

AD-A143 024

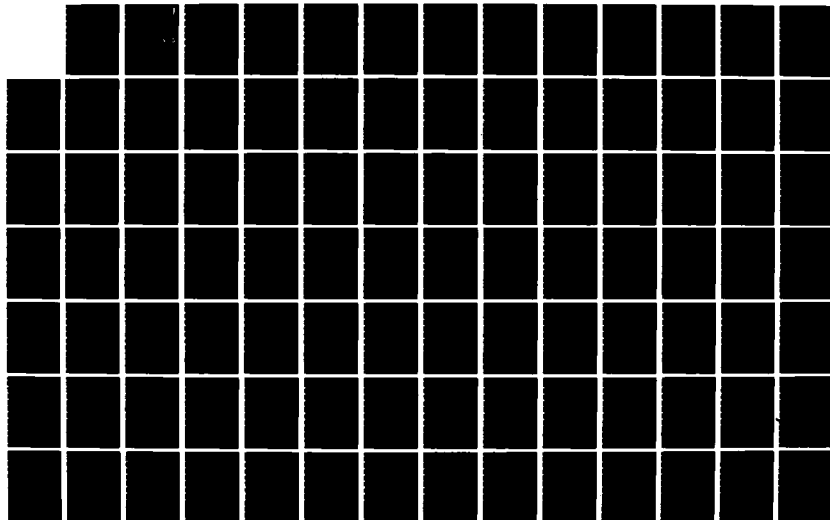
MODELING AND SIMULATION OF WASTEWATER REUSE SYSTEMS -
DYNAMIC PROCESS SIMULATOR(U) LOUISIANA STATE UNIV BATON
ROUGE DEPT OF CHEMICAL ENGINEERING. C L SMITH ET AL.
MAY 82 DAMD17-77-C-7040

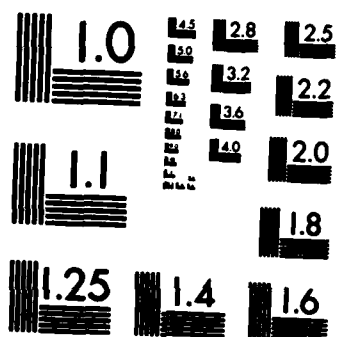
1/2

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

PHOTOGRAPH THIS SHEET

INVENTORY

LEVEL

DTIC ACCESSION NUMBER

AD-A143 024

Final Rpt., 1 July 77 - 31 May '80
Contract DAMD-17-77-C-7040

DOCUMENT IDENTIFICATION

May '82

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR

NTIS GRA&I

DTIC TAB

UNANNOUNCED

JUSTIFICATION



BY

DISTRIBUTION /

AVAILABILITY CODES

DIST

AVAIL AND/OR SPECIAL

A/1



DISTRIBUTION STAMP

DTIC ELECTE
S JUL 5 1984 D
D

DATE ACCESSIONED

DATE RETURNED

REGISTERED OR CERTIFIED NO.

DATE RECEIVED IN DTIC

84 07 05 00 1

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDAC

AD-A143 024

AD _____

MODELING AND SIMULATION OF WASTEWATER
REUSE SYSTEMS - DYNAMIC PROCESS SIMULATOR

FINAL REPORT

CECIL L. SMITH, DAVID M. STARKS,
AND WARREN T. ABBOTT

MAY 1982

ARTHUR M. STERLING, FINAL PRINCIPAL INVESTIGATOR

Supported by

U.S. ARMY MEDICAL RESEARCH AND DEVELOPMENT COMMAND
Fort Detrick, Frederick, Maryland 21701

Contract No. DAMD 17-77-C-7040

Project Officer - Mitchell J. Small
U.S. ARMY MEDICAL BIOENGINEERING RESEARCH
AND DEVELOPMENT LABORATORY
Fort Detrick, Frederick, Maryland 21701

Department of Chemical Engineering
Louisiana State University
Baton Rouge, Louisiana 70803

DOD DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Modeling and Simulation of Wastewater Reuse Systems - Dynamic Process Simulator		5. TYPE OF REPORT & PERIOD COVERED Final Report July 1, 1977 - May 31, 1980
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Cecil L. Smith David M. Starks Warren T. Abbott		8. CONTRACT OR GRANT NUMBER(s) DAMD 17-77-C-7040
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Chemical Engineering Louisiana State University Baton Rouge, LA 70803		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62720A.3E162720A835.00.074
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Medical Research & Development Command Fort Detrick, Frederick, MD 21701		12. REPORT DATE May 1982
		13. NUMBER OF PAGES 184
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Final Principal Investigator--Arthur M. Sterling		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Models Ozonation Water Reuse Ultrafiltration Hyperchlorination Reverse Osmosis Simulation		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) To aid in the development of the control strategy and the fault detection/fault isolation logic for a self-contained wastewater treatment system, a dynamic model of the system has been developed. The model integrates component models for water processing units into a process simulator to facilitate the simulation of a variety of configurations for water reuse processes. This report describes the simulator modules, the use of the simulator, and the details of input data preparation.		

Executive Summary

For support of field medical units, the U.S. Army is developing a self-contained wastewater treatment system to produce potable water for use within the unit. To aid in developing the control strategy and the fault detection/fault isolation logic, a dynamic model of the system has been developed.

The dynamic model consists of component models for an ultrafiltration unit, a tubular reverse osmosis unit, an ozonation contacting unit, and a hyperchlorination unit. These component models are incorporated into a process simulator to facilitate the simulation of a variety of configurations for water-reuse processes. In addition to the component models, the process simulator includes models for general purpose process elements such as mixed tanks, overflow tanks, pumps, stream splitters, and stream mixers, and general purpose control elements such as sensors, manipulators, binary controllers, ratio controllers, and PID controllers.

To estimate the coefficients in the individual component models, a Pattern Search strategy was used to minimize a cost function which penalized for model deviation from experimental data. Experimental data were obtained first from experiments that formed the basis of the design of a pilot plant version of a water-reuse system and then from the operation of this pilot plant.

To make the integrated model as flexible as possible, the plant configuration--the process units, as well as the origin and destination of every stream--is specified by input data. Details on the use of the simulator and the preparation of input data are included in this report.

Table of Contents

Executive Summary.	ii
Introduction	vii
1. Overview	1
2. Streams.	2
3. Process Equipment.	3
3.1 Mixed Tank.	5
3.2 Overflow Tank	6
3.3 Volumetric Pump	7
3.4 Stream Splitter	8
3.5 Stream Source	9
3.6 Stream Mixer.	10
3.7 Ultrafiltration Unit.	11
3.8 Tubular Reverse Osmosis Unit.	12
3.9 Gel Model Ultrafiltration Unit.	13
3.10 Reverse Osmosis Unit.	14
3.11 Ultraviolet/Ozonation Unit.	15
3.12 Hypochlorite Unit	16
3.13 Sink.	17
4. Control Elements	18
4.1 Sensor.	20
4.2 Manipulator	21
4.3 Binary Controller	22
4.4 Ratio Controller.	24
4.5 PID Controller.	25
5. Data Preparation	27
5.1 Model Parameters.	31
5.2 Stream Elements	38
5.3 Stream Names.	41
5.4 Configuration	44
5.5 Print Specifications.	57
5.6 Plot Specifications	62
5.7 Run Specifications.	65
5.8 Plot Control.	70
5.9 Off-Line Storage.	73
5.10 Old Value Retrieval	76
Listing of Source Program.	79
List of References	174

List of Figures

5.1	Data Deck	30
5.2	Listing for RMODPR for Reading Model Parameters	33, 34, 35
5.3	Sample Data for the *MP Data Segment	36
5.4	Output for Reading the Model Parameters	37
5.5	Listing of the *SE Data Segment	39
5.6	Output from the Data in Figure 5.5	40
5.7	Listing of the *SN Segment of the Data Deck	42
5.8	Output Corresponding to the Data in Figure 5.7	43
5.9	Example of the Output Generated in the Configuration Section	49
5.10	Simplified Flowsheet for the Equilization/Prescreening and Ultrafiltration Sections of the WPE	50
5.11	Configuration Data Segment for the Plant in Figure 5.10	51
5.12	Configuration Output for the Data in Figure 5.11	52, 53, 54
5.13	Stream Summary Generated for the Data in Figure 5.11	56
5.14	Typical print specification Segment of the Input Data Deck	58
5.15	Output of the Print Specification Data	59
5.16	Tabular Output Generated during the Simulation	60, 61
5.17	Typical Plot Specification Segment of the Input Data Deck	63
5.18	Output of the Plot Specification Data	64
5.19	Typical Run Control Data	66
5.20	Output from Run Control Section	67
5.21	Simulator Output of Material Balance Calculations	69
5.22	Typical Plot Control Segment of the Input Data Deck	71

(List of Figures Continued)

5.23	Second Plot Generated from the Data in Figure 5.19	72
5.24	Sample Input Data for Off-Line Storage	74
5.25	Simulator Output for the Data given in Figure 5.24	75
5.26	Sample Data to Retrieve the Values Saved by the Data in Figure 5.24	77
5.27	Simulator Output for the *OL Data given in Figure 5.26	78

List of Tables

5.1	Segment Definitions	29
5.2	Configuration Data Card #1	45
5.3	Significance of Streams	46
5.4	Definition of Equipment Parameters	47, 48

Introduction

The U.S. Army has a requirement to provide a mission-oriented medical treatment system which is designed and equipped to facilitate rapid establishment and disestablishment. The flexibility permits immediate response for a medical support unit to any tactical, environmental or geographical change. This system will provide a contamination-free and controlled environment in which medical, surgical and ancillary procedures, and other supporting functions can be performed.

The mobile medical treatment system is termed the MUST Medical Complex. Associated with the MUST Medical Complex is a Water and Waste Management Subsystem (WWMS)*. This subsystem is required to treat and dispose of, without degradation of the environment or danger to personal health, all toxic and contaminated waste materials generated within the functional areas of the Medical Complex. In addition to the waste treatment and disposal, the Water Processing Element (WPE) within the WWMS must be capable of producing potable water from a fresh or brackish water source and nonconsumptive reuse water from the MUST Medical Complex waste water effluent.

The objectives of this program were to:

1. Develop an integrated dynamic model describing the operational characteristics of the water processing element. Emphasis was placed on the reuse mode of operation utilizing the MUST hospital composite waste or the x-ray, laboratory, and OR composite waste. The methodology to apply the model to other configurations and other wastes was also developed.

*The WWMS was deleted from the MUST in 1978.

2. Develop a control/monitoring system for the operation of the WPE, using the dynamic model as the basis.
3. Develop a fault detection/fault isolation package for the WPE, using the dynamic model as the basis.

The first two of the objectives were accomplished, and were incorporated into a simulator program written for a digital computer.

The purpose of this report is to provide an overview of the dynamic process simulator, and to describe the procedures necessary to run the program.

MODELING AND SIMULATION OF WASTEWATER
REUSE SYSTEMS - DYNAMIC PROCESS SIMULATOR

1. Overview

The dynamic process simulator is a general purpose computer program. The configuration of the process is specified by the input data, along with the parameters that specify the characteristics of the individual items of equipment in the process. This permits a variety of process configurations to be investigated without requiring any changes in the program source code. While the structure of the simulator is very general, the program has been tailored to meet the needs of the MUST WPE.

2. Streams

For purpose of the simulator, the process is represented by individual pieces of equipment interconnected by process streams. Associated with each stream is a stream vector that specifies the status of the stream. The number of elements in each stream vector is n_s .

The first element in the stream vector is always the flow rate. The remaining elements define the properties of the stream (composition, temperatures, etc.). As presently implemented for the MUST WPE, a stream vector consists of the following:

<u>Element</u>	<u>Definition</u>
1	Flow, m^3/hr
2	Concentration of suspended solids, g/m^3
3	Concentration of dissolved solids, g/m^3
4	Concentration of total organic carbon, (TOC), g/m^3

At some time in the future, it may be necessary to add temperature and/or pressure to this list.

Each stream in the process is designated by a unique number.

3. Process Equipment

For each unit, the following information must be entered:

1. Unit number
2. Equipment type
3. Input streams
4. Output streams
5. Parameters specifying equipment characteristics

For each unit two data cards must be entered. The first data card specifies the first four items in the above list; the second card specifies the parameters.

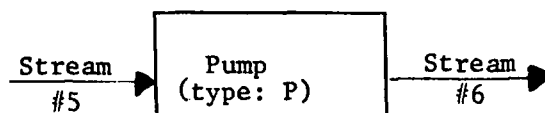
For each unit a maximum of five streams may be specified, with any mix of input and output streams. The data required are as follows:

1. Unit number
2. Equipment type
3. Stream 1
4. Stream 2
5. Stream 3
6. Stream 4
7. Stream 5

The data are entered using FORMAT (1X,I4,3X,A2,5I5).

For example, the unit

Unit #2



would be specified by the entry

2 P 5 -6

By convention, input streams are designated by positive stream numbers and output streams by negative stream numbers. Normally all input streams appear prior to output streams. For certain types of equipment, the order is significant.

The data parameters are also specific to each type of equipment.

The simulator presently recognizes the following types of equipment:

<u>Type</u>	<u>Code</u>
Mixed tank	MT
Overflow tank	ØT
Pump	P
Stream splitter	SP
Stream source	SØ
Stream mixer	SM
Ultrafiltration unit	UF
Gel model UF unit	GM
Reverse Osmosis unit	RØ
Tubular RO unit	TR
UV/oxination unit	UV
Hypochlorination unit	HC
Sink	SK

Other pieces of equipment can be easily added to the simulator.

3.1 Mixed Tank (MT)

The mixed tank is described by a total material balance and three component material balances. The total material balance is:

$$\frac{dV}{dt} = \sum_{\text{input}} F_i - \sum_{\text{output}} F_j$$

where V = volume of liquid in the tank, m^3

F_i = flow rate of input stream, m^3/hr

F_j = flow rate of output stream, m^3/hr

Each component material balance is:

$$\frac{d}{dt} (VC) = \sum_{\text{input}} F_i C_i - \sum_{\text{output}} F_j C_j$$

where C = concentration in the tank, g/m^3

C_i = concentration in input stream, g/m^3

C_j = concentration in output stream, g/m^3

Any combination of input and output streams are permitted. The concentration of all exit streams are the same as the concentration in the tank.

For the mixed tank, the following four parameters are required:

Par 1: Initial volume of liquid in the tank, m^3

Par 2: Initial concentration of suspended solids in the tank, g/m^3

Par 3: Initial concentration of dissolved solids in the tank, g/m^3

Par 4: Initial concentration of TOC in the tank, g/m^3

These parameters are entered via FORMAT (4F10.4).

In the initialization calculations for the mixed tank, the concentrations of all output streams are specified as the initial concentrations in the tank.

3.2 Overflow tank (OT)

The basic equations of the overflow tank are the same as for the mixed tank. However, the first stream must be the overflow stream and is specified as follows:

$$F_1 = F_T \text{ for } 0 < V < V_{\max}$$

$$F_1 = \sum \text{input } F_i - \sum \text{output } F_j \text{ for } V = 0 \text{ or } V = V_{\max}$$

where

F_1 = overflow stream flow rate, m^3/hr

F_T = design overflow-rate, m^3/hr

V = volume of liquid in tank, m^3

$V_{\max}$ = maximum volume of liquid in tank, m^3

F_i = flow rate of input stream, m^3/hr

F_j = flow rate of output stream, m^3/hr

Both F_T and $V_{\max}$ are data parameters.

The following six parameters are required for the overflow tank:

Par 1: Initial volume, m^3

Par 2: Initial concentration of suspended solids, g/m^3

Par 3: Initial concentration of dissolved solids, g/m^3

Par 4: Initial concentration of TOC, g/m^3

Par 5: Design overflow rate, m^3/hr

Par 6: Maximum volume, m^3

These parameters are entered according to FORMAT (6F10.4).

In the initialization calculations, the concentration of all output streams are set equal to the initial concentration in the tank. Also, the overflow-rate is set equal to F_T .

3.3 Volumetric Pump (P)

This piece of equipment represents either a constant volume pump or a variable volume pump followed by a flow controller. The first stream must be the input stream; the second must be the output stream.

The basic equation for this pump is as follows:

$$F_1 = F_2 = F_p$$

where F_p = flow rate through pump, m^3/hr

F_1 = flow rate of input stream, m^3/hr

F_2 = flow rate of output stream, m^3/hr

Furthermore, the concentration of the output stream is set equal to the concentration of the input stream.

The only input parameter is F_p , which is entered via FORMAT(F10.4).

The initialization calculations are the same as the normal calculations.

3.4 Stream Splitter (SP)

The purpose of this piece of equipment is to split an input stream into two output streams where the flow rate of one output stream is specified. The streams are specified in the following order:

1. Input stream
2. Output stream where flow is fixed
3. Output stream where flow varies

The flow equation describing this piece of equipment is as follows:

$$F_2 = \begin{cases} F_s & \text{if } F_1 \geq F_s \\ F_1 & \text{if } F_1 < F_s \end{cases}$$

$$F_3 = F_1 - F_2$$

where F_1 = flow rate of input stream, m^3/hr

F_2 = flow rate of first output stream, m^3/hr

F_3 = flow rate of second output stream, m^3/hr

F_s = the design split, m^3/hr

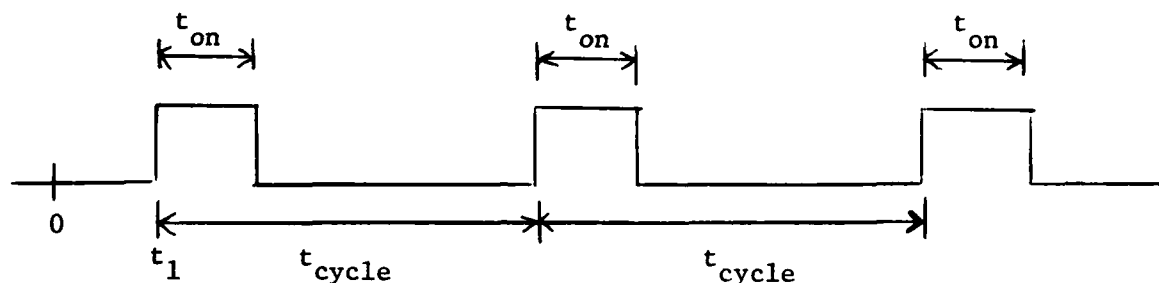
The concentrations in both output streams are set equal to the input concentration.

The only parameter required is F_s , which is entered via FORMAT (F10.4).

The initialization calculations are the same as the normal calculations.

3.5 Stream Source (S0)

The stream source is designed to simulate the pulse-type input streams that are encountered in the MUST hospital. A stream is assumed to behave as follows:



where t_1 = time of start of first pulse, hr.

t_{on} = time duration of pulse, hr.

t_{cycle} = time of cycle, hr

During the flow period, the flow rate and concentration must be specified.

For this unit, the following parameters are required:

Par 1: Time of first pulse, t_1 , hr

Par 2: Time duration of pulse, t_{on} , hr

Par 3: Time of cycle, t_{cycle} , hr

Par 4: Flow rate during pulse, m^3/hr

Par 5: Concentration of suspended solids, g/m^3

Par 6: Concentration of dissolved solid, g/m^3

Par 7: Concentration of TOC, g/m^3

These are entered via FORMAT (7F10.4)

The initialization calculations are the same as for the normal calculations.

All input streams must originate with a stream source block.

3.6 Stream Mixer (SM)

The stream mixer is designed to combine up to four input streams to produce a single output stream. In specifying the streams, the output stream must be specified last.

The flow equations describing this unit are as follows:

$$F_j = \sum_{\text{input}} F_i$$

where F_i = flow rate of input stream

F_j = flow rate of output stream

The equation for each concentration is as follows:

$$C_j = \frac{\sum_{\text{input}} F_i C_i}{F_j}$$

where C_i = concentration in input stream

C_j = concentration in output stream

No parameters are required for this unit.

The initialization calculations are the same as the normal calculations.

3.7 Ultrafiltration Unit (UF)

This piece of equipment consists of parallel banks of filtration tubes. The feed stream is forced into one end of the tube bundle, water and dissolved solids are forced through the tubes and collected as the filtered permeate. The concentrate stream is high in suspended solids, and is collected out of the other end of the bundle. In specifying the streams, the permeate stream must be first, the concentrate stream must be second, and the feed stream must be third.

See Starks and Smith (1) and Starks (3) for the model equations.

The parameters required for this unit are:

- Par 1: Number of tubes
- Par 2: Temperature of feed, °K
- Par 3: Pressure drop across the membrane at the inlet, atm
- Par 4: Pressure drop down the tubes, atm
- Par 5: Diameter of a tube, m
- Par 6: Length of a tube, m

These are entered via FORMAT (6F10.4).

The initialization calculations are the same as for the normal calculations.

3.8 Tubular Reverse Osmosis Unit (TR)

This piece of equipment is essentially the same as the ultrafiltration unit, except that the tube diameter is one or two orders of magnitude smaller, and the filtration membrane has smaller pores. These two differences lead to higher rejection of dissolved solids and TOC. See section 3.7 on the UF unit for a definition of the input and output streams.

The parameters required for this unit are the same as for the UF unit given in Section 3.7.

The initialization calculations are the same as for the normal calculations.

3.9 Gel Model Ultrafiltration Unit (GM)

This piece of equipment is essentially the same as the tubular RO unit, except that the boundary layer is assumed to form a gel on the inside surface. This gel increases the rejection of dissolved solids and TOC. See section 3.7 on the UF unit for a definition of the input and output streams.

The parameters required for this unit are the same as for the UF unit given in Section 3.7.

The initialization calculations are the same as for the normal calculations.

3.10 Reverse Osmosis Unit (RO)

This piece of equipment consists of a bundle of tube fibers around a central porous tube. The feed enters the central tube under very high pressure. The contaminated water flows radially around the outside of the fibers, forcing the water to permeate to the inside of the tubes; the fluid in the fibers is collected as the permeate stream. The fluid that reaches the outer region of the bundle is very high in concentration of dissolved solids and TOC and is collected as the concentrate stream. Suspended solids should not be present in the feed. The rejection is very high, but the flow rate is very small. The feed to the RO unit should be the permeate of an ultrafiltration unit, UF, TR, or GM. In specifying the streams, the permeate must be first, the concentrate must be second, the feed must be third.

See Starks and Smith (2) and Starks (3) for the model equations.

The parameters required for this unit are:

- Par 1: Pressure drop across the membrane (fiber), atm
- Par 2: Temperature of the feed, °K
- Par 3: Length of the fibers, m
- Par 4: Outer radius of bundle, m
- Par 5: Inner radius of bundle, m (center tube outer radius)
- Par 6: Diameter of a fiber, m

These are entered via FORMAT (6F10.4).

The initialization calculations are the same as for the normal calculations.

3.11 Ultraviolet/Ozonation Unit (UV)

This piece of equipment is used to oxidize organic compounds by ultraviolet activate ozonation. The contaminated water is simultaneously sparged with ozone and irradiated with ultraviolet light. The unit consists of a precontactor and six contact stages in series. There is one input stream and one output stream. In specifying the streams, the feed stream must be specified first and the effluent (output) stream must be specified second.

See Starks (3) for the model equations.

The parameters required for this unit are:

- Par 1: Initial concentration of suspended solids, g/m^3
- Par 2: Initial concentration of dissolved solids, g/m^3
- Par 3: Initial concentration of TOC, g/m^3
- Par 4: Inlet O_3 concentration, moles $\text{O}_3/\text{mole gas}$
- Par 5: Volumetric gas flow rate, m^3/hr
- Par 6: Precontactor flag (0 = No precontactor)
- Par 7: Number of stages
- Par 8: Area of a contactor, m^2
- Par 9: Area of a precontactor, m^2
- Par 10: Height of a stage, m
- Par 11: Feed temperature, $^{\circ}\text{K}$
- Par 12: Operating pressure, atm

These are entered via FORMAT (12F10.4). Note that two parameter cards are necessary (1-8 on the first, 9-12 on the second).

The initialization calculations are the same as for the normal calculations.

3.12 Hypochlorite Unit (HC)

This piece of equipment is a stirred tank in which calcium hypochlorite $\text{Ca}(\text{O Cl})_2$ is added to serve as a bacteria retardant. The feed stream must be specified first and the effluent stream must be specified second.

See Starks (3) for the model equations.

The parameters for this unit are:

- Par 1: pH of the output
- Par 2: Initial chlorite in the HC unit
- Par 3: Initial concentration of suspended solids, g/m^3
- Par 4: Initial concentration of dissolved solids, g/m^3
- Par 5: Initial concentration of TOC, g/m^3
- Par 6: Feed rate of calcium hypochlorite, m^3/hr
- Par 7: Volume of HC unit, m^3
- Par 8: Concentration of $\text{Ca}(\text{O Cl})_2$, g/m^3

These are entered via FORMAT (8F10.4).

Initialization calculations assume that the hypochlorination tank is filled up to its maximum volume at the concentration of the feed stream.

3.13 Sink (SK)

As indicated earlier, all input streams to the process must originate with the source block. Similarly, all streams leaving the process must be inputs to a block called the "Sink". One reason for this is that each stream must originate with one and only one block and each stream must terminate with one and only one block.

Actually, the sink block plays no role in the simulation. The only calculation procedure involved is that the collective stream flows (total and component) are integrated for the overall material balance calculations.

Each sink block may have up to five input streams. The process configuration may contain multiple sink blocks.

4. Control Elements

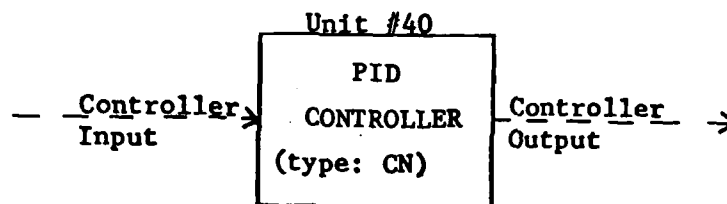
For each control element, the following information must be entered:

1. Unit number
2. Control element type
3. Parameters specifying control element characteristics

As for other units in the simulator, two data cards must be entered for each control element. The first data card specifies items one and two above; the second card specified the parameters.

Unlike the process equipment, there are no input or output streams to be specified. The first card is entered using FORMAT (1X,I4,3X,A2).

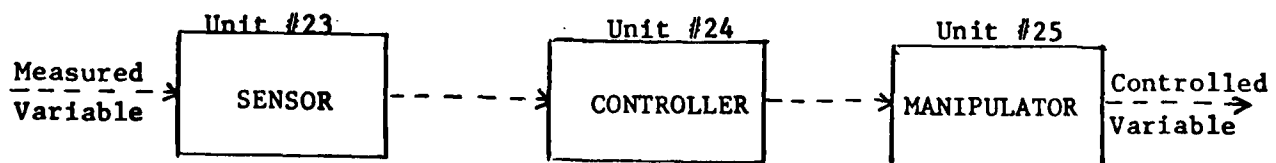
For example, the unit



would be specified by the entry

40 CN

The data parameters are specific to each type of control element. There are three basic control elements: the sensor, the manipulator, and the controller. A control scheme is defined by a unique group of sensor-controller-manipulator arrangement. The unit numbers for each control element in a control scheme must be sequential, starting with the sensor. For example:



The simulator presently recognizes the following types of control elements:

<u>TYPE</u>	<u>CODE</u>
Sensor	SN
Manipulator	MN
Binary Controller	BC
Ratio Controller	RC
PID Controller	CN

4.1 Sensor (SN)

The sensor is used to "measure" (return) the value of either a unit parameter (i.e. volume) or a stream element (flow rate or concentration).

The computational equation is:

$$c_n = \frac{1}{\tau} (r - c_{n-1}) \Delta t, \tau \neq 0$$

$$c_n = r, \tau = 0$$

where c_n is the output of the sensor (to the controller)

r is the "reading" or "measured value"

c_{n-1} is the previous output value

τ is the integration time constant, hr

Δt is the integration step size, hr

The parameters required are:

Par 1: Unit number or stream number

Par 2: Parameter or element number

Par 3: Initial output value

Par 4: The integration time constant, τ , hr

These parameters are entered via FORMAT (410.4). To specify a unit, enter a negative value; to specify a stream, enter a positive value.

The initialization calculations are the same as the normal calculations.

4.2 Manipulator (MN)

The manipulator is used to change the value of a parameter of a unit (i.e. increase a pump flow rate). The computational equation is:

$$y = \begin{cases} y_{\min}, & \text{if } m < y_{\min} \\ m, & \text{if } y_{\min} \leq m \leq y_{\max} \\ y_{\max}, & \text{if } m > y_{\max} \end{cases}$$

where y is the output value (new parameter value)

m is the signal from the controller

$y_{\min}$ is the lower limit of the parameter

$y_{\max}$ is the upper limit of the parameter

The parameters required are:

Par 1: Unit number of unit to manipulate (negative)

Par 2: Number of parameter to be manipulated

Par 3: Initial output value

Par 4: Upper limit

Par 5: Lower limit

These parameters are entered via FORMAT (5F10.4).

The initialization calculations are the same as the normal calculations.

4.3 Binary Controller (BC)

This control element represents a binary (on/off) controller. The computational equation is:

if automatic

$$m_n = \begin{cases} m_{\min}, & \text{if } c < c_{\min} \\ m_{n-1}, & \text{if } c_{\min} \leq c \leq c_{\max} \\ m_{\max}, & \text{if } c > c_{\max} \end{cases}$$

if manual

$$m_n = m_{\text{man}}$$

where m_n is the controller output

m_{n-1} is the previous value

$m_{\min}$ is the lower limit output

$m_{\max}$ is the upper limit output

c is the controller input (sensor output)

$c_{\min}$ is the lower setpoint

$c_{\max}$ is the upper setpoint

The parameters required are:

Par 1: Sensor unit number (negative)

Par 2: Manipulator unit number (negative)

Par 3: Lower setpoint value

Par 4: Upper setpoint value

Par 5: Initial output value

Par 6: Upper limit output value

Par 7: Lower limit output value

Par 8: Operation mode (negative = Automatic, positive = value for Manual)

These parameters are entered via FORMAT (8F10.4), note that two data cards are required.

The initialization calculations are the same as the normal calculations.

4.4 Ratio Controller (RC)

This control element represents a ratio controller. The computational equation is

$$m = \begin{cases} r \ c, & \text{if automatic} \\ m_{\text{man}}, & \text{if manual} \end{cases}$$

where m is the controller output value

r is the ratio (output/input)

c is the controller input (sensor output)

m_{man} is the output value for manual

The parameters required are

Par 1: Unit number of sensor (negative)

Par 2: Unit number of manipulator (negative)

Par 3: Ratio

Par 4: Operation mode (negative = Automatic, positive = value for Manual)

These parameters are entered via FORMAT (4F10.4).

The initialization calculations are the same as the normal calculations.

4.5 PID Controller (CN)

This control element represent a three mode, proportional-integral-derivative, controller. The computational equation is the velocity algorithm:

$$e_i = r - c_i$$

$$\frac{d e_i}{dt} = \frac{1}{\Delta t} (e_i - e_{i-1})$$

$$\frac{d^2 e_i}{dt^2} = \frac{1}{(\Delta t)^2} (e_i - 2 e_{i-1} + e_{i-2})$$

$$\Delta m_i = K_c \left(\frac{d e_i}{dt} + \frac{1}{T_I} e_i + T_D \frac{d^2 e_i}{dt^2} \right)$$

$$m_{i+1} = \begin{cases} m_i + \Delta m_i \Delta t, & \text{if automatic} \\ m_{\text{man}} & , \text{if manual} \end{cases}$$

where r is the set point value

c_i is the measured variable

e_i is the error in the signal

Δt is the integration step size

m_i is the controller signal

K_c is the controller gain

T_I is the reset time constant, hr

T_D is the derivative time constant, hr

m_{man} is the output value for manual

The parameters required are:

Par 1: Sensor unit number (negative)

Par 2: Manipulator unit number (negative)

Par 3: Setpoint

Par 4: Gain, K_c

Par 5: Reset time constant, hr

Par 6: Derivative time constant, hr

Par 7: Operation mode (negative = Automatic, positive = value for Manual)

These parameters are entered via FORMAT (7F10.4).

The initialization calculations are the same as the normal calculations.

5. Data Preparation

The input data cards provide for the entry of the following types of information.

1. Model parameters such as kinetic parameters, mass transfer coefficients, etc.
2. Specification of the plant itself, including
 - a. Configuration (equipment and streams)
 - b. Design parameters
 - c. Initial conditions
3. Descriptions, including
 - a. Equipment names
 - b. Names for equipment parameters
 - c. Stream names
 - d. Names for stream elements
4. Run control parameters (time of run, integration interval)
5. Variables to be printed, which may be either
 - a. Any element of any stream (e.g., flow rate of stream 5)
 - b. Any parameter associated with any unit (e.g., Volume of liquid in unit 8)
6. Plot control information, including
 - a. Variables to be plotted
 - b. Time of plot
 - c. Arrangement of graphs
7. Variables to be save on an offline storage medium, which may be either
 - a. Any element of any stream
 - b. Any parameter associated with any unit
8. Retrieval of variables from a previous run saved on offline storage.

As the data are entered, checks are made to assure that the data are meaningful.

To assist in the preparation of a data deck, data control cards are used to establish segments in the data deck. The first and last cards in each segment are data control cards. The data control card that appears at the beginning of the segment consists of an asterisk (*) in column 1 followed by up to three characters that identify the segment. The data control card at the end of the segment consists only of an asterisk (*) in column 1.

Table 5.1 lists the nine segments and specifies the data control cards for each. The general layout of the data deck is given in Figure 5.1. The following sections describe each segment in more detail.

