

105
55
D. 819
Dp. 2

Uptake and Transport of Chemicals by Plants (Version 2.1)



Special Report 819
May 1988



Agricultural Experiment Station
Oregon State University, Corvallis

UPTAKE AND TRANSPORT OF CHEMICALS BY PLANTS (VERSION 2.1)

F. T. LINDSTROM, L. BOERSMA, C. McFARLANE,
K. P. SUEN, D. CAWLFIELD

AUTHORS: F. T. Lindstrom, associate professor and L. Boersma, professor, Department of Soil Science, Oregon State University; C. McFarlane, Research Scientist, Environmental Research Laboratory, United States Environmental Protection Agency; K. P. Suen and D. Cawlfild, Computer Programmers, Department of Soil Science, Oregon State University.

FOREWORD

The uptake, transport, and accumulation of organic and inorganic chemicals by plants are influenced by characteristics of the plant, properties of the chemical, properties of the soil, and by prevailing environmental conditions. Complex interrelationships exist between the physical, chemical, and physiological processes that occur in specific plant tissues and the response of these processes to environmental conditions, such as the daily cycle of radiation, evaporation, and air temperature. The uptake is further influenced by the availability of the chemical at the root surface determined in turn by transport characteristics of the soil.

Progress with understanding the interrelationships between rate of uptake and soil, plant, and atmospheric conditions can be made by using mathematical models. The development of a predictive simulator for the uptake of xenobiotic chemicals by plants from the soil solution was reported in Special Report 818, "Model of Coupled Transport of Water and Solutes in Plants", Oregon Agricultural Experiment Station.

This report provides a user guide and describes the structure of the computer code developed to make simulations with the model.

ACKNOWLEDGEMENTS

This publication reports results of studies supported, in part, by Research Grant CR 811940-01-0 "A Mathematical Model of the Bioaccumulation of Xenobiotic Organic Chemicals in Plants" from the Corvallis Environmental Research Laboratory of the United States Environmental Protection Agency to the Department of Soil Science at Oregon State University and by the Oregon Agricultural Experiment Station.

CONTENTS

	PAGE
Foreword	i
Acknowledgement	i
List of Figures and Tables	iii
User Guide	1
Introduction	1
Setting the computer to run the program	2
Hardware requirements	2
Computer system configuration requirements.....	2
Operation	2
Structure of the input parameter file	2
How to run the program (step by step)	5
Structure of the output files	7
Example	9
Appendix	27
a. Program code structure	27
b. Program flow diagrams	28
c. Program source codes	37

FIGURES

<u>Figure</u>		<u>Page</u>
1	Schematic diagram of generic "Three-Leaf" plant, showing the hierarchy of leaves on the stem.	3
2	Numbering sequence for compartment used for the "Three-Leaf Model" of plant uptake and transport.	4
3	Initial- and Second-Screen Headers	10
4	Third- and Fourth-Screen Headers	11
5	Fifth- and Sixth-Screen Headers	12
6	Main program flow diagram	29
7	Subroutine (a) data driver and (b) load data flow diagrams	30
8	Subroutine (a) matrix a and (b) initial value flow diagrams	31
9	Subroutine stiff flow diagram	32
10	Subroutine intgrl flow diagram	34
11	Subroutine (a) outmass and (b) outmlos flow diagrams	35
12	Subroutine (a) outgroup and (b) outrslwl flow diagrams	36

TABLES

<u>Table</u>		<u>Page</u>
1	A typical datafile called SAMPLE.DAT	14
2	Definitions of variables	15
3	Listing of the file SAMPLE.OUT	16

INTRODUCTION

UTAB (Uptake, Translocation, Accumulation, and Biodegradation), a three Leaf Model [version 2.1], is a simulation program written in MS-Fortran77 v3.31 for uptake and transport of chemicals by plants. This program is designed to simulate the uptake of chemicals from soils by conceptualized plants (Boersma et al 1988)*. Upon completion of validation tests by the Corvallis Environmental Research Laboratory (USEPA CERL), the model will be used to predict the rate of uptake, fate, and distribution of a broad spectrum of chemicals, which may exist in the root zone. These may be naturally occurring chemicals or those applied to the soil environment such as waste disposal, storage, or agriculturally applied chemicals for weed or pest control. The model assumes a constant transpiration rate over time, as well as constant compartment size and fluid flow dynamics.

The diskette of "UPTAKE AND TRANSPORT OF CHEMICAL BY PLANTS" Three-Leaf Model [version 2.1] contains the following files:

	<u>File</u>	<u>Explanation</u>
1.	COMBLOC	include file
2.	PLANT3V2.FOR	main program
3.	DTADR3V2.FOR	sub program
4.	MTXA3V2.FOR	" "
5.	STIFF3V2.FOR	" "
6.	DECINT.FOR	" "
7.	PRINT3V2.FOR	" "
8.	PLANT3V2.EXE	executable program
9.	CPRINT.EXE	" "
10.	DECINT.LIB	integration library routine
11.	SAMPLE.DAT	input data
12.	SAMPLE.OUT	output listing

* Boersma L., Lindstrom F. T., McFarlane C., and McCoy E. L. 1988. Model of Coupled Transport of Water and Solutes in Plants. Special Report____. Oregon Agricultural Experiment Station, Corvallis, Oregon.

SETTING THE COMPUTER TO RUN THE PROGRAM

Hardware Requirements

The program described in this report has been developed on an IBM personal computer. It has been successfully tested on the IBM PC, XT, AT series as well as on several IBM-compatible machines.

Monitor : Monochrome (black/white) or color.
Memory requirements : 640k.
Coprocessor : 8087 which gives fast execution speed.
Hard disk : Optional, but highly recommended.
Regular disk drive : At least one 5.25 inches floppy disk drive.

Computer System Configuration requirements

The PLANT3V2.EXE program require a system file (config.sys) to enhance program and user interface. The config.sys file must contain the following COMMANDS:

```
files=20
buffers=20
device=ansi.sys
```

Refer to the IBM DOS reference manual for details.

OPERATION

Structure of the input parameter file

A sample input data file named "sample.dat" (see Table 1) is provided on the diskette. It is important to understand the structure of this data file for future parameter modifications.

Figure 1 shows a generic plant consisting of 1) a root, 2) three stems, and 3) three leaves. These three components are subdivided so that the plant is represented in Figure 2 as consisting of 29 compartments and 32 interconnecting pathways. The interconnecting pathways may be membranes or flow tubes. The numbers inside the square boxes are compartment numbers and the numbers between the boxes are interconnecting pathway numbers. As a PLANT3V2.EXE user, it is important to know where and how the plant was numbered as presented in Figure 2 because the program uses this numbering scheme to process each

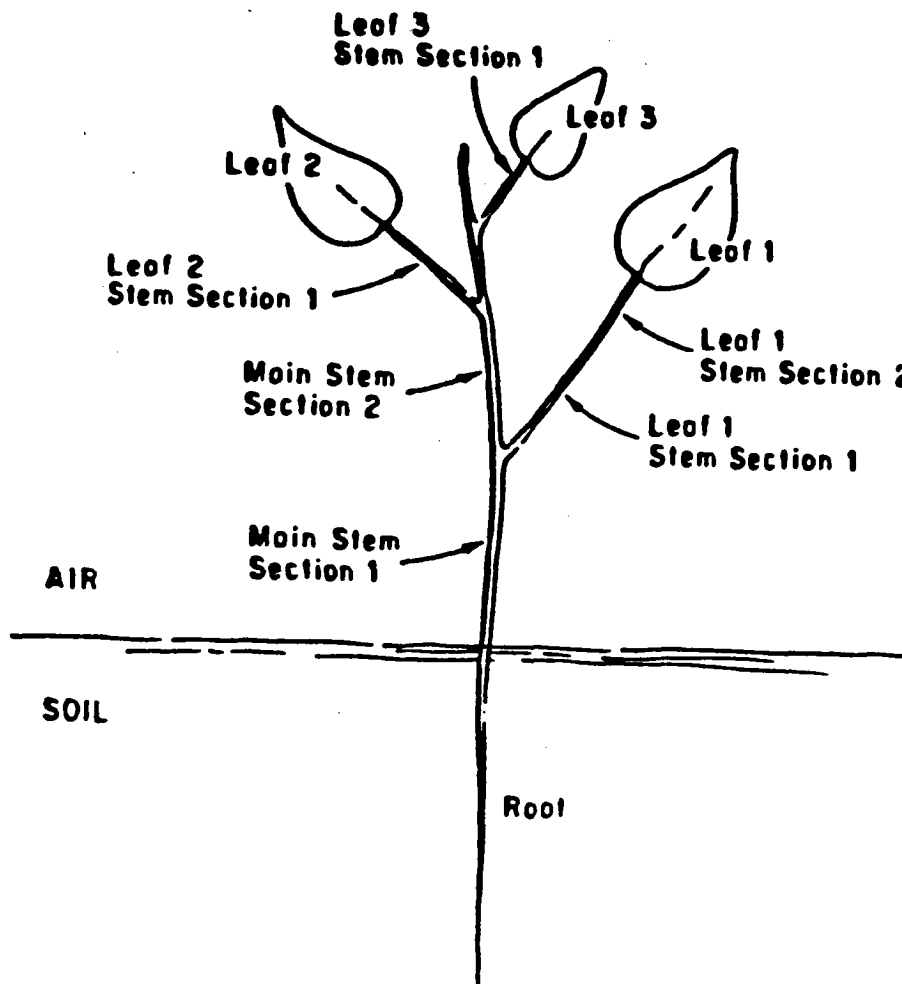


Figure 1. Schematic diagram of generic "Three-Leaf" plant, showing the hierarchy of leaves on the stem.

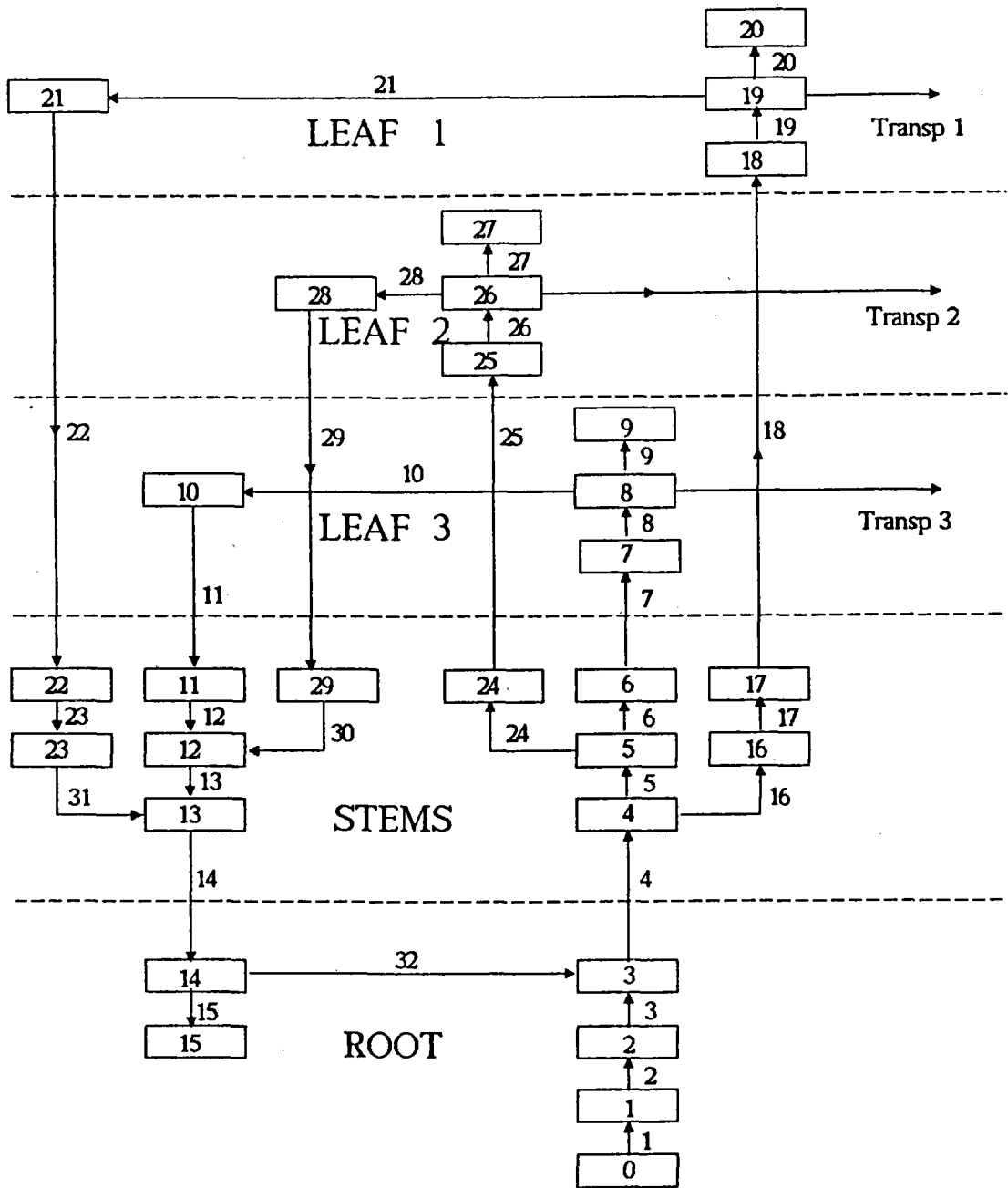


Figure 2. Numbering sequence for compartment used for the "Three-Leaf Model" of plant uptake and transport.

compartment in a predefined matrix within the program. The data file is thus defined according the numbering sequence.

The program always skips the first input line and one line between each section of parameters. The parameters must be organized sectionally in the following sequence (see Table 1):

	<u>Parameter</u>	<u>Number of coefficient</u>
1.	area(I)	32
2.	delta x(I)	32
3.	sigma(I)	32
4.	diffu(I)	32
5.	Kr(I)	32
6.	Kl(I)	32
7.	conc0	1
8.	Vol(I)	29
9.	B(I)	29
10.	Lambda(I)	29
11.	Js	1
12.	Frac(I)	3
13.	Transp(I)	3
14.	initial mass(I)	29

The parameters are defined as Fortran variables in Table 2.

WARNING! Be sure that the proper number of coefficient is provided for each section of parameters, otherwise the program will not execute.

How to run the program (step by step)

The program require two input informations. The informations are 1) the input data file described from the previous section and 2) the user interface used to control the simulation process. The following example illustrates a step-by-step user interface when executing PLANT3V2. The underlined characters are user inputs from the keyboard.

To start up the program, user simply types the name of the program and presses return.

```
c:\> PLANT3V2 <return>
```

After pressing the return key, the monitor screen is cleared and a printed screen header is displayed (See Figure 3). The following three requests will be made.

1. Enter today's date (e.g. 04-01-86) --> 07-17-87 <return>

2. Enter input filename --> SAMPLE.DAT <return>
3. Enter file name to store output or
"prn:" output to line printer --> SAMPLE.OUT <return>

At this point, you can either choose a file name for storing your output on a disk or send the output directly to the line printer. The above example shows that output file name "SAMPLE.OUT" has been chosen. If the output file name already exists in the disk, the program will prompt the user with a warning message, then allow the user to replace the existing file or to create a different file name. After the program knows where to direct the outputs, the following five requests will be made to initiate the simulation process.

4. Enter time (in hours) to start simulation --> 0 <return>
5. Enter time (in hours) to terminate simulation --> 96 < " >
6. Enter time interval (in hours) to integrate --> 0.1 < " >
7. Enter tolerance (default is .100000E-03) --> 1.0d-6 < " >
8. Enter time interval for output listing:
(e.g. 1.0 for listing every 1 hour.
1.5 for listing every 1.5 hours.
3.0 for listing every 3 hours.) --> 3 <return>

At this point, the program sets up the system mass balance equations matrix elements. Figure 3(b) shows the screen header and messages displayed while waiting. Figure (4a) shows a new screen display with the following question such that it allows the user to proceed simulation or quit at this point to examine the input data.

9. Do you want the simulation to proceed (y/n) --> y <return>

At this point, the program displays a time clock (see Figure 4b) to indicate simulation time. When the simulation has been completed, it will display a screen menu showing the options for data output (see Figure 5a):

- [1] Individual mass in compartments.
- [2] Individual mass lost in compartments.
- [3] Grouped mass; Root, Stems, and Leaves.
- [4] Grouped mass; Root, Stems, and Leaves with Lambda.
- [5] List all of the above.
- [6] Exit to DOS.

The final step in completing the simulation run is to select the desired output listing from the menu. Note: Any output combination may be selected. Option 6 must be chosen to leave this menu. Upon leaving

this menu, you are also leaving the program. The program provides instructions on where to retrieve the output listing selected from the menu (see Figure 5b).

The time it takes to complete a simulation varies from one machine to another. Moreover, it varies from one input data set to another set just discussed. It takes about 15 minutes to complete the sample data on an IBM AT with the 82807 math coprocessor chip operational, and 45 minutes on an IBM PC with an 8087 chip in place.

Structure of the output file

The output files are automatically sent to a default drive. Five text files are generated by the program, and named as follows:

1. DATA.OUT (default)
2. SAMPLE.OUT (user defined)
3. MASSFLW.OUT (default)
4. GRPMAS.OUT (default)
5. GRPRSLWL.OUT (default)

DATA.OUT is a file that contains the entries of the system transport-transfer matrix generated by subroutine mtax (see Figure 8a).

SAMPLE.OUT is a file that contains the selected results of the simulation solutions from the output option menu. Note: Use the CPRINT.EXE utility program, which is also provided on the diskette, to list SAMPLE.OUT. This write file shows (see Table 3):

- 1) the system parameters
- 2) individual mass of chemical in each compartment
- 3) individual mass loss via all first-order loss processes in each compartment
- 4) grouped chemical masses in
 - i) roots, ii) stems, and iii) leaves
- 5) grouped chemical masses with losses
 - i) roots, ii) stems, and iii) leaves

All this information is obtained because output option 5 was selected in this example.

MASSFLW.OUT is a file that contains the results of the difference in mass between inflow to and outflow from each individual compartment. The first data column is time (in hours) and data columns 2 through 30 represent compartments 1 through 29, respectively. Neither page headers nor column numbers are provided in this file, making it easy to load into other software, such as, LOTUS, SAS, etc., for statistical analysis or plotting the results (see Figure 11a). GRPMAS.OUT is a file that contains the calculated grouped masses in the plant sections without

mass loss due to first order loss (LAMBDA) processes. The plant sections are the root, stems, and leaves. This is the same output as option 3 on the screen output menu except that this file has no page header (see Figure 12a).

GRPRSLWL.OUT is a file that contains the calculated grouped masses with first-order loss (LAMBDA) added. These were obtained by adding cumulative first-order losses (LAMBDA) to the plant sections. This is the same output as option 4 on the screen output menu except that this file has no page header (see Figure 12b).

EXAMPLE

This is an actual program process recorded from beginning to end of a simulation run.

Figures 3, 4, and 5 are the questions and messages prompted by the program during the interactive process.

TABLE 1 is a sample input data file (SAMPLE.DAT) which is used to illustrate the organization of the parameters and TABLE 2 is a list of definitions and units used on the input parameters.

TABLE 3 is a list of the results generated by the program where option 5 was selected from the output menu (see Figure 5a).

PLANT TRANSPORT THREE-LEAF MODEL (ver 2.1)
ALL RIGHTS RESERVED.

Enter today's date (e.g. 04-01-86) --> 07-17-87
Enter input filename --> SAMPLE.DAT
Enter filename to store output or
"prn:" output to line printer --> SAMPLE.OUT

Enter time (in hours) to start simulation --> 0
Enter time (in hours) to terminate simulation --> 96
Enter time interval (in hours) (step size) to integrate--> .01
Enter tolerance (default is .100000E-03) --> 1.0d-6
Enter time interval for output listing:
(e.g. 1.0 for listing every 1 hour.
1.5 for listing every 1" hours.
3.0 for listing every 3 hours.) --> 3

(a)

PLANT TRANSPORT THREE-LEAF MODEL (ver 2.1)
ALL RIGHTS RESERVED.

The input data are now being printed to the user-designated output device.

Please check inputs.

The program is now constructing the mass balance equations.

Please wait a moment !

(b)

Figure 3. Initial- and Second-Screen Headers.

PLANT TRANSPORT THREE-LEAF MODEL (ver 2.1)
ALL RIGHTS RESERVED.

The mass balance equations are being constructed.

Do you want simulation to proceed (Y/N) --> y

(a)

PLANT TRANSPORT THREE-LEAF MODEL (ver 2.1)
ALL RIGHTS RESERVED.

----- Simulation in process. -----

Targeted simulation time is 96 hours.

If the program issued a warning message,
do not panic and let the execution proceed.
Refer to user guide for explanation of the
warning message(s).

The simulated time is 2 hour(s).

(b)

Figure 4. Third- and Fourth-Screen Headers.

PLANT TRANSPORT THREE-LEAF MODEL (ver 2.1)
ALL RIGHTS RESERVED.

!!!! Simulation has been completed !!!!

Choices of output listing:

- [1] Individual mass in compartments.
- [2] Individual mass loss in compartments.
- [3] Grouped mass; Root, Stems and Leaves.
- [4] Grouped mass; Root, Stems and Leaves with lambda.
- [5] List all of the above.
- [6] Exit to DOS.

Please select your favorite output number --> 5

Are you sure (y/n) ? --> y

(a)

PLANT TRANSPORT THREE-LEAF MODEL (ver 2.1)
ALL RIGHTS RESERVED.

You are now leaving PROGRAM PLANT3V2 [version 2.1]

You can retrieve your output from FILENAME --> sample.out

or line printer if you chose prn: as output device.

Stop - Program terminated.
c:\plant3v2>

(b)

Figure 5. Fifth- and Sixth-Screen Headers.

