A DIGITAL COMPUTER CODE FOR SIMULATING
THE DYNAMICS OF DEMONSTRATION SIZE
DUAL-PURPOSE DESALTING PLANTS USING
A PRESSURIZED WATER REACTOR
AS A HEAT SOURCE

J. G. Delene

23, 120

OAK RIDGE NATIONAL LABORATORY

OPERATED BY UNION CARBIDE CORPORATION • FOR THE U.S. ATOMIC ENERGY COMMISSION

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Contract No. W-7405-eng-26

Reactor Division

A DIGITAL COMPUTER CODE FOR SIMULATING THE DYNAMICS OF
DEMONSTRATION SIZE DUAL-PURPOSE DESALTING PLANTS USING
A PRESSURIZED WATER REACTOR AS A HEAT SOURCE

J. G. Delene

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

SEPTEMBER 1973

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
U.S. ATOMIC ENERGY COMMISSION

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

**THIS PAGE
WAS INTENTIONALLY
LEFT BLANK**

CONTENTS

	<u>Page</u>
TABLE OF NOMENCLATURE	v
LIST OF FIGURES	ix
ABSTRACT	1
INTRODUCTION	1
PLANT DESCRIPTION	2
SIMULATOR MODELING	6
MSF Plant	6
Brine Heater	7
Reactor	9
Steam Generator	12
Turbines	15
Moisture Separator and Reheaters	18
Crossover Holdup	19
Main Condenser	20
Feed Pump Turbine	20
Feedwater Heater	20
Control System	27
SAMPLE RESULTS	30
REFERENCES	33
Appendix A. CODE STRUCTURE	35
Appendix B. INPUT INSTRUCTIONS	39
Appendix C. FORTRAN LISTING	53

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

NOMENCLATURE

A	Area, ft^2
C	Salt concentration; delay neutron precursor concentration; a constant
C_p	Specific heat, $\text{Btu/lb } ^\circ\text{F}$
F_c	Flow constant, $\text{lb/sec } ^\circ\text{F}$
H	Total heat content, Btu
H_{fg}	Heat of vaporization, Btu/lb
h	Enthalpy, Btu/lb ; heat transfer coefficient, $\text{Btu/sec-ft}^2 \cdot ^\circ\text{F}$
K	Control system constant
k	Specific heat ratio, C_p/C_v
M	Mass, lb
P	Pressure, psia
P_{WR}	Power, MW(t)
T	Temperature, $^\circ\text{F}$
U	Overall heat transfer coefficient, $\text{Btu/sec-ft}^2 \cdot ^\circ\text{F}$
V	Volume, ft^3
v	Specific volume, ft^3/lb
W	Flow rate, lb/sec
X	Steam quality
α	Temperature coefficient of reactivity, $/^\circ\text{F}$
β	Delay neutron fraction
ℓ	Prompt neutron generation time, sec
λ	Delay neutron fraction
ρ	Excess reactivity
η	Fraction of reactor power deposited in coolant

Subscripts

A	Average
BH	Brine heater
BP	Back-pressure turbine
C	Condensate, core, coolant
CL	Cold leg of primary loop
D	Demand
DC	Drain cooler section of feedwater heater
d	Downcomer
e	external, error
EV	Evaporator section of feedwater heater
F	Fuel
FC	Feedwater in drain cooler section of feedwater heater
FE	Feedwater in evaporator section of feedwater heater
FP	Feed pump turbine
FW	Feedwater
F,C	From fuel to coolant
f	Flashing
HL	Hot leg of primary loop
HP	High-pressure turbine
i	Inside, inlet
i, ϕ	From inlet to outlet
L	Lower plenum, liquid stream in vapor section of feedwater heater
LP	Low-pressure turbine
NB	Nonboiling region of steam generator
ϕ	Outside or outlet

s	Steam; set point
S	Salt in brine
sat	Saturated conditions
sg	Steam generator
SH	Superheat
T	Tube brine, liquid inside of tube bundle
Tube	Tube metal
U	Upper plenum
V	Vapor
X $\emptyset$	Crossover
o	Initial values

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

LIST OF FIGURES

<u>Figure No.</u>	<u>Title</u>	<u>Page</u>
1	Overall Plant Arrangement	3
2	Turbine Plant Flow Diagram	4
3	MSF Plant Flow Diagram	5
4	Reactor and Steam Generator	10
5	Turbine Model	17
6	Feedwater Heater	21
7	Flow Diagram of Simulated Feedwater Heater	23
8	Reactor Control System	28
9	Reactor Power and Average Coolant Temperature as a Function of Time for a Negative 10% Electric Demand Pulse	31
10	Steam Generator Pressure and Feedwater Temperature as a Function of Time for a Negative 10% Electric Demand Pulse	32
A-1	High Pressure Turbine Data Layout	46
A-2	Low Pressure Turbine Data Layout	47
A-3	Back Pressure and Feed Pump Turbines Data Layout	48

A DIGITAL COMPUTER CODE FOR SIMULATING THE DYNAMICS OF
DEMONSTRATION SIZE DUAL-PURPOSE DESALTING PLANTS
USING A PRESSURIZED WATER REACTOR
AS A HEAT SOURCE

J. G. Delene

ABSTRACT

A digital simulator was developed for dual-purpose desalination plants of demonstration size (~50 Mgd) in which multistage flash (MSF) evaporators are coupled to a nuclear power plant through a back-pressure turbine. A pressurized water reactor (PWR) is used as the heat source. This simulator can calculate the time response of the coupled system, predicting pressures, temperatures, MSF plant liquid levels, reactor power, etc. The reactor, steam generator, and turbine plant system are described. The methods and equations used to describe the plant dynamics are discussed. Some sample results are also presented. The simulator can calculate a 1 hr transient in a 50 stage MSF plant coupled to the reactor-turbine plant in about 15 minutes on the IBM 360/91 at Oak Ridge National Laboratory.

Keywords: *plant dynamics + *programs (computers) + *dual-purpose plants + dynamic response + digital computer systems + multistage flash + pressurized water reactor + steam generators + turbine generators + AEC sponsored + stability analysis + sensitivity

INTRODUCTION

If large seawater distillation plants are to use steam from nuclear reactor turbine generator plants, an understanding of the overall compatibility of such dual-purpose plants must be achieved. Information is needed on how normal operational changes in either the water or electrical plant affect the other sections of the plant, and whether accidents in one part of the coupled plant will cause unacceptable conditions to occur elsewhere in the system. Using this information, more compatible coupled systems may be designed.

As an aid to understanding and predicting the response of demonstration size (~50 Mgd) dual-purpose plants, a digital simulator was developed. This simulator is an extension of the multistage flash (MSF) evaporator digital simulator reported previously.¹ The present simulator can

calculate the response of the combined plant to various perturbations, predicting temperatures, reactor power, pressures, water levels, etc., throughout the plant as functions of time. The coupled plant simulator can calculate a 1 hr transient in approximately 15 min. on the IBM 360/91 computer at Oak Ridge National Laboratory.

An overall model similar to, but not exactly the same as, the proposed Diablo Canyon Plant² was constructed. This model uses a PWR with a drum type steam generator. A conventional turbine plant design is used. Instead of extracting steam for the water plant from a low pressure turbine extraction point as is proposed in the Diablo Canyon design, a back-pressure turbine is utilized. Studies at ORNL³ have indicated advantages to the use of back-pressure turbines for steam supply in large desalting plants; thus a back-pressure turbine is an important element in this demonstration project.

The demonstration plant simulator as presented herein is a specialized code in that it was written to study the dynamics of a particular plant configuration. Responses of plants of other designs may be calculated with the simulator if these other plants are hooked up in the same manner as the reference case. Some quantities such as turbine stage efficiencies are built into the simulator, however, so that divergent design cases may suffer in accuracy.

In this report we discuss the assumptions and models used in describing the various components of the reactor-turbine plant in the simulator. The equations solved by the simulator are given and sample results are presented. The MSF plant simulation is the same as previously reported and is not described here. The dual-purpose plant code structure, input instructions, and a complete Fortran code listing are presented in the appendices.

PLANT DESCRIPTION

A block diagram of the simulated plant is shown in Fig. 1. A more detailed flow diagram of the turbine plant arrangement is shown in Fig. 2, and a flow diagram for a typical MSF plant is presented in Fig. 3.

Heat is produced in the reactor, and is carried by the coolant to the steam generator, where it produces steam. This steam passes through

ORNL-DWG 72-12006

Fig. 1. Overall Plant Arrangement

Fig. 3. MSF Plant Flow Diagram

high-pressure, low-pressure and back-pressure turbines, driving electric power generators in the process. Steam and condensate are removed at various extraction points in the turbines and pass through feedwater heaters and a main condenser. This feedwater reenters the steam generator to again become steam.

Some steam is diverted from the crossover between high- and low-pressure turbines, and sent through a back-pressure turbine. The steam leaving this back-pressure turbine is then used to heat brine going to the MSF plant.¹ The brine heater condensate is recycle to the feed heater chain.

SIMULATOR MODELING

Mathematical models have been developed for the various components of the dual-purpose plant. In the simulator we make the approximation that each component is effectively decoupled from each of the other components during a calculational time step. This component or stage isolation procedure permits the solution of many smaller problems instead of a few big ones, leading to a reduction in computer running time. This technique has been described previously in the report on the MSF plant simulator.¹ As in the previous report it is assumed that the mass and heat flow rates between the isolated components remain constant during a calculational time step.

The isolated components in the simulator include the MSF plant (up to 50 isolated stages), the brine heater, the reactor, a steam generator (represents all 4 units), a high pressure turbine + moisture separator + 2 reheater components, a low pressure turbine, a back-pressure turbine, 7 feed heater components, and a steam holdup component at the turbine crossover. Auxiliary calculations are made for the feed pump turbine and the main condenser but they can not truly be called isolated. The modeling of the various plant components are described in the following sections.

MSF Plant

The modeling used for the multistage flash evaporator (MSF) plant is the same as described previously.¹ Some changes were made in the code organization in order to facilitate coupling with the combined

plant code. The MSF plant simulation equations and techniques will not be described here.

Brine Heater

A model for the brine heater was included in the MSF simulation. In that simulation the brine heater was described by 4 differential equations. The state variables included the temperature of the brine leaving the brine heater ($T_{BH,\phi}$), the average brine temperature ($T_{BH,A}$), the average tube temperature ($T_{BH,T}$), and the mass of salt in the brine ($M_{BH,S}$). In that simulation, shell side steam was treated as an independent input, not a state variable.

The current simulation is a 7-variable model. The state variables include the 4 mentioned above as well as the steam, or vapor mass, on the shell side ($M_{BH,V}$), the mass of condensate holdup ($M_{BH,C}$), and the temperature of this condensate ($T_{BH,C}$).

The steam is assumed to be at saturated conditions, so that if its mass and volume are known, the pressure can be calculated as was done for a MSF stage simulation,

$$P_v = \left(\frac{\partial P_{sat}}{\partial \rho_{sat}} \right) / V_v \cdot M_v + C \quad (1)$$

The saturated temperature is calculated using a subroutine relating saturated temperature and pressure.

The steam pressure in the brine heater is the back pressure for the back-pressure turbine. The steam entering the brine heater from the back-pressure turbine usually has a quality (X_{BH}) less than 1 (i.e., it is wet steam). This steam is assumed to divide into a saturated steam portion and a saturated liquid portion when it enters the brine heater.

The differential equations used to describe the heat and mass transport mechanism in the brine heater are:

For the brine and tube metal temperatures,

$$\frac{dT_{BH,\phi}}{dt} = \frac{2W_T}{M_T} (T_{BH,A} - T_{BH,\phi}) + \frac{(Ah)_i}{M_T C_{P,T}} (T_{BH,T} - T_{BH,A}) \quad (2)$$

$$\frac{dT_{BH,A}}{dt} = \frac{2W_T}{M_T} (T_{BH,i} - T_{BH,A}) + \frac{(Ah)_i}{M_{T,P,T}} (T_{BH,T} - T_{BH,A}) \quad (3)$$

$$\frac{dT_{BH,T}}{dt} = \frac{(Ah)_\phi}{(MC)_p \text{ tube}} (T_{BH,V} - T_{BH,T}) - \frac{(Ah)_i}{(MC)_p \text{ tube}} (T_{BH,T} - T_{BH,A}) \quad (4)$$

For the shell side vapor mass balance,

$$\frac{dM_{BH,V}}{dt} = W_{s,i} X_{BH} - W_{BH,C} \quad (5)$$

Equation 6 is the condensate mass balance.

$$\frac{dM_{BH,C}}{dt} = W_{s,i} (1 - X_{BH}) + W_{BH,C} - W_{BH,\phi} \quad (6)$$

The condensate temperature is described by

$$\frac{dT_{BH,C}}{dt} = (W_{BH,C} + W_{s,i} (1 - X_{BH})) \cdot (T_{BH,V} - T_{BH,C}) / M_{BH,C} \quad (7)$$

The mass of the salt in the brine and its resulting concentration are found through Equations 8 and 9.

$$\frac{dM_{BH,S}}{dt} = W_T (C_{BH,i} - C_{BH}) \quad (8)$$

$$C_{BH} = M_{BH,S} / M_T \quad (9)$$

The vapor condensing rate ($W_{BH,C}$) is computed with a simple heat balance

$$W_{BH,C} = \frac{h A_\phi}{H_{fg}} (T_{BH,V} - T_{BH,T}) \quad (10)$$

The flow of condensate out ($W_{BH,\phi}$) is assumed to be controlled equal to the flow rate of steam entering the brine heater ($W_{s,i}$). The total inventory of vapor plus condensate therefore remains constant during a simulator run.

Six of the seven state variable equations (Eq. 2-7) are solved using the matrix exponential method as described in the MSF report. The seventh differential equation, or salt mass equation (Eq. 8) is solved independently using Euler's method.

Reactor

The reactor is modeled after a PWR similar to those to be used for the Diablo Canyon or Sequoia power plants. A simplified sketch of the reactor and steam generator layout is shown in Fig. 4.

Pressurized water from the steam generator enters the reactor vessel at the level of the upper plenum and flows along the vessel wall on the outside of the core to the lower plenum. The water enters the core at the bottom and flows upward through the core and into the upper plenum and eventually back to the steam generator.

A seven variable lumped-parameter model is used to describe the reactor dynamics. These nodes include the reactor power (P_{WR}), one group of delayed-neutron precursors (C), the average fuel temperature (T_F), the coolant temperature leaving the core ($T_{C,\phi}$), the average coolant temperature in the core region ($T_{C,A}$), the temperature of the coolant leaving the lower plenum, (T_L) and the temperature of the coolant leaving the upper plenum (T_U). The coolant inventory, or mass, is assumed to be a constant.

The coolant traveling along the side of the core is included with the lower plenum coolant. As the inlet coolant passes along the side of the upper plenum and core it is heated somewhat. In the simulation, this heat is assumed to come from the upper plenum liquid. The driving temperature for this heat transfer is assumed to be the difference between the core outlet temperature and the temperature of coolant entering from the steam generator.

The differential equations used in the simulator are for:
reactor power,

$$\frac{dP_{WR}}{dt} = \frac{\rho - \beta}{\ell} P_{WR} + \lambda C \quad (11)$$

ORNL-DWG 72-12009

Fig. 4. Reactor and Steam Generator

delay neutron precursors,

$$\frac{dC}{dt} = \frac{\beta}{\ell} P_{WR} - \lambda C \quad (12)$$

average fuel temperature,

$$\frac{dT_F}{dt} = \frac{1 - \eta}{(MC_P)_F} P_{WR} - \frac{(UA)_{F,C}}{(MC_P)_F} (T_F - T_{C,A}) \quad (13)$$

average core coolant temperature,

$$\frac{dT_{C,A}}{dt} = \frac{\eta P_{WR}}{(MC_P)_C} + \frac{(UA)_{F,C}}{(MC_P)_C} (T_F - T_{C,A}) - \frac{2W_P}{M_C} (T_{C,A} - T_L) \quad (14)$$

core outlet coolant temperature,

$$\frac{dT_{C,\phi}}{dt} = \frac{\eta P_{WR}}{(MC_P)_C} + \frac{(UA)_{F,C}}{(MC_P)_C} (T_F - T_{C,A}) - \frac{2W_P}{M_C} (T_{C,\phi} - T_{C,A}) \quad (15)$$

lower plenum coolant temperature,

$$\frac{dT_L}{dt} = \frac{W_P}{M_L} (T_{R,i} - T_L) + \frac{(UA)_{i,\phi}}{(MC_P)_L} (T_{C,\phi} - T_{R,i}) \quad (16)$$

and upper plenum coolant temperature,

$$\frac{dT_u}{dt} = \frac{W_P}{M_u} (T_{C,\phi} - T_u) - \frac{(UA)_{i,\phi}}{(MC_P)_u} (T_{C,\phi} - T_{R,i}) \quad (17)$$

These equations are solved using the matrix exponential method.

The heat transfer values $(UA)_{F,C}$ and $(UA)_{i,\phi}$ are those necessary to obtain a steady state heat balance. They are calculated from initial steady state temperature and flow conditions and are assumed to remain constant through a simulator run.

Both fuel and coolant temperature feedbacks are included in the simulation model. The excess reactivity (ρ) is recalculated each computer time step

$$\rho = \rho_e + \alpha_F (T_F - T_{F,O}) + \alpha_C (T_{C,A} - T_{C,A,O}) \quad (18)$$

The quantity ρ_e includes any external reactivity source, such as shim rod movement. The automatic reactor control system, which will be discussed later, utilizes ρ_e in order to simulate shim rod movement. ρ_e may be either negative (rod insertion), or positive (rod withdrawal). The reactivity coefficients α_F and α_C are negative numbers.

Steam Generator

The steam generator chosen for simulation is a U-tube drum type recirculating boiler as shown in Fig. 4. Feedwater from the feedwater heaters enters the steam generator in the downcomer where it mixes with the recirculating water. This mixed water moves down through the downcomer and enters the heating region of the steam generator at the bottom. Starting here, the feedwater is heated by the pressurized water in the U-tube bundle, and rises by natural convection. At a certain level the liquid begins to boil. This is the boundary between the nonboiling and the boiling region in our analysis. At the top of the steam generator the steam and water are separated. The steam goes to the turbine plant and the water recirculates back into the downcomer.

In the primary loop, pressurized water from the reactor enters the U-tube bundle at the bottom of the steam generator. These U-tubes extend through the nonboiling region and into the boiling region. They then curve back and again pass through the nonboiling region of the steam generator. The cooled primary liquid returns to the reactor.

Ten state variable equations are used to describe the steam generator dynamics. Six of these are for the temperature of the primary liquid. This liquid is divided into three sectors. The first sector extends from the inlet to the nonboiling, boiling, interface. The second sector includes the primary water in the boiling region and the third sector extends from the boiling, nonboiling interface to the point where primary water leaves the steam generator. Each sector contains an average and an outlet temperature node.

The six primary temperature equations are:
The outlet ($T_{\phi,1}$) and average ($T_{A,1}$) in Sector 1

$$\frac{dT_{\phi,1}}{dt} = \frac{2W_P}{M_1} (T_{A,1} - T_{\phi,1}) - \frac{(UA)_1}{(M_1 C_P)_1} (T_{A,1} - T_{NB,A}) \quad (19)$$

$$\frac{dT_{A,1}}{dt} = \frac{2W_P}{M_1} (T_i - T_{A,1}) - \frac{(UA)_1}{(M_1 C_P)_1} (T_{A,1} - T_{NB,A}) \quad (20)$$

The outlet ($T_{\phi,2}$) and average ($T_{A,2}$) temperatures in Sector 2

$$\frac{dT_{\phi,2}}{dt} = \frac{2W_P}{M_2} (T_{A,2} - T_{\phi,2}) - \frac{(UA)_2}{(MC_P)_2} (T_{A,2} - T_s) \quad (21)$$

$$\frac{dT_{A,2}}{dt} = \frac{2W_P}{M_2} (T_{\phi,1} - T_{A,2}) - \frac{(UA)_2}{(MC_P)_2} (T_{A,2} - T_s) \quad (22)$$

The outlet ($T_{\phi,3}$) and average ($T_{A,3}$) temperatures in Sector 3

$$\frac{dT_{\phi,3}}{dt} = \frac{2W_P}{M_3} (T_{A,3} - T_{\phi,3}) - \frac{(UA)_3}{(MC_P)_3} (T_{A,3} - T_{NB,A}) \quad (23)$$

$$\frac{dT_{A,3}}{dt} = \frac{2W_P}{M_3} (T_{\phi,2} - T_{A,3}) - \frac{(UA)_3}{(MC_P)_3} (T_{A,3} - T_{NB,A}) \quad (24)$$

On the secondary side there are 4 equations. The temperature in the nonboiling region has two equations, one for the average region temperature ($T_{NB,A}$) and one for the region outlet temperature ($T_{NB,\phi}$)

$$\frac{dT_{NB,\phi}}{dt} = \frac{2W_d}{M_{NB}} (T_{NBA} - T_{NB,\phi}) + \frac{(UA)_1}{(MC_P)_{NB}} (T_{A,1} - T_{NB,A}) + \frac{(UA)_3}{(MC_P)_{NB}} (T_{A,3} - T_{NB,A}) \quad (25)$$

$$\frac{dT_{NB,A}}{dt} = \frac{2W_d}{M_{NB}} (T_d - T_{NB,A}) + \frac{(UA)_1}{(MC_P)_{NB}} (T_{A,1} - T_{NB,A}) + \frac{(UA)_3}{(MC_P)_{NB}} (T_{A,3} - T_{NB,A}) \quad (26)$$

The third secondary side equation is for the temperature of the liquid leaving the downcomer (T_d)

$$\frac{dT_d}{dt} = \frac{W_d}{M_d} (T_s - T_d) - \frac{W_{sg,i}}{M_d} (T_s - T_{FW,I}) \quad (27)$$

The tenth state variable is the steam mass (M_s)

$$\frac{dM_s}{dt} = W_s - W_{sg} \quad (28)$$

Several simplifying assumptions are made in the steam generator model which may limit the simulator's ability to predict the results of wide ranging transients. First, the boiling region temperature is assumed equal to the steam temperature. We assume that the mass of the primary liquid in each sector is constant (no density variation with temperature). A constant liquid level is assumed for both the nonboiling and downcomer regions. With the constant density assumption this leads to constant downcomer mass (M_d) and nonboiling region mass (M_{NB}). For this to be so the recirculation flow (W_R) must equal the difference between the downcomer flow rate (W_d) and the feedwater inlet flow rate $W_{FW,i}$

$$W_R = W_d - W_{FW,i} \quad (29)$$

The feedwater flow is assumed to be equal to the rate of steam production (W_s). This assumption maintains a constant liquid inventory in the steam generator. The downcomer flow rate is assumed to be driven by the temperature difference between the steam and downcomer temperature

$$W_d = F_c (T_s - T_d) \quad (30)$$

F_c is a constant calculated from steady state flow and temperature. It is held constant during a simulator run.

In order to satisfy the component isolation assumption, both the steam flow to the turbines (W_{sg}) and the feedwater flow rate ($W_{FW,i}$) are

held constant during a calculational time step. The steam production rate (W_s) is calculated by a heat balance,

$$W_s = [(UA)_2(T_{A,2} - T_s) + W_d C_{p,s}(T_{NB,\phi} - T_s)]/H_{f,g} \quad (31)$$

The term

$$W_d C_{p,s}(T_{NB,\phi} - T_s)$$

must be included in the heat balance in order to account for the fixed position of boiling interface. This is because the liquid leaving the "nonboiling" region may be slightly superheated or subcooled in the calculation.

The simulation assumes constant heat transfer coefficients in the nonboiling region, so that $(UA)_1 = (UA)_3$. Using steady-state heat balances and the design value of steam quality leaving the boiling region before separation (25% in simulation), the steady-state values for the (UA) 's for each region and the initial values for the temperature state variables are found iteratively in the simulator. The heat transfer numbers (UA) are assumed constant and are not recalculated during a simulation run.

The steam pressure is obtained from the steam mass in the same way as done in the MSF portion of the simulation

$$P_s = \left(\frac{\partial P_{sat}}{\partial \rho_{sat}} / V_s \right) \cdot M_s + C. \quad (32)$$

Turbine Simulation

The procedures used in simulating the high-pressure, low-pressure, and back-pressure turbines are basically the same. If the rotor RPM is unchanged, the turbine steam conditions respond very swiftly to perturbations in inlet variables. It is assumed that this reaction time is short compared with the computational time step (usually 0.5 sec) so that the turbine is always essentially in steady state. At each time step during a transient calculation there is a redistribution of steam flow within the turbine, and out of the exhaust and extraction lines. These flows follow a redistribution of pressure within the turbine. The steam flow rate is assumed to follow the ideal gas law flow equation

$$W = A \left[\frac{k}{k-1} \left(\frac{P_1}{v_1} \right) \left(\left(\frac{P_2}{P_1} \right)^{\frac{2}{k}} - \left(\frac{P_2}{P_1} \right)^{\frac{k+1}{k}} \right) \right]^{\frac{1}{2}} \quad (33)$$

Here the subscript 1 refers to upstream values and 2 to downstream values. The constant A is proportional to flow area and is calculated initially for each turbine stage and extraction line using the initial steady state turbine conditions. Although it usually remains unchanged during a computational run, the flow area may be varied for turbine control.

The simulator iterates to solve a flow-pressure balance over each stage of the turbine. An example is shown in Fig. 5. Here the stream flow rate into 2 (W_1) must be equal to the flow rate down the extraction line (W_3), plus the flow rate through the next stage of the turbine (W_2) or

$$W_1 = W_2 + W_3 \quad (34)$$

These flow rates are calculated using Equation 33. In the simulation the pressure at point 2 would be iterated on until Equation 34 converges. The calculation would then move to point 3 and so on down the line from the high pressure to the low pressure end of the turbine. Points 5 and 6 in the example would be feedwater heaters. Point 7 might be the main condenser, brine heater, or moisture separator, depending on the turbine being calculated. The external pressures such as those in the feed heaters or brine heater are held constant during a turbine calculation.