Table 5.1. Segment Definitions

Segment	Data Control Card	Segment Required/Optional	Purpose of Segment
Model Parameters	*MP	Optional	Used to read any model parameters required for modules such as UF, RO, etc.
Stream Elements	*SE	Required	Reads engineering units and description for each stream element
Stream Names	*SN	Optional	Reads description for each stream
Configuration	*C	Required	Reads plant configuration
Print Definition	*PR	Required	Reads specifications for variables to be printed
Plot Definition	*PL	Optional	Reads specifications for variables to be plotted
Run	*RUN	Required	Reads execution control parameters and initiates simulation
Plot Control	*PC	Required when *PT used	Reads control information for generating plots
Off-line storage	*OF	Optional	Reads specifications for variables to be save on an off-line storage device (i.e. disk or tape)
Old values	*OL	Optional	To be used in place of RUN. To retrieve values of a previous run that were saved on an off-line storage device

```
*MP
    {model parameters}
*
*SE
    {stream element definitions}
*
*SN
    {stream descriptions}
*
*C
    {plant configuration}
*
*OF
    {off-line variable definitions}
*
*PR
    {print variable definitions}
*
*PL
    {plot variable definitions}
*
*PC
    {plot control information}
*
*OL
    {old value definitions}
*
*RUN
    {run time and integration size}
*
```

Figure 5.1. Data deck

5.1 Model Parameters (*MP)

For modules such as UF, RO, etc., the model parameters are determined by fitting the model to experimental data. Thus, it is necessary to provide a facility whereby these parameters can be easily changed.

Four options are available:

1. Within the subroutine for each respective module, use DATA statements to specify the model parameters. This is acceptable only if the primary subroutine does not call additional subroutines that require the model parameters.
2. Put all model parameters into a labeled COMMON statement, and then use DATA statements in the BLOCK DATA subprogram to initialize the parameters.
3. Put all model parameters into a labeled COMMON statement, and then read the values of the model parameters from data cards.
4. A combination of options 2 and 3. A set of model parameters are initialized in the BLOCK DATA subprogram. Different values may be entered through the use of NAMELIST statements.

The model parameter (*MP) segment of the data deck uses option 4.

Upon encountering the *MP data control card, the program calls subroutine RMODPR. This subroutine consists of statements such as

```
READ (5,NAMERO)
```

NAMERO is the name of a NAMELIST block for model parameters for the RO model. Any, all, or none of the NAMELIST parameters may be specified. These parameters are contained in labeled COMMON statements that also appear in the subroutines in which the values of the parameters are needed.

Parameters to be modified are specified as follows:

XX

&NAMEXX

A = x, B = y,

C = z

&END

where XX is the two character code of the desired module, i.e. UF, RO

A,B,C, etc. are the names of the parameters of XX to be changed

x,y,z, etc. are the corresponding numeric values

The user of the simulator can readily add parameters, delete parameters, and make any other changes required to meet the needs of the respective module for which the parameters are being read. Figure 5.2 gives the listing for RMODPR for reading the parameters for the ultrafiltration module only. Figure 5.3 lists the *MP segment of the data deck, and Figure 5.4 gives the corresponding output listing. The parameters are written from RMODPR so that errors may readily be detected, and so that outputs of runs using different values may be identified.

If no model parameters are to be changed from the default values, then the *MP section may be omitted. In this case, no parameters are changed or printed out. If it is desired to have a list of all of the default values, then enter

*MP

NONE

*

```

SUBROUTINE RMODPR
C
C READ MODEL PARAMETERS
C
      REAL MWHOCL, MWOCL, L, NF, NTPIDT, KLA, KHENRY, KRATE,
&      KDCOMP
      REAL*8 VHC, ALPHC, RDHC, KEQHC, CAOCL2, DTHC
C  LABELED COMMON STATEMENTS FOR REVERSE-OSMOSIS UNITS
      COMMON /REVOS1/ L, FLOW, RO, RI, DR, DP, DELP, RHOB,
&      TOLMX, TOLMN
      COMMON /ROFIT / AKA, AKC, ERE, APIRO, BPIRO, GAMARO,
&      BRO, CRO, NF, ROKE
      COMMON /GPPARM/ TEMP, VISC, MCNT2, MCNT3, JWRITE
      COMMON /REVOS2/ KWRITE, NSTEPS
C  LABELED COMMON STATEMENTS FOR OZONE UNITS
      COMMON /STAGES/ NSTAGE, PRECON
      COMMON /OZFIT/ KHENRY, ECOZ, ETOC, KRATE, KDCOMP,
&      EOZD, UVEFCT, ALPHA, EN, QPRIME
      COMMON /OZOPER/ CARPA, PAREA, UVH, UVRHO, UVPRES,
&      UVTEMP, NWRITE
      COMMON /GASLAW/ RGAS
C  LABELED COMMON STATEMENTS FOR ULTRAFILTRATION MODULE
      COMMON /UPPARM/ PLENUP, DTUBUP, NTUP, JPUFSS, JWUPSS
      COMMON /PARMUF/ TEMPUP, VISCUP, DENBUF, ZREOUF, DROPUP
      COMMON /UPSAV1/ NSTPUP
      COMMON /UPFIT / G1UP, G2UP, GINFUP, C1, C2, CINF
C  LABELED COMMON STATEMENTS FOR TUBULAR R-O MODULE
      COMMON /TRPARM/ PLENT, DTUBTR, NTTR, JPTRSS, JWTRSS
      COMMON /PARMTR/ TEMPTR, VISCTR, DENBTR, ZERO, DROPTR
      COMMON /TRFIT/ G1TR, G2TR, GINFTR, APITR, BTR, CTR,
&      DCXTR, ADAXTR, BDAXTR, CDAXTR
C  LABELED COMMON STATEMENTS FOR GEL-MODEL
      COMMON /GMPARM/ PLENGH, DTUBGH, NTGM, JPGMSS, JWGMSS
      COMMON /PARMGH/ TEMPGH, VISCGH, DENBGH, ZEROGH, DROPGH
      COMMON /GMFIT/ GAMMA, APIGH, BPIGH, BGM, CGH, RATIO,
&      DCXGH, ADAXGH, BDAXGH, CDAXGH, CAGEL
C  LABELED COMMON STATEMENTS FOR HYPOCHLORINATION MODULE
      COMMON /HCOPER/ VHC, ALPHC, RDHC, KEQHC, JWRTHC,
&      MCNTHC, CAOCL2, DTHC
      COMMON /HCSAV2/ MWHOCL, MWOCL, HCRHO
      DIMENSION JCARD(20), IUNIT(8)
      NAMELIST /NAMEO/ TOLMX, TOLMN, MCNT2, MCNT3, KWRITE,
&      AKA, AKC, ERE, APIRO, BPIRO, GAMARO,
&      BRO, CRO, NF, ROKE, JWRITE, NSTEPS
      NAMELIST /NAMEUV/ KHENRY, ECOZ, ETOC, KRATE, KDCOMP,
&      EOZD, UVEFCT, ALPHA, EN, QPRIME,
&      NWRITE, RGAS
      NAMELIST /NAMEUF/ JPUFSS, JWUPSS, G1UP, G2UP, GINFUP,
&      C1, C2, CINF, PERMIC
      NAMELIST /NAMETR/ JPTRSS, JWTRSS, G1TR, G2TR, GINFTR,
&      APITR, BTR, CTR, DCXTR, ADAXTR,
&      BDAXTR, CDAXTR
      NAMELIST /NAMEGM/ JPGMSS, JWGMSS, GAMMA, APIGH, BPIGH,

```

Figure 5.2. Listing for RMODPR for reading ultrafiltration model parameters

```

      &          BGM, CGM, RATIO, DCXGM, ADAXGM,
      &          BDAXGM, CDAXGM, CAGEL
      NAMELIST /NAMEHC/ VHC, ALPHC, RDHC, KEQHC, JWRTHC,
      &          MCNTHC, CAOCL2, MWHOCL, MWOCL, HCRHO
      DATA IUNIT/'NONE','*','UF','TR','GM','RO','UV','HC'/
10  READ(5,20) ICARD,JCARD
20  FORMAT(A4,T1,20A4)
      DO 30 I=1,8
      IF(ICARD.EQ.IUNIT(I))
      & GO TO (10, 110, 50, 60, 70, 80, 90, 100),I
C      NONE *   UF  TR  GM  RO  UV  HC
30  CONTINUE
      WRITE(6,40) JCARD
40  FORMAT('THE FOLLOWING CARD IS INVALID AND WILL BE',
      & ' IGNORED'/1X,20A4)
      GO TO 10
50  READ(5,NAMEUF)
      GO TO 10
60  READ(5,NAMETR)
      GO TO 10
70  READ(5,NAMEGM)
      GO TO 10
80  READ(5,NAMERO)
      GO TO 10
90  READ(5,NAMEUV)
      GO TO 10
100 READ(5,NAMEHC)
      GO TO 10
110 WRITE(6,120)
120 FORMAT('1*MP          MODEL PARAMETERS'/)
C
      WRITE(6,130)
130 FORMAT('OMODEL PARAMETERS FOR ULTRAFILTRATION MODULE')
      ICARD=0
      WRITE(6,140) JPUFSS, G1UF, G2UF, GINFUF, JWUFSS, C1,
      &          C2, CINF
140 FORMAT('0JPUFSS=',I3,'      G1UF=',G12.5,'      G2UF=',
      & G12.5,'      GINFUF=',G12.5/'      JWUFSS=',I3,6X,'      C1=',
      & '      C1=',G12.5,'      C2=',G12.5,'      CINF=',G12.5)
C
      WRITE(6,150)
150 FORMAT('OMODEL PARAMETERS FOR TUBULAR RO MODULE')
      WRITE(6,160) JPTRSS, G1TR, G2TR, GINFTR, JWTRSS, ADAXTR,
      &          BDAXTR, CDAXTR, APITR, BTR, CTR, DCXTR
160 FORMAT('0JPTRSS=',I3,'      G1TR=',G12.5,'      G2TR=',
      & G12.5,'      GINFTR=',G12.5/'      JWTRSS=',I3,'      ADAXTR=',
      & G12.5,'      BDAXTR=',G12.5,'      CDAXTR=',G12.5/14X,
      & '      APITR=',G12.5,'      BTR=',G12.5,'      CTR=',G12.5/
      & 14X,'      DCXTR=',G12.5)
C
      WRITE(6,170)
170 FORMAT('OMODEL PARAMETERS FOR GEL-MODEL')
      WRITE(6,180) JPGMSS, GAMMA, APIGM, BPIGM, JWGMSS, BGM,
      &          CGM, RATIO, DCXGM, ADAXGM, BDAXGM,

```

Figure 5.2. (continued)

```

&          CDAXGM, CAGEL
180 FORMAT('OJPGMSS=',I3,'  GAMMA=',G12.5,'  APIGM=',
& G12.5,'  BPIGM=',G12.5/'  JWGMSS=',I3,'  BGM=',
& G12.5,'  CGM=',G12.5,'  RATIO=',G12.5/14X,
& 'DCXGM=',G12.5,'  ADAXGM=',G12.5,'  BDAXGM=',G12.5/
& 12X,'  CDAXGM=',G12.5,'  CAGEL=',G12.5)

C
  WRITE(6,190)
190 FORMAT('OMODEL PARAMETERS FOR REVERSE OSMOSIS')
  WRITE(6,200) JWRITE, TOLMX, TOLMN, AKA, MCNT2, AKC,
&             ERE, APIRO, MCNT3, BPIRO, GAMARO, BRO,
&             NSTEPS, CRO, JWRITE, NF, ROKE
200 FORMAT('OJWRITE=',I3,'  TOLMX=',G12.5,'  TOLMN=',
& G12.5,'  AKA=',G12.5/'  MCNT2=',I3,'  AKC=',
& G12.5,'  ERE=',G12.5,'  APIRO=',G12.5/'  MCNT3=',
& I3,'  BPIRO=',G12.5,'  GAMARO=',G12.5,'  BRO=',
& G12.5/'  NSTEPS=',I3,'  CRO=',G12.5,'  NF=',
& G12.5,'  ROKE=',G12.5)

C
  WRITE(6,210)
210 FORMAT('OMODEL PARAMETERS FOR THE UV/OZONATION UNIT')
  WRITE(6,220) NWRITE, KHENRY, ECOZ, ETOC, KRATE,
&             KDCOMP, EOZD, UVEFCT, ALPHA, EN, QPRIME,
&             RGAS
220 FORMAT('ONWRITE=',I3,'  KHENRY=',G12.5,'  ECOZ=',
& G12.5,'  ETOC=',G12.5/14X,'  KRATE=',G12.5,
& '  KDCOMP=',G12.5,'  EOZD=',G12.5/14X,'  UVEFCT=',
& G12.5,'  ALPHA=',G12.5,'  EN=',G12.5/14X,
& 'QPRIME=',G12.5,'  RGAS=',G12.5)

C
  WRITE(6,230)
230 FORMAT('OMODEL PARAMETERS FOR HYPOCHLORINATION UNIT')
  WRITE(6,240) JWRTHC, VHC, ALPHC, RDHC, MCNTHC, KEQHC,
&             CAOCL2, MWHOCL, MWOCL, HCRHO
240 FORMAT('OJWRTHC=',I3,'  VHC=',G12.5,'  ALPHC=',
& G12.5,'  RDHC=',G12.5/'  MCNTHC=',I3,'  KEQHC=',
& G12.5,'  CAOCL2=',G12.5,'  MWHOCL=',G12.5/14X,
& 'MWOCL=',G12.5,'  HCRHO=',G12.5)
  RETURN
  END

```

Figure 5.2. (continued)

```
*MP  
RO  
  &NAME RO  
  MCNT2=15,  
  MCNT3=15,  
  &END  
*
```

Figure 5.3. Sample data for the *MP data segment.

*MP MODEL PARAMETERS

MODEL PARAMETERS FOR ULTRAFILTRATION MODULE

JMUFSS= 0	G1UF= 71251.	G2UF= .40141	GINFUF= .14674E-03
JWUFSS= 0	C1= .10537	C2= 1.1185	CINF= .46156E-02
	VISCUF= .32740E-02	DENBUF= .10000E+07	

MODEL PARAMETERS FOR TUBULAR RO MODULE

JPTRSS= 0	G1TF= 19100.	G2TR= .57895	GINPTR= 14400.
J4TRSS= 0	ADAXTR= .30254E-06	BCAXTR= .48540E-05	CDAXTR= .61405
	APITR= .10391E-07	BTR= .39526E-04	CTR= .42112E-02
	DCITR= .48000E-01	VISCTR= .32740E-02	DENBTH= .10000E+07

MODEL PARAMETERS FOR GEL-MODEL

JPGHSS= 0	GAMMA= .0	APIGM= .0	BPIGM= .0
JWGHSS= 0	BGM= .0	CGM= .0	RATIO= .0
	DCXGM= .0	MAXGM= .0	BDXGM= .0
	CDAIGM= .0	CAGEL= .0	VISCGM= .32740E-02
	DENBGM= .10000E+07		

MODEL PARAMETERS FOR REVERSE OSMOSIS

JWRITE= 0	TOLRX= .50000E-01	TOLRW= .10000E-01	AKA= .21170E-01
NCNT2= 15	AKC= .57434E-03	ERE= 3.7910	APIRO= .25260E-06
NCNT3= 15	BPIRO= .68100E-04	GAMARO= 18.350	BRO= .24880E-05
NSTEPS= 10	CRO= .13510E-05	RF= .36940E+08	ROKE= .62950
KWRITE= 0	VISC= .32740E-02	RHOB= .10000E+07	

MODEL PARAMETERS FOR THE UV/OZONATION UNIT

NWRITE= 0	KHENRY= .28560E+07	ECOZ= 1.0000	ETOC= 4.1250
	KRATE= 931.24	KDCOMP= .0	EOZD= 1.0000
	UVEFCT= .0	ALPHA= .16667E-03	EN= .0
	QPRIME= 16830.	RGAS= .82050	UVRHO= 55.556

MODEL PARAMETERS FOR HYPOCHLOBINATION UNIT

JWTHC= 0	ALPHC= .0	RDHC= .0	RCHRO= .10000E+07
NCNTHC= 30	KEQHC= .27000D-07	CAOCL2= .0	NWHOCL= 52500.
	NWOCL= 52500.		

Figure 5.4. Output for reading of the ultrafiltration model parameters

5.2 Stream Elements (*SE)

The purpose of this data segment is to read the engineering units and the descriptions of all elements of the stream vector.

The information for each stream element is punched on a card as follows:

<u>Quantity</u>	<u>Columns</u>	<u>Format</u>
Stream element number	2-5	I4
Engineering units	7-10	A4
Description	16-35	5A4

The cards must be entered in ascending order of the stream element numbers, and a card must be provided for each stream element.

Figure 5.5 lists the *SE segment of the data deck for defining three stream elements. Figure 5.6 gives the corresponding output.

```
*SE
  1 M3/H      FLOW RATE
  2 G/M3      SUSPENDED SOLIDS
  3 G/M3      DISSOLVED SOLIDS
  4 G/M3      TOT. ORG CARBON
*
```

Figure 5.5. Listing of the *SE data segment.

*SE STREAM ELEMENT DEFINITIONS

ELEMENT	UNITS	DESCRIPTION
1	M3/H	FLOW RATE
2	G/M3	SUSPENDED SOLIDS
3	G/M3	DISSOLVED SOLIDS
4	G/M3	TOT. ORG CARBON

Figure 5.6. Output from the data in Figure 5.5.

5.3 Stream Names (*SN)

The simulator provides for the association of a twenty-character descriptor with each process stream.

The descriptors are entered in the *SN data segment. The format of each card is as follows:

<u>Quantity</u>	<u>Columns</u>	<u>Format</u>
Stream number	2-5	I4
Description	11-30	5A4

The stream numbers may be entered in any order. It is not necessary that a descriptor be entered for each process stream.

Figure 5.7 lists the *SN data segment in which descriptions are provided for ten process streams. The corresponding output is given in Figure 5.8.

In the output generated from other sections of the simulator, the stream descriptor normally accompanies the stream number whenever sufficient space is available.

*SN
1 LABORATORY WASTE
2 X-RAY WASTE
3 BAD ACTOR FLOW
4 OPERATING ROOM WASTE
5 EQUILIZATION TK DISC
6 EQ TK PUMP DISCHARGE
7 RECYCLE FROM UF
8 UF FEED PUMP SUCTION
9 UF MODULE FEED FLOW
10 UF PERMEATE FLOW
*

Figure 5.7. Listing of the *SN segment of the data deck.

*SN	STREAM NAMES
STREAM	DESCRIPTION
1	LABORATORY WASTE
2	X-RAY WASTE
3	BAD ACTOR FLOW
4	OPERATING ROOM WASTE
5	EQUILIZATION TK DISC
6	EQ TK PUMP DISCHARGE
7	RECYCLE FROM UF
8	UF FEED PUMP SUCTION
9	UF MODULE FEED FLOW
10	UF PERMEATE FLOW

Figure 5.8. Output corresponding to the data in Figure 5.7.

5.4 Configuration (*C)

The plant configuration is specified by the information contained in the input data deck.

For each unit in the plant, two data cards are entered. The first card specifies the unit number, equipment type, stream connections, and the descriptor for that item of equipment. The second card specifies the physical parameters for the respective item of equipment*

Table 5.2 gives the format for the first card. The format for the second card is always 8F10.0. The significance of the streams depends upon the equipment type, and is summarized in Table 5.3. Table 5.4 summarizes the data input specifications for the basic equipment types provided by the simulator. Note: The second card is always required, even though no parameters are to be read for that respective piece of equipment.

In the stream specifications, a positive stream number designates an input stream; a negative stream number designates an output stream.

To illustrate, suppose the following specifications are given for a unit of equipment:

3	OT	1	2	-3	BAD ACTOR WASTE TANK
500.	1700.	50.	50.	756.	

The output listing generated in the configuration section is shown in Figure 5.9. Observe that descriptors accompany the stream numbers, and that engineering units accompany the parameters in the output listing.

For the plant whose flow sheet is given in Figure 5.10, Figure 5.11 lists the configuration data segment. Figure 5.12 provides the output for this configuration.

* some pieces of equipment require two parameter cards, refer to the specific sections in Chapter 3.

Table 5.2
Configuration Data Card #1

<u>Quantity</u>	<u>Columns</u>	<u>Format</u>
Equipment Number	2-5	I4
Equipment Type	9-10	A2
Stream #1	11-15	I5
Stream #2	16-20	I5
Stream #3	21-25	I5
Stream #4	26-30	I5
Stream #5	31-35	I5
Descriptor	41-60	5A4

Table 5.3

Significance of Streams

Equipment Type	Equipment Mnemonic	Stream #1	Stream #2	Stream #3	Stream #4	Stream #5
Mixed Tank	MT	I or 0	I or 0	I or 0	I or 0	I or 0
Overflow Tank	OT	Overflow Stream (0)	I or 0	I or 0	I or 0	I or 0
Pump	P	I	0	U	U	U
Stream Splitter	SP	I	Fixed Flow (0)	0	U	U
Stream Source	SO	0 Output	U	U	U	U
Stream Mixer	SM	Output stream must be specified last				
Sink	SK	I	I	I	I	I
Ultrafiltration	UF	Permeate (0)	Concentrate (0)	Feed (I)		U
Gel model UF	GM	Permeate (0)	Concentrate (0)	Feed (I)	U	U
Reverse Osmosis	RO	Permeate (0)	Concentrate (0)	Feed (I)	U	U
Tubular RO	TR	Permeate (0)	Concentrate (0)	Feed (I)	U	U
UV/Ozonation	UV	I	0	U	U	U
Hypochlorination	HC	I	0	U	U	U

I = Input, 0 = Output, U = Unused

Table 5.4

Definition of Equipment Parameters		
<u>Equipment</u>	<u>Parameter</u>	<u>Quantity</u>
MT	1	Initial volume in tank, m^3
	2	Initial suspended solids concentration, g/m^3
	3	Initial dissolved solids concentration, g/m^3
	4	Initial TOC concentration, g/m^3
OT	1	Initial volume in tank, m^3
	2	Initial suspended solids concentration, g/m^3
	3	Initial dissolved solids concentration, g/m^3
	4	Initial TOC concentration, g/m^3
	5	Design overflow, m^3/hr
	6	Maximum volume, m^3
P	1	Pump flow, m^3/hr
SP	1	Flow rate of fixed stream, m^3/hr
SO	1	Time of first pulse, hr
	2	Time duration of pulse, hr
	3	Time of cycle, hr
	4	Flow rate during pulse, m^3/hr
	5	Suspended Solids concentration, g/m^3
	6	Dissolved solids concentration, g/m^3
	7	TOC concentration, g/m^3
SM	None	
SK	None	
UF, TR, GM	1	Number of Tubes
	2	Temperature $^{\circ}K$
	3	Pressure drop across membrane at inlet
	4	Pressure drop down tube, atm
	5	Tube diameter, m
	6	Tube Length, m
RO	1	Pressure drop across the membrane, atm
	2	Temperature of feed, $^{\circ}K$
	3	Length of fibers, m
	4	Outer radius of fiber bundle, m
	5	Inner radius of fiber bundle, m
	6	Fiber diameter, m

Table 5.4
(continued)

<u>Equipment</u>	<u>Parameter</u>	<u>Quantity</u>
RO	6	Fiber diameter 3, m
UV	1	Initial suspended solids concentration, g/m^3
	2	Initial dissolved solids concentration, g/m^3
	3	Initial TOC concentration, g/m^3
	4	Inlet gas phase ozone to air mass ratio
	5	Volumetric gas flow rate, m^3/hr
	6	Precontactor
	7	Number fo stages
	8	Contactor area, m^2
	9	Pre-contactor area, m^2
	10	Stage height, m
	11	Feed temperature, $^{\circ}\text{K}$
	12	Operating Pressure, atm
HC	1	pH of the output
	2	Initial $\text{Na}(\text{O Cl})_2$ in the HC unit
	3	Initial suspended solids concentration, g/m^3
	4	Initial dissolved solids concentration, g/m^3
	5	Initial TOC concentration, g/m^3
	6	Feed rate of $\text{Ca} (\text{O Cl})_2$, m^3/hr
	7	Volume, m^3
	8	$\text{Ca} (\text{O Cl})_2$ feed concentration, g/m^3

UNIT NO.	3	OVERFLOW TANK	BAD ACTOR WASTE TANK
INPUT	STREAMS		
	1	LABORATORY WASTE	
	2	X-RAY WASTE	
OUTPUT	STREAMS		
	3	BAD ACTOR FLOW	
INITIAL CONDITIONS			
	VOLUME OF TANK	.50000	CU.M
	SUSPENDED SOLIDS	160.00	G/M3
	DISSOLVED SOLIDS	1700.0	G/M3
	TOTAL ORG CARBON	263.73	G/M3
DESIGN PARAMETERS			
	DESIGN OVERFLOW RATE	.50000E-01	M3/H
	MAXIMUM VOLUME	.75600	CU.M

Figure 5.9. Example of the output generated in the configuration section.

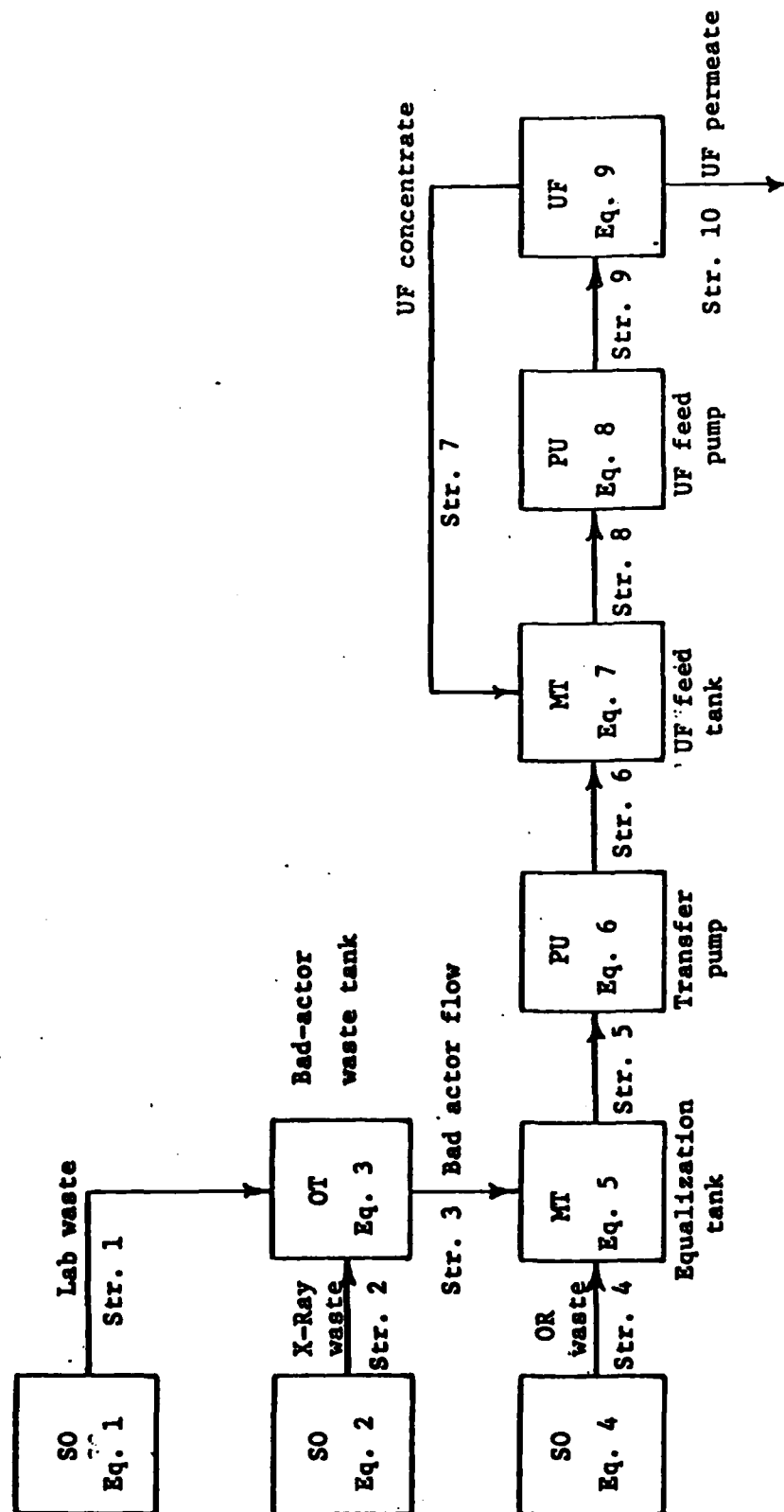


Figure 5.10. Simplified flowsheet for the equalization/prescreening and ultrafiltration sections of the WPE.

*C	1	SO	-1				LAB WASTE SOURCE
9.		7.		24.	.189	58.8	2151.
	2	SO	-2				X-RAY WASTE SOURCE
9.		8.		24.	.06615	43.	1247.
	3	OT	-3				BAD ACTOR WASTE TANK
.5		160.		1700.	263.73	.05	.756
	4	SO	-4				O. R. WASTE SOURCE
0.		.25		.75	.5292	2.	1788.
	5	MT	3	4 -5			EQUALIZATION TANK
2.5		160.		1700.	263.73		
	6	P	5	-6			EQ TANK PUMP
.270							
	7	MT	6	7 -8			UF FEED TANK
.5		160.		1700.	263.73		
	8	P	8	-9			UF FEED PUMP
2.							
	9	UF	-10	-7 9			UF MODULE
4.		311.1		3.4	2.04	.0254	20.49
9999		SK	10				

*

Figure 5.11. Configuration data segment for the plant in Figure 5.10.

*C PLANT CONFIGURATION

UNIT NO.	1	STREAM SOURCE	LAB WASTE SOURCE
INPUT	STREAMS		
	NONE		
OUTPUT	STREAMS		
	1	LABORATORY WASTE	
INITIAL CONDITIONS			
	NONE		
DESIGN PARAMETERS			
TIME OF FIRST PULSE	9.0000	HR	
PULSE DURATION	7.0000	HR	
PULSE CYCLE TIME	24.000	HR	
PULSE FLOW RATE	.18900	M3/H	
SUSPENDED SOLIDS	58.800	G/M3	
DISSOLVED SOLIDS	2151.0	G/M3	
TOTAL ORG CARBON	476.00	G/M3	
UNIT NO.	2	STREAM SOURCE	X-RAY WASTE SOURCE
INPUT	STREAMS		
	NONE		
OUTPUT	STREAMS		
	2	X-RAY WASTE	
INITIAL CONDITIONS			
	NONE		
DESIGN PARAMETERS			
TIME OF FIRST PULSE	9.0000	HR	
PULSE DURATION	8.0000	HR	
PULSE CYCLE TIME	24.000	HR	
PULSE FLOW RATE	.66150E-01	M3/H	
SUSPENDED SOLIDS	43.000	G/M3	
DISSOLVED SOLIDS	1247.0	G/M3	
TOTAL ORG CARBON	126.00	G/M3	
UNIT NO.	3	OVERFLOW TANK	BAD ACTOR WASTE TANK
INPUT	STREAMS		
	1	LABORATORY WASTE	
	2	X-RAY WASTE	
OUTPUT	STREAMS		
	3	BAD ACTOR FLOW	
INITIAL CONDITIONS			
VOLUME OF TANK	.50000	CU.M	
SUSPENDED SOLIDS	160.00	G/M3	
DISSOLVED SOLIDS	1700.0	G/M3	
TOTAL ORG CARBON	263.73	G/M3	
DESIGN PARAMETERS			
DESIGN OVERFLOW RATE	.50000E-01	M3/H	
MAXIMUM VOLUME	.75600	CU.M	

Figure 5.12. Configuration output for the data in Figure 5.11.

UNIT NO.	4	STREAM SOURCE	O. R. WASTE SOURCE
INPUT STREAMS	NONE		
OUTPUT STREAMS	4	OPERATING ROOM WASTE	
INITIAL CONDITIONS			
NONE			
DESIGN PARAMETERS			
	TIME OF FIRST PULSE	.0	HR
	PULSE DURATION	.25000	HR
	PULSE CYCLE TIME	.75000	HR
	PULSE FLOW RATE	.52920	M3/H
	SUSPENDED SOLIDS	2.0000	G/M3
	DISSOLVED SOLIDS	1788.0	G/M3
	TOTAL ORG CARBON	252.00	G/M3

UNIT NO.	5	MIXED TANK	EQUALIZATION TANK
INPUT STREAMS	3	BAD ACTOR FLOW	
	4	OPERATING ROOM WASTE	
OUTPUT STREAMS	5	EQUILIZATION TK DISC	
INITIAL CONDITIONS			
	VOLUME OF TANK	2.5000	CU.M
	SUSPENDED SOLIDS	160.00	G/M3
	DISSOLVED SOLIDS	1700.0	G/M3
	TOTAL ORG CARBON	263.73	G/M3
DESIGN PARAMETERS			
NONE			

UNIT NO.	6	PUMP	EQ TANK PUMP
INPUT STREAMS	5	EQUILIZATION TK DISC	
OUTPUT STREAMS	6	EQ TK PUMP DISCHARGE	
INITIAL CONDITIONS			
NONE			
DESIGN PARAMETERS			
	PUMP FLOW RATE	.27000	M3/H

UNIT NO.	7	MIXED TANK	UF FEED TANK
INPUT STREAMS	6	EQ TK PUMP DISCHARGE	
	7	RECYCLE FROM UF	
OUTPUT STREAMS	8	UF FEED PUMP SUCTION	
INITIAL CONDITIONS			
	VOLUME OF TANK	2.5000	CU.M
	SUSPENDED SOLIDS	160.00	G/M3
	DISSOLVED SOLIDS	1700.0	G/M3
	TOTAL ORG CARBON	263.73	G/M3
DESIGN PARAMETERS			
NONE			

Figure 5.12. (continued)

UNIT NO. 8 PUMP UF FEED PUMP
 INPUT STREAMS
 8 UP FEED PUMP SUCTION
 OUTPUT STREAMS
 9 UP MODULE FEED FLOW
 INITIAL CONDITIONS
 NONE
 DESIGN PARAMETERS
 PUMP FLOW RATE 2.0000 M3/H

UNIT NO. 9 ULTRAFILTRATION UF MODULE
 INPUT STREAMS
 9 UP MODULE FEED FLOW
 OUTPUT STREAMS
 10 UP PERMEATE FLOW
 7 RECYCLE FROM UF
 INITIAL CONDITIONS
 NONE
 DESIGN PARAMETERS
 NUMBER OF TUBES 4.0000
 FEED TEMPERATURE 311.10 DEGK
 INLET DELTA P 3.4000 ATM
 DELTA P DOWN TUBE 1.3600 ATM
 TUBE DIAMETER .25400E-01 M
 TUBE LENGTH 20.490 M

UNIT NO. 9999 STREAM SINK
 INPUT STREAMS
 10 UP PERMEATE FLOW
 OUTPUT STREAMS
 NONE
 INITIAL CONDITIONS
 NONE
 DESIGN PARAMETERS
 NONE

Figure 5.12. (continued)

Upon completion of the entry of the plant configuration, a check is made to see that for each stream

a) one and only one source is specified, and

b) one and only one destination is specified.

