulative mass is reinitialized at charcoal bed regeneration.

(d) Code was added to allow for reinitialization of methane oxidation efficiency in CATBNR.FOR if an upstream adsorption device is regenerated.

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 Co, 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.3

Program Editing, Compiling, and Linking

This program was edited, compiled, and linked using the Ryan-McFarland RM/FORTRAN™ version 2.42 which include 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 1. 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 are 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 $1 \times 10^{-20}$ 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 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 RINCDD 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 to set the 
adsorbed contaminant masses and device flow rates to zero if necessary. The main calculation loop sub-
routine, 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 lithium hydroxide (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 for-
matted 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 termi-
nated, 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. Table 1 lists the subroutines as they are called and provides a brief 
description each subroutine’s purpose. Block flow diagrams are provided for the most significant sub-
routines.

**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.
Table 1. TCCS computer program subroutine listing and description.

<table>
<thead>
<tr>
<th>Subrouting Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>MAIN PROGRAM</td>
</tr>
<tr>
<td></td>
<td>ZERO MATRIX NN</td>
</tr>
<tr>
<td></td>
<td>ZERO MATRICES CDI, CC, DD, TT</td>
</tr>
<tr>
<td></td>
<td>INPUT DATA INTO NN AND CDI</td>
</tr>
<tr>
<td></td>
<td>INPUT FROM FILE TO DD AND TT</td>
</tr>
<tr>
<td></td>
<td>PRINT DATA FROM NN AND CDI</td>
</tr>
<tr>
<td></td>
<td>PRINT DATA FROM DD AND TT</td>
</tr>
<tr>
<td></td>
<td>PRECALCULATION SETUP FOR ALL CONT</td>
</tr>
<tr>
<td></td>
<td>ZERO MATRIX DD COLUMNS 17-21</td>
</tr>
<tr>
<td></td>
<td>CALCULATE REMOVAL EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>AXIAL CHARCOAL BED EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>RADIAL CHARCOAL BED EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>AXIAL LiOH BED EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>CO OXIDIZER EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>CATALYTIC OXIDIZER EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>CONDENSATE EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>CONTAMINANT MATERIAL BALANCE</td>
</tr>
<tr>
<td></td>
<td>SUM OF MASS REMOVED BY DEVICES</td>
</tr>
<tr>
<td></td>
<td>LOAD GENERATION INTO DD COL 19</td>
</tr>
<tr>
<td></td>
<td>CALCULATE FINAL AND AVERAGE CONC</td>
</tr>
<tr>
<td></td>
<td>READ INCREMENT DEPENDENT DATA</td>
</tr>
<tr>
<td></td>
<td>CALCULATE REGENERATION/CHANGEOUT</td>
</tr>
<tr>
<td></td>
<td>MAIN CALCULATION ROUTINE</td>
</tr>
<tr>
<td></td>
<td>CALCULATE PREDICTED AVERAGE CONC</td>
</tr>
<tr>
<td></td>
<td>ZERO MATRIX DD COLUMNS 17-21</td>
</tr>
<tr>
<td></td>
<td>PUT LAST INCREMENT EFFICIENCY IN DD</td>
</tr>
<tr>
<td></td>
<td>MASS BALANCE ROUTINE</td>
</tr>
<tr>
<td></td>
<td>SUM OF MASS REMOVED BY DEVICES</td>
</tr>
<tr>
<td></td>
<td>LOAD GENERATION INTO DD COL 19</td>
</tr>
<tr>
<td></td>
<td>CALCULATE FINAL AND AVERAGE CONC</td>
</tr>
<tr>
<td></td>
<td>SOLVE FOR NEW REMOVAL EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>ZERO MATRIX DD COLUMNS 17-21</td>
</tr>
<tr>
<td></td>
<td>CALCULATE REMOVAL EFFICIENCIES</td>
</tr>
<tr>
<td></td>
<td>AXIAL CHARCOAL BED EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>RADIAL CHARCOAL BED EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>AXIAL LiOH BED EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>CO OXIDIZER EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>CATALYTIC OXIDIZER EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>CONDENSATE EFFICIENCY</td>
</tr>
<tr>
<td></td>
<td>CONTAMINANT MATERIAL BALANCE</td>
</tr>
<tr>
<td></td>
<td>SUM OF MASS REMOVED BY DEVICES</td>
</tr>
<tr>
<td></td>
<td>LOAD GENERATION INTO DD COL 19</td>
</tr>
<tr>
<td></td>
<td>CALCULATE FINAL AND AVERAGE CONC</td>
</tr>
<tr>
<td></td>
<td>CALCULATE LiOH USED IN INCREMENT</td>
</tr>
<tr>
<td></td>
<td>PRINT DATA TO THE SPECIFIED DEVICE(S)</td>
</tr>
<tr>
<td></td>
<td>PRINT CONCENTRATION DATA ANSWERS</td>
</tr>
<tr>
<td></td>
<td>PRINT DATA HEADINGS</td>
</tr>
<tr>
<td></td>
<td>PRINT TOXIC HAZARD INDEX ANSWERS</td>
</tr>
</tbody>
</table>
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 2 shows a flow diagram of PCSET. PCSET sets the initial time increment ending time to $\frac{1}{240}$ 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 $1 \times 10^{-20}$ 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 2. 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 3 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.

![Subroutine CNRSUB block flow diagram](image)

Figure 3. Subroutine CNRSUB block flow diagram.
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, RCHBC, 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 4 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 includes the cabin generation rate and the generation rate in each device. The steady-state concentration is defined according to the following equation:

\[
C_{ss} = \frac{(m_{\text{net-to-cabin}})}{(\eta_r \times Q)}
\]

where \(m_{\text{net-to-cabin}}\) is the mass of contaminant, \(\eta_r\) 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 (8). 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.

**Subroutine CALCM**

The removal device inlet and outlet concentrations and the total mass removed by the cabin and the specified removal devices is calculated 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 5 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
Figure 4. Subroutine MASBAL block flow diagram.
Figure 5. Subroutine CALCM block flow diagram.
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:

$$C_{\text{out}} = C_{\text{in}}(1 - \eta_r) \quad (9)$$

$$m_{\text{rem}} = C_{\text{in}}(Q)(\eta_r) \quad (10)$$

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:

$$C_4 = (C_1 Q_1 + C_2 Q_2 + C_3 Q_3)/Q_4 \quad (11)$$

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.

**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 6. 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 7 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.
Figure 6. Subroutine RINCDD block flow diagram.
BEGIN

BEGIN LOOP FOR DEVICES 3-15

DEVICE TYPE = 3, 4, OR 5

ASSIGN PROPER DEVICE VARIABLE

CHECK AND FIX INPUT INFO AS REQUIRED

CHECK REGEN TIME WITH RESPECT TO TIME INCREMENT

REGEN OCCURS AT BEGINNING OF TIME INCREMENT

REGEN OCCURS THROUGHOUT TIME INCREMENT

REPEAT FOR NEXT DEVICE

RETURN

START LOOP FOR ALL CONT 1 TO LIN

FOR DEV J

PUT SUM M = 0 IN CC COLUMN 12, 15, 18... AS APPROPRIATE

IF DEVICE = LiOH BED

PUT SUM M = 0 IN DD(J,16)

INACTIVE DEVICE

SET Q = 0

DD(J,2) = 0

Figure 7. Subroutine REGEN block flow diagram.
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 concentrations 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 8 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, 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 9 shows a block flow diagram for PREDCT.
Figure 8. Subroutine MCALC 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 are 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 10 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 achieved, 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 passes back to MCALC which determines whether convergence has been reached within 20 iterations.
Figure 10. Subroutine CONVRG block flow diagram.
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 11 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 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:

$$ A = \frac{T}{V_m} \log_{10} \left( \frac{p_v}{p_c} \right), $$

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 12. 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 13. Additional information may be obtained on charcoal capacity and performance from references 6 and 7. 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 functions of the adsorption potential factor, contaminant solubility, and cabin relative humidity. Specific equations used are found in subroutine FQI in the ACHBD.FOR listing found in appendix A. The general form of the equations is the following:

$$ q = \alpha e^{-\beta A}, $$

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-gmol/cm$^3$. As new information concerning adsorption capacity
The adsorption zone length for ninety percent removal is determined from experimental data obtained by Olcott at a 0.0066 m/s (1.3 ft/min) flow rate. This data is plotted in figure 14 and the computer program uses the following equation to calculate the adsorption zone length:

$$L_{\text{ads}} = (L_{\text{ads at 1.3 ft/min}})(V/1.3)^{0.8}.$$  \hspace{1cm} (14)

The adsorption zone length study conducted by Olcott shows that 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 as calculated based on the bed geometry, the amount of contaminant removed by the bed, and the saturation capacity, $q$. 

Figure 11. Charcoal saturation and adsorption zone distribution.
Figure 12. Potential plot for type BD granular activated charcoal.
Figure 13. Potential plot showing relative humidity effects.
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 indicates that an additional twenty 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 1.22 mmol/g of charcoal. This results in a requirement of 0.0061 g 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 that the removal efficiency is 100 percent if the bed is less than 80 percent utilized. The efficiency for last 20 percent of the bed is calculated using the following sine relationship:

$$\eta_r = \sin \left( \frac{m_{\text{charcoal}} - m_{\text{treated charcoal used}}}{(0.2)m_{\text{charcoal}}} \right)$$  \hspace{1cm} (15)$$

where $m_{\text{charcoal}}$ 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 indicates that this charcoal can chemisorb a total amount of formaldehyde equivalent to 5 percent of its weight.\(^9\) Testing at Lockheed Missiles and Space Company, Inc., showed this material’s efficiency to drop linearly from 100 to 90 percent for an amount of formaldehyde chemisorbed from 0 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:

$$\eta_r = \sin \left( \frac{m_{\text{LiOH}} - m_{\text{LiOH used}}}{(0.2)m_{\text{LiOH}}} \right)$$  \hspace{1cm} (16)$$

where $m_{\text{LiOH}}$ is the mass of the lithium hydroxide bed and $m_{\text{LiOH used}}$ 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 2 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 15.\(^{10}\)

**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
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 15. Noble metal CO catalyst performance.
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 not only by absorption but also by chemical reaction in the condensate. Ammonia is treated in this manner since it dissociates in water and reacts with dissolved carbon dioxide. 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 per mole fraction. Equation (17) 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.

\[
p_c = Hx . \tag{17}
\]

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

\[
x = y/[(C/A)+(H/P)] . \tag{18}
\]

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 mol/h, \( A \) is the atmospheric molar flow rate in mol/h, \( H \) is the Henry's Law constant in atmospheres, and \( P \) is the total pressure in atmospheres. Figure 16 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.

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 was used to obtain an equation relating liquid and gas phase ammonia composition. This data was obtained by sparging a gas mixture through a volume of water. Figure 17 shows a plot of the result. The carbon dioxide curve corresponding to 666.6 Pa (5 mm Hg) was used to obtain equation (19) which relates ammonia mass per kilogram of condensate to the entering ammonia concentration.

\[
m_a = 189.6C_c^{0.535} . \tag{19}
\]

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/m³. This equation is used to determine the ammonia removal efficiency from the mass of ammonia entering and leaving the condensing heat exchanger assembly.
Figure 16. Absorption of contaminants by humidity condensate.

Figure 17. Ammonia solubility in water at varying carbon dioxide concentration.
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 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 No. 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 No. 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 according to appendix B. 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

TCCS COMPUTER PROGRAM
VERSION 8.1
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:\FORTRAN\TCC\MAIN.F Options: /C 80 /L /BIJY 03/15/94 13:23:35

1 C FILE:MAIN.FOR
2 PROGRAM TCCS81
3 C
4 C ***************************************************************
5 C ***************************************************************
6 C * PROGRAM TO MODEL REMOVAL OF SPACECRAFT * 
7 C * GASEOUS CONTAMINANTS * 
8 C * VERSION 8.1 Alpha * 
9 C * March 15, 1994 * 
10 C ***************************************************************
11 C SUBROUTINES REQUIRED:
12 C CAFILL-FILL MATRIX WITH ZEROS
13 C RAFILL-FILL MATRIX WITH ZEROS
14 C CRIN-READ IN INPUT DATA
15 C RRIN-READ IN INPUT DATA
16 C CROUT-PRINT OUT INPUT DATA
17 C RRROUT-PRINT OUT INPUT DATA
18 C PCSET-PRECALCULATION SET UP ROUTINE
19 C MCALC-MAIN CALCULATION ROUTINE
20 C DATOUT-DATA PRINTOUT ROUTINE
21 C XXXXX-TIME DEPENDENT DATA ROUTINE
22 C REGEN-REGENERATION OF DEVICES ROUTINE
23 C SLIOH-SUM LIOH USED IN TIME INCREMENT
24 C
25 C NOTE:SUBROUTINES USE ADJUSTABLE SIZE ARRAYS
26 C WATCH COMPILER OPTIONS/DIMENSIONING IF
27 C ANY ARRAY IS LARGER THAN 64K BYTES
28 C
29 C ******************************************DIMENSION MAIN PROGRAM MATRICES******************************************
30 C NN=CONTAMINANT NAME MATRIX
31 C CDI=CONTAMINANT INPUT DATA MATRIX
32 C CC=CALCULATON MATRIX
33 C DD=DEVICE DEFINITION MATRIX
34 C TT=TIME DEPENDENT DATA MATRIX
35 C
36 C PRESENTLY SET TO HANDLE MAXIMUM OF 150 CONTAMINANTS IN MATRICES
37 C THIS VALUE =NROW AND IS USED IN ADDRESSING ADJUSTABLE SIZE
38 C ARRAYS IN SUBROUTINES
39 C NTTROW IS USED FOR MATRIX TT MAXIMUM LENGTH
40 C
41 C CHARACTER NN(300)*30
42 C REAL CDI(300,23)
43 C REAL CC(300,48)
44 C REAL DD(15,21)
45 C REAL DD(15,23)
46 C REAL TT(750,7)
47 C
48 C NOTE:MUST COMPILE SUBROUTINES PROPERLY FOR ADJUSTABLE SIZE
49 C ARRAYS IF A MAIN MATRIX EXCEEDS 65536 BYTES (REALS=4 BYTES)
50 C
51 C *************** DECLARE OTHER TERMS USED IN MAIN PROGRAM ***************
**WELCOME TO THE WORLD OF THE SPACECRAFT ATMOSPHERIC TRACE CONTAMINATION CONTROL SIMULATION PROGRAM - VERSION 8.1 Alpha -**

March 15, 1994

******* DEFINE PROGRAM VARIABLES ******************************

LIN = NO. OF LINES OF DATA IN MAT NN & MAT CDI & MAT CC

LINI = NO. LINES OF DATA IN MAT TT

LIN2 = NO. LINES OF DATA IN MAT DD

TN = INCREMENT END TIME (HRS)

TNI = INCREMENT BEGINNING TIME (HRS)

TMS = TOTAL MISSION TIME (HRS)

NINC = NUMBER OF TIME INCREMENTS ELAPSED

MAT NN, CC, AND CDI MUST HAVE SAME NO. OF ROWS

DIMENSIONS OF MAT DD

NROW = 15

NCOL = 21

DIMENSIONS OF MAT CC & ROWS IN MAT NN

NROWI = 300

NCOLI = 48

DIMENSIONS OF MAT CDI

NROW2 = NROWI

NCOL2 = 23

DIMENSIONS OF MAT TT

NTTROW = 750

NTTCOL = 7

DEVICE NUMBER FOR OUTPUT DATA (SET TO 6 FOR FORM FEED ON OUTPUT)

IDEVNO = 6

DEVICE NUMBER FOR MESSAGE OUTPUT

IMSGDN = 2

DEVICE NUMBER FOR CONTAMINANT PLOT DATA

IDEV1 = 10

DEVICE NUMBER FOR T-VALUE PLOT DATA

IDEV3 = 11

DEVICE NUMBER FOR EFFICIENCY PLOT DATA

IDEV2 = 12

*********** END OF DEFINITION SECTION **************************

******** ZERO MATRICES ****************************************

PUT BLANKS IN NAME MATRIX

CALL CAFILL(NN, 1, NROWI)

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)
********** READ IN DATA FROM FILES AND PRINT IT IF DESIRED **********

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)

020 WRITE(*,*) 'INPUT DEVICE DEFINITION TABLE FILE NAME: '  
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 RRROUT2(DD,NROW,NCOL,1,16,LIN2,IMSGDN)

030 WRITE(*,*) 'INPUT TIME DEPENDENT DATA FILE NAME: '  
CALL RRIN(TT,NTTROW,NTTCOL,NTTCOL,LINI)
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 RRROUT2(TT,NTTROW,NTTCOL,1,NTTCOL,LINI,IMSGDN)