After each flow rate vs. pressure computational pass through the turbine, a calculation of the steam expansion through the turbine is made. These calculations give the steam properties at each turbine stage boundary. The methods used here are those for a steady state turbine calculation as described in GER-2454A.⁴ Steam expansion, exhaust loss, and moisture removal calculations are made. The turbine stage efficiencies are assumed to be constant throughout the calculational run and are fixed numbers in each turbine subroutine.

The flow rate vs. pressure balance and steam conditions calculations are repeated until convergence is obtained. The electric power production is then calculated. The order in which the turbines are calculated are: low pressure, back pressure, and finally the high-pressure turbine.

ORNL-DWG 72- 12010

Fig. 5. Turbine Model

On completion of the high-pressure turbine calculation, the electrical outputs from the three turbine-generators are added. If this total differs from the required electrical demand, the flow constant in the first stage of the high-pressure turbine is varied. The high-pressure turbine is then recalculated under the changed conditions. This is repeated until the power production and electric demand converge. The convergence of power demand and production is optional in the code, i.e., the flow area iteration may be passed.

An option also exists in the code whereby the electrical production of the back-pressure turbine-generator can be held constant. This is accomplished by varying the back-pressure turbine first stage flow constant.

In the simulation there are assumed to be throttling valves upstream of the high-pressure and back-pressure turbines. The characteristics of such a valve are given in Fig. 6 of GER-2454A. The valve-wide-open flow rate is limited to 5% above rated (full power) flow.

Moisture Separator and Reheaters

The moisture separator and reheaters are made part of the high-pressure turbine model. The calculations are included within the high-pressure turbine subroutine and within its pressure-flow convergence loop.

As shown in Fig. 2, some of the steam leaving the high-pressure turbine passes directly into a moisture separator. This device is assumed to take in wet steam at its inlet and divide it into a saturated water stream and a saturated steam stream. The saturated water stream joins the remainder of the high-pressure turbine exhaust and goes to the shell side of the third feedwater heater. The saturated steam stream enters a two-stage reheater where it is superheated. The reheaters are treated as tube-in-shell heat exchangers with constant shell side temperatures (T_s). The temperature of the steam leaving each reheater is calculated as

$$T_o = T_s - (T_s - T_i) e^{-\left(\frac{AU}{C_p W}\right)} \quad (35)$$

In our simulation we assume that $\frac{AU}{C}$ remains constant for each reheater. The value of this quantity is calculated from initial turbine plant conditions. Separate calculations are made for each reheater, moving in the direction of steam flow.

The shell side steam in the first reheater stage comes from the first high-pressure turbine extraction point. The second reheater shell side steam comes directly from the steam generator. This steam condenses, transferring its heat of vaporization to the tube side steam. The shell side steam flow rate is calculated from a heat balance over each reheater.

Crossover Holdup

Other than the holdup provided by the component isolation assumption (one time step per component), no holdup times are included in the turbines, moisture separator, and reheater simulations. In order to account for steam holdup in these components and in the steam lines, a fictitious steam holdup tank is included in the simulation. This is located at the turbine crossover (between the second reheater and the other turbines). This steam holdup also helps the mathematical stability of the component isolation treatment by providing a stable sink pressure for steam flowing through the high-pressure turbine-reheater component, and by providing a stable source pressure for the low-pressure and back-pressure turbine flow calculations.

The heat and mass flow into and out of the crossover holdup is assumed to be constant during a calculational time step so as to be consistent with the component isolation treatment. At each time step, differential equations for the steam mass and total heat content in the crossover holdup are solved using Euler's method.

$$\frac{dM_{x\phi}}{dt} = W_{HP} - W_{LP} - W_{BP} - W_{FP} \quad (36)$$

$$\frac{dH_{x\phi}}{dt} = W_{HP} \cdot h_{HP} - h_{x\phi} (W_{LP} + W_{BP} + W_{FP}) \quad (37)$$

Knowing the mass, total heat content, and volume, the crossover enthalpy ($h_{x\phi}$) and pressure ($P_{x\phi}$) are calculated. The pressure is used to calculate turbine steam flow rates in the following time step.

Main Condenser

The location of the main condenser is shown in Fig. 2. In the simulator the main condenser is assumed to maintain a constant pressure of 2 in. HgA under all circumstances. The flow rate into the condenser from the low pressure and feed pump turbines and the flow out of the condenser to the feedwater heaters are calculated at each time step. A holdup mass of condensate is calculated using Euler's method.

$$\frac{dM_c}{dt} = W_{FP} + W_{LP} - W_{FW} \quad (38)$$

Since the temperature of the condensate is assumed to be constant at the temperature of saturated vapor at 2 in. HgA (101.4°F), the condenser holdup is of no significance to the time constant of the turbine plant.

Feed Pump Turbine

Some steam is taken from the crossover to run the feed pump turbine. The output of this turbine is used to run the feedwater pump. The feedwater pump is assumed to pressurize the feedwater to 1.23 times the steam generator pressure. At each time step the flow through the feed pump turbine is calculated so as to exactly meet the feed pump requirements.

Feedwater Heaters

A sketch of the feedwater heater model used in the simulator is shown in Fig. 6. There are seven such feed heaters in the simulated turbine plant. Their locations in the turbine plant are shown in Fig. 2.

Steam from the turbines enters the feedwater heater. If this steam is superheated, it is assumed first to give up this superheat to the feedwater. This is accomplished in the simulator by means of a simple heat balance. If the steam is wet, it is divided into a saturated steam part and a saturated liquid part,

Fig. 6. Feedwater Heater

$$W_s = W_v + W_L \quad (39)$$

The liquid falls into a liquid stream on a tray or partition within the feedwater heater. Liquid from a previous feedwater heater may also enter this stream. Since its temperature is usually hotter than the saturated vapor temperature, some of it flashes,

$$W_{DC,i} = W_{DC,f} + W_{DC,L} \quad (40)$$

This vapor goes into the vapor space. Steam in the vapor space condenses on the tubes containing the feedwater and falls into the liquid stream. This liquid eventually flows into the drain cooler section of the feedwater heater, and as it flows through the drain cooler it loses some of its heat to the feedwater.

The feedwater enters the feedwater heater in the tube bundles and flows counter to the direction of the drain coolant flow. After leaving the drain cooler, the feedwater enters tube bundles in the vapor space where it heats up some more as steam is condensed. Before leaving the heater, the feedwater in the tube bundle passes through a superheat section where the superheat in the turbine steam (if any) is removed.

In the simulation, the drain cooler section of the feedwater heater is treated essentially as a counterflow heat exchanger. The part where the feedwater tubes pass through the vapor space is treated as a tube in shell heater exchanger where the shell side has a uniform temperature. The superheat section is handled by the heat balance

$$W_{s,i} \cdot h_{S.H.} = W_{FW} \cdot \Delta h_{FW} \quad (41)$$

A lumped parameter model was constructed using nine differential equations to describe the dynamics of a single feedwater heater unit. Seven of the differential equations are for temperature and two are for mass. Flow diagrams for the model are shown in Fig. 7.

The feedwater temperatures in the drain cooler section are represented by equations 42 and 43.

Fig. 7. Flow Diagram of Simulated Feedwater Heater

$$\frac{dT_{FC,\phi}}{dt} = \frac{2W_{FW}}{M_{FC}} (T_{FC,A} - T_{FC,\phi}) + \frac{(UA)_{DC}}{(MC_P)_{FC}} (T_{DC,A} - T_{FC,\phi}) \quad , \quad (42)$$

$$\frac{dT_{FC,A}}{dt} = \frac{2W_{FW}}{M_{FC}} (T_{FW,i} - T_{FC,A}) + \frac{(UA)_{DC}}{(MC_P)_{FC}} (T_{DC\phi} - T_{FC,A}) \quad . \quad (43)$$

Equations 44 and 45 represent the feedwater temperatures in the evaporator, or vapor section,

$$\frac{dT_{FE\phi}}{dt} = \frac{2W_{FW}}{M_{FE}} (T_{FE,A} - T_{FE\phi}) + \frac{(UA)_{EV}}{(MC_P)_{FE}} (T_v - T_{FE,\phi}) \quad , \quad (44)$$

$$\frac{dT_{FE,A}}{dt} = \frac{2W_{FW}}{M_{FE}} (T_{FC,\phi} - T_{FE,A}) + \frac{(UA)_{EV}}{(MC_P)_{FE}} (T_v - T_{FE,A}) \quad . \quad (45)$$

The temperature of the coolant in the drain cooler is represented by equations 46 and 47,

$$\frac{dT_{DC,\phi}}{dt} = \frac{2W_{DC}}{M_{DC}} (T_{DC,A} - T_{DC,\phi}) - \frac{(UA)_{DC}}{(MC_P)_{DC}} (T_{DC,\phi} - T_{FC,A}) \quad , \quad (46)$$

$$\frac{dT_{DC,A}}{dt} = \frac{2W_{DC}}{M_{DC}} (T_L - T_{DC,A}) - \frac{(UA)_{DC}}{(MC_P)_{DC}} (T_{DC,A} - T_{FC,\phi}) \quad . \quad (47)$$

Equation 48 is the mass balance for the vapor,

$$\frac{dM_v}{dt} = W_v + W_{DC,f} - W_c \quad . \quad (48)$$

Equation 49 is the mass balance for the liquid stream in the vapor section,

$$\frac{dM_L}{dt} = W_{DC,L} + W_L + W_c - W_{DC} \quad . \quad (49)$$

Equation 50 represents the temperature of the liquid stream in the vapor section,

$$\frac{dT_L}{dt} = \frac{1}{M_L} (W_{DC,L} + W_L + W_c)(T_V - T_L) \quad (50)$$

For feedwater heaters without drain coolers (feedwater heaters 3 and 7 in simulated plant), equations 42, 43, 46, and 47 are not used.

The steam flowing into the feedwater heater divides into two streams,

$$W_s = W_v + W_L,$$

such that

$$W_s \cdot h_s = W_v h_v + W_L h_L \quad (51)$$

The liquid entering from a previous drain cooler divides into two streams,

$$W_{DC,i} = W_{DC,f} + W_{DC,L},$$

such that

$$W_{DC,i} \cdot h_{DC,i} = W_{DC,f} H_{fg} + W_{DC,L} \cdot h_{DC,L} \quad (52)$$

where H_{fg} and $h_{DC,L}$ are evaluated at the vapor temperature, T_v .

The vapor condensing rate (W_c) is calculated by

$$W_c = \frac{(UA)_{EV}}{2H_{fg}} (T_v - T_{FE,\phi}) + \frac{(UA)_{EV}}{2H_{fg}} (T_v - T_{FE,A}) \quad (53)$$

$(UA)_{EV}$ is the heat transfer coefficient times the area over the evaporator length. $(UA)_{DC}$ is the heat transfer coefficient times the area over the drain cooler length. These coefficients are assumed to be constant in the simulation.

The vapor pressure is calculated from the vapor mass by the same method used in the other component simulations.

$$P_v = \left(\frac{\partial P_{\text{sat}}}{\partial p_{\text{sat}}} / v_v \right) \cdot M_v + C \quad (54)$$

The drain coolant flow (W_{DC}) is assumed to be constant during a time step in order to be consistent with the component isolation assumption. It is calculated at the beginning of each time step by

$$W_{DC} = W_{DC,L} + W_L + W_C \quad (55)$$

The set of nine differential equations (45-53) are solved each time step using the matrix exponential method, as discussed in the MSF simulator¹ report.

In the initializing steady state calculation, the various masses (M_v , M_L , M_{DC} , M_{FC} , M_{FE}) are calculated, using a holdup time which is supplied as an input. Temperature input data includes the saturated vapor temperature (T_v), the inlet and outlet feedwater temperatures ($T_{FW,i}$, $T_{FW,\phi}$) and the drain coolant inlet and outlet temperatures ($T_{DC,i}$, $T_{DC,\phi}$). The flow rate input data include steam flow rate (W_s), feedwater flow rate (W_{FW}), and the drain coolant flow rates (W_{DC} , $W_{DC,i}$). From this data we know five of the nine state variables for each feedwater heater (M_v , M_L , $T_{FE,\phi}$, T_L , $T_{DC,\phi}$). The temperature of the feedwater leaving the drain cooler region ($T_{FC,\phi}$) may be calculated by means of a heat balance and a temperature from enthalpy subroutine.

$$W_{FW} (h_{FC,\phi} - h_{FW,i}) = W_{DC} (h_L - h_{DC,\phi}) \quad (56)$$

$T_{FE,A}$ is taken to be the feedwater temperature in the middle of the evaporator region

$$T_{FE,A} = T_v - (T_v - T_{FC,\phi}) e^{-\frac{1}{2} \frac{(UA)_{FW}}{W_{FW} \cdot C_P}} \quad (57)$$

where

$$\frac{(UA)_{EV}}{W_{FW} C_P} = \ln \frac{T_v - T_{FC,\phi}}{T_v - T_{FE,\phi}} \quad (58)$$

In the drain cooler section $T_{FC,A}$ and $T_{DC,A}$ are taken to be the temperatures halfway along the feedwater coolant pipe and the drain cooler, respectively. Following the analysis presented in Giedt⁵ pp. 346-347, these temperatures are obtained by solving two simultaneous equations

$$T_{DC,A} = T_{DC,i} - E_h (T_{DC,i} - T_{FC,A}) \quad , \quad (59)$$

$$T_{DC,\phi} = T_{DC,A} - E_h (T_{DC,A} - T_{FW,i}) \quad , \quad (60)$$

where

$$E_h = \frac{1 - e^{-\left[\frac{(UA)_{DC}}{(WC_P)_{FW}} \left(1 - \frac{(WC_P)_{FW}}{(WC_P)_{DC}} \right) \right]}}{1 - \frac{(WC_P)_{FW}}{(WC_P)_{DC}} e^{-\left[\frac{(UA)_{DC}}{(WC_P)_{FW}} \left(1 - \frac{(WC_P)_{FW}}{(WC_P)_{DC}} \right) \right]}} \quad . \quad (61)$$

Control Systems

Several automatic control systems are used in the simulator. These provide control of the reactor power, brine heater vapor pressure (back pressure turbine steam flow), and electric production.

The reactor power is regulated by simulated shim rod movement to add or subtract reactivity. An attempt was made here to model the shim control scheme as close as possible to that used in actual PWRs. The flow diagram for this control system is shown in Fig. 8.

After allowing for first-order lags for the thermocouples, the temperature signals from the hot and cold legs of the primary coolant loops are combined to form an average temperature (T_A). The difference between this temperature and a set point temperature (T_s) produces an error signal.

The set point temperature is a predetermined function of load. Since the pressure in the first stage of the high-pressure turbine is nearly proportional to load, both the temperature and reactor power set points are derived from this pressure.

The power set point signal (P_e) is compared with the reactor power signal (P_{WR}) to form a power error signal (P_e). This signal is passed

Fig. 8. Reactor Control System

through a first-order lag in order to damp out high frequency variations and to insure numerical stability. This compensated error signal is then passed through a high-pass filter (derivative circuit). This derivative of the power error signal is used to help compensate for transient power peaks.

The temperature error signal along with its derivative are combined with the derivative of the power error signal. This combined signal determines the rod speed and direction. In the simulation reactivity change rates due to shim movement range from 0.067 ϕ /sec to a maximum of 1 ϕ /sec depending on the size of the error signal. An error signal dead-band of $\pm 0.5^\circ\text{F}$ is currently used in the simulator.

The back pressure of the back-pressure turbine is the vapor pressure in the brine heater. The stability and performance of large MSF plants have been found to be sensitive to the vapor temperature in the brine heater.¹ A control system was therefore devised which will maintain back pressure by regulating the steam flow through the back-pressure turbine. The steam flow is controlled by changing the effective flow area in the first stage of the back-pressure turbine. The error signal uses both the difference between the back pressure and its set point and the time derivative of the back pressure

$$\frac{dA}{dt} = K_3 \frac{dP_{BH,V}}{dt} + K_4 \frac{P_{BH,V} - P_s}{P_s} \quad (62)$$

In the current simulator version K_3 and K_4 are equal to unity.

The overall electrical production is controlled by changing the steam flow area in the first stage of the high-pressure turbine. As described previously, the simulator will iterate on this area until the electrical power demand agrees with the electrical power produced. Such an option is also available to keep the electrical production of the back-pressure turbine-generator constant by adjusting its first stage flow area. If constant power is to be maintained in the back-pressure turbine, the code option which will maintain constant back pressure should not be selected.

SAMPLE RESULTS

In a sample calculation, the electric power demand to the turbine plant was linearly decreased by 10% over a 10-sec period. It was held at this lower demand for 15 min and then returned to the original power demand, again on a 10-sec ramp. Plots of some results from this simulation are given in Figs. 9 and 10.

The changed electric demand brings about some oscillation in the various plant variables shown. These oscillations are dampened, however, and new equilibriums are attained. After the demand is returned to design point, the variables should eventually return to approximately their original steady-state values. The 35 minute results are near the original values; however, due to the $1/2^{\circ}\text{F}$ deadband around the coolant temperature set point, the exact original steady state probably will not be attained.

The oscillations in the responses indicate that our simulation of the reactor control system is probably not perfect. It was, however, modeled after available information on PWR shim rod movement control.

Recent calculations show that the oscillations are reduced considerably by using less conservative reactivity coefficients than were used in the sample calculation. The values for the fuel and coolant reactivity coefficients (i.e., α_F and α_C in equation 18) used in the sample calculation were -1.0×10^{-5} and -2.5×10^{-5} $\delta\text{K/K} \cdot ^{\circ}\text{F}$, respectively. Calculations made using the reactor-steam generator-turbine plant portion of the simulator to compare with the measured response of the H. B. Robinson plant (a Westinghouse PWR owned by Carolina Power and Light Co.) indicate that these coefficients are too small. The use of fuel and coolant reactivity coefficients (-1.3×10^{-5} and 1.9×10^{-4} , respectively) appropriate to H. B. Robinson considerably reduce the oscillations.

In this calculation, the brine heater vapor pressure was controlled by regulating the back-pressure turbine flow as discussed in the Control Systems section. This control was effective for this particular transient. During the calculation the maximum deviation from the initial steady state liquid level in the MSF plant was held to less than 1 inch of water in any stage.

Fig. 9. Reactor Power and Average Coolant Temperature as a function of Time for a Negative 10% Electric Demand Pulse

Fig. 10. Steam Generator Pressure and Feedwater Temperature as a function of Time for a Negative 10% Electric Demand Pulse

REFERENCES

1. J. G. Delene and S. J. Ball, A Digital Computer Code for Simulating Large Multistage Flash Evaporator Desalting Plane Dynamics, ORNL-TM-2933, Oak Ridge National Laboratory (September 1971).
2. "Conceptual Design Study, Diablo Canyon Prototype Desalting Plant," Kaiser Engineers Report No. 71-60-RE, Oakland, California, January 1972.
3. T. D. Anderson, J. E. Jones, "Economic Scales of Options for Proposed California Prototype Desalting Plant," internal memorandum (Nov. 12, 1971).
4. F. G. Bailey, K. C. Cotton, and R. C. Spencer, "Predicting the Performance of Large Steam Turbine-Generators Operating with Saturated and Low Superheat Steam Conditions," Report GER-2454A, General Electric Company, presented at American Power Conference, Chicago, Illinois, April 25-27, 1967.
5. W. H. Giedt, Principles of Engineering Heat Transfer, D. Van Nostrand Company, Inc., January 1957.
6. C. T. Mothershed, ORCEF: A Fortran Code for the Calculation of Multi-Stage Flash Evaporator Desalination Plant Designs, USAEC Report ORNL-TM-1542, Oak Ridge National Laboratory, March 1966.
7. H. I. Bowers, ORCENT: A Digital Computer Program for Saturated and Low Superheat Steam Turbine Cycle Analysis, USAEC Report ORNL-TM-2395, Oak Ridge National Laboratory, January 1969.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

Appendix A

CODE STRUCTURE

The coupled-plant simulator consists of a main program, twenty major subroutines, and various physical property and other special functions and subroutines. The program parts and their functions are:

MAIN: The main calling program calls the subroutines in their proper sequence and does some miscellaneous calculations including those for the main condenser and feed pump turbine.

READ: Subroutine READ handles the inputting chores for the program and does some miscellaneous calculations. It is called at the beginning of a calculational run.

HPTURB: Subroutine HPTURB is called at each time step. It calculates the flow rate - pressure balance, steam conditions, and electrical production in and from the high-pressure turbine generator. The moisture separator and reheater calculations are also made here.

LPTURB: Subroutine LPTURB is called at each time step. It iterates on the flow rate - pressure balance and steam conditions within the low-pressure turbine. The low-pressure turbine generator electrical production is also calculated here.

BPTURB: Subroutine BPTURB is called at each time step. The back-pressure turbine flow rate - pressure, steam conditions, and electrical production are calculated here.

FHEAT: Subroutine FHEAT is called at each time step. It calculates the conditions within each of the system feed heaters.

FPIPE: Subroutine FPIPE calculates the feedwater inlet temperature to each feed heater at each time step.

RESET: Subroutine RESET is a bookkeeping subroutine for FHEAT. It resets or updates values of various parameters to be used by FHEAT in the next time step.

REACTR: Subroutine REACTR solves the equations describing the PWR and does the miscellaneous calculations associated with the reactor system at each time step.

SGEN: Subroutine SGEN solves the steam generator and related equations and at each time step.

EVAP: Subroutine EVAP is the main calling program for the MSF plant subroutines. This subroutine corresponds to the main program as reported in ORNL-TM-2933. It calls subroutines $\emptyset$ UT, IC $\emptyset$ N, SDATA, ZT $\emptyset$ T, AMATS, and UPSI.

$\emptyset$ UT: Subroutine $\emptyset$ UT handles the outputting chores for the MSF portion of the program.

IC $\emptyset$ N: Subroutine IC $\emptyset$ N is called at the beginning of a calculational run. It calculates the initial values of the MSF plant state variables and flow rates from input data. Certain quantities such as tube areas and tube brine mass which remain constant throughout the calculation are also calculated here.

SDATA: Subroutine SDATA is the principal subroutine for calculating MSF stage constants based on steady-state conditions. The constants calculated in SDATA are assumed to be invariant during the transient calculation.

ZT $\emptyset$ T: Subroutine ZT $\emptyset$ T is called at each time step for each MSF stage. In it are calculated the values of the first derivatives (Z_T) of the state variables. These Z_T 's are the rate change of the state variables at the beginning of a time step.

AMATS: Subroutine AMATS calculates the coefficients of the MSF stage A matrix. This subroutine is called at frequencies depending on criteria given in the input data.

UPSI: Subroutine UPSI calculates the values of the MSF stage state variables at the end of a time step based on the value at the beginning of the time step, the Z_T 's calculated in ZT $\emptyset$ T, and the current values of the components of the HP matrix for that stage.

BHEAT: Subroutine BHEAT calculates the temperature conditions within the brine heater and the pipes leading to and from the brine heater. Its initial call sets the initial brine heater conditions. During the transient calculation it is called once every time step. This subroutine corresponds to subroutine HEATER as reported in ORNL-TM-2933.

CHANGE: Subroutine CHANGE initially reads in, sets up, and stores information regarding future perturbations to the dual-purpose plant. At the time the perturbation is to occur, CHANGE is called and the

quantity in question is changed to its new value. At present the code can effect step changes in input seawater flow rate, seawater temperature, brine recycle fraction, and seawater rejection rate, and step or ramp reactivity changes, step or ramp changes in back-pressure turbine electric demand, ramp changes in entire plant electric demand, and step changes in the number of MSF trains serviced by the brine heater.

EMATS: Subroutine EMATS calculates the HP matrix used in the matrix exponential solution. This subroutine is called for the MSF plant, the feed-water heaters, the brine heater, the reactor and the steam generator.

RESTRT: Subroutine RESTRT may be called if desired at the end of the transient calculation. This subroutine punches a new set of initial condition data to be used if it is desired to restart the calculation from the point of termination. It corresponds to subroutine DATA in ORNL-TM-2933.

In addition to the major subroutines, the following physical property functions are also used by the code:

CP: Computes the specific heat of product and brine in the MSF plant as a function of temperature and salt content.

HFG: Computes H_2O heat of vaporization as a function of saturation temperature.

DEN: Computes the density of water as a function of temperature and salt fraction.

DVPSD: Computes the derivative of saturation pressure with respect to saturation density for a given saturation density.

DSTVP: Computes the derivative of H_2O saturation temperature with respect to saturation pressure as a function of saturation pressure.

HIN: Computes the tube inside heat transfer coefficient as a function of the tube brine temperature and flow rate.

HOUT: Computes the tube outside heat transfer coefficient as a function of tube temperature, vapor temperature, and fouling resistance.

WEQ: Computes flow per unit area given upstream and downstream pressure, upstream specific volume, and the specific heat ratio k (essentially solves equation 33).

PFVH: Calculates steam pressure from specific volume and enthalpy.

FIG8: Computes noncondensing stage group end point correction for exhaust loss in the turbine. Calculates value of exhaust loss vs pressure from Fig. 8 of GER-2451A.

FIG10: Computes turbine moisture removal stage effectiveness for a given pressure as plotted in Fig. 10 of GER-2454A.

UNIPOL: Computes a polynomial expansion used in FIG8 and FIG10.

TSAT: Computes the saturated temperature of water as a function of saturated pressure.

PSAT: Computes the saturated pressure of water as a function of saturated temperature.

TLIQH: Calculates the temperature of saturated water as a function of enthalpy.