When a duplicate source definition or a duplicate destination definition is encountered, messages are generated in the configuration output. Figure 5.13 gives the stream summary printed after all units have been specified.

If any errors are detected in either the configuration or in the parameters specified, the run is terminated. The errors in the configuration section should be corrected and the run repeated.

STREAM SUMMARY		
STREAM	SOURCE	DESTINATION
1	1	3
2	2	3
3	3	5
4	4	5
5	5	6
6	6	7
7	9	7
8	7	8
9	8	9
10	9	9999

Figure 5.13. Stream summary generated for the data in Figure 5.11.

5.5 Print Specifications (*PR)

The simulator will generate a tabular listing of up to ten process variables as a function of time. The output may be specified to be either of the following:

1. Any element of any stream, or
2. Any parameter associated with any unit of equipment.

At least one print specification must be made if an off-line storage specification is not made (See Section 5.6).

The specifications are entered, one per data card, as follows:

Col. 2-5: stream number or unit number, with the latter
being entered as a negative number,

Col. 6-10: element number or parameter number.

Figure 5.14 lists a typical print specification segment of the input data deck. Figure 5.15 gives the output of the specification data.

Observe that descriptors accompany all stream numbers, element numbers, equipment numbers, and parameter numbers. Figure 5.16 gives the tabular output generated during the execution of the simulation.

```

*PR
.4
    9    2
    9    3
   10    1
   10    2
   10    3
   -7    1
    5    2
    5    2
   -5    1
*

```

Figure 5.14. Typical print specification segment of the input data deck.

*PR PRINT SPECIFICATIONS

PRINT INTERVAL IS .40000 HRS

-----	STREAM	9	UF MODULE FEED FLOW
	ELEMENT	2	SUSPENDED SOLIDS
-----	STREAM	9	UF MODULE FEED FLOW
	ELEMENT	3	DISSOLVED SOLIDS
-----	STREAM	9	UF MODULE FEED FLOW
	ELEMENT	4	TOT. ORG CARBON
-----	STREAM	10	UF PERMEATE FLOW
	ELEMENT	1	FLOW RATE
-----	STREAM	10	UF PERMEATE FLOW
	ELEMENT	2	SUSPENDED SOLIDS
-----	STREAM	10	UF PERMEATE FLOW
	ELEMENT	3	DISSOLVED SOLIDS
+++++	UNIT	7	UF FEED TANK
	PARAMETER	1	VOLUME OF TANK
-----	STREAM	5	EQUILIZATION TK DISC
	ELEMENT	2	SUSPENDED SOLIDS
-----	STREAM	5	EQUILIZATION TK DISC
	ELEMENT	3	DISSOLVED SOLIDS
+++++	UNIT	5	EQUALIZATION TANK
	PARAMETER	1	VOLUME OF TANK

Figure 5.15. Output of the print specification data.

TIME	STRM 9 ELE. 2	STRM 9 ELE. 3	STPM 9 ELE. 4	STRM 10 ELE. 1	STRM 10 ELE. 2
.0	160.	.170E+04	264.	1.27	.0
.400	194.	.170E+04	294.	1.06	.0
.800	232.	.170E+04	332.	.902	.0
1.20	275.	.170E+04	376.	.788	.0
1.60	321.	.170E+04	426.	.700	.0
2.00	372.	.171E+04	481.	.629	.0
2.40	427.	.171E+04	542.	.572	.0
2.80	486.	.171E+04	606.	.524	.0
3.20	549.	.171E+04	675.	.483	.0
3.60	613.	.171E+04	745.	.448	.0
4.00	679.	.171E+04	816.	.418	.0
4.40	745.	.171E+04	887.	.392	.0
4.80	809.	.171E+04	955.	.369	.0
5.20	869.	.172E+04	.102E+04	.348	.0
5.60	925.	.172E+04	.108E+04	.330	.0
6.00	974.	.172E+04	.112E+04	.313	.0
6.40	.101E+04	.172E+04	.117E+04	.298	.0
6.80	.105E+04	.172E+04	.120E+04	.285	.0
7.20	.107E+04	.172E+04	.122E+04	.272	.0
7.60	.109E+04	.173E+04	.123E+04	.261	.0
8.00	.109E+04	.173E+04	.123E+04	.251	.0
8.40	.109E+04	.173E+04	.123E+04	.241	.0
8.80	.108E+04	.173E+04	.122E+04	.232	.0
9.20	.107E+04	.173E+04	.121E+04	.224	.0
9.60	.105E+04	.173E+04	.119E+04	.216	.0
10.0	.103E+04	.174E+04	.116E+04	.209	.0
10.4	.100E+04	.174E+04	.114E+04	.202	.0
10.8	972.	.174E+04	.111E+04	.196	.0
11.2	943.	.174E+04	.109E+04	.190	.0
11.6	914.	.174E+04	.106E+04	.184	.0
12.0	884.	.175E+04	.103E+04	.179	.0
12.4	854.	.175E+04	.100E+04	.174	.0
12.8	824.	.175E+04	978.	.169	.0
13.2	796.	.175E+04	954.	.165	.0
13.6	768.	.176E+04	930.	.161	.0
14.0	741.	.176E+04	909.	.156	.0
14.4	715.	.176E+04	888.	.153	.0
14.8	691.	.177E+04	869.	.149	.0
15.2	667.	.177E+04	851.	.146	.0
15.6	644.	.178E+04	833.	.142	.0
16.0	623.	.178E+04	817.	.139	.0
16.4	602.	.178E+04	801.	.136	.0
16.8	582.	.178E+04	786.	.133	.0
17.2	563.	.179E+04	771.	.130	.0
17.6	545.	.179E+04	757.	.128	.0
18.0	528.	.179E+04	744.	.125	.0
18.4	512.	.179E+04	731.	.123	.0
18.8	496.	.179E+04	719.	.120	.0
19.2	482.	.179E+04	707.	.118	.0
19.6	467.	.180E+04	696.	.116	.0
20.0	454.	.180E+04	685.	.114	.0

Figure 5.16. Tabular output generated during the simulation

STRM 10	UNIT 7	STRM 5	STRM 5	UNIT 5
ELE. 3	PAR. 1	ELE. 2	ELE. 3	PAR. 1
.170E+04	2.50	160.	.170E+04	2.50
.170E+04	2.15	152.	.170E+04	2.54
.170E+04	1.86	150.	.170E+04	2.48
.170E+04	1.63	144.	.171E+04	2.50
.170E+04	1.44	141.	.171E+04	2.46
.170E+04	1.28	137.	.171E+04	2.45
.170E+04	1.15	133.	.171E+04	2.44
.171E+04	1.04	130.	.171E+04	2.41
.171E+04	.947	125.	.171E+04	2.43
.171E+04	.869	124.	.171E+04	2.36
.171E+04	.804	117.	.172E+04	2.41
.171E+04	.750	118.	.172E+04	2.32
.171E+04	.706	112.	.172E+04	2.36
.171E+04	.670	112.	.172E+04	2.27
.171E+04	.643	106.	.173E+04	2.32
.172E+04	.622	107.	.173E+04	2.23
.172E+04	.608	101.	.173E+04	2.27
.172E+04	.599	101.	.173E+04	2.21
.172E+04	.595	96.7	.173E+04	2.23
.172E+04	.597	95.0	.173E+04	2.19
.172E+04	.602	92.4	.174E+04	2.18
.173E+04	.612	89.7	.174E+04	2.17
.173E+04	.625	88.3	.174E+04	2.14
.173E+04	.642	84.6	.174E+04	2.16
.173E+04	.662	83.7	.174E+04	2.09
.173E+04	.685	78.6	.175E+04	2.14
.173E+04	.711	78.5	.175E+04	2.05
.174E+04	.740	73.6	.175E+04	2.09
.174E+04	.771	73.5	.176E+04	2.00
.174E+04	.804	68.9	.176E+04	2.05
.174E+04	.839	68.8	.176E+04	1.96
.174E+04	.877	64.5	.176E+04	2.00
.175E+04	.916	63.4	.177E+04	2.01
.175E+04	.957	60.2	.178E+04	2.11
.175E+04	1.00	58.6	.179E+04	2.16
.176E+04	1.04	56.6	.179E+04	2.23
.176E+04	1.09	54.7	.180E+04	2.30
.177E+04	1.14	53.7	.180E+04	2.35
.177E+04	1.19	51.6	.180E+04	2.45
.177E+04	1.24	51.2	.181E+04	2.47
.178E+04	1.29	48.9	.181E+04	2.60
.178E+04	1.34	49.0	.181E+04	2.51
.178E+04	1.39	46.7	.181E+04	2.56
.178E+04	1.45	46.8	.181E+04	2.48
.179E+04	1.51	44.5	.181E+04	2.52
.179E+04	1.56	44.6	.181E+04	2.43
.179E+04	1.62	42.5	.180E+04	2.48
.179E+04	1.68	42.1	.180E+04	2.41
.179E+04	1.74	40.5	.180E+04	2.43
.179E+04	1.80	39.8	.180E+04	2.39
.179E+04	1.86	38.7	.180E+04	2.39

Figure 5.16. (continued)

5.6 Plot Specifications (*PL)

The simulator will generate plots of up to ten process variables as a function of time. The output may be specified to be either of the following:

1. Any element of any stream, or
2. Any parameter associated with any unit of equipment

If no plots are to be generated, the entire plot specification data segment should be omitted.

The first data card in the plot specification segment contains the time duration (in hours) for the plot in columns 1-10 (format is F10.0). The following cards contain the specifications entered one per card, as follows:

Col. 2-5: stream number or equipment number, with the latter
being entered as a negative number

Col. 6-10: element number or parameter number

Figure 5.17 lists a typical plot specification segment of the input data deck. Figure 5.18 gives the output of the specification data. The plots themselves will be presented in a later section.

The plots are not generated as the simulation is being executed, but instead the values are stored in arrays for plotting upon completion of the run.

```
*PL
20.
    -5    1
     9    2
     9    3
    10    1
    -7    1
*
```

Figure 5.17. Typical plot specification segment of the input data deck.

```

*PL      PLOT VARIABLES

      PLOT DURATION  20.00 HRS

+++++UNIT      5      EQUALIZATION TANK
      PARAMETER  1      VOLUME OF TANK
-----STREAM  9      UP MODULE FEED FLOW
      ELEMENT   2      SUSPENDED SOLIDS
-----STREAM  9      UP MODULE FEED FLOW
      ELEMENT   3      DISSOLVED SOLIDS
-----STREAM 10      UP PERMEATE FLOW
      ELEMENT   1      FLOW RATE
+++++UNIT      7      UP FEED TANK
      PARAMETER  1      VOLUME OF TANK

```

Figure 5.18. Output of the plot specification data.

5.7 Run Specifications (*RUN)

This section is used to specify to the simulator the total run time and the integration step size to use. These values are entered on a single card with FORMAT (2F10.0). Figure 5.19 lists a typical run specification segment of the data deck. Figure 5.20 gives the output generated by the *RUN section.

*RUN
20. .001
*

Figure 5.19. Typical run control data.

*RUN RUN TIME PARAMETERS
TOTAL TIME FOR RUN 20.00 HRS
INTEGRATION STEP SIZE .1000E-02 HRS

Figure 5.20. Output from run control section.

Upon completion of the simulation, the simulator prints a material balance showing

1. Initial inventory
2. Amount input (from sources)
3. Final inventory
4. Amount output (to sink)

A typical material balance output is given in Figure 5.19. Due to round-off errors, the material balance does not generally close exactly (except for plants with small configurations). However, large errors would indicate a problem somewhere in the coding.

*****MATERIAL BALANCE*****

	INITIAL INVENTORY	AMOUNT INPUT	TOTAL IN
VOLUME	5.50	5.42	10.9
SUSPENDED SOLIDS	880.	108.	988.
DISSOLVED SOLIDS	.935E+04	.990E+04	.192E+05
TOTAL ORGANIC CARBON	.145E+04	.160E+04	.305E+04

FINAL INVENTORY	AMOUNT OUTPUT	TOTAL OUT	DIFFERENCE
4.85	6.02	10.9	.480E-01
971.	.0	971.	16.4
.877E+04	.104E+05	.191E+05	132.
.218E+04	832.	.302E+04	31.7

Figure 5.19. Simulator Output of Material Balance Calculations.

5.8 Plot Control (*PC)

As indicated in an earlier section, a maximum of ten variables can be plotted. The simulator permits the output to be one plot containing all ten variables, to be ten plots containing one variable each, or to be any combination thereof.

For each variable to be plotted, a data card must be entered containing the following information:

<u>Quantity</u>	<u>Columns</u>	<u>Format</u>
Plot Character	1	A1
Zero-coordinate	6-15	F10.0
Maximum-coordinate	16-25	F10.0

The plot grid is 40 rows by 100 columns.

The existence of a blank card in the data deck indicates to the simulator that a plot is to be generated containing the variables for which the plot parameters have been given. To illustrate, a typical plot control data segment is illustrated in Figure 5-22. A total of four plots will be generated. For all plots except the second, a single variable will be plotted. On the second plot, two variables will be plotted. When more than one variable is to be plotted, the y-coordinates are always that of the first variable for that respective plot. Figure 5-23 illustrate the second plot generated from the data in Figure 5-22.

*PC		
L	2000.	2800.
S	5.	25.
D	2000.	10000.
P	0.	400.
L	200.	1400.
*		

Figure 5.22. Typical plot control segment of the input data deck.

PLOT PARAMETERS

S-----STREAM
ELEMENT 9 UP MODULE FEED FLOW YQ = 0.0
2 SUSPENDED SOLIDS YMAX = 1000.00

0-----STREAM
ELEMENT 9 UP MODULE FEED FLOW YQ = 1500.00
3 DISSOLVED SOLIDS YMAX = 2000.00

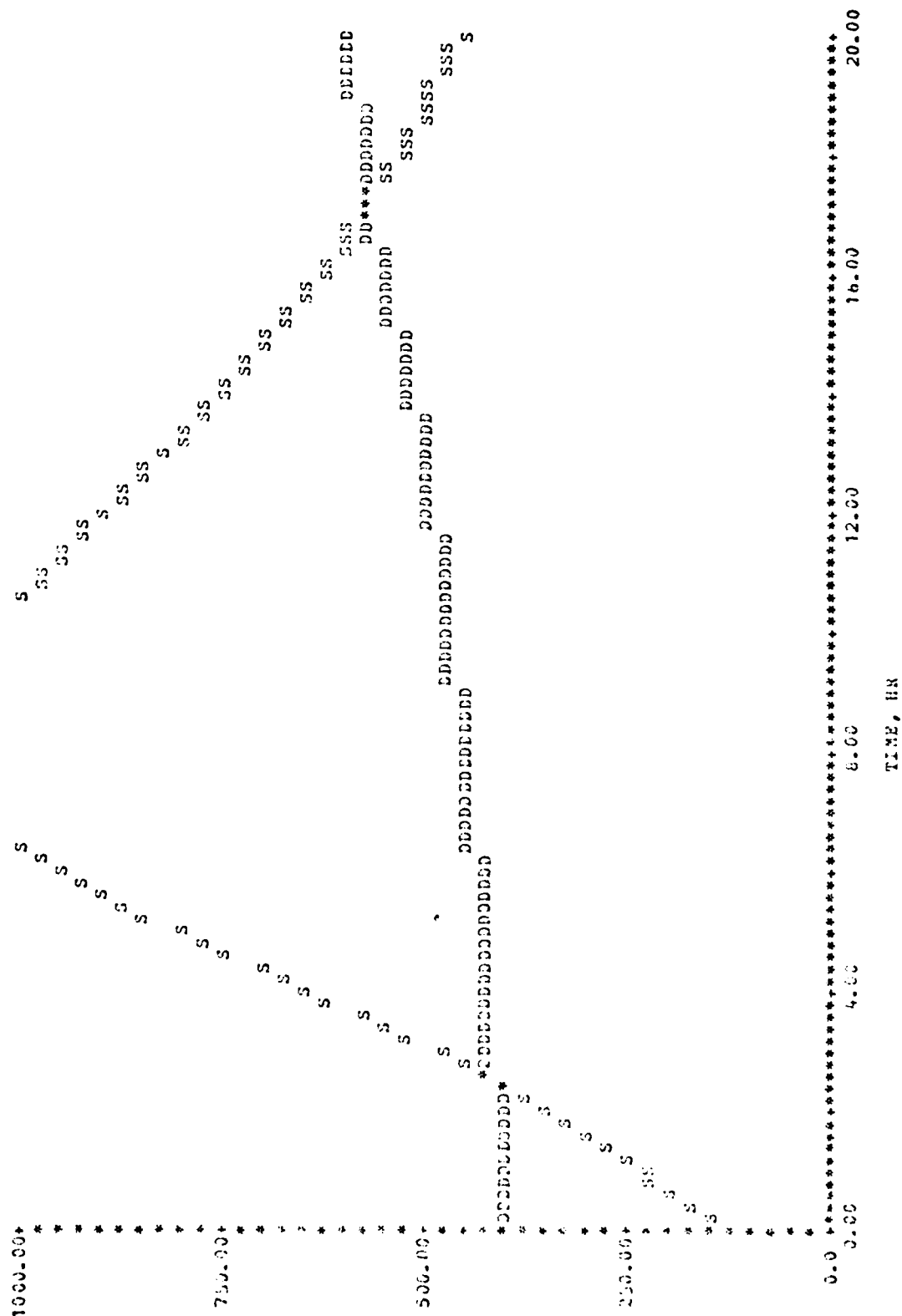


Figure 5.23. Second plot generated from the data in Figure 5.19.

5.9 Off-line Storage (*OF)

This section allows for the storage of the values of up to 50 process variables, which may be retrieved for printing and/or plotting at a latter date (see section 5.11). The values are written to FORTRAN logical unit number 8, which should be connected to a sequential dataset with a logical record length of one byte.

The first data card must be a unique message that uniquely identifies the particular run. The next data card is a flag that tells the simulator where on the dataset to store the values of the run.

There are two choices; FORMAT (I2):

- 1) -1 tells the simulator to start at the very beginning of the dataset (this is required if the dataset has never been used by the simulator)
- 2) +1 tells the simulator to start at the end of the last set of values on the dataset (in valid for a new dataset)

The third card is the save interval, i.e., the time interval between saves.

After the third card, the process variables to be saved are specified, one per card, just as in the print segment. See section 5.5 for what variables may be saved, and the format of each data card. Figure 5.24 lists a typical off-line specification segment of the input data deck. Figure 5.25 gives the output fo the off-line segment.

Note that this segment must be specified after the configuration section, and before the run section.

```

*OF
-1
RUN NUMBER ABC123
.4
    9    2
    9    3
    9    4
   10    1
   10    2
   10    3
   -7    1
    5    2
    5    3
   -5    1
*
```

Figure 5.24. Sample input data for Off-line Storage

```

*OP      OFF-LINE PARAMETER LIST
THE HEADER MESSAGE FOR THE DATA SET IS
RUN NUMBER ABC123
THE DATA SET WILL BE CLEARED, AND THE NEW VALUES SAVED FROM THE BEGINNING.
THE SAVE INTERVAL IS .4000
-----STREAM          9      UP MODULE FEED FLOW
      ELEMENT          2      SUSPENDED SOLIDS
-----STREAM          9      UP MODULE FEED FLOW
      ELEMENT          3      DISSOLVED SOLIDS
-----STREAM          9      UP MODULE FEED FLOW
      ELEMENT          4      TOT. ORG CARBON
-----STREAM          10     UP PERMEATE FLOW
      ELEMENT          1      FLOW RATE
-----STREAM          10     UP PERMEATE FLOW
      ELEMENT          2      SUSPENDED SOLIDS
-----STREAM          10     UP PERMEATE FLOW
      ELEMENT          3      DISSOLVED SOLIDS
+++++UNIT              7      UP FEED TANK
      PARAMETER          1      VOLUME OF TANK
-----STREAM          5      EQUILIZATION TK DISC
      ELEMENT          2      SUSPENDED SOLIDS
-----STREAM          5      EQUILIZATION TK DISC
      ELEMENT          3      DISSOLVED SOLIDS
+++++UNIT              5      EQUILIZATION TANK
      PARAMETER          1      VOLUME OF TANK

```

Figure 5.25. Simulator output for the data given in Figure 5.24.

5.10 Old Value Retrieval (*OL)

This segment indicates to the simulator that no simulation is to be performed, rather, values are to be read from a dataset on which values of a previous run have been stored by use of the off-line storage feature (see section 5.10). The dataset must be connected to FORTRAN logical unit number 8.

The only card in this section must contain the identification message of the data to be retrieved. This message must be identical to the message used in the off-line storage run. Figure 5.26 gives the corresponding *OL message for the *OF message of Figure 5.24. Figure 5.27 gives the corresponding output.

Note that *OL and *RUN are mutually exclusive. Also, the *SE, *SN, and *C sections must be specified again, exactly as in the original simulation run.

*OL
RUN NUMBER ABC123
*

Figure 5.26. Sample data to retrieve the values saved by the data in Figure 5.24.

*OL OLD VALUES SPECIFICATIONS
RUN NUMBER ABC123

Figure 5.27. Simulator output for the *OL data
given in Figure 5.26.

LISTING OF
SOURCE PROGRAM

```

      SUBROUTINE WPE
C   WPE  SIMULATOR
C
C   THIS IS THE MAIN ROUTINE FOR THE SIMULATION PACKAGE
C
      COMMON /LOOK/  ISW
      COMMON /MATDIS/ MATCAL
      COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
&          NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
      COMMON /CTIME/  TIME, FTIME, DT
      COMMON /CPLOT/  NPLOT, TPLOT, KPLOT, PLTDTA(100,10),
&          JPSTRM(2,10)
      COMMON /CPRINT/ IPRINT, NPELE, KPRINT(2,10)
      COMMON /COFLN/  NPL, LIST(2,50), LABLE, TOFLN,
&          MESSAG(20), IEOF, IHEAD
      COMMON /MATBAL/ BALNCE(4), AMTIN(4), AMTOUT(4)
      DIMENSION BALNO(4), TCTIN(4), TOTOUT(4), DIFF(4)
C   NPAR - NUMBER OF PARAMETERS FOR UNIT I
C           (INDEX INTO ARRAY PAR)
C   NCALL - SUBROUTINE CALCULATION SELECTION
C           -1 - READ PARAMETERS
C           0 - INITIALIZATION
C           1 - SIMULATE
C   NPELE - NUMBER OF PRINT ELEMENTS
C   NPLOT - NUMBER OF PLOT ELEMENTS
C   NOFLN- NUMBER OF OFF-LINE ELEMENTS
C   KPLOT - INDEX (2) INTO      ARRAY PLTDTA
C   STREAM(4,100) - STREAM VECTORS (FLOW,SS,TDS,TOC)
C   ICONFG(8,100) - CONFIGURATION ARRAY
C           (UNIT#,TYPE CODE,STRM1,....,STRM5,NPAR)
C   PAR(500) - PARAMETER VECTOR
C   KPRINT - STREAM/ELEMENT DESIGNATION OF PRINT OUTPUTS
C   JSCK(2,100) - STREAM CHECK ARRAY
C   JPSTRM(2,10) - STREAM/ELEMENT DESIGNATION OF PLOT OUTPUTS
C   PLTCRD(2,10) - YO/YMAX FOR PLOT OUTPUTS
C   PLTDTA(100,10) - PLOT DATA
      CALL ERESET(208,0,-1,1)
      DO 10 J=1,100
      DO 10 I=1,4
10  STREAM(I,J) = 0.0
      ISW= 0
      NFATER= 0
C   READ CONFIGURATION AND PARAMETERS
      CALL RDATA
      TIME= -DT
      DTPLOT= .01*TPLOT
      XPRINT= 0.0
      XOFLN= 0.0
      XPLOT= 0.0
      IF(NPL .LT. 0.) GO TO 60
C   GET INITIAL INVENTORY
      MATCAL= 0
      NCALL= -2

```

```

      DO 20 IUNIT= 1, NEQ
        NPAR= ICONFG(8,IUNIT)
20    CALL SUBCAL
      DO 30 I=1,4
        BALNO(I) = BALNCE(I)
30    BALNCE(I) = 0.
C    PERFORM INITIALIZATION CALCULATIONS
      NCALL= 0
40    IF(NPL .LT. 0.) GO TO 60
      DO 50 IUNIT=1,NEQ
        NPAR= ICONFG(8,IUNIT)
50    CALL SUBCAL
      IF(NFATER .GT. 10) STOP 45
C    PERFORM SIMULATION
      NCALL= 1
60    TIME= TIME + DT
      DTIME= TIME + .01*DT
      IF(NPL .EQ. 0) GO TO 80
      IF(DTIME .LT. XOFLN) GO TO 80
      XOFLN= XOFLN + TOFLN
      IF(NFL .GT. 0) CALL SAVEIT
      IF(NFL .LT. 0) CALL GETIT
80    IF(NPELE .EQ. 0) GO TO 90
      IF(DTIME .LT. XPRINT) GO TO 90
      XPRINT= XPRINT + TPRINT
      CALL PRINT
90    IF(NPLOT .EQ. 0) GO TO 100
      IF(DTIME .LT. XPLOT) GO TO 100
      XPLOT= XPLOT + DTPLOT
      CALL PLOT2
100   IF(DTIME .LT. FTIME) GO TO 40
      IF(NPL .LT. 0) GO TO 160
      IF(NPL .NE. 0) WRITE(8,110) IEOP
      IF(NPL .NE. 0) ENDFILE 8
110   FORMAT(A4)
C    GET FINAL INVENTORY
      MATCAL= 1
      NCALL= -2
      DO 120 IUNIT=1,NEQ
        NPAR= ICONFG(8,IUNIT)
120   CALL SUBCAL
      WRITE(6,130)
130   FORMAT('1*****MATERIAL BALANCE*****'/32X,'INITIAL',7X,
& 'AMOUNT',6X,'TOTAL',6X,'FINAL',8X,'AMOUNT',6X,'TOTAL'
& 31X,'INVENTORY',7X,'INPUT',8X,'IN',5X,'INVENTORY',6X,
& 'OUTPUT',7X,'OUT',4X,'DIFFERENCE')
      DO 140 I=1,4
        TOTIN(I) = AMTIN(I) + BALNO(I)
        TOTOUT(I) = AMTOUT(I) + BALNCE(I)
140   DIFF(I) = TOTIN(I) - TOTOUT(I)
      WRITE(6,150) (BALNO(I),AMTIN(I),TOTIN(I),BALNCE(I),
& AMTOUT(I),TOTOUT(I),DIFF(I),I=1,4)
150   FORMAT(6X,'VOLUME',T29,7G12.3/6X,'SUSPENDED SOLIDS',
& T29,7G12.3/6X,'DISSOLVED SOLIDS',T29,7G12.3/6X,

```

```
      E 'TOTAL ORGANIC CARBON',T29,7G12.3)
C MAKE PLOTS, IF NECESSARY
160 IF (NPLOT .EQ. 0) STOP
    CALL PLOT3
    RETURN
    END
```

```

BLOCK DATA
REAL*8 HC1
COMMON /CTIME/ TIME, FTIME, DT
COMMON /COFLN/ NPL, LIST(2,50), LABLE, TOFLN,
& MESSAG(20), IEOF, IHEAD
COMMON /CPLOT/ NPLOT, TPLOT, KPLOT, PLTDTA(100,10),
& JPSTRM(2,10)
COMMON /CPRINT/ TPRINT, NPELE, KPRINT(2,10)
COMMON /CREAD/ IFIRST, IAST, ICARD(20)
COMMON /MATBAL/ BALNCE(4), AMTIN(4), AMTOUT(4)
COMMON /NAMES/ IUNITS(12), NMSTRM(5,100),
& NMEQPT(5,100), NMELE(5,5), ISUNIT(5),
& NMPAR(6,75), IDNMPR(150)
COMMON /UPPARM/ UF1(2), IUF1(3)
COMMON /UPFIT/ UF2(6)
COMMON /PARMUF/ UF3(5)
COMMON /TRPARM/ TR1(2), ITR1(3)
COMMON /TRFIT/ TR2(10)
COMMON /PARMTR/ TR3(5)
COMMON /GMPARM/ GM1(2), IGM1(3)
COMMON /GMFIT/ GM2(11)
COMMON /PARMGH/ GM3(5)
COMMON /CHECK/ JSCK(2,100)
COMMON /UFSAV1/ UFSAVE
COMMON /PARMBO/ BO1(6), IRO1(2)
COMMON /ROFIT/ RO2(10)
COMMON /RCPARM/ RO3(4), IRO3(3)
COMMON /PIDSAV/ PIDSV(2)
COMMON /UVFIT/ UV1(10)
COMMON /UVPARM/ UV2(6), IUV
COMMON /STGSAV/ UV5(50)
COMMON /STAGES/ UV4(2)
COMMON /GASLAW/ RGAS
COMMON /HCPARM/ HC1(5), IHC1(2)
COMMON /HCSAV2/ HC2(3)
DIMENSION NMPR(6,11), NMUP(6,8), NMUV(6,7), NMRO(6,3),
& NMEXTRE(6,1), NMHC(6,3), NMPID(6,7),
& NMSN(6,4), NMHN(6,5), NMBIN(6,3)
EQUIVALENCE (NMPR,NMPAR(1,1)), (NMUP,NMPAR(1,12)),
& (NMUV,NMPAR(1,20)), (NMRO,NMPAR(1,27)),
& (NMEXTRE,NMPAR(1,30)), (NMHC,NMPAR(1,31)),
& (NMPID,NMPAR(1,34)), (NMSN,NMPAR(1,41)),
& (NMHN,NMPAR(1,45)), (NMBIN,NMPAR(1,50))
C DATA STATEMENTS FOR ULTRAFILTRATION
DATA IUF1 /3*0/, ITR1/3*0/
DATA UF2 /71251., .40141, .14674E-3, .10537, 1.1185,
C G1 G2 GINPC1 C2
& 0.46156E-2/
C CINP
DATA UF3 /0., .003274, 1.E6, 0., 0./
C TEMP VISC DENB DPZERO PDROP
C DATA STATEMENTS FOR TUBULAR RO
DATA TR2 /1.91E4, 0.57895, 1.44E4, 1.0391E-8,

```

```

C          GAM1      GAM2      GAMINF      API
E          3.9526E-5, 4.2112E-3, 4.8E-2, 3.0254E-7,
C          B          C          DCX          ADAX
E          4.854E-06, 0.61405/, IGM1/3*0/
C          BDAX      CDAX
C          DATA TR3 /0., .003274, 1.E6, 0., 0./
C          TEMP VISC      DENB DPZERO PDROP
C DATA STATEMENTS FOR GEL MODEL
C          DATA GM2 /11*0.0/
C          DATA GM3 /0., .003274, 1.E6, 0., 0./
C          TEMP VISC      DENB DPZERO PDROP
C DATA STATEMENTS FOR FIBER REVERSE OSMOSIS
C          DATA RO1 /0., 0., 0., 0., .05, .01/
C          L      RO RI DP TOLMX TOLMN
C          DATA IRO1 /0, 10/, IRO3/ 17, 22, 0/
C          NWRITE NSTEPS MCNT2 MCNT3 JWRITE
C          DATA RO2/0.02117, 5.7434E-4, 3.791, 2.526E-7, 6.81E-5,
C          AKA      AKC      ERE      API      BPI
E          18.35, 2.488E-6, 1.351E-6, 3.694E7, .6295/
C          GAMMA      B          C          NF      RATIO
C          DATA RO3 /0., .003274, 0., 1.E6/
C          TEMP VISC      DELP RHOB
C DATA STATEMENTS FOR UV/OZONATION
C          DATA UV1 /2.856E+6, 1.0, 4.125, 931.24, 0., 1.0,
C          KHENRY ECOZ ETOC KRATE KDCOMP EOZD
E          0., 1.6667E-4, 0.0, 16830./, IUUV/0/
C          UVEFCT RATIO EN QPRIME NWRITE
C          DATA UV2 / 0., 0., 0., 55.556, 0., 0./
C          CAREA PAREA H RHOB PRESS TEMP
C          DATA RGAS /.8205/
C DATA STATEMENTS FOR HYPOCHLORINATION
C          DATA HC1 /0.00, 0.00, 0.00, .27D-7, 0.00/
C          VOL ALPHA RE KEQ CAOCL2
C          DATA HC2 /52500.0, 52500.0, 1.E6/, IHC1/0, 30/
C          MWEOCL MWOCL RHO JWRITE MCNT
C          DATA NPL /0/, TOFLN/0./, IEOF/'*EOF'/, IHEAD/'HEAD'/
C          DATA NPLOT /0/, TPRINT/0./, NPELE/0/, IAST/'*'/
C          DATA NMSTRN/500*' '/, JSCK/200*0/, KPLOT/0/
C          DATA BALNCE,AMTIN,AMTOUT/4*0.0, 4*0.0, 4*0.0/
E          DATA IUNITS/'HB', 'M3/H', 'G/M3', 'CU.M', 'M', 'M2',
E          'G/M3', 'M2/H', 'DEGK', 'ATM', ' ', ' '
C MT
C          DATA IDNMPR/1, 4, 7, 5, 6, 30,
C OT
E          2, 6, 7, 5, 6, 30, 8, 9,
C P
E          3, 1, 10,
C SP
E          4, 1, 11,
C SO
E          5, 7, 1, 2, 3, 4, 5, 6, 30,
C SM
E          6, 0,
C UP

```

	&	7, 6, 12, 13, 14, 15, 16, 17,
C	RO	&
	&	8, 6, 27, 13, 17, 28, 29, 18,
C	UV	&
	&	9, 12, 5, 6, 30, 20, 21, 22, 23, 24, 25,
	&	26, 13, 27,
C	HC	&
	&	10, 8, 31, 32, 5, 6, 30, 33, 7, 19,
C	SK	&
	&	11, 0,
C	CN	&
	&	12, 7, 34, 35, 36, 37, 38, 39, 40,
C	SN	&
	&	13, 4, 41, 42, 43, 44,
C	MN	&
	&	14, 5, 45, 46, 47, 48, 49,
C	RC	&
	&	15, 5, 34, 35, 52, 40,
C	BC	&
	&	16, 8, 34, 35, 50, 51, 47, 48, 49, 40,
C	TR	&
	&	17, 6, 12, 13, 14, 15, 16, 17,
C	GM	&
	&	18, 6, 12, 13, 14, 15, 16, 17,
	&	23*0/