********** READ IN MISSION DATA VARIABLES **********

040 WRITE(*,*) 'INPUT TOTAL MISSION TIME IN HOURS: '  
READ(*,*) TMIS

********** PRINT SWITCH DEFINITION **********

1=RESULTS FOR ONE CONTAMINANT IN PCSET  
2=RESULTS FOR 1 CONT & INCR IN 1/10 INCR CONV ROUTINE (IN MCALC)  
3=CONVERGENCE VALUES IN CONVRG  
4=RESULTS FOR 1 CONT IN MCALC AFTER CAV CALC  
5=MAT CC AND MAT DD AT END OF TIME INCREMENT  
6=PRINT CONC+M.REM+SUM MASS REM+REM EFF (OTHERWISE ONLY CONC DATA)  
7=PRINT OUTPUT WITH NO FORM FEEDS  
8=PRINT ANSWERS DURING EACH ITERATION (IN MAIN PROGRAM) AND CONTROL PLO  
9=CONTROL PLOT FILE OUTPUT  
TV=CTRL 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))

**** 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
*************** 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
CHANGE TO NO FORM FEED IF PRTSW7=1
IF (PRTSW7.EQ.1) THEN
  IDEVNO=7
ENDIF
*************** MAKE DECISIONS ON DATA OUTPUT **********************
THIS IS WHERE ALL PROGRAM OUTPUT DATA FILES ARE OPENED
THEY MUST BE CLOSED AT THE END OF THE PROGRAM
SECTION WHICH CHECKS FOR EXISTANCE OF OUTPUT FILE & OPENS IT
WRITE(*,*) WRITE OUTPUT TO FILE, PRINTER, SCREEN, OR END?
WRITE (*,*) (FILE NAME/LPT1/CON/END)
*********** NOTE: LPT1 OUTPUT REQUIRES 132 COLUMNS ***********
READ(*,' (A)') FNAME
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
*** OPEN FILE FOR CONCENTRATION PLOT DATA IF PRTSW9=1 OR 3 ***
IF ((PRTSW9.EQ.1).OR.(PRTSW9.EQ.3)) THEN
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',IOT=IOVAL)
ENDIF
IF (IOVAL.NE.0) THEN
CLOSE (UNIT=10)
WRITE (*,*) 'IOERROR=' , IOVAL
GOTO 52
ENDIF

****** OPEN FILE FOR EFFICIENCY PLOT DATA IF PRTSW9=2 OR 3 ******
IF ((PRTSW9.EQ.2).OR.(PRTSW9.EQ.3)) THEN
WRITE (*,*) 'FILE NAME FOR EFFICIENCY PLOT DATA OUTPUT? '
READ (*,'(A)') FEPLT
INQUIRE (FILE=FEPLT,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=FEPLT,STATUS='OLD',IOSTAT=IOVAL)
ENDIF
ENDIF
OPEN (UNIT=12,FILE=FEPLT,STATUS='NEW',IOSTAT=IOVAL)
ENDIF
IF (IOVAL.NE.0) THEN
CLOSE (UNIT=12)
WRITE (*,*) 'IOERROR=' , IOVAL
GOTO 54
ENDIF

********** OPEN FILE FOR T-VALUE PLOT DATA IF TVAL=2 **********
IF (TVAL.EQ.2) THEN
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
OPEN (UNIT=11, FILE=FTPLOT, STATUS='NEW', IOSTAT=IOVAL)
ENDIF
IF (IOVAL.NE.0) THEN
  CLOSE (UNIT=11)
  WRITE (*,*) 'IOERROR= ',IOVAL
  GOTO 56
ENDIF
CALL SYSTEM TIME AND DATE
THIS MUST BE CALLED ONLY ONCE SO THAT THE TIME AND DATE WILL
BE THE SAME ON ALL OUTPUT INFORMATION FOR ONE RUN
CALL DATTM(IMONTH,IDAY,IYEAR, IHOUR,IMINUTE,ISECOND)
SET IPGCTR= COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUTS
IPGCTR=0
********** CHECK BASIC TIME INCREMENT ****************************
BINC=DD(1,11)
********** TEST FOR BINC=0 (CAUSES ENDLESS TIME LOOP) **********
IF (BINC.EQ.0) THEN
  CLOSE (IDEVNO)
  WRITE(*,*)' ERROR-BASIC TIME INCREMENT=0'
  GOTO 999
ENDIF
TN=0
TNI=0
NINC=0
************* PRECALCULATION SET UP ROUTINE *****************
FOR ALL CONTAMINANTS ONE AT A TIME AT CAV PRED=IE-20, CALC INIT
DEV EFF AND LOAD IT INTO MAT CC-ALSO CALC CAVPRD(CAV PREDICTED)
OUTPUTS TO PRECALC SET UP ROUTINE:
TN1=INCREMENT INITIAL TIME (HRS)
BINC=BASIC INCREMENT SIZE (HRS) (REF.=DD(1,11)) PASS IN????
LIN=NO. OF CONT IN MAT CC AND NN
DD,NROW,NCOL=NAME & SIZE OF MAT DD
CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
LIN2=NO. DEVICES IN MAT DD
NN=NAME OF MAT NN
INPUTS FROM PRECALC SETUP ROUTINE-SUBROUTINE PCSET:
CAVPRD=PRED CABIN AV CONC (MG/CU M):=CC(I,2)
CEQLIB=EQUILIBRIUM CABIN CONT CONC (MG/CU M):=CC(I,3)
CFINAL=FINAL CABIN CONT CONC (MG/CU M):=CC(I,4)
PUS REM EFF FROM DD COL 20 IN CC(I,7-10-13-16 ETC)
PUS M.REM IN CC(I,5-8-11-14 ..)
CALL FCSET(TN1, LIN, DD, NROW, NCOL, CC, NROWI, NCOL1,
+CDI, NROW2, NCOL2, LIN2, NN, PRTSWI, IMSGDN)

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

********** BEGINNING OF CALCULATION FOR EACH TIME INCREMENT **********

100 CONTINUE

********** INCREASE INCREMENT COUNTER **********************

NINC=NINC+1

********** 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

CHECK FOR INCREMENT FINAL
IF (TN.GT.TMIS) TN=TMIS

********** CHECK FOR CHANGES IN BASIC TIME **********
BINCNEW=BINC
DO 105 K=1,LINI
   IF ((TT(K,1).GE.TN).AND.(TT(K,1).LT.TN)) THEN
      IF ((TT(K,4).EQ.I).AND. (TT(K,6).EQ.II)) THEN
         BINCNEW=TT(K,7)
      ENDIF
   ENDIF
105 CONTINUE

IF (BINC.NE.BINCNEW) THEN
   TN:TN-BINC+BINCNEW
   BINC=BINCNEW
   DD(I,II)=BINC
ENDIF

********** STORE PREVIOUS INCREMENT CABIN VOLUME **********
PREVCVOL=DD(I,9)

********** READ TIME DEPENDENT DATA **********
CALL TIME DEPENDENT DATA SUBROUTINE-RINCDD
CALL RINCDD(I,TN,TNI,DD,NROW,NCOL,LIN2,
+CC,NROWI,NCOLI,CDI,NROW2,NCOL2,LIN,TT,NTTROW,NTTCOL,LINI)

CHECK TO SEE IF CABIN VOLUME HAS CHANGED, AND IF SO
UPDATE INITIAL CABIN CONCENTRATION FOR NEW VOLUME
IF(PREVCVOL.NE.DD(I,9)) THEN
   DO 200 I=I,LIN
      CC(I,i) =CC(I,6)/DD(I,9)
   CONTINUE
ENDIF

****** LIST INCREMENT NO. AND TIMES TO CONSOLE ******
IF((FNAME.NE.'CON').OR.(PRTSWI.NE.I))THEN
   OPEN(IMSGDN, FILE='CON',IOSTAT=IOVAL)
   WRITE(IMSGDN, 65)NINC,TNI,TN
   65 FORMAT (IX,'INCR NO.= ',I5,' BEGIN & END TIMES (hours)=',
   +F9.3,2X,F9.3)
**STORE ORIGINAL Q DEVICE IN DD COL 7 (TAKEN FROM DD COL 2)**

```fortran
  close (imsgdn)
endif

**CHECK FOR REGENERATION IN TIME INCREMENT**

```fortran
  call regeneration subroutine regen
  call regen(tn,tn1,dd,nrow,ncol,cc,nrow1,ncol1,
             +cdi,nrow2,ncol2,lin,lin2,imsgdn)

**CALL MAIN CALCULATION SUBROUTINE**

```fortran
  outputs to main calc subroutine - mcalc:
  i = contaminant no.
  tn,tn1 = increment end & beginning time (hrs)
  dd,nrow,ncol = name & dim of mat dd
  cc,nrow1,ncol1 = name & dim of mat cc
  cdi,nrow2,ncol2 = name & dim of mat cdi
  nn = name of mat nn
  lin = number of contaminants in mat nn & cdi
  lin2 = no. devices in mat dd
  inputs from main calc routine - mcalc:
  to mat cc
  puts cavclc,ceqliv,&cfinal in cc(i,2-3 &4)
  puts rem eff from dd col20 in cc(i,7-10-13 etc)
  puts sum mass rem for each dev from dd col21 in cc(i,6-9-12 etc)
  call mcalc(i,tn,tn1,dd,nrow,ncol,
             +cc,nrow1,ncol1,cdi,nrow2,ncol2,nn,lin,lin2,
             +prtsw2,prtsw3,prtsw4,imsgdn)

**CALCULATE LIOH USED IN INCREMENT**

```fortran
  call lioh removal subroutine slioh
  call slioh(tn,tn1,dd,nrow,ncol,cc,nrow1,ncol1,
             +cdi,nrow2,ncol2,lin,lin2)

**RESTORE DEVICE FLOW**

```fortran
  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

**PRINTOUT OF DATA FOR EACH TIME INCREMENT**