HLIQ: Calculates the enthalpy of saturated water as a function of temperature.

SLIQ: Calculates the entropy of saturated water as a function of temperature.

VLIQ: Calculates specific volume of saturated water as a function of temperature.

HVAP: Calculates the enthalpy of steam as a function of temperature and pressure.

SVAP: Calculates the entropy of steam as a function of temperature and pressure.

VVAP: Calculates specific volume of steam as a function of temperature and pressure.

TVAPH: Calculates temperature of steam as a function of pressure and enthalpy.

TVAPS: Calculates temperature of steam as a function of pressure and entropy.

PROPH: This subprogram produces steam quality, entropy, and temperature of steam for a given pressure and temperature.

PROPS: This subroutine returns enthalpy of steam for a given pressure and entropy.

Appendix B

INPUT INSTRUCTIONS

The input data set contains five data subsets. These are a (1) general data section, (2) a MSF plant and brine heater data subset, (3) a turbine plant and feed heater data subset, (4) a reactor and steam generator data subset, and (5) a perturbation data set. Data for the MSF and brine heater data subset may be obtained from a steady-state MSF design code such as ORSEF.⁶ Data for the turbine plant and feedwater heater data subset may be obtained from a steady-state turbine design code such as ORCENT.⁷

The following are instructions for starting a new case from steady-state design code data. If restart data is used, the punched restart deck should be substituted for data subsets 2, 3, and 4. Data subsets 1 and 5 are common to all cases. The data should be ordered beginning with subset 1 and ending with 5.

1. General DataCARDS 1 and 2 (20A4)Column

1-80 General comments

CARD 3 (815)Column

1-5 (NK) Number of heat recovery stages

6-10 (NNK) Number of heat reject stages

11-15 (NPH) Number of feed heaters (%)

16-20 (NEWW) 0 use restart data

1 steady-state design code data

21-25 (ISF) Use 1 for now

26-30 (NTMSP) Number of time steps

31-35 (NE) Number of equations used to describe MSF stage dynamics (excluding concentration equations) (8)

36-40 (LCT) Maximum number of time steps between recalculation of an MSF stage A matrix

CARD 4 (6E12.6)

Column

1-12 (TI(1)) Time for brine leaving stage 1 to reach middle of brine heater

13-24 (TI(2)) Time for brine to travel from middle of brine heater to stage 1

25-36 (TI(3)) Calculated internally (time brine spends in brine heater)

37-48 (TI(4)) Initial time for calculation (sec)

49-60 (DT) Length of time step (sec)

61-72 (WMIN) Minimum permissible tube flow rate. (This is code dependent and must be greater than $\frac{WTUBE * \text{MAX}(TI(1), TI(2))}{199 * DT}$, where WTUBE $\equiv$ tube brine flow rate in evaporator (lb/sec))

CARDS 5 and 6 (6E12.6)

CNC(N), N = 1,8 The maximum fractional change in each of the MSF state variable allowed before the A matrix is recalculated. (Fractional changes in the temperatures are made using a Fahrenheit base.)

CARD 5

Column

1-12 (CNC(1)) Fractional change in tube outlet temperature

13-24 (CNC(2)) Fractional change in average tube brine temperature

25-36 (CNC(3)) Fractional change in average tube temperature

37-48 (CNC(4)) Fractional change in vapor mass

49-60 (CNC(5)) Fractional change in tray brine temperature

61-72 (CNC(6)) Fractional change in tray brine mass

CARD 6Columns

- 1-12 (CNG(7)) Fractional change in product mass
 13-24 (CNG(8)) Fractional change in product temperature

CARD 7 (16I5)Column

5	0 - skip
1 - print A and HP matrix whenever they are calculated
10	0 - skip
1 - punch restart data
11-15	Number of time steps between printing of state variable values
20	0 - skip printing of FD and SD arrays. Also skip printing of A and HP matrix altogether.
1 - print initial value of A and HP matrix and then as indicated by INL. Print FD and SD arrays.
25	0 - skip
1 - print other information when printing state variable values
30	Not used
35	Not used
40	0 - skip
1 - punch cards to be used in plotting routine whenever printing stage results
45	0 - allow electric production to float
1 - control overall electric production equal to electric demand.
50	0 - allow back-pressure turbine electric production to float
1 - hold back-pressure turbine electric production to a set point demand
55	Not used

60 0 - allow brine heater pressure to float
 1 - hold brine heater pressure to set point by regulating back-pressure turbine flow

65 Not used

70 Not used

75 Not used

80 Not used

CARD 8 (8I5)

This card included only if Column 40 of Card 7 is not equal to 0.

Column

5 (IPP) State variable whose value is to be punched.

6-40 Stages whose value for IPP is to be punched, I5 Format, 7 stages needed.

2. MSF Plant Data

CARDS 1 through 9 (6E12.5) FD(I), I = 1,54

<u>FD</u>	<u>Identification</u>	<u>Section*</u>	<u>Unit</u>
1	Tube inside diameter	H	in.
2	Tube inside diameter	E	in.
3	Tube inside diameter	R	in.
4	Tube outside diameter	H	in.
5	Tube outside diameter	E	in.
6	Tube outside diameter	R	in.
7	Tube conductivity	H	Btu/hr.ft. °F
8	Tube conductivity	H	Btu/hr.ft. °F
9	Tube conductivity	R	Btu/hr.ft. °F
10	Fouling resistance	H	hr.ft ² . °F/Btu
11	Fouling resistance	E	hr.ft ² . °F/Btu
12	Fouling resistance	R	hr.ft ² . °F/Btu
13	Not used		
14	Number of modules in design data	-	
15	Tube inside heat transfer coefficient	H	Btu/hr.ft ² . °F

<u>FD</u>	<u>Identification</u>	<u>Section</u>	<u>Unit</u>
16	Shell volume per module	E	ft ³
17	Shell volume per module	R	ft ³
18	Number of tubes	E	
19	Number of tubes	R	
20	Steam temperature	H	°F
21	Brine tube flow rate	E	lb/hr
22	Brine tube flow rate	R	lb/hr
23	Brine heater outlet temperature	H	°F
24	Brine heater inlet temperature	H	°F
25	Tube liquid velocity	H	ft/sec
26	Tube liquid velocity	E	ft/sec
27	Tube liquid velocity	R	ft/sec
28	Sump salt concentration (M)		
29	Brine Heater Tube Brine Salt Concentration	H	
30	Seawater salt concentration		
31	Brine blowdown temperature	-	°F
32	(Not used)	-	
33	(Not used)	-	
34	(Not used)	-	
35	Tube outlet temperature	R	°F
36	Brine recycle flow rate	-	lb/hr
37	Tube length	H	ft
38	Not used	-	
39	Not used	-	
40	Not used	-	
41	Tube heat capacity per unit surface area	H	Btu/ft ² . °F
42	Tube heat capacity per unit surface area	E	Btu/ft ² . °F
43	Tube heat capacity per unit surface area	R	Btu/ft ² . °F
44	(Not used)	-	
45	Sump holdup time	-	sec
46	Steam volume	H	ft ³
47	Number of trains per brine heater	H	
48	(Not used)	-	
49	Downstream level effect fraction	-	

<u>FD</u>	<u>Identification</u>	<u>Section*</u>	<u>Unit</u>
50	(Not used)	-	
51	Vapor volume fraction (M)	-	
52	Not used		
53	(Not used)	-	
54	(Not used)	-	

CARDS 10 through (9 + 4 × number of stages) (6E12.5) SD(I) I = 1,24

This set of data includes four cards for each stage.

<u>SD</u>	<u>Identification</u>	<u>Unit</u>	<u>Ref.</u>
1	Tray brine outlet flow rate	lb/hr	A
2	Product flashing flow rate	lb/hr	B
3	Product temperature	°F	C
4	Brine flashing flow rate	lb/hr	B
5	Product outlet flow rate	lb/hr	A
6	Tray brine outlet temperature	°F	C
7	Saturated vapor temperature	°F	D
8	Tube inlet temperature	°F	C
9	(Not used)	-	
10	(Not used)	-	
11	Tray brine residence time	sec	
12	Tube length	ft	C
13	Vapor condensing pressure	psia	C
14	(Not used)	-	
15	Tube inside heat transfer coefficient	Btu/hr·ft ² ·°F	C
16	Tubing surface area	ft ²	C
17	(Not used)		
18	Tray brine outlet concentration	-	C
19	Tray brine level	in.	C,M
20	Product tray level	in.	C,M
21	Product holdup time	sec	C,M
22-24	(Not used)	-	

*H = brine heater, E = evaporation, and R = heat reject.

Reference

- A Calculated from mass balance
- B Calculated from heat balance
- C Used directly
- M Modified data needed, do not use design code data directly.

3. Turbine and Feedwater Heater Data

CARDS 1-8 High-pressure turbine data (see Fig. A-1)(6E12.5)

CARDS 1-2 Flow array, lb/sec, (WHP(12))
 3-4 Temperature array, °F, (TH(12))
 5-6 Enthalpy array, Btu/lb, (HH(12))
 7-8 Pressure array, psia, (PH(12))

CARDS 9-16 Low pressure turbine data (see Fig. A-2)(6E12.5)

CARDS 9-10 Flow array, lb/sec, (WLP(12))
 11-12 Temperature array, °F, (TL(12))
 13-14 Enthalpy array, Btu/lb, (HL(12))
 15-16 Pressure array, psia, (PL(12))

CARDS 17-19 Back-pressure turbine data (see Fig. A-3)(6E12.5)

CARD 17 Temperature array, °F, (TB(6))
 18 Enthalpy array, Btu/lb, (HB(6))
 19 Pressure array, psia, (PB(6))

CARD 20 Miscellaneous turbine plant data (6E12.5)

Column

1-12 (WBPT) Steam flow through back-pressure turbine, lb/sec
 13-24 (WBHO) Condensate flow leaving brine heater, lb/sec
 25-36 (WFPT) Steam flow through feed pump turbine, lb/sec
 37-48 (TFP) Temperature of feed pump turbine exhaust, °F
 49-60 (HFP) Enthalpy of feed pump turbine exhaust, Btu/lb
 61-72 (PFP) Feed pump turbine exhaust pressure, psia

Fig. A-1. High Pressure Turbine Data Layout

Fig. A-2. Low Pressure Turbine Data Layout

Fig. A-3. Back Pressure and Feed Pump Turbines Data Layout

CARD 21 Feedwater heater drain cooler indicator (7I5)

0 Drain cooler equations not used

1 Drain cooler equations utilized

CARDS 22-28 Feedwater heater data (6E12.5)
One card for each of seven feed heaters

Column

1-12 (WDC) Condensate flow out of feedwater heater, lb/sec
13-24 (WFW) Feedwater flow through feedwater heater, lb/sec
25-36 (TFWI) Temperature of feedwater entering feedwater heater, °F
37-48 (TFW ϕ) Temperature of feedwater leaving feedwater heater, °F
49-60 (TDC ϕ) Temperature of condensate leaving feedwater heater, °F
61-72 (PFHV) Steam pressure in feedwater heater, psia

CARDS 29-35 Feedwater heater holdup time data (4E10.3)
One card for each of seven feed heaters

Column

1-10 (HFWC) Holdup of feedwater in drain cooler, sec.
11-20 (HDC) Holdup of condensate in drain cooler, sec.
21-30 (HFWE) Feedwater holdup in evaporator section, sec.
31-40 (HVFH) Steam holdup in feedwater heater, sec.

4. Reactor and Steam Generator Data (6E12.5)

CARD 1

Column

1-12 (DCAY) Delay neutron precursor decay constant, sec⁻¹
13-24 (BETA) Delay neutron fraction
25-36 (PNLT) Prompt neutron lifetime, sec
37-48 (ETA) Fraction of power deposited in coolant
49-60 (CPF) Specific heat of fuel rod, Btu/lb.°F
61-72 (PCF) Fuel temperature reactivity coefficient, reactivity/°F

CARD 2Column

1-12 (TCC)	Coolant temperature reactivity coefficient, reactivity/°F
13-24 (FCM)	Mass of fuel elements, lb
25-36 (CCM)	Mass of coolant in core, lb
37-48 (CLM)	Mass of coolant in lower plenum, lb
49-60 (CUM)	Mass of coolant in upper plenum, lb
61-72 (PWR)	Reactor power estimate, kW(t)

CARD 3Column

1-12 (CDNP)	Delay neutron precursor estimate
13-24 (TFA)	Average fuel temperature, °F
25-36 (TCA)	Average coolant temperature, °F
37-48 (TCCI)	Estimate of coolant temperature entering core from lower plenum, °F
49-60 (TSGPI)	Estimate of coolant temperature leaving reactor vessel, °F
61-72 (WRP)	Primary coolant flow rate, lb/sec

CARD 4Column

1-12 (TCRI)	Temperature of coolant entering reactor vessel, °F
13-24	Not used
25-36 (SQSG)	Quality of steam leaving steam generator boiling region
37-72	Not used

CARD 5

Blank card

CARD 6Column

1-12 (HSG)	Enthalpy of steam leaving steam generator, Btu/lb
13-24 (HSGI)	Enthalpy of feedwater entering steam generator, Btu/lb

25-36 (WSG)	Steam flow from steam generator, lb/sec
37-48 (WSGI)	Feedwater flow into steam generator, lb/sec
49-60 (TSG)	Temperature of steam leaving steam generator, °F
61-72 (PSG)	Steam pressure in steam generator, psia

5. Perturbation Data

Read maximum of 20 cards (110, 3E10.3), last card must have 0 in column 10 to be blank.

<u>Column</u>	
10	Type of perturbation (1-10, see below)
11-20	Time at which perturbation occurs, sec
21-30	Perturbation data
1	Fractional change in seawater intake flow rate (e.g., = 1.05 for 5% increase inflow)
2	Change in seawater temperature, °F
3	Not used
4	Fractional change in brine recycle flow
5	New seawater reject fraction
6	Step reactivity change
7	Fouling factor change multiple
8	Step change in electric demand to back-pressure turbine (.05 means 5% increase based on full power).
9	Step change in overall electric demand (-.05 means 5% decrease based on full power output)
10	Fractional change in number of MSF trains through brine heater (0.5 reduces number to 1/2 of previous)
31-40	Perturbation data
1-5	Not used
6	Ramp reactivity rate
7	Not used
8	Ramp change in electric demand from back-pressure turbine (0.01 means 1% per second rate of increase)

- 9 Ramp change in overall electric demand (-.01 means
1%/sec rate of decrease)
- 10 Not used

APPENDIX C

FORTRAN LISTING

C MAIN PROGRAM FOR DEMONSTRATION PLANT SIMULATOR

C
C

```

IMPLICIT REAL*8(A-H,O-Z)
COMMON /GEN1/ TIME, DT, TNXT
COMMON /GEN3/ PSHP, ELHP, PSLP, ELLP, PSBP, ELBP
COMMON /GEN4/ IPRT
COMMON /EVAP5/ WSEA, WTUBE, FRSEA, FRTB, SMPMS, TPSUMP, TSEA,
1 TMAX, DEAIR, SSUMP, CSMP, CPSMP, CSEA
COMMON /EVAP6/ SD(24), FD(54), CNG(8), TI(4), TITLE(20), EXPL(20)
1, ST, WMIN
COMMON /EVAP7/ NK, NNK, NTMSP, NF, ICT, NSTGS, ISECT, NARROW, NEWW
1, NRT, IND(16), IERROR, IUP, ISM(50), ICT(50)
COMMON /FEED1/ VFH(8), WFW(8), HFWC(7), WDC(8), HDC(7), HFEW(7),
1 TFHV(8), TFEI(7), TFWO(8), TFWI(8), TDCO(7), FHSQ(7), WFHS(8),
2 HFHS(8), TDCI(8), TFEQ(7), PFHV(8)
COMMON /FEED2/ IDC(7), NFH
COMMON /FEED3/ WFHC(7), WDWI(7), WDLI(7), WFHV(7), WFHL(7)
COMMON /FEED4/ HFWI(7), HFWO(7), HFEI(7), HDGO(7), HDGI(7), HFHV(7)
1, HFEQ(7)
COMMON /FEED5/ FHMI(7), TFCA(7), TDCA(7), TFEA(7), FHMV(7), TFHI(7)
COMMON /BEAT1/ TBHV, TBHO, CBHO, WBHC, PBHV, WBHI, WBHO, SQUAL
COMMON /RHTR/ HRHO, TRHO, VMAS, HOLD, VOLD, CONM
COMMON /SGEN1/ HSG, TSG, PSG, WSG, WSGI, CSGW, PSGI
COMMON /FPUMP/ HFPI, HFPO
COMMON /TRB1/ WVWO, WVWOBP, PRTBP, CNGBP, PART, CNGEL, VLVH, VLVBP
COMMON /TRBH/ WHP(12), TH(12), HH(12), PH(12)
COMMON /TRBL/ WLP(12), TL(12), HL(12), PL(12)
COMMON /TRBB/ TB(6), HB(6), PB(6), WBPT, WFPT, TFP, HFP, PFP
COMMON /FDAT1/ XFH(9,7), UADC(7), UAEV(7)
COMMON /SGEN2/ HSGI, TSGI, HSGPI, HSGPO
COMMON /MAT2/ RAB(7,7), HPR(7,7), DTR
COMMON /REAC1/ XR(7), ZRT(7), RFD(30)
COMMON /REAC2/ WRP, TFA, TCA, TCRI, PWR, FCM, CCM, CLM, CUM, TCRO
COMMON /REAC3/ RHOP, RHO, AP, BP, AMPL, FREQ, TPRT
COMMON /SGEN3/ XSG(10), ZTSG(10), ASG(10,10), HPSG(10,10)
COMMON /STEAM1/ TSGPO, TSGPI
DATA ETAFPT / 0.80D0/
IPRT = 1
CN1 = 0.0D0
CN2 = 1.0D0
CNGEL = 0.0D0
CNGBP = 0.0D0
CALL READ(0,1)
C READ(1) READS IN GENERAL CASE INFORMATION AND FD ARRAY
TIME = TI(4)
TMAX = TIME + DT*FLOAT(NTMSP)
CALL EVAP(0)
C EVAP IS MAIN PROGRAM FOR THE MSF EVAPORATOR, EVAP(0) INITIALIZES
C EVAPORATOR AND READS IN SD ARRAYS
C READ(3) READS W,T,H, AND P'S
CALL READ(0,3)

```

```

C   READ(4) READS IN FEED HEATER DATA
    CALL READ(0,4)
    CALL READ(0,5)
    CALL READ(0,6)
    CALL READ(0,7)
    IF(NEWW.EQ.0) GO TO 4
    CONM = 10.00*(WLP(11) + WLP(12) + WFPT)
    TRHO = TH(6)
    VLVBP = PRTPP
    VLVH = PART
    WVMOBP = 1.0500*WBPT /VLVBP
    WVMQ = WHP(1)*1.0500/VLVH
    CALL GPIPC(0)
4  CONTINUE
    WRITE(6,104) TIME
    CALL FHEAT(0)
    WRITE(6,108)
    WRITE(6,109)((I,TFWD(I),TDCO(I),TFWI(I),TFHV(I),WFW(I),WDC(I),
1  TFEI(I)),I = 1,NFH)
    WRITE(6,103)
    WRITE(6,101)((I,HFWI(I),HFEI(I),HFEO(I),HFWD(I),HDCI(I),HDCO(I),
1  HFHV(I),HFHS(I),PFHV(I)),I = 1,NFH)
    IF(NEWW.EQ.0) GO TO 6
C   STEAM HOLDUP VALUES, 2.5 SEC. HOLDUP
    VMAS = WHP(5)*2.500
    HOLD = VMAS*HVAP(TH(6),PH(6))
    VOLD = VVAP(TH(6),PH(6))*VMAS
    WRITE(6,106) HH(6), PH(6), VMAS
6  CONTINUE
    CALL BPTURB(0)
    CALL LPTURB(0)
    CALL HPTURB(0)
    SELN = (ELHP + ELLP + ELBP)/PART
    HSGI = HFWO(1)
    CALL SGEN(0)
    CALL REACIR(0)
    CALL BHEAT(0)
    IN3 = IND(3)
    IPT = 1
C   END STEADY STATE CALCULATION
    WRITE(6,107)
    DO 10 N = 1,NTMSP
    IF(N.EQ.NTMSP) IND(5) = 1
    TIME = TIME + DT
    IPRT = 0
    IF(IPT.LT.IN3) GO TO 34
    WRITE(6,104) TIME
    IPRT = 1
    IPT = 0
34  IPT = IPT + 1
    CALL LPTURB(N)
    IF(PRTBP.GT.CN1) GO TO 36
    ELBP = CN1
    WBPT = CN1
    WPHQ = CN1
    GO TO 37
36  CONTINUE
    CALL BPTURB(N)

```

```

37 CONTINUE
DO 40 J = 1,20
CALL HPTURB(N)
TOTEL = ELHP + ELLP + ELBP
IF(IND(9).EQ.0) GO TO 42
DMND = PART*SELN
DIF = DMND - TOTEL
IF(DARS(DIF).LE.0.100) GO TO 41
DIFAC = DIF/ELHP
IF(DARS(DIFAC).GT.0.0600) DIFAC = DSIGN(0.0600,DIFAC)
WHP(1) = WHP(1)*(CN2 + DIFAC)
C FLOW THROUGH H-P TURBINE VALVE NEVER GREATER THAN WVWO
WHP(1) = DMIN1(WHP(1),WVWO)
IF(WHP(1).GE.WVWO) GO TO 41
WSG = WHP(1) + WHP(11)
40 CONTINUE
GO TO 41
42 WHP(1) = WSG - WHP(11)
41 CONTINUE
IF(IPRT.EQ.0) GO TO 35
WRITE(6,112)
WRITE(6,113) ((I,HH(I),PH(I),TH(I),HL(I),PL(I),TL(I),HB(I),PB(I),
1TB(I)), I=1,6)
WRITE(6,114)((I,HH(I),PH(I),TH(I),HL(I),PL(I),TL(I)), I=7,12)
WRITE(6,110)
WRITE(6,111) PSHP,PSLP,PSBP,ELHP,ELLP,ELBP
35 CONTINUE
CALL FHEAT(N)
IF(PRTBP.LE.CN1) GO TO 38
CALL BHEAT(N)
38 CONTINUE
HSG1 = HFWD(1)
CALL SGEN(N)
CALL REACTR(N)
CALL FPIPE(N)
CALL PROPH(A11,S11,TH(6),PH(6),HH(6))
CALL PROPS(H43S,PPF,S11)
HFP = HH(6) - (HH(6) - H43S)*ETAFTP
WFPT = WFW(1)*(HFWD - HFPI)/(HH(6) - HFP)
PART = PART + DT*CNDEL
PRTBP = PRTBP + DT*CNGBP
VMASS = VMASS + (WHP(5) - WBPT - WFPT - WLP(1))*DT
HOLD = HOLD + (WHP(5)*HRHO - HH(6)*(WBPT + WFPT + WLP(1))*DT
VSP = VOLD/VMASS
HH(6) = HOLD/VMASS
CALL PFVH(HH(6),VSP,PH(6))
HB(1) = HH(6)
HL(1) = HH(6)
TH(6) = TVAPH(PH(6),HH(6))
IF(IPRT.EQ.0) GO TO 11
WRITE(6,106) HH(6), PH(6), VMASS
11 CONTINUE
WFHS(1) = WHP(6) + WHP(11)
WFHS(2) = WHP(7) + WHP(10)
WFHS(3) = WHP(8) + WHP(9)
WFHS(4) = WLP(7)
WFHS(5) = WLP(8)
WFHS(6) = WLP(9)

```

C

```

113 FORMAT(15,9F10.3)
114 FORMAT(15,6F10.3)
10 CONTINUE
   IF(IND(2).EQ.0) GO TO 20
   VL VH = WHP(1)*1.05D0/WVW0
   CALL RESTRT
20 CONTINUE
   STOP
   END

```

```

SUBROUTINE READ(KK,N)
  IMPLICIT REAL*8(A-H,O-Z)
C THIS SUBROUTINE READS DATA FOR COUPLED PLANT AND SETS INITIAL VALUES
  COMMON /GEN1/ TIME, DT, TNXT
  COMMON /REAC1/ XR(7), ZRT(7), RFD(30)
  COMMON /REAC2/ WRP, TFA, TCA, TCRI, PWR, FCM,CCM,CLM,CUM,TCR0
  COMMON /SGEN1/ HSG, TSG, PSG, WSG, WSGI, CSGW, PSGI
  COMMON /SGEN2/ HSGI, TSGI, HSGPI, HSGPO
  COMMON /SGEN3/ XSG(10), ZTSG(10), ASG(10,10), HPSG(10,10)
  COMMON /EVAP6/ SD(24), FD(54), CNG(8), TI(4), TITLE(20), EXPL(20)
1, ST, WMIN
  COMMON /EVAP7/ NK, NNK, NTMSP, NE, LCT, NSTGS, ISECT, NARROW, NEWW
1, NRT, IND(16), IERROR, IUP, ISM(50), ICT(50)
  COMMON /BEAT1/ TBHV, TBHO, CBHO, WBHC, PBHV, WBHI, WBHO, SQUAL
  COMMON /BEAT2/ TBHC, HBI, BHT(6)
  COMMON /TRB1/ WVW0, WVW0BP, PRTBP, CNGBP, PART, CNGEL, VLVH,VLVBP
  COMMON /FEED1/ VFH(8), WFH(8), HFHC(7), WDC(8), HDC(7), HFHE(7),
1 TFHV(8), TFEI(7), TFWO(8), TFWI(8), TDCO(7), FHSO(7), WFHS(8),
2 HFHS(8), TDCI(8), TFEO(7), PFHV(8)
  COMMON /FEED2/ IOC(7), NFH
  COMMON /FEED4/ HFHI(7), HFHO(7), HFEI(7), HDCO(7), HDCI(7), HFHV(7)
1, HFEO(7)
  COMMON /FEED5/ FHMI(7), TFCA(7), TDCA(7), TFEA(7), FHMV(7), TFHI(7)
  COMMON /FDAT1/ XFH(9,7), UADC(7), UAEV(7)
  COMMON /FDAT2/ B5FH(7), POCFH(7), FWCN(7), FHDN(7), FWEM(7)
  COMMON /FDAT3/ B5SG, PDCSG, FCON, UAB, UANB
  COMMON /FDAT4/ UAFC, UAQI
  COMMON /STEAM1/ TSGPO, TSGPI
  COMMON /TRBH/ WHP(12), TH(12), HH(12), PH(12)
  COMMON /TRBL/ WLP(12), TL(12), HL(12), PL(12)
  COMMON /TRBB/ TB(6), HB(6), PB(6), WBPT,WFPT, TFP, HFP, PFP
  COMMON /RHTR/ HRHO, TRHO, VMAS, HOLD, VOLD, COMM
  COMMON /MPLQT/ IPP, IP1, IP2, IP3, IP4, IP5, IP6, IP7
  DIMENSION HVFH(7)
  GO TO (1,2,3,4,5,6,7),N
1 CONTINUE
C TITLE AND GENERAL EVAPORATOR INFORMATION
  READ(5,102) EXPL
  READ(5,102) TITLE
  READ(5,103) NK,NNK,NFH ,NEWW,NRT,NTMSP,NE,LCT
  CALL IDAY(TODAY)
  WRITE(6,124) TODAY
124 FORMAT(1H1,' THIS CASE WAS RUN ON ',A8)
  WRITE(6,102) EXPL
  WRITE(6,102) TITLE
  READ(5,104) TI,DT,WMIN