C	1	DATA NMPR	/'TIME', ' OF ', 'FIRS', 'T PU', 'LSE ', 1,
	&		'PULS', 'E DU', 'RATI', 'ON ', ' ', 1,
	&		'PULS', 'E CY', 'CLE ', 'TIME', ' ', 1,
	&		'PULS', 'E FL', 'OW R', 'ATE ', ' ', 2,
	&		'SUSP', 'ENDE', 'D SO', 'LIDS', ' ', 3,
	&		'DISS', 'OLVE', 'D SO', 'LIDS', ' ', 3,
	&		'VOLU', 'ME O', 'F TA', 'NK ', ' ', 4,
	&		'DESI', 'GN O', 'VERF', 'LOW ', 'RATE', 2,
	&		'MAXI', 'MUM ', 'VOLU', 'ME ', ' ', 4,
	&		'PUMP', ' FLO', 'W RA', 'TE ', ' ', 2,
	&		'FIXE', 'D FL', 'OW R', 'ATE ', ' ', 2/

C	12	DATA NMUF	/'NUMB', 'ER O', 'F TU', 'BES ', ' ', 11,
	&		'FEED', ' TEN', 'PERA', 'TURE', ' ', 9,
	&		'INLE', 'T DE', 'LTA ', 'P ', ' ', 10,
	&		'DELT', 'A P ', 'DOWN', ' TUB', 'E ', 10,
	&		'TUBE', ' DIA', 'METE', 'R ', ' ', 5,
	&		'TUBE', ' LEN', 'GTH ', ' ', ' ', 5,
	&		'FIBE', 'R DI', 'AMET', 'ER ', ' ', 5,
	&		'CAOC', 'L2 F', 'EED ', 'CONC', ' ', 3/

C	20	DATA NMUV	/'GAS ', 'OZON', 'E CO', 'NC. ', ' ', 11,
	&		'GAS ', 'FLOW', ' RAT', 'E ', ' ', 2,
	&		'PREC', 'ONTA', 'CTOR', ' ', ' ', 11,
	&		'NUMB', 'ER O', 'F ST', 'AGES', ' ', 11,
	&		'CONT', 'ACTO', 'R AR', 'EA ', ' ', 6,
	&		'PRE-', 'CONT', 'ACTO', 'R AR', 'EA ', 6,
	&		'HEIG', 'HT O', 'F A ', 'STAG', 'E ', 5/

```

C 27      DATA NMRO  /'OPER', 'ATIN', 'G PR', 'ESSU', 'RE  ',10,
&          'OUTE', 'R RA', 'DIUS', ' ', ' ', ' ',5,
&          'INNE', 'R RA', 'DIUS', ' ', ' ', ' ',5/

C 30      DATA NMEXTR/'TOTA', 'L OR', 'G CA', 'RBON', ' ', ' ',7/

C 31      DATA NMHC  /'PH  ', ' ', ' ', ' ', ' ', ' ',11,
&          'FREE', 'CHL', 'ORIN', 'E  ', ' ', ' ',12,
&          'CA(O', 'CL) 2', 'FE', 'ED R', 'ATE ',2/

C 34      DATA NMPID /'SENS', 'OR I', 'D  ', ' ', ' ', ' ',11,
&          'MANI', 'PULA', 'TOR ', 'ID ', ' ', ' ',11,
&          'SET ', 'POIN', 'T  ', ' ', ' ', ' ',11,
&          'GAIN', ' ', ' ', ' ', ' ', ' ',11,
&          'RESE', 'T TI', 'ME  ', ' ', ' ', ' ',1,
&          'RATE', 'TIM ', 'E   ', ' ', ' ', ' ',1,
&          'MODE', ' ', ' ', ' ', ' ', ' ',11/

C 41      DATA NMSN  /'UNIT', 'OR ', 'STRE', 'AM I', 'D  ',11,
&          'ITEM', 'OF ', 'INQU', 'IRY ', ' ',11,
&          'CURR', 'ENT ', 'READ', 'ING  ', ' ',11,
&          'TIME', 'CON', 'STAN', 'T   ', ' ',1/

C 45      DATA NMMN  /'UNIT', 'ID ', ' ', ' ', ' ', ' ',11,
&          'MANI', 'PULA', 'TED ', 'ITEM', ' ',11,
&          'CURR', 'END ', 'VALU', 'E   ', ' ',11,
&          'UPPE', 'R LI', 'MIT ', ' ', ' ', ' ',11,
&          'LOWE', 'R LI', 'MIT ', ' ', ' ', ' ',11/

C 50      DATA NMBIN /'LOW ', 'SET ', 'POIN', 'T  ', ' ', ' ',11,
&          'HI S', 'ET P', 'OINT', ' ', ' ', ' ',11,
&          'RATI', 'O  ', ' ', ' ', ' ', ' ',11/

      END

```

AD-A143 024

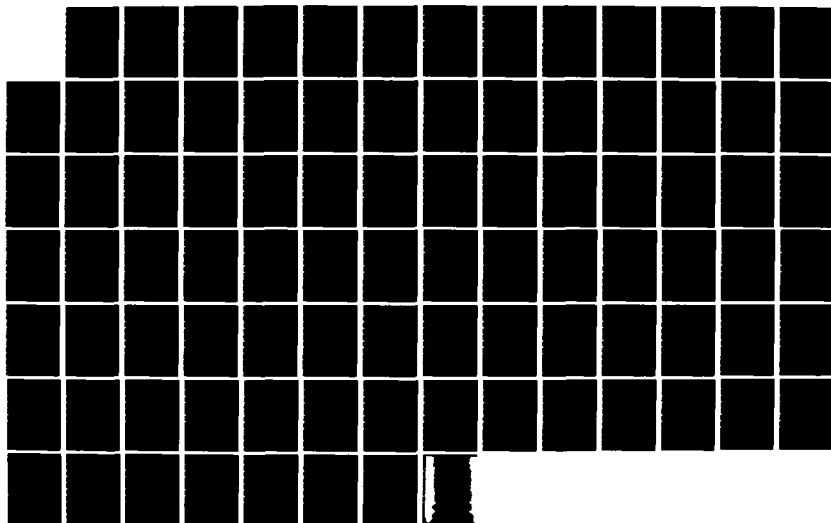
MODELING AND SIMULATION OF WASTEWATER REUSE SYSTEMS -
DYNAMIC PROCESS SIMULATOR(U) LOUISIANA STATE UNIV BATON
ROUGE DEPT OF CHEMICAL ENGINEERING. C L SMITH ET AL.
MAY 82 DAMD17-77-C-7040

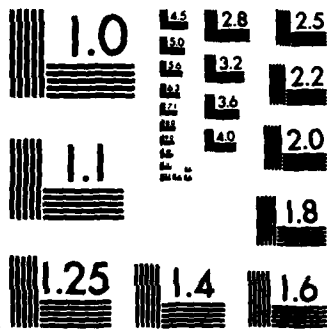
2/2

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

SUBROUTINE RDATA
C
C THIS SUBROUTINE READS ALL INPUT DATA
C
      LOGICAL OFF/.FALSE./
      COMMON /CREAD/ IFIRST, IAST, ICARD(20)
      COMMON /CTIME/ TIME, FTIME, DT
      EQUIVALENCE (ICARD1,ICARD(1))
      DATA IMP, ISE, ISN, IC, IOF, IOL, IPR, IPL, IRUN/
&      '*MP', '*SE', '*SN', '*C', '*OF', '*OL',
&      '*PR', '*PL', '*RUN'/'
C READ FIRST CARD. SHOULD BE A DATA CONTROL CARD
      CALL RCARD
C FIRST CARD SHOULD BE *MP OR *SE
      IF(ICARD1 .EQ. IMP) GO TO 20
      IF(ICARD1 .EQ. ISE) GO TO 40
      WRITE(6,10) IMP, ISE, ICARD
10  FORMAT('0*****EXPECTING ''',A4,''' OR ''',A4,
&  ''', FOUND '/' ''',20A4,'''')
      STOP
C FOUND *MP, READ MODEL PARAMETERS
20  CALL RMODPR
C RETURN FROM RMODPR OCCURS ONLY WHEN A DATA CONTROL CARD
C HAS BEEN ENCOUNTERED. THE NEXT CARD SHOULD BE *SE
      CALL RCARD
      IF(ICARD1 .EQ. ISE) GO TO 40
      WRITE(6,30) ISE, ICARD
30  FORMAT('0*****EXPECTING ''',A4,'''', FOUND'/'
&  ''',20A4,'''')
      STOP
C FOUND *SE, READ STREAM ELEMENT DEFINITIONS
40  CALL RSTRME
C RETURN FROM RSTRME OCCURS ONLY WHEN A DATA CONTROL CARD
C HAS BEEN ENCOUNTERED. THE NEXT CARD SHOULD BE *SN OR *C
      CALL RCARD
      IF(ICARD1 .EQ. ISN) GO TO 50
      IF(ICARD1 .EQ. IC) GO TO 60
      WRITE(6,10) ISN, IC, ICARD
      STOP
C FOUND *SN, READ DESCRIPTIONS FOR STREAMS
50  CALL RSTRMN
C RETURN FROM RSTRMN OCCURS ONLY WHEN A DATA CONTROL CARD
C HAS BEEN ENCOUNTERED. THE NEXT CARD SHOULD BE *C
      CALL RCARD
      IF(ICARD1 .EQ. IC) GO TO 60
      WRITE(6,30) IC, ICARD
      STOP
C FOUND *C, READ CONFIGURATION
60  CALL RCONPG
C RETURN FROM RCONPG IS ONLY WHEN A DATA CONTROL CARD HAS
C BEEN ENCOUNTERED. NEXT CARD SHOULD BE *OF OR *PR
      CALL RCARD
      IF(ICARD1 .EQ. IOF) GO TO 70

```

```

        IF(ICARD1 .EQ. IPR) GO TO 90
        WRITE(6,10) IOF, IPR, ICARD1
        STOP
C   FOUND *OF, READ OFF-LINE PARAMETER LIST
70  CALL OPLINE
    OFF= .TRUE.
C   RETURN FROM OPLINE IS ONLY WHEN A DATA CONTROL CARD HAS
C   BEEN ENCOUNTERED. NEXT CARD SHOULD BE *PR, *PL, OR *RUN
    CALL RCARD
    IF(ICARD1 .EQ. IPR) GO TO 90
    IF(ICARD1 .EQ. IPL) GO TO 100
    IF(ICARD1 .EQ. IRUN) GO TO 120
    WRITE(6,80) IPR, IPL, IRUN, ICARD
80  FORMAT('0*****EXPECTING ''',A4,''', ''',A4,''', OR ''',
& A4,''', FOUND'/' ''',20A4,'''')
    STOP
C   FOUND *PR, READ PRINT SPECIFICATIONS
90  CALL RPRINT
C   RETURN FROM RPRINT IS ONLY WHEN A DATA CONTROL CARD HAS
C   BEEN ENCOUNTERED. IT SHOULD BE *PL, *OL, OR *RUN
    CALL RCARD
    IF(ICARD1 .EQ. IPL) GO TO 100
    IF(ICARD1 .EQ. IOL) GO TO 110
    IF(ICARD1 .EQ. IRUN) GO TO 120
    WRITE(6,80) IPL, IOL, IRUN, ICARD
    STOP
C   FOUND *PL, READ PLOT SPECIFICATIONS
100 CALL PLOT1
C   RETURN FROM PLOT1 IS ONLY WHEN A DATA CONTROL CARD HAS
C   BEEN ENCOUNTERED. NEXT CARD SHOULD BE *OL OR *RUN
    CALL RCARD
    IF(ICARD1 .EQ. IOL) GO TO 110
    IF(ICARD1 .EQ. IRUN) GO TO 120
    WRITE(6,10) IOL, IRUN, ICARD
    STOP
C   FOUND *OL, LOCATE THE OLD VALUES
110 CALL OLDVAL
    GO TO 150
C   FOUND *RUN, READ RUN PARAMETERS
120 READ(5,130) FTIME, DT
130 FORMAT(2F10.0)
    IF(OFF) WRITE(8,140) FTIME, DT
140 FORMAT(A4)
150 WRITE(6,160) FTIME, DT
160 FORMAT('1*RUN      RUN TIME PARAMETERS'/
& 6X,'TOTAL TIME FOR RUN',G12.4,' HRS'/6X,
& 'INTEGRATION STEP SIZE',G12.4,' HRS')
    CALL RCARD
    RETURN
    END

```

SUBROUTINE RMCEPB

```

C
C READ MODEL PARAMETERS
C
      REAL MWHOCL, MWOCL, L, NF, NTPIDT, KLA, KHENRY, KRATE,
&      KDCOMP
      REAL*8 VHC, ALPHC, RDHC, KEQHC, CAOCL2, DTHC
C LABELED COMMON STATEMENTS FOR REVERSE-OSMOSIS UNITS
      COMMON /PARMO/ L, RO, RI, DF, TOLMX, TOLMN, KWRITE,
&      NSTEPS
      COMMON /ROFIT / AKA, AKC, ERE, APIRO, BPIRO, GAMARO,
&      BRO, CRO, NF, ROKE
      COMMON /ROPARM/ TEMP, VISC, DELP, RHOB, MCNT2, MCNT3,
&      JWRITE
C LABELED COMMON STATEMENTS FOR OZONE UNITS
      COMMON /UVFIT/ KHENRY, ECOZ, ETOC, KRATE, KDCOMP,
&      EOZD, UVEFCT, ALPHA, EN, QPRIME
      COMMON /UVPARM/ CAREA, PAREA, UVH, UVRHO, UVPRES,
&      UVTEMP, NWRITE
      COMMON /GASLAW/ RGAS
C LABELED COMMON STATEMENTS FOR ULTRAFILTRATION MODULE
      COMMON /UPARM/ PLENUF, DTUBUF, NTUF, JPUFSS, JWUFSS
      COMMON /PARMUF/ TEMPUF, VISCUF, DENBUF, ZREOUF, DROPUF
      COMMON /UFSAV1/ NSTPUF
      COMMON /UFFIT / G1UF, G2UF, GINFUF, C1, C2, CINF
C LABELED COMMON STATEMENTS FOR TUBULAR R-O MODULE
      COMMON /TRPARM/ PLENT, DTUBTR, NTTR, JPTRSS, JWTRSS
      COMMON /PARMTR/ TEMPTR, VISCTR, DENBTR, ZERO, DROPTR
      COMMON /TRFIT/ G1TR, G2TR, GINFTR, APITR, BTR, CTR,
&      DCXTR, ADAXTR, BDAXTR, CDAXTR
C LABELED COMMON STATEMENTS FOR GEL-MODEL
      COMMON /GMPARM/ LENGH, DTUBGM, NTGM, JPGHSS, JWGHSS
      COMMON /PARMGH/ TEMPGH, VISCGR, DENBGM, ZEROGM, DROPGM
      COMMON /GMFIT/ GAMMA, APIGM, BPIGM, BGM, CGM, RATIO,
&      DCXGM, ADAXGM, BDAXGM, CDAXGM, CAGEL
C LABELED COMMON STATEMENTS FOR HYPOCHLORINATION MODULE
      COMMON /HCPARM/ VHC, ALPHC, RDHC, KEQHC, CAOCL2,
&      JWTHC, MCNTHC
      COMMON /HCSAV2/ MWHOCL, MWOCL, HCRHO
      DIMENSION JCARD(20), IUNIT(8)
      NAMELIST /NAMERO/ TOLMX, TOLMN, MCNT2, MCNT3, KWRITE,
&      AKA, AKC, ERE, APIRO, BPIRO, GAMARO,
&      BRO, CRO, NF, ROKE, JWRITE, NSTEPS,
&      VISC, RHOB
      NAMELIST /NAMEUV/ KHENRY, ECOZ, ETOC, KRATE, KDCOMP,
&      EOZD, UVEFCT, ALPHA, EN, QPRIME,
&      NWRITE, RGAS, UVRHO
      NAMELIST /NAMEUF/ JPUFSS, JWUFSS, G1UF, G2UF, GINFUF,
&      C1, C2, CINF, VISCUF, DENBUF
      NAMELIST /NAMETR/ JPTRSS, JWTRSS, G1TR, G2TR, GINFTR,
&      APITR, BTR, CTR, DCXTR, ADAXTR,
&      BDAXTR, CDAXTR, VISCTR, DENBTR
      NAMELIST /NAMEGM/ JPGHSS, JWGHSS, GAMMA, APIGM, BPIGM,

```

```

      &          BGM, CGM, RATIO, DCXGM, ADAXGM,
      &          BDA XGM, CDA XGM, CAGEL, VISC GM, DENBGM
      NAMELIST /NAMEHC/ ALPHC, RDHC, REQHC, JWRTHC,
      &          MCNTHC, MWHOCL, MWOCL, HCRHO
      DATA IUNIT/'NONE','**','UF','TR','GM','RO','UV','HC'/
10 READ(5,20) ICARD,JCARD
20 FORMAT(A4,T1,20A4)
      DC 30 I=1,8
      IF(ICARD .EQ. IUNIT(I))
      & GO TO (10, 110, 50, 60, 70, 80, 90, 100),I
C      NONE *   UF  TR  GM  RO  UV   HC
30 CONTINUE
      WRITE(6,40) JCARD
40 FORMAT('THE FOLLOWING CARD IS INVALID AND WILL BE',
      & ' IGNORED'/1X,20A4)
      GO TO 10
50 READ(5,NAMEUF)
      GO TO 10
60 READ(5,NAMETR)
      GO TO 10
70 READ(5,NAMEGM)
      GO TO 10
80 READ(5,NAMERO)
      GO TO 10
90 READ(5,NAMEUV)
      GO TO 10
100 READ(5,NAMEHC)
      GO TO 10
110 WRITE(6,120)
120 FORMAT('1*MP          MODEL PARAMETERS'/)
C
      WRITE(6,130)
130 FORMAT('OMODEL PARAMETERS FOR ULTRAFILTRATION MODULE')
      ICARD=0
      WRITE(6,140) JPUFSS, G1UF, G2UF, GINFUF, JWUFSS, C1,
      &          C2, CINF, VISCUF, DENBUF
140 FORMAT('0 JPUFSS=',I3,'      G1UF=',G12.5,'      G2UF=',
      & G12.5,'      GINFUF=',G12.5/'      JWUFSS=',I3,'      C1=',
      & G12.5,'      C2=',G12.5,'      CINF=',G12.5/12X,
      & '      VISCUF=',G12.5,'      DENBUF=',G12.5)
C
      WRITE(6,150)
150 FORMAT('OMODEL PARAMETERS FOR TUBULAR RO MODULE')
      WRITE(6,160) JPTRSS, G1TR, G2TR, GINFTR, JWTRSS, ADAXTR,
      &          BDAXTR, CDAXTR, APITR, BTR, CTR, DCXTR,
      &          VISCTR, DENBTR
160 FORMAT('0 JPTRSS=',I3,'      G1TR=',G12.5,'      G2TR=',
      & G12.5,'      GINFTR=',G12.5/'      JWTRSS=',I3,'      ADAXTR=',
      & G12.5,'      BDAXTR=',G12.5,'      CDAXTR=',G12.5/15X,
      & '      APITR=',G12.5,'      BTR=',G12.5,'      CTR=',G12.5/
      & 15X,'      DCXTR=',G12.5,'      VISCTR=',G12.5,'      DENBTR=',
      & G12.5)
C
      WRITE(6,170)

```

```

170 FORMAT('OMODEL PARAMETERS FOR GEL-MODEL')
    WRITE(6,180) JPGMSS, GAMMA, APIGM, BPIGM, JWGMSS, BGM,
&          CGM, RATIO, DCXGM, ADAXGM, BDAXGM,
&          CDAXGM, CAGEL, VISCGR, DENBGM
180 FORMAT('0 JPGMSS=',I3,' GAMMA=',G12.5,' APIGM=',
& G12.5,' BPIGM=',G12.5/' JWGMSS=',I3,' BGM=',
& G12.5,' CGM=',G12.5,' RATIO=',G12.5/15X,
& 'DCXGM=',G12.5,' ADAXGM=',G12.5,' BDAXGM=',G12.5/
& 14X,'CDAXGM=',G12.5,' CAGEL=',G12.5,' VISCGR=',
& G12.5/14X,'DENBGM=',G12.5)

```

C

```

    WRITE(6,190)
190 FORMAT('OMODEL PARAMETERS FOR REVERSE OSMOSIS')
    WRITE(6,200) JWRITE, TOLMX, TOLMN, AKA, MCNT2, AKC,
&          ERE, APIRO, MCNT3, BPIRO, GAMARO, BRO,
&          NSTEPS, CRO, NF, ROKE, KWRITE, VISC,
&          RHOB
200 FORMAT('0 JWRITE=',I3,' TOLMX=',G12.5,' TOLMN=',
& G12.5,' AKA=',G12.5/' MCNT2=',I3,' AKC=',
& G12.5,' ERE=',G12.5,' APIRO=',G12.5/' MCNT3=',
& I3,' BPIRO=',G12.5,' GAMARO=',G12.5,' BRO=',
& G12.5/' NSTEPS=',I3,' CRO=',G12.5,' NF=',
& G12.5,' ROKE=',G12.5/' KWRITE=',I3,' VISC=',
& G12.5,' RHOB=',G12.5)

```

C

```

    WRITE(6,210)
210 FORMAT('OMODEL PARAMETERS FOR THE UV/OZONATION UNIT')
    WRITE(6,220) NWRITE, KHENRY, ECOZ, ETOC, KRATE,
&          KDCOMP, EOZD, UVEPCT, ALPHA, EN, QPRIME,
&          RGAS, UVRHO
220 FORMAT('0 NWRITE=',I3,' KHENRY=',G12.5,' ECOZ=',
& G12.5,' ETOC=',G12.5/15X,'KRATE=',G12.5,
& ' KDCOMP=',G12.5,' EOZD=',G12.5/14X,'UVEPCT=',
& G12.5,' ALPHA=',G12.5,' EN=',G12.5/14X,
& 'QPRIME=',G12.5,' RGAS=',G12.5,' UVRHO=',G12.5)

```

C

```

    WRITE(6,230)
230 FORMAT('OMODEL PARAMETERS FOR HYPOCHLORINATION UNIT')
    WRITE(6,240) JWRTHC, ALPHC, RDHC, HCRHO, MCNTHC,
&          KEQHC, CAOCL2, MWHOCL, MWOC
240 FORMAT('0 JWRTHC=',I3,' ALPHC=',G12.5,' RDHC=',
& G12.5,' HCRHO=',G12.5/' MCNTHC=',I3,' KEQHC=',
& G12.5,' CAOCL2=',G12.5,' MWHOCL=',G12.5/15X,
& 'MWOC=',G12.5)
    RETURN
    END

```

SUBROUTINE RSTRME

C
C
C

READ STREAM ELEMENT DEFINITIONS

```

COMMON STREAM(4,100), ICCNFG(8,100), PAR(500), NPAR,
&      NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
COMMON /CREAD/ IFIRST, IAST, ICARD(20)
COMMON /NAMES/ IUNITS(12), NMSTRM(5,100),
&      NMEQPT(5,100), NMELE(5,5), ISUNIT(5),
&      NMPAR(6,75), IDNMPR(150)
DIMENSION L(5)
WRITE(6,10)
10 FORMAT('1*SE      STREAM ELEMENT DEFINITIONS'//6X,
& 'ELEMENT  UNITS      DESCRIPTION')
20 READ(5,30) IFIRST, NS1, K, L, ICARD
30 FORMAT(A1,I4,1X,A4,5X,5A4,T1,20A4)
C IF FIRST CHARACTER IS "*" THEN A CONTROL CARD WAS FOUND
IF(IFIRST.EQ. IAST) RETURN
IF(NS1.LE. 0 .OR. NS1.GT. 5) GO TO 60
ISUNIT(NS1)= K
DO 40 J= 1,5
40 NMELE(NS1,J)= L(J)
WRITE(6,50) NS1, ISUNIT(NS1), (NMELE(NS1,J), J= 1,5)
GO TO 20
50 FORMAT(1X,I9,6X,A4,5X,10A4)
60 WRITE(6,70) NS1, ICARD
70 FORMAT('0*****ELEMENT',I3,' ENCOUNTERED.  WE ONLY HAVE
& ' ELEMENTS ONE THROUGH FIVE (1-5).  CHECK DATA CARD.'/
& ' ',20A4,'')
STOP
END
```

SUBROUTINE RSTRMN

```

C
C READ THE NAMES OF THE STREAMS
C
      COMMON /CREAD/ IFIRST, IAST, ICARD(20)
      COMMON /NAMES/ IUNITS(12), NMSTRM(5,100),
&                  NMECPT(5,100), NMELE(5,5), ISUNIT(5),
&                  NMPAR(6,75), IDNMPR(150)
      DIMENSION L(5)
      WRITE(6,10)
10  FORMAT('1*SN          STREAM NAMES'//6X,'STREAM      ',
& 'DESCRIPTION')
20  READ(5,30) IFIRST, N, L, ICARD
30  FORMAT(A1,I4,5X,5A4,T1,20A4)
C IF FIRST CHARACTER IS "*" THEN A CONTROL CARD WAS READ
  IF(IFIRST.EQ. IAST) RETURN
  DC 40 J=1,5
40  NMSTRM(J,N)= L(J)
  WRITE(6,50) N, (NMSTRM(J,N),J=1,5)
50  FORMAT(1X,I9,5X,5A4)
  GO TO 20
  END

```

```

SUBROUTINE RCONFG
C
C READ PLANT CONFIGURATION
C
      INTEGER NPAREQ(18), NPARIC(18), EQNAME(4,18)
      COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
&          NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
      COMMON /CREAD/ IFIRST, IAST, ICARD(20)
      COMMON /NAMES/ IUNITS(12), NMSTRM(5,100),
&          NMEQPT(5,100), NMELE(5,5), ISUNIT(5),
&          NMPAR(6,75), IDNMPR(150)
C NPAREQ(I) IS THE NUMBER OF PARAMETERS FOR EQUIPMENT TYPE
      DATA NPAREQ/4, 6, 1, 1, 7, 0, 6, 6, 12, 8, 0, 7, 4, 5,
&          5, 6, 6, 6/
C NPARIC(I) IS THE NUMBER OF INITIAL CONDITIONS FOR
C EQUIPMENT TYPE I
      DATA NPARIC/4, 4, 6*0, 3, 5, 8*0/
      DATA EQNAME/'MIXE', 'E TA', 'NK ', ' ', ' ',
&          'OVER', 'FLOW', 'TAN', 'K ',
&          'PUMP', ' ', ' ', ' ', ' ',
&          'STRE', 'AM S', 'PLIT', 'TER ',
&          'STRE', 'AM S', 'OURC', 'E ',
&          'STRE', 'AM M', 'IXER', ' ',
&          'ULTR', 'APIL', 'TRAT', 'ION ',
&          'REVE', 'RSE ', 'OSMO', 'SIS ',
&          'UV-O', 'ZONE', 'UNI', 'T ',
&          'HYPO', 'CHLO', 'RINA', 'TION',
&          'STRE', 'AM S', 'INK ', ' ',
&          'PID ', 'CONT', 'ROLL', 'ER ',
&          'SENS', 'OR ', ' ', ' ',
&          'MANI', 'PULA', 'TOR ', ' ',
&          'RATI', 'O CO', 'NTRO', 'LLER',
&          'BINA', 'RY C', 'ONTR', 'OLER',
&          'TUBU', 'LAR ', 'R-O ', ' ',
&          'GEL ', 'MODE', 'L UF', ' '/
      WRITE(6,10)
10  FORMAT('1*C          PLANT CONFIGURATION'/)
      NCALL= -1
      NPAR= 1
      IUNIT=1
20  READ(5,30) IFIRST, (ICONFG(I,IUNIT), I= 1,7),
&          (NMEQPT(I,IUNIT), I= 1,5), ICARD, DESC
30  FORMAT(A1,I4,3X,A2,5I5,5X,5A4,T1,20A4,T41,5A4)
C IF FIRST CHARACTER IS "*" THEN A CONTROL CARD WAS FOUND
      IF(IFIRST.EQ. IAST) GO TO 60
      NEQ= IUNIT
      ICONFG(8,IUNIT)= NPAR
C CHANGE EQUIPMENT TYPE ALPHA CODE TO NUMERIC CODE
      CALL EQCODE
C
C READ AND WRITE THE PARAMETERS ASSOCIATED WITH EACH UNIT OF
C EQUIPMENT.
C

```

```

C READ INPUT DATA
      ITYPE= ICONFG(2,IUNIT)
      NPARS= NPAREQ(ITYPE)
      NPARF= NPAR + NPARS - 1
      IF (NPARS .EQ. 0) NPARF= NPAR
      READ(5,40) (PAR(I), I= NPAR, NPARF)
40  FORMAT(8E10.0)
      WRITE(6,50) ICONFG(1,IUNIT), (EQNAME(J,1TYPE), J= 1, 4
&      DESC
50  FORMAT('OUNIT NC.',15,10X,4A4,4X,10A4)
C CALL PRNTEQ TO PRINT OUT STREAM INPUTS AND OUTPUTS FOR
C   UNIT IUNIT
      CALL PRNTEQ(NPARIC(ITYPE))
      WRITE(6,30)
      NPAR= NPAR + NPARS
      IUNIT= IUNIT + 1
      GC TO 20
60  NEQ= IUNIT - 1
      CALL STRM2
      IF (NFATER .EQ. 0) RETURN
      WRITE(6,70) NFATER
70  FORMAT('O*****',15,' FATAL ERRORS.  RUN TERMINATED.')
      STOP
      END

```

```

      SUBROUTINE STRM1
C
C   THIS SUBROUTINE IS USED TO MAKE SURE THAT NO STREAM HAS
C   BEEN GIVEN A MULTIPLE DEFINITION
C
      COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
&      NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
      COMMON /CHECK/ JSCK(2,100)
      DO 110 I= 3,7
          K= ICONFG(I,IUNIT)
          IF(K) 60, 120, 10
C   AN INPUT STREAM
      10 IF(JSCK(1,K)) 40, 20, 30
      20 JSCK(1,K)= ICCNFG(1,IUNIT)
          GO TO 110
      30 JSCK(1,K)= -JSCK(1,K)
      40      L= IABS(JSCK(1,K))
          WRITE(6,50) K, L
      50 FORMAT(6X,'*****STREAM',I4,' HAS BEEN PREVIOUSLY',
& ' DEFINED AS AN INPUT TO UNIT',I4)
          NFATER= NFATER + 1
          GO TO 110
C   AN OUTPUT STREAM
      60      K= -K
          IF(JSCK(2,K)) 90, 70, 80
      70 JSCK(2,K)= ICONFG(1,IUNIT)
          GO TO 110
      80 JSCK(2,K)= -JSCK(2,K)
      90      L= IABS(JSCK(2,K))
          WRITE(6,100) K, L
      100 .FORMAT(6X,'*****STREAM',I4,' HAS BEEN PREVIOUSLY',
& ' DEFINED AS AN OUTPUT FROM UNIT',I4)
          NFATER= NFATER + 1
      110 CONTINUE
      120 RETURN
          END

```

SUBROUTINE STRM2

C

C PRINT STREAM SUMMARY

C

```

COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
& NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
COMMON /CHECK/ JSCK(2,100)
WRITE(6,10)
10 FORMAT('1STREAM SUMMARY'/6X,'STREAM',5X,'SOURCE',
& 5X,'DESTINATION')
DO 210 I= 1,100
    K= JSCK(1,I)
    L= JSCK(2,I)
    IF(L) 20, 90, 140
20    L= -L
    IF(K) 30, 50, 70
30    K= -K
    WRITE(6,40) I, L, K
40 FORMAT(1X,I9,I11,I14,5X,'SOURCE AND DESTINATION',
& ' MULTIPLE DEFINITION')
    NFATER= NFATER + 1
    GO TO 210
50 WRITE(6,60) I, L, K
60 FORMAT(1X,I9,I11,I14,5X,'SOURCE MULTIPLE DEFINITIONS'/
& ' NO DESTINATION DEFINITION')
    NFATER= NFATER + 1
    GO TO 210
70 WRITE(6,80) I, L, K
80 FORMAT(1X,I9,I11,I14,5X,'SOURCE MULTIPLE DEFINITIONS')
    NFATER= NFATER + 1
    GO TO 210
90 IF(K) 100, 210, 120
100    K= -K
    WRITE(6,110) I, L, K
110 FORMAT(1X,I9,I11,I14,5X,'NO SOURCE DEFINITION/MULTIPLE
1DEFINITIONH')
    NFATER= NFATER + 1
    GO TO 210
120 WRITE(6,130) I, L, K
130 FORMAT(1X,I9,I11,I14,5X,'NO SOURCE DEFINITION')
    NFATER= NFATER + 1
    GO TO 210
140 IF(K) 150, 170, 190
150    K= -K
    WRITE(6,160) I, L, K
160 FORMAT(1X,I9,I11,I14,5X,'MULTIPLE DESTINATION DEFINITI
NFATER= NFATER + 1
GO TC 210
170 WRITE(6,180) I, L, K
180 FORMAT(1X,I9,I11,I14,5X,'NO DESTINATION DEFINITION')
    NFATER= NFATER + 1
    GO TO 210
190 WRITE(6,200) I, L, K

```

```
200 FORMAT(1X,I9,I11,I14)
210 CONTINUE
    RETURN
    END
```

```

      SUBROUTINE EQCCDE
C
C  CHANGE EQUIPMENT TYPE ALPHA CODE TO NUMERIC CODE
C
      COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
      &      NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
C  EQUIPMENT CODE ARRAY
      DIMENSION ICODE(18)
      DATA ICODE /'MT', 'OT', 'P ', 'SP', 'SO', 'SM', 'UF',
      &      'RO', 'UV', 'HC', 'SK', 'CN', 'SN', 'MN',
      &      'RC', 'BC', 'TR', 'GM'/
C  GET EQUIPMENT CODE FOR UNIT "IUNIT"
      M= ICONFG(2,IUNIT)
C  LOCK IN EQUIPMENT CODE TABLE
      DO 10 I=1,18
      IF(M.EQ. ICODE(I)) GO TO 40
10  CONTINUE
C  WE DID NOT FIND CODE IN TABLE
      WRITE(6,20) M, (ICONFG(I,IUNIT), I= 1,7)
20  FORMAT(/' *****EQUIPMENT CODE ',A2,' IN FOLLOWING',
      & ' ' ENTRY IS ILLEGAL'/11X,I5,3X,A2,5I5/)
      NFATER= NFATER + 1
      ICONFG(2,IUNIT)= 0
      READ(5,30) A
30  FORMAT(A4)
      RETURN
C  SET EQUIPMENT TYPE EQUAL TO NUMERIC CODE
40  ICONFG(2,IUNIT)= I
      RETURN
      END

```