```fortran
  if prtsw5=1 then print mat dd+mat cc info for this contaminant
  if (prtsw5.eq.1) then
    open (imsgdn,file='con',iostat=ioval)
    write(imsgdn,*),'printout for mat cc & dd at end of time incr' 
    write(imsgdn,*),'info from mat cc'
    close(imsgdn)
    call cROUT(nn,cc,nrow1,ncol1,1,ncol1,lin,1,lin,imsgdn)
    +fname,idevno,ioval)
    call cROUT(nn,cc,nrow1,ncol1,1,ncol1,lin,1,lin,imsgdn,ninc,
    +fname,idevno,ioval)
    open (imsgdn,file='con',iostat=ioval)
    write(imsgdn,*),'info from mat dd'
    close(imsgdn)
    call rROUT(dd,nrow,ncol,1,ncol,lin2,imsgdn)
    call rROUT(dd,nrow,ncol,1,ncol,lin2,imsgdn,fname,idevno,ioval)
```

40
ENDIF

C ********** REGULAR PRINTOUT OF DATA FOR EACH INCREMENT **********

IF (PRTSW8.EQ.1) THEN

CALL DATOUT(TN,TNI,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,
+ TVA/T,FCPLOT,IDEVN1,IDEVN3,IDEVN2)

ENDIF

C IF (IOVAL.NE.0) THEN
CLOSE (IDEVNO)
WRITE( *,*) 'PROGRAM DATA OUTPUT ERROR IN INCREMENT = ',NINC
GOTO 999

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 ***************** CHECK FOR END OF MISSION *********************

IF(TN.GE.TMIS) THEN

END TIME LOOP
CONTINUE
ELSE

********* UPDATE FOR NEXT TIME INCREMENT AND REPEAT **********

SET TFINAL FOR THIS INCR = TINIT FOR NEXT INCR
TN1=TN

SET CFINAL FOR INCR=CINIT FOR NEXT INCR-ALL CONTAMINANTS
DO 130 I=I,LIN
  CC(I,1)=CC(I,4)
130 CONTINUE
GOTO 100

ENDIF

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,
+ TVA/T,FCPLOT,IDEVN1,IDEVN3,IDEVN2)

ENDIF

C IF (IOVAL.NE.0) THEN
CLOSE (IDEVNO)
WRITE( *,*) 'PROGRAM DATA OUTPUT ERROR - FINAL PRINTOUT'
GOTO 999

ENDIF

C *********** CLOSE ALL PROGRAM OUTPUT FILES ***********

CLOSE (IDEVNO)
CLOSE (IDEVN1)
CLOSE (IDEVN3)
CLOSE (IDEVN2)
CONTINUE
WRITE(*,*) 'DO YOU WISH TO RUN ANOTHER CASE? (Y/N)'
READ (*, '(A)') DES
IF (DES.EQ.'Y') THEN
   GO TO 011
ENDIF
*****************************************************************
END OF MAIN PROGRAM
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
SUBROUTINE ACHBD (TN, TNI, CIIN, TCABIN, COEXIS, BEDQ, EMAX, BEDL, +BEDDIA, DENCH, TRTTYP, DCONT, VMOL, MW, VCONC, SOL, SMR, EFF, RH)

OUTPUT:
EFF = BED REMOVAL EFF (DEC)

INPUTS:
TN, TNI = INCREMENT INITIAL AND FINAL TIMES (HR)
CIIN = BED INLET CONT CONC (MG/CU M)
TCABIN = CABIN TEMP (DEG K)
COEXIS = COEXISTANCE FACTOR
BEDQ = BED FLOW RATE (CU M/HR)
EMAX = MAXIMUM BED EFF (DEC)
BEDL = BED LENGTH (M)
BEDDIA = BED DIAMETER (M)
DENCH = DENSITY OF CHARCOAL IN BED (KG/CU M)
TRTTYP = BED TREATMENT TYPE (1 = CI CHAR, 2 = PHOS ACID, OTHER
# = NONE)
DCONT = CONT LIQUID DENSITY (GM/CC)
VMOL = CONT MOLAR VOL (GM/CC)
MW = CONT MOLECULAR WGT
VCONC = CONT VAPOR CONCENTRATION AT TCABIN (MG/CU M)
SOL = HENRY'S LAW CONSTANT FOR WATER SOLUBILITY
SMR = SUM OF CONT MASS STORED IN BED (MG) - FROM LAST INCR

REAL LPREV, LAVNI, LUTIL, LIMM, LAVAV, LADS, MW
INTEGER FACID, FCI

SET CIIN = CIIN (THIS PREVENTS CIIN FROM BEING PASSED BACK UP TO OTHER SUBROUTINES IF IT IS SET TO 1E-20)
CIIN = CIIN

BED TREATMENT LOGIC
FACID = FLAG IF BED IS TREATED WITH PHOSPHORIC ACID (Y = 1
N = 0)
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

TEST FOR NO BED FLOW (BEDQ < 0) OR
TN - TNI <= 0; BEDL, BEDDIA, DENCH = 0
IF ((BEDQ.LE.0).OR. (TN-TNI.LE.0).OR. (BEDL.LE.0).OR. (BEDDIA.LE.0)
+.OR. (DENCH.LE.0)) THEN
EFF = 0
GOTO 199

ENDIF

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

TEST FOR AMMONIA AND H3PO4 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

TEST FOR MOL VOL=0 (NO CHAR REMOVAL)

IF (VMOL.EQ.0) THEN

EFF=0

GOTO 199

ENDIF

CHARCOAL REMOVAL EFFICIENCY CALCULATION

SUPERFICIAL BED VEL (FT/MIN)

BEDVEL=BEDQ*.06960/BEDDIA**2

TEST FOR CIN TOO SMALL IN AVAL CALC

IF (CIN.LT.1E-20) CIN=1E-20

AVAL=(TCABIN/VMOL)*LOG10(VCONC/CIN)

ADS ZONE LENGTH FOR 90% REMOVAL (M)

LADS=AVAL*.000275*(BEDVEL/I.3)**.8

GET QI (CC LIQ CONT/GM CHAR)

CALL FQI(AVAL,QI,FACID,SOL,RH)

LENGTH OF BED PREVIOUSLY USED BY CONT AT THIS C INLET (M)

LPREV=SMR*I.273E-6*COEXIS/(DCONT*DENCH*BEDDIA**2*QI)

RATE OF BED USAGE (M BED/ MG CONT)

LIMM=I.273E-6*COEXIS/(DCONT*DENCH*BEDDIA**2*QI)

LENGTH OF BED AVAILABLE FOR ADS ZONE AT BEGINNING OF INCREMENT (M)

LAVNI=BEDL-LPREV

IF (LAVNI.LT.0) LAVNI=0

FIX HERE IF DESORPTION IS DESIRED

IF (LAVNI/LADS.GT.20) THEN

EFFAV=EMAX

ELSE

INIT INCR EFF BASED ON C IN AND BED L AVAIL AT BEG OF INCR (DEC)

EFAVNI=EMAX* (1-EXP(-2.3025851*LAVNI/LADS))

LOOP FOR EFFICIENCY

EFAV=EFAVNI

DO 399 J=l,10,1

LENGTH OF BED UTILIZ IN INCR (M)

LUTIL=CIN*BEDQ*EFFAV*(TN-TNI)*LIMM

IF (LUTIL.GT.LAVNI) THEN

GOTO 299

ELSE

AVERAGE BED LENGTH AVAIL (M)

LAVAV=LAVNI-LUTIL/2

IF ((LAVAV/LADS).GE.20) THEN

EFFAV=EMAX

GOTO 299

ELSE

AVERAGE BED LENGTH AVAIL (M)

LAVAV=LAVNI-LUTIL/2

IF ((LAVAV/LADS).GE.20) THEN

EFFAV=EMAX

GOTO 299

ELSE
AV EFF BASED ON AV BED L AVAIL (DEC)

\[ \text{EFFAV} = \text{EMAX} \times (1 - \exp(-2.3025851 \times \text{LAVAV}/\text{LADS})) \]

ENDIF

399 CONTINUE

MAX EFF BASED ON C IN AND RATE OF BED USAGE (DEC)

\[ \text{EFFMAX} = \text{LAVNI}/(\text{CIN} \times \text{BEDQ} \times (\text{TN} - \text{TN1}) \times \text{LIMM}) \]

IF (EFFAV.GT.EFFMAX) EFFAV=EFFMAX

IF (EFFAV.LT.0) EFFAV=0

IF (EFFAV.GT.EMAX) EFFAV=EMAX

EFF=ACTUAL EFF OUTPUT FROM SUBROUTINE

EFF=EFFAV

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

136 C

*************

137 C

* SUBROUTINE ACIDCH - CALCULATES REMOVAL EFF *

* BED WITH NH3 AND 1.22 MILLIMOLE H3PO4 ON CHAR *

139 C

SUBROUTINE ACIDCH(EFF, EMAX, BEDL, BEDDIA, DENCH, SMR)

141 C

OUTPUTS

142 C

EFF=OUTPUT REMOVAL EFF (DEC)

143 C

INPUTS

144 C

EMAX=MAXIMUM BED REMOVAL EFF (DEC)

145 C

BEDL=BED LENGTH (M)

146 C

BEDDIA=BED DIAMETER (M)

147 C

DENCH=CHARCOAL DENSITY (KG/CU M)

148 C

SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)

149 C

FOR AMMONIA CAPACITY AT SMAC

151 C

CHAR USED (KG)

152 C

CHRUSD=1.6E-4*SMR

153 C

CHAR BED WGT(KG)

154 C

BEDWGT=BEDL*BEDDIA**2*.785*DENCH

155 C

IF (CHRUSD.LT.0.8*BEDWGT) THEN

156 C

EFF=EMAX

157 C

ELSE

158 C

EFF=EMAX*SIN((BEDWGT-CHRUSD)*1.57/(BEDWGT*0.2))

159 C

ENDIF

160 C

PREVENTS NEGATIVE EFF FOR REACTION

161 C

IF (EFF.LT.0) EFF=0

162 C

IF (EFF.GT.EMAX) EFF=EMAX

163 C

RETURN

164 C

END

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

165 C

*************

166 C

* SUBROUTINE CICH - CALCULATES REMOVAL EFF *

* FOR FORMALDELYDE AND CI CHAR BED *

168 C

SUBROUTINE CICH(EFF, EMAX, BEDL, BEDDIA, DENCH, SMR, BEDQ)
OUTPUTS
EFF=OUTPUT REMOVAL EFF (DEC)

INPUTS
EMAX=MAXIMUM BED REMOVAL EFF (DEC)
BEDL=BED LENGTH (M)
BEDDIA=BED DIAMETER (M)
DENCH=CHARCOAL DENSITY (KG/CU M)
SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
BEDQ=BED FLOW RATE (CU M/HR)

BEDWGT=BEDL*BEDDIA**2*.785*DENCH
PERCENT OF BED WEIGHT CONSUMED (DEC)
PBWGT=SMR/(BEDWGT*1E6)

INSOLUBLE CONTAMINANTS
SUMMARY OF BED REACTION EFFICIENCY

BED RESIDENCE TIME (SEC)
BREST=BEDL*BEDDIA**2*3600/(BEDQ*1.273)

PREVENTS NEGATIVE EFF FOR REACTION
IF (BREST.LT.0.25) THEN
EFF=EFF*BREST/.25
ENDIF

EFFECTIVENESS
IF (EFF.LT.0) EFF=0
IF (EFF.GT.EMAX) EFF=EMAX
RETURN
END

**SHOULD**
ELSE
    IF (RH.LE.50) THEN
        QI=(0.000096*RH**2-0.0188*RH+2.11)*EXP(-0.31*AVAL)
    ELSEIF (RH.GT.50) THEN
        QI=(0.000096*RH**2-0.0188*RH+2.11)*
            EXP(-AVAL*(0.25+0.0012*RH))
    END IF
ENDIF

C A VALUE .LE. 8
ELSEIF (AVAL.LE.8) THEN
    IF (SOL.GT.0.AND.SOL.LT.5) THEN
        QI=0.5-AVAL*0.0405
    ELSE
        QI=-0.0000128*RH**2-0.00264*RH+0.5+(0.00000112*RH**2+
            0.000208*RH-0.0405)*AVAL
    END IF
ENDIF

C AVAL .GE. 200
ELSE
    QI=1E-20
ENDIF
RETURN
END
**SUBROUTINE ALIOH - AXIAL FLOW BED**

**CALCULATES REMOVAL EFFICIENCY FOR LIOH**

SUBROUTINE ALIOH(TN,TNI,EMAX,BEDL,DENLI,BEDDIA,RWUTLI,SWUTLI, 
+ REMFCT,EFF)

INPUTS:

TN=FINAL INCREMENT TIME (HRS)
TNI=INITIAL INCREMENT TIME (HRS)
EMAX=MAXIMUM POSSIBLE REMOVAL EFFICIENCY (DEC)
BEDL=BED LENGTH (M)
DENLI=LIOH DENSITY (KG/CU M)
BEDDIA=BED DIAMETER (M)
RWUTLI=RATE OF LIOH USAGE FOR ALL CONTAMINANTS FROM LAST INCR(KG/HR)
SWUTLI=SUM OF WEIGHT OF LIOH UTILIZED FROM LAST INCR(KG)
REMFCT=LIOH REMOVAL FACTOR (LB LIOH/LB CONTAMINANT)

OUTPUTS:

EFF=REMOVAL EFFICIENCY (DEC)

IF CONT DOESNT REACT WITH LIOH OR BED<=0 OR BED DIA <=0 OR 
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
BED WEIGHT (KG)
BEDWGT=BEDL*(BEDDIA)**2*.785*DENLI
TOTAL WEIGHT OF LIOH UTILIZED AT AVERAGE TIME IN INCREMENT (KG)
TWUTLI=SWUTLI+RWUTLI*(TN-TNI)/2
IF (TWUTLI/BEDWGT.LE.0.8) THEN
EFF=EMAX
ELSE
EFF=EMAX*SIN((BEDWGT-TWUTLI)*I.57/(BEDWGT*0.2))
ENDIF
IF (BEDL.LT.0.0191) THEN
EFF=EFF*BEDL/0.0191
ENDIF
END IF
IF(EFF.LT.0) EFF=0
IF(EFF.GT.EMAX) EFF=EMAX
RETURN
END

*********** END OF SUBROUTINE ALIOH ****************************
SUBROUTINE CAFILL(NN,NROW,NCOL)
INTEGER NROW, NCOL
CHARACTER NN(NROW, NCOL)*30

DO ii0 I=1,NROW
   DO i00 J=1,NCOL
      NN(I,J)=' '
   CONTINUE
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\CALCM.F Options: /C 80 /L /BY 05/21/92 12:54:56

```fortran
1 C ****************************************************************
2 C SUBROUTINE CALCM *
3 C ****************************************************************

SUBROUTINE CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
INTEGER NROW,NCOL,LIN2
REAL DD(NROW,NCOL)

SUBROUTINES REQUIRED:
NONE

INPUTS:
CAV=CABIN CONT AVERAGE CONCENTRATION (MG/CU M)
DD,NROW,NCOL=NAME AND SIZE OF MAT DD
TN=INCREMENT FINAL TIME (HRS)
LIN2=NO. OF DEVICES IN MAT DD
REM EFF AND DEVICE+CABIN M.GEN MUST BE LOADED INTO MAT DD
BEFORE USING THIS SUBROUTINE

OUTPUTS:
M.REM (MG/HR) FOR ALL DEVICES + CABIN CALCULATED AT CAV,
ARE STORED IN MAT DD COL 21
SMREM=SUM OF MASS OF CONT REM IN DEVICES (MG/HR)
SMGEN=SUM OF MASS GENERATED BY ALL DEVICES + CABIN (MG/HR)

LOAD DEVICES 1 AND 2 WITH CIN AND COUT+M.REMOVED FOR DEV 2
DEVICE 1=CABIN
LOAD MAT DD WITH CAV CABIN

DEVICE 2
DD(2,17)=CAV
DD(2,18)=CAV
DD(2,21)=DD(2,21)*DD(2,18)