```

```

      READ(5,104) CNG
      READ(5,103) IND
      IF(IND(8).EQ.0) GO TO 33
      READ(5,115) IPP, IP1, IP2, IP3, IP4, IP5, IP6, IP7
33  CONTINUE
C   READ IN FIXED DATA---FD ARRAY
      READ(5,104) FD
      WRITE(6,105) FD
      GO TO 200
2  CONTINUE
C   READ STAGE DATA SD PUNCHED BY K-25 S.S. CODE
      READ(5,104) SD
      WRITE(6,107) KK
      WRITE(6,106) SD
      GO TO 200
3  CONTINUE
C   READ GENERAL RESTART DATA
      IF(NEWW.EQ.0) READ(5,111) TRHQ, VMAS, HOLD, VOLD, CONM
      IF(NEWW.EQ.0) READ(5,104) VLVH, VLVBP, WVNO, WVWOSP
      READ(5,104) PART, PRTOP
C   READ IN H-P TURBINE DATA
      READ(5,101) WHP
      READ(5,101) TH
      READ(5,101) HH
      READ(5,101) PH
      WRITE(6,114)
      WRITE(6,112)((I, WHP(I), TH(I), HH(I), PH(I)), I = 1, 12)
C   READ IN L-P TURBINE DATA
      READ(5,101) WLP
      READ(5,101) TL
      READ(5,101) HL
      READ(5,101) PL
      WRITE(6,116)
      WRITE(6,112)((I, WLP(I), TL(I), HL(I), PL(I)), I = 1, 12)
C   READ IN B-P TURBINE DATA
      READ(5,101) TB
      READ(5,101) HB
      READ(5,101) PB
      WRITE(6,117)
      WRITE(6,113)((I, TB(I), HB(I), PB(I)), I = 1, 6)
      READ(5,101) WBPT, WBHO, WFPT, TFP, HFP, PFP
      WRITE(6,118) WBPT, WBHO, WFPT, TFP, HFP, PFP
C   NOW SET UP INITIAL VALUES
      WBHI = WBPT
      PBHV = PB(6)
      TBHV = TSAT(PBHV)
      PPHV(8) = PL(12)
      GO TO 200
4  CONTINUE
      WFHS(1) = WHP(6) + WHP(11)
      WFHS(2) = WHP(7) + WHP(10)
      WFHS(3) = WHP(8) + WHP(9)
      WFHS(4) = WLP(7)
      WFHS(5) = WLP(8)
      WFHS(6) = WLP(9)
      WFHS(7) = WLP(10)
      WFHS(8) = WLP(11) + WLP(12) + WFPT
      WFHS(1) = (WHP(6)*HH(3) + WHP(11)*HH(12))/WFHS(1)

```

```

HFHS(2) = (WHP(7)*HH(4) + WHP(10)*HH(11))/WFHS(2)
HFHS(3) = (WHP(9)*HH(10) + WHP(8)*HH(9))/WFHS(3)
HFHS(4) = HL(2)
HFHS(5) = HL(3)
HFHS(6) = (WLP(3)*HL(4) - WLP(4)*HL(8))/WFHS(6)
HFHS(7) = (WLP(4)*HL(5) - WLP(5)*HL(9))/WFHS(7)
HFHS(8) = (WLP(5)*HL(6) - WLP(6)*(HL(10) - HL(7)) +
1 WFPT*HFP)/WFHS(8)
READ(5,108) IDC
IF(NEWW.EQ.1) GO TO 40
DO 42 I = 1,NFH
  READ(5,109)((XFH(J,I),J=1,9),WFW(I),WDC(I),UADC(I),UAEV(I)
1,TFWI(I),TDCI(I),PFHV(I))
  READ(5,111)(FWEM(I),FWCM(I),FHD(I),B5FH(I),PDCFH(I))
  CALL RESET(I)
42 CONTINUE
GO TO 200
40 CONTINUE
DO 10 I = 1,NFH
C  READ IN DATA SET FOR EACH FEEDHEATER
  READ(5,101) WDC(I), WFW(I), TFWI(I), TFWO(I), TDCO(I), PFHV(I)
10 CONTINUE
  WRITE(6,120)
  WRITE(6,119)((I,WDC(I),WFW(I),TFWI(I),TFWO(I),TDCO(I),PFHV(I)),
1 I=1,NFH)
  DO 19 I = 1,NFH
    READ(5,110) HFMC(I), HDC(I), HFWE(I), HVFH(I)
    TFHV(I) = TSAT(PFHV(I))
    PD = 1.00/VVAP(TFHV(I),PFHV(I))
    FHMV(I) = HVFH(I)*WFHS(I)
    VFH(I) = FHMV(I)/PD
    HDCI(I) = HLIQ(TFHV(I))
    TFHI(I) = TFHV(I)
21 HFWO(I) = HLIQ(TFWO(I))
    HFWI(I) = HLIQ(TFWI(I))
    HFHV(I) = HVAP(TFHV(I),PFHV(I))
    HDCO(I) = HLIQ(TDCO(I))
C  TEMP. OF FEEDWATER ENTERING EVAPORATOR SECTION (FROM HEAT BALANCE)
    IF(HFHS(I) - HFHV(I)) 13,13,14
13 TFEQ(I) = TFWO(I)
    HFEO(I) = HFWO(I)
    GO TO 15
14 HFEO(I) = HFWO(I) - WFHS(I)*(HFHS(I) - HFHV(I))/WFW(I)
    TFEQ(I) = TLIQH(HFEO(I))
15 CONTINUE
    IF(IDC(I))11,12,11
11 DCHT = WDC(I)*(HDCI(I) - HDGO(I))
    HFEI(I) = HFWI(I) + DCHT/WFW(I)
    TFEI(I) = TLIQH(HFEI(I))
    GO TO 19
12 TFEI(I) = TFWI(I)
    HFEI(I) = HFWI(I)
19 CONTINUE
    GO TO 200
5 CONTINUE
C  N = 5 IS FOR REACTOR INFORMATION
  READ(5,104) RFD
  WRITE(6,121)

```



```

WRITE(6,122) RFD
IF(NEWW.NE.0) GO TO 200
READ(5,111) ((XR(I), I = 1,7),UAFC,UAOI)
121 FORMAT(1H0,' RFD DATA')
122 FORMAT(1H0,1P8E12.5)
GO TO 200
6 CONTINUE
C N = 6 IS FOR STEAM GENERATOR INFORMATION
READ(5,104) HSG, HSGI, WSG, WSGI, TSG, PSG
WRITE(6,123) HSG, HSGI, WSG, WSGI, TSG, PSG
IF(NEWW.NE.0) GO TO 200
READ(5,111) (XSG(I), I = 1,10)
READ(5,111) BSSG, PDCSG,FCON, UAB, UANB
GO TO 200
7 CONTINUE
CALL CHANGE(0)
GO TO 200
200 CONTINUE
101 FORMAT(6E12.5)
102 FORMAT(20A4)
103 FORMAT(16I5)
104 FORMAT(6E12.5)
105 FORMAT(3HOFD/(1H ,1P10E11.4))
106 FORMAT(1H ,1P10E11.4)
107 FORMAT(21H0STAGE DATA FOR STAGE,I5)
108 FORMAT(7I5)
109 FORMAT(8F10.4)
110 FORMAT(4E10.3)
111 FORMAT(5E12.5)
112 FORMAT(I5,4F10.3)
113 FORMAT(I5,3F10.3)
114 FORMAT(1H0,' HIGH PRESSURE TURBINE DATA'/
1 ' POINT FLOW TEMPERATURE ENTHALPY PRESSURE')
115 FORMAT(8I5)
116 FORMAT(1H0,' LOW PRESSURE TURBINE DATA'/
1 ' POINT FLOW TEMPERATURE ENTHALPY PRESSURE')
117 FORMAT(1H0,' BACK PRESSURE TURBINE DATA'/
1 ' POINT TEMPERATURE ENTHALPY PRESSURE')
118 FORMAT(1H0,' WBPT =',F10.3,' WBHD =',F10.3,' WFPT =',F10.3,' TFP
1 =',F10.3,' HFP =',F10.3,' PFP =',F10.3)
119 FORMAT(I5,6F12.3)
120 FORMAT(1H0,' HEATER WDC WFW 1FW1 1FW2
1 TDCO PFHV')
123 FORMAT(1H0,' HSG = ',F10.3,' HSGI = ',F10.3,' WSG = ',F10.3,
1 ' WSGI = ',F10.3,' TSG = ',F10.3,' PSG = ',F10.3)
RETURN
END

```

```

SUBROUTINE HPTURB(IT)
IMPLICIT REAL*8(A-H,O-Z)
C THIS SUBROUTINE SOLVES FOR STEAM CONDITIONS FOR HIGH PRESSURE
C TURBINE USED IN COUPLED PLANT
COMMON /GEN3/ PSHD, FLHD, PSLP, ELLP, PSBP, ELBP
COMMON /GEN4/ IPRT
COMMON /EVAP7/ NK, NTMSP, NE, LCT, NSTGS, ISECT, NARROW, NEWH
1, ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)

```

```

COMMON /SGEN1/ HSG, TSG, PSG, WSG, WSGI, CSGW, PSGI
COMMON /FEED1/ VFH(8), WFW(8), HFWC(7), WDC(8), HDC(7), HFE(7),
1 TFHV(8), TFEI(7), TFWO(8), TFWI(8), TDCQ(7), FHSQ(7), WFHS(8),
2 HFHS(8), TDCI(8), TFEQ(7), PFHV(8)
COMMON /TRB1/ WVHO, WVHOBP, PRTBP, CNGBP, PART, CNGEL, VLVH, VLVBP
COMMON /TRBH/ WHP(12), TH(12), HH(12), PH(12)
COMMON /RHTR/ HRHO, TRHO, VMAS, HOLD, VOLD, CONH
DIMENSION ETAHP(4), SP(5)
DIMENSION WSL(5), AHP(4), WEL(3), AHX(3), CF(5)
DATA C1,ETAG1 / 3.41214D+6,0.9837D0/
DATA ETAHP / 0.808D0, .8582D0, 0.8582D0, 0.8582D0/
DATA CF / 2.0D0,2.0D0,2.0D0,2.0D0,2.0D0 /
C IF(IT.LE.0) CALCULATE THE FLOW EQUATION CONSTANTS
IF(IT.GE.1) GO TO 100
C FOR THE TIME BEING USE AN ISONTROPIC EXPANSION COEFFICIENT OF 1.33
CK = 1.33D0
DO 211 J = 1,5
WEL(J) = WHP(J)
IF(J.GE.4) GO TO 211
WEL(J) = WHP(J+5)
211 CONTINUE
VU = VVAP(TH(2),PH(2))
AHP(1) = WSL(1)/WEQ(PH(2),PH(3),VU,CK)
DO 210 K = 1,3
M = K + 1
I = K+2
VU = VVAP(TH(I),PH(I))
IF(K.EQ.3) M = K + 2
AHP(I-1) = WSL(M) /WEQ(PH(I),PH(I+1),VU,CK)
AHX(K) = WEL(K)/WEQ(PH(I),PFHV(K),VU,CK)
210 CONTINUE
PR7 = (PH(5) - PH(7))/(PH(5) - PH(6))
PR10 = (PH(5) - PH(8))/(PH(5) - PH(6))
UARH1 = WHP(5)*DLOG((TH(3) - TH(8))/(TH(3) - TH(7)))
UARH2 = WHP(5)*DLOG((TSG - TRHO)/(TSG - TH(8)))
WRITE(6,107) PR7,PR10,UARH1,UARH2
107 FORMAT(1P5D12.5)
WRITE(6,105) AHP
WRITE(6,106) AHX
105 FORMAT(1H0,' AHP ARRAY'//,1P4D12.5)
106 FORMAT(1H0,' AHX ARRAY'//,1P3D12.5)
GO TO 200
100 CONTINUE
C CALCULATE FLOW-PRESSURE BALANCE IN H-P TURBINE
HH(1) = HSG
C H(1) AND HSG ARE SAME QUANTITY, ALSO PSG = P(1) AND ISG = I(1)
TFR = WHP(1)/WVHO
C ASSUME VALVE OPERATES PER FIG. 6 OF GER 2454A
PH(1) = PSG*(1.00 - 0.085D0 + 0.009D0/(1.2D0 - TFR))
DO 110 JJ = 1,20
WU = WHP(1)
DO 120 I = 1,5
SP(I) = PH(I)
IF(I.EQ.1) GO TO 130
IM = I + 1
IK = I - 1
M = I - 2
VD = VVAP( TH(I), PH(I))

```

```

IF(I.EQ.2) GO TO 137
WU = AHP(I) *WEO(PH(I),PH(I),V1,CK)
137 CONTINUE
WSL(IK) = AHP(IK) *WEO(PH(I),PH(IH),VD,CK)
IF(I.LE.4) GO TO 134
PH(7) = PH(5) - PR7*(PH(5) - PH(6))
PH(8) = PH(5) - PR10*(PH(5) - PH(6))
TH(7) = TSAT(PH(7))
HV7 = HVAP(TH(7),PH(7))
HL7 = HLIQ(TH(7))
HV5 = HH(9)
HH(7) = HV7
FW10 = (HV7 - HL7)/(HV5 - HL7)
WSL(5) = WSL(4)
WSL(IK) = WSL(IK)*FW10
134 CONTINUE
IF(I.EQ.2) GO TO 131
WEL(M) = 0.000
IF(PFHV(M).GE.PH(I)) GO TO 135
WEL(M) = AHEX(M)*WEO(PH(I),PFHV(M),VD,CK)
135 WD = WEL(M) + WSL(IK)
IF(I.GT.3) GO TO 132
TH(8) = TH(3) - (TH(3) - TH(7))*DEXP(UARH1/WSL(5))
HH(8) = HVAP(TH(8),PH(8))
HH(11) = HLIQ(TH(3))
WHP(10)=WSL(5)*(HH(8) - HH(7))/(HH(3) - HH(11))
WD = WD + WHP(10)
GO TO 132
131 WD = WSL(1)
WER = WU - WD
IF(DABS(WER/WU).LE.0.1D-3) GO TO 121
DIVID = WD/(PH(2) - PH(3))
GO TO 133
132 CONTINUE
WER = WU - WD
IF(DABS(WER/WU).LE.0.1D-3) GO TO 121
DPU = PU - PH(I)
DPD = PH(I) - PH(IH)
DPCX = PH(I) - PFHV(M)
DIVID = DMAX1(WU/DPU,(WD-WEL(M))/DPD,WEL(M)/DPCX)
133 CONTINUE
DELP = CF(I)*WER/DIVID
PH(I) = PH(I) + DELP
121 CONTINUE
136 PU = PH(I)
V1 = VD
130 CONTINUE
CALL PROPH(A,S,TH(I),PH(I),HH(I))
IF(I.GE.5) GO TO 15
CALL PROPS(HS,PH(I+1),S)
HH(I+1) = HH(I) - ETAHP(I)*(HH(I) - HS)
ETAHP(I) = H-P TURBINE EFFICIENCY, GROUP I
120 CONTINUE
15 HH(9) = HH(5) + FIG8(PH(5))
TRHO = TSG - (TSG - TH(8))*DCXP(UARH2/WSL(5))
HRHO = HVAP(TRHO,PH(6))
HH(12)= HLIQ(TSG)
WHP(11)=WSL(5)*(HRHO -HH(8))/(HS6 - HH(12))

```

```

      DO 112 I = 1,5
      IF(DABS( PH(I) - SP(I)).GT.0.1D-2) GO TO 110
112 CONTINUE
      GO TO 113
110 CONTINUE
      WRITE(6,102)
102 FORMAT(1H0,' NON CONVERGENCE IN J LOOP AFTER 20 ITERATIONS')
113 CONTINUE
      PH(9) = PH(5)
      TH(9) = TH(5)
      HH(10) = HL7
      TH(10) = TH(7)
      PH(10) = PH(7)
      TH(11) = TH(3)
      PH(11) = PH(3)
      TH(12) = TSG
      PH(12) = PSG
      DO 115 J = 1,5
      WHP(J) = WSL(J)
      IF(J.GE.4) GO TO 115
      WHP(J+5) = WEL(J)
115 CONTINUE
      WHP(12) = WHP(6) + WHP(10)
      WHP(9)=WSL(4)*(1.D0 - 1.D0/FW10)
200 CONTINUE
      WHHP = WSL(1)*(HH(1) - HH(3)) + WSL(2)*(HH(3) - HH(4)) +
1 WSL(3)*(HH(4) - HH(9))
C   WHHP IS SUM OF W*(DELTA H) FOR H-P TURBINE
      PSHP = 3.6D3*WHHP/C1
C   PSHP IS POWER TO HIGH PRESSURE TURBINE SHAFT IN MW
C   C1 IS A CONVERSION FACTOR =3412140 BTU/MW-HR
      ELHP = PSHP*ETAG1
C   ELHP = GROSS OUTPUT (MWE)
C   ETAG1 = GENERATOR EFFICIENCY FOR HP, LP SHAFT
      IF(IPRT.EQ.0) GO TO 116
      WRITE(6,201) (WSL(I),I = 1,5)
      WRITE(6,202) (WEL(I),I = 1,3)
      WRITE(6,203) WHP(10), WHP(11)
116 CONTINUE
201 FORMAT(1H0,' H-P TURBINE FLOW ARRAY'//,5F10.3)
202 FORMAT(1H0,' EXTRACTION LINE FLOWS'//,3F10.3)
203 FORMAT(1H0,' WHR1 =',F10.3,' WHR2 =',F10.3)
      RETURN
      END

```

```

SUBROUTINE LPTURB(IT)
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /GEN3/ PSHP, ELHP, PSLP, ELLP, PSBP, ELBP
  COMMON /GEN4/ IPRT
  COMMON /FEED1/ VFH(8), WFW(8), HFWC(7), WDC(8), HDC(7), HFWE(7),
1 TFHV(8), TFEI(7), TFWO(8), TFWI(8), TDCQ(7), FHSQ(7), WFHS(8),
2 HFHS(8), TDCI(8), TFEQ(7), PFHV(8)
  COMMON /TRBH/ WHP(12), TH(12), HH(12), PH(12)
  COMMON /TRBL/ WLP(12), TL(12), HL(12), PL(12)
  DIMENSION ALP(7), ALEX(5), SP(7), ETALP(6), CF(6)
  DATA C1, ETAG1 / 3.41214D+6, 0.9837D0/

```

```

DATA ETALP / .92D0, .92D0, .92D0, .8639D0, .847D0, .8499D0 /
DATA ANNTOT / 742.986D0 /
DATA CF / 2.0D0, 1.5D0, 1.5D0, 2.0D0, 2.0D0, 2.0D0 /
C ASSUMING FOR THE TIME BEING THAT THE STAGE GROUP EFFICIENCY,
C 'ETA' IS A CONSTANT
C THIS SUBROUTINE SOLVES FOR ENTHALPY AT VARIOUS EXTRACTION POINTS
C IN THE LOW PRESSURE TURBINE IN COUPLED PLANT STUDY
C THE STEAM FLOW AND PRESSURE BALANCE IS SOLVED FIRST
C LOW PRESSURE TURBINE FLOW AND PRESSURE BALANCE
IF(IT.GE.1) GO TO 100
IF(IT.LE.0) CALCULATE THE FLOW EQUATION CONSTANTS
C FOR THE TIME BEING USE AN ISONTROPIC EXPANSION COEFFICIENT OF 1.33
CK = 1.33D0
VCON = VVAP(TL(12), PL(12))
PUP = PH(6)
VUP = VVAP(TH(6), PUP)
ALP(1) = WLP(1)/WEQ(PUP, PL(1), VUP, CK)
DO 10 I = 1, 6
K = 2 + I
VU = VVAP(TL(I), PL(I))
ALP(I+1) = WLP(I)/WEQ(PL(I), PL(I+1), VU, CK)
IF(I.EQ.1) GO TO 10
ALEX(I-1) = WLP(I+5)/WEQ(PL(I), PFHV(K), VU, CK)
10 CONTINUE
WRITE(6,105) ALP
WRITE(6,106) ALEX
105 FORMAT(1H0, ' ALP ARRAY' /, 1P7D12.5)
106 FORMAT(1H0, ' ALEX ARRAY' /, 1P5D12.5)
GO TO 200
100 CONTINUE
DO 110 M = 1, 20
DO 114 I = 1, 6
SP(I) = PL(I)
114 CONTINUE
V1 = VVAP(TH(6), PH(6))
PU = PH(6)
DO 120 I = 1, 6
C CALCULATE FLOW-PRESSURE BALANCE IN L-P TURBINE
WU = ALP(I)*WEQ(PU, PL(I), V1, CK)
VD = VVAP(TL(I), PL(I))
WLP(I) = ALP(I+1)*WEQ(PL(I), PL(I+1), VD, CK)
IF(I.EQ.1) GO TO 131
WLP(I+5) = ALEX(I-1)*WEQ(PL(I), PFHV(I+2), VD, CK)
WD = WLP(I+5) + WLP(I)
GO TO 132
131 WD = WLP(I)
132 CONTINUE
WER = WU - WD
IF(DABS(WER/WU).LE.0.1D-3) GO TO 121
DPU = PU - PL(I)
DPD = PL(I) - PL(I+1)
DPEX = PL(I) - PFHV(I+2)
IF(I.GT.1) GO TO 134
DIVID = DMAX1(WU/DPU, WD/DPD)
DPEX = DPD
GO TO 135
134 DIVID = DMAX1(WU/DPU, WD/DPD, WLP(I+5)/DPEX)
135 DELP = CF(I)*WER/DIVID

```

```

      ABSV = DABS(DEL P)
      SGM = DEL P/ABSV
      DEL P = DMIN1(ABSV,DPU,DPD,DPEX)
      PL(I) = PL(I) + DEL P*SGM
      GO TO 136
121 CONTINUE
136 PU = PL(I)
      V1 = VD
      CALL PROPH(A,S,TL(I),PL(I),HL(I))
      HRET = HL(I)
      IF(I.LE.3) GO TO 20
      A2 = A - 0.01DO*FIG10(PL(I))*A
      HL(I+4) = A2*HLIQ(TL(I)) + (1.DO - A2)*HVP(TL(I),PL(I))
      S = (1.DO - A2)*SVAP(TL(I),PL(I)) + A2*SLIQ(TL(I))
      HRET = HL(I+4)
20 CALL PROPS(HS,PL(I+1),S)
      HL(I+1) = HRET - ETALP(I)*(HRET - HS)
120 CONTINUE
      CALL PROPH(A,S,TL(7),PL(7),HL(7))
C     NOW CALCULATE ENTHALPY OF EXHAUST FROM L-P TURBINE (UEEP H)
C     VANN = ANNULUS VELOCITY, FT./SEC.
C     VCON = SPECIFIC VOLUME IN CONDENSER
C     ANNTOT = LAST STAGE ANNULUS AREA, FT**2
      VANN = WLP(6)*VCON*(1.DO - A)/ANNTOT
      EL=117.915D0-VANN*(.759471D0 - VANN*(2.45557D-3 - VANN*(5.16029D-6
1 - VANN*(7.29996D-9 - VANN*(6.45073D-12 - VANN*(3.32163D-15 -
2 VANN*(8.86298D-19 - VANN*(8.97546D-23))))))
C     EL = EXHAUST LOSS (BTU/LB-DRY FLOW)
      HL(11) = HL(7) + EL*0.87D0*(1.0D0 - A)*(1.0D0 - 0.65D0*A)
      DO 112 I = 1,6
      IF(DABS(PL(I) - SP(I)).GT.0.1D-2) GO TO 110
112 CONTINUE
      GO TO 113
110 CONTINUE
      WRITE(6,102)
102 FORMAT(1H0,' NON CONVERGENCE IN J LOOP AFTER 20 ITERATIONS')
113 CONTINUE
      DO 25 J = 1,4
      TL(J+7) = TL(J+3)
      PL(J+7) = PL(J+3)
25 CONTINUE
      WLP(12) = WLP(6)
200 CONTINUE
      WHLP = WLP(1)*(HL(1) - HL(2)) + WLP(2)*(HL(2) - HL(3)) +
1 WLP(3)*(HL(3) - HL(4)) + WLP(4)*(HL(8) - HL(5)) +
2 WLP(5)*(HL(9) - HL(6)) + WLP(6)*(HL(10) - HL(11))
      PSLP = 3.6D3*WHLP/C1
      ELLP = PSLP*ETAG1
      IF(IPRT.EQ.0) GO TO 116
      WRITE(6,201)(WLP(I),I = 1,6)
      WRITE(6,202)(WLP(I),I = 7,12)
116 CONTINUE
201 FORMAT(1H0,' L-P TURBINE FLOW ARRAY'//,6F10.3)
202 FORMAT(1H0,' EXTRACTION LINE FLOWS'//,6F10.3)
      RETURN
      END