```

      SUBROUTINE PRNTEQ (NINCON)
C
C   PRINT THE SPECIFICATIONS
C
      COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
&          NCALL, IUNIT, NPATER, NS, NEQ, DESC(5)
      COMMON /NAMES/ IUNITIS(12), NMSTRM(5,100),
&          NMEQPT(5,100), NMELE(5,5), ISUNIT(5),
&          NMPAR(6,75), IDNMPR(150)
      INTEGER ISTR(5), OSTR(5)
C   STREAM CONTAINS THE INFORMATION FOR EACH STREAM
C   ICONFG CONTAINS THE INFORMATION FOR EACH UNIT
C   NINCON IS THE NUMBER OF INITIAL CONDITIONS
C   NINSTR IS THE NUMBER OF INPUT STREAMS (MAX= 5)
C   NOTSTR IS THE NUMBER OF OUTPUT STREAMS (MAX= 5)
      NINSTR= 0
      NOTSTR= 0
C   POSITIONS 3 THRU 7 OF ICCNFG CONTAIN THE STREAM NUMBERS
C   TO/FROM UNIT IUNIT
      DO 30 I=3,7
          J= ICONFG(I,IUNIT)
          IF (J) 10, 30, 20
10      NOTSTR= NOTSTR + 1
          OSTR(NOTSTR)= -J
          GO TO 30
20      NINSTR= NINSTR + 1
          ISTR(NINSTR)= J
30      CONTINUE
          WRITE(6,40)
40      FORMAT(6X,'INPUT  STREAMS')
          IF (NINSTR .NE. 0) GO TO 60
          WRITE(6,50)
50      FORMAT(11X,'NONE')
          GO TO 90
60      DO 70 I= 1,NINSTR
          J= ISTR(I)
70      WRITE(6,80) J, (NMSTRM(K,J), K= 1,5)
80      FORMAT(1X,I15,5X,5A4)
90      WRITE(6,100)
100     FORMAT(6X,'OUTPUT STREAMS')
          IF (NOTSTR .NE. 0) GO TO 110
          WRITE(6,50)
          GO TO 130
110     DO 120 I=1,NOTSTR
          J= OSTR(I)
120     WRITE(6,80) J, (NMSTRM(K,J), K=1,5)
C   CALL STRM1 TO CHECK FOR CONSISTANCY
130     CALL STRM1
C   POSITION 2 OF ICONFG CONTAINS THE TYPE CODE FOR UNIT IUNI
      KEQ= ICONFG(2,IUNIT)
      K= 1
140     IF (IDNMPR(K) .EQ. KEQ) GO TO 150
          K= K + 2 + IDNMPR(K+1)
150

```

```

      GO TO 140
150 KPAR= IDNMPR (K+1)
      K= K + 2
      L= NPAR
      WRITE(6,160)
160 FORMAT( 6X,'INITIAL CONDITIONS')
      IF(NINCON .NE. 0) GO TO 180
      WRITE(6,170)
170 FORMAT( 11X,'NONE')
      GO TO 210
180 DO 200 I=1,NINCCN
      J= IDNMPR (K)
      M= NMPAR(6,J)
      WRITE(6,190) (NMPAR(IA,J), IA=1,5), PAR(L), IUNITS(M)
190 FORMAT(11X,5A4,G14.5,1X,A4)
      L= L + 1
200 K= K + 1
210 WRITE(6,220)
220 FORMAT( 6X,'DESIGN PARAMETERS')
      NDESPR= KPAR - NINCCN
      IF(NDESPR .NE. 0) GO TO 230
      WRITE(6,170)
      RETURN
230 DO 240 I=1,NDESPR
      J= IDNMPR (K)
      M= NMPAR(6,J)
      WRITE(6,190) (NMPAR(IA,J), IA=1,5), PAR(L), IUNITS(M)
      L= L + 1
240 K= K + 1
      RETURN
      END

```

```

      SUBROUTINE RPRINT
C
C  READ PRINT SPECIFICATIONS
C
      COMMON /CPRINT/ TPRINT, NPELE, KPRINT(2,10)
      WRITE(6,10)
10  FORMAT('1*PR          PRINT SPECIFICATIONS'/)
      READ(5,20) TPRINT
20  FORMAT(G10.5)
      WRITE(6,30) TPRINT
30  FORMAT('0          PRINT INTERVAL IS ',G13.5,' HRS'/)
      CALL SPEC(KPRINT,NPELE,10)
      RETURN
      END

```

```

SUBROUTINE OFFLINE
C
C  SETUP THE LABEL ON THE OFF-LINE STORAGE DEVICE (UNIT 8)
C
      COMMON /CTIME/ TIME, FTIME, DT
      COMMON /COPLN/ NPL, LIST(2,50), LABLE, TOPLN,
&              MESSAG(20), IEOP, IHEAD
      WRITE(6,10)
10  FORMAT('1*OL      OFF-LINE PARAMETER LIST')
      READ(5,20) MESSAG
20  FORMAT(20A4)
      WRITE(6,30) MESSAG
30  FORMAT(' THE HEADER MESSAGE FOR THE DATA SET IS'/
& 1X,20A4)
      READ(5,40) LABLE
40  FORMAT(I2)
      IF(LABLE .LT. 0) GO TO 60
      WRITE(6,50)
50  FORMAT(' THE NEW VALUES WILL BE ADDED TO THE END',
& ' OF THE DATA SET.')
      GO TO 80
60  WRITE(6,70)
70  FORMAT(' THE DATA SET WILL BE CLEARED, AND THE NEW',
& ' VALUES SAVED FROM THE BEGINNING.')
80  READ(5,90) TOPLN
90  FORMAT(F10.3)
      WRITE(6,100) TOPLN
100 FORMAT(' THE SAVE INTERVAL IS',G12.4)
      CALL SPEC(LIST,NPL,50)
C
C  WRITE LABEL ON OFF-LINE STORAGE DEVICE ON UNIT 8
C
C  IF LABLE .LT. 0, THEN CLEAR DATA SET
C  IF LABLE .GE. 0, THEN ADD AT THE END OF THE DATA SET
      REWIND 8
      IF(LABLE .LT. 0) GO TO 120
110 READ(8,150,END=130) ICODE
      IF(ICODE .NE. IEOP) GO TO 110
      BACKSPACE 8
C  WRITE HEADER
120 WRITE(8,150) IHEAD, MESSAG, TOPLN, NPL,
&              (LIST(1,I), LIST(2,I), I= 1, NPL)
      RETURN
130 WRITE(6,140)
140 FORMAT(' ***** END-OF-FILE OCCURRED WHILE SETTING',
& ' UP THE OFF-LINE STORAGE HEADER *****')
      STOP
150 FORMAT(A4)
      END

```

```

      SUBROUTINE PLOT1
C
C  READ IN THE PARAMETERS/ELEMENTS TO PLOT
C
      COMMON /CPLOT/ NPLOT, TPLOT, KPLOT, PLTDTA(100,10),
&              JPSTRM(2,10)
      READ(5,10) TPLOT
10  FORMAT(F10.0)
      WRITE(6,20)
20  FORMAT('1*PL          PLOT VARIABLES')
      WRITE(6,30) TPLOT
30  FORMAT(/6X, 'PLOT DURATION', F7.2, ' HRS'/)
      CALL SPEC(JPSTRM,NPLOT,10)
      RETURN
      END

```

```

      SUBROUTINE OLDVAL
C
C LOCATE THE DESIRED SET OF OLD VALUES
C
      COMMON /COFLN/ NPL, LIST(2,50), IABLE, TOFLN,
&      MESSAG(20), IEOF, IHEAD
      COMMON /CTIME/ TIME, PTIME, DT
      DIMENSION MSG(20)
      WRITE(6,1)
1  FORMAT('1*OL          OLD VALUES SPECIFICATIONS')
C READ THE UNIQUE MESSAGE
      READ(5,10) MESSAG
10  FORMAT(20A4)
      WRITE(6,20) MESSAG
20  FORMAT(1X,20A4)
      REWIND 8
C FIND HEADER
30  READ(8,70,END=50) ICODE
      IF(ICODE .EQ. IEOF) GO TO 50
      IF(ICODE .NE. IHEAD) GO TO 30
      READ(8,70,END=50) MSG
      DO 40 I=1,20
      IF(MSG(I) .NE. MESSAG(I)) GO TO 30
40  CONTINUE
      READ(8,70,END=50) TOFLN, NPL, (LIST(1,I), LIST(2,I),
&      I= 1, NPL), PTIME, DT
      NPL= -NPL
      RETURN
50  WRITE(6,60) MESSAG
60  FORMAT(' THE RUN FOR'/1X,20A4/' WAS NOT FOUND.')
      STOP
70  FORMAT(A4)
      END

```

SUBROUTINE SAVEIT

```

C
C SAVE VALUES ON OFF-LINE STORAGE DEVICE, UNIT 8
C
COMMON /COFLN/ NPL, LIST(2,50), IABLE, TOFLN,
&          MESSAG(20), IEOP, IHEAD
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
&          NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
COMMON /CTIME/ TIME, FTIME, DT
DIMENSION P(50)
IF(NPL .LE. 0) RETURN
DO 20 I= 1, NPL
    K= LIST(1,I)
    L= LIST(2,I)
    IF(K .GT. 0) GO TO 10
    P(I)= PAR(L)
GO TO 20
10  P(I)= STREAM(L,K)
20  CONTINUE
    WRITE(8,30) TIME, (P(I),I=1,NPL)
30  FORMAT(A4)
    RETURN
END

```

SUBROUTINE GETIT

```

C
C RETREIVE THE VALUES THAT WERE SAVED ON THE OFF-LINE
C STORAGE DEVICE, UNIT 8, BY "SAVEIT" IN A PREVIOUS RUN
C
      COMMON /COFLN/ NPL, LIST(2,50), IABLE, TOFLN,
&          MESSAG(20), IEOF, IHEAD
      COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
&          NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
      COMMON /CTIME/ TIME, FTIME, DT
      DIMENSION P(50)
      KPL= -NPL
      READ(8,30) TIME, (P(J),J=1,KPL)
      DO 20 I= 1, KPL
          K= LIST(1,I)
          L= LIST(2,I)
          IF(K .GT. 0) GO TO 10
          PAR(L)= P(I)
          GO TO 20
10  STREAM(L,K)= P(I)
20  CONTINUE
30  FORMAT(A4)
      RETURN
      END

```

SUBROUTINE PRINT

```

C
C THIS SUBROUTINE IS USED TO PRINT OUT THE SPECIFIED VALUES
C
COMMON STREAM(4,100), ICCNFG(8,100), PAR(500), NPAR,
&      NCALL, IUNIT, NPATER, NS, NEQ, DESC(5)
COMMON /CPRINT/ TPRINT, NPELE, KPRINT(2,10)
COMMON /CTIME/ TIME, FTIME, DT
DIMENSION P(10), NME1(10), NME2(10), I1(10), I2(10),
&      NAME1(2), NAME2(2)
DATA NAME1, NAME2/'STEM', 'UNIT', 'ELE.', 'PAR.'/
DATA LINE, NLines/0, 51/
C PRINT HEADINGS WHEN LINE = 0
IF(LINE .GT. 0) GO TO 50
DO 20 I= 1, NPELE
    K= KPRINT(1,I)
    L= KPRINT(2,I)
    J= 1
    IF(K .GT. 0) GO TO 10
    J= 2
    K= -K
    L= L - ICCNFG(8,K) + 1
    K= ICCNFG(1,K)
10 NME1(I)= NAME1(J)
   NME2(I)= NAME2(J)
    I1(I)= K
20 I2(I)= L
   WRITE(6,30) (NME1(I), I1(I), I= 1, NPELE)
30 FORMAT('1',10X,10(4X,A4,I4))
   WRITE(6,40) (NME2(I), I2(I), I= 1, NPELE)
40 FORMAT(6X,'TIME ',10(4X,A4,I4))
   LINE= NLines
C RETRIEVE ELEMENTS
50 DO 70 I= 1, NPELE
    K= KPRINT(1,I)
    L= KPRINT(2,I)
    IF(K .GT. 0) GO TO 60
    P(I)= PAR(L)
    GO TO 70
60 P(I)= STREAM(L,K)
70 CONTINUE
   WRITE(6,80) TIME, (P(I), I= 1, NPELE)
80 FORMAT(1X,G10.3,10G12.3)
   LINE= LINE - 1
RETURN
END

```

SUBROUTINE PLOT2

```

C
C  SAVE THE VALUES OF THE PARAMETERS/ELEMENTS FOR PLOTTING
C
      COMMON STREAM(4,100), ICONF(8,100), PAR(500), NPAR,
&      NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
      COMMON /CPLOT/ NPLOT, TPLOT, KPLOT, PLTDTA(100,10),
&      JPSTRM(2,10)
      COMMON /CTIME/ TIME, PTIME, DT
      IF (KELOT .EQ. 100) RETURN
10  CONTINUE
      KPLOT= KPLOT + 1
      DO 30 I= 1, NFLOT
          K= JPSTRM(1,I)
          L= JPSTRM(2,I)
          IF (K .GT. 0) GO TO 20
          PLTDTA(KPLOT,I)= PAR(L)
          GO TO 30
20  PLTDTA(KPLOT,I)= STREAM(L,K)
30  CONTINUE
      IF (KPLOT*0.01*TPLOT .LE. TIME) GO TO 10
      RETURN
      END

```

```

      SUBROUTINE SPEC (IARRAY,N,NZ)
C
C  READ VARIABLES TO BE PRINTED, PLOTTED, OR SAVED OFF-LINE
C
      COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
1      NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
      COMMON /CREAD/ IFIRST, IAST, ICARD(20)
      COMMON /NAMES/ IUNITS(12), NMSTRM(5,100),
&      NMEQPT(5,100), NMELE(5,5), ISUNIT(5),
&      NMPAR(6,75), IDNMPR(150)
      DIMENSION IARRAY(2,NZ)
      N=0
      10 READ(5,20) IFIRST, K, L, ICARD
      20 FORMAT(A1,I4,I5,T1,20A4)
C  IF THE FIRST CHARACTER IS '*' THEN IT IS A CONTROL CARD
      IF(IFIRST.EQ. IAST) RETURN
      IF(N.LT. NZ) GO TO 40
      WRITE(6,30) ICARD, NZ
      30 FORMAT(' *****CARD ''',20A4,''' IGNORED.',I4,
& ' MAXIMUM')
      GO TO 10
      40 IF(K.GT. 0) GO TO 110
C  ENTRY IS UNIT/PARAMETER
      K= -K
C  LOOK FOR UNIT NUMBER IN CONFIGURATION ARRAY
      DO 50 I= 1,NEQ
      IF(ICONFG(1,I).EQ. K) GO TO 70
      50 CONTINUE
      WRITE(6,60) ICARD, K
      60 FORMAT(' *****CARD ''',20A4,''' IGNORED. NO UNIT',
& ' NUMBER',I5)
      GO TO 10
C  WE HAVE FOUND UNIT K
      70      N= N + 1
      IARRAY(1,N)= -I
      IARRAY(2,N)= ICONFG(8,I) + L - 1
      ITYPE= ICONFG(2,I)
      M= 1
      80 IF(IDNMPR(M).EQ. ITYPE) GO TO 90
      M= M + 2 + IDNMPR(M+1)
      GO TO 80
      90      M= M + 1 + L
      M= IDNMPR(M)
      WRITE(6,100) K, (NMEQPT(J,I), J= 1,5), L,
&      (NMPAR(J,M), J= 1,5)
      100 FORMAT(' ++++UNIT', 19, 5X, 5A4 / 6X, 'PARAMETER',
& I4, 6X, 5A4)
      GO TO 10
C  ENTRY IS STREAM/ELEMENT
      110      N= N + 1
      IARRAY(1,N)= K
      IARRAY(2,N)= L
      WRITE(6,120) K, (NMSTRM(J,K), J= 1, 5), L,

```

```
      6      (NMELE(L,J) , J= 1, 5)  
120 FORMAT(' -----STREAM', 17, 5X, 5A4 / 6X,  
      6 'ELEMENT', I6, 5X, 5A4)  
      GC TC 10  
      END
```

```

      SUBROUTINE RCARD
C
C  READ A DATA CONTROL CARD
C
      COMMON /CREAD/ IFIRST, IAST, ICARD(20)
      READ(5,10) IFIRST, ICARD
10  FORMAT(A1,T1,20A4)
C  FIRST CHARACTER MUST BE '*' FOR A DATA CONTROL CARD
      IF(IFIRST.EQ. IAST) RETURN
      WRITE(6,20) ICARD
20  FORMAT('0*****EXPECTING DATA CONTROL CARD, FOUND ' /
& ' ',20A4,' ')
      STOP
      END

```

```

      SUBROUTINE SUBCAL
C      CALL SUBROUTINE CORRESPONDING TO EQUIPMENT TYPE
      COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
&      NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
C      GET EQUIPMENT TYPE
      ITYPE= ICONFG(2,IUNIT)
      IF (ITYPE .EQ. 0) RETURN
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,
C      MT OT P SP SO SM UF RO UV HC SK CN SN MN
&      150,160,170,180), ITYPE
C      RC BC TR GM

      RETURN
10 CALL MT
      RETURN
20 CALL OT
      RETURN
30 CALL P
      RETURN
40 CALL SP
      RETURN
50 CALL SC
      RETURN
60 CALL SM
      RETURN
70 CALL UF
      RETURN
80 CALL RO
      RETURN
90 CALL UV
      RETURN
100 CALL HC
      RETURN
110 CALL SK
      RETURN
120 CALL PID
      RETURN
130 CALL SENSOR
      RETURN
140 CALL MANIP
      RETURN
150 CALL RATIO
      RETURN
160 CALL BINARY
      RETURN
170 CALL TR
      RETURN
180 CALL GM
      RETURN
      END

```

SUBROUTINE PLOT3

C
C
C

THIS SUBROUTINE MAKES THE LINEPRINTER PLOTS

```

COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
&      NCALL, IUNIT, NPATER, NS, NEQ, DESC(5)
COMMON /CFLOT/ NPLT, TPLOT, KPLOT, PLTDTA(100,10),
&      JPSTRM(2,10)
COMMON /CREAD/ IFIRST, IAST, ICARD(20)
COMMON /NAMES/ IUNITS(12), NMSTRM(5,100),
&      NMEQPT(5,100), NMELE(5,5), ISUNIT(5),
&      NMPAR(6,75), IDNMPR(150)
DIMENSION ICHR(10), IARR(100), XAXIS(5), PLTCRD(2,10)
EQUIVALENCE (ICARD1, ICARD(1))
DATA ICD1/'*PC '/, IBLK/' '/, ISTAR/'*'/
IF(NPLT .EQ. 0) RETURN
      K1= 1
CALL RCARD
IF(ICARD1 .EQ. ICD1) GO TO 20
WRITE(6,10) ICARD
10 FORMAT('0 EXPECTING DATA CONTROL CARD '*PC ', FOUND'/
&      ' ', 20A4, ' ')
STOP
20 WRITE(6,30)
30 FORMAT('1 PLOT PARAMETERS')
DO 100 K2= K1, NPLT
  READ(5,40) ICHR(K2), (PLTCRD(J,K2), J= 1, 2)
40 FORMAT(A1,4X,2F10.0)
  IF(ICHR(K2) .EQ. IBLK) GO TO 110
      K= JPSTRM(1,K2)
      L= JPSTRM(2,K2)
  IF(K .GT. 0) GO TO 80
      K= -K
  IUNIT= ICONFG(2,K)
  KPAR= L - ICONFG(8,K) + 1
  M= 1
50 IF(IDNMPR(M) .EQ. IUNIT) GO TO 60
  M= M + 2 + IDNMPR(M+1)
  GO TO 50
60  M= M + 1 + KPAR
  MI= IDNMPR(M)
  IXYZ= K
  JXYZ= ICONFG(1,K)
  WRITE(6,70) ICHR(K2), JXYZ, (NMEQPT(J,IXYZ), J=1,5),
&      PLTCRD(1,K2), KPAR,
&      (NMPAR(J,MI), J=1,5), PLTCRD(2,K2)
70 FORMAT(/1X,A1,'----UNIT',I10,5X,5A4,5X,'YO  =' ,F12.2/
1 6X,'PARAMETER',I5,5X,5A4,5X,'YMAX =' ,F12.2)
  GO TO 100
80 WRITE(6,90) ICHR(K2), K, (NMSTRM(J,K), J=1,5),
&      PLTCRD(1,K2), L,
&      (NMELE(L,J), J=1,5), PLTCRD(2,K2)
90 FORMAT(/1X,A1,'----STREAM',I8,5X,5A4,5X,'YO  =' ,

```

```

      & F12.2/6X,'ELEMENT',I7,5X,5A4,5X,'YMAX =',F12.2)
100 CONTINUE
      K2= NPLOT
      GO TO 120
110      K2= K2 - 1
120 WRITE(6,130)
130 FORMAT('0')
      LINE= 40
140 DO 150 I=1,100
150 IARR(I)= IBLK
      K= K2
160      YO= PLTCRD(1,K)
      YMX= PLTCRD(2,K)
      DY= (YMX-YO)/40.
      DO 170 I= 1, KPLOT
          J= 0.5 + (PLTDTA(I,K)-YO)/DY
          IF(J .NE. LINE) GO TO 170
          ICHAR= ICHR(K)
          IF(IARR(I) .NE. IBLK) ICHAR= ISTAR
          IARR(I)= ICHAR
170 CONTINUE
      K= K - 1
      IF(K .GE. K1) GO TO 160
      IF((LINE/10)*10 .EQ. LINE) GO TO 190
      WRITE(6,180) IARR
180 FORMAT(11X,'*',100A1)
      GO TO 210
190      YO= PLTCRD(1,K1)
      YMX= PLTCRD(2,K1)
      DY= (YMX-YO)/40.
      YC= YO + LINE*DY
      WRITE(6,200) YC, IARR
200 FORMAT(1X,F10.2,'+',100A1)
210      LINE= LINE - 1
      YO= PLTCRD(1,K1)
      IF(LINE .NE. 0) GO TO 140
      WRITE(6,220) YO
220 FORMAT(1X,F10.2,'+',10('*****+'))
      DX= TPLOT/5.
      DO 230 I=1,5
230 XAXIS(I)= DX*I
      WRITE(6,240) XAXIS
240 FORMAT(10X,'0.00',5F20.2//50X,'TIME, HR '/')
      K1= K2 + 1
      IF(K1 .LE. NPLOT) GO TO 20
      CALL RCARD
      RETURN
      END

```

SUBROUTINE MT

```
C THIS SUBROUTINE SIMULATES A MIXING TANK
C
C
C
C      INPUT   |           | OUTPUT
C      STREAMS | MIXING   | STREAMS
C      ----->| TANK     |----->
C      ----->|           |----->
C
C PARAMETER          QUANTITY
C    1              INITIAL VOLUME
C    2              INITIAL TSS CONC.
C    3              INITIAL TDS CONC.
C    4              INITIAL TOC CONC.
C
C COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
& NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
COMMON /CTIME/ TIME, FTIME, DT
COMMON /MATBAL/ BALNCE(4), AMTIN(4), AMTOUT(4)
COMMON /LOOK/ ISW
C ASCERTAIN IF READ PARAMETERS, INITIALIZE, OR COMPUTE
IF(NCALL) 10, 20, 70
C MATERIAL BALANCE
10 BALNCE(1)= PAR(NPAR) + BALNCE(1)
   BALNCE(2)= PAR(NPAR)*PAR(NPAR+1) + BALNCE(2)
   BALNCE(3)= PAR(NPAR)*PAR(NPAR+2) + BALNCE(3)
   BALNCE(4)= PAR(NPAR)*PAR(NPAR+3) + BALNCE(4)
   RETURN
C INITIALIZATION CALCUALTIONS
20 DO 50 I= 3, 7
   J= ICONFG(I,IUNIT)
   IF(J) 30, 60, 50
30   J= -J
   STREAM(2,J)= PAR(NPAR+1)
   STREAM(3,J)= PAR(NPAR+2)
   STREAM(4,J)= PAR(NPAR+3)
50 CONTINUE
60 RETURN
C SIMULATION
70   V= PAR(NPAR)
   VC2= V*PAR(NPAR+1)
   VC3= V*PAR(NPAR+2)
   VC4= V*PAR(NPAR+3)
   DO 100 I=3,7
     J= ICONFG(I,IUNIT)
     IF(J) 80, 100, 90
C OUTPUT STREAM
80   J= -J
     P= STREAM(1,J)
     TSS= PAR(NPAR+1)
     TDS= PAR(NPAR+2)
     TOC= PAR(NPAR+3)
```

```

        STREAM(2,J) = TSS
        STREAM(3,J) = TDS
        STREAM(4,J) = TOC
            V= V - DT*F
            VC2= VC2 - DT*F*TSS
            VC3= VC3 - DT*F*TDS
            VC4= VC4 - DT*F*TOC
        GO TO 100
C  INPUT STREAM
    90      F= STREAM(1,J)
            V= V + DT*F
            VC2= VC2 + DT*F*STREAM(2,J)
            VC3= VC3 + DT*F*STREAM(3,J)
            VC4= VC4 + DT*F*STREAM(4,J)
    100 CONTINUE
        IF(V .GT. 0.0) GO TO 130
        WRITE(6,110) ICONFG(1,IUNIT)
    110 FORMAT('0*****WARNING. UNIT NUMBER',I5,' HAS RUN DRY')
        DO 120 J= 1, 4
    120 PAR(NPAR+J-1) = 0.
        GO TO 140
    130 CONTINUE
            PAR(NPAR) = V
            PAR(NPAR+1) = VC2/V
            PAR(NPAR+2) = VC3/V
            PAR(NPAR+3) = VC4/V
    140 CCNTINUE
            J= NPAR + 3
            IF(ISW.EQ.1) WRITE(6,150) IUNIT, (PAR(I),I=NPAR,J)
    150 FORMAT(' UNIT',I5,' MT VOL=',G10.3,5X,' SS=',
& G10.3,5X,' DS=',G10.3,
& ' TOC=',G10.3)
        GO TO 20
        END

```

SUBROUTINE OT

THIS SUBROUTINE SIMULATES AN OVERFLOW TANK



PARAMETER	QUANTITY
1	INITIAL VOLUME
2	INITIAL TSS CONC.
3	INITIAL TDS CONC.
4	INITIAL TOC CONC.
5	OVERFLOW RATE
6	MAXIMUM VOLUME

THE OVERFLOW STREAM MUST BE SPECIFIED FIRST
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
& NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
COMMON /CTIME/ TIME, FTIME, DT
COMMON /MATEAL/ BALNCE(4), AMTIN(4), AMTOUT(4)
COMMON /LCCK/ ISW

ASCERTAIN IF READ PARAMETERS, INITIALIZE, OR SIMULATE
IF(NCALL) 10,60,110
10 IF(ICONFG(3,IUNIT).LT.0) GOTO30
WRITE(6,20) IUNIT,ICONFG(3,IUNIT)
20 FORMAT(6X,'*****ERROR, UNIT',I5,'. FIRST STREAM IN',
& ' CONFIGURATION IS',I5,'. MUST BE THE OUTPUT STREAM',
& ' FOR OVERFLOW')
NFATER= NFATER + 1
30 IF(PAR(NPAR+5) .GT. 0.) GO TO 50
WRITE(6,40) IUNIT, PAR(NPAR+5)
40 FORMAT(6X,'*****ERROR, UNIT',I5,'. MAXIMUM VOLUME =',
& F10.2,' IS INVALID. MUST BE POSITIVE.')
NFATER= NFATER + 1

MATERIAL BALANCE
50 BALNCE(1) = PAR(NPAR) + BALNCE(1)
BALNCE(2) = PAR(NPAR)*PAR(NPAR+1) + BALNCE(2)
BALNCE(3) = PAR(NPAR)*PAR(NPAR+2) + BALNCE(3)
BALNCE(4) = PAR(NPAR)*PAR(NPAR+3) + BALNCE(4)
RETURN

INITIALIZATION CALCULATIONS
60 J=-ICONFG(3,IUNIT)
STREAM(1,J) = PAR(NPAR+4)
70 DO 90 I= 3, 7
J= ICONFG(I,IUNIT)
IF(J) 80,100,90
80 J= -J
STREAM(2,J) = PAR(NPAR+1)
STREAM(3,J) = PAR(NPAR+2)
STREAM(4,J) = PAR(NPAR+3)

```

90 CONTINUE
100 RETURN
C SIMULATION
110     F= 0.
      DO 140 I= 4, 7
          J= ICONFG(I,IUNIT)
      IF (J) 120,140,130
120     J= -J
          F= F - STREAM(1,J)
      GO TO 140
130     F= F + STREAM(1,J)
140 CONTINUE
          V= PAR(NPAR)
          K= -ICONFG(3,IUNIT)
          FLOW= PAR(NPAR+4)
          VX= V + (F-FLOW)*DT
          VMAX= PAR(NPAR+5)
          IF(VX .GT. VMAX) FLOW= (V + F*DT - VMAX)/DT
          IF(VX .LT. 0.) FLOW= (V + F*DT)/DT
          STREAM(1,K)= FLOW
          PAR(NPAR)= V + (F - FLOW)*DT
          VC2= V*PAR(NPAR+1)
          VC3= V*PAR(NPAR+2)
          VC4= V*PAR(NPAR+3)
      DO 170 I= 3, 7
          J= ICONFG(I,IUNIT)
      IF (J) 150, 170, 160
C OUTPUT STREAM
150     J= -J
          STREAM(2,J)= PAR(NPAR+1)
          STREAM(3,J)= PAR(NPAR+2)
          STREAM(4,J)= PAR(NPAR+3)
          F= STREAM(1,J)
          IF(F .LT. 1.E-20) F=0.
          VC2= VC2 - DT*F*STREAM(2,J)
          VC3= VC3 - DT*F*STREAM(3,J)
          VC4= VC4 - DT*F*STREAM(4,J)
      GO TO 170
C INPUT STREAM
160     F= STREAM(1,J)
          IF(F .LT. 1.E-20) F= 0.
          VC2= VC2 + DT*F*STREAM(2,J)
          VC3= VC3 + DT*F*STREAM(3,J)
          VC4= VC4 + DT*F*STREAM(4,J)
170 CONTINUE
          VF= PAR(NPAR)
          IF(VF .LT. 1.E-20) GO TO 180
          PAR(NPAR+1)= VC2/VF
          PAR(NPAR+2)= VC3/VF
          PAR(NPAR+3)= VC4/VF
180 CONTINUE
          J= NPAR + 3
          IF(ISW .EQ. 1) WRITE(6,190) IUNIT, (PAR(I),I=NPAR,J)
190 FORMAT(' UNIT',I5,' OT VOL=',G10.3,5X,'SS=',G10.3,5X,

```

```
& 'DS=',G10.3,'TOC=',G10.3)  
GO TO 70  
END
```

C C C C C C C C C C C C C C C C

INPUT STREAM	VOLUMETRIC PUMP	OUTPUT STREAM
1 ----->		-----> 2

```

THE INPUT STREAM MUST BE SPECIFIED FIRST
THE OUTPUT STREAM MUST BE SPECIFIED SECOND
COMMON /LOOK/ ISW
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
& NCALL, IUNIT, NPATER, NS, NEQ, DESC(5)
ASCERTAIN IF READ DATA, INITIALIZE, OR SIMULATE
IF(NCALL) 10, 20, 20
RETURN IF MATERIAL BALANCE CALCULATIONS
10 RETURN
INITIALIZE AND SIMULATE ARE IDENTICAL
20 K= ICONF(3,IUNIT)
L= -ICONFG(4,IUNIT)
STREAM(1,K)= PAR(NPAR)
STREAM(1,L)= PAR(NPAR)
STREAM(2,L)= STREAM(2,K)
STREAM(3,L)= STREAM(3,K)
STREAM(4,L)= STREAM(4,K)
IF(ISW.EQ. 1) WRITE(6,30) IUNIT, (STREAM(I,L), I=1,4)
30 FORMAT(' UNIT',I5,' P FLOW=',G10.3,5X,'SS=',
& G10.3,5X,'DS=',G10.3,5X,'TOC=',G10.3)
RETURN
END

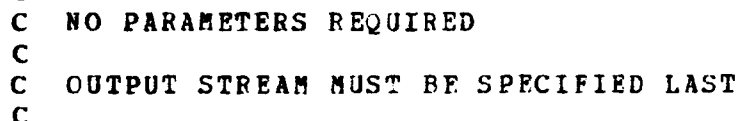
```



```

50 STREAM(1,J)= 0.
   GO TO 90
C 60 STREAM(1,J)= PAR(NPAR+3)
70 CCNTINUE
   IF (FRACT .GT. 1.) FRACT= 1.
   IF (FRACT .LT. 0.0) FRACT= 0.
      F= PAR(NPAR+3)*FRACT
   STREAM(1,J)= F
   IF (NCALL .EQ. 0) RETURN
C      F= STREAM(1,J)
      AMTIN(1)= AMTIN(1) + F*DT
   DO 80 L=2,4
80 AMTIN(L)= AMTIN(L) + F*STREAM(L,J)*DT
90 IF (ISW .EQ. 1) WRITE(6,100) IUNIT, (STREAM(I,J),I=1,4)
100 FORMAT(' UNIT',I5,' SO FLOW=',G10.3,5X,'SS=',
&         G10.3,5X,'DS=',G10.3,5X,' TOC=',G10.3)
   RETURN
   END