FOR DEV 3-15 CALC CIN COUT AND M.REMOVED BY DEVICE
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
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(J,17)+DD(J,19)/DD(J,2)
ELSE
DETERMINE FLOWS,CIN AND COUT FOR DEVICES WITH RELATIVE ADDRESSES
IF (DD(J,4).EQ.0) THEN
QNOI=0
CNOI=0
ELSE
QNOI=DD(NINT(DD(J,4)),2)
CNOI=DD(NINT(DD(J,4)),18)
ENDIF
```

50
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
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(*,'(A)') 'FLOW HALTED-UPSTREAM DEV TURNED OFF-
+ INC END TIME;DEV=',TN,DD(J,1)
  CLOSE (2)
  GO TO 100
ELSE
  CALCULATE CIN
  DD(J,17)=(QNO1*CNO1+QNO2*CNO2+QNO3*CNO3)/(QNO1+QNO2+QNO3)+DD(J,19)/DD(J,2)
ENDIF
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(I,21)
DD(1,21)=SMGEN-SMREM
RETURN
C
**************** END OF SUBROUTINE CALCM ********************
END
SUBROUTINE CATBNR(P, EMAX, OXNEW, OXID, MW, EFF)
REAL MW

INPUTS:
P=TOTAL POISON (CONTAMINANTS IN NHB CATEGORIES 6, 7 AND 12: CHLOROCARBONS, CHLOROFLUOROCARBONS AND SULFIDES) REMOVED
BY CAT BURNER (MG)
EMAX=MAXIMUM BED EFFICIENCY (DEC)
OXID=DEGREE OF OXIDIZATION OF CHEMICAL (1=FULLY, 0=NONE)
MW=MOLECULAR WEIGHT

OUTPUTS:
EFF=REMOVAL EFF (DEC)

IF(OXID.LT.0) OXID=0
IF(OXID.GT.1) OXID=1
EFF=EMAX*OXID

EFFICIENCY FOR METHANE (MW=16.04) IS A FUNCTION OF P
IF(MW.EQ.16.04) THEN
OXNEW=OXID
IF(P.LE.5500) THEN
EFF=0.97506*10**(-0.00010507*P)*EMAX*OXID
ELSEIF (P.GT.5500) THEN
EFF=(31.453-1.151*1E-3*P+1.9046*1E-8*P**2
+ -1.0389*1E-13*P**3)*0.01*EMAX*OXID
ENDIF
ELSE
EFF=EMAX*OXID
END IF

IF(EFF.LE.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
**SUBROUTINE CNRSUB**

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

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

SUBROUTINE CNRSUB(I,TN,TNI,DD,NROW,NCOL,CC,NROW1,NCOL1,
+CDI,NROW2,NCOL2,LIN2,KK)

INTEGER NROW,NCOL,NROWI,NCOLI,NROW2,NCOL2,CATEG,DI,D2,D3,KK
REAL DD(NROW,NCOL)
REAL CC(NROW1,NCOL1)
REAL CDI(NROW2,NCOL2)
REAL RGTMI,RGTM2,RGTM3,P,POISN,OLDP

SUBROUTINES REQUIRED:
ACHBD-REMOVAL EFF OF AXIAL CHARCOAL BED
RCHBD-REMOVAL EFF OF RADIAL CHARCOAL BED
ALIOH-REMOVAL EFF OF AXIAL LIOH BED
COOXID-REMOVAL EFF OF CO OXIDIZER
CATBNR-REMOVAL EFF OF CAT BURNER
CONDHX-REMOVAL EFF OF CONDENSING HX

INPUTS:
FROM PCSET PREDCT, AND CONVRG
I=CONT NO.
TN,TNI=CONT INCREMENT FINAL, INIT TIME (HRS)
CAVPRED=CABIN AVERAGE CONT CONC (MG/CU M)
DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
CC,NROW1,NCOL1=NAME & DIM OF MAT CC
CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
LIN2=NUMBER OF DEVICES IN MAT DD
DD(I,14)=PERCENT RELATIVE HUMIDITY
FROM EFFICIENCY SUBROUTINES
EFF=REMOVAL EFF (DEC)

OUTPUTS:
TO EFFICIENCY SUBROUTINES
DEVICE AND CONTAMINANT INFORMATION AS REQUIRED
DD(J,22)=DEVICE INLET CONCENTRATION
TO PCSET, PREDCT, & CONVRG
Puts REMOVAL EFF FOR EACH DEVICE IN MAT DD COL 20

EFFICIENCY FOR DEVICES 1 AND 2
DD(1,20)=0
DD(2,20)=DD(2,8)

SET COUNTER FOR READING CHAR SMR IN MAT CC FOR DEVICE TYPE 3
K=9

BEGIN LOOP FOR DEVICES 3 TO 15 OF MAT DD
DO 100 J=3,LIN2
INDEX COUNTER
K=K+3
SET REM EFF=0 AND GO TO END OF J LOOP IF DEVICE FLOW = 0
IF(DD(J,2).LE.1E-10) THEN

53
IF (NINT(DD(J,3)).EQ.3) THEN
CALL ACHBD(TN,TNI,DD(J,22),DD(I,10),DD(I,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,DD(I,14))
STORE EFF IN MAT DD
DD(J,20)=EFF
ELSEIF (NINT(DD(J,3)).EQ.4) THEN
CALL RCHBD(TN,TNI,DD(J,22),DD(I,10),DD(I,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,DD(I,14))
STORE EFF IN MAT DD
DD(J,20)=EFF
ELSEIF (NINT(DD(J,3)).EQ.5) THEN
CALL ALIOH(TN,TNI,DD(J,8),DD(J,9),DD(J,10),DD(J,12),DD(J,15),
+ DD(J,16),CDI(I,7),EFF)
STORE EFF IN MAT DD
DD(J,20)=EFF
ELSEIF (NINT(DD(J,3)).EQ.6) THEN
CALL COOXID(DD(J,2),DD(J,8),DD(J,9),DD(J,10),CDI(I,4),EFF)
STORE EFF IN MAT DD
DD(J,20)=EFF
ELSEIF (NINT(DD(J,3)).EQ.7) THEN
SUM POISONS (CONTAMINANTS IN NHB CATEGORIES 6, 7 AND 12:
CHLOROCARBONS, CHLOROFLUOROCARBONS AND SULFIDES) REMOVED
BY CAT BURNER (MG)
P=0.0
DO 200 L=I,NROWI,I
CATEG=CDI(L,8)
IF((CATEG.EQ.6).OR.(CATEG.EQ.7).OR.(CATEG.EQ.12)) THEN
P=CC(L, (J+l)*3)
POISN=POISN+P
ENDIF
200 CONTINUE
RGTM 1 = 0
RGTM2 = 0
RGTM3 = 0
DI=DD (J,4)
D2=DD(J,5)
D3=DD(J,6)
IF ((DI.NE.0).OR.(D2.NE.0).OR.(D3.NE.0)) THEN
TRCI=DD(DI,15)
TRD=DD(DI,16)
TIR=DD(DI,14)
IF (TN1.EQ.0) GOTO 50
IF (TN1.LT.TIR) GOTO 50
IF (TRCI.LE.0) GOTO 50
IF (AINT((TN1-TIR)/TRCI)).EQ.((TN1-TIR)/TRCI)) THEN
RGTM1=1
GOTO 55
ENDIF
GOTO 55
ENDIF
C

CONTINUE

ENDIF

IF ((DD(D2,3),EQ.3).OR.(DD(D2,3),EQ.4)) THEN

TCI=DD(D2,15)

TRD=DD(D2,16)

TIR=DD(D2,14)

IF (TN1.EQ.0) GOTO 60

IF (TN1.LT.TIR) GOTO 60

IF (TIR.EQ.0) GOTO 60

IF (TNI.EQ.0) GOTO 60

IF (TNI.LE.0) GOTO 60

IF (TNI.LT.TIR) GOTO 60

IF (TNI.LE.0) GOTO 60

IF (AINT((TNI-TIR)/TRCI).EQ.((TNI-TIR)/TRCI)) THEN

RGTM2=1

GOTO 65

ENDIF

RGTM2=0

GOTO 65

ENDIF

END IF

IF ((DD(D3,3),EQ.3).OR.(DD(D3,3),EQ.4)) THEN

TCI=DD(D3,15)

TRD=DD(D3,16)

TIR=DD(D3,14)

IF (TN1.EQ.0) GOTO 70

IF (TN1.LT.TIR) GOTO 70

IF (TIR.EQ.0) GOTO 70

IF (TNI.EQ.0) GOTO 70

IF (TNI.LE.0) GOTO 70

IF (TNI.LT.TIR) GOTO 70

IF (TNI.LE.0) GOTO 70

IF (AINT((TNI-TIR)/TRCI).EQ.((TNI-TIR)/TRCI)) THEN

RGTM3=1

GOTO 75

ENDIF

RGTM3=0

GOTO 75

ENDIF

END IF

IF((RGTM1.EQ.0) . OR. (RGTM2.EQ.0) . OR. (RGTM3.EQ.0)) THEN

OLDP=POISN

REINITIALIZES METHANE OXIDATION EFFICIENCY TO 90% OF PREVIOUS

IF ((CDI(I,4),EQ.16.04) . AND. (KK.EQ.1)) THEN

CDI(I,23)=0.9*CDI(I,23)

ENDIF

END IF

END IF

GO TO SUBROUTINE FOR DEVICE TYPE 7-CAT BURNER

CALL CATBNR(POISN,DD(J,8),DD(J,9),CDI(I,23),CDI(I,4) ,EFF)

STORE EFF IN MAT DD

DD(J,22)=EFF

ELSEIF (NINT(DD(J,3)).EQ.8) THEN

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)

STORE EFF IN MAT DD

DD(J,20)=EFF

ELSEIF (NINT(DD(J,3)).EQ.9) THEN

SUBROUTINE FOR DEVICE TYPE 9-DUMMY

EFF=0

STORE EFF IN MAT DD

DD(J,20)=EFF

ELSE

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
SUBROUTINE CONDHX(BEDQ, EMAX, MLIQ, MW, VCONC, SOL, CAVPRD, EFF)
REAL MW, MLIQ, H, PA, XA, NOUT

EFF = REMOVAL EFF (DEC)

CAIN = CAVPRD
IF (CAIN.LE.1E-10) THEN
    CAIN = 0.1E-10
ENDIF
IF (SOL.LE.1E-10) THEN
    EFF = 0
ELSE
    CONTAMINANT IS AMMONIA - USES EXPERIMENTAL DATA FROM JSC-08797
    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
        CONTAMINANT IS NOT AMMONIA
        CALCULATE CONTAMINANT PARTIAL PRESSURE AND WATER MOLE FRACTION
        PA = CAIN*1.0E-9*82.06*278/MW
        XA = (PA/I)/(MLIQ*(1000/18))/(BEDQ*(1000/22.4))+SOL/I
        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

******************************** END OF SUBROUTINE CONDHX **************************
SUBROUTINE CONVRG(I,TN,TNI,CAVPRD,DD,NROW,NCOL,
+CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CACULC,CFINAL, CEQULIB,KK,LIN,LIN2,
+N,PRTSW3,IMSGDN)
INTEGER NROW,NCOL,NROWI,NCOLI,NROW2,NCOL2,PRTSW3,KK
CHARACTER NN(NROWI)*30
REAL DD(NROW,NCOL)
REAL CC(NROWI,NCOLI)
REAL CDI(NROW2,NCOL2)
REAL XI,X2,YI,Y2,SLOPE

19 C SUBROUTINES REQUIRED:
20 C PRAFIL-ZERO MAT DD COL 17-21
21 C CNRSUB- USING CAV PRED CALC-REM EFF FOR ALL DEVICES-PUT IN DD COL 20
22 C MASBAL-CALC CAV CALC,CFINAL,CEQULIB,M.REM
23 C INPUTS:
24 C FROM MCALC
25 C I=CONTAMINANT NO.
26 C TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
27 C DD,NROW,NCOL=NAME & DIM OF MAT DD
28 C CC,NROW1,NCOL1=NAME & DIM OF MAT CC
29 C CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
30 C LIN=NO. OF CONTAMINANTS IN MAT CDI
31 C LIN2=NO. DEVICES IN MAT DD
32 C IMSGDN=DEVICE NO. FOR MESSAGE OUTPUT
33 C CAVPRD=PREDICTED CABIN CONC FOR INCREMENT (MG/CU M)
34 C CFINAL=CABIN CONC (MG/CU M)
35 C CEQULIB=EQUILIBRIUM CABIN CONC (MG/CU M)
36 C M.REM IS IN MAT DD COL 21
37 C TO PRAFIL
38 C NAME & SIZE OF MATRIX + FIRST AND LAST COL TO BE ZEREOED
39 C TO CNRSUB
40 C TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
41 C DEVICE AVERAGE CONT CONC (MG/M3) = DD(J,22)
42 C DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
43 C CC,NROW1,NCOL1=NAME & DIM OF MAT CC
44 C CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
45 C LIN2=NUMBER OF DEVICES IN MAT DD
CONVERGENCE ERROR (DEC)
CNVERR=DD(1,12)

IF(CNVERR.LT.IE-10) THEN
  OPEN(IMSGDN, FILE='CON', IOSTAT=IOVAL)
  WRITE(IMSGDN, *) 'CONV ERROR<IE-10: PROGRAM TERMINATED'
  CLOSE(IMSGDN)
  STOP
ENDIF

DO 100 KK=1,20
  ZERO MAT DD COL 17-21
  CALL PROFIL(DD,NROW,NCOL,17,21)
  USING CAVPRD FIND REM EFF OF EACH DEV & PUT IN DD COL 20
  CALL CNRSUB(I,TN, TNI,DD,NROW,NCOL,CC,NROW1,NCOL1,
  + CDI,NROW2,NCOL2,LIN2,KK)
  FIND CAVCLC FOR THESE REMOVAL EFFICIENCIES
  CALL MASBAL(I,TN, TNI,DD,NROW,NCOL,CC,NROW1,NCOL1,
  + CAVCLC,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
  IF PRTSW3=1 THEN PRINT NAME & NO + CONV VALUES
  IF (PRTSW3.EQ.1) THEN
    OPEN(IMSGDN, FILE='CON', IOSTAT=IOVAL)
    WRITE(IMSGDN, 50) I,NN(I)
    FORMAT(1X,'CONT NO.= ',I4,2X,A)
    WRITE(IMSGDN,*) 'CAVPRD,CAVCLC= ',CAVPRD,CAVCLC
    CLOSE(IMSGDN)
  ENDIF
  IF CAVCLC=CAVPRD THEN EXIT THE KK LOOP
  IF (KK.EQ.I) THEN
    INITIALIZE X2 AND Y2 FOR THE NEXT ITERATION
    X2=CAVPRD
    Y2=CAVCLC-CAVPRD
    CAVPRD=(CAVPRD+CAVCLC)/2
  ELSE
    USE THE NEWTON-RAPHSON METHOD FOR ITERATIONS WHERE KK>I
    XI=X2
    YI=Y2
    X2=CAVPRD
    Y2=CAVCLC-CAVPRD
    SLOPE=(Y2-YI)/(X2-XI)
    CAVPRD=X2-0.95*Y2/SLOPE
  ENDIF
ENDIF

IF (CAVCLC=CAVPRD) THEN EXIT THE KK LOOP
IF(CAVPRD.LE.IE-10) THEN EXIT THE KK LOOP
IF(CAVPRD.LT.IE-10) GOTO 80
IF(ABS((CAVPRD-CAVCLC)/CAVPRD).LT.CNVERR) THEN
GOTO 101
ENDIF

CONVERGENCE CALCULATION ROUTINE
USE THE BISECTION METHOD FOR THE ITERATION WHERE KK=1

080 IF (KK.EQ.1) THEN
  INITIALIZE X2 AND Y2 FOR THE NEXT ITERATION
  X2=CAVPRD
  Y2=CAVCLC-CAVPRD
  CAVPRD=(CAVPRD+CAVCLC)/2
  ELSE
    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

SET CAV IN PRED DD(I,22) = CAV IN CALC DD(I,17)
119     DO 90 J=1,LIN2
120        DD(J,22)=DD(J,17)
121 090 CONTINUE
122 C
123 100 CONTINUE
124 C   END OF KK LOOP
125 101 CONTINUE
126 C
127
128    RETURN
129 C   *********** END OF SUBROUTINE CONVRG   *******************
130   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
SUBROUTINE COOXID(BEDQ, EMAX, BEDL, BEDDIA, MW, EFF)
REAL MW

WORKS ONLY FOR CO MW=28.01 OR H2=2.02; OTHERWISE REM EFF=0
IF ((MW.EQ.28.01).OR. (MW.EQ.2.02)) THEN
  EFF=EMAX
ENDIF

IF RESIDENCE TIME <0.2 SEC THEN REM EFF DROPS LINEARLY
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

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

*****************************************************************
* SUBROUTINE COOXID *
* CALCULATES EFFICIENCY OF CO OXIDIZER (Pt on charcoal) *
*****************************************************************

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
* SUBROUTINE CRIN *
* SUBROUTINE TO READ STRING OF LENGTH 30 INTO MAT NN *
* AND READ REAL DATA INTO MAT XX(ROW, COL) *
* RETURNS NUMBER OF LINES OF DATA READ FROM FILE *

**NOTE: INPUT STRING MUST HAVE SINGLE QUOTES AROUND IT**
**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(*, '(A)') ' IOERROR= ', IOVAL
CLOSE (1)
WRITE(*,*) ' WHAT IS THE INPUT FILE NAME? '
GOTO 10
990 RETURN

END
SUBROUTINE CROUT(NN, XX, NROW, NCOL, FSTCOL, LSTCOL, LIN, FSTLIN, LSTLIN, NINC, IOVAL)
INTEGER NROW, NCOL, IOVAL, FSTCOL, LSTCOL, LIN, FSTLIN, LSTLIN, NINC, 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 (FSTLIN.GT.LIN) FSTLIN=LIN
IF (LSTLIN.GT.LIN) LSTLIN=LIN
IF (FSTLIN.GT.LSTLIN) FSTLIN=LSTLIN

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(I, FILE=FNAME, IOSTAT=IOVAL)
FORMAT ('INCREMENT NO. = ',I7)
DO 10 I=FSTLIN, LSTLIN
WRITE(I,60, IOSTAT=IOVAL) I, NN(I)
WRITE(IDEVNO, 60, IOSTAT=IOVAL) I, NN(I)
FORMAT(I1X, 'CONT NO.: ',I4,2X,A)
WRITE(I,70, IOSTAT=IOVAL) (XX(I,J), J=FSTCOL, LSTCOL)
WRITE(IDEVNO,70, IOSTAT=IOVAL) (XX(I,J), J=FSTCOL, LSTCOL)
10 CONTINUE
CLOSE (I)
CLOSE (IDEVNO)
990 RETURN

C 010 OPEN (IMSGDN, FILE='CON', IOSTAT=IOVAL)
C 020 WRITE(IMSGDN, ' (A) ') ' WRITE TO LPT1 OR CON OR END '
C 025 CLOSE (IMSGDN)
C 030 READ(*, ' (A) ') FNAME
C 035 QUIT IF FNAME='END'
C 040 IF (FNAME.EQ.'END') GO TO 990
C 045 IF ((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
C 050 OPEN(I, FILE=FNAME, IOSTAT=IOVAL)
C 055 FORMAT ('INCREMENT NO. = ', I7)
C 060 DO 10 I=FSTLIN, LSTLIN
C 065 WRITE(I,60, IOSTAT=IOVAL, ERR=900) I, NN(I)
C 070 WRITE(IDEVNO, 60, IOSTAT=IOVAL, ERR=900) I, NN(I)
C 075 FORMAT(I1X, 'CONT NO.: ',I4,2X,A)
C 080 WRITE(I,70, IOSTAT=IOVAL, ERR=900) (XX(I,J), J=FSTCOL, LSTCOL)
C 085 WRITE(IDEVNO,70, IOSTAT=IOVAL, ERR=900) (XX(I,J), J=FSTCOL, LSTCOL)
C 090 CONTINUE
C 095 CLOSE (I)
C 100 CLOSE (IDEVNO)
C 990 RETURN

C *********** END OF SUBROUTINE CROUT ***********
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*I
CHARACTER NN(NROW)*30
REAL XX(NROW, NCOL)
IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
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 I=FSTLIN, LSTLIN
WRITE(1,70,IOSTAT=IOVAL, ERR=900) (XX(I,J), J=FSTCOL, LSTCOL)