```

```

SUBROUTINE BPTURB(IT)
  IMPLICIT REAL*8(A-H,O-Z)
C   THIS SUBROUTINE CALCULATES ENTHALPY OF STEAM ENTERING BRINE HEATER
C   THE EXHAUST PRESSURE IS EQUAL TO BRINE HEATER VAPOR PRESSURE
  COMMON /GEN1/ TIME, DT, TNXT
  COMMON /GEN3/ PSHP, ELHP, PSLP, ELLP, PSBP, ELBP
  COMMON /GEN4/ IPRT
  COMMON /EVAP6/ SD(24), FD(54), CNG(8), TI(4), TITLE(20), EXPL(20)
  1, ST, WMIN
  COMMON /EVAP7/ NK, MNK, NTMSP, NE, LCT, NSTGS, ISECT, NARROW, NEWM
  1, ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)
  COMMON /BEAT1/ TBHV, TBHO, CBHO, WBHC, PBHV, WBHI, WBHO, SQUAL
  COMMON /TRB1/ WVWQ, WVWOBP, PRTBP, CNGBP, PART, CNGEL, VLVH, VLVBP
  COMMON /TRBH/ WHP(12), TH(12), HH(12), PH(12)
  COMMON /TRBB/ TB(6), HB(6), PB(6), WBPT, WFPT, TFP, HFP, PFP
  DIMENSION ABPT(4), SP(5), ETABP(4)
  DATA ETABP, C1, ETAG2/0.9039D0, .9039D0, .9039D0, 0.9039D0, 3.41214D+6,
  1 0.98D0/
C   CALCULATE FLOW CONSTANTS
  IEL = 0
  IF(IT.GE.1) GO TO 100
  CK = 1.297D0
  PBPS = PBHV
  IF(NEWM.EQ.0) PBPS = FD(54)
  IF(NEWM.NE.0) FD(54) = PBHV
  DO 50 I = 1, 4
    VD = VVAP(TB(I), PB(I))
    ABPT(I) = WBPT/WEO(PB(I), PB(I+1), VD, CK)
  50 CONTINUE
  ABPT(1) = ABPT(1)/VLVBP
  WRITE(6,103) ABPT, PBPS
103 FORMAT(1H0, ' BACK PRESSURE TURBINE FLOW CONSTANTS = ', 1P4D12.5, //
  1 ' BACK PRESSURE SET POINT = ', F8.3, ' PSIA')
  GO TO 200
100 CONTINUE
  WU = WBPT
  PB(5) = PBHV
  HB(1) = HH(6)
  DELBP = PRTBP*SELBP
  IF(DELBP.GT.0.01D0) GO TO 12
  WBPT = 0.000
  WBHI = 0.000
  ELBP = 0.000
  PSBP = 0.000
  GO TO 17
  12 CONTINUE
  IF(IND(12).EQ.0) GO TO 43
  IF(WU.GE.WVWOBP) GO TO 43
  VLVBP = VLVBP - ((PBHV - PVS)/DT + (PBHV - PBPS))/PBPS
  43 CONTINUE
  DO 60 K = 1, 20
    TFRBP = WU/WVWOBP
    PD = PH(6)*(1.00 - .005D0 + .009D0/(1.200 - TFRBP))
    DO 14 I = 1, 5
      SP(I) = PB(I)
  14 CONTINUE
  TB(1) = TVAPH(PD, HH(6))
  PB(1) = PD

```

```

VU = VVAP(TB(1),PB(1))
DO 20 I = 1,5
IF(I.EQ.1) GO TO 30
IF(I.EQ.5) GO TO 30
VFAC = 1.00
IF(I.EQ.2) VFAC = VLVBP
WU = VFAC*ABPT(I-1)*WEQ(PB(I-1),PB(I),VU,CK)
WU = DMIN1(WVWOBP,WU)
VD = VVAP(TB(I),PB(I))
WD = ABPT(I)*WEQ(PB(I),PB(I+1),VD,CK)
WER = WU - WD
IF(DABS(WER/WU).LE.0.1D-3) GO TO 21
DPU = PB(I-1) - PB(I)
DPD = PB(I) - PB(I+1)
DIVID = DMAX1(WU/DPU,WD/DPD)
DELP = 2.00D0*WER/DIVID
PB(I) = PB(I) + DELP
21 CONTINUE
VU = VD
30 CONTINUE
WBPT = WD
CALL PROPH(A,S,TB(I),PB(I),HB(I))
IF(I.EQ.5) GO TO 15
CALL PROPS(HS,PB(I+1),S)
HB(I+1) = HB(I) - ETABP(I)*(HB(I) - HS)
20 CONTINUE
15 HB(6) = HB(5) + FIG8(PBHV)
DO 16 I = 2,4
IF(DABS(PB(I) - SP(I)).GT.0.1D-2) GO TO 60
16 CONTINUE
GO TO 13
60 CONTINUE
13 CONTINUE
200 CONTINUE
PVS = PBHV
WBHI = WBPT
CALL PROPH(A55,S55,TB(6),PBHV,HB(6))
PB(6) = PBHV
SQAL = 1.00D0 - A55
WHBP = WBPT*(HB(1) - HB(6))
PSBP = 3.6D3*WHBP/C1
ELBP = PSBP*ETAC2
IF(IT.GE.1) GO TO 41
SELBP = ELBP/PRTBP
GO TO 40
C IF IND(10).NE.0 ATTEMPT TO MAINTAIN B-P TURBINE ELECTRICAL OUTPUT
41 IF(IND(10).EQ.0) GO TO 40
DIFEL = DELBP - ELBP
IF(DIFEL.GE.0.0D0.AND.WBPT.GE.WVWOBP) GO TO 40
IF(DABS(DIFEL).LE.0.1D-1) GO TO 40
IEL = IEL + 1
IF(IEL.GT.20) GO TO 42
VLVBP = VLVBP*(DELBP/ELBP)
IF(IPRT.EQ.0) GO TO 43
WRITE(6,102) IT,IEL,ELBP,VLVBP
GO TO 43
42 WRITE(6,105)
WRITE(6,102) IT,IEL,ELBP,VLVBP

```



```

40 CONTINUE
   IF(IPRT.EQ.0) GO TO 116
   WRITE(6,202) VLVBP
202  FORMAT(1H0,' B-P TURBINE VALVE SETTING = ',F8.5)
   WRITE(6,201) WBPT, PB(1), HB(6), SQUAL, HB(1), PBHV
116 CONTINUE
102  FORMAT(1H0,' IT = ',I5,' IEI = ',I3,' ELBP = ',F8.3,' VLVBP = ',F8.5)
105  FORMAT(1H0,' NON- CONVERGENCE IN B-P TURBINE ELECTRICAL LOOP')
201  FORMAT(1H0,' WBPT = ',F9.3,' PBPI = ',F8.3,' HBHI = ',F9.3,' SQUAL = '
      1F9.5,' HBPI = ',F9.3,' PBHV = ',F8.3)
17  RETURN
   END

```

```

SUBROUTINE FHEAT(NSS)
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /FEED1/ VFH(8), WFW(8), HFWC(7), WDC(8), HDC(7), HFWE(7),
1  TFHV(8), TFEI(7), TFWO(8), TFHI(8), TDCO(7), FHSQ(7), WFHS(8),
2  HFHS(8), TDCI(8), TFEQ(7), PFHV(8)
  COMMON/FEED2/ IDC(7), NFH
  COMMON/FEED3/ WFHC(7),WDVI(7),WDLI(7),WFHV(7),WFHL(7)
  COMMON/FEED4/ HFWI(7), HFWO(7), HFEI(7), HDCO(7), HDCI(7), HFHV(7)
1, HFEQ(7)
  COMMON /FEED5/ FHHI(7), TFCA(7), TDCA(7), TFEA(7), FHMV(7), TFHI(7)
  COMMON /MAT3/ AFH(9,9,7), HPFH(9,9,7)
  COMMON/GEN1/ TIME,DT,TNXT
  COMMON /GEN4/ IPRT
  COMMON /EVAP7/ NK, NMK, NTMSP, NE, LCT, NSTGS, ISECT, NAPROW, NEWW
1, ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)
  COMMON /SGEN1/ HSG, TSG, PSG, WSG, WSGI, CSGW, PSGI
  COMMON /FDAT1/ XFH(9,7), UADC(7), UAEV(7)
  COMMON/ FDAT2/ B5FH(7), PDCFH(7), FWCN(7), FHDM(7), FWEM(7)
  DIMENSION HDLI(7), CPFE(7), CPDC(7), CPFW(7)
  DIMENSION ZTFH(9,7), SX(9,7), CPI(7), TS(9), CNG(9)
  DIMENSION NCT(7), LFH(7)
  DIMENSION DTD(7)
  DATA CNG / .01D0, .01D0, .01D0, .01D0, .01D0, .01D0, .01D0, .01D0 /
  IF(NSS.GT.0) GO TO 200
C  IF NSS EQUALS 0, CALCULATE INITIAL STEADY STATE NUMBERS
  CN1 = 0.0D0
  CN2 = 1.0D0
  CN3 = 0.5D0
  LCT = 120
  IFH = 2
  DO 120 I = 1,7
  DO 119 J = 1,9
  DO 121 K = 1,9
    AFH(I,K,I) = CN1
121 CONTINUE
    ZTFH(J,I) = CN1
119 CONTINUE
    LFH(I) = 0
    IF(NEWM.EQ.0) GO TO 120
    XFH(1,I) = CN1
    XFH(2,I) = CN1
    XFH(5,I) = CN1
    XFH(6,I) = CN1

```

```

120 CONTINUE
  IF(NEWM.EQ.0) GO TO 112
  DO 110 I = 1,NFH
    PD = 1.D0/VVAP(TFHV(I),PFHV(I))
    B5FH(I) = DVPSD(PD)/VFH(I)
    PDCHF(I) = PFHV(I) - B5FH(I)*FHMV(I)
    XFH(8,I) = WDC(I)*HDC(I)
  C  MASS OF FEEDWATER IN EVAPORATOR SECTION
    FWEH(I) = WFW(I)*HFEI(I)
    XFH(3,I) = TFEQ(I)
    XFH(7,I) = FHMV(I)
    XFH(9,I) = TFHV(I)
    CPI(I) = CPW(XFH(9,I))
    CPFE(I) = (HFEQ(I) - HFEI(I))/(TFEQ(I) - TFEI(I))
    TOP = TFHV(I) - TFEI(I)
    BOT = TFHV(I) - TFEQ(I)
    COEF = CN3*DLOG(TOP/BOT)
    TFEA(I) = TFHV(I) - TOP*DEXP(-COEF)
    XFH(4,I) = TFEA(I)
    MCP = WFW(I)*CPFE(I)
    UAEV(I) = MCP*(TFEQ(I) - TFEA(I))/BOT
  C  IF THERE IS NO DRAIN COOLER FEED WATER HEATING SKIP NEXT SECTION
  C  IF(IDC(I).EQ.0) GO TO 111
  C  MASS OF FEED WATER IN DRAIN COOLER
    FWC(I) = WFW(I)*HFWC(I)
  C  MASS OF PRIMARY WATER IN DRAIN COOLER
    FHD(I) = WDC(I)*HDC(I)
    XFH(1,I) = TFEI(I)
    XFH(5,I) = TDCQ(I)
    CPDC(I) = (HDCI(I) - HDCQ(I))/(TFHV(I) - TDCQ(I))
    CPFW(I) = (HFEI(I) - HFWI(I))/(TFEI(I) - TFWI(I))
    DT1 = TFHV(I) - TFEI(I)
    DT2 = TDCQ(I) - TFWI(I)
    DTML = (DT1 - DT2)/DLOG(DT1/DT2)
    CH = WDC(I)*CPDC(I)
    CC = WFW(I)*CPFW(I)
    UA = CH*(TFHV(I) - TDCQ(I))/DTML
    XNTU = CN3*UA/CH
    RAT = CH/CC
    COEF = -XNTU*(CN2 - RAT)
    EXPN = DEXP(COEF)
    EH = (CN2 - EXPN)/(CN2 - RAT*EXPN)
    WRITE(6,136) CH,EXPN,RAT,COEF,XNTU,UA,CH,CC,DTML
136  FORMAT(1P9D12.5)
    TFCA(I) = TFHV(I) + TDCQ(I)/(EH*(CN2 - EH)) - TFHV(I)/EH
    1 - TFWI(I)/(CN2 - EH)
    TDCA(I) = TFHV(I) - EH*(TFHV(I) - TFCA(I))
    XFH(2,I) = TFCA(I)
    XFH(6,I) = TDCA(I)
    UADC(I) = CH*(TDCA(I) - TDCQ(I))/(TDCQ(I) - TFCA(I))
    GO TO 110
111 CONTINUE
    FWC(I) = CN1
    FHD(I) = CN1
    UADC(I) = CN1
110 CONTINUE
112 CONTINUE
    DO 113 I = 1,7

```

```

      DTDP(I) = DSTVP(PFHV(I))
      DN 122 L = 1,9
122 CONTINUE
      SX(L,I) = XFH(L,I)
113 CONTINUE
      WRITE(6,134)
      WRITE(6,135)((I,FHVV(I),FWCM(I),FHDM(I),FWEM(I),UADC(I),
1 UAEV(I),      B5FH(I),DTDP(I)),I = 1,NFH)
134 FORMAT(1H0,' HEATER FHVV      FWCM      FHDM      FWEM      UADC
1      EAEV      B5FH      DTDP')
135 FORMAT(15,OP6F10.3,1P2D12.5)
      GO TO 79
200 CONTINUE
C      IF NSS IS GREATER THAN 0, DO A TRANSIENT CALCULATION
C      OTHER PORTIONS OF THE PROGRAM.
      DO 210 I = 1,NFH
C      COMPUTE NEW AVERAGE SPECIFIC HEATS
C      STEAM FLOW INTO FEEDHEATERS AND ITS ENTHALPY ARE CALCULATED IN
      IF(IDC(I).EQ.0) GO TO 20
      CPDC(I) = (HDCI(I) - HUCU(I))/(TFHI(I) - TDCQ(I))
      CPFW(I) = (HFEI(I) - HFMI(I))/(TFEI(I) - TFWI(I))
C      ZTUT TYPE EQUATIONS, VARIABLES 1,2,5,6
      B0 = 2.00*WFW(I)/FWCM(I)
      B1 = 2.00*UADC(I)/(FWCM(I)*CPFW(I))
      ZTFH(1,I) = B0*XFH(2,I) - (B0 + B1)*XFH(1,I) + B1*XFH(6,I)
      ZTFH(2,I) = -(B0 + B1)*XFH(2,I) + B1*XFH(5,I) + B0*TFWI(I)
      B7 = 2.00*WDC(I)/FHDM(I)
      B6 = 2.00*UADC(I)/(FHDM(I)*CPDC(I))
      ZTFH(5,I) = B7*XFH(6,I) - (B7 + B6)*XFH(5,I) + B6*XFH(2,I)
      ZTFH(6,I) = B7*XFH(9,I) - (B7 + B6)*XFH(6,I) + B6*XFH(1,I)
      IF(LFH(I).GT.0) GO TO 20
C      A MATRIX FOR DRAIN COOLER
      AFH(1,1,I) = -B1
      AFH(1,2,I) = B0
      AFH(1,6,I) = B1
      AFH(2,2,I) = -(B0 + B1)
      AFH(2,5,I) = B1
      AFH(5,6,I) = B7
      AFH(5,2,I) = B6
      AFH(5,5,I) = -B6
      AFH(6,1,I) = B6
      AFH(6,6,I) = -(B7 + B6)
      AFH(6,9,I) = B7
20 CONTINUE
      CPFE(I) = (HFEO(I) - HFEI(I))/(TFEO(I) - TFEI(I))
C      NEXT CALCULATE FEEDWATER TEMPERATURE IN EVAPORATOR SECTION
C      DRAIN COOLER EQUATIONS
C      EVAPORATOR AND CONDENSATE EQUATIONS, NUMBERS 3,4,7,8,9
      B3 = 2.00*WFW(I)/FWEM(I)
      B4 = 2.00*UAEV(I)/(FWEM(I)*CPFE(I))
      ZTFH(3,I) = B3*XFH(4,I) - (B3 + B4)*XFH(3,I) + B4*TFHV(I)
      ZTFH(4,I) = B3*TFEI(I) - (B3 + B4)*XFH(4,I) + B4*TFHV(I)
C      CALCULATE STEAM CHEST CONDITIONS AND NEW DRAIN COOLER FLOW RATE
      TV = TFHV(I)
      PV = PFHV(I)
      HLQ = HLQ(TV)
      HF = HFHV(I) - HLQ
      DTDP(I) = DSTVP(PFHV(I))

```

```

      WFHC(I) = (UAEV(I)*(2.00*TFHV(I) - XFH(3,I) - XFH(4,I)))/HF
      IF(HFHS(I) - HFHV(I)) 74,75,75
75  WFHV(I) = WFHS(I)
      WFHL(I) = CN1
      FHSQ(I) = 1.00
      GO TO 77
74  FHSQ(I) = 1.00 - (HFHV(I) - HFHS(I))/HF
      WFHV(I) = FHSQ(I)*WFHS(I)
      WFHL(I) = (CN2 - FHSQ(I))*WFHS(I)
77  CONTINUE
      IF(I.EQ.1.OR.I.EQ.4) GO TO 83
      HDLI(I) = HDCO(I-1)
      IF(TDCI(I).GT.TV) GO TO 81
      WDCV(I) = CN1
      WDLI(I) = WDC(I-1)
      F3A = WFHS(I)
      WDA = WFHC(I) + WFHL(I)
      GO TO 84
81  CONTINUE
      WDCV(I) = WDC(I-1)*(HDLI(I) - HLO)/HF
      WDLI(I) = WDC(I-1) - WDCV(I)
      F3A = WFHS(I) + WDC(I-1)
      WDA = WFHC(I) + WFHL(I) + WDLI(I)
      GO TO 84
83  WDCV(I) = CN1
      WDLI(I) = CN1
      HDLI(I) = CN1
      F3A = WFHS(I)
      WDA = WFHC(I) + WFHL(I)
84  CONTINUE
      WDF = WDCV(I) + WDLI(I)
      CPI(I) = CPW(XFH(9,I))
      ZTFH(7,I) = WDCV(I) + WFHV(I) - WFHC(I)
      ZTFH(8,I) = WFHL(I) + WDLI(I) + WFHC(I) - WDC(I)
      ZTFH(9,I) = (WFHL(I) + WFHC(I) + WDLI(I))*(TFHV(I) - XFH(9,I))/XFH(8,I)
      IF(LFH(I).GT.0) GO TO 53
      IF(HFHS(I).GT.HFHV(I)) F3A = F3A - WFHS(I)
      AFH(3,3,I) = -B4
      AFH(3,4,I) = B3
      AFH(3,7,I) = B4*B5FH(I)*DTDP(I)
      AFH(4,1,I) = B3
      AFH(4,4,I) = -(B3 + B4)
      AFH(4,7,I) = B4*B5FH(I)*DTDP(I)
      F4 = -UAEV(I)/HF
      F3 = (CPI(I)*F3A + 2.00*UAEV(I) ) * B5FH(I)*DTDP(I)/HF
      AFH(7,7,I) = -F3
      AFH(7,3,I) = -F4
      AFH(7,4,I) = -F4
      AFH(8,3,I) = F4
      AFH(8,4,I) = F4
      AFH(8,7,I) = F3
      AFH(9,3,I) = F4*(TFHV(I) - XFH(9,I))/XFH(8,I)
      AFH(9,4,I) = F4*(TFHV(I) - XFH(9,I))/XFH(8,I)
      AFH(9,7,I) = (F3*(TFHV(I) - XFH(9,I)) + WDA*B5FH(I)*DTDP(I))/XFH(8,I)
      AFH(9,8,I) = -ZTFH(9,I)/XFH(8,I)
      AFH(9,9,I) = -(ZTFH(8,I) + WDC(I))/XFH(8,I)
      CALL EMATS(IFH,I)
      LFH(I) = 1

```

```

NCT(I) = 0
IF(NSS.GT.1) GO TO 53
WRITE(6,101) I
WRITE(6,102)((AFH(L,M,I),M = 1,9),L = 1,9)
WRITE(6,103) I
WRITE(6,104)((HPFH(L,M,I),M = 1,9),L = 1,9)
101 FORMAT(1H0,' A MATRIX FOR FEEDHEATER',I5)
102 FORMAT(1P9D12.5)
103 FORMAT(1H0,' H-P MATRIX FOR FEEDHEATER',I5)
104 FORMAT(1P9D12.5)
53 CONTINUE
DO 54 L = 1,9
SUM1 = CN1
DO 52 M = 1,9
SUM1 = SUM1 + HPFH(L,M,I)*ZTFH(M,I)
52 CONTINUE
TS(L) = SUM1 + XFH(L,I)
IF(NCT(I).EQ.0) GO TO 55
IF(SX(L,I).LE.CN1) GO TO 54
DIF = DABS((TS(L) - SX(L,I))/SX(L,I))
IF(DIF - CNG(L)) 54,57,57
57 LFH(I) = 0
GO TO 54
55 SX(L,I) = TS(L)
54 CONTINUE
NCT(I) = NCT(I) + 1
IF(NCT(I).GE.LCT) LFH(I) = 0
DO 58 L = 1,9
XFH(L,I) = TS(L)
58 CONTINUE
210 CONTINUE
79 CONTINUE
IF(NSS.EQ.0) GO TO 10
IF(IPRT.EQ.0) GO TO 11
10 WRITE(6,131)
WRITE(6,130)((I,(XFH(L,I),L = 1,9)),I = 1,7)
IF(NSS.EQ.0) GO TO 211
IF(IND(5).EQ.0) GO TO 11
WRITE(6,133)
WRITE(6,132)((I,(ZTFH(L,I),L=1,9)),I = 1,7)
131 FORMAT(1H0,' FEEDHEATER STATE VARIABLES'/
1'      I      TFCO      TFCA      TFEO      TFEA      TDCO
2      TOCA      FHVH      FHIM      TFHI')
130 FORMAT(15,9F12.3)
132 FORMAT(15,1P9D12.5)
133 FORMAT(1H0,' FEEDHEATER ZTOT ARRAY')
11 CONTINUE
DO 209 I = 1,NFH
WDC(I) = WFC(I) + WDLI(I) + WFHI(I)
PFHV(I) = B5FH(1)*XFH(7,I) + PDCFH(I)
CALL RESET(I)
209 CONTINUE
211 CONTINUE
RETURN
END

```

```

SUBROUTINE FPIPE(NSS)
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /FEED1/ VFH(8), WFW(8), HFWC(7), WDC(8), HDC(7), HFWE(7),
1 TFHV(8), TFEI(7), TFWO(8), TFWI(8), TDCO(7), FHSQ(7), WFHS(8),
2 HFHS(8), TDCI(8), TFEQ(7), PFHV(8)
  COMMON /FEED2/ IDC(7), NFH
  COMMON/FEED4/ HFWI(7), HFWO(7), HFEI(7), HDCC(7), HDCI(7), HFHV(7)
1,HFEQ(7)
  COMMON /SGEN1/ HSG, TSG, PSG, WSG, WSGI, CSGW, PSGI
  COMMON /BEAT1/ TBHV, TBHQ, CBHQ, WBHC, PBHV, WBHI, WBHO, SQUAL
  COMMON /BEAT2/ TBHC, HBI, BHT(6)
  COMMON /FPUMP/ HFPI, HFPO
  COMMON /TRBL/ WLP(12), TL(12), HL(12), PL(12)
  COMMON /FDAT1/ XFH(9,7), UADC(7), UAEV(7)
  DATA ETAFP / 0.85D0/
  DO 1 I = 1,NFH
    IF(I.EQ.1.OR.I.EQ.4) GO TO 2
    TDCI(I) = TDCO(I-1)
    GO TO 1
  2 TDCI(I) = TFHV(I)
  1 CONTINUE
  IF(NSS.EQ.0) GO TO 10
  P32 = PSAT(TFWO(2)) + 50.0D0
  V32 = VLIQ(TFWO(2))
  HFPI = HFWO(2)
  PSGI = 1.23D0*PSG
  HFWI(1)=HFPI + V32*(PSGI - P32 )*144.0D0/(778.0D0*ETAFP)
  TFWI(1) = TLIQH(HFWI(1))
  HFPO = HFWI(1)
  ENIN2 = (HLIQ(TFWO(3))*WFW(3) + HLIQ(TDCO(3))*WDC(3))/WFW(2)
  TFWI(2) = TLIQH(ENIN2)
  ENIN3 = (HLIQ(TBHC)*WBHQ + HLIQ(TFWO(4))*WFW(4))/WFW(3)
  TFWI(3) = TLIQH(ENIN3)
  TFWI(4) = TFWO(5)
  TFWI(5) = TFWO(6)
  ENIN6 = (HLIQ(TFWO(7))*WFW(7) + HLIQ(TDCO(7))*WDC(7))/WFW(6)
  TFWI(6) = TLIQH(ENIN6)
  TFWI(7) = TL(12)
  HFWI(4) = HFWO(5)
  HFWI(5) = HFWO(6)
  HFWI(7) = HLIQ(TL(12))
  HFWI(2) = ENIN2
  HFWI(3) = ENIN3
  HFWI(6) = ENIN6
10 RETURN
END