Table 1. A typical datafile called SAMPLE.DAT

1	2	3	4	5	6	7	8	9	
6.3500E+00	4.7200E+01	1.6700E+00	1.0800E-03	1.0800E-03	1.0800E-03	1.0800E-03	0.1000E+00	2.4000E+01	
8.3600E-02	1.0800E-04	1.0800E-04	1.0800E-04	1.0800E-01	3.0000E-03	1.0800E-03	1.0800E-03	1.0800E-03	
0.1000E+00	2.4000E+01	8.3600E-02	1.0800E-04	1.0800E-04	1.0800E-03	1.0800E-03	0.1000E+00	2.4000E+01	
8.3600E-02	1.0800E-04	1.0800E-04	1.0800E-04	8.2580E-01					= AREA
0.2000E+00	0.5000E-01	0.3000E-01	5.0000E-01	5.0000E+00	5.0000E+00	5.0000E-01	1.0000E-03	1.0000E-03	
1.0000E-03	5.0000E-01	5.0000E+00	5.0000E+00	5.0000E-01	1.0000E-03	5.0000E+00	5.0000E+00	5.0000E-01	
1.0000E-03	1.0000E-03	1.0000E-03	5.0000E-01	5.0000E+00	5.0000E+00	5.0000E-01	1.0000E-03	1.0000E-03	
1.0000E-03	5.0000E-01	5.0000E+00	5.0000E+00	1.0000E-03					= DELTA X
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	= SIGMA
1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	
1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	
1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	
1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03	1.8000E-03					= DIFFU
1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	
1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	
1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	
1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01					= KR
1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	
1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	
1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	1.0000E-01	= KL
1.0000E+00									= CONCO
2.1900E-03	0.4600E-01	1.1800E-03	5.4000E-04	5.4000E-03	5.4000E-03	1.4900E-03	5.0500E-03	3.5600E-02	
1.0600E-04	5.4000E-05	5.4000E-04	5.4000E-04	9.2800E-06	1.0000E-03	5.4000E-03	5.4000E-03	1.4900E-03	
5.0500E-03	0.3560E-01	1.0600E-04	5.4000E-05	5.4000E-04	5.4000E-03	1.4900E-03	5.0500E-03	3.5600E-02	
1.0600E-04	5.4000E-05								= VOL
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	= B
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	= LAMBDA
0.0000E+00									= Js
1.0000E+00	1.0000E+00	1.0000E+00							= FRAC
0.1000E+00	0.1000E+00	0.1000E+00							= TRANSP
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	
0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	= INITIAL

Table 2. Definitions of Variables

Variable	Math. Symbol	Meaning and Units
AREA(I)	A_i	Inter compartmental membrane area ₂ or cross sectional area of flow (cm ²)
DELTA(I)	δx_i	Inter compartmental membrane thickness or length of flow tube (cm)
SIGMA(I)	σ_i	Reflection coefficient of membrane (dimensionless)
DIFFU(I)	D_i	Diffusion coefficient of membrane or or flow path (cm ² /hr)
KR(I)	K_{ri}	Chemical partition coefficient of of membrane matter-i + 1st compartment (dimensionless)
KL(I)	K_{li}	Chemical partition coefficient of membrane matter-ith compartment (dimensionless)
CONCO	C_0	Concentration in soil water of chemical ($\mu\text{g}/\text{cm}^3$)
VOL(I)	V_i	Volume of ith compartment (cm ³)
B(I)	B_i	Linear equilibrium Sorption parameter in ith compartment (dimensionless)
LAMBDA(I)	λ_i	Coefficient of first order loss process law in ith compartment (1/hr)
JS	J_s	Zeroth order chemical flux across the soil-root boundary ($\mu\text{g}/\text{cm}^2\text{-hr}$)
FRAC(I)	f_i	Fraction, $0 \leq f_i < \infty$, of ith transpiration pathway (dimensionless)
TRANSP(I)	Tr_i	Transpiration rate of ith leaf (cm/hr)
INITIAL(I)	$M_i(0)$	Initial mass of chemical in the ith compartment at time $t = 0.0$ hours.

Table 3. Listing of the file SAMPLE.OUT

Start time: .000, Stop time: 96.000, Tolerance: .0001000, Step size: .10000, Print interval: 3.00000

PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGHTS RESERVED.

This FILENAME is --> sample.out

Date : 12-11-87

1	2	3	4	5	6	7	8	9	
.6350E+01	.4720E+02	.1670E+01	.1080E-02	.1080E-02	.1080E-02	.1080E-02	.1000E+00	.2400E+02	
10	11	12	13	14	15	16	17	18	
.8360E-01	.1080E-03	.1080E-03	.1080E-03	.1080E+00	.3000E-02	.1080E-02	.1080E-02	.1080E-02	
19	20	21	22	23	24	25	26	27	
.1000E+00	.2400E+02	.8360E-01	.1080E-03	.1080E-03	.1080E-02	.1080E-02	.1000E+00	.2400E+02	
28	29	30	31	32					
.8360E-01	.1080E-03	.1080E-03	.1080E-03	.8258E+00					= Area
1	2	3	4	5	6	7	8	9	
.2000E+00	.5000E-01	.3000E-01	.5000E+00	.5000E+01	.5000E+01	.5000E+00	.1000E-02	.1000E-02	
10	11	12	13	14	15	16	17	18	
.1000E-02	.5000E+00	.5000E+01	.5000E+01	.5000E+00	.1000E-02	.5000E+01	.5000E+01	.5000E+00	
19	20	21	22	23	24	25	26	27	
.1000E-02	.1000E-02	.1000E-02	.5000E+00	.5000E+01	.5000E+01	.5000E+00	.1000E-02	.1000E-02	
28	29	30	31	32					
.1000E-02	.5000E+00	.5000E+01	.5000E+01	.1000E-02					= Delta x
1	2	3	4	5	6	7	8	9	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
10	11	12	13	14	15	16	17	18	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
19	20	21	22	23	24	25	26	27	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
28	29	30	31	32					
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00					= Sigma
1	2	3	4	5	6	7	8	9	
.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	
10	11	12	13	14	15	16	17	18	
.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	
19	20	21	22	23	24	25	26	27	
.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02	
28	29	30	31	32					
.1800E-02	.1800E-02	.1800E-02	.1800E-02	.1800E-02					= Diffu
1	2	3	4	5	6	7	8	9	
.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	
10	11	12	13	14	15	16	17	18	
.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	
19	20	21	22	23	24	25	26	27	
.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	
28	29	30	31	32					
.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00					= Kr
1	2	3	4	5	6	7	8	9	
.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	
10	11	12	13	14	15	16	17	18	
.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	
19	20	21	22	23	24	25	26	27	
.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	
28	29	30	31	32					
.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00					= K1

Table 3. Listing of the file SAMPLE.OUT (continued)

PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGHTS RESERVED.

This file is named --> sample.out

DATE : 12-11-87

.1000E+01									= Conc0
1	2	3	4	5	6	7	8	9	
.2190E-02	.4600E-01	.1180E-02	.5400E-03	.5400E-02	.5400E-02	.1490E-02	.5050E-02	.3560E-01	
10	11	12	13	14	15	16	17	18	
.1060E-03	.5400E-04	.5400E-03	.5400E-03	.9280E-05	.1000E-02	.5400E-02	.5400E-02	.1490E-02	
19	20	21	22	23	24	25	26	27	
.5050E-02	.3560E-01	.1060E-03	.5400E-04	.5400E-03	.5400E-02	.1490E-02	.5050E-02	.3560E-01	
28	29								
.1060E-03	.5400E-04								= Vol
1	2	3	4	5	6	7	8	9	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
10	11	12	13	14	15	16	17	18	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
19	20	21	22	23	24	25	26	27	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
28	29								
.0000E+00	.0000E+00								= B
1	2	3	4	5	6	7	8	9	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
10	11	12	13	14	15	16	17	18	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
19	20	21	22	23	24	25	26	27	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
28	29								
.0000E+00	.0000E+00								= Lambda
.0000E+00									= Js
1	2	3							
.1000E+01	.1000E+01	.1000E+01							= Frac
1	2	3							
.1000E+00	.1000E+00	.1000E+00							= Transp
1	2	3	4	5	6	7	8	9	
.4724E-01	.6356E-02	.3593E+00	.5556E+03	.3704E+03	.1852E+03	.1852E+03	.2000E+01	.0000E+00	
10	11	12	13	14	15	16	17	18	
.1196E+01	.9259E+03	.9259E+03	.1852E+04	.2778E+01	.0000E+00	.1852E+03	.1852E+03	.1852E+03	
19	20	21	22	23	24	25	26	27	
.2000E+01	.0000E+00	.1196E+01	.9259E+03	.9259E+03	.1852E+03	.1852E+03	.2000E+01	.0000E+00	
28	29	30	31	32					
.1196E+01	.9259E+03	.9259E+03	.9259E+03	.3633E+00					= q
1	2	3	4	5	6	7	8	9	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
10	11	12	13	14	15	16	17	18	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
19	20	21	22	23	24	25	26	27	
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	
28	29								
.0000E+00	.0000E+00								= initial

Table 3. Listing of the file SAMPLE.OUT (continued)

PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGHTS RESERVED.

This FILENAME IS --> sample.out

Date : 12-11-87

page 1

Mass In Compartment									
time	1	2	3	4	5	6	7	8	9
.3000E+01	.1756E-02	.3503E-01	.1031E-03	.4718E-04	.4704E-03	.4677E-03	.9862E-03	.2273E-02	.1597E-01
.6000E+01	.2600E-02	.5650E-01	.1663E-03	.7611E-04	.7598E-03	.7573E-03	.1772E-02	.4882E-02	.3437E-01
.9000E+01	.3385E-02	.7643E-01	.2250E-03	.1030E-03	.1028E-02	.1026E-02	.2502E-02	.7307E-02	.5146E-01
.1200E+02	.4113E-02	.9495E-01	.2795E-03	.1279E-03	.1278E-02	.1276E-02	.3180E-02	.9559E-02	.6735E-01
.1500E+02	.4791E-02	.1122E+00	.3302E-03	.1511E-03	.1510E-02	.1508E-02	.3811E-02	.1165E-01	.8211E-01
.1800E+02	.5420E-02	.1282E+00	.3773E-03	.1727E-03	.1726E-02	.1724E-02	.4397E-02	.1360E-01	.9582E-01
.2100E+02	.6005E-02	.1430E+00	.4211E-03	.1927E-03	.1926E-02	.1924E-02	.4941E-02	.1541E-01	.1086E+00
.2400E+02	.6548E-02	.1568E+00	.4617E-03	.2113E-03	.2112E-02	.2111E-02	.5447E-02	.1709E-01	.1204E+00
.2700E+02	.7053E-02	.1697E+00	.4995E-03	.2286E-03	.2285E-02	.2284E-02	.5917E-02	.1865E-01	.1314E+00
.3000E+02	.7522E-02	.1816E+00	.5346E-03	.2447E-03	.2446E-02	.2445E-02	.6353E-02	.2010E-01	.1416E+00
.3300E+02	.7958E-02	.1927E+00	.5673E-03	.2596E-03	.2595E-02	.2594E-02	.6759E-02	.2145E-01	.1512E+00
.3600E+02	.8363E-02	.2030E+00	.5976E-03	.2735E-03	.2734E-02	.2733E-02	.7136E-02	.2270E-01	.1600E+00
.3900E+02	.8740E-02	.2125E+00	.6258E-03	.2864E-03	.2863E-02	.2862E-02	.7487E-02	.2386E-01	.1682E+00
.4200E+02	.9090E-02	.2214E+00	.6519E-03	.2983E-03	.2983E-02	.2982E-02	.7812E-02	.2494E-01	.1758E+00
.4500E+02	.9415E-02	.2297E+00	.6763E-03	.3095E-03	.3094E-02	.3093E-02	.8115E-02	.2595E-01	.1829E+00
.4800E+02	.9717E-02	.2374E+00	.6989E-03	.3198E-03	.3198E-02	.3197E-02	.8396E-02	.2688E-01	.1895E+00
.5100E+02	.9998E-02	.2445E+00	.7199E-03	.3294E-03	.3294E-02	.3293E-02	.8658E-02	.2775E-01	.1956E+00
.5400E+02	.1026E-01	.2511E+00	.7394E-03	.3384E-03	.3383E-02	.3383E-02	.8900E-02	.2856E-01	.2013E+00
.5700E+02	.1050E-01	.2573E+00	.7576E-03	.3467E-03	.3466E-02	.3466E-02	.9126E-02	.2931E-01	.2066E+00
.6000E+02	.1073E-01	.2630E+00	.7744E-03	.3544E-03	.3544E-02	.3543E-02	.9336E-02	.3000E-01	.2115E+00
.6300E+02	.1094E-01	.2683E+00	.7901E-03	.3616E-03	.3615E-02	.3615E-02	.9531E-02	.3065E-01	.2161E+00
.6600E+02	.1113E-01	.2733E+00	.8046E-03	.3682E-03	.3682E-02	.3681E-02	.9712E-02	.3125E-01	.2203E+00
.6900E+02	.1131E-01	.2779E+00	.8182E-03	.3744E-03	.3744E-02	.3743E-02	.9880E-02	.3181E-01	.2242E+00
.7200E+02	.1148E-01	.2821E+00	.8307E-03	.3802E-03	.3801E-02	.3801E-02	.1004E-01	.3233E-01	.2279E+00
.7500E+02	.1164E-01	.2861E+00	.8424E-03	.3855E-03	.3855E-02	.3854E-02	.1018E-01	.3281E-01	.2313E+00
.7800E+02	.1178E-01	.2898E+00	.8533E-03	.3905E-03	.3905E-02	.3904E-02	.1032E-01	.3326E-01	.2345E+00
.8100E+02	.1192E-01	.2932E+00	.8634E-03	.3951E-03	.3951E-02	.3950E-02	.1044E-01	.3368E-01	.2374E+00
.8400E+02	.1204E-01	.2964E+00	.8727E-03	.3994E-03	.3994E-02	.3993E-02	.1056E-01	.3407E-01	.2401E+00
.8700E+02	.1216E-01	.2994E+00	.8815E-03	.4034E-03	.4034E-02	.4033E-02	.1067E-01	.3442E-01	.2427E+00
.9000E+02	.1227E-01	.3021E+00	.8896E-03	.4071E-03	.4071E-02	.4070E-02	.1077E-01	.3476E-01	.2450E+00
.9300E+02	.1237E-01	.3047E+00	.8971E-03	.4105E-03	.4105E-02	.4105E-02	.1086E-01	.3507E-01	.2472E+00
.9600E+02	.1246E-01	.3070E+00	.9041E-03	.4137E-03	.4137E-02	.4137E-02	.1095E-01	.3536E-01	.2493E+00

Table 3. Listing of the file SAMPLE.OUT (continued)

PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGHTS RESERVED.

This FILENAME IS --> sample.out

Date : 12-11-87

page 2

	Mass In Compartment								
time	10	11	12	13	14	15	16	17	18
.3000E+01	.1038E-04	.5288E-05	.5283E-04	.5268E-04	.7391E-05	.4453E-03	.4691E-03	.4664E-03	.9825E-03
.6000E+01	.2231E-04	.1136E-04	.1136E-03	.1134E-03	.1275E-04	.1021E-02	.7586E-03	.7561E-03	.1769E-02
.9000E+01	.3339E-04	.1701E-04	.1700E-03	.1699E-03	.1772E-04	.1577E-02	.1027E-02	.1025E-02	.2499E-02
.1200E+02	.4368E-04	.2225E-04	.2225E-03	.2224E-03	.2235E-04	.2098E-02	.1277E-02	.1275E-02	.3178E-02
.1500E+02	.5325E-04	.2713E-04	.2712E-03	.2711E-03	.2664E-04	.2583E-02	.1509E-02	.1507E-02	.3808E-02
.1800E+02	.6214E-04	.3166E-04	.3165E-03	.3164E-03	.3064E-04	.3033E-02	.1725E-02	.1723E-02	.4394E-02
.2100E+02	.7040E-04	.3586E-04	.3586E-03	.3585E-03	.3435E-04	.3452E-02	.1925E-02	.1923E-02	.4938E-02
.2400E+02	.7808E-04	.3978E-04	.3977E-03	.3976E-03	.3779E-04	.3841E-02	.2111E-02	.2110E-02	.5444E-02
.2700E+02	.8521E-04	.4341E-04	.4341E-03	.4340E-03	.4100E-04	.4203E-02	.2284E-02	.2283E-02	.5915E-02
.3000E+02	.9184E-04	.4679E-04	.4678E-03	.4678E-03	.4397E-04	.4539E-02	.2445E-02	.2444E-02	.6351E-02
.3300E+02	.9800E-04	.4992E-04	.4992E-03	.4991E-03	.4674E-04	.4851E-02	.2595E-02	.2593E-02	.6757E-02
.3600E+02	.1037E-03	.5284E-04	.5284E-03	.5283E-03	.4931E-04	.5141E-02	.2733E-02	.2732E-02	.7135E-02
.3900E+02	.1090E-03	.5555E-04	.5555E-03	.5554E-03	.5170E-04	.5411E-02	.2863E-02	.2861E-02	.7485E-02
.4200E+02	.1140E-03	.5807E-04	.5807E-03	.5806E-03	.5392E-04	.5661E-02	.2982E-02	.2981E-02	.7811E-02
.4500E+02	.1186E-03	.6041E-04	.6041E-03	.6040E-03	.5598E-04	.5894E-02	.3094E-02	.3093E-02	.8114E-02
.4800E+02	.1229E-03	.6258E-04	.6258E-03	.6258E-03	.5790E-04	.6110E-02	.3197E-02	.3196E-02	.8395E-02
.5100E+02	.1268E-03	.6460E-04	.6460E-03	.6460E-03	.5968E-04	.6312E-02	.3294E-02	.3293E-02	.8656E-02
.5400E+02	.1305E-03	.6648E-04	.6648E-03	.6648E-03	.6134E-04	.6498E-02	.3383E-02	.3382E-02	.8899E-02
.5700E+02	.1339E-03	.6823E-04	.6823E-03	.6822E-03	.6287E-04	.6672E-02	.3466E-02	.3465E-02	.9125E-02
.6000E+02	.1371E-03	.6985E-04	.6985E-03	.6984E-03	.6430E-04	.6833E-02	.3543E-02	.3543E-02	.9335E-02
.6300E+02	.1401E-03	.7136E-04	.7135E-03	.7135E-03	.6563E-04	.6983E-02	.3615E-02	.3614E-02	.9530E-02
.6600E+02	.1428E-03	.7276E-04	.7275E-03	.7275E-03	.6687E-04	.7123E-02	.3682E-02	.3681E-02	.9711E-02
.6900E+02	.1454E-03	.7406E-04	.7406E-03	.7405E-03	.6801E-04	.7252E-02	.3744E-02	.3743E-02	.9879E-02
.7200E+02	.1477E-03	.7527E-04	.7527E-03	.7526E-03	.6908E-04	.7372E-02	.3801E-02	.3801E-02	.1004E-01
.7500E+02	.1500E-03	.7639E-04	.7639E-03	.7639E-03	.7007E-04	.7484E-02	.3855E-02	.3854E-02	.1018E-01
.7800E+02	.1520E-03	.7743E-04	.7743E-03	.7743E-03	.7099E-04	.7588E-02	.3904E-02	.3904E-02	.1032E-01
.8100E+02	.1539E-03	.7840E-04	.7840E-03	.7840E-03	.7185E-04	.7684E-02	.3951E-02	.3950E-02	.1044E-01
.8400E+02	.1557E-03	.7931E-04	.7930E-03	.7930E-03	.7264E-04	.7774E-02	.3994E-02	.3993E-02	.1056E-01
.8700E+02	.1573E-03	.8014E-04	.8014E-03	.8014E-03	.7338E-04	.7858E-02	.4033E-02	.4033E-02	.1067E-01
.9000E+02	.1588E-03	.8092E-04	.8092E-03	.8092E-03	.7406E-04	.7935E-02	.4070E-02	.4070E-02	.1077E-01
.9300E+02	.1603E-03	.8165E-04	.8164E-03	.8164E-03	.7470E-04	.8007E-02	.4105E-02	.4105E-02	.1086E-01
.9600E+02	.1616E-03	.8232E-04	.8232E-03	.8232E-03	.7529E-04	.8074E-02	.4137E-02	.4137E-02	.1095E-01

Table 3. Listing of the file SAMPLE.OUT (continued)

PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGHTS RESERVED.