070 FORMAT(IX,TGII.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

*********** 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
SUBROUTINE DATOUT

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

SUBROUTINE DATOUT(TN,TNI,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

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

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

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

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

PRINT OUT CONCENTRATION DATA
CALL PRCDA(TN, TNI, LIN, CC, NROW1, NCOL1, CDI, NROW2, NCOL2, NN,
+ IDEVNO, NINC, IMONTH, IDAY, IYEAR, IHOUR, IMINUTE, FNAME, IOVAL, IPGCTR,
+ PRTSW8, PRTSW9, FCPLOT, IDEVN1)

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

****************************************************************
* SUBROUTINE GROUP *
* PROGRAM TO PRINT THE GROUP TOXICITY LEVELS AND T LEVEL *
****************************************************************

SUBROUTINE GROUP(TN,TNI,LIN,CC,NROWI,NCOLI,CDI,NROW2,NCOL2,NN,
+IDEVNO,NINC,IMONTH,IDAY,IYEAR,IMINUTE,FNAME,IOVAL,
+IPGCTR,TVAL,IDEVNT,PRTSW8)

DIMENSION GL(16)
REAL CC(NROWI,NCOLI)
REAL CDI(NROW2,NCOL2)
REAL TLEVL
INTEGER TVAL,PRTSW8,NINC,IDEVNO,IDEVNT

Determine the sums for each group level
DO 25 J=1,16
   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

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) ' GROUP T-VALUES AS SPECIFIED IN NHB 8060.1B APPENDIX D' 
   WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) '+ -01- -02- -03- -04- -05- -06- -07- -08- -09- + -10- -11- -12- -13- -14- -15- -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,*,IOSTAT=IOVAL,ERR=900) OALLT
   IF (NINC.NE.-1) THEN
      WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) 'IO ERROR IN GROUP=
      RETURN
   ENDIF
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
C FILE: HEADGS.FOR

*-------------------------------------------------------------*
* SUBROUTINE HDGI                                          *
* PROGRAM TO PRINT HEADING-DATE, TIME, FILE NAME, & PAGE NO. *
*-------------------------------------------------------------*

NOTE: FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

SUBROUTINE HDGI(IMONTH, IDAY, IYEAR, IHOUR, IMINUTE,
+FNAME, IPGNO, IDEVNO)
IMONTH...ISECOND=TIME AND DATE NAMES
FNAME: FILE NAME
IPGNO=PAGE NUMBER
IDEVNO=DEVICE NUMBER FOR OUTPUT
CHARACTER FNAME*24

WRITE HEADING
IF (IPGNO .EQ. 1) THEN
WRITE (IDEVNO, 5, IOSTAT=IOVAL, ERR=900)
005 FORMAT (2X, 'PROGRAM VERSION 8.1 Alpha', 5X, 'March 15, 1994', /)
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 HDGI= ', IOVAL
999 RETURN
END

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

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

SUBROUTINE DATTM(IMONTH, IDAY, IYEAR, IHOUR, IMINUTE, ISECOND)
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
SUBROUTINE HDG2(INCRNO, TNI, TN, IDEVNO)

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

WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900) INCRNO, TNI, TN

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

SUBROUTINE HDG3(IFLAG, TNI, TN, IDEVNO)

INPUTS:
IFLAG = FLAG FOR TIME INCREMENT (1=PCALC, 2=FINAL)
TNI = INCREMENT INITIAL TIME (HRS)
TN = INCREMENT FINAL TIME (HRS)
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

WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900) INAME, TNI, TN

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

SUBROUTINE HDG2(INCRNO, TNI, TN, IDEVNO)

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

WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900) INCRNO, TNI, TN

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

SUBROUTINE HDG3(IFLAG, TNI, TN, IDEVNO)

INPUTS:
IFLAG = FLAG FOR TIME INCREMENT (1=PCALC, 2=FINAL)
TNI = INCREMENT INITIAL TIME (HRS)
TN = INCREMENT FINAL TIME (HRS)
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

WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900) INAME, TNI, TN

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

SUBROUTINE HDG2(INCRNO, TNI, TN, IDEVNO)

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

WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900) INCRNO, TNI, TN

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

SUBROUTINE HDG3(IFLAG, TNI, TN, IDEVNO)

INPUTS:
IFLAG = FLAG FOR TIME INCREMENT (1=PCALC, 2=FINAL)
TNI = INCREMENT INITIAL TIME (HRS)
TN = INCREMENT FINAL TIME (HRS)
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

WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900) INAME, TNI, TN

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

****************************************************************
* SUBROUTINE HDG4
* PROGRAM TO PRINT HEADING-CONT NO., NAME, FINAL CABIN CONC
* MAC, EXCEEDS MAC
****************************************************************
NOTE: FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

SUBROUTINE HDG4 (IDEVNO)

INPUTS:
IDEVNO=DEVICE NUMBER FOR OUTPUT

WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900)
010 FORMAT(IX, 'CONT. 14X, .NAME.,14X .FINAL CABIN',5X, 'MAC',5X,
+ 'EXCEEDS')
WRITE(IDEVNO, 20, IOSTAT=IOERR, ERR=900)
020 FORMAT(IX,' NO. ',32X, 'CONC (MG/M3) ',I2X,' MAC '
GO TO 999
900 WRITE(*,*) 'IO ERROR IN HDG4= ',IOVAL
999 RETURN
END

****************************************************************
* SUBROUTINE HDG5
* PROGRAM TO PRINT HEADING-TOTAL CONT REMOVED BY EACH DEV (MG)
* PRINTS SHEET I-NO,NAME,CABIN,LEAK,&DEV3..DEV8
****************************************************************
NOTE: FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

SUBROUTINE HDG5(IDEVNO)

INPUTS:
IDEVNO=DEVICE NUMBER FOR OUTPUT

WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900)
010 FORMAT(IX,24X,'TOTAL CONTAMINANT MASS REMOVED BY EACH DEVICE (MG) '+
 SHEET I')
WRITE(IDEVNO,20, IOSTAT=IOVAL, ERR=900)
020 FORMAT(IX,' NO.',I4X, 'NAME',I6X, 'CABIN',9X, 'LEAK.,SX, .DEV3. '+
 DEV4',8X, 'DEV5',8X, 'DEV6',SX, '.DEV7..DEV8.
GO TO 999
900 WRITE(*,*) 'IO ERROR IN HDG4= ',IOVAL
999 RETURN
END
GO TO 999
900 WRITE(*,*) 'IO ERROR IN HDG5= ', IOVAL
999 RETURN
END

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

NOTE: FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

SUBROUTINE HDG6(IDEVNO)

INPUTS:
IDEVNO=DEVICE NUMBER FOR OUTPUT

WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900)
010 FORMAT(IX,24X,'TOTAL CONTAMINANT MASS REMOVED BY EACH DEVICE (MG)
+ SHEET 2')
WRITE(IDEVNO,20, IOSTAT=IOERR, ERR=900)
020 FORMAT(IX,' NO.',I4X, 'NAME',I6X,' 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

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

NOTE: FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE

SUBROUTINE HDG7 (IDEVNO)

INPUTS:
IDEVNO=DEVICE NUMBER FOR OUTPUT

WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900)
010 FORMAT(IX,24X,'DEVICE REMOVAL EFFICIENCY AT END OF TIME INCREMENT
+ (DEC)')
WRITE(IDEVNO,20, IOSTAT=IOERR, ERR=900)
020 FORMAT(IX,' NO.',I4X, 'NAME',I5X,'#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

SUBROUTINE HDG8
INPUTS:
IDEVNO=DEVICE NUMBER FOR OUTPUT
WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900)
010 FORMAT(IX,24X,'RATE OF CONTAMINANT REMOVAL-EACH DEVICE (MG/HR)', + SHEET 1')
WRITE(IDEVNO,20, IOSTAT=IOERR, ERR=900)
020 FORMAT(IX,' NO.',I4X,'NAME',I6X,'DEV9',8X,'DEV10',7X,'DEV11',7X,'DEV12',7X,'DEV13',7X,'DEV14',7X,'DEV15')
GO TO 999
900 WRITE(*,*)'IO ERROR IN HDG8= ',IOVAL
999 RETURN
END

SUBROUTINE HDG9
INPUTS:
IDEVNO=DEVICE NUMBER FOR OUTPUT
WRITE HEADING
WRITE(IDEVNO, 10, IOSTAT=IOVAL, ERR=900)
010 FORMAT(IX,24X,'RATE OF CONTAMINANT REMOVAL-EACH DEVICE (MG/HR)', + SHEET 2')
WRITE(IDEVNO,20, IOSTAT=IOERR, ERR=900)
020 FORMAT(IX,' NO.',I4X,'NAME',I6X,'DEV9',8X,'DEV10',7X,'DEV11',7X,'DEV12',7X,'DEV13',7X,'DEV14',7X,'DEV15')
GO TO 999
900 WRITE(*,*)'IO ERROR IN HDG8= ',IOVAL
999 RETURN
END
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:
I=CONTAMINANT NUMBER
DD, NROW, NCOL=NAME AND DIMENSIONS OF MAT DD
CDI, NROW2, NCOL2=NAME AND DIMENSIONS OF MAT CDI
LIN2=NUMBER OF DEVICES IN MAT DD

C OUTPUT
LOADS INTERNAL GENERATION FROM MAT CDI INTO MAT DD COL 19

DD(1,19)=CDI(I,1)
DD(2,19)=0
DO 10 J=3, LIN2
   DD(J,19)=CDI(I,J+7)
10 CONTINUE
RETURN
END

C ************ END OF SUBROUTINE LDIGEN

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
**SUBROUTINE LODEFF**

*SUBROUTINE TO LOAD LAST INCR EFF FROM MAT CC INTO MAT DD COL 20*

*USES ADJUSTABLE SIZE ARRAYS*

**SUBROUTINE LODEFF(I,DD,NROW,NCOL,CC,NROWI,NCOLI,LIN2)**

**INPUTS:**

I=CONTAMINANT LINE NUMBER IN MAT CC
DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
CC,NROWI,NCOLI=NAME & DIMENSIONS OF MAT CC
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

**CONTINUE**

**RETURN**

**END**

**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
**MASS BALANCE SUBROUTINE-MASBAL**

* FOR 1 CONT AT A TIME AT A GIVEN DEVICE EFFICIENCY CALCULATES *
* CAV, CFINAL, CEQ, M.REMOVED (ALL DEV+CABIN) - DATA PUT IN MAT DD *

**NOTE:** BEFORE RUNNING THIS SUBROUTINE MUST ZERO MAT DD COL 17-21
(DONE BY PRAFIL) & LOAD REM EFF FOR EACH DEVICE INTO
MAT DD COL 20 (DONE BY LODEFF OR CNRSUB)

**INPUTS:**
- I = CONTAMINANT NO.
- TN = INCREMENT END TIME (HRS); TNI = INCR BEGINNING TIME HRS
- DD, NROW, NCOL = NAME AND SIZE OF MAT DD
- CC, NROW1, NCOL1 = NAME AND SIZE OF MAT CC
- CDI, NROW2, NCOL2 = NAME AND SIZE OF MAT CDI
- LIN = NO. OF CONT IN MAT CDI
- LIN2 = NO. OF DEVICES IN MAT DD

**OUTPUTS:**
- CAVCLC = CALC INCR CABIN CONT CONC (MG/CU M)
- CEQLIB = CABIN EQUILIB CONC (MG/CU M)
- CFINAL = INCREMENT FINAL CABIN CONCENTRATION (MG/CU M)
- DD(I,19) = 50 (CABIN M.GEN)
- OTHER DEVICES DD(2-15,19) MUST = 0 AT THIS POINT (SEE PRAFIL)
SQEFFN=SUM OF Q*REM EFF NET FOR ALL DEVICES (CU M/HR)

CVOL=CAVIN VOL (CU M)=DD(I,9)

SMNTC=SUM OF MASS NET TO CABIN(MG/HR)

CABIN VOL (CU M)

CVOL=DD(I,9)

CINITIAL (MG/CU M)

CINIT=CC(I,1)

EVALUATE SUM Q*REM EFF NET USING MGEN IN DEVICES=0 (DD COL 19)

SET CABIN AVERAGE CONCENTRATION = TO ARBITRARY VALUE OF 100

AND INTERNAL GENERATION IN DEVICES =0 (NOT YET LOADED)

SET CABIN MGEN=ARBITRARY VALUE OF 50 (DD(I,19))

SMGEN=SUM MGEN IN ALL DEVICES +CABIN (MG/HR)

SMREM=SUM M REMOVED BY ALL DEVICES (MG/HR)

CAV=CABIN AVERAGE CONCENTRATION (MG/CU M)

CAV=100

DD(I,19)=50

CALL CALCM/DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2

SQEFFN=SUM OF Q*REMOVAL EFF NET (MG/HR)

LOAD INTERNAL GENERATION FOR ALL DEV+CABIN FROM CDI INTO DD COL 19

CALL LDIGEN(I,DD,NROW,NCOL,CDI,NROW2,NCOL2,LIN2

EVALUATE SUM OF M.NET TO CABIN=MGEN CABIN+SUM MGEN ALL DEVICES -

SUM MREM ALL DEVICES

SMNTC=SUM M.NET TO CABIN=AMT GEN WHICH GETS TO THE CABIN DIRECTLY

SET C CABIN AV=0

CAV=0

GET SUM MASS GEN CABIN+ INTERNAL DEVICES AND SUM MASS REMOVED ALL

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

NOTE:SMNTC IS ALSO PUT IN DD(1,21) BY CALCM

GET CALCULATED CABIN EQUILIBRIUM CONCENTRATION (CAVCLC) (MG/CU M)

IF(SQEFFN.LT.1E-10) THEN

IF(CVOL.EQ.0) THEN

CFINAL=1E10

ELSE

CFINAL=CINIT+(TN-TNI)*SMGEN/CVOL

ENDIF

CAVCLC=(CINIT+CFINAL)/2

CEQLIB=IE0

ELSE

CEQLIB=SMNTC/SQEFFN

CALCULATE CAVCLC AND CFINAL FROM SUBROUTINE

CALL PCAVCF(TN,TNI,CEQLIB,SQEFFN,CVOL,SMNTC,CINIT, + CAVCLC,CFINAL)

ENDIF

USING CAV CALC EVALUATE MREM FOR CABIN + DEVICES AND PUT

IN MAT DD COL 21

CAV=CAVCLC

CALL CALCM/DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2

ENDIF
119 RETURN
120 C END OF SUBROUTINE MASBAL
121 C ******************** END OF SUBROUTINE MASBAL ********************
122 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
SUBROUTINE MCALC

* MAIN CALCULATION LOOP SUBROUTINE FOR 1 TIME INCREMENT
* FOR ALL CONTAMINANTS ONE AT A TIME
* BASED ON SUM MASS REM LAST INCR, FOR EACH CONT
* CALCULATE NEW REMOVAL EFF, CAV CALC CABIN,
* CEQUILIB, CFINAL, & M.REMOVED ALL DEVICES-PUT IN MAT CC

SUBROUTINE MCALC(I,TN,TNI,DD,NROW,NCOL,
+CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,LIN,LIN2,
+PRTSW2,PRTSW3,PRTSW4,IMSGDN)
INTEGER NROW,NCOL,NROWI,NCOLI,NCOL2,PRTSW2,PRTSW3,PRTSW4,KK
CHARACTER NN(NROWI)*30
REAL DD(NROW,NCOL)
REAL CC(NROWI,NCOLI)
REAL CDI(NROW2,NCOL2)

SUBROUTINES REQUIRED:
PREDCT=PREDICT CAV BASED ON M.GEN OF THIS INCR & REM EFF OF LST INC
CONVRG=CALC CAV CALC,CEQ,CFINAL,M.REM,REM EFF
CROUT=PRINT TEST VALUES OF MAT CC
RROUT=PRINT TEST VALUES OF MAT DD

INPUTS:
FROM MAIN PROG
I=CONTAMINANT NO.
TN,TNI =INCREMENT END & BEGINNING TIME (HRS)
DD,NROW,NCOL=NAME & DIM OF MAT DD
CC,NROW1,NCOL1=NAME & DIM OF MAT CC
CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
NN=NAME OF MAT NN
LIN=NUMBER OF CONTAMINANTS IN MAT NN & CDI
LIN2=NO. DEVICES IN MAT DD
FROM PREDCT
CAVPRD=PRED CABIN CONT CONC (MG/CU M)
FROM CONVRG
CAVCLC=CALC CABIN CONT CONC (MG/CU M)
CFINAL=FINAL INCR CABIN CONT CONC (MG/CUM)