```

```

SUBROUTINE RESET(I)
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /FEED1/ VFH(8), WFW(8), HFWC(7), WDC(8), HDC(7), HFWE(7),
1 TFHV(8), TFEI(7), TFWO(8), TFWI(8), TDCO(7), FHSQ(7), WFHS(8),
2 HFHS(8), TDCI(8), TFEQ(7), PFHV(8)
  COMMON/FEED2/ IDC(7), NFH
  COMMON/FEED4/ HFWI(7), HFWO(7), HFEI(7), HDCC(7), HDCI(7), HFHV(7)
1,HFEQ(7)
  COMMON /FEED5/ FHMI(7),TFCA(7),TDCA(7),TFEA(7),FHMV(7), TFHI(7)

```

```

COMMON /SGEN1/ HSG, TSG, PSG, WSG, WSGI, CSGW, PSGI
COMMON /FDAT1/ XFH(9,7), UANC(7), UAEV(7)
TFEI(I) = XFH(1,I)
IF(IDC(I).EQ.0) TFEI(I) = TFWI(I)
TFCA(I) = XFH(2,I)
TFEO(I) = XFH(3,I)
TFEA(I) = XFH(4,I)
TDCO(I) = XFH(5,I)
TDCA(I) = XFH(6,I)
FHMV(I) = XFH(7,I)
FHMI(I) = XFH(8,I)
TFHI(I) = XFH(9,I)
IF(IDC(I).EQ.0) TDCO(I) = XFH(9,I)
TFHV(I) = TSAT(PFHV(I))
HFHV(I) = HVAP(TFHV(I),PFHV(I))
HFEI(I) = HLIQ(TFEI(I))
HFEO(I) = HLIQ(TFEO(I))
HFWI(I) = HLIQ(TFWI(I))
3 IF(HFHS(1).GT.HFHV(I)) GO TO 4
TFWO(I) = TFEU(I)
HFEO(I) = HFEO(I)
GO TO 5
4 HFWO(I) = HFEO(I) + WFHS(I)*(HFHS(I) - HFHV(I))/WFW(I)
TFWO(I) = TLIQH(HFWO(I))
5 CONTINUE
HDCI(I) = HLIQ(TFHI(I))
HDCO(I) = HLIQ(TDCO(I))
RETURN
END

```

```

SUBROUTINE REACTR(I)
IMPLICIT REAL * 8(A-H,O-Z)
COMMON/GEN1/TIME,DT,TNXT
COMMON /GEN4/ IPRT
COMMON/MAT2/RAB(7,7),HPR(7,7), DTR
COMMON /REAC2/ WRP, TFA, TCA, TCRI, PWR, FCM,CCM,CLM,CUM,TCRO
COMMON /REAC3/ RHOP,PHO, AP, BP, AMPL, FREQ, TPERT
COMMON /STEAM1/ TSGPO, TSGPI
COMMON /RFAC1/ XR(7), ZRT(7), RFD(30)
COMMON /FDAT4/ UAFC, UAQI
COMMON /SGEN1/ HSG, TSG, PSG, WSG, WSGI, CSGW, PSGI
COMMON /SGEN3/ XSG(10), ZTSG(10), ASG(10,10), HPSG(10,10)
COMMON /TRBH/ WHP(12), TH(12), HH(12), PH(12)
COMMON /EVAP7/ NK, NNK, NTMSP, NN, LCT, NSTGS, ISECT, NARROW, NEWW
1, HRT, IND(16), IERROR, JUP, ISM(50), ICT(50)
IF (I .GE. 1) GO TO 100

```

110
120

C INITIALIZE CALCULATION

300

```

AP6 = 1.00+6
AM6 = 1.00-6
IWH = 1
RAMP = 0.000
AP = 0.000
BP = 0.000
BPS = 0.000
PWS = 0.000
EXH = DEXP(-DT/0.500)

```

```

QME = 1.00 - EXM
EXA = DEXP(-0.500*DT)
PWA = 0.000
ISHM = 0
AMPL = 0.000
FREQ = 0.000
DCAY = RFD(1)
BETA = RFD(2)
PNLT = RFD(3)
ETA = RFD(4)
CPF = RFD(5)
PCF = RFD(6)
TCC = RFD(7)
FCM = RFD(8)
CCM = RFD(9)
CLM = RFD(10)
CUM = RFD(11)
TFO = RFD(14)
TCO = RFD(15)
IF(NEWW.EQ.0) GO TO 51
CDNP = RFD(13)
TFA = RFD(14)
TCA = RFD(15)
TCCI = RFD(16)
C  CALCULATE STEADY STATE HEAT BALANCE
DHC = PWR/WRP
HCA = HLIQ(TCA)
DO 10 N = 1,10
HCI = HCA - DHC/2.0
HCO = HCI + DHC
DHC = HCO - HCI
PWT = DHC*WRP
PERROR = DABS(PWT - PWR)
IF((PERROR).LE.1.0D-3) GO TO 11
WRITE(6,102) PWT, DHC, HCI, HCO
DHC = DHC*PWR/PWT
10 CONTINUE
11 CONTINUE
TCA = TLIQH(HCA)
TCCI = TLIQH(HCI)
TCCO = TLIQH(HCO)
CPC = (HCA - HCI)/(TCA - TCCI)
CPO = (HCO - HCA)/(TCCO - TCA)
HRI = HLIQ(TCRI)
DHL = HCI - HRI
CPL = DHL/(TCCI - TCRI)
C  NOW FIND REACTOR VESSEL OUTLET TEMPERATURE.TCRO
HRO = HCO - DHL
TCRO = TLIQH(HRO)
CPU = (HRO - HCO)/(TCRO - TCCO)
UAFC = (1.00 - ETA)*PWR/(TFA - TCA)
C  UAQI DRIVING FORCE IS ASSUMED TO BE DIFFERENCE BETWEEN CORE
C  OUTLET COOLANT TEMP AND REACTOR INLET TEMP
UAQI = WRP*DHL/(TCCO - TCRI)
CDNP = BETA*PWR/(DCAY*PNLT)
RHOP = 0.000
CPO = CPW(TCRO)
CPCI = CPW(TCCI)

```



```

      CPCA = CPM(TCA)
      CPCO = CPM(TCCO)
      WRITE(6,107) CPCI,CPCA, CPCO, CPRQ,CPL, CPC, CPD, CPU
107  FORMAT(1P8D12.5)
      51 CONTINUE
      DO 15 K = 1,7
      DO 16 J = 1,7
      RAB(K,J) = 0.000
      16 CONTINUE
      15 CONTINUE
      RAB(1,2) = DCAY
      RAB(2,1) = BETA/PNL T
      RAB(2,2) = -DCAY
      CPM = FCN*CPF
      RAB(3,1) = (1.000-ETA)/(CPM)
      RAB(3,3) = -UAFC/(CPM)
      RAB(3,4) = -RAB(3,3)
      DTR = DT/FLQAT(NRT)
      C  NRT IS REACTOR SUB INTERVAL
      IF(NEWH.EQ.0) GO TO 50
      XR(1) = PHR
      XR(2) = CONP
      XR(3) = TFA
      XR(4) = TCA
      XR(5) = TCCO
      XR(6) = TCCI
      XR(7) = TCRO
      TFO = TFA
      TCO = TCA
      PWO = PH(2)
      PWDD = PH(2)
      DTO = PHR
      RFD(25) = PWO
      RFD(26) = PWDD
      RFD(28) = DTO
      TSRA = (TCRO + TCRI)*0.5D0
      TRIO = TSRA
      RFD(27) = TRIO
      WRITE(6,108) PWDD, TRIO, DTO, PWO
108  FORMAT(1H0,' PWDD =',F10.3,' TRIO =',F8.3,' DTO =',F12.3,
      1' PWO =',F10.0)
      RFD(14) = TFO
      RFD(15) = ILU
      SCODL = 0.5D0*CCM*(HCA + HCO) + CLM*HCI + CUM*HRO
      GO TO 400
      100 CONTINUE
      IF(IMD(16).EQ.0) GO TO 20
      C  REACTOR CONTROL SYSTEM THRU STATEMENT 20
      TSRC = (TCRO + TCRI)*0.5D0
      C  3 SEC LAG IN THERMAL COUPLES
      TSRA = TSRC - (TSRC - TSRA)*DEXP(-.33D0*DT)
      RFD(29) = TSRA
      BPS = 0.0D0
      C  5 SEC. TIME CONSTANT IN POWER DEMAND SIGNAL
      PWDD = PH(2) - (PH(2) - PWDD)*DEXP(-.2D0*DT)
      RFD(26) = PWDD
      TSRO = TRIO + (PWDD - PWO)*7.5D-2
      PWC =      DTO*(1.00 - (PWO - PWDD)/6.5D2)

```

390
430
440

520

```

PWE = PWC - PWR
IF(1.EQ.1) PEO = 0.000
PEO = EXM*PEO + OME*PWE
DPP = PWE - PEO
C 2 SEC. TIME CONSTANT FOR POWER FEEDBACK SIGNAL
PHA = DPP - (DPP - PHA)*EXA
PWS = PWE
DTOT = (TSRA - TSRS)/DT
ERSGN = TSRO - (10.DO*DTOT + TSRA) + 5.0D-3*PHA
IF(DABS(ERSGN).LT.0.5D0) GO TO 20
IF(DABS(ERSGN).LT.1.D0) ERSNG = DSIGN(1.0D0,ERSGN)
IF(DABS(ERSGN).GT.15.D0) ERSNG = DSIGN(15.0D0,ERSGN)
BPS = ERSNG*0.34667D-5
20 CONTINUE
HRI = HLIQ(TCRI)
CPL = (HCI - HRI)/(TCCI - TCRI)
RAB(4,5) = 2.DO*WRP*CPC/(CCM*CPCA)
RAB(5,5) = -2.DO*WRP*CPO/(CCM*CPCO)
RAB(6,6) = -WRP*CPL/(CLM*CPCI)
RAB(4,1) = ETA/(CPCA*CCM)
RAB(4,3) = UAFC/(CPCA*CCM)
RAB(6,5) = UAOI/(CLM*CPCI)
RAB(7,5) = (WRP*CPU - UAOI)/(CUM*CPRD)
RAB(4,4) = -(RAB(4,3) + RAB(4,5))
RAB(5,1) = ETA/(CPCO*CCM)
RAB(5,3) = UAFC/(CPCO*CCM)
RAB(5,4) = -RAB(5,3) - RAB(5,5)
DO 110 NL = 1,NRT
RAMP = RAMP + (BP + BPS)*DTR
RFD(20) = AP
RFD(30) = RAMP
RHOP = AP + RAMP
IF(AMPL.EQ.0.0D0) GO TO 250
DTPERT = TIME - TPRT + DTR*FLUAT(NL)
RHOP = RHOP + AMPL*DSIN(6.283184D0*FREQ*DTPERT)
250 RHO = RHOP*PCF*(XR(3)-TFO)+TCC*(XR(4)-TCO)
RBL = (RHO-BETA)/PNLT
RAB(1,1) = RBL
ZRT(1) = RBL*XR(1) + DCAY*XR(2)
RAB(1,3) = XR(1)*PCF/PNLT
RAB(1,4) = XR(1)*TCC/PNLT
C COMPUTE REST OF ZRT(L)
ZRT(2) = BETA*XR(1)/PNLT - DCAY*XR(2)
ZRT(3) = (1.0D0-ETA)*XR(1)/CPFM - UAFC*(XR(3)-XR(4))/CPFM
ZRT(4) = (ETA*XR(1) + UAFC*(XR(3) - XR(4)) - 2.DO*WRP*CPC*
1 (XR(4) - XR(6)))/(CCM*CPCA)
ZRT(5) = (ETA*XR(1) + UAFC*(XR(3) - XR(4)) - 2.DO*WRP*CPO*
1 (XR(5) - XR(4)))/(CCM*CPCO)
ZRT(6) = (WRP*CPL*(TCRI-XR(6))+ UAOI*(XR(5) - TCRI))/(CLM*CPCI)
ZRT(7) = (WRP*CPU*(XR(5)-XR(7))-UAOI*(XR(5) - TCRI))/(CUM*CPRD)
C CALL AND COMPUTE HP MATRIX
RAB(1,3) = AM6*RAB(1,3)
RAB(1,4) = AM6*RAB(1,4)
RAB(3,1) = AP6*RAB(3,1)
RAB(4,1) = AP6*RAB(4,1)
RAB(5,1) = AP6*RAB(5,1)
ZRT(1) = AM6*ZRT(1)
ZRT(2) = AM6*ZRT(2)

```

1120
1140
1150
1155
1160
1170
1180

1250

```

XR(1) = AM6*XR(1)
XR(2) = AM6*XR(2)
CALL EMATS(3,0)
DO 300 N = 1,7
SUM2 = 0.000
DO 290 L = 1,7
290 SUM2 = SUM2 + HPR(N,L)*ZRT(L)
300 XR(N) = XR(N) + SUM2
RAB(1,3) = AP6*RAB(1,3)
RAB(1,4) = AP6*RAB(1,4)
RAB(3,1) = AM6*RAB(3,1)
RAB(4,1) = AM6*RAB(4,1)
RAB(5,1) = AM6*RAB(5,1)
ZRT(1) = AP6*ZRT(1)
ZRT(2) = AP6*ZRT(2)
XR(2) = AP6*XR(2)
XR(1) = AP6*XR(1)
110 CONTINUE
IF(IPRT.EQ.0) GO TO 13
IF(IND(16).EQ.0) GO TO 17
WRITE(6,109) TSRA,TSRO,TSRC,BPS
109 FORMAT(1H0,'TSRA = ',F8.3,' TSRO = ',F8.3,' TSRC = ',F8.3,' BPS = ',
1,D12.5)
WRITE(6,102) PWA,DTDT,ERSGN,PWC,DPP, PWE
17 CONTINUE
WRITE(6,104) RHOP, RHO, TCRI
WRITE(6,105)
WRITE(6,102)((XR(N),ZRT(N)), N = 1,7)
IF(I.LE.1) GO TO 14
IF(IND(5).EQ.0) GO TO 13
14 WRITE(6,103)((RAB(N,J),J = 1,7),N = 1,7)
13 CONTINUE
HPRD = DT*CDNP*DCAY*PNLT/(BETA - RHO)
DFLOW = DT*WRP*(HRO - HRI)
HFSTR = CPM*(XR(3) - TFA)
GO TO 52
50 CONTINUE
PWO = RFD(25)
PWDD = RFD(26)
TRIO = RFD(27)
OTO = RFD(28)
TSRA = RFD(29)
RAMP = RFD(30)
AP = RFD(20)
52 CONTINUE
PWR = XR(1)
CDNP = XR(2)
TFA = XR(3)
TCA = XR(4)
TCCO = XR(5)
TCCI = XR(6)
TCRO = XR(7)
HCA = HLIQ(TCA)
HCO = HLIQ(TCCO)
HCI = HLIQ(TCCI)
HRO = HLIQ(TCRO)
HCOOL = 0.500*CCM*(HCA + HCO) + CLM*HCI + CUM*HRO
DHTC = HCOOL - SCOO

```

```

SCOOOL = HCOOL
CPC = (HCA - HCI)/(TCA - TCCI)
CPO = (HCO - HCA)/(TCCO - TCA)
CPU = (HRO - HCO)/(TCRO - TCCO)
CPCA = CPW(TCA)
CPCO = CPW(TCCO)
CPRO = CPW(TCRO)
CPCI = CPW(TCCI)
IF(IPRT.EQ.0) GO TO 12
IF(IND(5).EQ.0) GO TO 12
WRITE(6,103) HFSTR, HCOOL, HCA, HCO, HCI, HRO, HRI, CPC,CPL,CPU
1 ,CPCA, CPCI, PCO, CPRO
102 FORMAT(1P2D13.5)
103 FORMAT(1P7D12.5)
104 FORMAT(1H0,'EXTERNAL REACTIVITY = ',1PD12.5//
1' TOTAL REACTIVITY = '1PD12.5//
2' REACTOR INLET TEMPERATURE = ',0PF10.3,'DEG. F')
105 FORMAT(1H0,' REACTOR STATE VARIABLES'//
1' NO.      XR      ZRT')
12 CONTINUE
400 CONTINUE
TSRS = TSRA
RETURN
END

```

1490

```

SUBROUTINE SGEN(NSS)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /STEAM1/ TSGPO, TSGPI
COMMON /SGEN1/ HSG, TSG, PSG, WSG, WSGI, LSGW, PSGI
COMMON /GEN4/ IPRT
COMMON /SGEN2/ HSGI, TSGI, HSGPI, HSGPO
COMMON /SGEN3/ XSG(10), ZTSG(10), ASG(10,10), HPSG(10,10)
COMMON /REAC1/ XR(7), ZRT(7), RFD(30)
COMMON /REAC2/ WRP, TFA, TCA, TCRI, PWR, FCM,CCM,CLM,CUM,TCRO
COMMON /EVAP7/ NK, NMK, NTMSP, NN, LCT, NSTGS, ISECT, NARROW, NEWW
1, NRT, IND(16), IERROR, IUP, ISM(50), ICT(50)
COMMON /FDAT3/ B5SG, PDCSG, FCON, UAB, UANB
IF(NSS.GE.1) GO TO 100
DO 20 I = 1, 10
DO 21 J = 1, 10
ASG(I,J) = 0.000
21 CONTINUE
ZTSG(I) = 0.000
20 CONTINUE
WRP = RFD(18)
TSGI = TL10H(HSGI)
TCRI = RFD(19)
TSGPO = TCRI
TSGPI = RFD(17)
HSGPI = HL10(TSGPI)
C ASSUMING STEAM GENERATOR SATURATED
HSG = HVAP(TSG,PSG)
IF(NEWW.EQ.0) GO TO 50
HSGPO = HL10(TSGPO)
HSL = HL10(TSG)
PWR = (HSG - HSGI)*WSG

```

```

HSGPI = PWR/WRP + HSGPO
TSGPI = TLIOH(HSGPI)
TCRN = TSGPI
WS = WSGI
SOSG = RFD(21)
WNB = WS/SOSG
WREC = WNB - WS
HD = (WS*HSGI + WREC*HSL)/WNB
TD = TLIOH(HD)
TNBA = (TSG + TD)*0.500
TNBQ = TSG
FCQN = WNB/(TNBQ - TD)
XSG(7) = TNBQ
XSG(8) = TNBA
XSG(9) = TD
C HOLDUP TIMES
C STEAM = 6.4 SEC
C DOWNCOMER = 7.0 SECS.
C NON BOILING REGION = 14.0 SECS.
C TOTAL PRIMARY SIDE HOLDUP = 4.1 SECS.
XSTM = 6.400*WSG
XSG(10) = XSTM
XMDWN = 7.0000*WNB
XMNB = 14.000*WNB
XMP1 = 0.800*WRP
XMP3 = 0.800*WRP
XMP2 = 2.500*WRP
RFD(22) = XMP1
RFD(23) = XMDWN
RFD(24) = XMNB
TO3 = TSGPO
XSG(5) = TO3
CPNB = (HSL - HD)/(TSG - TD)
CPP2 = (HSGPI - HSGPO)/(TSGPI - TSGPO)
CPP1 = CPW(TSGPI)
CPP3 = CPW(TSGPO)
HFSG = HSG - HSL
TA1 = TSGPI
TA3 = TSGPO
HNBQ = HSL
HBL = WS*HFSG
DO 10 I = 1, 10
DTR = HBL/(WRP*CPP3)
DTNB = TSGPI - TSGPO - DTR
TOP = (TA1 - TNBA)/CPP1
BOT = TOP + (TA3 - TNBA)/CPP3
TO1 = TSGPI - DTNB*TOP/BOT
TO2 = TO1 - DTR
TA1 = (TSGPI + TO1)*0.500
TA3 = (TO2 + TO3)*0.500
HO1 = HLIO(TO1)
HO2 = HLIO(TO2)
HH2 = (HO1 - HO2)*WRP
WRITE(6,105) CPP1, CPP2, CPP3, HH2, HSL, HFSG
CPP2 = (HO1 - HO2)/(TO1 - TO2)
CPP1 = (HSGPI - HO1)/(TSGPI - TO1)
CPP3 = (HO2 - HSGPO)/(TO2 - TO3)
IF(DABS(HH2 - HBL).LE.0.1D-2) GO TO 12

```

```

10 CONTINUE
  WRITE(6,101) HQ1, HQ2, HH2, HBL, DTB, DTNB
101 FORMAT(1H1, ' MORE THAN 10 S. GEN. TEMP. ITER.', 1P6D12.5)
12 UANB = (HSGPI - HQ1)*WRP/(TA1 - TNBA)
  TA2 = (TQ1 + TQ2)*0.5D0
  UAB = WRP*(HQ1 - HQ2)/(TA2 - TSG)
  CPS = CPW(TSG)
  XSG(6) = TA3
  XSG(1) = TQ1
  XSG(2) = TA1
  XSG(3) = TQ2
  XSG(4) = TA2
  SVOL = VVAP(TSG,PSG)
  PD = 1.D0/SVOL
  B5SG = DVPSD(PD)/(SVOL*XMSTM)
  PDCSG = PSG - B5SG*XMSTM
  WRITE(6,105) UAB, UANB, TSG, WRP, WS, HFSG
  GO TO 200
100 CONTINUE
  TSGPI = TCRO
  HSGPI = HLIQ(TSGPI)
  WNB = FCN*(TNBO - TD)
  PSG = PDCSG + B5SG*XMSTM
  TSG = TSAT(PSG)
  HLS = HLIQ(TSG)
  HSG = HVAP(TSG,PSG)
  HFSG = HSG - HLS
  WS = (UAB*(TA2 - TSG) + WNB*(HNBO - HLS))/HFSG
  WSGI = WS
  CPD = CPW(TD)
  CPSI = (HLS - HSGI)/(TSG - TSGI)
  CPSD = (HLS - HD)/(TSG - TD)
  B1 = 2.D0*WRP/XMP1
  B2 = UANB/(XMP1*CPP1)
  B3 = 2.D0*WRP/XMP2
  B4 = UAB/(XMP2*CPP2)
  B7 = 2.D0*WRP/XMP3
  B8 = UANB/(XMP3*CPP3)
  B9 = 2.D0*WNB/XMNB
  B10 = UANB/(XMNB*CPNB)
  B11 = WNB*CPD/(XMDWN*CPD)
  B12 = WSGI*CPSI/(XMDWN*CPD)
  B13 = UAB/HFSG
  B14 = WNB*CPD/HFSG
  ZTSG(1) = B1*(TA1 - TQ1) - B2*(TA1 - TNBA)
  ZTSG(2) = B1*(TSGPI - TA1) - B2*(TA1 - TNBA)
  ZTSG(3) = B3*(TA2 - TQ2) - B4*(TA2 - TSG)
  ZTSG(4) = B3*(TQ1 - TA2) - B4*(TA2 - TSG)
  ZTSG(5) = B7*(TA3 - TQ3) - B8*(TA3 - TNBA)
  ZTSG(6) = B7*(TQ2 - TA3) - B8*(TA3 - TNBA)
  ZTSG(7) = B9*(TNBA - TNBO) + B10*(TA1 + TA3 - 2.D0*TNBA)
  ZTSG(8) = B9*(TD - TNBA) + B10*(TA1 + TA3 - 2.D0*TNBA)
  ZTSG(9) = B11*(TSG - TD) - B12*(TSG - TSGI)
  ZTSG(10) = WS - WSG
  DTDV = DSTVP(PSG)
  ASG(1,1) = -B1
  ASG(1,2) = B1 - B2
  ASG(1,8) = B2

```

```

ASG(2,2) = -(B1 + B2)
ASG(2,8) = B2
ASG(3,3) = -B3
ASG(3,4) = B3 - B4
ASG(3,10) = B4*B5SG*DTDP
ASG(4,1) = B3
ASG(4,4) = -(B3 + B4)
ASG(4,10) = ASG(3,10)
ASG(5,6) = B7 - B8
ASG(5,8) = B8
ASG(6,3) = B7
ASG(6,6) = -(B7 + B8)
ASG(6,8) = B8
ASG(7,2) = B10
ASG(7,6) = B10
ASG(7,7) = -B9
ASG(7,8) = B9 - 2.00*B10
ASG(8,2) = B10
ASG(8,6) = B10
ASG(8,8) = -B9 - 2.00*B10
ASG(8,9) = B9
ASG(9,9) = B11
ASG(9,10) = (B11 - B12)*B5SG*DTDP
ASG(10,4) = B13
ASG(10,7) = B14
ASG(10,10) = -(B13 + B14)*B5SG*DTDP
IF(NSS.GT.1) GO TO 14
WRITE(6,102)
WRITE(6,103)((N,XSG(N),ZTSG(N)),N = 1, 10)
WRITE(6,104)((ASG(N,J), J = 1, 10), N = 1, 10)
14 CONTINUE
CALL EMATS(4,0)
DO 300 N = 1, 10
  SUM2 = 0.000
  DO 290 L = 1, 10
290 SUM2 = SUM2 + HPSG(N,L)*ZTSG(L)
300 XSG(N) = XSG(N) + SUM2
GO TO 51
50 CONTINUE
XMP1 = RFD(22)
XMP3 = RFD(23)
XMP2 = 3.00*RFD(22)
XMDWN = RFD(23)
XMNB = RFD(24)
51 CONTINUE
T01 = XSG(1)
TA1 = XSG(2)
T02 = XSG(3)
TA2 = XSG(4)
T03 = XSG(5)
TA3 = XSG(6)
TNBO = XSG(7)
TNBA = XSG(8)
TD = XSG(9)
XNSTM = XSG(10)
H01 = HLIQ(T01)
H02 = HLIQ(T02)
HSGPO = HLIQ(T03)