This FILENAME IS --> sample.out

Date : 12-11-87

page 3

Mass In Compartment

time	19	20	21	22	23	24	25	26	27
.3000E+01	.2261E-02	.1588E-01	.1033E-04	.5260E-05	.5249E-04	.4677E-03	.9862E-03	.2273E-02	.1597E-01
.6000E+01	.4871E-02	.3429E-01	.2225E-04	.1134E-04	.1133E-03	.7573E-03	.1772E-02	.4882E-02	.3437E-01
.9000E+01	.7296E-02	.5139E-01	.3334E-04	.1698E-04	.1697E-03	.1026E-02	.2502E-02	.7307E-02	.5146E-01
.1200E+02	.9550E-02	.6728E-01	.4364E-04	.2223E-04	.2222E-03	.1276E-02	.3180E-02	.9559E-02	.6735E-01
.1500E+02	.1164E-01	.8204E-01	.5321E-04	.2711E-04	.2710E-03	.1508E-02	.3811E-02	.1165E-01	.8211E-01
.1800E+02	.1359E-01	.9576E-01	.6210E-04	.3164E-04	.3163E-03	.1724E-02	.4397E-02	.1360E-01	.9582E-01
.2100E+02	.1540E-01	.1085E+00	.7037E-04	.3585E-04	.3584E-03	.1924E-02	.4941E-02	.1541E-01	.1086E+00
.2400E+02	.1708E-01	.1204E+00	.7805E-04	.3976E-04	.3975E-03	.2111E-02	.5447E-02	.1709E-01	.1204E+00
.2700E+02	.1864E-01	.1314E+00	.8518E-04	.4339E-04	.4339E-03	.2284E-02	.5917E-02	.1865E-01	.1314E+00
.3000E+02	.2009E-01	.1416E+00	.9181E-04	.4677E-04	.4677E-03	.2445E-02	.6353E-02	.2010E-01	.1416E+00
.3300E+02	.2144E-01	.1511E+00	.9798E-04	.4991E-04	.4991E-03	.2594E-02	.6759E-02	.2145E-01	.1512E+00
.3600E+02	.2269E-01	.1599E+00	.1037E-03	.5283E-04	.5282E-03	.2733E-02	.7136E-02	.2270E-01	.1600E+00
.3900E+02	.2386E-01	.1682E+00	.1090E-03	.5554E-04	.5553E-03	.2862E-02	.7487E-02	.2386E-01	.1682E+00
.4200E+02	.2494E-01	.1758E+00	.1140E-03	.5806E-04	.5805E-03	.2982E-02	.7812E-02	.2494E-01	.1758E+00
.4500E+02	.2594E-01	.1829E+00	.1186E-03	.6040E-04	.6040E-03	.3093E-02	.8115E-02	.2595E-01	.1829E+00
.4800E+02	.2688E-01	.1895E+00	.1228E-03	.6257E-04	.6257E-03	.3197E-02	.8396E-02	.2688E-01	.1895E+00
.5100E+02	.2775E-01	.1956E+00	.1268E-03	.6460E-04	.6459E-03	.3293E-02	.8658E-02	.2775E-01	.1956E+00
.5400E+02	.2855E-01	.2013E+00	.1305E-03	.6647E-04	.6647E-03	.3383E-02	.8900E-02	.2856E-01	.2013E+00
.5700E+02	.2930E-01	.2066E+00	.1339E-03	.6822E-04	.6822E-03	.3466E-02	.9126E-02	.2931E-01	.2066E+00
.6000E+02	.3000E-01	.2115E+00	.1371E-03	.6984E-04	.6984E-03	.3543E-02	.9336E-02	.3000E-01	.2115E+00
.6300E+02	.3065E-01	.2160E+00	.1401E-03	.7135E-04	.7135E-03	.3615E-02	.9531E-02	.3065E-01	.2161E+00
.6600E+02	.3125E-01	.2203E+00	.1428E-03	.7275E-04	.7275E-03	.3681E-02	.9712E-02	.3125E-01	.2203E+00
.6900E+02	.3181E-01	.2242E+00	.1454E-03	.7405E-04	.7405E-03	.3743E-02	.9880E-02	.3181E-01	.2242E+00
.7200E+02	.3233E-01	.2279E+00	.1477E-03	.7526E-04	.7526E-03	.3801E-02	.1004E-01	.3233E-01	.2279E+00
.7500E+02	.3281E-01	.2313E+00	.1499E-03	.7638E-04	.7638E-03	.3854E-02	.1018E-01	.3281E-01	.2313E+00
.7800E+02	.3326E-01	.2345E+00	.1520E-03	.7743E-04	.7743E-03	.3904E-02	.1032E-01	.3326E-01	.2345E+00
.8100E+02	.3368E-01	.2374E+00	.1539E-03	.7840E-04	.7840E-03	.3950E-02	.1044E-01	.3368E-01	.2374E+00
.8400E+02	.3406E-01	.2401E+00	.1557E-03	.7930E-04	.7930E-03	.3993E-02	.1056E-01	.3407E-01	.2401E+00
.8700E+02	.3442E-01	.2427E+00	.1573E-03	.8014E-04	.8014E-03	.4033E-02	.1067E-01	.3442E-01	.2427E+00
.9000E+02	.3476E-01	.2450E+00	.1588E-03	.8092E-04	.8092E-03	.4070E-02	.1077E-01	.3476E-01	.2450E+00
.9300E+02	.3507E-01	.2472E+00	.1603E-03	.8164E-04	.8164E-03	.4105E-02	.1086E-01	.3507E-01	.2472E+00
.9600E+02	.3536E-01	.2492E+00	.1616E-03	.8231E-04	.8231E-03	.4137E-02	.1095E-01	.3536E-01	.2493E+00

Table 3. Listing of the file SAMPLE.OUT (continued)

PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGHTS RESERVED.
This FILENAME IS --> sample.out

Date : 12-11-87

page 4

Mass In Compartment

time	28	29
.3000E+01	.1038E-04	.5288E-05
.6000E+01	.2231E-04	.1136E-04
.9000E+01	.3339E-04	.1701E-04
.1200E+02	.4368E-04	.2225E-04
.1500E+02	.5325E-04	.2713E-04
.1800E+02	.6214E-04	.3166E-04
.2100E+02	.7040E-04	.3586E-04
.2400E+02	.7808E-04	.3978E-04
.2700E+02	.8521E-04	.4341E-04
.3000E+02	.9184E-04	.4679E-04
.3300E+02	.9800E-04	.4992E-04
.3600E+02	.1037E-03	.5284E-04
.3900E+02	.1090E-03	.5555E-04
.4200E+02	.1140E-03	.5807E-04
.4500E+02	.1186E-03	.6041E-04
.4800E+02	.1229E-03	.6258E-04
.5100E+02	.1268E-03	.6460E-04
.5400E+02	.1305E-03	.6648E-04
.5700E+02	.1339E-03	.6823E-04
.6000E+02	.1371E-03	.6985E-04
.6300E+02	.1401E-03	.7136E-04
.6600E+02	.1428E-03	.7276E-04
.6900E+02	.1454E-03	.7406E-04
.7200E+02	.1477E-03	.7527E-04
.7500E+02	.1500E-03	.7639E-04
.7800E+02	.1520E-03	.7743E-04
.8100E+02	.1539E-03	.7840E-04
.8400E+02	.1557E-03	.7931E-04
.8700E+02	.1573E-03	.8014E-04
.9000E+02	.1588E-03	.8092E-04
.9300E+02	.1603E-03	.8165E-04
.9600E+02	.1616E-03	.8232E-04

Table 3. Listing of the file SAMPLE.OUT (continued)

PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGHTS RESERVED.
 This FILENAME is --> sample.out

Date : 12-11-87

page 1

	Mass		Loss		In		Compartment		
time	1	2	3	4	5	6	7	8	9
.3000E+01	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.6000E+01	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.9000E+01	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.1200E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.1500E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.1800E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.2100E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.2400E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.2700E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.3000E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.3300E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.3600E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.3900E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.4200E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.4500E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.4800E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.5100E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.5400E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.5700E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.6000E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.6300E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.6600E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.6900E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.7200E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.7500E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.7800E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.8100E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.8400E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.8700E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.9000E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.9300E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.9600E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00

Table 3. Listing of the file SAMPLE.OUT (continued)

PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGHTS RESERVED.

This FILENAME is --> sample.out

Date : 12-11-87

page 3

	Mass		Loss		In		Compartment		
time	19	20	21	22	23	24	25	26	27
.3000E+01	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.6000E+01	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.9000E+01	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.1200E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.1500E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.1800E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.2100E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.2400E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.2700E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.3000E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.3300E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.3600E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.3900E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.4200E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.4500E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.4800E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.5100E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.5400E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.5700E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.6000E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.6300E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.6600E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.6900E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.7200E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.7500E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.7800E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.8100E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.8400E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.8700E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.9000E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.9300E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
.9600E+02	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00

Table 3. Listing of the file SAMPLE.OUT (continued)

PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGHTS RESERVED.

This FILENAME is --> sample.out

Date : 12-11-87

page 4

	Mass	Loss	In	Compartment
time	28	29		
.3000E+01	.0000E+00	.0000E+00		
.6000E+01	.0000E+00	.0000E+00		
.9000E+01	.0000E+00	.0000E+00		
.1200E+02	.0000E+00	.0000E+00		
.1500E+02	.0000E+00	.0000E+00		
.1800E+02	.0000E+00	.0000E+00		
.2100E+02	.0000E+00	.0000E+00		
.2400E+02	.0000E+00	.0000E+00		
.2700E+02	.0000E+00	.0000E+00		
.3000E+02	.0000E+00	.0000E+00		
.3300E+02	.0000E+00	.0000E+00		
.3600E+02	.0000E+00	.0000E+00		
.3900E+02	.0000E+00	.0000E+00		
.4200E+02	.0000E+00	.0000E+00		
.4500E+02	.0000E+00	.0000E+00		
.4800E+02	.0000E+00	.0000E+00		
.5100E+02	.0000E+00	.0000E+00		
.5400E+02	.0000E+00	.0000E+00		
.5700E+02	.0000E+00	.0000E+00		
.6000E+02	.0000E+00	.0000E+00		
.6300E+02	.0000E+00	.0000E+00		
.6600E+02	.0000E+00	.0000E+00		
.6900E+02	.0000E+00	.0000E+00		
.7200E+02	.0000E+00	.0000E+00		
.7500E+02	.0000E+00	.0000E+00		
.7800E+02	.0000E+00	.0000E+00		
.8100E+02	.0000E+00	.0000E+00		
.8400E+02	.0000E+00	.0000E+00		
.8700E+02	.0000E+00	.0000E+00		
.9000E+02	.0000E+00	.0000E+00		
.9300E+02	.0000E+00	.0000E+00		
.9600E+02	.0000E+00	.0000E+00		

Table 3. Listing of the file SAMPLE.OUT (continued)

PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGHTS RESERVED.
 This FILENAME is -->sample.out

Date : 12-11-87

page 1

time	Grouped Mass						Total =	plant	inflow
	roots	stems	leaf1	leaf2	leaf3	lambda			
.3000E+01	.3735E-01	.2562E-02	.1914E-01	.1924E-01	.1924E-01	.0000E+00	.9753E-01	.9788E-01	
.6000E+01	.6030E-01	.4240E-02	.4095E-01	.4104E-01	.4104E-01	.0000E+00	.1876E+00	.1879E+00	
.9000E+01	.8164E-01	.5797E-02	.6122E-01	.6130E-01	.6130E-01	.0000E+00	.2713E+00	.2716E+00	
.1200E+02	.1015E+00	.7244E-02	.8005E-01	.8013E-01	.8013E-01	.0000E+00	.3490E+00	.3494E+00	
.1500E+02	.1199E+00	.8588E-02	.9755E-01	.9762E-01	.9762E-01	.0000E+00	.4213E+00	.4216E+00	
.1800E+02	.1370E+00	.9838E-02	.1138E+00	.1139E+00	.1139E+00	.0000E+00	.4884E+00	.4888E+00	
.2100E+02	.1529E+00	.1100E-01	.1289E+00	.1290E+00	.1290E+00	.0000E+00	.5508E+00	.5512E+00	
.2400E+02	.1677E+00	.1208E-01	.1430E+00	.1430E+00	.1430E+00	.0000E+00	.6088E+00	.6092E+00	
.2700E+02	.1815E+00	.1308E-01	.1560E+00	.1561E+00	.1561E+00	.0000E+00	.6627E+00	.6630E+00	
.3000E+02	.1942E+00	.1401E-01	.1681E+00	.1682E+00	.1682E+00	.0000E+00	.7128E+00	.7131E+00	
.3300E+02	.2061E+00	.1488E-01	.1794E+00	.1795E+00	.1795E+00	.0000E+00	.7593E+00	.7596E+00	
.3600E+02	.2171E+00	.1568E-01	.1899E+00	.1899E+00	.1899E+00	.0000E+00	.8025E+00	.8029E+00	
.3900E+02	.2274E+00	.1643E-01	.1996E+00	.1997E+00	.1997E+00	.0000E+00	.8427E+00	.8431E+00	
.4200E+02	.2369E+00	.1712E-01	.2087E+00	.2087E+00	.2087E+00	.0000E+00	.8800E+00	.8804E+00	
.4500E+02	.2457E+00	.1777E-01	.2171E+00	.2171E+00	.2171E+00	.0000E+00	.9147E+00	.9151E+00	
.4800E+02	.2539E+00	.1837E-01	.2249E+00	.2249E+00	.2249E+00	.0000E+00	.9470E+00	.9473E+00	
.5100E+02	.2616E+00	.1893E-01	.2321E+00	.2321E+00	.2321E+00	.0000E+00	.9769E+00	.9773E+00	
.5400E+02	.2687E+00	.1945E-01	.2389E+00	.2389E+00	.2389E+00	.0000E+00	.1005E+01	.1005E+01	
.5700E+02	.2753E+00	.1993E-01	.2451E+00	.2452E+00	.2452E+00	.0000E+00	.1031E+01	.1031E+01	
.6000E+02	.2814E+00	.2037E-01	.2509E+00	.2510E+00	.2510E+00	.0000E+00	.1055E+01	.1055E+01	
.6300E+02	.2871E+00	.2079E-01	.2564E+00	.2564E+00	.2564E+00	.0000E+00	.1077E+01	.1077E+01	
.6600E+02	.2924E+00	.2118E-01	.2614E+00	.2614E+00	.2614E+00	.0000E+00	.1098E+01	.1098E+01	
.6900E+02	.2973E+00	.2154E-01	.2661E+00	.2661E+00	.2661E+00	.0000E+00	.1117E+01	.1117E+01	
.7200E+02	.3019E+00	.2187E-01	.2704E+00	.2704E+00	.2704E+00	.0000E+00	.1135E+01	.1135E+01	
.7500E+02	.3061E+00	.2218E-01	.2744E+00	.2744E+00	.2744E+00	.0000E+00	.1152E+01	.1152E+01	
.7800E+02	.3101E+00	.2247E-01	.2782E+00	.2782E+00	.2782E+00	.0000E+00	.1167E+01	.1167E+01	
.8100E+02	.3138E+00	.2273E-01	.2817E+00	.2817E+00	.2817E+00	.0000E+00	.1182E+01	.1182E+01	
.8400E+02	.3172E+00	.2298E-01	.2849E+00	.2849E+00	.2849E+00	.0000E+00	.1195E+01	.1195E+01	
.8700E+02	.3203E+00	.2321E-01	.2879E+00	.2879E+00	.2879E+00	.0000E+00	.1207E+01	.1208E+01	
.9000E+02	.3233E+00	.2343E-01	.2907E+00	.2907E+00	.2907E+00	.0000E+00	.1219E+01	.1219E+01	
.9300E+02	.3260E+00	.2363E-01	.2933E+00	.2933E+00	.2933E+00	.0000E+00	.1230E+01	.1230E+01	
.9600E+02	.3286E+00	.2381E-01	.2957E+00	.2957E+00	.2957E+00	.0000E+00	.1240E+01	.1240E+01	

Table 3. Listing of the file SAMPLE.OUT (continued)

PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGHTS RESERVED.

This FILENAME is --> sample.out

Date : 12-11-87

page 1

	Grouped			Mass	with	Lambda		
time	roots	+ stems	+ leaf1	+ leaf2	+ leaf3	= plant	inflow	
.3000E+01	.3735E-01	.2562E-02	.1914E-01	.1924E-01	.1924E-01	.9753E-01	.9788E-01	
.6000E+01	.6030E-01	.4240E-02	.4095E-01	.4104E-01	.4104E-01	.1876E+00	.1879E+00	
.9000E+01	.8164E-01	.5797E-02	.6122E-01	.6130E-01	.6130E-01	.2713E+00	.2716E+00	
.1200E+02	.1015E+00	.7244E-02	.8005E-01	.8013E-01	.8013E-01	.3490E+00	.3494E+00	
.1500E+02	.1199E+00	.8588E-02	.9755E-01	.9762E-01	.9762E-01	.4213E+00	.4216E+00	
.1800E+02	.1370E+00	.9838E-02	.1138E+00	.1139E+00	.1139E+00	.4884E+00	.4888E+00	
.2100E+02	.1529E+00	.1100E-01	.1289E+00	.1290E+00	.1290E+00	.5508E+00	.5512E+00	
.2400E+02	.1677E+00	.1208E-01	.1430E+00	.1430E+00	.1430E+00	.6088E+00	.6092E+00	
.2700E+02	.1815E+00	.1308E-01	.1560E+00	.1561E+00	.1561E+00	.6627E+00	.6630E+00	
.3000E+02	.1942E+00	.1401E-01	.1681E+00	.1682E+00	.1682E+00	.7128E+00	.7131E+00	
.3300E+02	.2061E+00	.1488E-01	.1794E+00	.1795E+00	.1795E+00	.7593E+00	.7596E+00	
.3600E+02	.2171E+00	.1568E-01	.1899E+00	.1899E+00	.1899E+00	.8025E+00	.8029E+00	
.3900E+02	.2274E+00	.1643E-01	.1996E+00	.1997E+00	.1997E+00	.8427E+00	.8431E+00	
.4200E+02	.2369E+00	.1712E-01	.2087E+00	.2087E+00	.2087E+00	.8800E+00	.8804E+00	
.4500E+02	.2457E+00	.1777E-01	.2171E+00	.2171E+00	.2171E+00	.9147E+00	.9151E+00	
.4800E+02	.2539E+00	.1837E-01	.2249E+00	.2249E+00	.2249E+00	.9470E+00	.9473E+00	
.5100E+02	.2616E+00	.1893E-01	.2321E+00	.2321E+00	.2321E+00	.9769E+00	.9773E+00	
.5400E+02	.2687E+00	.1945E-01	.2389E+00	.2389E+00	.2389E+00	.1005E+01	.1005E+01	
.5700E+02	.2753E+00	.1993E-01	.2451E+00	.2452E+00	.2452E+00	.1031E+01	.1031E+01	
.6000E+02	.2814E+00	.2037E-01	.2509E+00	.2510E+00	.2510E+00	.1055E+01	.1055E+01	
.6300E+02	.2871E+00	.2079E-01	.2564E+00	.2564E+00	.2564E+00	.1077E+01	.1077E+01	
.6600E+02	.2924E+00	.2118E-01	.2614E+00	.2614E+00	.2614E+00	.1098E+01	.1098E+01	
.6900E+02	.2973E+00	.2154E-01	.2661E+00	.2661E+00	.2661E+00	.1117E+01	.1117E+01	
.7200E+02	.3019E+00	.2187E-01	.2704E+00	.2704E+00	.2704E+00	.1135E+01	.1135E+01	
.7500E+02	.3061E+00	.2218E-01	.2744E+00	.2744E+00	.2744E+00	.1152E+01	.1152E+01	
.7800E+02	.3101E+00	.2247E-01	.2782E+00	.2782E+00	.2782E+00	.1167E+01	.1167E+01	
.8100E+02	.3138E+00	.2273E-01	.2817E+00	.2817E+00	.2817E+00	.1182E+01	.1182E+01	
.8400E+02	.3172E+00	.2298E-01	.2849E+00	.2849E+00	.2849E+00	.1195E+01	.1195E+01	
.8700E+02	.3203E+00	.2321E-01	.2879E+00	.2879E+00	.2879E+00	.1207E+01	.1208E+01	
.9000E+02	.3233E+00	.2343E-01	.2907E+00	.2907E+00	.2907E+00	.1219E+01	.1219E+01	
.9300E+02	.3260E+00	.2363E-01	.2933E+00	.2933E+00	.2933E+00	.1230E+01	.1230E+01	
.9600E+02	.3286E+00	.2381E-01	.2957E+00	.2957E+00	.2957E+00	.1240E+01	.1240E+01	

APPENDIX A. Program code structure

Structure of the program codes were broken down into seven program modules (files). The name for each program module and its functional purpose are listed below:

1. COMBLOC (include file)
2. PLANT3V2.FOR (main program)
3. DTADR3V2.FOR (subroutines)
4. MTXA3V2.FOR (subroutines)
5. STIFF3V2.FOR (subroutines)
6. DECINT.FOR (subroutines)
7. PRINT3V2.FOR (subroutines)

File COMBLOC is an include file which consists of the global arrays and variables. This include file is shared by the main program and other subroutines. All computations are done in double precisions.

File PLANT3V2.FOR is the main program module which controls the input/output units and transfer controls to other subroutines to perform proper simulations (see Figure 6).

File DTADR3V2.FOR is the data driver program module which is used to control the data input process (see Figure 7a).

File MTXA3V2.FOR is the matrix constructor program module which transforms the input data into a two dimensional matrix that is used to represent the system (see Figure 8a).

File STIFF3V2.FOR is the simulation program module which simulates the mass flow in the plant (see Figures 9 and 10).

File DECINT.FOR is an integration routine driver which calls an integration method by C.W. Gear; "DIFSUB for Solution of Ordinary Differential Equations"; Algorithm 407; CACM 14:3, March 1971.

File PRINT3V2.FOR is the output listing program module which consist all the output headers and formats.