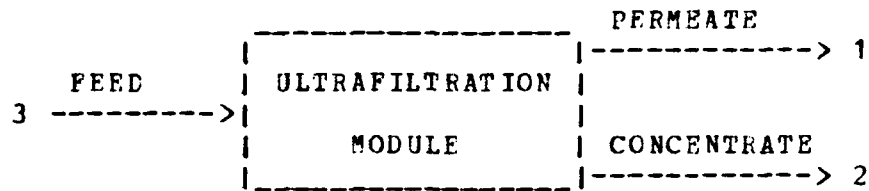
```

C
C SIMULATION OF STREAM MIXER

125

SUBROUTINE UF

ULTRAFILTRATION MODEL INTERFACE TO WPE SIMULATOR



PARAMETER	QUANTITY
1	NUMBER OF TUBES
2	OPERATING TEMPERATURE
3	PRESSURE DROP ACROSS THE MEMBRANE AT THE INLET
4	PRESSURE DROP DOWN THE TUBE
5	TUBE DIAMETER
6	TUBE LENGTH

THE PERMEATE STREAM MUST BE SPECIFIED FIRST
THE CONCENTRATE STREAM MUST BE SPECIFIED SECOND
THE FEED STREAM MUST BE SPECIFIED LAST

```

REAL NTPIDT
DIMENSION PERMN(100)
COMMON /LOOK/ISW
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
      NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
COMMON /UPPARM/ PLEN, DTUBE, NT, JPRINT, JWRITE
COMMON /PARMUP/ TEMP, VISC, DENB, DPZERO, PDROP
IF(NCALL) 10, 80, 90

```

```
C THERE ARE NO MATERIAL BALANCE CALCULATIONS. HOWEVER,  
C THE STREAM SPECIFICATIONS ARE CHECKED FOR CONSISTENT  
C INPUT/OUTPUT  
  10 CONTINUE
```

C SOME ERROR CHECKING

```
IF (ICONFG(3,IUNIT) .LT. 0) GO TO 30
```

```
WRITE(6,20) IUNIT, ICONFG(3,IUNIT)
```

```
20 FORMAT(6X,'*****ERROR, UNIT',I5,'.  FIRST STREAM IN ',  
& 'CONFIGURATION IS',I5,'.  MUST BE THE PERMEATE ',  
& '(OUTPUT).')
```

```

NFATER= NFATER + 1

```

```

30 IF (ICONPG(4,IUNIT) .LT. 0) GO TO 50

```

```
WRITE(6,40) IUNIT,ICONFG(4,IUNIT)
```

```
40 FORMAT(6X,'*****ERROR, UNIT',I5,'. SECOND STREAM IN ',  
  & 'CONFIGURATION IS',I5,'. MUST BE THE CONCENTRATE ',  
  & '(OUTPUT).')
```

NFATER= NFATER + 1

```
50 IF (ICONFG(5,IUNIT) .GT. 0) GO TO 70
```

```
WRITE(6,60) IUNIT,ICONFG(5,IUNIT)
```

```

60 FORMAT(6X,'*****ERROR, UNIT',I5,'.  THIRD STREAM IN ',
      & 'CONFIGURATION IS',I5,'.  MUST BE THE CONCENTRATE ',
      & '(OUTPUT).')

```

```

        NFATER= NFATER + 1
70  CONTINUE
    RETURN
80  CONTINUE
C  INITIALIZATION SAME AS SIMULATE
    PERMN(IUNIT) = 0.
C  RETURN
90  CONTINUE
C  SET UP COMMON VARIABLES
        NT= PAR(NPAR)
        TEMP= PAR(NPAR+1)
        DPZERO= PAR(NPAR+2)
        PDROP= PAR(NPAR+3)
        DTUBE= PAR(NPAR+4)
        PLEN= PAR(NPAR+5)
C  SIMULATE
        IPERM= -ICCNFG(3,IUNIT)
        ICCNC= -ICONFG(4,IUNIT)
        IFEED= ICONFG(5,IUNIT)
C  GET READY TO CALL UFSS
        KULTRA= -1
        CS= STREAM(2,IFEED)
        CD= STREAM(3,IFEED)
        CC= STREAM(4,IFEED)
        FLOW= STREAM(1,IFEED)
        PERM= PERMN(IUNIT)
        IF(ISW .GE. 1)
            & WRITE(6,100) IPERM,ICONC,IFEED,CA,CC,FLOW
100  FORMAT(' $$$ UF DEBUG',6G10.3)
        CALL UFSS(KULTRA,CS,CD,CC,FLOW,TOTALA,TOTALB,TOTALC,
            & PERM)
        PERMN(IUNIT)= PERM
        TOTAL= TOTALA + TOTALB + TOTALC
        STREAM(3,IPERM)= TOTALA/TOTAL*DENB
        STREAM(4,IPERM)= TOTALC/TOTAL*DENB
        STREAM(1,IPERM)= TOTAL/DENB
C  NO SUSPENDED SOLIDS PASS THROUGH UF MEMBRANE
        STREAM(2,IPERM)= 0.
C  CHECK FOR ERRORS
        IF(KULTRA .LT. 0) GO TO 120
        WRITE(6,110) KULTRA
110  FORMAT(' *****ERROR IN UF WHILE CALLING UFSS. ',
            & ' KULTRA=',I5)
        NFATER= NFATER + 1
120  CONTINUE
C  GET FLOW RATE OF CONCENTRATE BY STEADY STATE MATERIAL
C  BALANCE
        STREAM(1,ICONC)= STREAM(1,IFEED) - STREAM(1,IPERM)
C  GET CONCENTRATION OF CONCENTRATE BY COMPONENT BALANCE
        DO 130 I=2,4
            STREAM(I,ICONC)= (STREAM(I,IFEED)*STREAM(1,IFEED) -
                & STREAM(I,IPERM)*STREAM(1,IPERM))/STREAM(1,ICONC)
130  CONTINUE
        IF(ISW .EQ. 1) WRITE(6,140) IUNIT,(STREAM(I,IPERM),I=1

```

```

      8 (STREAM(I,ICCN),I=1,4)
140  FORMAT(' UNIT',I5,' UF  PERM FLOW=',G10.3,5X,'SS=',
      8G10.3,5X,'DS=',G10.3,5X,'TOC=',G10.3/15X,'CONC FLOW=',
      8G10.3,5X,'SS=',G10.3,5X,'DS=',G10.3,5X,'TOC=',G10.3)
      RETURN
      END

```

```

      SUBROUTINE UFSS(KULTRA,SS,DS,TC,F,TOTALA,TOTALB,
&                  TOTALC,PERM)
C
C   THIS SUBROUTINE PERFORMS THE STEADY STATE CALCULATIONS FO
C   MULTIPLE TUBE ULTRAFILTRATION MODULES.
C   THIS ROUTINE ASSUMES COMPLETE REJECTION OF SUSPENDED SOLI
C   AND NO REJECTION OF DESOLVED SOLIDS.
C   SEE THE REPORT BY ABBOTT AND STERLING ON THE MODIFIED UF/
C   TUBULAR RO/ GEL MODEL FOR A DISCRIPTION OF VARIABLES
C
      REAL NTPIDT
      COMMON /UFPARM/ PLEN, DTUBE, NT, JPRINT, JWRITE
      COMMON /UFFIT/  GAM1, GAM2, GAMINF, C1, C2, CINF
      COMMON /PARMUF/ TEMP, VISC, DENB, DPZERO, PDROP
      COMMON /CTIME/  TIME, FTIME, DT
      DATA PIE/3.141593/
      NTPIDT= NT*DTUBE*PIE
      DPBAR= DPZERO - 0.5*PDROP
      GAMMA= GAMINF
      IF (GAM2*PERM .LT. 174.)
& GAMMA= GAMINF + (GAM1-GAMINF)*EXP (-GAM2*PERM)
      TOTALB= PLEN*GAMMA*DPBAR*NTPIDT
      FB= (TOTALB/DENB)
      TOTALA= DS*FB
      TOTAL= TOTALA + TOTALB
      C= CINF
      IF (C2*PERM .LT. 174.) C= CINF + (C1 - CINF)*EXP (-C2*PE
      TOTALC= PLEN*NTPIDT*TC*TOTAL*C/(TOTAL + DENB*C)
      PERM= PERM + FE*DT
10  IF (JPRINT .LE. 0) GO TO 50
      JPRINT= JPRINT - 1
      WRITE(6,20) SS,DS,TC,F,DPZERO,PDROP,PLEN,DT,DTUBE,NTPID
20  FORMAT('      SS= ',G12.5,'      DS= ',G12.5,'      TC= '
* '      F= ',G12.5/
1  ' DPZERO= ',G12.5,' PDROP= ',G12.5,' PLEN= ',G12.5
*,G12.5/
2  ' DTUBE= ',G12.5,' NTPIDT= ',G12.5,' DENB= ',G12.5
3,G12.5)
      WRITE(6,30) GAM1,GAM2,GAMINF,C1,C2,CINF
30  FORMAT('      GAM1= ',G12.5,' GAM2= ',G12.5,' GAMINF= '
* '      C1= ',G12.5,' C2= ',G12.5,' CINF= ',G12.5
      WRITE(6,40) DPBAR,GAMMA,C,PERM,TOTALA,TOTALB,TOTALC
40  FORMAT(' DPBAR= ',G12.5,' GAMMA= ',G12.5,' C= ',G
1  ' PERM= ',G12.5/' TOTALA= ',G12.5,' TOTALB= ',G12.5
2  ' TOTALC= ',G12.5)
50  RETURN
      END

```

```

      SUBROUTINE UOZONE(ZINO,CZINO,TOCINO,G,F,ZOUT,CZOUT,
&                      TOCCUT)
C
C   THIS SUBROUTINE PERFORMS THE STEADY STATE CALCULATIONS
C   FOR THE ULTRAVIOLET OZONATION UNIT
C
      REAL KLA, KHENRY, KRATE, KDCOMP
      COMMON /DELTAI/ DT
      COMMON /UVL/     UVLITE
      COMMON /UVFIT/    KHENRY, ECOZ, ETOC, KRATE, KDCOMP,
&                      EOZD, UVEFCT, ALPHA, FN, QPRIME
      COMMON /UVPARM/ CAREA, PAREA, H, RHO, PRESS, TEMP, NWR
      DIMENSION ZOUT(10), CZOUT(10), TOCOUT(10), DCZBR(10),
&              DTOCER(10)
      COMMON /STAGES/ NSTAGE,PRECON
      DATA IFIRST/0/
      EQUIVALENCE (NNN,IEND)
      NNN= NSTAGE
      IF(PRECON .NE. 0.0) NNN= NSTAGE + 1
      IF(IFIRST .NE. 0) GO TO 20
C   FOR FIRST ITERATION, SET UP INLET GAS CONCENTRATION
C   TO PRE-CONTACTOR
      IFIRST= 1
      TZOUT= 0.
      DO 10 I= 1, NSTAGE
10  TZOUT= TZOUT + ZOUT(I)
      ZBAR= TZOUT/NSTAGE
20 CONTINUE
      UVLITE= 0.
C   CALL FOR PRE-CONTACTOR
      CALL USTAGE(ZBAR, CZINO, TOCINO, NSTAGE*G, F,
& ZOUT(NSTAGE+1), DCZBR(NSTAGE+1), DTOCER(NSTAGE+1),
& PAREA, CZOUT(NSTAGE+1), TOCOUT(NSTAGE+1))
      TZOUT= 0.
C   SET UP UV RADIATION EFFECT ON REACTION RATE CONSTANTS
      UVLITE= UVEFCT
      DO 60 I= 1, NSTAGE
C   SET UP INPUTS TO THE NEXT STAGE
      ZIN= ZINO
      N= I - 1
      IF(I .NE. 1) GO TO 30
      N= NSTAGE + 1
      IF(PRECON .EQ. 0.0) GO TO 40
30 CONTINUE
      CZIN= CZOUT(N)
      TOCIN= TOCOUT(N)
      GO TO 50
40 CONTINUE
      CZIN= CZINO
      TOCIN= TOCINO
50 CONTINUE
      CALL USTAGE(ZIN, CZIN, TOCIN, G, F, ZOUT(I), DCZBR(I),
& DTOCER(I), CAREA, CZOUT(I), TOCOUT(I))

```

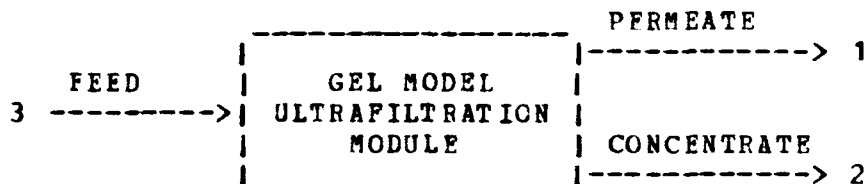
```

      TZOUT= TZOUT + ZOUT(I)
    60 CCNTINUE
  C DETERMINE NEXT GAS CONCENTRATION TO THE PRE-CONTACTOR
      ZEAR= TZOUT/NSTAGE
      DO 70 I= 1, IEND
  C TAKE ONE INTEGRATION STEP
      CZOUT(I) = CZOUT(I) + DCZBR(I)*DT
      70 TOCOUT(I)= TOCCUT(I) + DTOCBR(I)*DT
      RETURN
      END

```

SUBROUTINE GM

ULTRAFILTRATION GEL MODEL INTERFACE TO WPE SIMULATOR



PARAMETER	QUANTITY
1	NUMBER OF TUBES
2	OPERATING TEMPERATURE
3	PRESSURE DROP ACROSS THE MEMBRANE AT THE INLET
4	PRESSURE DROP DOWN THE TUBE
5	TUBE DIAMETER
6	TUBE LENGTH

THE PERMEATE STREAM MUST BE SPECIFIED FIRST
 THE CONCENTRATE STREAM MUST BE SPECIFIED SECOND
 THE FEED STREAM MUST BE SPECIFIED LAST

```

REAL NTPIDT
DIMENSION PERMN(100)
COMMON /LOOK/ISW
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
& NCALL, IUNIT, NPATER, NS, NEQ, DESC(5)
COMMON /GMPARM/ PLEN, DTUBE, NT, JPRINT, JWRITE
COMMON /PARMG/ TEMP, VISC, DENB, DPZERO, PDROP
IF(NCALL) 10, 80, 90
  
```

10 CONTINUE

THERE ARE NO MATERIAL BALANCE CALCULATIONS. HOWEVER, THE
 STREAM SPECIFICATIONS ARE CHECKED FOR CONSISTENT
 INPUT/OUTPUT

```

IF(ICONFG(3,IUNIT) .LT. 0) GO TO 30
WRITE(6,20) IUNIT, ICONFG(3,IUNIT)
20 FORMAT(6X,'*****ERROR, UNIT',I5,'. FIRST STREAM IN ',
& ' CONFIGURATION IS',I5,'. MUST BE THE PERMEATE ',
& '(OUTPUT).')
NPATER= NPATER + 1
30 IF(ICONFG(4,IUNIT) .LT. 0) GO TO 50
WRITE(6,40) IUNIT,ICONFG(4,IUNIT)
40 FORMAT(6X,'*****ERROR, UNIT',I5,'. SECOND STREAM IN',
& ' CONFIGURATION IS',I5,'. MUST BE THE CONCENTRATE',
& '(OUTPUT).')
NPATER= NPATER + 1
50 IF(ICONFG(5,IUNIT) .GT. 0) GO TO 70
WRITE(6,60) IUNIT,ICONFG(5,IUNIT)
60 FORMAT(6X,'*****ERROR, UNIT',I5,'. THIRD STREAM IN',
& ' CONFIGURATION IS',I5,'. MUST BE THE FEED ',
& '(INPUT).')
  
```

INCREMENT NUMBER OF FATAL ERRORS

```

      NFATER= NFATER + 1
70  CONTINUE
      RETURN
80  CONTINUE
C   INITIALIZATION SAME AS SIMULATE
      PERMN(IUNIT) = 0.
90  CONTINUE
C   SET UP COMMON VARIABLES
      NT= PAR(NPAR)
      TEMP= PAR(NPAR+1)
      DPZERO= PAR(NPAR+2)
      PDEOP= PAR(NPAR+3)
      DTUBE= PAR(NPAR+4)
      PLEN= PAR(NPAR+5)
C   SIMULATE
      IPERM= -ICONFG(3,IUNIT)
      ICONC= -ICONFG(4,IUNIT)
      IFEED= ICONFG(5,IUNIT)
C   GET READY TO CALL GMSS
      KULTRA= -1
      CS= STREAM(2,IFEED)
      CD= STREAM(3,IFEED)
      CC= STREAM(4,IFEED)
      FLOW= STREAM(1,IFEED)
      PERM= PERMN(IUNIT)
      IF(ISW .GE. 1)
        & WRITE(6,100) IPERM, ICONC, IFEED, CA, CC, FLOW
100  FORMAT(' $$$ GM DEBUG',6G13.5)
      CALL GMSS(KULTRA,CS,CD,CC,FLOW,TOTALA,TOTALB,TOTALC,
        & PERM)
      PERMN(IUNIT)= PERM
      TOTAL= TOTALA + TOTALB + TOTALC
      STREAM(3,IPERM)= TOTALA/TOTAL*DENB
      STREAM(4,IPERM)= TOTALC/TOTAL*DENB
      STREAM(1,IPERM)= TOTAL/DENB
C   NO SUSPENDED SOLIDS PASS THROUGH UP MEMBRANE
      STREAM(2,IPERM)= 0.
C   CHECK FOR ERRORS
      IF(KULTRA .LT. 0) GO TO 120
      WRITE(6,110) KULTRA
110  FORMAT(' *****ERROR IN GEL WHILE CALLING GMSS.',
        & ' KULTRA=',I5)
      NFATER= NFATER + 1
120  CONTINUE
C   GET FLOW RATE OF CONCENTRATE BY STEADY STATE MATERIAL
C   BALANCE
      STREAM(1,ICONC)= STREAM(1,IFEED) - STREAM(1,IPERM)
C   GET CONCENTRATION OF CONCENTRATE BY COMPONENT BALANCE
      DO 130 I=2,4
        STREAM(I,ICONC)= (STREAM(I,IFEED)*STREAM(1,IFEED) -
          & STREAM(I,IPERM)*STREAM(1,IPERM))/STREAM(1,ICONC)
130  CONTINUE
      IF(ISW .EQ. 1) WRITE(6,140) IUNIT, (STREAM(I,IPERM), I=1
        & (STREAM(I,ICONC), I=1,4)

```

```
140 FORMAT(' UNIT',I5,' GM PERM FLOW=',G10.3,5X,'SS=',  
& G10.3,5X,'DS=',G10.3,5X,'TOC=',G10.3/15X,'CONC FLOW='  
& ,G10.3,5X,'SS=',G10.3,5X,'DS=',G10.3,5X,'TOC=',G10.3)  
RETURN  
END
```

```

      SUBROUTINE GMSS (KULTRA,SS,DS,TC,F,TOTALA,TOTALB,
& TOTALC,PERM)
C
C THIS SUBROUTINE PERFORMS THE STEADY STATE CALCULATIONS
C FOR MULTIPLE TUBE ULTRAFILTRATION MODULES USING THE GEL
C MODEL
C THIS ROUTINE ASSUMES COMPLETE REJECTION OF SUSPENDED
C SOLIDS
C SEE THE REPORT BY ABBOTT AND STERLING ON THE MODIFIED UF/
C TUBULAR RO/ GEL MODEL FOR A DISCRPTION OF VARIABLES
C
      REAL JA,JB,JC,NTPIDT
      COMMON /GMPARM/ PLEN, DTUBE, NT, JPRINT, JWRITE
      COMMON /PARMGH/ TEMP, VISC, DENB, DPZERO, PDROP
      COMMON /CTIME/ TIME, FTIME, DT
      DATA NSTEPS/10/
      DATA NDIM1/100/
      CA= DS
      CC= TC
      PI= 3.141593
C DETERMINE AXIAL STEP SIZE FROM THE TUBE LENGTH AND
C NUMBER OF INTEGRATION STEPS
      DZ= PLEN/NSTEPS
      NWRITE= 0
      AREA= 0.25*PI*DTUBE*DTUBE
      NTPIDT= NT*DTUBE*PI
      TOTALA= 0.
      TOTALB= 0.
      TOTALC= 0.
      DO 30 I=1,NSTEPS
          Z= (I-1)*DZ + 0.5*DZ
C EVALUATE PRESSURE DROP AT MIDPOINT OF INTEGRATION STEP
          DELP= DPZERO - PDROP*Z/PLEN
          V= F/AREA
          KGETJ= 0
C CALL GETJ TO GET THE FLUXES
          CALL GETJ(CA,CC,V,DELP,JA,JB,JC,KGETJ)
C CALCULATE DERIVATIVES FOR THIS STEP
          DF= -NTPIDT*(JA + JB + JC)/DENB
          DFCA= -NTPIDT*JA
          DFCC= -NTPIDT*JC
C PERFORM INTEGRATION
          FNEW= F + DF*DZ
          CA= (F*CA + DFCA*DZ)/FNEW
          CC= (F*CC + DFCC*DZ)/FNEW
C UPDATE CUMULATIVE PERMEATE FLOWS OF A, B, AND C
          TOTALA= TOTALA + JA*NTPIDT*DZ
          TOTALB= TOTALB + JB*NTPIDT*DZ
          TOTALC= TOTALC + JC*NTPIDT*DZ
          F= FNEW
          IF(JPRINT .LE. 0) GO TO 30
C DETERMINE IF ANY OUTPUT IS TO BE PRODUCED
          NWRITE= NWRITE + 1

```

```

      IF (NWRITE .LT. JPRINT) GO TO 30
      NWRITE= 0
      WRITE(6,20) I, Z, CA, CC, JA, JB, JC, F,
&          TCTALA, TOTALB, TOTALC
20  FORMAT(' STEP ',I4,' Z= ',G13.5,' CA= ',G13.5,
&  ' CC= ',G13.5/' JA= ',G13.5,' JB= ',G13.5,' JC= ',
&  ' F= ',G13.5/' TOTALA= ',G13.5,' TOTALB= ',G13.5,
&  ' TOTALC= ',G13.5)
30  CONTINUE
      DS= CA
      TC= CC
      PERM= PERM + (TOTALB/DENB)*DT
      IF (JPRINT .EQ. -999) WRITE(6,20) I, Z, CA, CC, JA, JB,
&          JC, F, TOTALA, TOTALB, TOTALC
      RETURN
      END

```

```

      SUBROUTINE GETJ(CA1,CC1,V,DELP,JA,JB,JC,KGETJ)
C
C THIS IS THE GEL MODEL FOR THE ULTRAFILTRATION UNIT
C SEE REPORT BY SMITH AND STARKS ON THE UF UNIT FOR A
C DISCRIPTION OF VARIABLES
C
      REAL NF, JA, JB, JC
      REAL*8 JB1, JB2, KA, KC, JDA, DAX, RE, SC, SH, CA2,
&          CA2BIG, CA2SML, CA3, CA3BIG, CA3SML, PI,
&          POSMOT, CEQ2, CEQ3, DELPI, PI2, PI3, CA3P,
&          CA2P, XA1, XC1, CX, BX, CCGEL
      COMMON /GMPARM/ PLEN, DTUBE, NT, JPRINT, JWRITE
      COMMON /PARMGH/ TEMP, VISC, DENB, DPZERO, PDROP
      COMMON /GMFIT/ GAMMA, API, BPI, B, C, RATIO, DCX,
&          ADAX, BDAX, CDAX, CAGEL
      DATA MCNT2, MCNT3/10, 10/
      POSMOT(C)= API*TEMP*C*(1.00 + BPI*C)**2
C CHECK FOR REASONABLE INPUT VALUES.
      IF(CA1 .LT. 0.0 .OR. CA1 .GT. DENB) GO TO 10
      IF(CC1 .LT. 0.0 .OR. CC1 .GT. DENB) GO TO 10
      IF(V .LT. 0.0) GO TO 10
      GO TO 30
10 CONTINUE
C IF UNREALISTIC VALUES FROM GMSS ARE ENCOUNTERED, PRINT
C AN ERROR MESSAGE AND DUMP THE MAJOR VARIABLES IN GETJ.
C THEN RETURN.
      WRITE(6,20)CA1,CC1,V
20 FORMAT(' IN GETJ ...UNREALISTIC INPUT VALUES. '/
& ' CA1= ',G15.5,' CC1= ',G15.5,' V= ',G15.5)
      KGETJ= 1
      IF(JWRITE .LE. 0) JWRITE= 1
      GO TO 160
30 CONTINUE
C CALCULATE BULK MASS FRACTIONS
      XA1= CA1/DENB
      XC1= CC1/DENB
C CALCULATE REYNOLDS NUMBER
      RE= DP*V/VISC
      RE913= .0096*RE**.913
C CALCULATE MASS TRANSFER COEFFICIENTS
      XXX= XA1*100.
      DDD= ADAX*XXX + BCAX*EXP(-CDAX*XXX)
      DAX= DDD*1.E-10
      SC= VISC/DAX
      SH= RE913*(SC**.346)
      KA= SH*DAX/DTUBE
      SC= VISC/DCX
      SH= RE913*(SC**.346)
      KC= SH*DCX/DTUBE
C RESET KGETJ TO NON-ERROR CONDITION IF NEEDED.
      IF(KGETJ .LE. 0) GO TO 40
      KGETJ= 0
40 CONTINUE

```

```

C  USE INTERVAL HALVING TO FIND THE LARGEST POSSIBLE VALUE
C  OF CA2, REMEMBERING THAT (DELP - DELPI) MUST ALWAYS BE
C  POSITIVE.
    CA2BIG= DENB
    CA2SML= 0.
C  USE CA2 AS A TEMPORARY STORAGE LOCATION FOR NOW
    CA2= CA1 + CC1*RATIO
    PI2BIG= POSMOT(CA2) + DELP
    DO 60 I=1,20
        CA2= 0.5*(CA2BIG + CA2SML)
        PI2= POSMOT(CA2)
        IF(PI2 .GT. PI2BIG) GO TO 50
    CA2SML= CA2
    GC TO 60
50 CA2BIG= CA2
60 CONTINUE
C  SET UP LIMITS FOR OUTER LOOP TO FIND CA2
    CA2SML= CA1
    DO 130 I=1,MCNT2
        CA2= (CA2BIG + CA2SML)/2.
        JDA= KA*(CA2 - CA1)
C  SET UP LIMITS FOR INNER LOOP TO FIND CA3
    CA3BIG= CA1
    CA3SML= 0.
    DO 100 II=1,MCNT3
        CA3= 0.5D0*(CA3BIG + CA3SML)
        JE1= B*DENB*(CA2/CA3 - 1.0)
        CC2= (JB1*XC1 + KC*CC1)/
        & (KC + C*(1. - DENB*C/(JB1 + DENB*C)))
        CC3= DENB*C*CC2/(JB1 + DENB*C)
        CEQ2= CA2 + CC2*RATIO
        PI2= POSMOT(CEQ2)
        CEQ3= CA3 + CC3*RATIO
        PI3= POSMOT(CEQ3)
        DELPI= PI2 - PI3
        JB2= GAMMA*(DELP - DELPI)
        CA3P= B*DENB*CA2/(JB2 + B*DENB)
        IF(JB2 .LT. 0.0) GO TO 70
        IF(CA3 .GT. CA3P) GO TO 80
70 CA3SML= CA3
    GO TO 90
80 CA3BIG= CA3
90 CONTINUE
100 CONTINUE
    JB= 0.5D0*(JB1 + JB2)
    CA2P= (JB*XA1 + KA*CA1 + B*CA3)/(KA + B)
    IF(CA2 .GT. CA2P) GO TO 110
    CA2SML= CA2
    GO TO 120
110 CA2BIG= CA2
120 CONTINUE
130 CONTINUE
    JA= B*(CA2 - CA3)
    JC= C*(CC2 - CC3)

```

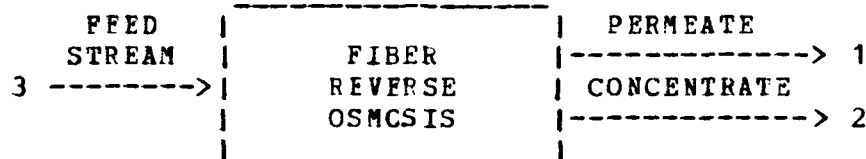
```

IF (CAGEL .LT. 0.) GO TO 150
IF (CA2 .LE. CAGEL) GO TO 150
  JLA= KA*(CAGEL - CA1)
  BX= JLA*(1. - E/KA)
  CX= BX*CAGEL*JDA
  JA= (-BX + DSQRT(BX*BX - 4.*CX)) *0.5D0
  JB= (JA + JDA)/XA1
  CA3= DENB*JA/JB
  CX= C/(1. + DENB*C/JB)
  CCGEL= (JB*XC1 + KC*CC1)/(CX + KC)
  JC= CX*CCGEL
  JDC= KC*(CCGEL - CC1)
  CC3= DENB*JC/JB
  CEQ2= CAGEL + CCGEL*RATIO
  PI2= POSMOT(CEQ2)
  CEQ3= CA3 + CA3 + CC3*RATIO
  PI3= PCSMCT(CEQ3)
  DELPI= PI2 - PI3
  Q= GAMMA*(DELP - DELPI)/JB
  IF(Q .LT. 1.) WRITE(6,140) Q
140 FORMAT(' IN GETJ ... Q= ',G13.5,' IS LESS THAN 1.')
150 CONTINUE
160 CONTINUE
C THIS SECTION OF WRITE STATEMENTS PROVIDES A DUMP OF MAJOR
C VARIABLES IN GETJ
  IF(JWRITE .LE. 0) GO TO 200
  JPRINT= JPRINT + 1
  IF(JPRINT .GE. JWRITE) JWRITE= - 1000
  WRITE(6,170) JPRINT
170 FORMAT(' ENTER GETJ ...PASS NUMBER ',I10)
  WRITE(6,190) CA1, CC1, V, DP, DENB
  WRITE(6,190) TEMP, MCNT2, MCNT3, VISC, DELP
  WRITE(6,190) JWRITE, JA, JB, JC
  WRITE(6,190) AKA, AKC, ERE, API, BPI
  WRITE(6,190) GAMMA, B, C, NF, RATIO
  WRITE(6,190) RE, KA, KC, DELPI
  WRITE(6,190) CA2, CC2, CEQ2, PI2
  WRITE(6,190) CA3, CC3, CEQ3, PI3
  WRITE(6,190) CA2SML, CA2BIG, CC2SML, CC2BIG
  WRITE(6,190) CA3SML, CA3BIG, CC3SML, CC3BIG
  WRITE(6,190) CCGEL, Q
  WRITE(6,180)
180 FORMAT(1X,10(' '), ' LEAVING GETJ ',30(' '))
190 FORMAT(5(1X,G15.5))
200 CONTINUE
  RETURN
  END

```

SUBROUTINE RO

THIS IS THE FIBER REVERSE OSMOSIS INTERFACE



PARAMETER	QUANTITY
1	OPERATING PRESSURE
2	OPERATING TEMPERATURE
3	LENGTH OF A FIBER
4	OUTER RADIUS OF FIBER BUNDLE
5	INNER RADIUS OF FIBER BUNDLE
6	FIBER DIAMETER

THE PERMEATE STREAM MUST BE SPECIFIED FIRST
 THE CONCENTRATE STREAM MUST BE SPECIFIED SECOND
 THE FEED STREAM MUST BE SPECIFIED THIRD

```

REAL L,NF
COMMON /LOOK/ ISW
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
& NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
COMMON /ROFIT/ AKA, AKC, ERE, API, BPI, GAMMA, B, C,
& NF, RATIO
COMMON /PARMO/ L, OR, RI, DF, TOLMX, TOLMN, NWRITE,
& NSTEPS
COMMON /ROPARM/ TEMP, VISC, DELP, RHOB, MCNT2, MCNT3,
& JWITE
IF(NCALL) 10, 80, 90
10 CONTINUE
C SOME ERROR CHECKING
IF(ICONFG(3,IUNIT) .LT. 0) GO TO 30
WRITE(6,20) IUNIT, ICONFG(3,IUNIT)
20 FORMAT(6X,'*****ERROR, UNIT',I5,'. FIRST STREAM IN',
& ' CONFIGURATION IS',I5,'. MUST BE PERMEATE (OUTPUT.)')
NFATER= NFATER + 1
30 IF(ICONFG(4,IUNIT) .LT. 0) GO TO 50
WRITE(6,40) IUNIT, ICONFG(4,IUNIT)
40 FORMAT(6X,'*****ERROR, UNIT',I5,'. SECOND STREAM IN',
& ' CONFIGURATION IS',I5,'. MUST BE THE CONCENTRATE',
& ' (OUTPUT.)')
NFATER= NFATER + 1
50 IF(ICONFG(5,IUNIT) .GT. 0) GO TO 70
WRITE(6,60) IUNIT, ICONFG(5,IUNIT)
60 FORMAT(6X,'*****ERROR, UNIT',I5,'. THIRD STREAM IN',
& ' CONFIGURATION IS',I5,'. MUST BE THE FEED (INPUT.)')
NFATER= NFATER + 1
70 CCNTINUE
RETURN
  
```

```

      80 CONTINUE
C   INITIALIZATION IS THE SAME AS SIMULATE
C   RETURN
      90 CONTINUE
C   SIMULATE
      DELP= PAR(NPAR)
      TEMP= PAR(NPAR+1)
      L= PAR(NPAR+2)
      CR= PAR(NPAR+3)
      RI= PAR(NPAR+4)
      DF= PAR(NPAR+5)
      IPERM= -ICONFG(3,IUNIT)
      ICCNC= -ICONFG(4,IUNIT)
      IFEED= ICONFG(5,IUNIT)
C   GET READY TO CALL ROSS
      FLOW= STREAM(1,IFEED)
      CAR= STREAM(3,IFEED)
      CARSAV= CAR
      CCR= STREAM(4,IFEED)
      CCRSAV= CCR
      CALL ROSS(FLOW,RHOB,CAR,CCR,SOUTA,SOUTB,SOUTC)
      TOUT= SCUTA + SOUTB + SOUTC
      XAP= SOUTA/TOUT
      XCP= SOUTC/TOUT
      STREAM(3,IPERM)= XAP*RHOB
      STREAM(4,IPERM)= XCP*RHOB
      STREAM(2,IPERM)= 0.
      FLOWC=FLOW-TOUT/RHOB
      CAC= (FLOW*CARSAV-SOUTA)/FLOWC
      CCC= (FLOW*CCRSAV-SOUTC)/FLOWC
      STREAM(3,ICONC)= CAC
      STREAM(4,ICONC)= CCC
C   NO SUSPENDED SOLIDS IN THE PERMEATE
      STREAM(2,ICONC)= FLOW*STREAM(2,IFEED)/FLOWC
      STREAM(1,ICONC)= FLOWC
      STREAM(1,IPERM)= FLOW - FLOWC
      IF(ISW.NE.0) WRITE(6,100)
100  FORMAT(' LEAVING RO')
      IF(ISW.EQ.1) WRITE(6,110) IUNIT, (STREAM(I,IPERM), I=1,4)
      & (STREAM(I,ICONC), I=1,4)
110  FORMAT(' UNIT',I5,' RO PERM FLOW=',G10.3,5X,'SS=',
      & G10.3,5X,'DS=',G10.3,5X,'TOC=',G10.3/15X,'CONC FLOW=',
      & G10.3,5X,'SS=',G10.3,5X,'DS=',G10.5,'TOC=',G10.3)
      RETURN
      END

```