KK=COUNTER FOR CONVERGENCE
CEQLIB=EQUILIBRIUM CABIN CONT CONC(MG/CU M)
IMSGDN=DEVICE NO FOR MESSAGE AND TEXT PRINTOUT OUTPUT

OUTPUTS:
TO PREDCT
I=CONTAMINANT NO.
TN,TNI =INCREMENT END & BEGINNING TIME (HRS)
DD,NROW,NCOL=NAME & DIM OF MAT DD
CC,NROW1,NCOL1=NAME & DIM OF MAT CC
CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
LIN=NO. OF CONTAMINANTS IN MAT CDI
LIN2=NO. DEVICES IN MAT DD
TO CONVRG
I=CONTAMINANT NO.
TN,TNI =INCREMENT END & BEGINNING TIME (HRS)
DD,NROW,NCOL=NAME & DIM OF MAT DD
CC,NROW1,NCOL1=NAME & DIM OF MAT CC
CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
LIN=NO. OF CONTAMINANTS IN MAT CDI
LIN2=NO. DEVICES IN MAT DD
CAVPRD=PREDICTED CABIN CONC FOR INCREMENT (MG/CU M)

PUTS CAVCLC,CEQLIV,&CFINAL IN CC(I,2-3 &4)

PUTS REM EFF FROM DD COL20 IN CC(I,7-10-13 ETC)

PUTS M.REM FOR EACH DEV FROM DD COL21 IN CC(I,6-9-12 ETC)

PUTS SUM MASS REM FOR EACH DEV IN CC(I,8-11-14 ETC)

BEGIN LOOP FOR EACH CONTAMINANT FOR EACH TIME INCREMENT

DO I00 I=1,LIN

CALC CAV PRED CABIN FOR CONT BASED ON REM EFF OF LAST INCREMENT 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)

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)

IF KK>20 THEN BEGIN 1/20 TIME INCREMENT CONVERGENCE ROUTINE

IF (KK.GT.20) THEN

KK=1

BEGIN 1/20 INCREMENT CONVERGENCE ROUTINE

NEW INCREMENT INITIAL TIME (HRS)

TN1NEW=TN1

NEW TIME INCREMENT (HRS)

BINew=(TN-TN1)/20

BEGIN LOOP FOR 1/20 INCREMENT SIZE TIME INCREMENT

NEW INCREMENT FINAL TIME (HRS)

TNNEW=TN1NEW+BINew

ZERO MAT DD COL 17-21

CALL PRAFIL(DD,NROW,NCOL,17,21)

LOAD EFFICIENCY FROM LAST INCREMENT INTO MAT DD COL 20

CALL LODEFF(I,DD,NROW,NCOL,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,LIN,
+LIN2,NN)

CALC CAV PRED CABIN FOR CONT BASED ON REM EFF OF LAST INCREMENT 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)

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)

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, 50) I,NN(I),TN1NEW,TNNEW

FORMAT (IX,'CONT NO.= ',I4,2X,A,/,IX,
'FOR INCREMENT INIT & FINAL TIMES= ',FS.2,FS.2)

CLOSE(IMSGDN)

ENDIF

ENDIF

FILL MAT CC WITH RESULTS
PUT CAVCLC, CEQLIB, AND CFINAL IN CC

CC(I, 2) = CAVCLC
CC(I, 3) = CEQLIB
CC(I, 4) = CFINAL

PUT REM EFF FROM LAST ITER DD COL 20 IN CC(I, 7-10-13-ETC)

K = 7
DO 102 J = 2, LIN2
CC(I, K) = DD(J, 20)
K = K + 3
102 CONTINUE

TAKE CABIN M.REM(MG/HR) FROM DD(1, 21) & PUT IN MAT CC(I, 5)

CC(I, 5) = DD(1, 21)

TAKE M.REM FROM DD COL 21 & PUT IN CC(I, 8-11-14-ETC)

K = 8
DO 103 J = 2, LIN2
CC(I, K) = DD(J, 21)
K = K + 3
103 CONTINUE

CALCULATE SUM OF MASS REMOVED IN DEVICES + CABIN TO DATE AND

PUT IN CC(I, 6-9-12-ETC)

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

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 INC'
+ LOOP OF MCALC'
WRITE (IMSGDN, *) 'INFO FROM MAT CC'
CLOSE (IMSGDN)
CALL CROUT(NN, CC, NROW1, NCOL1, I, I, I, I, IMSGDN)
ENDIF

REPEAT LOOP FOR 1/10 INCREMENT IF END OF 1/20 INCREMENT TIME

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

END OF CONVERGENCE ROUTINE

CALC SUM MASS REMOVED & FILL MAT CC WITH RESULTS

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

TAKE CABIN M.REM(MG/HR) FROM DD(I,21) & PUT IN MAT CC(I,5)
CC(I,5)=DD(I,21)
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

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

IF PRTSW4=I THEN PRINT MAT DD+MAT CC INFO FOR THIS CONTAMINANT
IF (PRTSW4.EQ.I) 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 crount(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,I,1,IMSGDN)
open (IMSGDN, FILE='CON', IOSTAT=IOVAL)
write(IMSGDN,*) 'INFO FROM MAT DD'
close(IMSGDN)
call rrount(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN)
endif

END OF I LOOP FOR EACH CONTAMINANT
220 C
221 100 CONTINUE
222 RETURN
223 C ***** END OF SUBROUTINE MCALC ****************************
224 END
SUBROUTINE PCAVCF(TN, TNI, SCEQLIB, SQEFFN, CVOL, SMNTC, CINIT, CAVCLC, SCFINAL)

DOUBLE PRECISION EXPON, CEQLIB, CFINAL
CEQLIB=DBLE(SCEQLIB)

SUBROUTINES REQUIRED: NONE

INPUTS:
TN, TNI=INITIAL & FINAL INCREMENT TIME (HRS)
SCEQLIB=CEQLIB=EQUILIBRIUM CABIN CONC (MG/CUM)
SQEFFN=SUM Q'REMOVAL EFF NET (MG/HR)
CVOL=CABIN VOLUME (CU M)
SMNTC=SUM MASS CONT NET TO CABIN (MG/HR)
CINIT=INITIAL INCREMENT CONT CONC (MG/CU M)

OUTPUTS:
SCAVCLC(CAVCLC)=CALC AVERAGE CABIN CONC (MG/CU M)
SCFINAL(CFINAL)=FINAL INCREMENT CONC (MG/CU M)

IF(CVOL.LE.0) THEN
CAVCLC=CEQLIB
CFINAL=CEQLIB
GOTO 99
ENDIF

CALCULATION FOR CFINAL
EXPON=(TN-TNI)*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-TNI)/CVOL
CEQLIB=1E10
GOTO 99
ELSE
CFINAL=CINIT+(SMNTC/SQEFFN-CINIT)*((1-EXP(-EXPON)))
ENDIF

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
ENDIF
ELSE
CAVCLC=CEQLIB-(CFINAL-CINIT)/LOG((CEQLIB-CINIT)/
+ (CEQLIB -CFINAL))
ENDIF

SCFINAL=REAL(CFINAL)

CONTINUE
59       RETURN
60       C     *********** END OF SUBROUTINE PCAVCF ****************************
61       END
62       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
**SUBROUTINE PCSET**

SUBROUTINE PCSET, SUBROUTINE FOR PRECALCULATION SETUP ROUTINE

* FOR ALL CONTAMINANTS ONE AT A TIME *

* CALL EFF SUBROUTINES FOR DEVICES; GET CAV CABIN PRED, *

* CEQUILIB, CFINAL, & M.REMOVED ALL DEVICES-PUT IN MAT CC *

------------------------------------------------------------------

**SUBROUTINE PCSET(TNI,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,**

+CDI,NROW2,NCOL2,LIN2,NN, PRTSW1,IMSGDN)

** INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,**

** PRTSW1,KK **

** CHARACTER NN(NROWI)*30 **

** REAL DD (NROW, NCOL) **

** REAL CC (NROWI _ NCOLI) **

** REAL CDI(NROW2,NCOL2) **

** KK=0 **

** SUBROUTINES REQUIRED: **

** PRAFIL-ZERO MAT DD COL 17-21 **

** CNRSUB-USING CAV=1E-20, FIND REMOVAL EFF AND PUT IN MAT DD COL 20 **

** MASBAL-CALC CAV CALC PRED, CEQ, CFINAL, M.REM **

** CROUT-TEST PRINTOUT OF CONT INFO **

** **

** INPUTS: **

** FROM MAIN CALC LOOP **

** TNI=INCREMENT INITIAL TIME (HRS) **

** LIN=NO. OF CONT IN MAT CC AND NN **

** DD,NROW,NCOL=NAME & SIZE OF MAT DD **

** CC,NROW1,NCOL1=NAME & SIZE OF MAT CC **

** CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI **

** LIN2=NO. DEVICES IN MAT DD **

** NN=NAME OF MAT NN **

** PRTSW1=PRINTSWITCH WHICH CONTROLS TEST PRINTOUT **

** IMSGDN=DEVICE NUMBER FOR MESSAGES AND TEST PRINTOUT **

** FROM PRAFIL **

** PUTS ZEROS IN MAT DD COL 17-21 **

** FROM CNRSUB **

** CNRSUB PUTS REM EFF(DEC) FOR EACH DEVICE IN MAT DD COL 20 **

** FROM MASBAL (PREDICTED VALUES) **

** CAVCLC=AVERAGE CABIN CONC (MG/CU M) **

** CFINAL=FINAL INCREMENT CONT CONC (MG/CU M) **

** CEQLIB=EQUILIBRIUM CONT CONC (MG/CU M) **

** M.REM FOR ALL DEVICES PLACED IN COL 21 OF MAT DD **

** OUTPUTS **

** TO MAIN PROGRAM **

** PUT IN MAT CC **

** CAVPRD=PRED CABIN AV CONC (MG/CU M) : =CC(I,2) **

** CEQLIB=EQUILIBRIUM CABIN CONT CONC (MG/CU M) : =CC(I,3) **

** CFINAL=FINAL CABIN CONT CONC (MG/CU M) : =CC(I,4) **

** PUTS REM EFF FROM DD COL 20 IN CC(I,7-10-13-16 ETC) **

** PUTS M.REM IN CC(I,5-8-11-14...) **

** TO PRAFIL **

** NAME AND SIZE OF MAT DD+FIRST & LAST COLUMN TO ZERO **

** TO CNRSUB **

** I=CONT NO. **

** TN,TNI=CONT INCREMENT FINAL,INIT TIME (HRS) **

** CAVPRD=CABIN AVERAGE CONT CONC (MG/CU M) **

** DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD **

** CC,NROW1,NCOL1=NAME & DIM OF MAT CC **

** CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI **
LIN2=NO. ACTIVE DEVICES IN MAT DD
I=CONT NO.
TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
CALC2=CALC CABIN AV CONC (MG/CU M)
CC,NROW1,NCOL1=NAME & DIM OF MAT CC
CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
CFINAL=CABIN FINAL CONCENTRATION (MG/CU M)
CEQLIB=CABIN EQUILIBRIUM CONCENTRATION (MG/CU M)
 LIN=NO. OF CONTAMINANTS IN MAT CDI
LIN2=NO. ACTIVE DEVICES IN MAT DD
TO RROUT
MATRX NAME,#ROWS,#COLS,FIRST & LAST COL TO PRINT,#LINES TO PRINT
BASIC TIME INCREMENT (HRS)
BINC=DD(I,1,11)
SET FINAL INCREMENT TIME (HRS)
TN=0.1*BINC/24
BEGIN LOOP FOR EACH CONTAMINANT - ONE AT A TIME
CALCULATE REM EFF FOR EACH DEVICE, GET M.REM, CAV CABIN CALC
CEQULIB, CFINAL-PUT IN MAT CC
DO 100 I=1,LIN
ZERO MAT DD COL 17 TO 21
CALL PRAFIL(DD,NROW,NCOL,17,21)
SET CAVPRD = MINIMUM VALUE TO ALLOW COMPUTATION
CAVPRD=IE-20
DD(J,22)=CAVPRD
CALC REMOVAL EFFICIENCIES (THROUGH EFF CALLING SUBROUTINE)
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,KK)
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
PUT CAVPRD,CEQLIB, CFINAL IN MAT CC
CC(I,2)=CAVPRD
CC(I,3)=CEQLIB
CC(I,4)=CFINAL
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
CONTINUE
TAKE M.REMOVED FROM MAT DD COL 21 AND PUT IN MAT CC(I,5-8-...)
CLOSE (IMSGDN)
CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,1,I,1,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
**SUBROUTINE PRAFIL**

**SUBROUTINE TO FILL ADJUSTABLE SIZE REAL ARRAY WITH ZEROS**

**PARTIAL FILL-FROM COL FSTCOL TO COL LSTCOL**

**SUBROUTINE PRAFIL(X,NROW,NCOL,FSTCOL,LSTCOL)**

**INTEGER NCOL,NROW,FSTCOL,LSTCOL**

**REAL X(NROW,NCOL)**

**INPUTS:**

- **X,NROW,NCOL** = NAME AND DIMENSIONS OF MATRIX X
- **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 II0 I=I,NROW**

**DO I00 J=FSTCOL,LSTCOL**

**X(I,J)=0.0**

**END OF SUBROUTINE PRAFIL **************************
**SUBROUTINE PRCDA**

*PROGRAM TO PRINT ANSWERS FOR CONCENTRATION DATA*

**NOTES:**
1. FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
2. IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED

**SUBROUTINE PRCDA**(TN, TNI, LIN, CC, NROW1, NCOL1, CDI, NROW2, NCOL2, NN,
                        +IDEVNO, NINC, IMONTH, IDAY, IYEAR, IHOUR, IMINUTE, FNAME, IOVAL, IPGCTR,
                        +PRTSW8, PRTSW9, FCLASS, IDEVNI)

**SUBROUTINES REQUIRED:**

HDGI, HDG2, HDG3, HDG4

**TN, TNI=FINAL AND INITIAL INCREMENT TIME (HRS)**

**LIN=TOTAL NUMBER OF CONTAMINANTS**

**CC,NROW1,NCOL1=NAME & SIZE OF MAT CC**

**CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI**

**NN=NAME OF MAT NN**

**IDEVNO=DEVICE NUMBER FOR OUTPUT**

**NINC=TIME INCREMENT NUMBER**

**=0 THEN PRINT HDG3 WITH PCALC**

**=-1 THEN PRINT HDG3 WITH FINAL**

**ELSE PRINT HDG2 WITH INCREMENT NUMBER**

**IMONTH..IMINUTE=TIME AND DATE INFO**

**FNAME=FILE NAME OUTPUT DATA IS STORED ON**

**IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT**

**REAL CC(NROW1,NCOL1)**

**REAL CDI(NROW2,NCOL2)**

**CHARACTER CNAME*30,FNAME*24,ECHR*1,FCLASS*24**

**CHARACTER NN(NROW1)*30**

**INTEGER PRTSW8,PRTSW9,NINC**

**ECHR=EXCEEDS MAC CHARACTER (Y OR N)**

**IF ((PRTSW8.EQ.I).OR. ((PRTSW8.EQ.0).AND. (NINC.EQ.-1))) THEN**

**INCREMENT PAGE COUNTER BY ONE**

**IPGCTR:IPGCTR+1**

**START FIRST PAGE**

**PRINT FORM FEED**

**WRITE(IODEVNO,20,IOSTAT=IOVAL,ERR=900)**

**FORMAT('1')**

**PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 4**

**WRITE(IODEVNO,40,IOSTAT=IOVAL,ERR=900)**

**FORMAT(1X,'')**

**CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)**

**IF(NINC.EQ.0) THEN**

**CALL HDG3(1,TNI,TN,IDEVNO)**

**ELSEIF(NINC.EQ.-1) THEN**

**CALL HDG3(2,TNI,TN,IDEVNO)**

**ELSE**

**CALL HDG2(NINC,TNI,TN,IDEVNO)**

**ENDIF**

**CALL HDG4(IDEVNO)**

**PRINT ANOTHER BLANK LINE**

**WRITE(IODEVNO,40,IOSTAT=IOVAL,ERR=900)**

89
BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
DO 100 I=1,LIN

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

IF CABIN CONC>MAC PRINT 'Y' OTHERWISE PRINT 'N'
IF(FCONC.GT.RMAC) THEN
  ECHR='Y'
ELSE
  ECHR='N'
ENDIF

PRINT 56 LINES OF DATA AND THEN START NEW PAGE
WRITE (IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,CNAME,FCONC,RMAC,ECHR
010 FORMAT (IX, I4, IX,A, IX,G11.4, IX, G11.4, 5X,A)

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
ENDIF

START SUBSEQUENT PAGES
WRITE (IDEVNO,50,IOSTAT=IOVAL,ERR=900)
050 FORMAT ('1')
PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &4
WRITE (IDEVNO,30,IOSTAT=IOVAL,ERR=900)
030 FORMAT (IX,'')
CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)

IF(NINC.EQ.0) THEN
  CALL HDG3(I,TNI,TN, IDEVNO)
ELSEIF (NINC.EQ.-1) THEN
  CALL HDG3(2,TNI,TN, IDEVNO)
ELSE
  CALL HDG2(NINC,TNI,TN, IDEVNO)
ENDIF
CALL HDG4 (IDEVNO)
PRINT ANOTHER BLANK LINE
WRITE (IDEVNO,30,IOSTAT=IOVAL,ERR=900)

100 CONTINUE
ENDIF

ENDIF
100 CONTINUE
ENDIF

****** 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 (IDEVNI,II0,IOSTAT=IOVAL,ERR=900) TNI,TN, (CC(J,4),J=IS,IE)
      120 CONTINUE
  ENDIF
ENDIF
WRITE (IDEVNO,30,IOSTAT=IOVAL,ERR=900)
10 CONTINUE
ENDIF

90
**SUBROUTINE PRREMI**

* PROGRAM TO PRINT ANSWERS - RATE OF CONTAMINANT REMOVAL (MG/HR) *

**NOTES:**

1. FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
2. IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED

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

**SUBROUTINES REQUIRED:**
HDGI, HDG2, HDG3, HDG8