APPENDIX B. Program flow diagrams

Illustrations of the program logical flow are shown in the following figures:

- Figure 6. Main program flow diagram
- Figure 7. Subroutine (a) data driver and (b) load data flow diagrams
- Figure 8. Subroutine (a) matrix a and (b) initial value flow diagrams
- Figure 9. Subroutine stiff flow diagram
- Figure 10. Subroutine intgr1 flow diagram
- Figure 11. Subroutine (a) outmass and (b) outmlos flow diagrams
- Figure 12. Subroutine (a) outgroup and (b) outrslwl flow diagrams

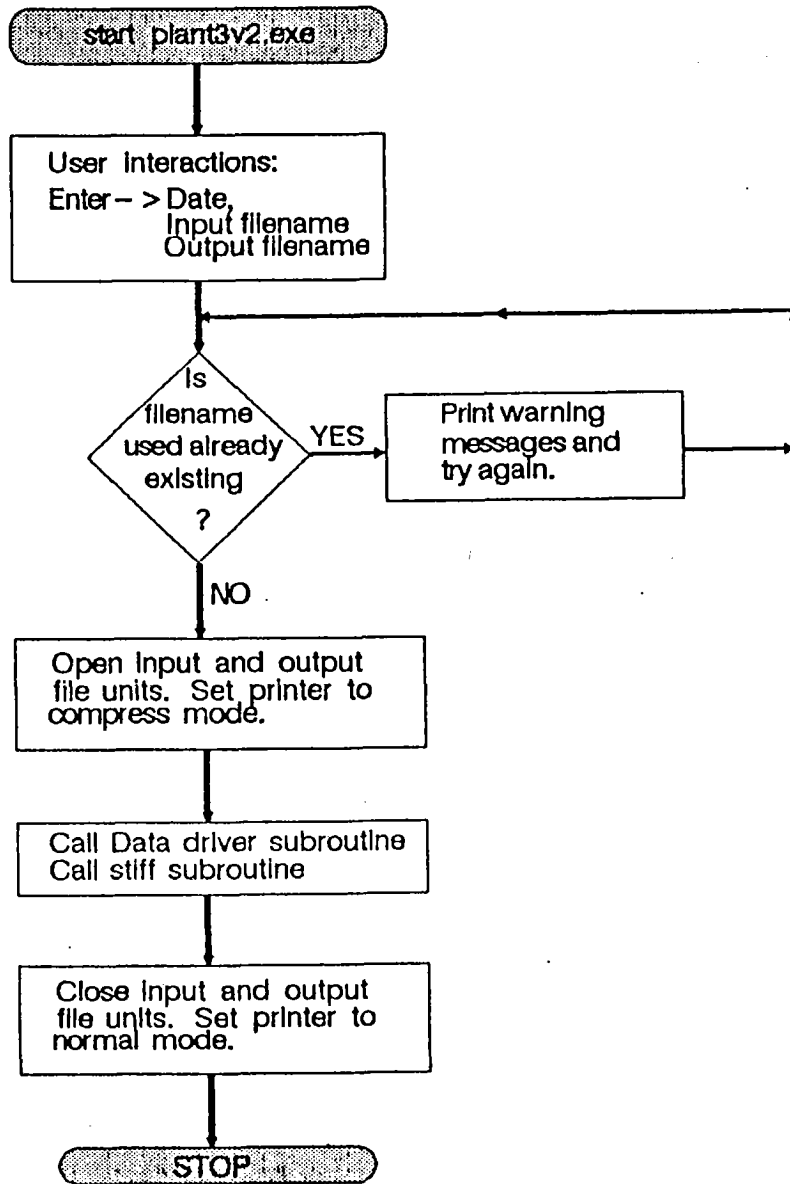


Figure 6. Main program flow diagram

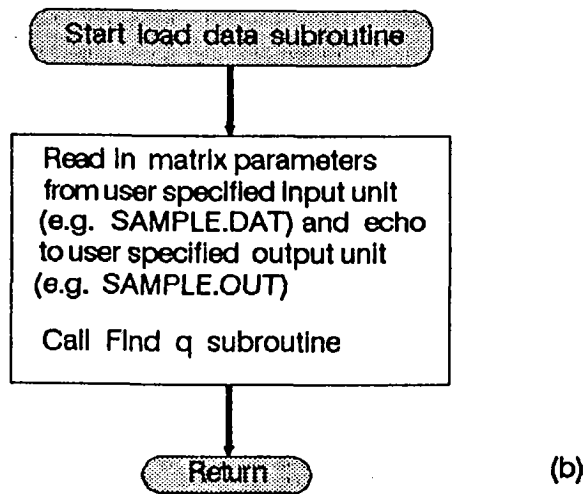
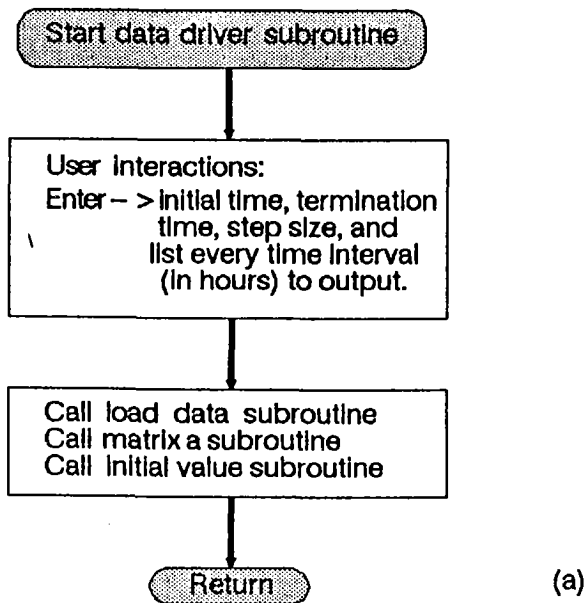


Figure 7. Subroutine (a) data driver and (b) load data flow diagrams

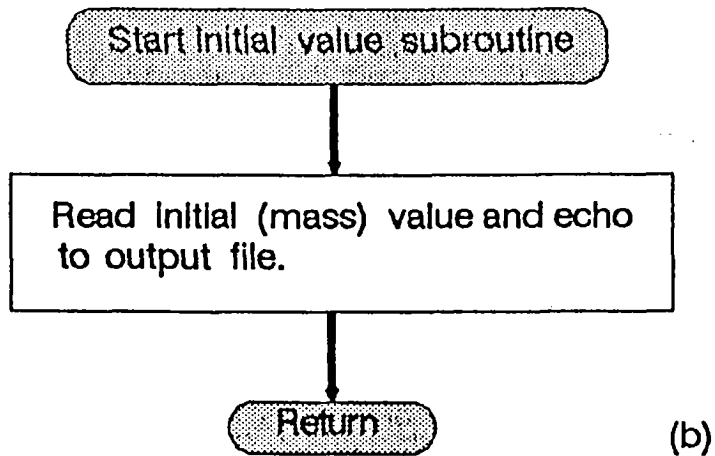
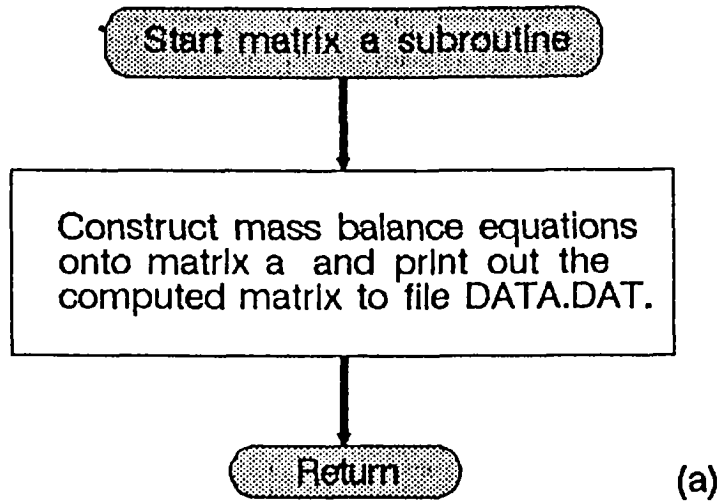


Figure 8. Subroutine (a) matrix a and (b) initial value flow diagrams

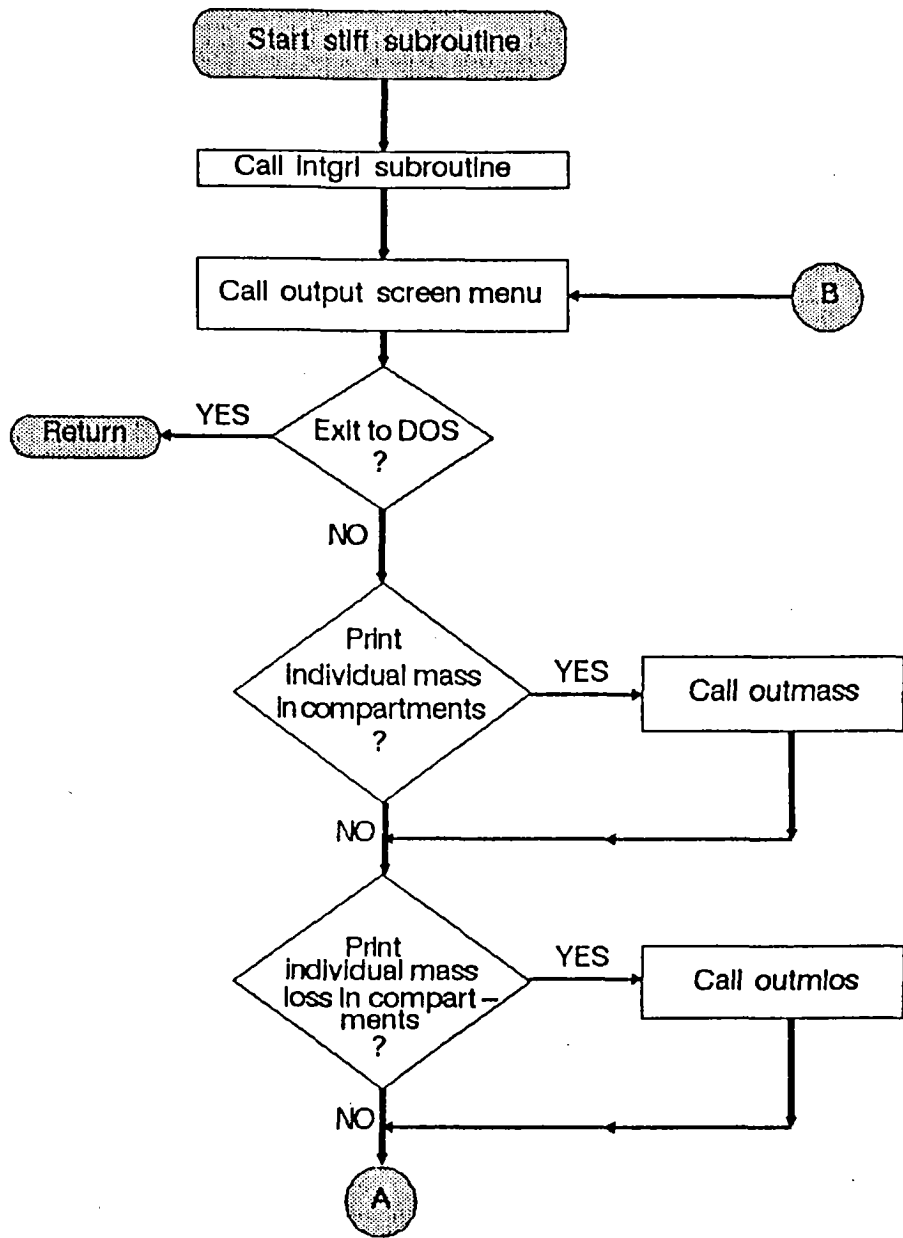


Figure 8. Subroutine stiff flow diagram

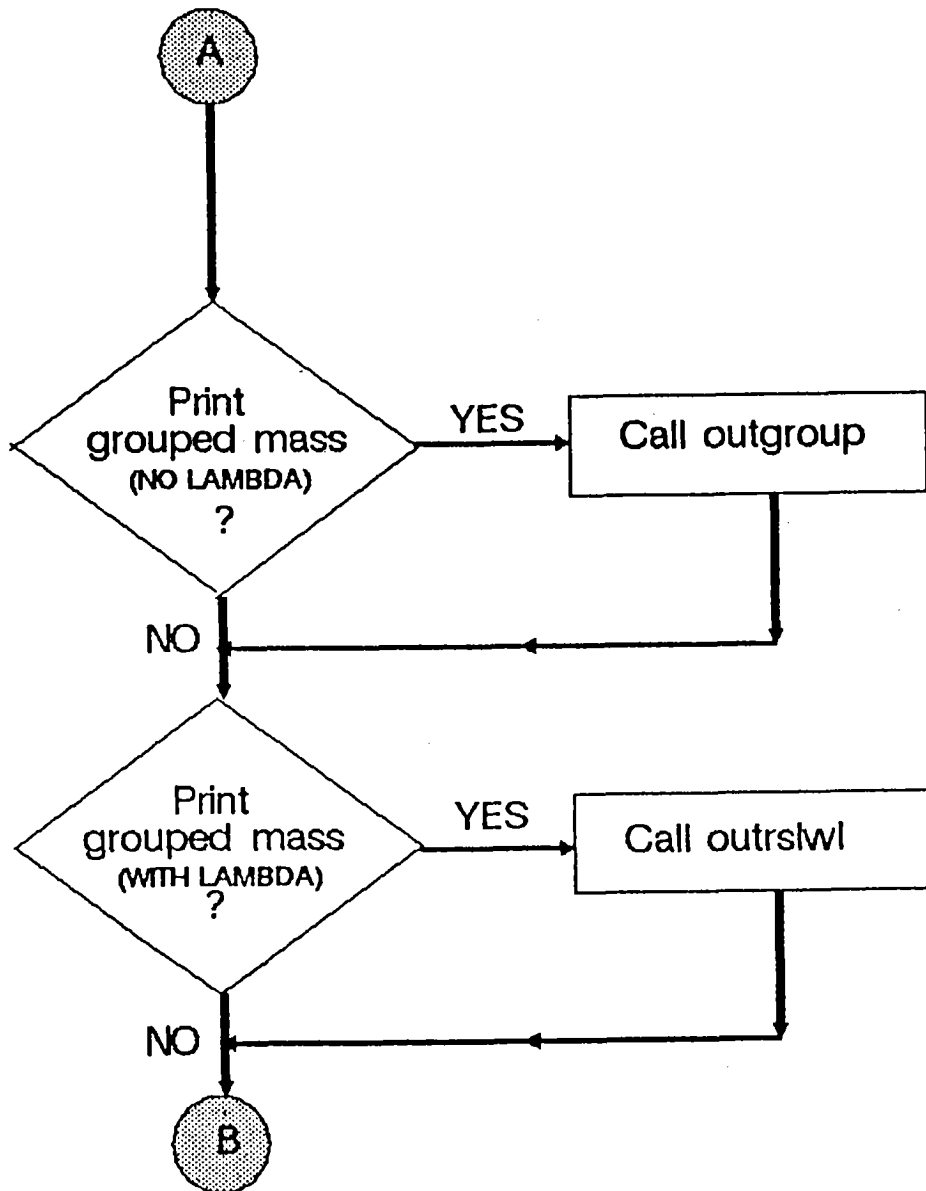


Figure 9. Subroutine stiff flow diagrams (continued)

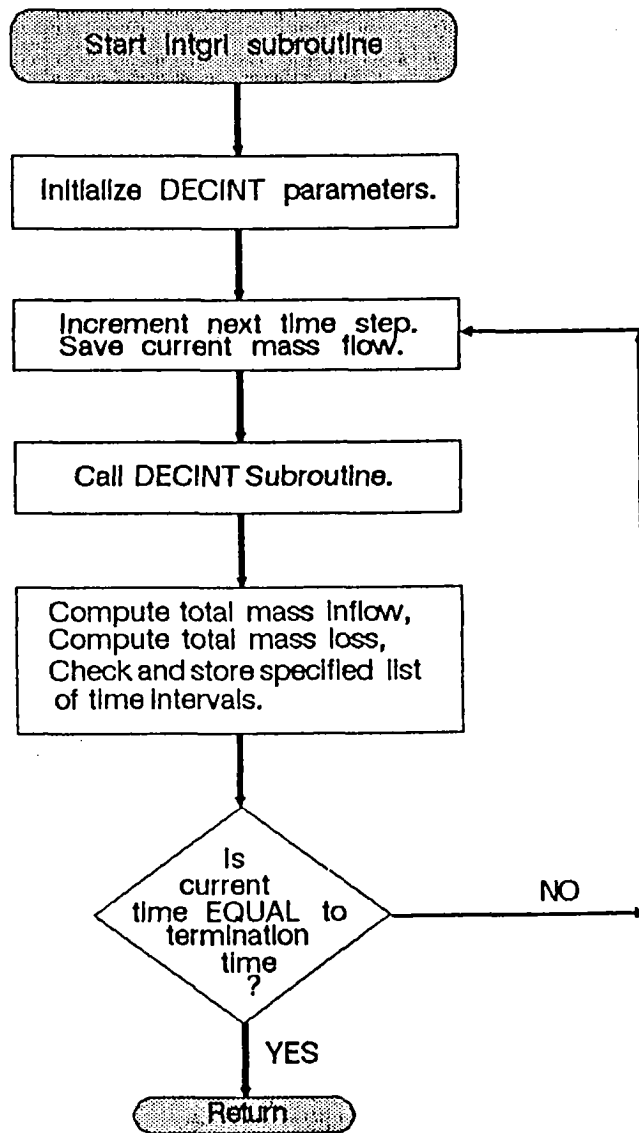


Figure 10. Subroutine intgrl flow diagram

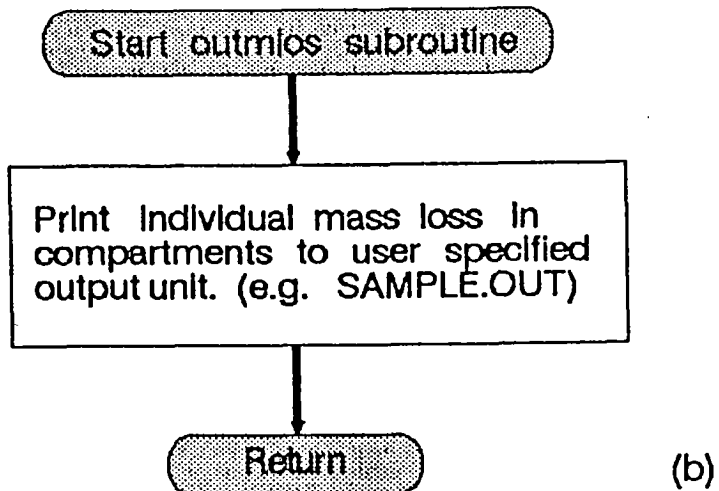
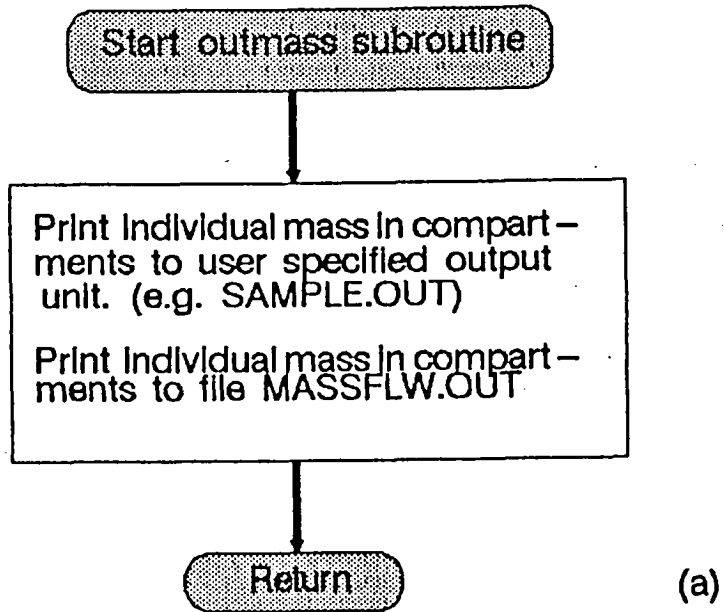


Figure 11. Subroutine (a) outmass and (b) outmlos flow diagrams

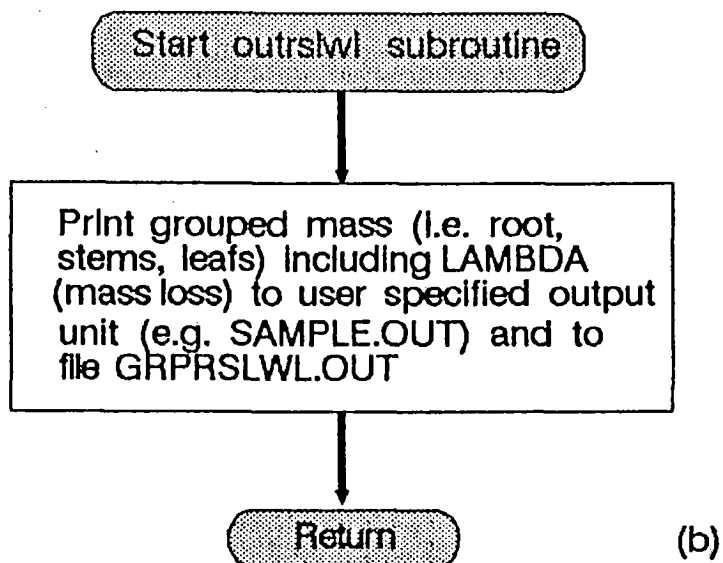
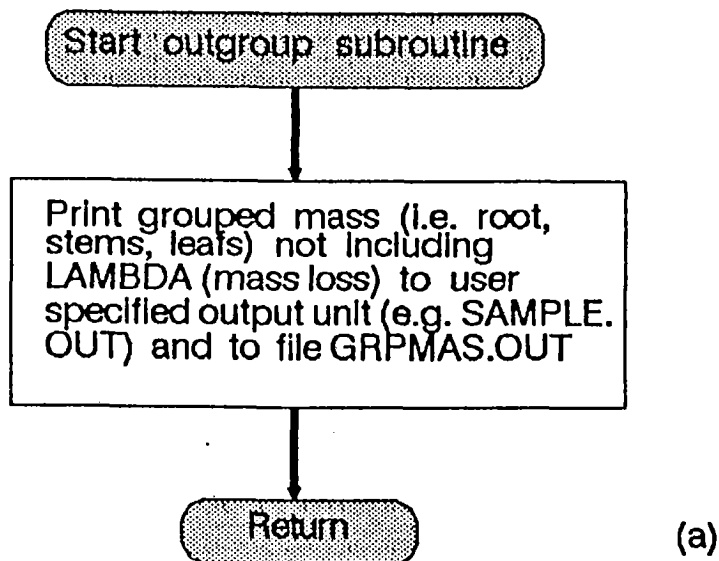


Figure 12. Subroutine (a) outgroup and (b) outrslwl flow diagrams

APPENDIX C. Program source codes

Program source codes are listed in sequential order as mentioned in APPENDIX A.