```

      SUBROUTINE ROSS (FLOW, RHOB, SCAR, SCCR, SOUTA, SOUTB, SOUTC)
C
C THIS SUBROUTINE PERFORMS THE STEADY STATE RO CALCULATIONS
C
      REAL*8 R, JA, JB, JC, VR, DR, DRSAVE, CAR, CCR,
&          OUTASV, OUTESV, OUTCSV, ATOTAL, RDR, AUX,
&          VRDR, CARDR, CCRDR, RELCA, RELCC, RELV,
&          OUTA, OUTB, OUTC
      REAL NF, L
      EQUIVALENCE (IXCNT, IHALVE)
      COMMON /ROFIT/ AKA, AKC, ERE, API, BPI, GAMMA, B, C,
&                  NF, RATIO
      COMMON /FASIRO/ IFLAGF
      CMMCN /PARMRC/ L, RO, RI, DF, TOLMX, TOLMN, NWRITE,
&                  NSTEPS
      DATA PI/3.141593/
      DATA IVFRST/0/
      DATA IERROR/0/
C INITIALIZE COUNTERS
      IFLAGF= 0
      IF (IVFRST .NE. 0) GO TO 10
      IVFRST= 1
      IFLAGF= 1
10  CONTINUE
      CAR= SCAR
      CCR= SCCR
      IHALVE= 0
      J= 0
      I= 0
      OUTA= 0.
      OUTB= 0.
      OUTC= 0.
      R= RI
      ATOTAL= PI*(RO*RC - BI*RI)
      VR= FLOW/(2.*PI*RI*L)
      DR= (RO - RI)/NSTEPS
      DRSAVE= DR
      KGETF= 10
20  CONTINUE
C START OF THE INTEGRATION LOOP
30  CONTINUE
C GETF CALCULATES VALUES OF THE FLUXES
      CALL GETF(CAR, CCR, VR, DF, JA, JB, JC, KGETF)
C CHECK FOR ERRORS IN GETF
      IF (KGETF .EQ. 0) GO TO 100
      IERROR= IERROR + 1
      WRITE(6,40)
40  FORMAT(' IN ROSS...RESPONDING TO AN ERROR IN GETF')
      WRITE(6,130) I, IXCNT, R, CAR, CCR, OUTA, OUTB, OUTC,
&                VR, AUX, JA, JB, JC
      KGETF= 0
50  CONTINUE
C HALVE THE STEP SIZE

```

```

      IHALVE= IHALVE + 1
      DR= DR/2.
C CHECK FOR EXCESSIVE STEP SIZE HALVINGS AND
C ERRORS FROM GETF
      IF(IHALVE .LE. 20) GO TO 70
      WRITE(6,60)
60  FORMAT(' IN ROSS. AN EXCESSIVE NUMBER OF RADIAL STEP'
      $,' SIZE HALVINGS HAVE OCCURRED.')
      STOP
70  CONTINUE
      IF(IEERROR .LE. 100) GO TO 90
      WRITE(6,80)
80  FORMAT(' IN ROSS. AN EXCESSIVE NUMBER OF ERRORS HAVE'
      $,' BEEN REPORTED FROM GETF')
      STOP
90  CONTINUE
100 CONTINUE
C ONE INTEGRATION STEP
      RDR= R + DR
      AUX= NF*DF*PI*DR*(2.*R+DR)
      VRDR= (2.*R*RHCB*VR- (JA+JB+JC) *AUX)/(2.*RDR*RHOB)
      CARDR= (2.*R*VR*CAR-JA*AUX)/(2.*RDR*VRDR)
      CCRDR= (2.*R*VR*CCR-JC*AUX)/(2.*RDR*VRDR)
C CALCULATE THE RELATIVE CHANGE IN VARIABLES OVER THE
C INTEGRATION STEP
      IF(VR .NE. 0.) RELV= (VRDR-VR)/VR
      IF(VR .EQ. 0.) RELV= VRDR-VR
      IF(CAR .NE. 0.) RELCA= (CARDR-CAR)/CAR
      IF(CAR .EQ. 0.) RELCA= CARDR-CAR
      IF(CCR .NE. 0.) RELCC= (CCRDR-CCR)/CCR
      IF(CCR .EQ. 0.) RELCC= CCRDR-CCR
C CHECK FOR UNREALISTIC VALUES OF VARIABLES
      IF(VRDR .LT. 0.0) GO TO 50
      IF(CARDR .LT. 0.0) GO TO 50
      IF(CCRDR .LT. 0.0) GO TO 50
C IF GRADIENTS ARE TOO STEEP, HALVE THE STEP SIZE
      IF(RELV .GT. TOLMX .OR. (-RELV).GT.TOLMX) GO TO 50
      IF(RELCA .GT. TOLMX .OR. (-RELCA).GT.TOLMX) GO TO 50
      IF(RELCC .GT. TOLMX .OR. (-RELCC).GT.TOLMX) GO TO 50
C IF GRADIENTS ARE VERY SMALL, DOUBLE STEP SIZE
      IF(RELCA .GT. TOLMN .CR. (-RELCA) .GT. TOLMN) GO TO 11
      IF(RELCC .GT. TOLMN .OR. (-RELCC) .GT. TOLMN) GO TO 11
      IHALVE= IHALVE - 1
      DR= DR*2.
C NEVER LET STEP SIZE BECOME LARGER THAN THE ORIGINAL ONE
      IF(DR.LT.DRSAVE ) GO TO 110
      DR= DRSAVE
      IHALVE= 0
110 CONTINUE
C UPDATA PERMEATE FLOW RATES
      OUTASV= CUTA
      OUTBSV= OUTB
      OUTCSV= OUTC
      OUTA= OUTA + JA*AUX*L*PI

```

```

      OUTB= OUTB + JE*AUX*L*PI
      OUTC= OUTC + JC*AUX*L*PI
C   GET READY FOR NEXT INTEGRATION STEP
      I= I + 1
      R= RDR
C   SAVE CURRENT VALUES OF VARIABLES FOR USE LATER
      VSAVE= VR
      CASAVE= CAR
      CCSAVE= CCR
      VR= VRDR
      CAR= CARDR
      CCR= CCRDR
C   DETERMINE WHETHER ANY OUTPUT AT THIS STEP IS REQUIRED
      IF(NWRITE .LE. 0) GO TO 140
      J= J + 1
      IF(J.LT.NWRITE) GO TO 140
      J= 0
120  CONTINUE
      WRITE(6,130) I, IXCNT, R, CAR, CCR, OUTA, OUTB, OUTC,
&      VR, AUX, JA, JB, JC
130  FORMAT(' IN ROSS ',2I3,(/5(1X,E11.4)))
      IF(R .EQ. RO) GO TO 150
140  CCNTINUE
      IF(R .LT. RO-DR/10.) GO TO 20
C   USE LINEAR INTERPOLATION TO FIND VALUES OF CAR,CCR,VR
C   EXACTLY AT RO
      VR= ((VRDR-VSAVE)/DR)*(RO-(RDR-DR)) + VSAVE
      CAR= ((CARDR-CASAVE)/DR)*(RO-(RDR-DR)) + CASAVE
      CCR= ((CCRDR-CCSAVE)/DR)*(RO-(RDR-DR)) + CCSAVE
      OUTA= ((OUTA-OUTASV)/DR)*(RO-(RDR-DR)) + OUTASV
      OUTB= ((OUTB-OUTBSV)/DR)*(RO-(RDR-DR)) + OUTBSV
      OUTC= ((OUTC-OUTCSV)/DR)*(RO-(RDR-DR)) + OUTCSV
      R= RO
      IF(NWRITE .EQ. -999 .OR. NWRITE .GT. 0) GO TO 120
150  CONTINUE
C   UNSAVE THE ORIGINAL STEP SIZE
      DR= DRSAVE
      SOUTA= OUTA
      SOUTB= OUTB
      SOUTC= OUTC
      SCAR= CAR
      SCCR= CCR
      RETURN
      END

```

```

      SUBROUTINE GETF(CA1,CC1,V,DF,JA,JB,JC,KGETF)
C
C   THIS SUBROUTINE COMPUTES THE FLUXES FOR THE
C   REVERSE OSMOSIS PROGRAM
C   SEE REPORT BY SMITH AND STARKS ON THE RO UNIT FOR A
C   DISCUSSION OF VARIABLES
C
      REAL*8 JA, JB, JC, CA1, CC1, V, XA1, XC1, CA2BIG,
&          CA2SML, PI2BIG, PI, POSMOT, CA2, JDA, CA3BIG,
&          CA3SML, CA3, JB1, CC2, CC3, CEQ2, PI2, CEQ3,
&          PI3, DELPI, JB2, CA3P, CA2P, CONC
      REAL KA,KC,NF
      COMMON /ROPARM/ TEMP, VISC, DELP, RHO, MCNT2, MCNT3,
&          JWRITE
      COMMON /ROFIT/  AKA, AKC, ERE, API, BPI, GAMMA, B, C,
&          NF, RATIO
      COMMON /PASTE0/ IFLAG
      DATA TOLER/0.001/
      DATA JPRINT/0/
      POSMOT(CONC)= API*TEMP*CONC*(1.0+BPI*CONC)**2
      IF(CA1 .LT. 0.0 .OR. CA1 .GT. RHO) GO TO 10
      IF(CC1 .LT. 0.0 .OR. CC1 .GT. RHO) GO TO 10
      IF(V .LT. 0.0) GO TO 10
      GO TO 30
10  CONTINUE
C   IF UNREALISTIC VALUES FROM ROSS ARE ENCOUNTERED, PRINT
C   AN ERROR MESSAGE AND DUMP THE MAJOR VARIABLES IN GETF.
C   THEN RETURN.
      WRITE(6,20) CA1,CC1,V
20  FORMAT(' IN GETF ...UNREALISTIC INPUT VALUES.',
& ' CA1= ',G15.5,'      CC1= ',G15.5,'      V= ',G15.5)
      KGETF= 1
      IF(JWRITE .LE. 0) JWRITE= 1
      GO TO 140
30  CONTINUE
C   CALCULATE REYNOLDS NUMBER
      RE= DF*V/VISC
C   CALCULATE MASS TRANSFER COEFFICIENTS
      KA= AKA*RE**ERE
      KC= AKC*RE**ERE
C   RESET KGETF TO NON-ERROR CONDITION IF NEEDED.
      IF(KGETF .LE. 0) GO TO 40
      KGETF= 0
40  CONTINUE
C   CALCULATE BULK MASS FRACTIONS
      XA1= CA1/RHO
      XC1= CC1/RHO
      IF(IFLAG .EQ. 0) GO TO 190
C   USE INTERVAL HALVING TO FIND THE LARGEST POSSIBLE VALUE
C   OF CA2, REMEMBERING THAT (DELP-DELPI) MUST ALWAYS BE
C   POSITIVE.
      CA2BIG= RHO
      CA2SML= 0.

```

```

C  USE CA2 AS A TEMPORARY STORAGE LOCATION FOR NOW
    CA2= CA1 + CC1*RATIO
    PI2BIG= POSMOT(CA2) + DELP
    DO 60 I= 1,20
        CA2= (CA2BIG + CA2SML)/2.
        PI2= POSMOT(CA2)
        IF(PI2 .GT. PI2BIG) GO TO 50
        CA2SML= CA2
        GO TO 60
    50 CA2BIG= CA2
    60 CONTINUE
C  USE DOUBLLY NESTED INTERVAL HALVING TO SOLVE FOR
C  CA2 AND CA3
C  SET UP LIMITS FOR OUTER LOOP TO FIND CA2
    CA2SML= CA1
    CA2BIG= CA2
    DO 130 I= 1,MCNT2
        CA2= (CA2BIG + CA2SML)/2.
        JCA= KA*(CA2 - CA1)
C  SET UP LIMITS FOR INNER LOOP TO FIND CA3
    CA3BIG= CA1
    CA3SML= 0.
    DO 100 II= 1,MCNT3
        CA3= (CA3BIG + CA3SML)/2.
        JB1= B*RHO*(CA2/CA3 - 1.0)
        CC2= (JB1*XC1+KC*CC1)/(KC+C*(1.-RHO*C/(JB1+RHO*C)))
        CC3= RHO*C*CC2/(JB1+RHO*C)
        CEQ2= CA2 + CC2*RATIO
        PI2= POSMOT(CEQ2)
        CEQ3= CA3 + CC3*RATIO
        PI3= POSMOT(CEQ3)
        DELFI= PI2 - PI3
        JB2= GAMMA*(DELP - DELPI)
        CA3P= B*RHO*CA2/(JB2 + B*RHO)
        IF(JB2 .LT. 0.0) GO TO 70
        IF(CA3 .GT. CA3P) GO TO 80
    70 CA3SML= CA3
        GO TO 90
    80 CA3BIG= CA3
    90 CCNTINUE
    100 CONTINUE
        JB= (JB1 + JB2)/2.
        CA2P= (JB*XA1 + KA*CA1 + B*CA3)/(KA + B)
        IF(CA2 .GT. CA2P) GO TO 110
        CA2SML= CA2
        GO TO 120
    110 CA2BIG= CA2
    120 CONTINUE
    130 CONTINUE
        JA= B*(CA2 - CA3)
        JC= C*(CC2 - CC3)
    140 CONTINUE
C  THIS SECTION OF WRITE STATEMENTS PROVIDES A DUMP OF MAJOR
C  VARIABLES IN GETF

```

```

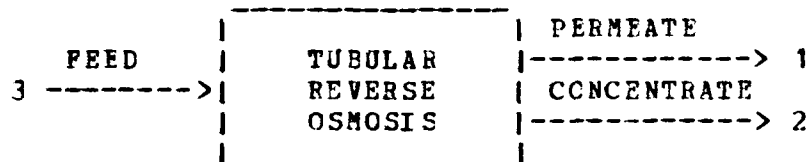
      IF(JWRITE .LE. 0) GO TO 180
      JPRINT= JPRINT + 1
      IF(JPRINT .GE. JWRITE) JWRITE=-1000
      WRITE(6,150) JPRINT
150  FORMAT(' ENTER GETF ...PASS NUMBER ',I10)
      WRITE(6,170) CA1, CC1, V, DF, RHO
      WRITE(6,170) TEMP, MCNT2, MCNT3, VISC, DELP
      WRITE(6,170) JWRITE, JA, JB, JC
      WRITE(6,170) AKA, AKC, ERE, API, BPI
      WRITE(6,170) GAMMA, B, C, NF, RATIO
      WRITE(6,170) RE, KA, KC, DELPI
      WRITE(6,170) CA2, CC2, CEQ2, PI2
      WRITE(6,170) CA3, CC3, CEQ3, PI3
      WRITE(6,170) CA2SML, CA2BIG, CC2SML, CC2BIG
      WRITE(6,170) CA3SML, CA3BIG, CC3SML, CC3BIG
      WRITE(6,160)
160  FORMAT(1X,10(' '), ' LEAVING GETF ',30(' '))
170  FORMAT(5(1X,G15.5))
180  CONTINUE
      RETURN
190  CONTINUE
C   TRY BACK SUBSTITUTION
      DO 200 I=1,15
        CA2L= CA2
        CA3L= CA3
        CC2L= CC2
        CC3L= CC3
        CEQ2= CA2 + RATIO*CC2
        PI2= POSMOT(CEQ2)
        CEQ3= CA3 + RATIO*CC3
        PI3= POSMOT(CEQ3)
        DELPI= PI2 - PI3
      IF(DELPI .GT. DELP) DELPI= 0.3*DELP
        JB= GAMMA*(DELP-DELPI)
        JA= B*(CA2-CA3)
        JC= C*(CC2-CC3)
        TOTLJ= JA+JC+JB
        CA3= JA/TOTLJ
        CC3= JC/TOTLJ
        CA2= (TOTLJ*XA1+KA*CA1+B*CA3)/(KA+B)
        CC2= (TOTLJ*XC1+KC*CC1+C*CC3)/(KC+C)
        RELCA3= (CA3-CA3L)/CA3
        RELCC3= (CC3-CC3L)/CC3
        RELCA2= (CA2-CA2L)/CA2
        RELCC2= (CC2-CC2L)/CC2
      IF(RELCA3 .LT. 0.) RELCA3= -RELCA3
      IF(RELCC3 .LT. 0.) RELCC3= -RELCC3
      IF(RELCA2 .LT. 0.) RELCA2= -RELCA2
      IF(RELCC2 .LT. 0.) RELCC2= -RELCC2
      IF(RELCA3 .GT. TOLER) GO TO 200
      IF(RELCC3 .GT. TOLER) GO TO 200
      IF(RELCC2 .GT. TOLER) GO TO 200
      IF(RELCA2 .GT. TOLER) GO TO 200
      GC TO 140

```

```
200 CONTINUE  
    IFLAG= 1  
    GO TO 30  
END
```

SUBROUTINE TR

TUBULAR REVERSE OSMOSIS MODEL INTERFACE TO WPE SIMULATOR



PARAMETER	QUANTITY
1	NUMBER OF TUBES
2	OPERATING TEMPERATURE
3	PRESSURE DROP ACROSS THE MEMBRANE AT THE INLET
4	PRESSURE DROP DOWN THE TUBE
5	DIAMETER OF TUBE
6	TUBE LENGTH

THE PERMEATE STREAM MUST BE SPECIFIED FIRST
 THE CONCENTRATE STREAM MUST BE SPECIFIED SECOND
 THE FEED STREAM MUST BE SPECIFIED THIRD

```

REAL NTPIDT
DIMENSION PERMN(100)
COMMON /LOOK/ ISW
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
& NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
COMMON /TRPARM/ PLEN, DTUBE, NT, JPRINT, JWRITE
COMMON /PARMTR/ TEMP, VISC, DENB, DPZERO, PDROP
IF(NCALL) 10, 80, 90
  
```

THERE ARE NO MATERIAL BALANCE CALCULATIONS. HOWEVER, THE
 STREAM SPECIFICATIONS ARE CHECKED FOR CONSISTENT
 INPUT/OUTPUT

10 CONTINUE

SOME ERROR CHECKING

```

IF(ICONFG(3,IUNIT) .LT. 0) GO TO 30
WRITE(6,20) IUNIT, ICONFG(3,IUNIT)
  
```

```

20 FORMAT(6X,'*****ERROR, UNIT',I5,'. FIRST STREAM IN',
& ' CONFIGURATION IS',I5,'. MUST BE THE PERMEATE',
& ' (OUTPUT).')
  
```

NFATER= NFATER + 1

```

30 IF(ICONFG(4,IUNIT) .LT. 0) GO TO 50
WRITE(6,40) IUNIT, ICONFG(4,IUNIT)
  
```

```

40 FORMAT(6X,'*****ERROR, UNIT',I5,'. SECOND STREAM IN',
& ' CONFIGURATION IS',I5,'. MUST BE THE CONCENTRATE',
& ' (OUTPUT).')
  
```

NFATER= NFATER + 1

```

50 IF(ICONFG(5,IUNIT) .GT. 0) GO TO 70
WRITE(6,60) IUNIT, ICONFG(5,IUNIT)
  
```

```

60 FORMAT(6X,'*****ERROR, UNIT',I5,'. THIRD STREAM IN',
& ' CONFIGURATION IS',I5,'. MUST BE THE FEED (INPUT).')
NFATER= NFATER + 1
  
```

```

70 CONTINUE
  RETURN
80 CCNTINUE
C  INITIALIZATION SAME AS SIMULATE
  PERMN(IUNIT) = C.
C  RETURN
90 CONTINUE
C  SET UP COMMON VARIABLES
  NT= PAR(NPAR)
  TEMP= PAR(NPAR+1)
  DPZERO= PAR(NPAR+2)
  PDROP= PAR(NPAR+3)
  DTUPE= PAR(NPAR+4)
  PLEN= PAR(NPAR+5)
C  SIMULATE
  IPERM= -ICONFG(3,IUNIT)
  ICONC= -ICONFG(4,IUNIT)
  IFEEF= ICONFG(5,IUNIT)
C  GET READY TO CALL TRSS
  KULTRA= -1
  CS= STREAM(2,IFEEF)
  CD= STREAM(3,IFEEF)
  CC= STREAM(4,IFEEF)
  FLOW= STREAM(1,IFEEF)
  PERM= PERMN(IUNIT)
  IF (ISW .GE. 1)
    & WRITE(6,100) IPERM, ICONC, IFEEF, CA, CC, FLOW
100 FORMAT(' $$$ TR DEBUG',6G10.3)
  CALL TRSS(KULTRA,CS,CD,CC,FLOW,TOTALA,TOTALB,
    & TOTALC,PERM)
  PERMN(IUNIT) = PERM
  TOTAL= TOTALA + TOTALB + TOTALC
  STREAM(3,IPERM) = TOTALA/TOTAL*DENB
  STREAM(4,IPERM) = TOTALC/TOTAL*DENB
  STREAM(1,IPERM) = TOTAL/DENB
C  NO SUSPENDED SOLIDS PASS THROUGH TR MEMBRANE
  STREAM(2,IPERM) = 0.
C  CHECK FOR ERRORS
  IF(KULTRA .LT. 0) GO TO 120
  WRITE(6,110) KULTRA
110 FORMAT(' *****ERROR IN TR WHILE CALLING TRSS.',
    & ' KULTRA=',I5)
  NFATER= NFATER + 1
120 CONTINUE
C  GET FLOW RATE OF CONC. BY STEADY STATE MATERIAL BALANCE
  STREAM(1,ICONC) = STREAM(1,IFEEF) - STREAM(1,IPERM)
C  GET CONCENTRATION OF CONCENTRATE BY COMPONENT BALANCE
  DO 130 I=2,4
    STREAM(I,ICONC) = (STREAM(I,IFEEF)*STREAM(1,IFEEF) -
      & STREAM(I,IPERM)*STREAM(1,IPERM))/STREAM(1,ICONC)
130 CONTINUE
  IF (ISW.EQ.1) WRITE(6,140) IUNIT, (STREAM(I,IPERM), I=1,4)
  & (STREAM(I,ICONC), I=1,4)
140 FORMAT(' UNIT',I5,' TR PERM FLOW=',G10.3,5X,'SS=',

```

```
&G10.3,5X,'DS=',G10.3,5X,'TOC=',G10.3/15X,'CONC FLOW=',  
& G10.3,5X,'SS=',G10.3,5X,'DS=',G10.3,5X,'TOC=',G10.3)  
  RETURN  
  END
```

```

      SUBROUTINE TRSS (KULTRA,SS,DS,TC,F,TOTALA,TOTALB,
&                     TOTALC,PERM)
C
C   THIS SUBROUTINE PERFORMS THE STEADY STATE CALCULATIONS
C   FOR TUBULAR REVERSE OSMOSIS MODULES.
C   THIS ROUTINE ASSUMES COMPLETE REJECTION OF SUSPENDED
C   SOLIDS AND HIGH REJECTION OF DESOLVED SOLIDS.
C   SEE THE REPORT BY ABBOTT AND STERLING ON THE MODIFIED UF/
C   TUBULAR RO/ GEL MODEL FOR A DISCRIPTION OF VARIABLES
C
      REAL NTPIDT
      REAL*8 RE, SCA, SHA, KA, SCC, SHC, KC
      COMMON /TRPARM/ PLEN, DTUBE, NT, JPRINT, JWRITE
      COMMON /TRFIT/  GAM1, GAM2, GAMINF, API, B, C, DCX,
&                     ADAX, BDAX, CDAX
      COMMON /PARMTR/ TEMP, VISC, DENB, DPZERO, PDROP
      COMMON /CTIME/  TIME, FTIME, DT
      DATA PIE/3.141593/
C   COMPUTE NT TIMES THE CIRCUMFERENCE OF ALL TUBES
      NTPIDT= NT*DTUBE*PIE
C   COMPUTE THE CROSS-SECTIONAL AREA OF A TUBE
      AREA= PIE*DTUBE*DTUBE/4.
C   COMPUTE THE VELOCITY THROUGH A TUBE
      V= F/(AREA*NT)
      CA= DS
C   COMPUTE THE WEIGHT FRACTION OF TDS
      XA1= DS/DENB
C   COMPUTE THE WEIGHT FRACTION OF TOC
      XC1= TC/DENB
C   COMPUTE THE REYNOLDS NUMBER
      RE= DTUBE*V/VISC
      RE913= 0.0096*RE**.913
      XXX= XA1*100.
      DD= ACAX*XXX
      IF (CDAX*XXX .LT. 174.) DDD= ADAX*XXX + BDAX*EXP(-CDAX*
      DAX= DDD*1.E-10
      IF (DAX .NE. 0.) GO TO 10
      KULTRA= 1
      GO TO 100
10      SCA= VISC/DAX
      IF (SCA .GT. 0.) GO TO 20
      KULTRA= 2
      GO TO 100
20      SHA= RE913*(SCA**.346)
      KA= SHA*DAX/DTUBE
      IF (DCX .NE. 0.) GO TO 30
      KULTRA= 3
      GO TO 100
30      SCC= VISC/DCX
      IF (SCC .GT. 0.) GO TO 40
      KULTRA= 4
      GO TO 100
40      SHC= RE913*(SCC**.346)

```

```

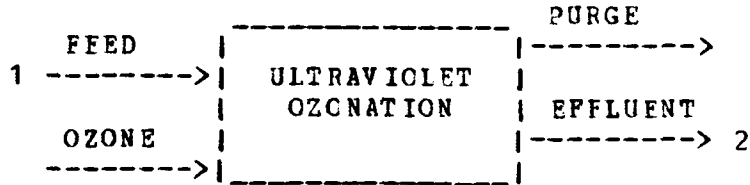
      KC= SHC*DCX/ETUBE
      DPBAR= DPZERO - 0.5*PDROP
      POS= API*TEMP*CA
      IF (POS .LE. DPBAR) GO TO 50
      KULTRA= 5
      GC TC 100
50  GAMMA= GAMINF
      IF (GAM2*PERM .LT. 174)
& GAMMA= GAMINF + (GAM1 - GAMINF)*EXP(-GAM2*PERM)
      TERM1= GAMMA*DPBAR/DENB + KA
      TERM2= GAMMA*POS/DENB + B + KA
      IF (TERM2 .NE. 0.) GO TO 60
      KULTRA= 6
      GO TO 100
60  TERM12= TERM1/TERM2
      TOTALB= PLEN*GAMMA*(DPBAR - POS*TERM12)*NTPIDT
      PERM= PERM + (TOTALB/DENB)*DT
      TOTALA= B*DS*PLEN*TERM12*NTPIDT
      TOTALC= C*TC/(C + KC)*(KC*PLEN*NTPIDT + TOTALB/DENB)
      IF (TOTALA .LT. 0.) WRITE(6,70) TOTALA, B, TERM12
70  FORMAT(' *****TOTALA= ',G12.5,' B= ',G12.5,
& ' TERM12= ',G12.5)
      IF (TOTALB .LT. 0.) WRITE(6,80) TOTALB,GAMMA,POS,TERM12
80  FORMAT(' *****TOTALB= ',G12.5,' GAMMA= ',G12.5,
& ' POS= ',G12.5,' TERM12= ',G12.5)
      IF (TOTALC .LT. 0.) WRITE(6,90) TOTALC,C,KC
90  FORMAT(' *****TOTALC= ',G12.5,' C= ',G12.5,
& ' KC= ',G12.5)
100 IF(JPRINT .LE. 0) RETURN
      JPRINT= JPRINT - 1
      WRITE(6,110) SS, DS, TC, F, DPZERO, PDROP, PLEN, DT,
& DTUBE, NTPIDT, DENB, NT
110 FORMAT(' SS=',G13.5,' DS=',G13.5,' TC=',
* G13.5,' F=',G13.5/' DPZERO=',G12.5,' PDROP=',
1 G13.5,' PLEN=',G13.5,' DT=',G13.5/' DTUBE=',
2 G13.5,' NTPIDT=',G13.5,' DENB=',G13.5,' NT=',
3 G13.5)
      WRITE(6,120) GAM1, GAM2, GAMINF, API, B, C, DCX, ADAX,
& BDAX, CDAX
120 FORMAT(' GAM1=',G13.5,' GAM2=',G13.5,' GAMINF=',
* G13.5,' B=',G13.5,' C=',G13.5/' RATIO=',
1 G13.5,' DCX=',G13.5,' ADAX=',G13.5,' BDAX=',
2 G13.5,' CDAX=',G13.5)
      WRITE(6,130) TEMP, VISC, AREA, V, CA, XA1, XC1, RE
130 FORMAT(' TEMP=',G13.5,' VISC=',G13.5,' AREA=',
* G13.5,' V=',G13.5/' CA=',G13.5,' XA1=',
1 G13.5,' XC1=',G13.5,' RE=',G13.5)
      WRITE(6,140) DAX, SCA, SHA, KA, DCX, SCC, SHC, KC
140 FORMAT(' DAX=',G13.5,' SCA=',G13.5,' SHA=',
* G13.5,' KA=',G13.5/' DCX=',G13.5,' SCC=',
1 G13.5,' SHC=',G13.5,' KC=',G13.5)
      WRITE(6,150) DPBAR, POS, TERM1, TERM2, TERM12,
& TOTALA, TOTALB, TOTALC
150 FORMAT(' DPBAR=',G13.5,' POS=',G13.5,' TERM1=',

```

```
1 G13.5,' TERM2=',G13.5/' TERM12=',G13.5,' TOTALA=',  
2 G13.5,' TOTALB=',G13.5,' TOTALC=',G13.5)  
160 RETURN  
END
```

SUBROUTINE UV

C THIS IS THE ULTRAVIOLET OZONATION INTERFACE



```

C  PARAMETER                QUANTITY
C      1                    INITIAL TSS CONC.
C      2                    INITIAL TDS CONC.
C      3                    INITIAL TOC CONC.
C      4                    INLET GAS PHASE OZONE TO AIR MASS RATIO
C      5                    VOLUMETRIC GAS FLOW RATE
C      6                    PRECONTACTOR FLAG (0= NO PRECONTACTOR)
C      7                    NUMBER OF STAGES
C      8                    AREA OF A CONTACTOR
C      9                    AREA OF THE PRECONTACTOR
C     10                    HEIGHT OF A STAGE
C     11                    FEED TEMPERATURE
C     12                    OPERATING PRESSURE
C  THE FEED STREAM MUST BE SPECIFIED FIRST
C  THE EFFLUENT STREAM MUST BE SPECIFIED SECOND
      DIMENSION ZOUT(10), CZOUT(10), CTOUT(10), SS(10),
      &          DS(10), DSS(10), DDS(10)
      COMMON /DELTAT/ DTUV
      COMMON /MATTOC/ TOCRCT
      COMMON /MATBAL/ BALNCE(4), AMTIN(4), AMTOUT(4)
      COMMON /CTIME/  TIME, FTIME, DT
      COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
      &          NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
      COMMON /LOOK/   ISW
      COMMON /STAGES/ NSTAGE, PRECON
      COMMON /MATDIS/ MATCAL
      COMMON /UVPARM/ CAREA, PAREA, H, RHO, PRESS, TEMP, NWR
      COMMON /STGSAV/ ZOUT, CZOUT, CTOUT, SS, DS
      IF(NCALL) 10, 100, 110
10  CONTINUE
C  INITILIZE
      PRECCN= PAR(NPAR+5)
      NSTAGE= PAR(NPAR+6)
      NNN= NSTAGE
      IF(PRECON .NE. 0.0) NNN= NSTAGE + 1
      IF(MATCAL .NE. 0) GO TO 30
      DO 20 I= 1, NNN
      CZOUT(I)= 0.
      ZOUT(I)= 0.
      SS(I)= PAR(NPAR)
      DS(I)= PAR(NPAR+1)
20  CTOUT(I)= PAR(NPAR+2)

```

```

30 CONTINUE
    FLAG= 0.
    IF (PRECON .NE. 0.0) FLAG=1.
    VCON= CAREA*H
    VPRE= PAREA*H
    BALNCE(1)= BALNCE(1) + NSTAGE*VCON+FLAG*VPRE
    DC 40 I= 1, NSTAGE
    BALNCE(2)= BALNCE(2) + VCON*SS(I)
    BALNCE(3)= BALNCE(3) + VCON*DS(I)
    BALNCE(4)= BALNCE(4) + VCON*CTOUT(I)
40 CONTINUE
    IF (PRECON .EQ. 0.0) GO TO 50
    BALNCE(2)= BALNCE(2) + VPRE*SS(NSTAGE+1)
    BALNCE(3)= BALNCE(3) + VPRE*DS(NSTAGE+1)
    BALNCE(4)= BALNCE(4) + VPRE*CTOUT(NSTAGE+1)
50 CONTINUE
    IF (ICONFG(3,IUNIT) .GT. 0) GO TO 70
    WRITE(6,60) IUNIT, ICONFG(3,IUNIT)
60 FORMAT(6X,'****ERROR, UNIT',I5,'. FIRST STREAM IN',
& ' CONFIGURATION IS',I5,'. MUST BE THE FEED.')
    NFATER= NFATER + 1
70 IF (ICONFG(4,IUNIT) .LT. 0) GO TO 90
    WRITE(6,80) IUNIT, ICONFG(4,IUNIT)
80 FORMAT(6X,'****ERROR, UNIT',I5,'. SECOND STREAM IN',
& ' CONFIGURATION IS',I5,'. MUST BE THE EFFLUENT.')
    NFATER= NFATER + 1
90 CONTINUE
100 CONTINUE
    RETURN
110 CONTINUE
    DTUV= DT
C ALL COMMON VARIABLES ARE ALREADY SET UP
C BEGIN SIMULATION
    IFEEED= ICONFG(3,IUNIT)
    IPROD= -ICONFG(4,IUNIT)
    CZIN= 0.0
    CTIN= STREAM(4,IFEEED)
    ZIN= PAR(NPAR+3)
    G= PAR(NPAR+4)/NSTAGE
    CAREA= PAR(NPAR+7)
    PAREA= PAR(NPAR+8)
    H= PAR(NPAR+9)
    TEMP= PAR(NPAR+10)
    PRESS= PAR(NPAR+11)
    F= STREAM(1,IFEEED)
    IF (ISW .GT. 1) WRITE(6,120) F, CTIN
120 FORMAT('0UV...BEFORE CALL TO UOZONE, FLOW=',G10.3,
& ' TOC CONC.=',G10.3)
    TOCRCT= 0.
    CALL UOZONE(ZIN,CZIN,CTIN,G,F,ZOUT,CZOUT,CTOUT)
C DETERMINE IF PRE-CONTACTOR IS PRESENT
    NNN= NSTAGE
    IF (PRECON .NE. 0.0) NNN= NSTAGE + 1
    STREAM(4,IPROD)= CTOUT(NSTAGE)

```