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

INCREMENT PAGE COUNTER BY ONE
IPGCTR = IPGCTR + 1

START FIRST PAGE
PRINT FORM FEED
WRITE(IDEVNO, 20, IOSTAT=IOVAL, ERR=900)
FORMAT('1')

PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 8
WRITE(IDEVNO, 40, IOSTAT=IOVAL, ERR=900)
FORMAT('IX, ''')

CALL HDG1(IMONTH, IDAY, IYEAR, IHOUR, IMINUTE, FNAME, IPGCTR, IDEVNO)

IF(NINC.EQ.0) THEN
  CALL HDG3(1, TNI, TN, IDEVNO)
ELSEIF(NINC.EQ.-1) THEN
  CALL HDG3(2, TNI, TN, IDEVNO)
ENDIF
GO TO 999
999 RETURN
ELSE
   CALL HDG2(NINC,TN1,TN,IDEVNO)
ENDIF
CALL HDG8(IDEVNO)
PRINT ANOTHER BLANK LINE
WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
DO 100 I=1,LIN
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) )
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
   START SUBSEQUENT PAGES
   PRINT FORM FEED
   WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
   050 FORMAT('1')
   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(I,TNI,TN,IDEVNO)
   ELSEIF(NINC.EQ.-1) THEN
      CALL HDG3(2,TNI,TN,IDEVNO)
   ELSE
      CALL HDG2(NINC,TN1,TN,IDEVNO)
   ENDIF
   CALL HDG8(IDEVNO)
   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

SUBROUTINE PRREM2
* SUBROUTINE TO PRINT ANSWERS-RATE OF CONTAMINANT REMOVAL (MG/HR) *
* SHEET 2 *
**************************************************************************
NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
(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) 230

SUBROUTINES REQUIRED:

     HDG1, HDG2, HDG3, HDG9

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

INCREMENT PAGE COUNTER BY ONE
IPGCTR=IPGCTR+1

START FIRST PAGE
DON'T PRINT FORM FEED UNLESS NO. CONT > 20
IF(LIN.GT.20) THEN
PRINT FORM FEED
WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
020 FORMAT('1')
ENDIF

PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 9
WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
040 FORMAT(IX,')
CALL HDGI(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)
PRINT ANOTHER BLANK LINE
WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)

BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
DO I00 I=1,LIN
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(IX,I4,1X,A,7(IX,G11.4))

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

START SUBSEQUENT PAGES
PRINT FORM FEED
WRITE(IDEVO,50,IOSTAT=IOVAL,ERR=900)
FORMAT('1')
PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 8
WRITE(IDEVO,30,IOSTAT=IOVAL,ERR=900)
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)
PRINT ANOTHER BLANK LINE
WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
ENDIF
END CONTINUE
GO TO 999
WRITE(*,*), 'IO ERROR IN PRREM2=', IOVAL
999 RETURN
END

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

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

SUBROUTINES REQUIRED:
HDG1, HDG2, HDG3, HDG5

TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
LIN=TOTAL NUMBER OF CONTAMINANTS
CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
NN=NAME OF MAT NN
IDEVNO=DEVICE NUMBER FOR OUTPUT
NINC=TIME INCREMENT NUMBER
=0 THEN PRINT HDG3 WITH PCALC
=-1 THEN PRINT HDG3 WITH FINAL
ELSE PRINT HDG2 WITH INCREMENT NUMBER
IMONTH..IMINUTE=TIME AND DATE INFO
FNAME=FILE NAME OUTPUT DATA IS STORED ON
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(IODEVNO, 20, IOSTAT=IOVAL, ERR=900)

C PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 5
WRITE(IODEVNO, 40, IOSTAT=IOVAL, ERR=900)

CALL HDGI(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)

PRINT ANOTHER BLANK LINE
WRITE(IODEVNO, 30, IOSTAT=IOVAL, ERR=900)

C BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
DO I=1, LIN
  WRITE(IODEVNO, 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)

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
ENDIF

C START SUBSEQUENT PAGES
C PRINT FORM FEED
WRITE(IODEVNO, 50, IOSTAT=IOVAL, ERR=900)

CALL HDG1(IMONTH, IDAY, IYEAR, IHOUR, IMINUTE, FNAME, IPGCTR, IDEVNO)
IF(NINC.EQ.0) THEN
  CALL HDG3(I, TNI, TN, IDEVNO)
ELSEIF(NINC.EQ.-1) THEN
  CALL HDG3(2, TNI, TN, IDEVNO)
ELSE
  CALL HDG2(NINC, TNI, TN, IDEVNO)
ENDIF
CALL HDG5(IDEVNO)

PRINT ANOTHER BLANK LINE
WRITE(IODEVNO, 30, IOSTAT=IOVAL, ERR=900)
ENDIF
I00 CONTINUE

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
ENDIF

C START SUBSEQUENT PAGES
C PRINT FORM FEED
WRITE(IODEVNO, 50, IOSTAT=IOVAL, ERR=900)

CALL HDG1(IMONTH, IDAY, IYEAR, IHOUR, IMINUTE, FNAME, IPGCTR, IDEVNO)
IF(NINC.EQ.0) THEN
  CALL HDG3(I, TNI, TN, IDEVNO)
ELSEIF(NINC.EQ.-1) THEN
  CALL HDG3(2, TNI, TN, IDEVNO)
ELSE
  CALL HDG2(NINC, TNI, TN, IDEVNO)
ENDIF
CALL HDG5(IDEVNO)

PRINT ANOTHER BLANK LINE
WRITE(IODEVNO, 30, IOSTAT=IOVAL, ERR=900)
ENDIF
I00 CONTINUE

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
ENDIF

C START SUBSEQUENT PAGES
C PRINT FORM FEED
WRITE(IODEVNO, 50, IOSTAT=IOVAL, ERR=900)

CALL HDG1(IMONTH, IDAY, IYEAR, IHOUR, IMINUTE, FNAME, IPGCTR, IDEVNO)
IF(NINC.EQ.0) THEN
  CALL HDG3(I, TNI, TN, IDEVNO)
ELSEIF(NINC.EQ.-1) THEN
  CALL HDG3(2, TNI, TN, IDEVNO)
ELSE
  CALL HDG2(NINC, TNI, TN, IDEVNO)
ENDIF
CALL HDG5(IDEVNO)

PRINT ANOTHER BLANK LINE
WRITE(IODEVNO, 30, IOSTAT=IOVAL, ERR=900)
ENDIF
I00 CONTINUE

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

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

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

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

SUBROUTINES REQUIRED:
HDGI, HDG2, HDG3, HDG6

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

INCREMENT PAGE COUNTER BY ONE
IPGCTR=IPGCTR+1

START FIRST PAGE
DON'T PRINT FORM FEED UNLESS NO. CONT > 20
IF(LIN.GT.20) THEN
PRINT FORM FEED
WRITE(IDEVNO,20, IOSTAT=IOVAL, ERR=900)
FORMAT('1')
ENDIF
PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 9
WRITE(IDEVNO,40, IOSTAT=IOVAL, ERR=900)
FORMAT(1X,'')
CALL HDGI(IMONTH, IDAY, IYEAR, IHOUR, IMINUTE, FNAME, IPGCTR, IDEVNO)
IF(NINC.EQ.0) THEN
CALL HDG3(1, TNI, 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)
BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
DO 100 I=1,LIN
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)
100 FORMAT(I4,1X,A,7(IX,G11.4))
CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
   IPGCTR=IPGCTR+I
   START SUBSEQUENT PAGES
   WRITE(IDEVNO,50,IOSTAT:IOVAL,ERR:900)
   FORMAT('1')
   PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 8
   WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
   FORMAT(1X,'}'))
   CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
   IF(NINC.EQ.0) THEN
      CALL HDG3(I,TNI,TN,IDEVNO)
   ELSEIF(NINC.EQ.-1) THEN
      CALL HDG3(2,TNI,TN,IDEVNO)
   ELSE
      CALL HDG2(NINC,TN1,TN,IDEVNO)
   ENDIF
   CALL HDG6(IDEVNO)
   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

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

NOTES: (1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
       (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:
   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
   = 0 THEN PRINT HDG3 WITH PCALC
   = -1 THEN PRINT HDG3 WITH FINAL
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

C REAL CC(NROW1,NCOL1)
C REAL CDI(NROW2,NCOL2)
C CHARACTER CNAME*30,FNAME*24
C CHARACTER NN(NROW1)*30
C INTEGER PRTSW8,PRTSW9,IDEVN2,IDEVNO,I,J,K,H,NINC
C IF ((PRTSW8.EQ.I).OR. ((PRTSW8.EQ.0).AND. (NINC.EQ.-I))) THEN
C     INCREMENT PAGE COUNTER BY ONE
C     IPGCTR = IPGCTR + 1
C START FIRST PAGE
C PRINT FORM FEED
C WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
C 020 FORMAT('1')
C PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 9
C WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
C 040 FORMAT(1X,'')
C CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
C IF(NINC.EQ.0) THEN
C   CALL HDG3(1,TN1,TN,IDEVNO)
C ELSEIF(NINC.EQ.-1) THEN
C   CALL HDG3(2,TN1,TN,IDEVNO)
C ELSE
C   CALL HDG2(NINC,TN1,TN,IDEVNO)
C ENDIF
C CALL HDG7(IDEVNO)
C PRINT ANOTHER BLANK LINE
C WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
C BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
C DO 100 I=1,LIN
C PRINT 56 LINES OF DATA AND THEN START NEW PAGE
C WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
C +CC(I,7),CC(I,10),CC(I,13),CC(I,16),CC(I,19),CC(I,22),CC(I,25),
C +CC(I,28),CC(I,31),CC(I,34),CC(I,37),CC(I,40),CC(I,43),CC(I,46)
C 010 FORMAT(1X,I4,1X,A,14(1X,F5.3))
C CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
C IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
C   IPGCTR = IPGCTR + 1
START SUBSEQUENT PAGES

PRINT FORM FEED

WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)

FORMAT('1')

PRINT BLANK LINE FOLLOWED BY HEADINGS i, 3 & 8

WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)

FORMAT(IX,' ')

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)

PRINT ANOTHER BLANK LINE

WRITE(IDEVNO, 30,IOSTAT=IOVAL,ERR=900)

ENDIF

*************** WRITE DATA TO A FILE FOR PLOTTING ***************

IF (NINC.NE.-1) THEN

IF ((PRTSW9.EQ.2).OR. (PRTSW9.EQ.3)) THEN

DO 70 I=I,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) TNI,TN,H,

+ (CC(J,K),J=IS,IE)

055 FORMAT (T2,2(F8.2,1X),I2,1X,300(F5.3, :,IX))

K=K+3

60 CONTINUE

61 060 CONTINUE

614 070 CONTINUE

615 ENDIF

616 ENDIF

617 GO TO 999

618 900 WRITE(*,*)'IO ERROR IN PREFF: ',IOVAL

619 999 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
SUBROUTINE PREDCT(I,TN,TNI,CAVPRD,DD,NROW,NCOL,
+CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,LIN,LIN2,NN)
INTEGER NROW,NCOL,NROWI,NCOLI,NROW2,NCOL2
CHARACTER NN(NROWI)*30
REAL DD(NROW, NCOL)
REAL CC(NROWI,NCOLI)
REAL CDI(NROW2,NCOL2)
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
PRAFIL-ZERO MAT DD COL 17-21
LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
SUBROUTINES REQUIRED:
CALL PRAFIL(DD,NROW,NCOL,17,21)

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

FIND CAV PRED FOR THESE REMOVAL EFFICIENCIES
CALL MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROWI,NCOL1,
+ CAVPRD,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)

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
RETURN

*************** END OF SUBROUTINE PREDCT  ******************
SUBROUTINE RAFILL(XX,NROW,NCOL)
  INTEGER NROW,NCOL
  REAL XX(NROW,NCOL)
  
  XX=ARRAY NAME
  NCOL= COLUMNS IN MATRIX
  NROW= ROWS IN MATRIX
  
  DO 110 I=1,NROW
  DO 100 J=1,NCOL
  XX(I,J)=0.0
  110 CONTINUE
  100 CONTINUE
  
  RETURN
  
  **************** END OF SUBROUTINE RAFILL ****************
END
SUBROUTINE RCHBD (TN, TNI, CIIN, TCABIN, COEXIS, BEDQ, EMAX, CARTL, 
+ BEDOD, BEDID, DENCH, TRTYP, DCONT, VMOL, MW, VCONC, SOL, SMR, EFF, RH) 

OUTPUT:

EFF: BED REMOVAL EFF (DEC)

INPUTS:

TN, TNI = INCREMENT INITIAL AND FINAL TIMES (HR)
CIIN = BED INLET CONT CONC (MG/CU M)
TCABIN = CABIN TEMP (DEG K)
COEXIS = COEXISTANCE FACTOR
BEDQ = BED FLOW RATE (CU M/HR)
EMAX = MAXIMUM BED EFF (DEC)
CARTL = CARTRIDGE LENGTH (M)
BEDOD = BED OUTSIDE DIAMETER (M)
BEDID = BED INSIDE DIAMETER (M)
DENCH = DENSITY OF CHARCOAL IN BED (KG/CU M)
TRTYP = BED TREATMENT TYPE (1 = CI CHAR, 2 = PHOS ACID, OTHER #: NONE)
DCONT = CONT LIQUID DENSITY (GM/CC)
VMOL = CONT MOLAR VOL (GM/CC)
MW = CONT MOLECULAR WGT
VCONC = CONT VAPOR CONCENTRATION AT TCABIN (MG/CU M)
SOL = HENRY'S LAW CONSTANT FOR WATER SOLUBILITY
SMR = SUM OF CONT MASS STORED IN BED (MG) - FROM LAST INCR

REAL LPREV, LAVNI, LUTIL, LIMM, LAVAV, LADS, MW
INTEGER FACID, FCI

SET CIIN = CIIN (THIS PREVENTS CIIN FROM BEING PASSED BACK UP TO OTHER SUBROUTINES IF IT IS SET TO 1E-20)

CIN = CIIN

BED TREATMENT LOGIC

FACID = FLAG IF BED IS TREATED WITH PHOSPHORIC ACID (Y = 1 N = 0)
FCI = FLAG FOR CI CHAR IN BED (REMOVES FORMALDAHYDE)

IF (NINT (TRTYP) .EQ. 2) THEN
FACID = 1
FCI = 0
ELSEIF (NINT (TRTYP) .EQ. 1) THEN
FACID = 0
FCI = 1
ELSE
FACID = 0
FCI = 0
ENDIF

TEST FOR NO BED FLOW (BEDQ < 0) OR TN - TNI < 0; BEDL, BEDIA, DENCH = 0
IF ((BEDQ .LE. 0) .OR. (TN - TNI .LE. 0) .OR. (CARTL .LE. 0) .OR. (BEDOD .LE. 0) 
+. .OR. (DENCH .LE. 0)) THEN
EFF = 0
GOTO 199

103
ENDIF

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

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

TEST FOR MOL VOL=0 (NO CHAR REMOVAL)
IF (VMOL.EQ.0) THEN
EFF=0
GOTO 199
ENDIF

CHARCOAL REMOVAL EFFICIENCY CALCULATION
BED LENGTH (M) - ASSUMES THIN BED
BEDL=(BEDOD-BEDID)/2
BED WGT (KG)
BEDWGT=DENCH*.785*(BEDOD**2-BEDID**2)*CARTL
SUPERFICIAL BED VEL (FT/MIN)
BEDVEL=BEDQ*.0348/((BEDOD+BEDID)*CARTL)
TEST FOR CIN TOO SMALL IN AVAL CALC
IF (CIN.LT.IE-20) CIN=IE-20
AVAL=(TCABIN/VMOL)*LOG10(VCONC/CIN)
ADS ZONE LENGTH FOR 90% REMOVAL (M)
LADS=AVAL*.000275*(BEDVEL/I.3)**.8
GET QI (CC LIQ CONT/GM CHAR)
CALL FQI (AVAL, QI, FACID, SOL, RH)
LENGTH OF BED PREVIOUSLY USED BY CONT AT THIS C INLET (M)
LPREV=SMR*1.0E-6*COEXIS*BEDL/(DCONT*BEDWGT*QI)
RATE OF BED USAGE (M BED/MG CONT)
LIMM=1.0E-6*COEXIS*BEDL/(DCONT*BEDWGT*QI)
LENGTH OF BED AVAILABLE FOR ADS ZONE AT BEGINNING OF INCR (M)