<u>File</u>	PROGRAM	Page
1. COMBLOC	(include file)	38
2. PLANT3V2.FOR	(main program)	39
3. DTADR3V2.FOR	(subroutines)	45
4. MTXA3V2.FOR	(subroutines)	53
5. STIFF3V2.FOR	(subroutines)	57
6. DECINT.FOR	(subroutines)	65
7. PRINT3V2.FOR	(subroutines)	71

```

        implicit double precision(a-h, o-z)
C
C
C-----THIS IS A STAND ALONE INCLUDE FILE NAMED "COMBLOC"
C
C-----NCPMT = NUMBER OF COMPARTMENTS
C      NWALLS = NUMBER OF WALLS IN TOTAL COMPARTMENTS
C
      integer    ncpmt, nwalls
      parameter (ncpmt=29, nwalls=32)
      real*8 kr,kl
      common /blk1/area(nwalls),difu(nwalls),kr(nwalls),kl(nwalls),
*          q(nwalls),sigma(nwalls)
      real*8 lambda
      common /blk2/deltax(nwalls),lambda(ncpmt),vol(ncpmt),sb(ncpmt)
      common /blk3/printvl,tintial,tstop,dt,npage,itime,tprt,tol
      real*8 mass,mlos,msorb
      common /blk4/mass(200,31),mlos(200,31),msorb(200,30)
      common /blk5/xnow(ncpmt),xold(ncpmt),frac(3),tr(3)
      real*8 js,lost
      common /blk6/js,conc0,a(ncpmt,ncpmt),lost(ncpmt)
      character date*8
      character*20 infile, outfile, pghd*70
      common /blk7/ infile, outfile, pghd, date

```

```

c
c   T H I S   I S   F I L E   " P L A N T 3 V 2 . F O R "
c
c*-----*
c*
c* P L A N T   T R A N S P O R T   T H R E E   L E A F   *
c*                               M O D E L   *
c*                               (version 2.1)   *
c*
c* Program manager : Dr. Craig McFarlane   *
c*
c* Programmer : Kai P. Suen   6/1/1987   *
c*               Oregon State University   *
c*               Soil Science Department   *
c*               Corvallis, Oregon 97331   *
c*
c* Program developed for EPA grant no. CR811940-01-03   *
c*
c*               All rights reserved   *
c*
c* THIS IS THE MAIN PROGRAM AND IS USED TO SET UP THE SYSTEM FOR *
C* INPUT, OUTPUT DEVICES AND PASS CONTROL TO THE DATA DRIVER *
C* SUBROUTINE TO INPUT DATA AND PREPARE SYSTEM MASS BALANCE *
C* EQUATIONS FOR SIMULATION, THEN PASS CONTROL TO THE *
C* SIMULATOR SUBROUTINE TO SIMULATE SOLUTE TRANSPORT. IT USES *
C* AN INCLUDE COMMAND TO INCLUDE A FILE NAMED "COMBLOC" WHICH *
C* STORES ALL THE GLOBAL ARRAYS AND VARIABLES. *
c*
c* VARIABLE DEFINITION *
c*-----*
C*   REPLACE      : SINGLE CHARACTER VARIABLE USED TO CHECK *
C*                 (Y/N) TO REPLACE NEW OUTPUT FILENAME. *
C*   EXIST        : LOGICAL VARIABLE TO CHECK WHETHER AN *
C*                 OUTPUT FILENAME GIVEN BY USER IS ALREADY *
C*                 EXISTED. *
c*
c* FUNCTION CALLS *
c*-----*
C*   CALL CLRSCRN : CLEAR CURRENT SCREEN. ACTUAL CODES ARE *
C*                 STORED IN FILE "PLANT3V2.FOR" *
C*   CALL SCRNH D : PRINT SCREEN HEADER. ACTUAL CODES ARE *
C*                 STORED IN FILE "PLANT3V2.FOR" *
C*   CALL MSSGE1  : PRINT MESSAGE TO SCREEN. ACTUAL CODES ARE *
C*                 STORED IN FILE "PLANT3V2.FOR" *
C*   CALL SETPR( ) : SET PRINTER TO COMPRESS MODE WHEN ('ON'), *
C*                 SET PRINTER TO NORMAL MODE WHEN ('OFF'). *
C*                 ONLY ONE ARGUMENT IS PASSED. ACTUAL CODES *
C*                 ARE STORED IN FILE "PLANT3V2.FOR" *
C*   CALL DTADRVR : TRANSFER CONTROL TO DATA DRIVER SUBROUTINE *
C*                 ACTUAL CODES ARE STORED IN "DTADR3V2.FOR" *

```

```

C*      CALL STIFF      : TRANSFER CONTROL TO SIMULATOR SUBROUTINE      *
C*      CALL BYE       : PRINT MESSAGE TO SCREEN. ACTUAL CODES ARE      *
C*                        STORED IN FILE "PLANT3v2.FOR"                  *
C*-----*
c
      program plant3V2
$include:'combloc'
      real start0, elapsd, second
      character replace*1
      logical exist
      external second

C
C-----INPUT FORMAT
C
      1000 format(a)

C
C-----OUTPUT FORMATS
C
      2000 format(//,16x,'Enter today''s date (e.g. 04-01-86) >- ',\ )
      2010 format(16x,'Enter input filename >- ',\ )
      2020 format(16x,'Enter filename to store output or',/,
      *      16x,'      "prn:" output to line printer >- ',\ )
      2030 format(5(/),16x,' **Warning**',/,
      *      16x,'      The filename you choose is being used.',//,
      *      16x,'      Do you want to replace (write over) ',
      *      ' this file (y/n) >- ',\ )

C
C-----The statements commented out with "Ct" were for timing purposes
C
Ct 2040 format(/,16x,'Integration took ',f12.3,' seconds.')
Ct 2050 format(/,16x,'Accuracy check for 1st & last elements:',//,
Ct      & ( 16x,f10.4,8x,e20.15,5x,e20.15,/ ) )

C
C*-----*
C*      B E G I N   M A I N   P R O G R A M      *
C*-----*
C
C-----CLEAR SCREEN AND PRINT SCREEN HEADER
C
      call clrscrn
      call scrnhd

C
C-----ENTER CURRENT DATE FROM SCREEN
C
      write(*,2000)
      read(*,1000) date

C
C-----ENTER DATA FILENAME FROM SCREEN
C
      write(*,2010)

```

```

        read(*,1000) infile
C
C-----ENTER OUTPUT FILENAME FROM SCREEN
    10 write(*,2020)
        read(*,1000) outfile
        if (outfile(1:4) .eq. 'prn:') then
C
C-----CLEAR DISPLAY AND PRINT MESSAGE
C
        call clrscrn
        call scrnhd
        call mssgel
        call clrscrn
        call scrnhd
    else
C
C-----CHECK FOR EXISTING OUTPUT FILENAME, IF FILENAME IS ALREADY
C     EXISTED THEN ALLOW USER TO CHANGE ANOTHER FILENAME OR
C     WRITE OVER THE OLD FILENAME.
C
        inquire(file=outfile,exist=exist)
        if (exist) then
            call clrscrn
            call scrnhd
            write(*,2030)
            read(*,1000) replace
            if ((replace.eq.'n').or.(replace.eq.'N')) goto 10
            call clrscrn
            call scrnhd
        endif
    endif
C
C-----OPEN FILES
C
        open(1,file=infile)
        open(2,file='data.out')
        open(3,file=outfile)
        open(4,file='massflw.out')
        open(5,file='grpmas.out')
        open(6,file='grprslwl.out')
C
C-----SET PRINTER TO COMPRESSED MODE
C
        call setpr('on ')
C
C-----EXECUTE DATA DRIVER; ACTUAL CODING IS IN FILE DTADR3V2.FOR
C
        call dtadv
C
C-----EXECUTE SIMULATOR; ACTUAL CODING IS IN FILE STIFF3V2.FOR
Ct     start0 = second()

```

```

call stiff
Ct      elapsd = second() - start0
Ct      write(3,2040) elapsd
Ct      write(3,2050) mass(1,1), mass(1,2), mass(1,30),
Ct      &          mass(itime,1), mass(itime,2), mass(itime,30)
C
C-----SET PRINTER TO NORMAL MODE
call setpr('off')
call clrscrn
call scrnhd
call bye
Ct      write(*,2040) elapsd
C
C-----CLOSE FILES
close(1)
close(2)
close(3)
close(4)
close(5)
close(6)

C
C*-----*
C*          E N D      M A I N      P R O G R A M          *
C*-----*
C
      stop
      end

c
c*-----*
c*          S U B R O U T I N E      M S S G E 1          *
c* THIS SUBROUTINE PRINTS AN ADVICE MESSAGE TO CONSOLE TO REMIND *
c* USER TO TURN PRINTER POWER ON AND SET PRINTER SPOOLER LARGER *
c* THIS SUBROUTINE IS CALLED FROM THE MAIN PROGRAM WHICH IS *
c* CONTAINED IN THIS FILE. *
c*-----*
c
      subroutine mssgel
      write(*,2000)
2000 format(
* ///,16x,' Set printer power on to receive output.',
* ///,16x,' Recommendation:',/,16x,' *****',//,
* 16x, ' 1. Set printer buffer >= 64k to speed up runtime.',
* 5(//)
      pause
      return
      end

c
c*-----*
c*          S U B R O U T I N E      S C R N H D          *
c* THIS SUBROUTINE PRINTS THE SCREEN HEADER AND IS BEING CALLED *
c* FROM MANY OTHER FILES. *

```

```

c*-----*
c
  subroutine scrnhd
    write(*,2000)
  2000 format(16x,'
    *      16x,'      PLANT TRANSPORT THREE LEAF MODEL (ver 2.1)
    *      16x,'      ALL RIGHTS RESERVED.
    *      16x,'
    return
  end

c
c*-----*
c*          S U B R O U T I N E          B Y E          *
c* THIS SUBROUTINE PRINTS MESSAGE TO CONSOLE BEFORE LEAVING THE *
c* PLANT TRANSPORT THREE LEAVES MODEL PROGRAM. THE MESSAGE *
c* INDICATED WHERE AND WHAT THE OUTPUT FILES ARE NAMED. *
c* THIS SUBROUTINE CALLED FROM THE MAIN PROGRAM WHICH IS THIS *
c* FILE. *
c*-----*
  subroutine bye
  $include:'combloc'
    write(*,2000) outfile
  2000 format(5(/),
    * 16x,' You are now leaving PROGRAM PLANT3V2 [version 2.1]'
    * 5(/),16x,' You can retrieve your output from FILENAME --> ',a/,
    * 16x,' or line printer if you chose prn: as output device.',///)
    return
  end

c
c*-----*
c*          S U B R O U T I N E          S E T P R          *
c* THIS SUBROUTINE SET PRINTER TO COMPRESS MODE AND NORMAL MODE *
c* AND IS CALLED FROM THE MAIN PROGRAM WHICH IS THIS FILE. *
c*-----*

  subroutine setpr(compress)

c
c-----set printer to compress mode
  character*1 c1,c2,compress*3
  integer ia(2)
  equivalence (c1,ia(1)),(c2,ia(2))
  data ia/16#0F,16#12/
  if (compress .eq. 'on ') then
    write(3,10) c1
  else if (compress .eq. 'off') then
    write(3,10) c2
  endif
  10 format(1x,a1)
  return
  end

c

```



```

c*-----*
c*          S U B R O U T I N E          C L R S C R N          *
C*  THIS SUBROUTINE CLEARS THE SCREEN AND IS CALLED FROM MANY  *
C*  SUBROUTINES.                                               *
c*-----*
C
      subroutine clrscrn
c
c-----clear display and move to home position
      character*1 c1,c2,c3,c4
      integer ic(4)
      equivalence (c1,ic(1)),(c2,ic(2)),(c3,ic(3)),(c4,ic(4))
      data ic/16#1B,16#5B,16#32,16#4A/
c
      write(*,1) c1,c2,c3,c4
1 format(lx,4a1)
      return
      end
c
c*-----*
c*          S U B R O U T I N E          C U R S O R          *
C*  THIS SUBROUTINE RELOCATE THE CURSOR AND IS CALLED FROM FILE *
C*  STIFF3V2.FOR SUBROUTINE INTGRL.                             *
c*-----*

      subroutine cursor(i,j)
c
c-----move cursor to specific location
      character*1 c1,c2,c5,c8,lc(5)
      character*8 cbuff
      integer ic(4)
      equivalence (c1,ic(1)),(c2,ic(2)),(c5,ic(3)),(c8,ic(4)),
*          (cbuff,lc(1))
      data ic/16#1B,16#5B,16#3B,16#66/
c
      l = 10000+100*i+j
      write(cbuff,2) l
2 format(i5)
      write(*,1) c1,c2,lc(2),lc(3),c5,lc(4),lc(5),c8
1 format(lx,8a1,\)
      return
      end
c
c*-----*
c*  THIS IS A BLOCK DATA USED AS A HEADER ON THE OUTPUT LISTING. *
c*-----*
      block data lstid
      character*20 infile, outfile, pghd*70, date*8
      common /blk7/ infile, outfile, pghd, date
      data pghd/' PLANT TRANSPORT THREE LEAF MODEL (VER 2.1). ALL RIGH
*TS RESERVED. '/

```

end

```

c
c      T H I S      I S      F I L E      D T A D R 3 L V . F O R
c
c*-----*
c*
c* P L A N T      T R A N S P O R T      T H R E E      L E A F      *
c*                M O D E L      *
c*                (version 2.1)  *
c*
c* Program manager : Dr. Craig McFarlane *
c*
c* Programmer : Kai P. Suen           6/1/1987 *
c*                Oregon State University *
c*                Soil Science Department *
c*                Corvallis, Oregon 97331 *
c*
c* Program developed for EPA grant no. CR811940-01-03 *
c*
c*                All rights reserved *
c*-----*
c
c
c*-----*
c*                S U B R O U T I N E      D T A D R V R      *
c* THE PURPOSE OF THIS SUBROUTINE IS TO PERFORM DATA INPUT AND *
c* DATA ORGANIZATION FOR THE PROGRAM SIMULATOR. *
c* THIS SUBROUTINE IS CALLED FROM FILE 'PLANT3V2.FOR' MAIN *
c* PROGRAM. *
c*
c* VARIABLE DEFINITION *
c*-----*
c*      TINTIAL      :      INITIAL TIME TO START SIMULATION *
c*      TSTOP       :      TERMINATION TIME WHEN SIMULATION COMPLETED *
c*      DT          :      TIME STEP INCREMENTS *
c*      PRINTVL    :      PRINT RESULTS FOR EVERY GIVEN INTERVALS *
c*      NCPMT      :      NUMBER OF COMPARTMENTS. IT IS A CONSTANT *
c*                   VARIABLE DECLARED IN FILE "COMBLOC" *
c*
c* FUNCTION CALLS *
c*-----*
c*      CALL CLRSCRN : CLEAR CURRENT SCREEN. ACTUAL CODES ARE *
c*                   STORED IN FILE "PLANT3V2.FOR" *
c*      CALL SCRNH D : PRINT SCREEN HEADER. ACTUAL CODES ARE *
c*                   STORED IN FILE "PLANT3V2.FOR" *
c*      CALL MSSGE2  : PROMPT PROGRAM PROCESSING MESSAGE TO SCREEN. *
c*      CALL LOADDTA : LOAD INPUT DATA. ACTUAL CODES ARE STORED *
c*                   IN FILE "DTADR3v2.FOR" *
c*      CALL MTXA    : COMPUTE MATRIX A(N,N). ACTUAL CODES ARE *
c*                   STORED IN FILE "MTXA3V2.FOR". *
c*      CALL INTVLE  : LOAD INITIAL MASS VALUES. ACTUAL CODES ARE *

```

```

C*                               STORED IN FILE "DTADR3V2.FOR"                               *
C*-----*
C
      subroutine dtadrvr
$include:'combloc'
C
C-----OUTPUT FORMATS
C
1025  format(f12.4)
2000  format(//,16x,'Enter time (in hours) to start simulation >- ',\ )
2010  format(16x,'Enter time (in hours) to terminate simulation >- ',\ )
2020  format(16x,'Enter time interval (step size) to integrate >- ',\ )
2025  format(16x,'Enter tolerance (default is ',G12.6,') >- ',\ )
2030  format(16x,'Enter time interval for output listing:',/,
*      16x,' (e.g. 1.0 for listing every 1 hour.',/,
*      16x,' 1.5 for listing every 1" hours.',/,
*      16x,' 3.0 for listing every 3 hours.) >- ',\ )
2040  format(i2)
2050  format(5e16.9)
2060  format(a,/, ' This FILENAME is --> ',a,55x,'Date : ',A)
2070  format(1x,'Start time: ', f6.3,', Stop time: ', f6.3,
& ', Tolerance: ', f10.7, ', Step size: ', f8.5,
& ', Print interval: ', f8.5, /)
C
C
C-----ENTER INITIAL SIMULATION TIME
C
      write(*,2000)
      read(*,*) tintial
C
C-----ENTER TERMINATION TIME
C
      write(*,2010)
      read(*,*) tstop
C
C-----ENTER TIME STEP INCREMENT
C
      write(*,2020)
      read(*,*) dt
C
C-----ENTER TIME INTERVAL FOR OUTPUT
C  IT IS VERY DIFFICULT FOR FORTRAN TO COMPARE TWO
C  FLOATING POINT NUMBERS WITH AN EQUAL SIGN (.EQ.).
C  THEREFORE PRINT INTERVAL VARIABLE (PRINTVL) IS SUBTRACTED
C  BY 0.0001 AND (.GT.) SIGN IS USED TO CHECK FOR PRINTING
C  INTERVALS.
C
      write(*,2030)
      read(*,*) printvl
      printvl = printvl - 0.0001
C

```

```
C-----USER MAY SET THE DESIRED TOLERANCE FOR THE INTEGRATION
C
```

```
    tol = 1.d-4
    write(*,2025) tol
    read (*,1025) tol
    if ( tol .le. 0.d0 ) tol = 1.d-4
```

```
C
C-----LIST INPUT INFORMATION TO FILE 'DATA.OUT'
C
```

```
    write(2,2040) ncpmt
    write(2,2050) tintial,tstop,dt,printvl+0.0001,tol
```

```
C
C-----CLEAR SCREEN ,PRINT SCREEN HEADER AND REPORT
C
C
C
```

```
    call clrscrn
    call scrnhd
    call mssge2
```

```
C
C-----LOAD INPUT DATA, COMPUTE MATRIX A, AND LOAD INITIAL
C
C
C
```

```
    write(3,2070) tintial, tstop, tol, dt, printvl
    write(3,2060) pghd,outfile,date
    call loaddta
    call mtxa
    call intvle
    return
    end
```

```
c*-----*
c*          S U B R O U T I N E          F I N D Q          *
c*
c* THIS SUBROUTINE IS DESIGNED TO COMPUTE THE Q COEFFICIENTS; *
c* VELOCITIES.  INFORMATIONS NEEDED ARE PASSED IN AS GLOBAL AND *
c* RESULT IS ALSO PASS OUT AS GLOBAL.  THE INDEXES USED ARE *
c* REPRESENT CELL WALLS, AND IT IS PREVIOUSLY DEFINED IN THE *
c* PLANT FIGURE. *
c* THIS SUBROUTINE IS CALLED FROM FILE 'DTADR3v2.FOR' SUBROUTINE *
c* LOADDTA. *
c*-----*
```

```
c
    subroutine findq
    $include:'combloc'
    c
        do 10 i=1,2
            q(i) = (tr(1)+tr(2)+tr(3))/area(i)
        10 continue
    c
        do 20 i=3,4
            q(i) = ((1.0+frac(1))*tr(1)+(1.0+frac(2))*tr(2)+
*              (1.0+frac(3))*tr(3))/area(i)
```

```

20 continue
c
  q(5) = ((1.0+frac(2))*tr(2)+(1.0+frac(3))*tr(3))/area(5)
c
  do 30 i=6,8
    q(i) = ((1.0+frac(3))*tr(3))/area(i)
30 continue
c
  q(9) = 0.0
c
  do 40 i=10,12
    q(i) = (frac(3)*tr(3))/area(i)
40 continue
c
  q(13) = (frac(2)*tr(2)+frac(3)*tr(3))/area(13)
  q(14) = (frac(1)*tr(1)+frac(2)*tr(2)+frac(3)*tr(3))/area(14)
  q(15) = 0.0
c
  do 50 i=16,19
    q(i) = ((1.0+frac(1))*tr(1))/area(i)
50 continue
c
  q(20) = 0.0
c
  do 60 i=21,23
    q(i) = (frac(1)*tr(1))/area(i)
60 continue
c
  do 70 i=24,26
    q(i) = ((1.0+frac(2))*tr(2))/area(i)
70 continue
c
  q(27) = 0.0
c
  do 80 i=28,30
    q(i) = (frac(2)*tr(2))/area(i)
80 continue
c
  q(31) = (frac(1)*tr(1))/area(31)
  q(32) = (frac(1)*tr(1)+frac(2)*tr(2)+frac(3)*tr(3))/area(32)
c
  return
  end
c*-----*
c*          S U B R O U T I N E          L O A D D T A          *
c*          THIS SUBROUTINE IS TO LOAD INPUT IN APPROPRIATE STORAGE *
c*          VARIABLES. *
c*          THIS SUBROUTINE IS CALLED FROM FILE 'DTADR3V2.FOR' *
c*          SUBROUTINE DTADRVR. *
c*-----*
c

```

```

      subroutine loadtdta
$include:'combloc'
C
C-----INPUTS ARE FREE FORMAT
c
c-----OUTPUT FORMAT
  2000 format('1',/,a/, ' This file is named --> ',a,
*          55x, 'DATE : ',a,/)
C
C-----LOAD DATA AND ECHO TO USER GIVEN OUTPUT FILENAME.
  read(1,*,end=999)
  read(1,*) (area(i),i=1,32)
  call lstdta(area,32, 'Area      ')
  read(1,*)
  read(1,*) (deltax(i),i=1,32)
  call lstdta(deltax,32,'Delta x  ')
  read(1,*)
  read(1,*) (sigma(i),i=1,32)
  call lstdta(sigma,32, 'Sigma    ')
  read(1,*)
  read(1,*) (difu(i),i=1,32)
  call lstdta(difu,32, 'Diffu   ')
  read(1,*)
  read(1,*) (kr(i),i=1,32)
  call lstdta(kr,32,   'Kr      ')
  read(1,*)
  read(1,*) (kl(i),i=1,32)
  call lstdta(kl,32,   'Kl      ')
c
C-----SKIP TO A NEW PAGE AND PRINT PAGE HEADER.
  write(3,2000) pghd,outfile,date
c
  read(1,*)
  read(1,*) conc0
  call lstdta(conc0,1, 'Conc0   ')
  read(1,*)
  read(1,*) (vol(i),i=1,29)
  call lstdta(vol,29, 'Vol     ')
  read(1,*)
  read(1,*) (sb(i),i=1,29)
  call lstdta(sb,29,  'B       ')
  read(1,*)
  read(1,*) (lambda(i),i=1,29)
  call lstdta(lambda,29,'Lambda  ')
  read(1,*)
  read(1,*) js
  call lstdta(js,1,   'Js      ')
  read(1,*)
c
c-----FRACTIONS OF THE FLOW RATES ARE READ FROM LEAF1 THROUGH
C      LEAF3 IN ORDERS.