```

```

TCRI = T03
CPP1 = (HSGPI - H01)/(TSGPI - T01)
CPP2 = (H01 - H02)/(T01 - T02)
CPP3 = (H02 - HSGP0)/(T02 - T03)
HNB0 = HLIQ(TNB0)
HD = HLIQ(TD)
CPNB = (HNB0 - HD)/(TNB0 - TD)
CPS = CPW(TSG)
IF(IPRT.EQ.0) GO TO 200
WRITE(6,102)
WRITE(6,103)((N,XSG(N),ZTSG(N)),N = 1, 10)
WRITE(6,106) WS, WNB, WSGI, PSG, HSG, WSG
106 FORMAT(1H0,' WS=',OPF9.3,' WNB =',F9.3,' WSGI =',F9.3,
1' PSG =',F9.3,' HSG =',F9.3,' WSG =',F9.3)
102 FORMAT(1H0,' STEAM GENERATOR STATE VARIABLES'//
1' NO.      XSG      ZTSG')
103 FORMAT(I4,1P2D12.5)
104 FORMAT(1P10D11.4)
105 FORMAT(1P6D12.5)
200 CONTINUE
RETURN
END

```

```

SUBROUTINE EVAP(KK)
C MAIN CALLING PROGRAM FOR MSF SIDE OF COUPLED PLANT
IMPLICIT REAL*8(A-H,O-Z)
COMMON /GEN1/ TIME, DT, TNXT
COMMON /MAT1/ A(8,8,50), HP(8,8,50), AB(6,6), HHP(6,6)
COMMON /EVAP1/ X(8,50), XIC(8,50), ZT(8,50), Y(8,50), WMAT(30,50)
COMMON /EVAP2/ PDC(50), TMASS(50), AO(50), AI(50),
1CTUBMS(50), B5(50), FPF(50), FB1(50), FB2(50), FB3(50), FTB(50),
2FP1(50), FP2(50), FWP(50), FOUL(50), FBF(50), FUDHI(50), FUDHO(50)
COMMON /EVAP3/ WTBO(50), WTBI(50), WTBFI(50), WPI(50), WC(50),
1WPF(50), WPO(50), CPB(50), CPTB(50), CPP(50), CCP(50), CONC(50),
2CNCI(50), SMTT(50), SMTB(50)
COMMON /EVAP4/ TTI(50), TTBI(50), TPI(50), BL(50), PL(50), BLI(50)
1, PLI(50), DMTB(50), DMP(50)
COMMON /EVAP5/ WSEA, WTUBE, FRSEA, FRTB, SMPMS, TPSUMP, TSEA,
1TMAX, DEAIR, SSUMP, CSMP, CPSMP, CSEA
COMMON /EVAP6/ SD(24), FD(54), CNG(8), TI(4), TITLE(20), EXPL(20)
1, ST, WMIN
COMMON /EVAP7/ NK, NNK, NTMSP, NE, LCT, NSTGS, ISECT, NARRUN, NEWW
1, ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)
COMMON /EVAP8/ VENT, IATM
COMMON /BEAT1/ TBHV, TBHO, CBHO, WBHC, PBHV, WBHI, WBHO, SQUAL
IF(KK.GE.1) GO TO 100
IUP = 0
IF(NEHW.EQ.1) GO TO 5
VENT = FD(53)
GO TO 8
5 VENT = 1.0D0
8 CONTINUE
TMAX = NTMSP*DT + TI(4)
IERROR = 0
C CALCULATE ITEMS BASED ON FIXED DATA
A1 = 1.0D0/(3.6D3*FD(14))

```



```

CSEA = FD(30)
CBHO = FD(29)
CSMP = FD(28)
WSEA = FD(22)*A1
WTUBE = FD(21)*A1
SMPHS = WTUBE*FD(45)
SMS P = SMPHS*CSMP
TI(3) = FD(37)/FD(25)
FRSEA = (FD(21) - FD(36))/FD(22)
FRTB = FD(36)*A1
IN1 = IND(1)
IN2 = IND(2)
IN3 = IND(3)
IN4 = IND(4)
IN5 = IND(5)
IN7 = IND(7)
NSTGS = NK+NNK
ISECT=2
DO 1 I= 1,NSTGS
DO 27 N = 1,8
DO 28 K = 1,8
A(K,N,I) = 0.0
28 CONTINUE
27 CONTINUE
DO 26 N = 1,30
MMAT(N,I) = 0.0
26 CONTINUE
IF(I-NK) 6,6,7
7 ISECT=3
C READ IN DATA FOR STAGE I
6 CALL READ(I,2)
CALL ICON(I)
CALL SDATA(I)
IF(IEORR.GT.0) GO TO 40
C SDATA CALCULATES THE FIXED COEFFICIENTS REQUIRED FOR CALCULATING
C THE CONSTANTS, ETC. IN AMATS, AND ZTOT.
ISM(I) = 1
DO 13 N = 1,NE
X(N,I) = XIC(N,I)
Y(N,I) = XIC(N,I)
13 CONTINUE
SMTB(I) = Y(6,I)*CUNC(I)
SMTT(I) = TMASS(I)*CNCT(I)
IF(1.EQ.NK) TPSUMP = SD(8)
1 CONTINUE
C CALCULATE ITEMS BASED ON LAST STAGE SD
TSEA = SD(8)
CALL OUT(I,3)
IPT = 1
TDUMP = X(5,NSTGS)
FLOTB = FRIB
FLOSEA = FRSEA*WSEA
SSUMP = TPSUMP
SPFUD = FLOTB*CPTB(NSTGS)*(TDUMP - TPSUMP)
1 + FLOSEA*CPB(NK+1)*(X(1,NK+1) - TPSUMP)
TBML = X(6,NSTGS)
PML = X(7,NSTGS)
GO TO 200

```

MAIN

MAIN

MAIN

MAIN

MAIN

```

100 CONTINUE
   IF(KK.EQ.NTMSP) IN5 = 1
   DO 3 I = 1,NSTGS
      IF(I.NE.NK) GO TO 10
C     NEXT CALCULATE TEMPERATURE OF BRIME IN RECYCLE SUMP AND THE TUBE
C     BRIME FLOW RATE FROM THE SUMP ASSUMING CONSTANT SUMP BRIME MASS
      FLOTB = FRTB
      FLOSEA = FRSEA*WSEA
      WTUBE = FLOTB + FLOSEA
      SSUMP = TPSUMP
      CPSMP = CP(TPSUMP,CSP)
      TPSUMP = TPSUMP + DT*(FLOTB*CPTB(NSTGS)*(TDUMP - TPSUMP) + FLOSEA
1*CPB(NK+1)*(X(1,NK+1) - TPSUMP) - SPFUD)/(SMPMS*CPSMP)
      SMSP = SMSP + DT*(FLOTB*CONC(NSTGS)+FLOSEA*CNCT(NK+1)-WTUBE*CSMP)
10 CONTINUE
   CALL ZTOT(I)
C     ZTOT CALCULATES THE TOTAL FORCING FUNCTIONS FOR EACH EQUATION
C     IN STAGE I
      IF(IERROR.EQ.1) GO TO 32
      IF(ISM(I).EQ.0) GO TO 4
      CALL AMATS(I)
      IF(IERROR.EQ.1) GO TO 32
C     AMATS CALCULATES A NEW A MATRIX FOR STAGE WHENEVER ISM = 1
      CALL EMATS(1,I)
C     EMATS CALCULATES A NEW C AND HP MATRIX FROM THE A MATRIX
      IF(IN4) 35,33,35
35 IF(J.EQ.1) GO TO 32
      IF(IN1) 32,33,32
32 CALL OUT(I,4)
      IF(IERROR.EQ.1) GO TO 39
33 ICT(I) = 1
      ISM(I) = 0
4 CONTINUE
      CALL UPSI(I)
C     UPSI CALCULATES THE VALUES OF THE STATE VARIABLES AT THE END
C     OF THE TIME INTERVAL.
      ICT(I) = ICT(I) + 1
      IF(LCT.GT.ICT(I))GO TO 11
      ISM(I) = 1
11 CONTINUE
      DMTB(I) = Y(6,I) - X(6,I)
      DMP(I) = Y(7,I) - X(7,I)
      WMAT(14,I) = DMTB(I)
      WMAT(15,I) = DMP(I)
      WMAT(16,I) = Y(4,I) - X(4,I)
      WMAT(17,I) = CONC(I)
      WMAT(18,I) = CNCT(I)
3 CONTINUE
C     END OF STAGE LOOP
      IF(IN3.EQ.0) GO TO 34
      IF(IPT.LT.IN3) GO TO 36
39 CALL OUT(I,5)
      IF(IERROR.EQ.1) GO TO 40
      IPT = 0
36 CONTINUE
      IPT = IPT + 1
34 DO 31 I = 1,NSTGS
      DO 30 N = 1,NE

```

```

      X(N,I) = Y(N,I)
30  CONTINUE
      BL(I) = BLI(I)*X(6,I)/XIC(6,I)
      PL(I) = PLI(I)*X(7,I)/XIC(7,I)
      CONC(I) = SMTB(I)/Y(6,I)
      CNGT(I) = SMTT(I)/TMASS(I)
31  CONTINUE
      CSMP = SMSPP/SMPMS
      X(6,NSTGS) = TBHL
      X(7,NSTGS) = PHL
      TDUMP = X(5,NSTGS)
      GO TO 200
40  CONTINUE
      STOP 2
200 CONTINUE
      RETURN
      END

```

ACON

MAIN

```

SUBROUTINE OUT(M,KK)
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /GEN1/ TIME, DT, TNXT
  COMMON /MAT1/ A(8,8,50), HP(8,8,50), AR(6,6), HHP(6,6)
  COMMON /EVAP1/ X(8,50), XIC(8,50), ZT(8,50), Y(8,50), WMAT(30,50)
  COMMON /EVAP2/ PDC(50), TMASS(50), AQ(50), AI(50),
  1CTUBMS(50), B5(50), FPF(50), FB1(50), FB2(50), FB3(50), FTB(50),
  2FPI(50), FP2(50), FWP(50), FOUL(50), FBF(50), FUDHI(50), FUDHQ(50)
  COMMON /EVAP3/ WTBU(50), WTBI(50), WTBFI(50), WPI(50), WC(50),
  1WPF(50), WPQ(50), CPR(50), CPTB(50), CPP(50), CCP(50), CONC(50),
  2CNGT(50), SMTT(50), SMTB(50)
  COMMON /EVAP4/ TTI(50), TTBI(50), TPI(50), BL(50), PL(50), BLI(50)
  1, PLI(50), DMTB(50), DMP(50)
  COMMON /EVAP5/ WSEA, WTUBE, FRSEA, FRTB, SMPMS, TPSUMP, TSEA,
  1 TMAX, DEAIR, SSUMP, CSMP, CPSMP, CSEA
  COMMON /EVAP6/ SD(24), FD(54), CNG(8), TI(4), TITLE(20), EXPL(20)
  1, ST, WMIN
  COMMON /EVAP7/ NK, NMK, NTMSP, NE, LCT, NSTGS, ISFCT, NARROW, NERR
  1, ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)
  COMMON /BFAT1/ TBHV, TDIQ, CBHQ, WBHG, PDIIV, MUHI, WBHO, SQUAL
  COMMON /MPL0T/ IPP, IP1, IP2, IP3, IP4, IP5, IP6, IP7
  DIMENSION Z(8,50)
  GO TO (1,2,3,4,5),KK
1  CONTINUE
2  CONTINUE
3  CONTINUE
  IN4 = IND(4)
  IN5 = IND(5)
  IN8 = IND(8)
  WRITE(6,105)
  WRITE(6,102)
  WRITE(6,103) (I,(XIC(J,I),J = 1,8), I = 1,NSTGS)
  WRITE(6,115)
  WRITE(6,116) WSEA,WTUBE, SMPMS,TI(3), FRSEA,FRTB, TBHV
  IF(IN4.EQ.0) GO TO 30
  WRITE(6,117)
  WRITE(6,114)((I,FB1(I),FB2(I),FB3(I),FTB(I),FBF(I),FPI(I),FP2(I),
  1FWP(I),FPF(I)),I = 1,NSTGS)

```

```

WRITE(6,119)
DO 11 I = 1,NSTGS
WRITE(6,114) I,TMASS(I),CTUBMS(I),CPTB(I),CPR(I),B5(I),PDC(I),FUD
1HI(I),FUDHO(I),FOUL(I)
11 CONTINUE
WRITE(6,113)
WRITE(6,114)(I,(WMAT(N,I),N = 1,9), I = 1,NSTGS)
GO TO 30
4 CONTINUE
WRITE(6,109) M
WRITE(6,108)((A(K,N,M),N=1,8), K=1,8)
WRITE(6,125) M
WRITE(6,108)((HP(N,L,M),L=1,8),N = 1,8)
GO TO 10
5 CONTINUE
IN5 = IND(5)
WRITE(6,101)
DO 34 I = 1,NSTGS
DO 35 N = 1, NE
Z(N,I) = Y(N,I)
35 CONTINUE
Z(6,I) = BL(I)
Z(7,I) = PL(I)
34 CONTINUE
WRITE(6,136)
WRITE(6,137)(I,(Z(N,I),N=1,NE),I=1,NSTGS)
WRITE(6,120) TBHO
WRITE(6,121) WPO(NSTGS)
121 FORMAT(1H,' WATER PRODUCTION RATE = ',F9.1,' POUNDS/SECOND')
IF(IN5.EQ.0) GO TO 30
WRITE(6,112)
WRITE(6,111)(I,(ZT(N,I),N = 1,8),I=1,NSTGS)
WRITE(6,113)
WRITE(6,114)(I,(WMAT(N,I),N = 1,9), I = 1,NSTGS)
WRITE(6,122)
WRITE(6,114)(I,(WMAT(N,I),N=10,18),I=1,NSTGS)
SUM1 = 0.000
SUM2 = 0.000
SUM3 = 0.000
SUM5 = 0.000
SUM6 = 0.000
SUM7 = 0.000
DO 20 I = 1,NSTGS
SUM1 = SUM1 + Y(4,I)
SUM2 = SUM2 + Y(6,I)
SUM3 = SUM3 + Y(7,I)
SUM5 = WMAT(14,I) + SUM5
SUM6 = WMAT(15,I) + SUM6
SUM7 = WMAT(16,I) + SUM7
20 CONTINUE
SUM4 = SUM5 + SUM6 + SUM7
WRITE(6,129) SUM1
WRITE(6,130) SUM2
WRITE(6,131) SUM3
WRITE(6,133) SUM7
WRITE(6,134) SUM5
WRITE(6,135) SUM6
WRITE(6,132) SUM4

```

```

30 CONTINUE
  IF(IN8) 31,32,31
31 PUNCH 118,TIME,Y(IPP,IP1),Y(IPP,IP2),Y(IPP,IP3),Y(IPP,IP4),
  1Y(IPP,IP5),Y(IPP,IP6),Y(IPP,IP7)
32 CONTINUE
101 FORMAT(1H1,' MSF PLANT RESULTS')
102 FORMAT(1H0,117HSTAGE  OUTLET      AVERAGE      AVERAGE      MAIN
1CELL      AVERAGE      TRAY      PRODUCT      PRODUCT      MAIN
2/ 116H      TUBE BRINE      TUBE BRINE      TUBE      VAPOR      MAIN
3 TRAY BRINE      BRINE      TRAY LIQUID      TRAY LIQUID /116H      MAIN
4 TEMPERATURE      TEMPERATURE      TEMPERATURE      MASS      TEMPERATU      MAIN
5RE      MASS      MASS      TEMPERATURE / 116H      (DEG F)      MAIN
6      (DEG F)      (DEG F)      (POUNDS)      (DEG F)      (POU      MAIN
7NDS)      (POUNDS)      (DEG F)      )      MAIN
103 FORMAT (I4,OPF11.3,2F14.3,1PD15.6,OPF13.3,1PD15.6,OPF12.3)
105 FORMAT(21HSTEADY STATE RESULTS)
108 FORMAT(1P8D13.5)
109 FORMAT(1H0, 19H A MATRIX FOR STAGE, I5)
111 FORMAT(I5,1P8D12.4)
112 FORMAT (1H0,13H ZTOT RESULTS)
113 FORMAT(1H0,107H STAGE  WTBI      WTDO      WTBF      WPF
1      WC      WPI      WPD      PV      TV)
114 FORMAT(I5,1P9D12.5)
115 FORMAT(1H1,38H CONSTANTS AND STEADY STATE QUANTITIES)
116 FORMAT(1H0,7H WSEA =, 1PD12.5,8H WTUBE =,1PD12.5,8H SMPMS =,1PD12.
  15,8H TI(3) =,1PE12.5//8H FRSEA =,1PD12.5,8H FRTB =,1PD12.5,8H TBH
  2V =,1PD12.5)
117 FORMAT(1H0,109H  I      FB1(I)      FB2(I)      FB3(I)      FTB(I)
1      FDF(I)      FP1(I)      FP2(I)      FWP(I)      FPF(I))
118 FORMAT(8F10.1)
119 FORMAT(1H0,110H  I      TMASS(I)      CTUBMS(I)      CPTB(I)      CPB(I)
1      B5(I)      PDC(I)      FUDHI(I)      FUDHO(I)      FOUL(I))
120 FORMAT(1H0,40H STAGE 1 TRAY BRINE INLET TEMPERATURE = ,OPF10.3)
122 FORMAT(1H0,110H STAGE TTBI(I)      TPI(I)      TTI(I)      HF
1      DMTB      DMP      DMCV      CONC      CNCT )
125 FORMAT(1H0,20H HP MATRIX FOR STAGE,I5)
129 FORMAT(32HOTOTAL CELL VAPOR IN MSF PLANT =,2PD15.5, 7H POUNDS)
130 FORMAT(32HOTOTAL TRAY BRINE IN MSF PLANT =,4PD15.5, 7H POUNDS)
131 FORMAT(32HOTOTAL PRODUCT IN MSF PLANT =,3PD15.5, 7H POUNDS)
132 FORMAT(37HOTOTAL MASS CHANGE IN LAST TIME STEP=,1PD13.6,7H POUNDS)
133 FORMAT(23HOCHANGE IN CELL VAPOR =,1PD13.6,7H POUNDS)
134 FORMAT(23HOCHANGE IN TRAY BRINE =,1PD13.6,7H POUNDS)
135 FORMAT(23HOCHANGE IN PRODUCT =,1PD13.6,7H POUNDS)
136 FORMAT(1H0,117HSTAGE  OUTLET      AVERAGE      AVERAGE      MAIN
1CELL      AVERAGE      TRAY      PRODUCT      PRODUCT      MAIN
2/ 116H      TUBE BRINE      TUBE BRINE      TUBE      VAPOR      MAIN
3 TRAY BRINE      BRINE      TRAY LIQUID      TRAY LIQUID /116H      MAIN
4 TEMPERATURE      TEMPERATURE      TEMPERATURE      MASS      TEMPERATU      MAIN
5RE      LFVEL      LEVEL      TEMPERATURE / 116H      (DEG F)      MAIN
6      (DEG F)      (DEG F)      (POUNDS)      (DEG F)      (INC      MAIN
7HES)      (INCHES)      (DEG F)      )      MAIN
137 FORMAT (I4,OPF11.3,2F14.3,1PD15.6,OPF13.3,2F15.3,F12.3)
10 CONTINUE
  RETURN
  END

```

```

SUBROUTINE ICON(I)
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /EVAP1/ X(8,50), XIC(8,50), ZT(8,50), Y(8,50), WMAT(30,50)
  COMMON /EVAP2/ PDC(50), TMASS(50), AO(50), AI(50),
1 CTUBMS(50), B5(50), FPF(50), FB1(50), FB2(50), FB3(50), FTB(50),
2 FPI(50), FP2(50), FWP(50), FOUL(50), FBF(50), FUDHI(50), FUDHQ(50)
  COMMON /EVAP3/ WTBO(50), WTBI(50), WTBFI(50), WPI(50), WC(50),
1 WPF(50), WPD(50), CPR(50), CPTB(50), CPP(50), CCP(50), CONC(50),
2 CNCT(50), SMTT(50), SMTB(50)
  COMMON /EVAP4/ TTI(50), TTBI(50), TPI(50), BL(50), PL(50), BLI(50)
1, PLI(50), DMTB(50), DMP(50)
  COMMON /EVAP5/ WSEA, WTUBE, FRSEA, FRTB, SMPMS, TPSUMP, TSEA,
1 TMAX, DEAIR, SSUMP, CSMP, CPSMP, CSEA
  COMMON /EVAP6/ SD(24), FD(54), CNG(8), TI(4), TITLE(20), EXPL(20)
1, ST, WMIN
  COMMON /EVAP7/ NK, NNK, NTMSP, NE, LCT, NSTGS, ISECT, NARROW, NEWW
1, ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)
C  INITIAL CONDITION CALCS. FOR STAGE I
C  SETS XIC(N,I) = I.C.-S FOR N=1,8
C  USES SUBROUTINES SDEN
C  FIXED DATA FD TO HAVE BEEN READ IN
C  PREVIOUSLY BY MAIN PROGRAM
  IN4 = IND(4)
  IN6 = IND(6)
  CN1 = 0.000
  CN2 = 1.000
  DS6 = SD(6)
  DS18 = SD(18)
  DS13 = SD(13)
  V1 = FD(51)
  TV = TSAT(DS13)
  HF = HFG(TV)
  IF(I.EQ.1.AND.NEWW.NE.0) SD(3) = TV
  CONC(I) = DS18
  BL(I) = SD(19)
  PL(I) = SD(20)
  BLI(I) = SD(19)
  PLI(I) = SD(20)
  A1 = 1.00/(3.603*FD(14))
  TMASS(I) = FD(ISECT+19)*A1*SD(12)/FD(ISECT+24)
  AO(I) = SD(16)/FD(14)
  AI(I) = AO(I)*FD(ISECT)/FD(ISECT+3)
  DF57 = FD(ISECT + 27)
C  AVG. TRAY BRINE TEMP. N=5
  XIC(5,I) = SD(6)
  CPTB(I) = CP(XIC(5,I),DS18)
C  PRODUCT TEMPERATURE N=8
  XIC(8,I) = SD(3)
  CPP(I) = CP(XIC(8,I),CN1)
  CCP(I) = CP(TV,CN1)
  CTUBMS(I) = FD(ISECT+40)*AO(I)
  IF(NEWW) 4,3,4
4 SD(21) = 5.0
  CNCT(I) = DF57
C  PURE BRINE OUTLET TEMP. N=1
  IF(I.EQ.1) GO TO 1
  TTBI(I) = XIC(5,I-1)
  TPI(I) = XIC(8,I-1)

```

```

XIC(1,1)=TUBQS
GO TO 2
1 XIC(1,1) = FD(24)
  WTBI(1) = FD(21)*A1
  WPI(1) = 0.000
  TTBI(1) = FD(23)
  TPI(1) = TV
  WPF(1) = 0.000
2 CONTINUE
  TUBQS=SD(8)
C   AVG. TUBE BRINE TEMP. N=2
  XIC(2,1) = 0.500*(TUBQS + XIC(1,1))
C   AVG. TUBE TEMP. N=3
C   ASSUME FOULING ON OUTSIDE, CALC U FOR ARITH DELTA T
  IF(I.EQ.NK) TUBQS = FD(35)
  WTBF(I) = WTBI(1)*CPTB(I)*(TTBI(I) - XIC(5,1))/(HF + CPTB(I)*
1 (TTBI(I) - XIC(5,1)) - CCP(I)*(TTBI(I) - TV))
  IF(I.GT.1) WPF(I) = (CPP(I)*WPI(I))*(TPI(I) - SD(3)) + CCP(I)*
1 WTBF(I)*(TV - SD(3))/HF
  WPO(I) = WPI(I) + WTBF(I)
  WTBO(I) = WTBI(1) - WTBF(I)
  CPB(I) = CP(XIC(2,1),DF57)
  IF(I.EQ.MSTGS) GO TO 8
  WTBI(I+1) = WTBO(I)
  WPI(I+1) = WPO(I)
  WC(I) = WTBF(I) + WPF(I)
  GO TO 7
8 WC(I) = FD(22)*A1*CPR(I)*(XIC(1,1) - TUBQS)/HF
  DEAIR = WTBF(I) + WPF(I) - WC(I)
  WPO(I) = WPI(I) + WC(I) - WPF(I)
7 CONTINUE
  UBAR = HF*WC(I) / (SD(16)*A1*(TV-XIC(2,1)))
C   U-BAR IN BTU/HR.FT**2.F OUTSIDE AREA
  DTEMP = (UBAR/SD(15))*(TV - XIC(2,1))
C   ASSUME H-IN (SD(15)) BASED ON OUTSIDE AREA
  XIC(3,1) = XIC(2,1) + DTEMP
  IF(IN4.EQ.0) GO TO 11
  WRITE(6,101) UBAR, HF, DTEMP
101 FORMAT(8H0UBAR = ,1PD14.6,15H      HF = ,1PD14.6,18H
1 DTEMP = ,1PD14.6)
C   CELL VAPOR MASS (LB) N=4
11 SLEN = 1.00/VVAP(IV,DS13)
  IF(ISECT.EQ.2)XIC(4,1) = SDEN*V1*FD(16)/FLOAT(NK)
  IF(ISECT.EQ.3)XIC(4,1) = SDEN*V1*FD(17)/FLOAT(NNK)
  GO TO 5
3 XIC(1,1) = SD(22)
  XIC(2,1) = SD(23)
  XIC(3,1) = SD(24)
  XIC(4,1) = SD(17)
  XIC(6,1) = SD(11)
  XIC(7,1) = SD(21)
  WTBO(I) = A1*SD(1)
  WTBF(I) = A1*SD(4)
  WPO(I) = A1*SD(5)
  WPI(I) = A1*SD(2)
  CNCT(I) = SD(14)
  CPB(I) = CP(XIC(2,1),CNCT(I))
  IF(I.EQ.1) GO TO 9