LAVNI=BEDL-LPREV
IF (LAVNI.LT.0) LAVNI=0
FIX HERE IF DESORPTION IS DESIRED
IF (LAVNI/LADS.GT.20) THEN
EFFAV=EMAX
GOTO 299
ELSE
INIT INCR EFF BASED ON C IN AND BED L AVAIL AT BEG OF INCR (DEC)
EFAVNI=EMAX*(1-EXP(-2.3025851*LAVNI/LADS))
LOOP FOR EFFICIENCY
EFFAV=EFAVNI
DO 399 J=1,10,1
LENGTH OF BED UTILIZ IN INCR (M)
LUTIL=CIN*BEDQ*EFFAV*(TN-TNI)*LIMM
IF (LUTIL.GT.LAVNI) THEN
GOTO 299
ELSE
AVERAGE BED LENGTH AVAIL (M)
LAVAV=LAVNI-LUTIL/2
IF ((LAVAV/LADS).GE.20) THEN
EFFAV=EMAX
GOTO 299
ELSE
AVERAGE BED LENGTH AVAIL (M)
LAVAV=LAVNI-LUTIL/2
IF ((LAVAV/LADS).GE.20) THEN
EFFAV=EMAX
GOTO 299
ELSE
AVERAGE BED LENGTH AVAIL (M)
LAVAV=LAVNI-LUTIL/2
IF ((LAVAV/LADS).GE.20) THEN
EFFAV=EMAX
GOTO 299
ELSE
AVERAGE BED LENGTH AVAIL (M)
LAVAV=LAVNI-LUTIL/2
IF ((LAVAV/LADS).GE.20) THEN
EFFAV=EMAX
GOTO 299
ELSE
AVERAGE BED LENGTH AVAIL (M)
LAVAV=LAVNI-LUTIL/2
IF ((LAVAV/LADS).GE.20) THEN
EFFAV=EMAX
GOTO 299
118 ELSE
119 C   AV EFF BASED ON AV BED L AVAIL (DEC)
120   EFFAV=EMAX*(1-EXP(-2.3025851*LAVAV/LADS))
121 ENDIF
122 ENDIF
123 399 CONTINUE
124 299 ENDIF
125 C   MAX EFF BASED ON C IN AND RATE OF BED USAGE (DEC)
126   EFFMAX=LAVNI/(CIN*BEDQ* (TN-TNI)*LIMM)
127 IF (EFFAV.GT.EFFMAX) EFFAV=EFFMAX
128 IF (EFFAV.LT.0) EFFAV=0
129 IF (EFFAV.GT.EMAX) EFFAV=EMAX
130 C   EFF=ACTUAL EFF OUTPUT FROM SUBROUTINE
131 EFF=EFFAV
132 C   REMOVE THIS CHECK IF DESORPTION IS ADDED
133 199 IF (EFF.LT.0) EFF=0
134 IF (EFF.GT.EMAX) EFF=EMAX
135 RETURN
136 END

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

137 C   ***************************************************
138 C   * SUBROUTINE RACCH - CALCULATES REMOVAL EFF *
139 C   * BED WITH NH3 AND 1.22 MILLIMOLE H3PO4 ON CHAR *
140 C   ***************************************************
141 C   SUBROUTINE RACCH(EFF, EMAX,CARTL,BEDOD,BEDID,DENCH,SMR)
142 C OUTPUTS
143 C   EFF=OUTPUT REMOVAL EFF (DEC)
144 C INPUTS
145 C   EMAX=MAXIMUM BED REMOVAL EFF (DEC)
146 C CARTL=CARTRIDGE LENGTH (M)
147 C BEDOD=BED OUTSIDE DIAMETER (M)
148 C BEDID=BED INSIDE DIAMETER (M)
149 C DENCH=CHARCOAL DENSITY (KG/CU M)
150 C SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
151 C FOR AMMONIA CAPACITY AT SMAC
152 C CHAR USED (KG)
153 C CHRUSD=1.6E-4*SMR
154 C CHAR BED WGT (KG)
155 C BEDWGT=CARTL*(BEDOD**2-BEDID**2)*.785*DENCH
156 IF (CHRUSD.LT.0.8*BEDWGT) THEN
157   EFF=EMAX
158 ELSE
159   EFF=EMAX*SIN((BEDWGT-CHRUSD)*1.57/(BEDWGT*0.2))
160 ENDIF
161 C PREVENTS NEGATIVE EFF FOR REACTION
162 C IF (EFF.LT.0) EFF=0
163 C IF (EFF.GT.EMAX) EFF=EMAX
164 C RETURN
165 C END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0
SUBROUTINE RCICH - CALCULATES REMOVAL EFF FOR FORMALDEHYDE AND CI CHAR BED

SUBROUTINE RCICH(EFF, EMAX, CARTL, BEDOD, BEDID, DENCH, SMR, BEDQ)

OUTPUTS
EFF=OUTPUT REMOVAL EFF (DEC)

INPUTS
EMAX=MAXIMUM BED REMOVAL EFF (DEC)
CARTL=CARTRIDGE LENGTH (M)
BEDOD=BED OUTSIDE DIAMETER (M)
BEDID=BED INSIDE DIAMETER (M)
DENCH=CHARCOAL DENSITY(KG/CU M)
SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
BEDQ=BED FLOW RATE (CU M/HR)

BEDWGT=CARTL*(BEDOD**2-BEDID**2)*.785*DENCH
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
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
PREVENTS NEGATIVE EFF FOR REACTION
IF (EFF.LT.0) EFF=0
IF(EFF.GT.EMAX) EFF=EMAX
RETURN
END
SUBROUTINE REGEN(TN,TNI,DD,NROW,NCOL,CC,NROWI,NCOLI, 
+CDI,NROW2,NCOL2,LIN, LIN2,IMSGDN)
INTEGER NROW,NCOL,NROWI,NCOLI,NROW2,NCOL2
REAL DD(NROW,NCOL)
REAL CC(NROWI,NCOLI)
REAL CDI(NROW2,NCOL2)

NOTE: BEFORE RUNNING THIS SUBROUTINE THE ORIGINAL FLOW RATES FROM 19 C
TIME INCREMENT THEY MUST BE RESTORED TO COL 2

DIRECT INPUTS:
TN=INCREMENT END TIME (HRS); TNI=INCR BEGINNING TIME HRS
DD,NROW,NCOL=NAME AND SIZE OF MAT DD
CC,NROWI,NCOLI=NAME AND SIZE OF MAT CC
CDI,NROW2,NCOL2=NAME AND SIZE OF MAT CDI
LIN=NO. OF CONT IN MAT CDI
LIN2=NO. OF DEVICES IN MAT DD

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

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

SUBROUTINES REQUIRED:
RECHG

START LOOP FOR ALL DEVICES 3 TO 15
DO 100 J=3,LIN2
   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
   DEVICE = CHARCOAL
   IF(DD(J,3).EQ.3.OR. DD(J,3).EQ.4) THEN
      TRCI=DD(J,15)
      REGENERATION/CHANGEOUT INTERVAL (HRS)
   END IF
REGENERATION DURATION (HRS)
TRD=DD(J,16)
INITIAL (FIRST) REGENERATION
TIR=DD(J,14)
ENDIF

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

CHECK AND FIX INPUT AS REQ + PRINT WARNINGS
TIME INCREMENT (HRS)
TINC=DD(1,11)
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 ',TNI,
  + 'DEV NO.', (J)
  WRITE(IMSGDN,*) 'INITIAL TIME NOT = MULTIPLE OF TIME
  + INCREMENT'
  WRITE(IMSGDN,*) 'TRUNCATED TO ',TIR
  CLOSE(IMSGDN)
ENDIF

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 ',TNI,
  + 'DEV NO.', (J)
  WRITE(IMSGDN,*) 'REGEN/CHG TIME NOT = MULTIPLE OF TIME
  + INCREMENT'
  WRITE(IMSGDN,*) 'TRUNCATED TO ',TRCI
  CLOSE(IMSGDN)
ENDIF

IF REGEN/CHGOUT INTERVAL <= 0 THEN GOTO END OF LOOP FOR CONT
IF (TRCI.LE.0) THEN
  GOTO i00
ENDIF

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
REGENERATION DURATION > REGEN/CHG INTERVAL

IF (TRD.GT.TRCI) THEN
TRD=TRCI
OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
WRITE(IMSGDN,*)'INCREMENT BEGINNING TIME ',TNI,
WRITE(IMSGDN,*)'REGEN DURATION > REGEN/CHG INTERVAL'
WRITE(IMSGDN,*)'TRUNCATED TO ',TRD
CLOSE(IMSGDN)
ENDIF

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

CALL RBGCHG(TN1,TRCI,TRD,TIR,TINC,IRBFLG,IRTFLG)
REGENERATION OCCURS AT BEGINNING OF INCREMENT
IF (IRBFLG.EQ.I) THEN
PUT SUM MASS REM =0 IN MAT CC FOR THIS DEVICE
START LOOP FOR ALL CONT FOR THIS DEVICE
K=J*3+3
DO 101 I=1,LIN
CC(I,K)=0
101 CONTINUE
IF DEVICE = LIOH BED PUT SUM MASS:0 IN DD(J,16)
IF (DD(J,3).EQ.5) THEN
DD(J,16)=0
ENDIF
ELSE
SET DEVICE Q= ORIGINAL VALUE
DD(J,2)=DD(J,7)
ENDIF
REGENERATION OCCURS THROUGHOUT ENTIRE INCREMENT
IF (IRTFLG.EQ.1) THEN
SET DEVICE Q=0
DD(J,2)=0
ELSE
SET DEVICE Q= ORIGI NAL VALUE
DD(J,2)=DD(J,7)
ENDIF
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

******************************************************************************
* AUXILIARY REGENERATION SUBROUTINE-REGCHG                                  *
* DETERMINES IF REGEN/CHANGEOUT IS TO OCCUR AT BEGINNING OF                 *
* TIME INCREMENT-ALSO DETERMINES IF REGENERATION IS OCCURRING              *
* THROUGHOUT THE TIME INCREMENT                                            *
******************************************************************************

SUBROUTINE REGCHG(TN1,TRCI,TRD,TIR,TINC,IRBFLG,IRTFLG)

INPUTS:
TN1=INCREMENT INITIAL TIME (HRS)
TRCI=CHANGEOUT/REGENERATION INTERVAL (HRS)
TRD=REGENERATION DURATION (HRS)
TIR=INITIAL (FIRST) REGENERATION TIME (HRS)
TINC=TIME INCREMENT (HRS)

OUTPUTS:
REGENERATION OCCURS AT BEGINNING OF TIME INCREMENT (Y OR N)
(IRBFLG=I FOR Y & 0 FOR N)
REGENERATION IS OCCURRING THROUGHOUT THE WHOLE INCREMENT (Y OR N)
(IRTFLG=1 FOR Y & 0 FOR N)

REGENERATION OCCURS AT BEGINNING OF TIME INCREMENT

IF (TNI.EQ.0) GOTO 10
IF(TNI.LT.TIR) GOTO 10
IF(TRCI.LE.0) GOTO 10
IF(AINT((TNI-TIR)/TRCI).EQ.((TNI-TIR)/TRCI)) THEN
  REGENERATION OCCURS
  IRBFLG=1
  GO TO 20
ENDIF

REGENERATION OCCURRING THROUGHOUT ENTIRE TIME INCREMENT

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

REGEN DOESN'T OCCUR

RETURN
* SUBROUTINE RINCDD *
* SUBROUTINE TO OPERATE ON INCREMENT DEPENDENT DATA *
* READS DATA FROM MAT TT AND PUT IT IN THE PROPER PLACES IN *
* MAT CDI OR MAT DD - USED AT THE BEGINNING OF EACH TIME INCR *
*****************************************************************
*
SUBROUTINE RINCDD(I,TN,TNI,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)
SUBROUTINES REQUIRED:
NONE
DD,NROW,NCOL,LIN2=NAME,DIM & NO DEV IN MAT DD
CC,NROW1,NCOL1=NAME & DIM OF MAT CC
CDI,NROW2,NCOL,LIN=NAME,DIM & NO CONT IN MAT CDI
TT,NTTROW,NTTCOL,LIN1=NAME,DIM & NO ITEMS IN MAT TT
IF (LIN1.EQ.0) GOTO 999
BEGIN LOOP FOR ALL LINES IN MAT TT
DO 100 K=I,LINI
CHECK FOR TIME >= TNI AND < TN
IF((TT(K,1).LT.TNI).OR.(TT(K,1).GE.TN)) THEN
GOTO 100
ENDIF
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)
ICONTN=INTEGER CONTAMINANT NO. - TT(K,2)
GENRT=CONT GENERATION RATE (MG/HR) - TT(K,3)
IDEVNO=INTEGER DEVICE NUMBER - TT(K,4)
DEVQ=DEVICE FLOW RATE (CU M/HR) - TT(K,5)
ICOLNO=INTEGER COLUMN NUMBER IN MAT DD - TT(K,6)
VAL=NEW VALUE IN MAT DD - TT(K,7)
CASE NO. 1 - CHANGE CONTAMINANT GENERATION RATE
IF((ICONTN.GT.0).AND.(ICONTN.LE.LIN)) THEN
IF(IDEVNO.EQ.I) THEN
CDI(ICONTN, I)=GENRT
ELSEIF ((IDEVNO.GE.3).AND.(IDEVNO.LE.LIN2)) THEN
CDI(ICONTN, (7+IDEVNO))=GENRT
ENDIF
GOTO 100
ENDIF
CASE 2 - CHANGE DEVICE FLOW OR OTHER DD DATA
GOTO 100
ENDIF
ENDIF
CASE 2 - CHANGE DEVICE FLOW OR OTHER DD DATA
THIS CASE WORKS ONLY IF ANY CONT NO. <=0
MUST USE -1 FOR ANY Q OR NEW VALUE NOT TO BE CHANGED
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
      CHANGE DEVICE FLOW IN MAT DD
      DD(IDEVNO,2)=DEVQ
    ENDIF
  ENDIF
  IF ((ICOLNO.GE.1).AND.(ICOLNO.LE.16)) THEN
    IF (VAL.GE.0) THEN
      CHANGE VALUE IN MAT DD
      DD(IDEVNO,ICOLNO)=VAL
    ENDIF
  ENDIF
ENDIF
100 CONTINUE
999 RETURN
END
SUBROUTINE RRIN
   SUBROUTINE TO READ REAL DATA INTO MAT XX(ROW,COL)
   RETURNS NUMBER OF LINES OF DATA READ FROM FILE
   READS FROM COL 1 TO COL LSTCOL

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
   OPEN(I,FILE=FNAME,STATUS='OLD',IOSTAT=IOVAL)
   IF(IOVAL.NE.0) GOTO 900
   LIN=0
   DO 100 I=1,NROW
      READ(I,*), (XX(I,J),J=1,LSTCOL)
      LIN=LIN+1
   100 CONTINUE
   WRITE(*, '(A)') ' DONE WITH FILE INPUT'
   WRITE(*, '(*,*)')
   CLOSE(I)
   GOTO 900
500 WRITE(*, '*') ' IOERROR= ',IOVAL
   CLOSE(I)
   WRITE(*, '*') ' WHAT IS THE INPUT FILE NAME? '
   GOTO 10
900 RETURN

END
SUBROUTINE RROUT(XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,IMSGDN,FNAME,
+ IDEVNO,IOVAL)
INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,IDEVNO
REAL XX(NROW,NCOL)
IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
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(I,FILE=FNAME, IOSTAT=IOVAL)
IF(IOVAL.NE.0) GOTO 900
DO ii0 I=I,LIN
WRITE(I,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
WRITE(IDEVNO,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
FORMAT(IX,7GII.4)
WRITE(I,*,IOSTAT=IOVAL,ERR=900)
WRITE(IDEVNO,*,IOSTAT=IOVAL,ERR=900)
CONTINUE
CLOSE (i)
OPEN(IMSGDN,FILE=°CON',IOSTAT=IOVAL)
WRITE(IMSGDN,*) °IOERROR= ',IOVAL
CLOSE(IMSGDN)
CLOSE (i)
CLOSE (IDEVNO)
GOTO 10
RETURN
*************** END OF SUBROUTINE RROUT ************************
END
**SUBROUTINE RROUT2** (XX, NROW, NCOL, FSTCOL, LSTCOL, LIN, IMSGDN)

SUBROUTINE RROUT (XX, NROW, NCOL, IVAL, FSTCOL, LSTCOL, LIN, IDEVNO)

REAL XX(NROW, NCOL)

IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL

OPEN(1, FILE=FNAME, IOSTAT=IOVAL)
IF(FOAMNE.EQ.'LPT1'.AND.(FNAME.EQ.'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)
70 FORMAT(1X,7G11.4)
WRITE(1,*,IOSTAT=IOVAL, ERR=900)
110 CONTINUE
CLOSE (1)
GOTO 990

OPEN(1, FILE=FNAME, IOSTAT=IOVAL)
WRITE(1,*,IOERROR= 'I  OVALID', IOVAL)
CLOSE (1)
GOTO 10

RETURN

END
**SUBROUTINE SLIOH**

* SUM LIOH USED IN TIME INCREMENT FOR EACH BED ONE AT A TIME *

* AND FOR ALL CONTAMINANTS FOR EACH BED *

**SUBROUTINE SLIOH(TN,TNI,DD,NROW,NCOL,CC,NROW1,NCOL1, CDI,NROW2,NCOL2,LIN,LIN2)**

**INTEGER** NROW,NCOL,NROWI,NCOLI,NROW2,NCOL2

**REAL** DD(NROW,NCOL)

**REAL** CC(NROWI,NCOLI)

**REAL** CDI(NROW2,NCOL2)

**SUBROUTINES REQUIRED: NONE**

**DIRECT INPUTS:**

TN=INCREMENT END TIME (HRS); TNI=INCR BEGINNING TIME HRS

DD,NROW,NCOL=NAME AND SIZE OF MAT DD

CC,NROW1,NCOL1=NAME AND SIZE OF MAT CC

CDI,NROW2,NCOL2=NAME AND SIZE OF MAT CDI

LIN=NO. OF CONT IN MAT CDI

LIN2=NO. OF DEVICES IN MAT DD

**OTHER INPUTS FROM MAT DD**

DD(J,3)=DEVICE NUMBER

DD(J,16)=AMT OF LIOH PREVIOUSLY USED BY DEVICE

CDI(I,17)=LB LIOH UTIL/LB CONT ADSORBED IN BED (FOR ONE CONT)

**OUTPUTS (STORED IN MAT DD):**

DD(J,16)=AMOUNT OF LIOH UTILIZED BY DEVICE THROUGH THE END OF

THIS TIME INCREMENT

DD(J,15)=RATE OF LIOH USAGE FOR DEVICE

**START LOOP FOR ALL DEVICES 3 TO 15**

DO 100 J=3,LIN2

**CHECK FOR DEVICE = LIOH BED**

IF (DD(J,3).EQ.5) THEN

**RATE OF LIOH UTILIZATION (KG/HR)**

RWUTLI=0

**BEGIN LOOP FOR ALL CONTAMINANTS**

DO 110 I=1,LIN

RWUTLI=RWUTLI+CC(I,K)*CDI(I,7)*1E-6

110 CONTINUE

**STORE RATE OF LIOH UTILIZATION IN MAT DD FOR THIS DEVICE**

DD(J,15)=RWUTLI

**UPDATE AMOUNT OF LIOH UTIL THROUGH THE END OF TIME INCR (KG)**

DD(J,16)=DD(J,16)+RWUTLI*(TN-TNI)

ENDIF

**END LOOP FOR ALL DEVICES**

RETURN

**END**
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 16 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:

\[ T_{\text{group}} = \sum \frac{C_c}{C_m} , \quad (B1) \]

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


5. Ibid., p. 24


8. Ibid., p. 31.


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