```

        STREAM(1,IPROD)= F
        AMTOUT(4)= AMTOUT(4) + TOCRCT
        IF(ISW .GT. 1) WRITE(6,130) F, CTOUT(NSTAGE)
130    FORMAT(' UV... AFTER CALL TO UOZONE, FLOW=',G10.3,
&          5X,'TOC CONC.=' ,G10.3)
        VPRE= PAREA*H
        VCON= CAREA*H
        DO 160 I= 1 ,NNN
        IF(I .EQ. 1) GO TO 140
        VCL= VCON
        SSO= SS(I-1)
        DSO= DS(I-1)
        GO TO 150
140    CONTINUE
        VOL= VPRE
        SSO= STREAM(2,IFFED)
        DSO= STREAM(3,IFFEED)
150    CCNTINUE
        DDS(I)= F*(DSO-DS(I))/VOL
        DSS(I)= F*(SSO-SS(I))/VOL
160    CONTINUE
        DO 170 I= 1, NNN
        SS(I)= SS(I) + DSS(I)*DT
        DS(I)= DS(I) + DDS(I)*DT
170    CCNTINUE
        STREAM(2,IPROD)= SS(NNN)
        STREAM(3,IPROD)= DS(NNN)
        IF(ISW .EQ. 1)
&    WRITE(6,180) (SS(I),I=1,NNN), (DS(I),I=1,NNN)
180    FORMAT(' UV',T10,8G10.3,/T10,8G10.3)
        RETURN
        END

```

```

      SUBROUTINE UOZONE(ZINO,CZINO,TOCINO,G,F,ZOUT,CZOUT,
&                     TOCOUT)
C
C   THIS SUBROUTINE PERFORMS THE STEADY STATE CALCULATIOES
C   FOR THE ULTRAVIOLET OZONATION UNIT
C
      REAL KLA, KHENRY, KRATE, KDCOMP
      COMMON /DELTAT/ DT
      COMMON /UVL/     UVLITE
      COMMON /OPIT/    KHENRY, ECOZ, ETOC, KRATE, KDCOMP,
&                     EOZD, UVEFCT, ALPHA, EN, QPRIME
      COMMON /OZOPER/  CAREA, PAREA, H, RHO, PRESS, TEMP, NWR
      DIMENSION ZOUT(10), CZOUT(10), TOCOUT(10), DCZBR(10),
&               DTOCBR(10)
      COMMON /STAGES/  NSTAGE,PRECON
      DATA IFIRST/0/
      EQUIVALENCE (NNN,IEND)
      NNN= NSTAGE
      IF(PRECON .NE. 0.0) NNN= NSTAGE + 1
      IF(IFIRST .NE. 0) GO TO 20
C   FOR FIRST ITERATION, SET UP INLET GAS CONCENTRATION
C   TO PRE-CONTACTOR
      IFIRST= 1
      TZOUT= 0.
      DO 10 I= 1, NSTAGE
10    TZOUT= TZOUT + ZOUT(I)
      ZBAR= TZOUT/NSTAGE
20    CONTINUE
      UVLITE= 0.
C   CALL FOR PRE-CONTACTOR
      CALL USTAGE(ZBAR, CZINO, TOCINO, NSTAGE*G, F,
&               ZOUT(NSTAGE+1), DCZBR(NSTAGE+1), DTOCBR(NSTAGE+1),
&               PAREA, CZOUT(NSTAGE+1), TOCOUT(NSTAGE+1))
      TZOUT= 0.
C   SET UP UV RADIATION EFFECT ON REACTION RATE CONSTANTS
      UVLITE= UVEFCT
      DO 60 I= 1, NSTAGE
C   SET UP INPUTS TO THE NEXT STAGE
      ZIN= ZINO
      N= I - 1
      IF(I .NE. 1) GO TO 30
      N= NSTAGE + 1
      IF(PRECON .EQ. 0.0) GO TO 40
30    CONTINUE
      CZIN= CZOUT(N)
      TOCIN= TOCOUT(N)
      GO TO 50
40    CONTINUE
      CZIN= CZINO
      TOCIN= TOCINO
50    CONTINUE
      CALL USTAGE(ZIN, CZIN, TOCIN, G, F, ZOUT(I), DCZBR(I),
&               DTOCBR(I), CAREA, CZOUT(I), TOCOUT(I))

```

```

      TZOUT= TZOUT + ZOUT(I)
    60 CONTINUE
  C DETERMINE NEXT GAS CONCENTRATION TO THE PRE-CONTACTOR
      ZBAR= TZOUT/NSTAGE
      DO 70 I= 1, IEND
  C TAKE ONE INTEGRATION STEP
      CZOUT(I) = CZOUT(I) + DCZBR(I)*DT
      70 TOCOUT(I)= TOCOUT(I) + DTOCBR(I)*DT
      RETURN
      END

```

```

      SUBROUTINE USTAGE(ZIN,CZIN,TOCIN,G,P,ZOUT,DCZ,DTOC,
&                      AREA,CZOUT,TOCOUT)
C
C   THIS SUBROUTINE PERFORMS THE STEADY STATE CALCULATIONS
C   FOR A SINGLE UV STAGE
C
      REAL KLA, KHENRY, KRATE, KDCOMP
      EQUIVALENCE (DCOMP,DECOMP), (NSTWRT,NWRITE)
      COMMON /DELTA1/ DT
      COMMON /MATTOC/ TOCRCT
      COMMON /UVFIT/ KHENRY, ECOZ, ETOC, KRATE, KDCOMP,
&                      EOZD, UVFFCT, ALPHA, EN, QPRIME
      COMMON /UVL/ UVLITE
      COMMON /UVPARM/ CAREA, PAREA, H, RHO, PRESS, TEMP,
&                      NWRITE
      COMMON /GASLAW/ RGAS
      DATA JWRITE /0/
C   USE IDEAL GAS LAW FOR DETERMINING GAS DENSITY
      RHOGAS= PRESS/TEMP/RGAS
      V= AREA*H
      CZ= CZOUT
      TOC= TOCOUT
C   DETERMINE LIQUID PHASE RESIDENCE TIME
      TAU= V/F
      YIN= ZIN/(1.0+ZIN)
      GMOLES= G*RHOGAS*(1.-YIN)
      AUX= RHO/KHENRY
C   DETERMINE M1(H)
      XYZ= EXP(-RHO*QPRIME*AREA*G**(1.-EN)/
&                      (KHENRY*V*RHOGAS))
C   SET UP REACTION RATE CONSTANTS
      RATE= KRATE*(1. + UVLITE)*(1. + UVLITE)
      DRATE= KDCOMP*(1. + UVLITE)*(1. + UVLITE)
C   BACKWARD DIFFERENCE INTEGRATION OF CZ EQUATION--REQUIRES
C   ALGEBRAIC MANIPULATIONS.
      D1= GMOLES*(1.-XYZ)/V
      D2= 1./TAU
      D3= 0.
      IF(TOC.GT.0.0) D3= ALPHA*RATE*TOC*ETOC
      IF(CZ.GT.0.0) GO TO 10
      AECOZ= 0.
      BECOZ= 0.
      AEOZD= 0.
      BEOZD= 0.
      GO TO 20
10   CONTINUE
C   APPROXIMATIONS RESULTING FROM FIRST ORDER TAYLOR
C   SERIES EXPANSION
      AECOZ= (1.-ECOZ)*CZ**ECOZ
      BECOZ= ECOZ*CZ**(ECOZ-1.)
      AEOZD= (1.-EOZD)*CZ**EOZD
      BEOZD= EOZD*CZ**(EOZD-1.)
20   CONTINUE

```

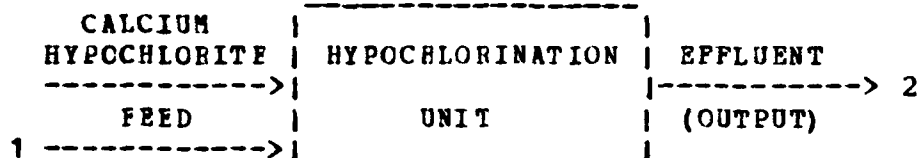
```

      D4= (D1*ZIN + D2*CZIN - D3*AECOZ - DRATE*AEOZD) *DT
      D5= -(D1/AUX + D2 + D3*BECZD + DRATE*BEOZD) *DT
      CZNEW=(CZ+D4)/(1.-D5)
      ZTOP=ZIN*XYZ+(1.0-XYZ)*CZNEW/AUX
C   CALCULATE TIME DERIVATIVE OF OZONE CONCENTRATION
      DCZ= (CZNEW-CZ)/DT
      TCREAC= 0.
      IF(TOC .GT. 0.0) TCREAC= RATE*(TOC**ETOC)*(CZ**ECOZ)
C   CALCULATE TIME DERIVATIVE OF TOC CONCENTRATION
      DTOC= (TOCIN-TOC)/TAU - TCREAC
      TOCRCT= TOCRCT + TCREAC*V*DT
      ZOUT= ZTOP
C   DETERMINE IF ANY OUTPUT IS NECESSARY AT THIS STEP
      IF(NWRITE .LE. 0) GO TO 50
      JWRITE= JWRITE + 1
      IF(JWRITE .LT. NWRITE) GO TO 50
30  CONTINUE
      JWRITE= 0
      WRITE(6,40) ZIN, CZIN, TOCIN, G, F, ZOUT, AREA, CZOUT,
&                TOCOUT, V, CZ, TOC, TAU, GHOLES, ZUX, XYZ,
&                RATE, DRATE, UVLITE, TCREAC, DCZ, DTOC, DT,
&                D1, D2, D3, D4, D5, AECOZ, BECOZ, AEOZD,
&                BECZD, CZNEW
40  FORMAT(' IN USTAGE', (/1X,8E12.4))
50  CONTINUE
      RETURN
      END

```

SUBROUTINE HC

THIS IS THE HYPOCHLORINATION INTERFACE



PARAMETER	QUANTITY
1	PH OF THE OUTPUT
2	INITIAL CHLORITE IN THE HC UNIT
3	INITIAL TSS CONC.
4	INITIAL TDS CONC.
5	INITIAL TOC CONC.
6	FEED RATE OF CALCIUM HYPOCHLORITE
7	VOLUME OF HYPOCHLORINATION UNIT
8	CONC. OF CAOCL2 FEED

THE FEED STREAM MUST BE SPECIFIED FIRST

THE EFFLUENT STREAM MUST BE SPECIFIED SECOND

```
REAL*8 HOCID, CCLD, PH1, F1, FIN, V, ALPHA, RD, KEQ,
& CAOCL2, PHI, HCCLI, OCLI, CLI, FAC
```

```
REAL MWHOCL, MWOCL
```

```
COMMON /HCSAV2/ MWHOCL, MWOCL, RHO
```

```
COMMON /HATEAL/ EALNCE(4), AMTIN(4), AMTOUT(4)
```

```
COMMON /CTIME/ TIME, FTIME, DT
```

```
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
```

```
& NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
```

```
COMMON /HCSAVE/ PHI, HOCCLI, OCLI
```

```
COMMON /HCPARM/ V, ALPHA, RD, KEQ, CAOCL2, JWRITE, MCNT
```

```
COMMON /HCSTOR/ SS, DS, TOC
```

```
COMMON /MATDIS/ MATCAL
```

```
FIN= PAR(NPAR+5)
```

```
V= PAR(NPAR+6)
```

```
CAOCL2= PAR(NPAR+7)
```

```
IF(NCALL) 10,70,80
```

```
10 PHI= PAR(NPAR)
```

```
IF(MATCAL.NE. 0) GO TO 20
```

```
SS= PAR(NPAR+2)
```

```
DS= PAR(NPAR+3)
```

```
TOC= PAR(NPAR+4)
```

```
FAC= PAR(NPAR+1)
```

```
OCLI= (FAC*RHO/1000000.)/((10.**(-PHI)/KEQ)*
& MWHOCL+MWOCL)
```

```
HOCLI= (OCLI*10.**(- PHI))/KEQ
```

```
20 CONTINUE
```

```
BALNCE(1)= BALNCE(1) + V
```

```
BALNCE(2)= BALNCE(2) + V*SS
```

```
BALNCE(3)= BALNCE(3) + V*DS
```

```
EALNCE(4)= BALNCE(4) + V*TOC
```

```

      IF (ICONFG(3,IUNIT) .GT. 0) GO TO 40
      WRITE(6,30) IUNIT, ICCNFG(3,IUNIT)
30   FORMAT(6X,'**** ERROR, UNIT',I5,' FIRST STREAM IN',
      & ' CONFIGURATION IS',I5,'. MUST BE THE FEED.')
      NFATER= NFATER + 1
40   IF (ICONFG(4,IUNIT) .LT. 0) GO TO 60
      WRITE(6,50) IUNIT, ICONFG(4,IUNIT)
50   FORMAT(6X,'**** ERROR, UNIT',I5,'. SECOND STREAM IN',
      & ' CONFIGURATION IS',I5,'. MUST BE THE EFFLUENT.')
      NFATER= NFATER + 1
60   CONTINUE
C   THE HCSS ROUTINE EXPECTS THE TOTAL CHLORINE DEMAND, RD,
C   TO BE EXPRESSED IN MOLAR UNITS. SIMILARLY FOR ALPHA
C
C   AS ENTERED IN THE INPUT DATA, RD IS IN TERMS OF PARTS PER
C   MILLION. ALPHA IS IN TERMS OF FRACTIONAL PARTS PER MILLI
C   THE FOLLOWING SECTION OF CODE RE-COMPUTES ALPHA AND RD
C   A MOLAR BASIS.
      HOCLD= (RD*ALPHA*RHO)/(1000000.*MWHOCL)
      OCLD= (RD*RHO/1000000. - HOCLD*MWHOCL)/MWOCL
      RD= HOCLD + OCLD
      ALPHA= 0.
      IF(RD .GT. 0.) ALPHA= HOCLD/RD
70   CONTINUE
      RETURN
80   CONTINUE
      IFEED= ICONFG(3,IUNIT)
      IPROD= -ICONFG(4,IUNIT)
      F1= STREAM(1,IFEED)
      PH1= 7.0
      CALL HCSS(F1,FIN,PH1)
      FAC= (HOCLI*MWHOCL+OCLI*MWOCL)/RHO*1000000.
      DSS= (F1*STREAM(2,IFEED) - (F1+FIN)*SS)/V
      DDS= (F1*STREAM(3,IFEED) - (F1+FIN)*DS)/V
      DTOC= (F1*STREAM(4,IFEED) - (F1+FIN)*TOC)/V
      SS= SS + DSS*DT
      DS= DS + DDS*DT
      TOC= TOC + DTOC*DT
      STREAM(1,IPROD)= F1 + FIN
      STREAM(2,IPROD)= SS
      STREAM(3,IPROD)= DS
      STREAM(4,IPROD)= TOC
      PAR(NPAR+1)= FAC
      PAR(NPAR)= PHI
      RETURN
      END

```

```

      SUBROUTINE HCSS(F1,FIN,PH1)
C
C THIS SUBROUTINE PERFORMS THE STEADY STATE HYPOCHLORINATION
C CALCULATIONS
C SEE THE REPORT BY SMITH AND STARKS ON THE HC UNIT FOR A
C DISCRIPTION OF VARIABLES
C
      REAL*8 V, ALPHA, RD, KEQ, CAOCL2, PHI, HOCLI, OCLI,
&          H1, H, OH, OH1, F1, F2, PH1, FIN, SUM, DSUM,
&          HOCLBG, HOCLSM, HOCL, OCL, HOCLP, ATERM,
&          BTERM, CTERM, X, RADCL, HNEW
      COMMON /HCPARM/V, ALPHA, RD, KEQ, CAOCL2, JWRITE, MCNT
      COMMON /HCSAVE/ PHI, HOCLI, OCLI
      COMMON /CTIME/ TIME, PTIME, DT
      EQUIVALENCE (JERROR,JWRITE)
      DATA JERRCT/0/
          H1= 10.0**(-PH1)
          H= 10.0**(-PHI)
          OH1= 1.E-14/H1
          OH= 1.E-14/H
          F2= F1 + FIN
C BEGIN SOLUTION
          SUM= HOCLI + OCLI
          DSUM= (2.*FIN*CAOCL2-F1*RD-F2*SUM)/V
          SUM= SUM + DSUM*DT
          HOCLBG= SUM
          HOCLSM= 0.0
          DO 60 I= 1, MCNT
              HOCL= (HOCLBG+HOCLSM)/2.
C ATERM, BTERM, AND CTERM ARE THE COEFFICIENTS USED IN THE
C SOLUTION OF X
              ATERM= H + (F1*H1 + 2.*FIN*CAOCL2 - F2*HOCL -
&                  F2*H - ALPHA*RD*F1)*DT/V
              BTERM= -F2*DT/V
              CTERM= OH+ (F1*OH1+2.*FIN*CAOCL2-F2*OH)*DT/V
              RADCL= ((CTERM + ATERM)*(CTERM + ATERM) -
&                  4.0*(ATERM*CTERM - 1.E-14))
              IF(RADCL .GE. 0.0) GO TO 10
C HOCL IS TOO SMALL
              GO TO 20
          10 CONTINUE
              X= (-(CTERM+ATERM)+DSQRT(RADCL))/2./BTERM
              HNEW= ATERM+BTERM*X
C CHECK FOR UNREALISTIC VALUES OF H+ AT TIME T+DT
              IF(HNEW.LE.0.0) GO TO 30
              OCL= KEQ*HOCL/HNEW
C DETERMINE HOCL', THE CHECK ON ASSUMED VALUE OF HOCL
              HOCLP= SUM-OCL
              IF(HOCL .GT. HOCLP) GO TO 30
          20 HOCLSM= HOCL
              GO TO 40
          30 HOCLBG= HOCL
          40 CONTINUE

```

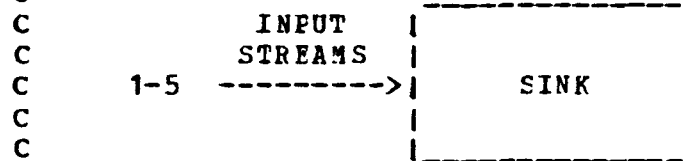
```

      IF(JERROR .LE. 0) GO TO 60
C   DETERMINE IF ANY WRITING IS TO BE PERFORMED
      JERRCT= JERRCT + 1
      IF(JERRCT .LT. JERROR) GO TO 60
      WRITE(6,50) I, TIME, HOCL, HOCLP, OCL, HNEW, X, RADCL,
&      ATERM, BTERM, CTERM, HOCLBG, HOCLSM, H, OH
50  FORMAT(' IN HYPOCL...I= ',I5,(/1X,5G15.5))
60  CCNTINUE
C   UPDATE VALUES TO TIME T+CT
      HOCLI= HOCL
      OCLI= OCL
      PHI= -DLOG10(HNEW)
      RETURN
      END

```

SUBROUTINE SK

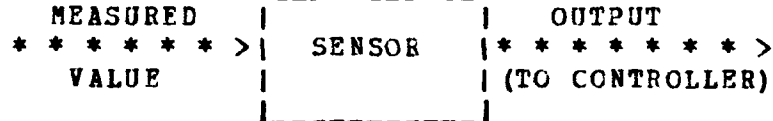
C
C SIMULATION OF STREAM SINK
C



C THERE ARE NO PARAMETERS
COMMON /LOOK/ ISW
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
8 NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
COMMON /MATBAL/ BALNCE(4), AMTIN(4), AMTOUT(4)
COMMON /CTIME/ TIME, FTIME, DT
C ASCERTAIN IF READ DATA, INITIALIZE OR SIMULATE
IF(NCALL) 10, 20, 30
C RETURN IF MATERIAL BALANCE CALCULATIONS
10 RETURN
C THERE ARE NO INITIALIZATION CALCULATIONS
20 RETURN
C SIMULATE CALCULATIONS CONSIST ONLY OF MATERIAL BALANCE
C EQUATIONS
30 DO 50 I=3,7
J=ICONFG(I,IUNIT)
IF(J.EQ. 0) GO TO 60
F= STREAM(1,J)
AMTOUT(1)= AMTOUT(1) + F*DT
DO 40 L=2,4
40 AMTOUT(L)= AMTOUT(L) + F*STREAM(L,J)*DT
50 CONTINUE
60 IF(ISW.EQ.1) WRITE(6,70) (AMTOUT(I),I=1,4)
70 FORMAT(' UNIT',15,' SK TOTALS FLOW=',G10.3,5X,
8 'SS=',G10.3,5X,'DS=',G10.3,5X,'TOC=',G10.3)
RETURN
END

SUBROUTINE SENSOR

THIS SUBROUTINE SENSES A UNIT PARAMETER OR A STREAM
ELEMENT AND PRODUCES THE TIME INTEGRAL OF THE ERROR



```

PARAMETER      QUANTITY
1              UNIT OR STREAM NUMBER
2              PARAMETER OR ELEMENT NUMBER
3              INITIAL OUTPUT VALUE
4              INTEGRATION TIME CONSTANT, TAU
IF TAU IS EQUAL TO ZERO, THEN THE VALUE IS RETURNED
IF TAU IS NON-ZERO, THEN
    (NEW VALUE - OLD OUTPUT VALUE)/TAU
    IS INTEGRATED AS THE NEW OUTPUT VALUE
COMMON /LOCK/ ISW
DATA IGET/'G'/
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
&          NCALL, IUNIT, NPATER, NS, NEQ, DESC(5)
COMMON /CTIME/ TIME, FTIME, DT
IF(NCALL) 10,20,30
C NO MATERIAL BALANCE CALCULATIONS
10 RETURN
20 CONTINUE
30 CONTINUE
    I= PAR(NPAR)
    J= PAR(NPAR+1)
    CALL GETPUT(IGET,I,J,R)
    C= PAR(NPAR+2)
    TAU= PAR(NPAR+3)
    IF(TAU.EQ.0.0) GOTO50
    DC= (R-C)/TAU
    IF(ISW .GT. 0)
        & WRITE(6,40) I, J, C, TAU, DC, C, R
40 FORMAT(' IN SENSOR ',(8G15.5,/1X))
    C= C + DC*DT
    GO TO 60
50    C= R
60    PAR(NPAR+2)=C
    RETURN
    END

```

SUBROUTINE MANIP

C THIS SUBROUTINE IS USED TO CHANGE THE VALUE OF A PARAMETER
C OF A UNIT, IE. INCREASE A PUMP OR AN OVERFLOW FLOW RATE
C
C
C

VALUE ***** > (FROM CONTROLLER)	MANIPULATOR	CONTROLLED ***** > PARAMETER
---------------------------------------	-------------	------------------------------------

PARAMETER	QUANTITY
1	NUMBER OF UNIT TO MANIPULATE (NEGATIVE)
2	PARAMETER NUMBER
3	OUTPUT VALUE
4	UPPER LIMIT
5	LOWER LIMIT

```

REAL LOVAL
COMMON /LOOK/ ISW
DATA IPUT/'P'/
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
& NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
IF(NCALL) 10,20,30
10 RETURN
20 CONTINUE
30 CONTINUE
    ITEM= PAR(NPAR)
    IF(ITEM .LT. 0) GO TO 50
    WRITE(6,40) ITEM, IUNIT
40 FORMAT(' IN ROUTINE MANIP...CAN ONLY MANIPULATE',
& ' EQUIPMENT....YOU HAVE SPECIFIED STREAM NUMBER',I10,
& ' . CURRENT UNIT NUMBER IS',I10)
    NFATER= NFATER + 1
50 CONTINUE
    J= PAR(NPAR+1)
    VALUE= PAR(NPAR+2)
    HIVAL= PAR(NPAR+3)
    LOVAL= PAR(NPAR+4)
    IF(VALUE .GT. HIVAL) VALUE= HIVAL
    IF(VALUE .LT. LOVAL) VALUE= LOVAL
    IF(ISW .GT. 0)
& WRITE(6,60) ITEM, J, VALUE, HIVAL, LOVAL
60 FORMAT(' IN MANIP ',(8G15.5,/1X))
    PAR(NPAR+2)= VALUE
    CALL GETPUT(IPUT,ITEM,J,VALUE)
    RETURN
END

```

```
C
C THIS SUBROUTINE IS USED TO SIMULATE A BINARY CONTROLLER
```

C	PARAMETER	CONTENT
C	1	SENSOR NUMBER
C	2	MANIPULATOR NUMBER
C	3	LOWER SET POINT VALUE
C	4	UPPER SET POINT VALUE
C	5	VALUE OF THE CONTROLLED VARIABLE
C	6	UPPER LIMIT OUTPUT VALUE
C	7	LOWER LIMIT OUTPUT VALUE
C	8	OPERATION MODE (NEGATIVE= AUTOMATIC, POSITIVE= OUTPUT VALUE FOR MANUAL MODE)

169

SUBROUTINE RATIO

THIS SUBROUTINE SIMULATE A RATIO CONTROLLER

```

PRCM SENSOR | RATIO | TO MANIPULATOR
* * * * * >| CONTROLLER |* * * * * >

```

PARAMETER	QUANTITY
1	SENSOR UNIT NUMBER
2	MANIPULATOR UNIT NUMBER
3	RATIO
4	OPERATION MODE (NEGATIVE= AUTOMATIC PCSSITIVE= OUTPUT VALUE FOR MANUAL)

```

REAL M
COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
6      NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
COMMON /LCCK/ ISW
DATA IPUT, IGET/'P', 'G'/
IF(NCALL) 10, 20, 30
10 RETURN
20 CONTINUE
30 CONTINUE
      ISEN= PAR(NPAR)
      IMAN= PAR(NPAR+1)
      CALL GETPUT(IGET,ISEN,3,C)
      R= PAR(NPAR+2)
      AMODE= PAR(NPAR+3)
      IF(AMODE.GT. 0) GO TO 60
      M= R*C
      IF(ISW.GT. 0)
6      WRITE(6,40) ISEN, IMAN, R, C, AMODE, M
40  FORMAT(' IN RATIO ',(8G15.5,/1X))
50  CALL GETPUT(IPUT,IMAN,3,M)
      RETURN
60      M= AMODE
      GOTO50
      END

```

```

C      SUBROUTINE PID
C      THIS SUBROUTINE SIMULATES A PID CONTROLLER
C      USING THE VELOCITY ALGORITHM
C
C      FROM SENSOR |-----| TO MANIPULATOR
C      * * * * * > | P-I-D | * * * * * >
C      |-----|
C
C      PARAMETER      QUANTITY
C      1              UNIT NUMBER OF THE SENSOR (NEGATIVE)
C      2              UNIT NUMBER OF THE MANIPULATOR (NEGATIVE)
C      3              SETPOINT
C      4              GAIN
C      5              RESET TIME
C      6              DERIVATIVE TIME
C      7              MODE OF OPERATION (NEGATIVE= AUTOMATIC,
C                      POSITIVE= OUTPUT VALUE FOR MANUAL)
C
C      REAL KC,M
C      COMMON STREAM(4,100), ICONFG(8,100), PAR(500), NPAR,
C      &      NCALL, IUNIT, NFATER, NS, NEQ, DESC(5)
C      COMMON /CTIME/  TIME, FTIME, DT
C      COMMON /PIDSAV/ EIM1, EIM2
C      COMMON /LOOK/   ISW
C      DATA IGET, IPUT/'G', 'P'/
C      THIS ROUTINE USES THE VELOCITY ALGORITHM
C      ISEN= PAR(NPAR)
C      IMAN= PAR(NPAR+1)
C      IF(NCALL) 10,20,40
C      10 RETURN
C      CHECK TO SEE IF ACCESSING SENSORS AND MANIPULATORS
C      20  EIM1= 0.
C      EIM2= 0.
C      IF(IMAN .LT. 0 .AND. ISEN .LT. 0) GO TO 40
C      WRITE(6,30) IMAN, ISEN
C      30 FORMAT(' IN PID...IMPROPERLY SPECIFIED SENSOR OR ',
C      & ' MANIPULATOR'/' MANIPULATOR IS',I5,' SENSOR IS',I5)
C      NFATER= NFATER + 1
C      RETURN
C      40  R= PAR(NPAR+2)
C      CALL GETPUT(IGET,ISEN,3,C)
C      EI= R - C
C      KC= PAR(NPAR+3)
C      TI= PAR(NPAR+4)
C      TD= PAR(NPAR+5)
C      AMODE= PAR(NPAR+6)
C      IF(AMODE .GT. 0.) GO TO 60
C      CONTROLLER IS ON AUTO
C      DE= (EI - EIM1)/DT
C      DE2= (EI - 2.*EIM1 + EIM2)/(DT*DT)

```

```

      DM= KC*(DE + EI/TI + TD*DE2)
      CALL GETPUT(IGET,IMAN,3,M)
      IF(ISW .GT. 0)
        & WRITE(6,50) ISEN, IMAN, R, C, EI, EIM1, EIM2, KC,
        &          TI, TD, AMODE, DE, DE2, DM, M
50  FORMAT(' IN PID ',(8G15.5,/1X))
      M= M + DM*DT
      CALL GETPUT(IPUT,IMAN,3,M)
      EIM2= EIM1
      EIM1= EI
      RETURN
C  CONTROLLER IS ON MANUAL
60  M= AMODE
      CALL GETPUT(IPUT,IMAN,3,M)
      RETURN
      END

```

```

      SUBROUTINE GETPUT(ICODE,ITEMO,J,VALUE)
C
C   THIS SUBROUTINE PROVIDES THE INTERFACE BETWEEN THE
C   SENSORS AND CONTROLLERS AND THE PROCESS UNITS AND STREAMS
C   IF ICODE= 'P' THEN "VALUE" IS ASSIGNED TO THE PARAMETER
C   OR ELEMENT SPECIFIED BY "ITEMO" AND "J"
C   IF ICODE= 'G' THEN "VALUE" IS SET EQUAL TO THE VALUE
C   OF THE SPECIFIED PARAMETER OR ELEMENT
C   ITEMO= UNIT NUMBER IF NEGATIVE
C   = STREAM NUMBER IF POSITIVE
C   J= UNIT PARAMETER OR STREAM ELEMENT NUMBER
C
      COMMON STREAM(100,4), ICONFG(100,8), PAR(500), NPAR,
&          NCALL, IUNIT, NPATER, NS, NEQ, DESC(5)
      COMMON /LOOK/ ISW
      DATA IPUT/'P'/
      DATA IGET/'G'/
      ITEM= ITEMO
C   CHECK ON VALIDITY OF ICODE
      IF(ICODE .EQ. IGET .OR. ICODE .EQ. IPUT) GO TO 20
      WRITE(6,10) ICODE
10  FORMAT(' IN ROUTINE GETPUT...INVALID CODE.',/' ICODE',
& ' IN I FORMAT IS ',I10,' ICODE IN A FORMAT IS ',A1)
      NPATER= NPATER + 1
      RETURN
20  IF(ITEM .GT. 0) GO TO 60
C   HAVE A PIECE OF EQUIPMENT
      ITEM= -ITEM
C   FIND POSITION "K" OF UNIT IN CONFIGURATION
C   (POSITION IS NOT NECESSARILY EQUAL TO THE UNIT NUMBER)
      DO 30 K=1,NEQ
      IF(ICONFG(K,1) .EQ. ITEM) GO TO 50
30  CONTINUE
      WRITE(6,40) ITEM
40  FORMAT(' IN ROUTINE GETPUT...UNABLE TO FIND UNIT'
& ', NUMBER',I5,/' SETTING ERROR CODE')
      NPATER= NPATER + 1
      RETURN
C   HAVE FOUND CORRECT PIECE OF EQUIPMENT
50      N= ICONFG(K,8)
C   N= LOCATION OF FIRST PARAMETER FOR THIS UNIT IN "PAR"
      IF(ICODE .EQ. IPUT) PAR(N+J-1)= VALUE
      IF(ICODE .EQ. IGET) VALUE= PAR(N+J-1)
      RETURN
C   WE HAVE A STREAM
60  IF(ICODE .EQ. IGET) VALUE= STREAM(ITEM,J)
      IF(ICODE .EQ. IPUT) STREAM(ITEM,J)= VALUE
      GO TO 50
      END

```

REFERENCES

- (1) Starks, D.M. and Smith, C.L., "A Mathematical Model on an Ultra-filtration Unit for Water Purification Use". Proceedings of the 1978 Summer Computer Simulation Conference, Newport Beach, July 24-26, 1978.
- (2) Starks, D.M. and Smith, C.L., "A Mathematical Model of a Reverse Osmosis System for Water Purification Use", Proceedings of the Ninth Annual Pittsburgh Conference on Modeling and Simulaton, Pittsburgh, April 27-28, 1978.
- (3) Starks, D.M., "A Dynamic Model for the Simulation of Water Re-Use Units", Ph.D. Dissertation, Louisiana State University, Baton Rouge, Louisiana, 1978.

DISTRIBUTION LIST

5 copies	Commander US Army Medical Bioengineering Research and Development Laboratory ATTN: SGRD-UBG Fort Detrick, Frederick, MD 21701
2 copies	Commander US Army Medical Research and Development Command ATTN: SGRD-RMS Fort Detrick, Frederick, MD 21701
1 copy	Defense Technical Information Center (DTIC) ATTN: DTIC-DDA Cameron Station Alexandria, VA 22314

END

FILMED

8

74

DTIC