```

```

        read(1,*) (frac(i),i=1,3)
        call lstdta(frac,3, 'Frac  ')
        read(1,*)
c
c-----THE TRANSPORTATIONS RATE ARE READ FROM LEAF1 THROUGH LEAF3
c      IN ORDERS.
        read(1,*) (tr(i),i=1,3)
        call lstdta(tr,3, 'Transp  ')
c
c-----THE FLOW RATE 'q' WILL BE COMPUTED BY SUBROUTINE FINDQ
        call findq
        call lstdta(q,32, 'q  ')
999  return
      end
c*-----*
c*          S U B R O U T I N E          I N T V L E          *
c*      THIS SUBROUTINE IS TO INPUT INITIAL MASS VALUES AND IS      *
c*      CALLED FROM FILE 'DTADR3V2.FOR' SUBROUTINE DTADRVR.          *
c*-----*
c
      subroutine intvle
$include:'combloc'
c
c-----OUTPUT FORMAT
      2000 format(29e16.9)
c
c-----INPUT AND OUTPUT INITIAL PROCESSING TIME
      do 10 i=1,ncpmt
          xnow(i) = 0.0
10  continue
      read(1,*) (xnow(i),i=1,ncpmt)
      write(2,2000) (xnow(i),i=1,ncpmt)
      call lstdta(xnow,ncpmt,'initial ')
      return
      end
c*-----*
c*          S U B R O U T I N E          L S T M T X          *
c*      THIS SUBROUTINE IS TO PRINT MATRIX A TO FILE "DATA.OUT"      *
c*      DATA.OUT CAN BE USED AS PROGRAM VERIFICATION AND THE MATRIX  *
c*      CAN BE USED TO COMPUTE STEADY STATE.                          *
c*      THIS SUBROUTINE IS CALLED FROM FILE 'MTXA3V2.FOR' SUBROUTINE *
c*      MTXA.                                                          *
c*-----*
c
      subroutine lstmtx(omtxa)
$large:omtxa
      real*8 omtxa(29,29)
c
c-----PARAMETER DEFINITION
c      1. omtxa = two dimensional matrix.
c

```



```

c-----OUTPUT FORMAT
2000 format(33e16.9)
c
c-----LIST THE COMPUTED M BY M MATRIX A
do 10 i=1,29
    write(2,2000) (omtxa(i,j),j=1,29)
10 continue
return
end

c*-----*
c*          S U B R O U T I N E          L S T D T A          *
c*    THIS SUBROUTINE IS TO PRINT ALL INPUT DATA TO AN OUTPUT *
c*    FILE.  THE OUTPUT FILENAME IS USER SPECIFIED.          *
c*    THIS SUBROUTINE IS CALLED FROM FILE 'DTADR3V2.FOR'      *
c*    SUBROUTINE LOADDTA.                                     *
c*-----*
c
c          subroutine lstdta(odata,msize,ovarble)
c          real*8 odata(msize)
c          character*9 ovarble

c
c-----parameter definition
c          1. odata = one dimensional array of input variables.
c          2. msize = specified array size.
c          3. ovarble = output variable name.
c
2010 format(9(i8,4x))
2020 format(9e12.4)
2030 format(5e12.4,49x,'= ',a,/)
2040 format(2e12.4,85x,'= ',a,/)
2050 format(e12.4,97x,'= ',a,/)
2060 format(3e12.4,73x,'= ',a,/)
c
c-----ECHO INPUT DATA TO USER SPECIFIED FILENAME.
if (msize .gt. 3) then
    write(3,2010) (i,i=1,9)
    write(3,2020) (odata(i),i=1,9)
    write(3,2010) (i,i=10,18)
    write(3,2020) (odata(i),i=10,18)
    write(3,2010) (i,i=19,27)
    write(3,2020) (odata(i),i=19,27)
    write(3,2010) (i,i=28,msize)
    if (msize .eq. 32) then
        write(3,2030) (odata(i),i=28,msize),ovarble
    else if (msize .eq. 29) then
        write(3,2040) (odata(i),i=28,msize),ovarble
    end if
else if (msize .eq. 1) then
    write(3,2050) odata(msize),ovarble
else if (msize .eq. 3) then
    write(3,2010) (i,i=1,3)

```

```

        write(3,2060) (odata(i),i=1,3),ovarble
    end if
    return
end
c*-----*
c*           S U B R O U T I N E           M S S G E 2           *
c*   THIS SUBROUTINE IS TO PROMPT SCREEN MESSAGE TO USER AND IS *
c*   CALLED FROM FILE 'DTADR3V2.FOR' SUBROUTINE DTADRVR.       *
c*-----*
c
    subroutine mssge2
    write(*,2000)
2000 format(5(/),
    * 16x,'The input data is now being printed to the user',/,
    * 16x,'designated output device.',//,
    * 16x,'Please check the input data.',//,
    * 16x,'The program is now constructing the mass balance',/,
    * 16x,'equations.',///,
    * 16x,'Please wait a moment !')
    return
end

```

```

c
c   T H I S   I S   F I L E   M T X A 3 L V . F O R
c
c*-----*
c*
c* P L A N T   T R A N S P O R T   T H R E E   L E A F   *
c*           M O D E L   *
c*           (version 2.1) *
c* *
c* Program manager : Dr. Craig McFarlane *
c* *
c* Programmer : Kai P. Suen           06/01/87. *
c*           Oregon State University *
c*           Soil Science Department *
c*           Corvallis, Oregon 97331 *
c* *
c* Program developed for EPA grant no. CR811940-01-03 *
c* *
c*           All rights reserved *
c*-----*
c
c
c*-----*
c*           S U B R O U T I N E           M T X A           *
c* THIS SUBROUTINE IS TO COMPUTE THE COEFFICIENTS FOR ALL OF THE *
c* COMPARTMENT INFLOWS AND OUTFLOWS. THE VALUE COMPUTED IS *
c* STORED IN MATRIX A AND MATRIX A IS A GLOBAL TWO DIMENSIONAL *
c* ARRAY. THE SYSTEM EQUATIONS USED FOR THE THREE LEAF MODEL HAS *
c* SIMILAR TERMS BUT DIFFERENT INDEXES FOR DIFFERENT COMPART- *
c* MENTS, THEREFORE FUNCTIONS (FN1, FN2, FN3, AND FN4) ARE BEING *
c* USED TO SUBSTITUTE CODING REDUNDANCY. PLEASE REFER TO PAPER *
c* RECIEVED FOR THE SYSTEM EQUATIONS. *
c* THIS SUBROUTINE IS CALLED FROM FILE 'DTADR3V2.FOR' SUBROUTINE *
c* DTADRVR. *
c* *
c* SUBROUTINE CALLS *
c* ----- *
c* CALL LSTMTX() = LIST MATRIX A TO OUTPUT FILE 'DATA.OUT'. *
c*           ACTUAL CODE IS STORED IN FILE 'DTADR3V2.FOR' *
c* *
c* FUNCTION CALLS *
c* ----- *
c* FN1 = ACTUAL CODE IS STORED IN FILE 'MTXA3V2.FOR' *
c* FN2 = ACTUAL CODE IS STORED IN FILE 'MTXA3V2.FOR' *
c* FN3 = ACTUAL CODE IS STORED IN FILE 'MTXA3V2.FOR' *
c* FN4 = ACTUAL CODE IS STORED IN FILE 'MTXA3V2.FOR' *
c*-----*
c
subroutine mtxa
$include:'combloc'
real*8 fn1,fn2,fn3,fn4

```

```

C
C-----INITIALIZE MATRIX A TO ZERO; matrix a(ncpmt,ncpmt) is global
do 10 i=1,ncpmt
  do 10 j=1,ncpmt
    a(i,j) = 0.0
  10 continue
C
C-----START COMPUTE VALUES FOR DISTINCT LOCATION
a(1,1) = -(fn1(1)+fn2(2)+fn3(1))/(vol(1)*(1.0+sb(1)))
a(1,2) = fn1(2)/(vol(2)*(1.0+sb(2)))
a(2,1) = fn2(2)/(vol(1)*(1.0+sb(1)))
a(2,2) = -(fn1(2)+fn1(32)+fn2(3)+fn3(2))/(vol(2)*(1.0+sb(2)))
a(2,3) = fn1(3)/(vol(3)*(1.0+sb(3)))
a(2,14) = fn2(32)/(vol(14)*(1.0+sb(14)))
a(3,2) = fn2(3)/(vol(2)*(1.0+sb(2)))
a(3,3) = -(fn1(3)+fn4(4)+fn3(3))/(vol(3)*(1.0+sb(3)))
a(4,3) = fn4(4)/(vol(3)*(1.0+sb(3)))
a(4,4) = -(fn4(5)+fn4(16)+fn3(4))/(vol(4)*(1.0+sb(4)))
a(5,4) = fn4(5)/(vol(4)*(1.0+sb(4)))
a(5,5) = -(fn4(24)+fn4(6)+fn3(5))/(vol(5)*(1.0+sb(5)))
a(6,5) = fn4(6)/(vol(5)*(1.0+sb(5)))
a(6,6) = -(fn4(7)+fn3(6))/(vol(6)*(1.0+sb(6)))
a(7,6) = fn4(7)/(vol(6)*(1.0+sb(6)))
a(7,7) = -(fn2(8)+fn3(7))/(vol(7)*(1.0+sb(7)))
a(7,8) = fn1(8)/(vol(8)*(1.0+sb(8)))
a(8,7) = fn2(8)/(vol(7)*(1.0+sb(7)))
a(8,8) = -(fn1(8)+fn2(9)+fn2(10)+fn3(8))/(vol(8)*(1.0+sb(8)))
a(8,9) = fn1(9)/(vol(9)*(1.0+sb(9)))
a(8,10) = fn1(10)/(vol(10)*(1.0+sb(10)))
a(9,8) = fn2(9)/(vol(8)*(1.0+sb(8)))
a(9,9) = -(fn1(9)+fn3(9))/(vol(9)*(1.0+sb(9)))
a(10,8) = fn2(10)/(vol(8)*(1.0+sb(8)))
a(10,10) = -(fn1(10)+fn4(11)+fn3(10))/(vol(10)*(1.0+sb(10)))
a(11,10) = fn4(11)/(vol(10)*(1.0+sb(10)))
a(11,11) = -(fn4(12)+fn3(11))/(vol(11)*(1.0+sb(11)))
a(12,11) = fn4(12)/(vol(11)*(1.0+sb(11)))
a(12,12) = -(fn4(13)+fn3(12))/(vol(12)*(1.0+sb(12)))
a(12,29) = fn4(30)/(vol(29)*(1.0+sb(29)))
a(13,12) = fn4(13)/(vol(12)*(1.0+sb(12)))
a(13,13) = -(fn4(14)+fn3(13))/(vol(13)*(1.0+sb(13)))
a(13,23) = fn4(31)/(vol(23)*(1.0+sb(23)))
a(14,13) = fn4(14)/(vol(13)*(1.0+sb(13)))
a(14,2) = fn1(32)/(vol(2)*(1.0+sb(2)))
a(14,14) = -(fn2(32)+fn2(15)+fn3(14))/(vol(14)*(1.0+sb(14)))
a(14,15) = fn1(15)/(vol(15)*(1.0+sb(15)))
a(15,14) = fn2(15)/(vol(14)*(1.0+sb(14)))
a(15,15) = -(fn1(15)+fn3(15))/(vol(15)*(1.0+sb(15)))
a(16,4) = fn4(16)/(vol(4)*(1.0+sb(4)))
a(16,16) = -(fn4(17)+fn3(16))/(vol(16)*(1.0+sb(16)))
a(17,16) = fn4(17)/(vol(16)*(1.0+sb(16)))
a(17,17) = -(fn4(18)+fn3(17))/(vol(17)*(1.0+sb(17)))

```

```

a(18,17) = fn4(18)/(vol(17)*(1.0+sb(17)))
a(18,18) = -(fn2(19)+fn3(18))/(vol(18)*(1.0+sb(18)))
a(18,19) = fn1(19)/(vol(19)*(1.0+sb(19)))
a(19,18) = fn2(19)/(vol(18)*(1.0+sb(18)))
a(19,19) = -(fn1(19)+fn2(20)+fn2(21)+fn3(19))/
*      (vol(19)*(1.0+sb(19)))
a(19,20) = fn1(20)/(vol(20)*(1.0+sb(20)))
a(19,21) = fn1(21)/(vol(21)*(1.0+sb(21)))
a(20,19) = fn2(20)/(vol(19)*(1.0+sb(19)))
a(20,20) = -(fn1(20)+fn3(20))/(vol(20)*(1.0+sb(20)))
a(21,19) = fn2(21)/(vol(19)*(1.0+sb(19)))
a(21,21) = -(fn1(21)+fn4(22)+fn3(21))/(vol(21)*(1.0+sb(21)))
a(22,21) = fn4(22)/(vol(21)*(1.0+sb(21)))
a(22,22) = -(fn4(23)+fn3(22))/(vol(22)*(1.0+sb(22)))
a(23,22) = fn4(23)/(vol(22)*(1.0+sb(22)))
a(23,23) = -(fn4(31)+fn3(23))/(vol(23)*(1.0+sb(23)))
a(24,5) = fn4(24)/(vol(5)*(1.0+sb(5)))
a(24,24) = -(fn4(25)+fn3(24))/(vol(24)*(1.0+sb(24)))
a(25,24) = fn4(25)/(vol(24)*(1.0+sb(24)))
a(25,25) = -(fn2(26)+fn3(25))/(vol(25)*(1.0+sb(25)))
a(25,26) = fn1(26)/(vol(26)*(1.0+sb(26)))
a(26,25) = fn2(26)/(vol(25)*(1.0+sb(25)))
a(26,26) = -(fn1(26)+fn2(27)+fn2(28)+fn3(26))/
*      (vol(26)*(1.0+sb(26)))
a(26,27) = fn1(27)/(vol(27)*(1.0+sb(27)))
a(26,28) = fn1(28)/(vol(28)*(1.0+sb(28)))
a(27,26) = fn2(27)/(vol(26)*(1.0+sb(26)))
a(27,27) = -(fn1(27)+fn3(27))/(vol(27)*(1.0+sb(27)))
a(28,26) = fn2(28)/(vol(26)*(1.0+sb(26)))
a(28,28) = -(fn1(28)+fn4(29)+fn3(28))/(vol(28)*(1.0+sb(28)))
a(29,28) = fn4(29)/(vol(28)*(1.0+sb(28)))
a(29,29) = -(fn4(30)+fn3(29))/(vol(29)*(1.0+sb(29)))

```

```

C
C-----OUTPUT COMPUTED MATRIX
call lstmtx(a)
return
end

```

```

C
c*-----*
C*          F U N C T I O N      F N 1      *
C* THIS FUNCTION IS CALLED FROM FILE 'MTXA3V2.FOR' SUBROUTINE *
C* MTXA. *
c*-----*

```

```

real*8 function fn1(i)
$include:'combloc'
fn1 = ( kr(i)*difu(i)*area(i) ) / deltax(i)
return
end

```

```

c*-----*
C*          F U N C T I O N      F N 2      *

```

```

C*   THIS FUNCTION IS CALLED FROM FILE 'MTXA3V2.FOR' SUBROUTINE   *
C*   MTXA AND FILE 'STIFF3V2.FOR' FUNCTION FNC.                 *
c*-----*

```

```

      real*8 function fn2(i)
$include:'combloc'
      fn2 = ( ( (kl(i)*difu(i)) / deltax(i) ) + ( kl(i)*q(i)*
1          (1.0-sigma(i)) ) ) * area(i)
      return
      end

```

```

c*-----*
C*           F U N C T I O N           F N 3                       *
C*   THIS FUNCTION IS CALLED FROM FILE 'MTXA3V2.FOR' SUBROUTINE   *
C*   MTXA.                                                                *
c*-----*

```

```

      real*8 function fn3(i)
$include:'combloc'
      fn3 = lambda(i) * vol(i)
      return
      end

```

```

C
c*-----*
C*           F U N C T I O N           F N 4                       *
C*   THIS FUNCTION IS CALLED FROM FILE 'MTXA3V2.FOR' SUBROUTINE   *
C*   MTXA.                                                                *
c*-----*
C

```

```

      real*8 function fn4(i)
$include:'combloc'
      fn4 = q(i) * area(i)
      return
      end

```

*STIFF

Last revision: December 4, 1987

c
c THIS IS FILE STIFF3LV.FOR
c

c*-----*
c*
c* PLANT TRANSPORT THREE LEAF *
c* MODEL *
c* (version 2.1) *
c*
c* Program manager : Dr. Craig McFarlane *
c*
c* Programmer : Kai P. Suen 06/01/87. *
c* Oregon State University *
c* Soil Science Department *
c* Corvallis, Oregon 97331 *
c*
c* Program developed for EPA grant no. CR811940-01-03 *
c*
c* All rights reserved *
c*-----*

c
c
c*-----*
c* SUBROUTINE STIFF *
c* THIS SUBROUTINE PASS CONTROL TO SUBROUTINE INTGRL TO PROCESS *
c* SIMULATION. IT DISPLAY AN OUTPUT OPTION SCREEN MENU FOR USER*
c* TO SELECT THEIR OUTPUT LISTING AFTER SIMULATION HAS COMPLETED *
c* THIS SUBROUTINE IS CALLED FROM FILE 'PLANT3V2.FOR' MAIN *
c* PROGRAM. *
c*
c* LOCAL VARIABLE DEFINITION *
c*-----*
c* IPICK = SELECTED OUTPUT OPTION FROM SCREEN MENU (INTEGER). *
c*
c* SUBROUTINE CALLS *
c*-----*
c* CALL CLRSCRN : CLEAR CONSOLE. ACTUAL CODE IS STORED IN *
c* FILE "PLANT3V2.FOR" *
c* CALL SCRNH D : PRINT SCREEN HEADER. ACTUAL CODE IS STORED *
c* IN FILE "PLANT3V2.FOR" *
c* CALL MSSGE3 : PRINT PROCESSING MESSAGE TO SCREEN AND *
c* PERMITS USER TO EXIT PROGRAM BEFORE *
c* STARTING SIMULATION. ACTUAL CODE IS STORED *
c* IN FILE "STIFF3V2.FOR" *
c* CALL INTGRL : SIMULATION ROUTINE. ACTUAL CODE IS STORED *
c* IN FILE "STIFF3V2.FOR" *
c* CALL MENU : OUTPUT OPTIONS PRESENTED IN SCREEN MENU *
c* FORMAT. ACTUAL CODE IS STORED IN FILE *
c* "STIFF3V2.FOR" *

```

C*   CALL OUTMASS   : OUTPUT INDIVIDUAL MASS IN COMPARTMENTS OVER *
C*                   TIME.  ACTUAL CODE IS STORED IN FILE          *
C*                   "PRINT3V2.FOR"                               *
C*   CALL OUTMLOS   : OUTPUT INDIVIDUAL MASS LOST IN COMPARTMENTS *
C*                   OVER TIME.  ACTUAL CODE IS STORED IN FILE    *
C*                   "PRINT3V2.FOR"                               *
C*   CALL OUTGROUP  : OUTPUT GROUPED MASS SUCH AS ROOTS, STEMS,   *
C*                   AND LEAVES (1, 2, AND 3).  MASS LOSS (LAMBDA)*
C*                   OVER TIME ARE NOT INCLUDED IN THE GROUPED   *
C*                   COMPARTMENTS.  ACTUAL CODE IS STORED IN FILE *
C*                   "PRINT3V2.FOR"                               *
C*   CALL OUTRSLWL  : OUTPUT GROUPED MASS SUCH AS ROOTS, STEMS,   *
C*                   AND LEAVES (1, 2, AND 3).  MASS LOSS (LAMBDA)*
C*                   OVER TIME ARE INCLUDED IN THE GROUPED       *
C*                   COMPARTMENTS.  ACTUAL CODE IS STORED IN FILE *
C*                   "PRINT3V2.FOR"                               *
c*-----*
c
      subroutine stiff
$include:'combloc'
c
      call clrscrn
C-----CLEAR CONSOLE AND PRINT SCREEN HEADER.
      call scrnhd
c
C-----LIST PROCESSING MESSAGES ONTO CONSOLE
      call mssge3
c
C-----SIMULATE SYSTEM OPERATION OVER TIME.
      call intgr1
c
10    call clrscrn
      call scrnhd
      call menu(ipick)
c
c-----EXIT COMMAND
      if (ipick .eq. 6) goto 999
c
C-----OUTPUT INDIVIDUAL MASS IN COMPARTMENTS
      if ((ipick .eq. 1).or.(ipick .eq. 5)) call outmass
c
C-----OUTPUT INDIVIDUAL MASS LOSS IN COMPARTMENTS
      if ((ipick .eq. 2).or.(ipick .eq. 5)) call outmlos
c
c-----OUTPUT GROUPED MASS WITH ROOT, STEM, AND LEAVES. NO LOSS
C      (LAMBDA) ADDED TO EACH GROUP.
c
      if ((ipick .eq. 3).or.(ipick .eq. 5)) call outgroup
c
c-----OUTPUT GROUPED MASS (ROOT, STEM AND LEAVES) WITH LOSS (LAMBDA)
C      ADDED TO EACH GROUP

```



```

c
      if ((ipick .eq. 4).or.(ipick .eq. 5)) call outrslwl
c
C-----END OF RUN.
c
C-----RETURN TO SCREEN MENU UNTIL USER PICKED OPTION #6 TO EXIT.
      goto 10
999  return
      end

c
c*-----*
c*          S U B R O U T I N E      I N T G R L          *
c*  THIS SUBROUTINE IS THE HEART OF THE SIMULATION MODEL.      *
c*  IT CONTAINS THE FIRST ORDER DIFFERENTIAL EQUATION SOLVER FROM *
c*  THE IMSL MATH LIBRARY (DGEAR METHOD). PLEASE REFER TO IMSL *
c*  LIBRARY FOR DETAILS ON PARAMETER DEFINITIONS.              *
c*  THIS SUBROUTINE IS CALLED FROM FILE 'STIFF3V2.FOR' SUBROUTINE *
c*  STIFF.                                                       *
c*                                                                *
c*  LOCAL VARIABLE DEFINITIONS                                  *
c*  -----*
c*  WK(1452) = REFER TO IMSL MATH LIBRARY FOR DETAILS.          *
c*  IWK(33)  = REFER TO IMSL MATH LIBRARY FOR DETAILS.          *
c*  INFLOW   = A FUNCTION USED TO COMPUTE MASS INFLOW           *
c*            FROM SOURCE TO PLANT AT INSTANT TIME. (REAL)     *
c*  TLAMBDA  = A FUNCTION USED TO COMPUTE MASS LOST FROM        *
c*            PLANT AT INSTANT TIME. (REAL)                     *
c*  TOL      = REFER TO IMSL MATH LIBRARY FOR DETAILS.          *
c*  H        = REFER TO IMSL MATH LIBRARY FOR DETAILS.          *
c*  TEND     = REFER TO IMSL MATH LIBRARY FOR DETAILS.          *
c*  IER      = REFER TO IMSL MATH LIBRARY FOR DETAILS.          *
c*  METH     = REFER TO IMSL MATH LIBRARY FOR DETAILS.          *
c*  MITER    = REFER TO IMSL MATH LIBRARY FOR DETAILS.          *
c*  INDEX    = REFER TO IMSL MATH LIBRARY FOR DETAILS.          *
c*  IT       = INTEGER VALUE CONVERTED FROM DOUBLE PRECISION   *
c*            GLOBAL VARIABLE TINTIAL.                          *
c*  IIT      = SAME AS ABOVE. USED TO CHECK SIMULATING TIME    *
c*            DISPLAYED TO SCREEN.                              *
c*  FCN,FCNJ = REFER TO IMSL MATH LIBRARY FOR DETAILS.          *
c*  COMPMT0  = ACCUMULATED MASS INFLOW FROM SOURCE TO PLANT.   *
c*            (REAL)                                             *
c*  TLOST    = ACCUMULATED MASS LOST FROM PLANT. (REAL)        *
c*-----*
c
      subroutine intgrl
      $include:'combloc'
      double precision  prmt, x2, xmax, save, error
      real              pw
      dimension x2(8,ncpmt), xmax(ncpmt), save(12*ncpmt,ncpmt),
& error(ncpmt), pw(ncpmt,ncpmt), prmt(5+3)
      common /f1/ lunpr, inext, tnext

```

```

      real*8 inflow, tlambda, tol, h, tend
      external fcn, fcnj, foutp
C
C-----INITIALIZE VARIABLES
      h      = 1.0D-5
      itime = 1
      tprt  = 0.0
      tnxt  = tintial
      npage = 1
      prmt(1) = tintial
      prmt(2) = tstop
      prmt(3) = h
      prmt(4) = 1.0D-8
      prmt(5) = tol
      prmt(6) = dt
      prmt(7) = 2.
      prmt(8) = 5.
c
C-----START SIMULATING UNTIL SPECIFIED TERMINATING TIME
      do 45 j=1,ncpmt
          xold(j) = xnow(j)
          x2(1,j) = xnow(j)
          xmax(j) = 1.d0
          error(j) = 0.d0
          do 45, i = 2, ncpmt
              x2(i,j) = 0.d0
          45 continue
c
c-----SUBROUTINE DECINT USES GEAR'S DIFSUB, STIFF METHOD
      kflag = 0
      nc    = ncpmt
      call decint(prmt, x2, xmax, nc, kflag, fcn, fcnj, foutp,
&               error, save, pw)
C
C-----PRINT CURRENT VALUES WHEN ERROR OCCURRED AND STOP PROGRAM
      if ( kflag .lt. 0 ) then
          write(3,60) tol,ncpmt,tend,dt,tintial
60      format(' TOL   = ',E13.6,/,
*           ' n     = ',I2,/,
*           ' TEND  = ',E13.6,/,
*           ' H     = ',E13.6,/,
*           ' T     = ',E13.6,/,
*           ' stopped time and current xnow are: '/')
          write(3,70) tintial,(xnow(i),i=1,ncpmt)
70      format(34e13.6)
          stop
      end if
c
      itime = itime - 1
      return
      end

```

```

c
c*-----*
c*           S U B R O U T I N E       F C N           *
c* THIS SUBROUTINE IS TO COMPUTE THE FIRST ORDER DERIVATIVES *
c* RESPECT TO XNOW AND IS CALLED FROM FILE 'STIFF3V2.FOR' *
c* SUBROUTINE INTGRL. *
c*-----*

```

```

c
  subroutine fcn(ncpmt,tintial,x2,yprime)
  implicit double precision(a-h,o-z)
  common /blk1/area(32),difu(32),kr(32),kl(32),q(32),sigma(32)
  real*8 kr,kl
  common /blk2/deltax(32),lambda(29),vol(29),sb(29)
  real*8 lambda
  common /blk6/ js,conc0,a(29,29),lost(29)
  real*8 js, lost, x2(8,29), yprime(29), fn2

```

```

c
  s1 = fn2(1)*conc0+js*area(1)
  do 10 i=1,ncpmt
    yprime(i) = a(i,1)*x2(1,1) + s1
    s1 = 0.0
    do 10 j=2,ncpmt
      yprime(i) = yprime(i) + a(i,j)*x2(1,j)
10 continue
  return
end

```

```

c
c*-----*
c*           S U B R O U T I N E       F C N J           *
c* THIS IS A DUMMY FUNCTION. REFER DIFSUB ROUTINE FOR DETAILS *
c*-----*

```

```

c
  subroutine fcnj(tintial,x2,pd,ncpmt)
  implicit double precision(a-h,o-z)
  integer ncpmt
  real pd(ncpmt,ncpmt)
  real*8 tintial, x2(8,ncpmt)
c   do 10, ij = 1, ncpmt
c     pd(ij,ij) = a(ij,ij)
c 10 continue
  return
end

```

```

c
c*-----*
c*           F U N C T I O N       I N F L O W           *
c* THIS FUNCTION IS TO COMPUTE THE MASS INFLOW FROM THE SOURCE *
c* INTO THE PLANT USING TRAPEZOIL METHOD. *
c* THIS FUNCTION IS CALLED FROM FILE 'STIFF3V2.FOR' SUBROUTINE *
c* INTGRL. *
c*-----*

```

c

```

      real*8 function inflow()
$include:'combloc'
      temp1 = dt*((difu(1)*kl(1)/deltax(1))+(kl(1)*q(1)*
1          (1.0-sigma(1))))*conc0*area(1)+(js*area(1)*dt)
      temp2 = (dt/2.0)*(difu(1)*kr(1)*area(1)/(deltax(1)*vol(1)*
2          (1.0+sb(1))))*(xnow(1)+xold(1))
      inflow = temp1 - temp2
      return
      end

c
c*-----*
C*          F U N C T I O N          T L A M B D A          *
C*  THIS FUNCTION IS TO COMPUTE THE MASS LOST FROM THE PLANT *
C*  USING TRAPEZOIL METHOD.  IT COMPUTES THE TOTAL LAMBDA LOST *
C*  AND THE LOST FROM EACH COMPARTMENT. *
C*  THIS FUNCITON IS CALLED FROM FILE 'STIFF3V2.FOR' SUBROUTINE *
C*  INTGRL. *
c*-----*
c
      real*8 function tlambda()
$include:'combloc'
      sum = 0.0
      do 10 i=1,29
          temp = (lambda(i)*dt*(xnow(i)+xold(i)))/
*              (2.0*(1.0+sb(i)))
C
C-----SUM UP EACH MASS LOST
          sum = sum + temp
C
C-----STORE EACH MASS LOST
          lost(i) = lost(i) + temp
      10 continue
C
C-----TOTAL MASS LOST
      tlambda = sum
      return
      end

c
c*-----*
C*          S U B R O U T I N E          M E N U          *
C*  THIS SUBROUTINE DISPLAY THE OUTPUT OPTION AS A SCREEN MENU. *
C*  USER IS ALLOW TO PICK ANY COMBINATION OF OUTPUTS.  THE OUTPUT *
C*  NUMBER SELECTED WILL BE PASSED TO THE CALLING STATEMENT. *
C*  AFTER AN OUTPUT NUMBER HAS BEEN SELECTED, IT WILL ASK USER *
C*  FOR VERIFICATION THEN USER HAS TO SELECT NUMBER 6 TO EXIT *
C*  THIS MENU. *
C*  THIS SUBROUTINE IS CALLED FROM FILE 'STIFF3V2.FOR' SUBROUTINE *
C*  STIFF. *
c*-----*
c
      subroutine menu(opick)

```

```

integer opick
character respond*1

c
5 write(*,10)
10 format(3(/),16x,'!!!! Simulation has been completed !!!!!',
* ///,16x,'Choices of output listing.',/,
* /,16x,'[1] Individual mass in compartments. ',/,
* 16x,'[2] Individual mass loss in compartments.',/,
* 16x,'[3] Grouped mass; Root, Stems and Leaves.',/,
* 16x,'[4] Grouped mass; Root, Stems and Leaves',
* ' with lambda.',/,
* 16x,'[5] List all of the above.',/,
* 16x,'[6] Exit to DOS.',3(/),
* 16x,'Please select your favorite output number >- ',\))
read(*,*) opick

c
c-----RETRY FOR USER ENTERING WRONG CHOICES.
if ((opick .lt. 1).or.(opick .gt. 6)) then
call clrscrn
call scrnhd
goto 5
endif

c
c-----CHECK FOR WRONG SELECTION.
write(*,20)
20 format(/,16x,'Are you sure (y/n) ? >- ',\))
read(*,30) respond
30 format(a)
if ((respond.ne.'y').and.(respond.ne.'Y')) then
call clrscrn
goto 5
endif
return
end

c
c*-----*
C*          S U B R O U T I N E          M S S G E 3          *
C* THIS SUBROUTINE FIRST DISPLAY A MESSAGE ON SCREEN WHEN MASS *
C* BALANCE EQUATIONS ARE CONSTRUCTED INTO A MATRIX FORM. FROM *
C* THIS POINT USER CAN TERMINATE THE PROGRAM AS THEY WISH, OR *
C* PROCEED TO SIMULATE. IT THEN DISPLAY A MESSAGE SHOWING *
C* THE SIMULATED TIME. *
C* THIS SUBROUTINE IS CALLED FROM FILE 'STIFF3V2.FOR' SUBROUTINE *
C* MSSGE3. *
c*-----*
c
subroutine mssge3
$include:'combloc'
character*1 action
write(*,10)
10 format(5(/),16x,'The mass balance equations are being',

```

```

*          ' constructed.',/,
*          //,16x,'Do you want simulation to proceed',
*          ' (Y/N) >- ',\
read(*,30) action
if ((action .eq. 'n').or.(action .eq. 'N')) then
    call setpr('off')
    stop
endif
c
    call clrscrn
    call scrnhd
    write(*,20) int(tstop)
20 format(3(/),23x,'----- Simulation in process. -----',//,
*      /,20x,'Targeted simulation time is ',i7,' hours.',
*      //,20x,'If the program issues a warning message, ',/,
*      20x,'do not panic, but let the execution proceed. ',/,
*      20x,'Refer to the user guide for explanation of the ',/,
*      20x,'warning message(s).'3(/),
*      20x,'The simulated time is          hour(s).')
c
30 format(a)
    return
    end

```

*DECINT Last revision: November 25, 1987

SUBROUTINE DECINT(PRMT, X, XMAX, NM, KFLAG, FCT, FCTJ, OUTP,
& ERROR, SAVE, PW)

\$include:'combloc'

INTEGER NM, KFLAG, IPRT
DOUBLE PRECISION PRMT, X, XMAX, SAVE, ERROR
REAL PW
DIMENSION X(8,NM), XMAX(NM), SAVE(12*NM,NM), PW(NM,NM),
& IPRT(4), ERROR(NM), PRMT(5+3)

C Needed by DIFSUB:

INTEGER MF, JSTART, MAXDER
DOUBLE PRECISION T, H, HMIN, HMAX, TOL, TEND

C Locals

REAL TEST
INTEGER ITER, I, IMESS
CHARACTER EMESS(5)*60
LOGICAL EFLAG, RETRY
EXTERNAL FCT, FCTJ, OUTP

* Mapping routine to make the call to DIFSUB behave similarly to the *
* other ODE integration routines. *

* This routine tests the user parameters and then iterates from T0 *
* thru TEND with calls to DIFSUB. DIFSUB is essentially the routine *
* by C.W. Gear; "DIFSUB for Solution of Ordinary Differential Equa- *
* tions"; Algorithm 407; CACM 14:3, March 1971. DIFSUB was designed *
* to integrate a single step in which it adapts the step size and/or *
* order of the method to a desired tolerance. Since this method *
* results in rather odd step sizes being returned, the desired points *
* are passed to the OUTP routine (which is an external argument) for *
* interpretation and interpolation. The arguments H, HMIN, etc. are *
* described in DIFSUB; the PRMT array, unique to this routine, holds: *

* PRMT(1) - The starting value, T0, *

* PRMT(2) - The ending value, TEND, *

* PRMT(3) - The initial step size, H, *

* PRMT(4) - The minimum step size to be used, HMIN, *

* PRMT(5) - The tolerance, TOL. The single step error *

* estimate / YMAX(I) will be less than this in the *

* Euclidean norm. *

* PRMT(6) - The maximum step size, HMIN, *

* PRMT(7) - The method to be used: *

0 == Adams-Bashforth-Moulton (ABM) *

1 == Gear's stiff method with the partial *

derivatives for the Jacobian matrix *

provided by the FCTJ subroutine *

described in DIFSUB. *

2 == Gear's stiff method, where the part- *

ials are estimated numerically. FCTJ *

is a dummy routine just for linking. *

* PRMT(6) - The maximum order to be used. DIFSUB adjusts *

```

*           this up or down (with the current value in      *
*           KFLAG), but will not exceed MAXDER.             *
*   FCT      -   Is the routine which DIFSUB calls DIFFUN, with *
*               the form FCT(T, Y, DY)                      *
*   FCTJ     -   Is the routine which DIFSUB calls PEDERV, with *
*               the form FCTJ(T, Y, PW, M),                 *
*   OUTP     -   Is the user's output routine to capture and *
*               evaluate the mesh points.  A generic form,  *
*               DOUTP is provided following DECINT.         *

```

```

*****
*           David E. Cawlfieid,  Fall 1987                  *
*****

```

```

DATA EMESS/

```

```

& ' Integration halted by user (KFLAG set to zero in DOUTP). ',
& ' Step taken with H=Hmin; but the requested error not met. ',
& ' The maximum order specified was found to be too large. ',
& ' Corrector convergence could not be achieved for H > Hmin. ',
& ' The requested error too small for this problem. ' /

```

```

IF ( KFLAG .GT. 0 ) GO TO 100
IF ( KFLAG .EQ. -1 ) THEN
  H = PRMT(3)
  GO TO 100

```

```

ENDIF

```

```

T      = PRMT(1)
TEND  = PRMT(2)
H      = PRMT(3)
HMIN  = PRMT(4)
TOL   = PRMT(5)
HMAX  = PRMT(6)
MF    = INT(PRMT(7))
MAXDER = INT(PRMT(8))

```

```

C

```

```

C Test user parameters for validity.  This need only be done once,
C before using DIFSUB to iterate from T0 through TEND.  For speed,
C DIFSUB does *not* test parameters.

```

```

C

```

```

EFLAG = .FALSE.
IF ( T .GT. TEND ) THEN
  WRITE(*,*) '*DECINT* error -- T > TEND'
  EFLAG = .TRUE.
ENDIF
IF ( HMIN .GT. H ) THEN
  WRITE(*,*) '*DECINT* error -- HMIN > H'
  EFLAG = .TRUE.
ENDIF
IF ( H .GT. HMAX ) THEN
  WRITE(*,*) '*DECINT* error -- H > HMAX'
  EFLAG = .TRUE.
ENDIF
IF ( TOL .LE. 0.D0 ) THEN
  WRITE(*,*) '*DECINT* error -- TOL <= 0 ', TOL

```



```

        EFLAG = .TRUE.
    ENDIF
    IF ( MF .LT. 0 .OR. MF .GT. 2 ) THEN
        WRITE(*,*) '*DECINT* error -- METHOD? MF =', MF
        EFLAG = .TRUE.
    ENDIF
    IF ( (MF .LT. 1 .AND. MAXDER .GT. 7) .OR.
&      (MF .GE. 1 .AND. MAXDER .GT. 6) ) THEN
        WRITE(*,*) '*DECINT* error -- Wrong order.', MAXDER
        EFLAG = .TRUE.
    ENDIF
    IF ( EFLAG ) THEN
        KFLAG = -5
        RETURN
    ENDIF
C
C Initial point...Statement 500 below is the error trap...
C We don't like GO TO's <yecch!>, but this is the quickest way.
C DIFSUB returns KFLAG = +1 if successful, -<n> otherwise.
C
        KFLAG = 0
        JSTART = 0
        RETRY = .FALSE.
Cold CALL OUTP(NM, T, H, X, HL, ERROR, KFLAG, JSTART, KFLAG)
C
    100 CALL DIFSUB(NM,T,X,SAVE,H,HMIN,HMAX,TOL,MF,XMAX,ERROR,KFLAG,
&      JSTART,MAXDER,PW,FCT,FCTJ)
C
        IF ( KFLAG .LT. 1 ) GO TO 500
        ITER = 1
        CALL OUTP(NM, T, H, X, HL, ERROR, KFLAG, JSTART, ITER)
C
C Iterate thru TEND...
C
    200 TEST = H*(T + H - TEND)
        IF ( TEST .LE. 0.0 ) THEN
            CALL DIFSUB(NM,T,X,SAVE,H,HMIN,HMAX,TOL,MF,XMAX,ERROR,KFLAG,
&      JSTART,MAXDER,PW,FCT,FCTJ)
            IF ( KFLAG .LT. 1 ) GO TO 500
            ITER = ITER + 1
            CALL OUTP(NM, T, H, X, HL, ERROR, KFLAG, JSTART, ITER)
            GO TO 200
        ENDIF
C
        RETURN
C
C If integration failed, try once to restart it ...
C RETRY is set true iff a step was re-tried.
C
    500 IMESS = 1-KFLAG
        WRITE(*,2000) EMESS(IMESS)

```

```

      IF ( RETRY ) THEN
        WRITE(*,2030)
        KFLAG = -6
      ELSE
        WRITE(*,2020)
Cdbg  PRINT '(/G12.6)', TINTIAL, HL, PRMT
        PRMT(1) = TINTIAL
        PRMT(3) = 0.1D0 * HL
        PRMT(4) = 0.01D0 * HL
        DO 510, J = 1, NM
          X(1,J) = XNOW(J)
510    CONTINUE
        ITER      = 0
        JSTART    = -1
        RETRY     = .TRUE.
        GO TO 200
      ENDIF
      RETURN
2000  FORMAT(/,1X,A,/)
2010  FORMAT(1X,F7.4,5(2X,G12.6))
2020  FORMAT(/,1X,'Integration step failed.  Attempting to restart ',
          & 'with smaller step size.')
2030  FORMAT(/,1X,'Re-start failed also.')
      END

```

*FOUTP

```

      SUBROUTINE FOFTP(NDIM, T, H, X, HL, ERROR, KFLAG, NQ, IPTS)
      $include:'combloc'

```

C Parameters...

```

      INTEGER      NDIM, KFLAG, NQ
      DOUBLE PRECISION  T, H, X, HL, ERROR
      DIMENSION X(8,NDIM), ERROR(NDIM)

```

C Restart control...

```

      INTEGER      NDM
      INTEGER      LUNPR, INEXT
      DOUBLE PRECISION  TNXT, PINK
      COMMON      /F1/ LUNPR, INEXT, TNXT

```

C Local control

```

      INTEGER      HOLD, IDLE, I, J
      DOUBLE PRECISION  S, E
      DOUBLE PRECISION  INFLOW, TLAMBDA
      EXTERNAL      INFLOW, TLAMBDA

```

```

* Boiler plate type FOFTP routine for the DIFSUB type output.      *
* DIFSUB returns the "time" points at rather odd intervals (it uses *
* a multiplicative factor to change H), and so the desired mesh point *
* must be interpolated as described in the DIFSUB program document.  *
* The value of IDLE and the second IF... statement are used to save *
* non-mesh points (every HOLD interval) for re-starting purposes.    *
* The user may set KFLAG to 0 in this routine, to stop DECINT.      *
* The key variables in this routine are:                             *
*

```

```

*      PNXT, PINK      The next value to "print" at, and the interval,  *
*      TINTIAL,HL     Holding values for re-starting purposes,        *
*      HOLD, IDLE     Counters for when to store holding values,      *
*                                                              *
* The interpolation loop (E=,S=,DO 20..) is described in DIFSUB.      *
*****
*      David E. Cawlfeld, Fall 1987                                  *
*****
      DATA HOLD/ 50 /, ITT/0/, COMPMT0/0/, TLOST/0/, IDLE/-1/
      IF ( T + H .GT. TNXT ) THEN
          E = TNXT - T
          S = E / H
          DO 20, I = 1, NDIM
              XNOW(I) = X(1,I)
              DO 10, J = 1, NQ
                  XNOW(I) = XNOW(I) + X(J+1,I)*S**J
          10      CONTINUE
          20      CONTINUE
          TINTIAL = T + E
          HL      = H
C-----Indicate every simulated hour to screen
          IT = INT(TINTIAL)
          IF (IT .GT. IIT) THEN
              CALL CURSOR(20,42)
              WRITE(*,50) IT
              CALL CURSOR(20,49)
              IIT = IT
          50      FORMAT(I8,\)
          ENDIF
Cdbg      WRITE(*,2000) TINTIAL, H, KFLAG, NQ, IPTS
Cdbg      WRITE(*,2010) (XNOW(I), I = 1, NDIM)
Cdbg      WRITE(*,2020) (ERROR(I), I = 1, NDIM)
C
C-----Accumulate total mass inflow and total mass loss (lambda)
          COMPMT0 = COMPMT0 + INFLOW()
          TLOST = TLOST + TLAMBDA()
C
C-----Store mass at user specified time increments
          IF (TPRT .GT. PRINTVL) THEN
              DO 80 J=2,NCPMT+1
                  MASS(ITIME,J) = XNOW(J-1)
          80      CONTINUE
C
C-----Store current simulated time to storage area.
          MASS(ITIME,1) = TINTIAL
          MLOS(ITIME,1) = TINTIAL
          MSORB(ITIME,1) = TINTIAL
C
C-----Store accumulated mass inflow and mass loss to storage area.
          MASS(ITIME,31) = COMPMT0
          MLOS(ITIME,31) = TLOST

```

```

C
C-----Store mass loss at instant simulated time.
          DO 90 II=1,NCPMT
            MLOS(ITIME,II+1) = LOST(II)
          90      CONTINUE
C
C-----Increment next time storage location
          ITIME = ITIME + 1
C
C-----Reset printing interval flag to zero for next storage
C      increments.
          TPRT=0.0
          IDLE = 0
          END IF
          TNXT = TNXT + DT
          TPRT = TPRT + DT
          DO 45, I = 1, NCPMT
            XOLD(I) = XNOW(I)
          45      CONTINUE
          INEXT = INEXT + 1
          ENDIF
C
C This section saves intermediate values for re-start, if not enough
C "mesh" points have been reported/accepted.
C
          IDLE = IDLE + 1
          IF ( MOD(IDLE, HOLD) .EQ. 0 ) THEN
            TINTIAL = T
            HL      = H
            DO 30, J = 1, NDIM
              XNOW(J) = X(1,J)
            30      CONTINUE
Cdbg      WRITE(*,'('' Idle'',\)'')
          ENDIF
          RETURN
2000 FORMAT(//,' Time = ',F9.3, ', H = ', G12.6, ', KFLAG =', I2,
          & ', NQ =',I2, ', Points = ',I6)
2010 FORMAT(1H , 'X(I): ', 4(F12.6, :,4X),/, ( T8, 4(F12.6, :,4X) ) )
2020 FORMAT(1H , 'Err : ', 4(F12.6, :,4X),/, ( T8, 4(F12.6, :,4X) ) )
          END

```

```

C
C   T H I S       I S       F I L E       P R I N T 3 L V . F O R
c
C*-----*
c*
c*   P L A N T       T R A N S P O R T       T H R E E       L E A F       *
c*                               M O D E L                               *
c*                               (version 2.1)                         *
c*
c*   Program manager : Dr. Craig McFarlane                           *
c*
c*   Programmer : Kai P. Suen           06/01/87.                     *
c*                               Oregon State University               *
c*                               Soil Science Department               *
c*                               Corvallis, Oregon 97331               *
c*
c*   Program developed for EPA grant no. CR811940-01-03               *
c*
c*                               All rights reserved                   *
C*-----*
c
c
C*-----*
c*           S U B R O U T I N E       O U T M A S S                   *
C*   THIS SUBROUTINE PRINT OUT ALL THE INDIVIDUAL MASS AT THE       *
C*   USER REQUESTED INTERVAL.  THE OUTPUT INCLUDES PAGE HEADER,    *
C*   PAGE NUMBER, PRINT OUT (LISTING) FILENAME, AND THE PLANT      *
C*   COMPARTMENT NUMBERS.  IT SKIPS TO A NEW PAGE EVERY FIFTY     *
C*   ARE PRINTED.
C*   THIS SUBROUTINE IS CALLED FROM FILE 'STIFF3V2.FOR' SUBROUTINE *
C*   STIFF.
C*-----*
      subroutine outmass
$include:'combloc'
      logical done
c
C-----OUTPUT FORMATS
c
      2000 format(34(1x,e10.4))
      2010 format(1x,e10.4,4x,10(1x,e10.4))
c
c-----OUTPUT MASS DISTRIBUTION TO FILE 'MASSFLW.OUT'.
c   THIS FILE DOES NOT INCLUDE PAGE HEADER.  THE OUTPUT
c   IS AS FOLLOW:
c       COLUMN 1 : TIME (IN HOURS)
c       COLUMN 2 - 30: COMPARTMENT 1 THROUGH COMPARTMENT 29
c
      do 10 i=1,itime
          write(4,2000) (mass(i,j),j=1,ncpmt+1)
10 continue

```

```

c
c-----PRINT OUT MASS DISTRIBUTIONS TO USER SPECIFIED OUTPUT FILENAME.
C      IT INCLUDES PAGE HEADER, PAGE NUMBER, OUTPUT FILENAME AND
C      COMPARTMENT NUMBERS.
c
      npage = 1
      done = .false.
      jj = 2
      jjj = 10
      call masshd(1,jj-1,jjj-1)
20 do 50 i=1,itime
      write(3,2010) mass(i,1),(mass(i,j),j=jj,jjj)
c
C-----SKIP TO A NEW PAGE FOR EVERY 50 LINES AND PRINT APPROPRIATE
C      PAGE HEADER.
c
      if (mod(i,50).eq.0) then
        if (npage .lt. 4) then
          call masshd(2,jj-1,jjj-1)
        else
          call masshd(2,28,29)
        endif
      endif
50 continue
c
c-----IF DONE RETURN TO CALLING STATEMENT
      if (done) goto 999
      npage = npage + 1
C
C-----IF NUMBER OF PAGE IS LESS THAN 4 THEN PRINT EVERY NINE
C      COMPARTMENT ON ONE PAGE ELSE PRINT THE LAST TWO COMPARTMENT
C      ON PAGE 4.
C
      if (npage .lt. 4) then
        jj = jjj + 1
        jjj = jjj + 9
        call masshd(1,jj-1,jjj-1)
      else
        call masshd(1,28,29)
        jj = 29
        jjj = 30
        done = .true.
      endif
c
C-----REPEAT PROCESS UNTIL DONE IS SET TO TRUE, THEN EXIT SUBROUTINE
      goto 20
999 return
end
C
C*-----*
C*          S U B R O U T I N E          O U T M L O S          *

```

```

C* THIS SUBROUTINE PRINT OUT THE MASS LOST (LAMBDA) IN PLANT. *
C* THE OUTPUT INCLUDES PAGE HEADER, PAGE NUMBER, PRINT OUT *
C* FILENAME, AND THE PLANT COMPARTMENT NUMBERS. *
C* THIS SUBROUTINE IS CALLED FROM FILE 'STIFF3V2.FOR' SUBROUTINE *
C* STIFF. *
C*-----*
C
      subroutine outmlos
$include:'combloc'
c
C-----OUTPUT FORMAT
C
      2000 format(1x,e10.4,4x,10(1x,e10.4))
c
      npage = 1
      jj = 2
      jjj = 10
      call mloshd(1,0,jj-1,jjj-1)
c
C-----PRINT OUT MASS LOST TO USER SPECIFIED OUTPUT FILENAME.
C      IT INCLUDES PAGE HEADER, PAGE NUMBER AND OUTPUT FILENAME AND
C      COMPARTMENT NUMBERS.
C
C-----SKIP TO A NEW PAGE FOR EVERY 50 LINES AND PRINT APPROPRIATE
C      PAGE HEADER.
C
      10 do 20 i=1,itime
          write(3,2000) mlos(i,1),(mlos(i,j),j=jj,jjj)
          if (mod(i,50).eq.0) call mloshd(2,0,jj-1,jjj-1)
      20 continue
c
      npage = npage + 1
C
C-----IF NUMBER OF PAGE IS LESS THAN 4 THEN PRINT EVERY NINE
C      COMPARTMENT ON ONE PAGE ELSE PRINT THE LAST TWO COMPARTMENT
C      ON PAGE 4.
C
      if (npage .lt. 4) then
          jj = jjj + 1
          jjj = jjj + 9
          call mloshd(1,0,jj-1,jjj-1)
          goto 10
      else
          call mloshd(1,1,28,29)
          do 30 i=1,itime
              write(3,2000) mlos(i,1),(mlos(i,j),j=29,30)
              if (mod(i,50).eq.0) call mloshd(2,1,28,29)
          30 continue
      endif
c
      return

```

```

end
C
C*-----*
C*          S U B R O U T I N E          O U T G R O U P          *
C* THIS SUBROUTINE COMPUTES THE GROUPED MASS.  SUCH AS, ROOT,    *
C* STEMS, LEAFS AND PLANT TOTAL.  THE METHOD USED TO COMPUTE THE *
C* GROUPED MASS IS TO SUM UP THE APPROPRIATE COMPARTMENTS WHICH *
C* PREDEFINED IN THE MODEL.                                     *
C* THIS SUBROUTINE IS CALLED FROM FILE STIFF3V2.FOR SUBROUTINE *
C* STIFF.                                                       *
C* **NOTE**                                                    *
C* MASS LOST (LAMBDA) ARE NOT INCLUDED IN THE ROOT, STEMS, OR  *
C* LEAFS.                                                       *
C* LOCAL VARIABLE DEFINITIONS                                  *
C*-----*
C* SUMR = ROOT SUMS (INCLUDED COMPARTMENTS ARE 1,2,3,14,15) *
C* STEMS = STEM SUMS (INCLUDED COMPARTMENTS ARE 4,5,6,16,17, *
C*                24,29,11,12, *
C*                13,22,23) *
C* SUMLF1 = LEAF 1 SUMS (INCLUDED COMPARTMENTS ARE 18,19,20,21) *
C* SUMLF2 = LEAF 2 SUMS (INCLUDED COMPARTMENTS ARE 25,26,27,28) *
C* SUMLF3 = LEAF 3 SUMS (INCLUDED COMPARTMENTS ARE 7,8,9,10) *
C* TOTAL = TOTAL PLANT MASS (SUMR+STEMS+SUNLF1+SUNLF2+SUNLF3+ *
C*                MLOS(I,31) ) *
C* MLOS(I,31) = TOTAL MASS LOST IN PLANT. *
C*-----*
      subroutine outgroup
$include:'combloc'
C
C-----OUTPUT FORMAT
      2000 format(1x,e10.4,4x,7(1x,e10.4),2x,e10.4)
C
      npage = 1
      call gmasshd
      do 50 i=1,itime
          sumr = 0.0
          sums = 0.0
          sumlf1 = 0.0
          sumlf2 = 0.0
          sumlf3 = 0.0
c
c-----SUM ROOT MASS
          do 10 j=2,4
              sumr = sumr + mass(i,j)
          10 continue
          sumr = sumr + mass(i,15) + mass(i,16)
c
c-----SUM STEMS MASS
          do 20 j=5,7
              sums = sums + mass(i,j) + mass(i,j+7)

```



```

20  continue
    sums = sums + mass(i,17) + mass(i,18) + mass(i,23) +
*      mass(i,24) + mass(i,25) + mass(i,30)
c
c-----SUM LEAF MASS
    do 30 j=8,11
c-----LEAF 3
        sumlf3 = sumlf3 + mass(i,j)
c-----LEAF 2
        sumlf2 = sumlf2 + mass(i,j+18)
c-----LEAF 1
        sumlf1 = sumlf1 + mass(i,j+11)
    30  continue
        total = sumr+sums+sumlf1+sumlf2+sumlf3+mlos(i,31)
C
C-----PRINT OUT GROUPED MASS TO USER SPECIFIED OUTPUT FILENAME.
C
        write(3,2000) mass(i,1),sumr,sums,sumlf1,sumlf2,sumlf3,
*      mlos(i,31),total,mass(i,31)
C
C-----PRINT OUT GOUPED MASS TO FILE 'GRPMAS.OUT'
C
        write(5,2000) mass(i,1),sumr,sums,sumlf1,sumlf2,sumlf3,
*      mlos(i,31),total,mass(i,31)
C
C-----SKIP TO A NEW PAGE FOR EVERY 50 LINES PRINTED
C
        if (mod(i,50).eq.0) then
            npage = npage + 1
            call gmasshd
        endif
50  continue
    return
end

C
C*-----*
C*      S U B R O U T I N E      O U T R S L W L      *
C*      THIS SUBROUTINE PRINT OUT THE GROUPED MASS WITH MASS LOST *
C*      ASSOCIATED WITH THE ROOT, STEMS, LEAFS AND PLANT TOTAL. THE *
C*      METHOD USED TO COMPUTE THE GROUPED MASS IS TO SUM UP THE *
C*      APPROPRIATE COMPARTMENTS WHICH PREDEFINED IN THE MODEL. *
C*      THIS SUBROUTINE IS CALLED FROM FILE STIFF3V2.FOR SUBROUTINE *
C*      STIFF. *
C*      **NOTE** *
C*      MASS LOST (LAMBDA) ARE INCLUDED IN THE ROOT, STEMS AND LEAFS *
C* *
C*      LOCAL VARIABLE DEFINITIONS *
C*      ----- *
C*      SUMR = ROOT SUMS (INCLUDED COMPARTMENTS ARE 1,2,3,14,15) *
C*      STEMS = STEM SUMS (INCLUDED COMPARTMENTS ARE 4,5,6,16,17, *
C*      24,29,11,12, *

```

```

C*                                     13,22,23)      *
C*   SUMLF1 = LEAF 1 SUMS (INCLUDED COMPARTMENTS ARE 18,19,20,21) *
C*   SUMLF2 = LEAF 2 SUMS (INCLUDED COMPARTMENTS ARE 25,26,27,28) *
C*   SUMLF3 = LEAF 3 SUMS (INCLUDED COMPARTMENTS ARE 7,8,9,10)   *
C*   TOTAL  = TOTAL PLANT MASS (SUMR+STEMS+SUNLF1+SUNLF2+SUNLF3) *
C*-----*
C
      subroutine outrslwl
$include:'combloc'
C
C-----OUTPUT FORMAT
      2000 format(1x,e10.4,4x,6(1x,e10.4),4x,e10.4)
C
      npage = 1
      call grslwlhd
      do 50 i=1,itime
          sumr   = d0
          sums   = d0
          sumlf1 = d0
          sumlf2 = d0
          sumlf3 = d0
c
c-----SUM ROOT MASS WITH LAMBDA
      do 10 j=2,4
          sumr = sumr + mass(i,j) + mlos(i,j)
10    continue
          sumr = sumr + mass(i,15) + mlos(i,15)
          *      + mass(i,16) + mlos(i,16)
c
c-----SUM STEMS MASS WITH LAMBDA
      do 20 j=5,7
          sums = sums + mass(i,j) + mlos(i,j)
          *      + mass(i,j+7) + mlos(i,j+7)
20    continue
          sums = sums + mass(i,17) + mass(i,18) + mass(i,23) +
          *      mass(i,24) + mass(i,25) + mass(i,30) +
          *      mlos(i,17) + mlos(i,18) + mlos(i,23) +
          *      mlos(i,24) + mlos(i,25) + mlos(i,30)
c
c-----SUM LEAF MASS WITH LAMBDA
      do 30 j=8,11
C-----LEAF 3
          sumlf3 = sumlf3 + mass(i,j) + mlos(i,j)
C-----LEAF 2
          sumlf2 = sumlf2 + mass(i,j+18) + mlos(i,j+18)
C-----LEAF 1
          sumlf1 = sumlf1 + mass(i,j+11) + mlos(i,j+11)
30    continue
          total = sumr+sums+sumlf1+sumlf2+sumlf3
          write(3,2000) mass(i,1),sumr,sums,sumlf1,sumlf2,sumlf3,
          *      total,mass(i,31)

```

```

        write(6,2000) mass(i,1),sumr,sums,sumlf1,sumlf2,sumlf3,
*           total,mass(i,31)
C
C-----SKIP TO A NEW PAGE FOR EVERY 50 LINES.
        if (mod(i,50).eq.0) then
            npage = npage + 1
            call grslwlhd
        endif
50 continue
        return
        end

C*-----*
C*           S U B R O U T I N E           M A S S H D           *
C* THIS SUBROUTINE PRINT OUT THE MASS HEADING AND IS CALLED *
C* FROM FILE PRINT3V2.FOR SUBROUTINE OUTMASS. *
C*-----*
        subroutine masshd(nn,jj,jjj)
$include:'combloc'
        character*7 long(2)
        data long/'      ','(conc.)'/
C
C-----OUTPUT FORMATS
2000 format('1',/,a/, ' This FILENAME IS --> ',a,/,/,
*          3x,'Date : ',a,85x,'page ',i4,1x,a7,/,/,
*          42x,'      Mass In  Compartment',/)
2010 format(3x,'time',11x,10(1x,i3,7x))
c
c-----print mass output heading
        write(3,2000) pghd,outfile,date,npage,long(nn)
        write(3,2010) (i,i=jj,jjj)
        return
        end

C
C*-----*
C*           S U B R O U T I N E           G M A S S H D           *
C* THIS SUBROUTINE PRINT OUT HEADING FOR GROUPED MASS WITHOUT *
C* LAMBDA AND IS CALLED FROM FILE PRINT3V2.FOR SUBROUTINE *
C* OUTGROUP. *
C*-----*
C
        subroutine gmasshd
$include:'combloc'
C
C-----OUTPUT FORMAT
2000 format('1',/,a/, ' This FILENAME is -->',a,/,/,
*          3x,'Date : ',a,85x,'page ',i4,/,/,35x,
*          'Grouped      Mass',/,/,74x,'Total',/,3x,'time',
*          12x,'roots  + stems  + leaf1  + leaf2  + ',
*          'leaf3  + lambda = plant  inflow')
c
C-----PRINT GROUPED MASS OUTPUT HEADING TO FILE 'GRPMAS.OUT'

```

```

C
    write(5,2000) pghd,outfile,date,npage
C
C-----PRINT GROUPED MASS OUTPUT HEADING USER SPECIFIED FILENAME.
    write(3,2000) pghd,outfile,date,npage
    return
    end
C
C*-----*
C*          S U B R O U T I N E      G R S L W L H D          *
C* THIS SUBROUTINE PRINT OUT HEADING FOR GROUPED MASS WITH    *
C* LAMBDA AND IS CALLED FROM FILE PRINT3V2.FOR SUBROUTINE    *
C* OUTRSLWL.                                                  *
C*-----*
C
    subroutine grslwlhd
$include:'combloc'
C
C-----OUTPUT FORMAT
2000 format('1',/,a/, ' This FILENAME is --> ',a,//,
*          3x,'Date : ',a,85x,'page ',i4,//,32x,
*          'Grouped      Mass      with      Lambda',//,3x,'time',
*          12x,'roots  + stems  + leaf1  + leaf2  + ',
*          'leaf3  = plant      inflow')
C
C-----PRINT GROUPED MASS OUTPUT HEADING TO FILE 'GRPRSLWL.OUT'
C
    write(6,2000) pghd,outfile,date,npage
C
C-----PRINT GROUPED MASS OUTPUT HEADING TO USER SPECIFIED FILENAME.
C
    write(3,2000) pghd,outfile,date,npage
    return
    end
C
C*-----*
C*          S U B R O U T I N E      M L O S H D          *
C* THIS SUBROUTINE PRINT OUT MASS LOST HEADING AND IS CALLED  *
C* FILE PRINT3V2.FOR SUBROUTINE OUTMLOS.                    *
C*-----*
C
    subroutine mloshd(nn,kk,jj,jjj)
$include:'combloc'
    character*7 long(2)
    data long/      ', '(conc.)'/
C
C-----OUTPUT FORMATS
C
2000 format('1',/,a/, ' This FILENAME is --> ',a,//,
*          3x,'Date : ',a,85x,'page ',i4,lx,a7,//,
*          42x,'Mass      Loss      In      Compartment',/)

```

```
2010  format(3x,'time',11x,10(1x,i3,7x))
2020  format(3x,'time',11x,6(1x,i3,7x),2x,'sum')
c
C-----PRINT MASS OUTPUT HEADING
      write(3,2000) pghd,outfile,date,npage,long(nn)
      if (kk .eq. 0) then
        write(3,2010)(i,i=jj,jjj)

      else
        write(3,2020) (i,i=jj,jjj)
      endif
      return
      end
```