```

```

    TTBI(I) = XIC(5,I-1)
    TPI(I) = XIC(8,I-1)
    WPI(I) = WPO(I-1)
    WTBI(I) = WTBO(I-1)
    GO TO 10
9  WTBI(1) = FD(21)*A1
    WPI(1) = 0.000
    TTBI(1) = FD(23)
    TPI(1) = TV
10  WC(I) = A1*SD(10)
    IF(I.EQ.NSTGS) DEAIR = WTBF(I) + WPF(I) - WC(I)
    GO TO 6
5  CONTINUE
    SD(11) = SD(11)*SD(19)/(FD(13)*12.0)
C   MASS OF TRAY BRINE (LB) N=6
    XIC(6,I) = WTBO(I)*SD(11)
C   MASS OF PRODUCT (LB) N=7
C   SD(21) IS THE PRODUCT HOLDUP TIME ( 5 SEC. ORIGINALLY)
    XIC(7,I) = SD(21)*WPO(I)
6  CONTINUE
    IF(I.EQ.NK) TUBQS = FD(35)
    RETURN
    END

SUBROUTINE SDATA(I)
    IMPLICIT REAL*8(A-H,O-Z)
    COMMON /EVAP1/ X(8,50), XIC(8,50), ZT(8,50), Y(8,50), WMAT(30,50)
    COMMON /EVAP2/ PDC(50), TMASS(50), AD(50), AI(50),
1  ICTUBMS(50), B5(50), FPF(50), FB1(50), FB2(50), FB3(50), FTB(50),
2  FPI(50), FP2(50), FWP(50), FOUL(50), FBF(50), FUDHI(50), FUDHO(50)
    COMMON /EVAP3/ WTBO(50), WTBI(50), WTBF(50), WPI(50), WC(50),
1  WPF(50), WPO(50), CPB(50), CPTB(50), CPP(50), CCP(50), CONC(50),
2  CNCT(50), SMTT(50), SMTB(50)
    COMMON /EVAP4/ TTI(50), TTBI(50), TPI(50), BL(50), PL(50), BLI(50)
1  , PLI(50), DMTB(50), DMP(50)
    COMMON /EVAP6/ SD(24), FD(54), CNG(8), TI(4), TITLE(20), EXPL(20)
1  , ST, WMIN
    COMMON /EVAP7/ NK, NNK, NTMSP, NE, LCT, NSTGS, ISECT, NARROW, NEWW
1  , ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)
    COMMON /EVAP8/ VENT, IATM
    COMMON /EVAP9/ SSD(50)
C   CALCULATES FIXED COEFFS. FOR AMATS + ZTOT
C   USES SUBROUTINES CP, SDEN, DVPSD, HFG, DEN
C   ISECT=2 FOR EVAP., =3 FOR REJECT
    A1 = 1.00/(3.603*FD(14))
    IN4 = IND(4)
    CN1 = 0.000
    CN2 = 1.000
    DS3 = SD(3)
    DS6 = SD(6)
    DS18 = SD(18)
    PV = SD(13)
    TV = TSAT(PV)
    IF( ISECT-2) 20,20,21
20  VV = FD(16)*FD(51)/FLOAT(NK)
    GO TO 22

```



```

21 VV = FD(17)*FD(51)/FLOAT(NNK)
22 CONTINUE
   IF(NEWH.EQ.0) GO TO 7
   P6 = 1.0D0/VVAP(TV,PV)
   B5(I)=DVPSD(P6)/VV
   GO TO 8
7 B5(I) = SD(9)
8 CONTINUE
   SSD(I) = B5(I)
   PDC(I) = PV - B5(I)*XIC(4,I)
C   H-IN FUDGE FACTOR
   WT = FD(ISECT+19)*A1
   HI = HIN(XIC(2,I),WT,CN2)
   FUDHI(I)=SD(15)/(HI*3.6D3)
C   TUBER=(FD(ISECT+3)-FD(ISECT))/(2.4D1*FD(ISECT+6)).
   H-OUT FUDGE FACTOR
   UBAR = HFG(TV)*WC(I)/(SD(16)*A1*(TV - XIC(2,I)))
   HQ = HOUT(XIC(3,I),TV,CN1,CN2)
   AFOUL = FD(ISECT + 9) + TUBER
   IF(VENT.LE.1.0D0) GO TO 6
   IF(PV.GE.14.7D0) GO TO 6
   PVC = DMA X1(PV,8.0D0)
   AFOUL = AFOUL*(1.0D0 + (VENT - 1.0D0)*DSQRT((14.7D0-PVC)/6.7D0))
6 FUDHQ(I) = 1.0D0/(3.6D3*(1.0D0/UBAR - 1.0D0/SD(15) - AFOUL)*HQ)
   IF(I.GT.1)GO TO 1
   FPF(1)=0.0D0
   PVI = PV
   ENN=0.5D0
   ENP=0.5D0
   GO TO 2
1 FPF(I) = WPF(I)/(DSQRT(WPIS)*(TPI(I) - TV))
2 WTBIS = WTBQ(I)
   FBF(I) = WTB(I)/(DSQRT(WTBI(I))*(TTBI(I) - TV))
C   ASSUME SIMPLIFIED ORIFICE EQN.
   FB2(I)=0.0D0
C   AFC=BRINE TRAY BASE AREA, IN.**2
   AFC =1.728D3*XIC(6,I)/(DEN(DS6,DS18)*SD(19))
C   SD(19) IS BRINE DEPTH, IN.
C   ASSUME SIMPLIFIED ORIFICE EQN. FOR PRODUCT
   AFCP=1.728D3*XIC(7,I)/(DEN(DS3,CN1)*SD(20))
   IF(I.GT.1)GO TO 3
   AF,PS=AFC,P
   AFCS=AFC
   FB1(1)=0.0D0
   FB3(1)=0.0D0
   FTB(1)=0.0D0
   WPIS=0.0D0
   FPI(1) = 0.0D0
   FP2(1) = 0.0D0
   FWP(1) = 0.0D0
   GO TO 4
3 CONTINUE
   FB1(I) = 1.0D0/AFCS
   FB3(I)=FD(49)/AFC
   ODPEFFO=PVI- PV +FB1(I)*XIC(6,I-1)
   1 - FB2(I)*(TTBI(I) - SD(7)) - FB3(I)*XIC(6,I)
   FTB(I) = WTB(I)/(DPEFFO**ENN)
C   PRODUCT MASS BALANCE

```

```

      FP1(I) = 1.000/AFCPS
      FP2(I)=FD(49)/AFCP
      DPEPO = PVI - PV + FP1(I)*XIC(7,I-1) - FP2(I)*XIC(7,I)
      FWP(I)=WPIS/(DPEPO**ENP)
4    CONTINUE
C    FOUL INCLUDES TUBE RESISTANCE
      FOUL(I)=(FD(ISECT+9)+TURER)*3.6D3
      IF(IN4.EQ.0) GO TO 5
      HO = HOUT(XIC(3,I),TV,FOUL(I),FUDHO(I))
      HI = HIN(XIC(2,I),WT,FUDHI(I))
      WRITE(6,101) UBAR, HI,HO
101  FORMAT(1H0,9H UBAR =,F10.3,6H HI = ,F10.3,6H HO = ,F10.3)
5    CONTINUE
C    SET SAVED VARIABLES FOR NEXT I
      PVI = PV
      AFCS=AFCS
      AFPCS=AFCP
      WPIS = WPO(I)
      WMAT(1,I) = WTBI(I)
      WMAT(2,I) = WTBO(I)
      WMAT(3,I) = WTBF(I)
      WMAT(4,I) = WPF(I)
      WMAT(5,I) = WC(I)
      WMAT(6,I) = WPI(I)
      WMAT(7,I) = WPO(I)
      WMAT(8,I) = PV
      WMAT(9,I) = TV
      RETURN
      END

```

```

SUBROUTINE ZTOT(I)
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /GEN1/ TIME, DT, TNXT
  COMMON /EVAP1/ X(8,50), XIC(8,50), ZT(8,50), Y(8,50), WMAT(30,50)
  COMMON /EVAP2/ PDC(50), TMASS(50), AD(50), AI(50),
  1CTUBMS(50), B5(50), FPF(50), FB1(50), FB2(50), FB3(50), FTB(50),
  2FP1(50), FP2(50), FWP(50), FOUL(50), FBF(50), FUDHI(50), FUDHO(50)
  COMMON /EVAP3/ WTBO(50), WTBI(50), WTBF(50), WPI(50), WC(50),
  1WPF(50), WPO(50), CPB(50), CPTB(50), CPP(50), CCP(50), CONC(50),
  2CNC(50), SMTT(50), SMTB(50)
  COMMON /EVAP4/ TTI(50), TTBI(50), TPI(50), BL(50), PL(50), BLI(50)
  1, PLI(50), DMTB(50), DMP(50)
  COMMON /EVAP5/ WSEA, WTUBE, FRSEA, FRTB, SMPMS, TPSUMP, TSEA,
  1TMAX, DCAIR, SSUMP, CSMP, CPSMP, CSEA
  COMMON /EVAP7/ NK, NNK, NTMSP, NE, LCT, NSTGS, ISECT, NARROW, NEWW
  1, ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)
  COMMON /EVAP8/ VENT, IATM
  COMMON /BEAT1/ TBHV, TBHO, CBHO, WBHC, PBHV, WBHI, WBHO, SQUAL
  IF(I-NK)11,16,12
12  WT = WSEA
  IF(I.EQ.NSTGS) SALTN = CSEA
  GO TO 13
16  TTI(NK) = TPSUMP
11  WT = WTUBE
  IF(I.EQ.1) SSALT = CBHO
13  CONTINUE

```

ZTOT

ZTOT

ZTOT

ZTOT

```

IF(I.LT.NSTGS) SALTIN = CNCT(I+1)
TAV = X(2,I)
BO=2.00*WT/TMASS(I)
FUD = FUDHI(I)
HI = HIN( TAV, WT, FUD)
B1=AO(I)*HI/(TMASS(I)*CPB(I))
B12 = CCP(I)/CPTB(I)
B13 = CCP(I)/CPP(I)
ZT(1,I)= -B0*X(1,I)+ (B0-B1)*X(2,I) + B1*X(3,I)
IF(I.EQ.NSTGS) GO TO 14
IF(I.EQ.NK) GO TO 24
TTI(I) = X(1,I+1)
GO TO 15
14 TTI(NSTGS) = TSEA
GO TO 15
24 SALTIN = CSMP
15 CONTINUE
ZT(2,I) = -B0*TTI(I) - (B0 + B1)*X(2,I) + B1*X(3,I)
PV = B5(I)*X(4,I) + PDC(I)
IF(I.GT.1) GO TO 35
TV = TSAT( PV)
GO TO 36
35 TV = TVS
36 TTUBE = X(3,I)
AFOUL = FOUL(I)
IF(VENT.LE.1.00) GO TO 37
IF(PV.GE.14.7) GO TO 37
PVC = DMAX1(PV,8.00)
AFOUL = FOUL(I)*(1.00+(VENT-1.00)*DSORT((14.700 - PVC)/6.700))
37 CONTINUE
FUD = FUDHO(I)
HO = HOUT( TTUBE, TV,AFOUL, FUD)
DENOM = CTUBMS(I)
B4 = HO*AO(I)/DENOM
B3 = HI*AO(I)/DENOM
ZT(3,I) = B3*X(2,I) - (B3+B4)*X(3,I) + B4*TV
TTB = X(5,I)
TP = X(8,I)
HF = HFG( TV)
B6 = AO(I)*HO/HF
WC(I) = B6*(TV - TTUBE)
IF(WC(I).LT.0.00)WC(I) = 0.000
IF(I.EQ.1) GO TO 21
WTBI(I) = WTBO(I-1)
WPI(I) = WPO(I-1)
TTBI(I) = X(5,I-1)
TPI(I) = X(8,I-1)
WPF(I) = FPF(I)*DSORT(WPI(I))*(TPI(I) - TV)
IF(WPF(I).LT.0.00)WPF(I) = 0.000
GO TO 22
21 WPI(I) = 0.000
WTBI(I) = WTUBE
TTBI(1) = TBHO
TPI(1) = TV
WPF(I) = 0.000
22 CONTINUE
RTWTBI = DSORT(WTBI(I))
WTBF(I) = FBF(I)*RTWTBI*(TTBI(I) - TV)

```

ZTOT

ZTOT

ZTOT

ZTOT

ZTOT

ZTOT

ZTOT

ZTOT

ZTOT

```

IF(WTBF(I).LT.0.000) WTBF(I) = 0.000
IF(I.EQ.NSTGS) GO TO 18
ZTOT
ZT(4,I) = WTBF(I) + WPF(I) - WC(I)
PVS = B5(I+1)*X(4,I+1) + PDC(I+1)
TVS = TSAT(PVS)
AMTBS = X(6,I+1)
PEFF = PV-PVS + FB1(I+1)*X(6,I) - FB2(I+1)*(TTB-TVS)-FB3(I+1)*AMTBS
WTBO(I) = FTB(I+1)*DSQRT(PEFF)
ZT(6,I) = WTBI(I) - WTBO(I) - WTBF(I)
GO TO 19
ZTOT
18 ZT(6,I) = 0.000
ZT(7,I) = 0.000
ZT(4,I) = WTBF(I) + WPF(I) - WC(I) - DEAIR
WPO(I) = WPI(I) + WC(I) - WPF(I)
WTBO(I) = WTBI(I) - WTBF(I)
GO TO 20
ZTOT
19 CONTINUE
ZTOT
PPEFF = PV-PVS + FP1(I+1)*X(7,I) - FP2(I+1)*X(7,I+1)
IF(PPEFF.LE. 0.000) GO TO 27
WPO(I) = FWP(I+1)*DSQRT(PPEFF)
GO TO 28
27 WRITE(6,104)I,PPEFF,TIME
104 FORMAT(15H PPEFF IN STAGE,I5,18H FOUND NEGATIVE = ,D12.5,11H AT TI
TIME = ,D12.5)
GO TO 33
28 ZT(7,I) = WPI(I) - WPO(I) - WPF(I) + WC(I)
ZTOT
20 CONTINUE
B7 = (WTBI(I) - WTBF(I))/X(6,I)
B8 = WTBF(I)*HF/(X(6,I)*CPTB(I))
ZT(5,I) = B7*(TTBI(I)-TTB) - B8 - WTBF(I)*B12*(TV-TTBI(I))/X(6,I)
IF(I.EQ.1) GO TO 23
ZTOT
23 B9 = WPI(I)/X(7,I)
B10 = WPF(I)*HF/(X(7,I)* CPP(I))
B11 = (WC(I) - WPF(I))/X(7,I)
ZT(8,I)=B9*( TPI(I)-TP) - B10 + B11*( TV - TP)*B13
SMTB(I) = SMTB(I) + DT*(WTBI(I)*SSALT - WTBO(I)*CONC(I))
SMTT(I) = SMTT(I) + DT*WT *(SALTN - CNCT(I))
SSALT = CONC(I)
WMAT(1,I) = WTBI(I)
WMAT(2,I) = WTBO(I)
WMAT(3,I) = WTBF(I)
WMAT(4,I) = WPF(I)
WMAT(5,I) = WC(I)
WMAT(6,I) = WPI(I)
WMAT(7,I) = WPO(I)
WMAT(8,I) = PV
WMAT(9,I) = TV
WMAT(10,I) = TTBI(I)
WMAT(11,I) = TPI(I)
WMAT(12,I) = TTI(I)
WMAT(13,I) = HF
GO TO 10
33 IERROR = 1
10 CONTINUE
RETURN
ZTOT
END
ZTOT

```

```

SUBROUTINE AMATS(I)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /MAT1/ A(8,8,50), HP(8,8,50), AB(6,6), HHP(6,6)
COMMON /EVAP1/ X(8,50), XIC(8,50), ZT(8,50), Y(8,50), WMAT(30,50)
COMMON /EVAP2/ PDC(50), TMASS(50), AO(50), AI(50),
1CTUBMS(50), B5(50), FPF(50), FB1(50), FB2(50), FB3(50), FTB(50),
2FPI(50), FP2(50), FWP(50), FOUL(50), FBF(50), FUDHI(50), FUDHO(50)
COMMON /EVAP3/ WTBO(50), WTBI(50), WTBF(50), WPI(50), WC(50),
1WPF(50), WPO(50), CPR(50), CPTB(50), CPP(50), CCP(50), CONC(50),
2CNCT(50), SMTT(50), SMTB(50)
COMMON /EVAP5/ WSEA, WTUBE, FRSEA, FRTB, SMPMS, TPSUMP, TSEA,
1TMAX, DEAIR, SSUMP, CSMP, CPSMP, CSEA
COMMON /EVAP7/ NK, NNK, NTMSP, NE, LCT, NSTGS, ISECT, NARROW, NEWH
1, ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)
C THIS SUBROUTINE CALCULATES THE COEFFICIENTS OF THE A MATRIX
IF(I-NK)11,11,12
12 WT = WSEA
ISECT = 3
GO TO 13
11 WT = WTUBE
ISECT = 2
13 CONTINUE
PV = B5(I)*X(4,1) + PDC(I)
TV = TSAT(PV)
CN1 = 0.0D0
CPTB(I) = CP(X(5,I),CONC(I))
CPB(I) = CP(X(2,I),CNCT(I))
CPP(I) = CP(X(8,I),CN1)
CCP(I) = CP(TV,CN1)
B12 = CCP(I)/CPTB(I)
B13 = CCP(I)/CPP(I)
TAV = X(2,I)
FUD = FUDHI(I)
HI = HIN(TAV,WT,FUD)
B0 = 2.0D0*WT/TMASS(I)
B2 = AO(I)*HI/(TMASS(I)*CPB(I))
A(1,3,I) = B2
A(1,2,I) = B0 - B2
A(2,2,I) = -(B0 + B2)
A(2,3,I) = B2
C NEXT ARE TUBE TEMPERATURE COEFFICIENTS
DTDP = DSTVP(PV)
TTUBE = X(3,I)
FUD = FUDHO(I)
HO = HOUT(TTUBE,TV,FOUL(I),FUD)
DENOM = CTUBMS(I)
B4 = HO*AO(I)/DENOM
D3 = HI*AO(I)/DENOM
A(3,2,I) = B3
A(3,3,I) = -(B3+B4)
A(3,4,I) = B5(I)*DTDP*B4
TTB = X(5,I)
TP = X(8,I)
HF = HFG(TV)
F8 = -AO(I)*HO/HF
F7 = -F8*B5(I)*DTDP
AMIB = X(6,I)
XBF = FBF(I)

```

AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

```

      AMP = X(7,I)
      IF(I.EQ.1) GO TO 25
      PVS=B5(I-1)*X(4,I-1) + PDC(I-1)
      F3 = -XBF*B5(I)*DTDP*DSQRT(WTBI(I))
      F6 = -FPF(I)*DSQRT(WPI(I))*B5(I)*DTDP
      GO TO 26
25  CONTINUE
      F3 = -XBF*B5(I)*DTDP*DSQRT(WTBI(I))
      F6 = 0.0
26  CONTINUE
      B7 = (HF/CPTB(I) + B12*(TV - TTB))/AMTB
      B8 = B13*(TV - TP)/AMP
      B10 = -B8 - HF/(CPP(I)*AMP)
      CON = WC(I) - WPF(I)
C    NEXT CALCULATE CELL VAPOR MASS COEFFICIENTS
      A(4,3,I) = F8
      A(4,4,I) = F3 + F6 + F7
      IF(I.EQ.NSTGS) GO TO 22
C    NEXT CALCULATE COEFFICIENTS FOR PRODUCT TRAY TEMPERATURE
      A(5,5,I) = -ZT(6,I)/AMTB
      A(5,4,I) = -F3*B7 - B12*B5(I)*DTDP*WTBF(I)/AMTB
      A(5,6,I) = -ZT(5,I)/AMTB
C    NEXT CALCULATE COEFFICIENTS FOR TRAY BRINE MASS
      A(6,4,I) = -F3
C    NEXT CALCULATE COEFFICIENTS FOR PRODUCT TRAY MASS
      A(7,3,I) = F8
      A(7,4,I) = (F7 - F6)
      A(8,3,I) = F8*B8
      A(8,4,I) = B13*CON*B5(I)*DTDP/AMP + B8*F7 + B10*F6
      A(8,7,I) = -ZT(8,I)/AMP
      A(8,8,I) = -ZT(7,I)/AMP
      IF(ISM(I) - 2) 21,27,27
22  CONTINUE
      A(8,3,I) = F8*B8
      A(8,4,I) = B13*CON*B5(I)*DTDP/AMP + B8*F7 + B10*F6
27  A(5,5,I) = 0.0D0
      A(5,6,I) = 0.0D0
      A(6,4,I) = 0.0D0
      A(5,4,I) = -F3*B7 - B12*B5(I)*DTDP*WTBF(I)/AMTB
21  CONTINUE
C    FUDGE THE A MATRIX TO MINIMIZE AMAX/AMIN RATIO IN EMATS
      A2 = 1.0D4
      A3 = 1.0D3
      A(5,6,I) = A2*A(5,6,I)
      A(6,4,I) = A(6,4,I)/A2
      A(7,3,I) = A(7,3,I)/A3
      A(7,4,I) = A(7,4,I)/A3
      A(8,7,I) = A(8,7,I)*A3
200 CONTINUE
100 RETURN
      END

```

AMATS
AMATS

AMATS

AMATS

AMATS
AMATS
AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

AMATS

```

      SUBROUTINE UPS1(I)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /MAT1/  A(8,8,50),  HP(8,8,50), AB(6,6), HHP(6,6)
      COMMON /EVAP1/ X(8,50), XIC(8,50), ZT(8,50), Y(8,50), WMAT(30,50)
      COMMON /EVAP6/ SD(24), FD(54), CNG(8), TI(4), TITLE(20), FXPL(20)
      1, ST, WMIN
      COMMON /EVAP7/ NK, NNK, NTMSP, NE, LCT, NSTGS, ISECT, NARROW, NEWW
      1, ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)
      DIMENSION SX(8,50)
      C THIS SUBROUTINE CALCULATES THE FUDGED FORCING FUNCTIONS AND
      C CALCULATES NEW VALUES FOR THE STATE VARIABLES FOR STAGE I FOR A
      C TIME STEP
      C FIRST ADJUST X MATRIX AND ZTOTS TO ACCOUNT FOR A MATRIX FUDGING
      A1 = 10.00
      A2 = 100.00
      A3 = 0.100
      A4 = 0.0100
      DO 15 N = 1,5
      X(N,I) = A2*X(N,I)
      ZT(N,I) = A2*ZT(N,I)
      15 CONTINUE
      X(6,I) = A4*X(6,I)
      X(7,I) = A3*X(7,I)
      X(8,I) = A2*X(8,I)
      ZT(6,I) = A4*ZT(6,I)
      ZT(7,I) = A3*ZT(7,I)
      ZT(8,I) = A2*ZT(8,I)
      C NOW CALCULATE NEW STATE VARIABLES
      DO 20 N = 1,NE
      SUM2 = 0.000
      DO 21 L = 1, NE
      SUM2 = SUM2 + HP(N,L,I)*ZT(L,I)
      21 CONTINUE
      Y(N,I) = X(N,I) + SUM2
      IF(ICT(I).EQ.1) GO TO 25
      DIF = DABS((Y(N,I) - SX(N,I))/SX(N,I))
      IF( DIF - CNG(N)) 20,27,27
      27 ISM(I) = 1
      GO TO 20
      25 SX(N,I) = Y(N,I)
      20 CONTINUE
      C WE HAVE FINISHED CALCULATING NEW STATE VARIABLES
      C ADJUST Y(N,I) AND ZT(N,I) BACK TO NON-FUDGED VALUES
      DO 14 N = 1,5
      Y(N,I) = A4*Y(N,I)
      X(N,I) = A4*X(N,I)
      ZT(N,I) = A4*ZT(N,I)
      14 CONTINUE
      Y(6,I) = A2*Y(6,I)
      X(6,I) = A2*X(6,I)
      Y(7,I) = A1*Y(7,I)
      X(7,I) = A1*X(7,I)
      Y(8,I) = A4*Y(8,I)
      X(8,I) = A4*X(8,I)
      ZT(6,I) = A2*ZT(6,I)
      ZT(7,I) = A1*ZT(7,I)
      ZT(8,I) = A4*ZT(8,I)
      RETURN
      END

```

```

SUBROUTINE BHEAT(I)
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /GEN1/ TIME, DT, TNXT
  COMMON /GEN4/ IPRT
  COMMON /MAT1/ A(8,8,50), HP(8,8,50), AB(6,6), HHP(6,6)
  COMMON /EVAP1/ X(8,50), XIC(8,50), ZT(8,50), Y(8,50), WMAT(30,50)
  COMMON /EVAP2/ PDC(50), TMASS(50), AD(50), AI(50),
  1 CTUBMS(50), B5(50), FPF(50), FB1(50), FB2(50), FB3(50), FTB(50),
  2 FPI(50), FP2(50), FWP(50), FOUL(50), FBF(50), FUDHI(50), FUDHO(50)
  COMMON /EVAP3/ WTBO(50), WTBI(50), WTBFI(50), WPI(50), WC(50),
  1 WPF(50), WPO(50), CPB(50), CPTB(50), CPP(50), CCP(50), CONC(50),
  2 CNCT(50), SMTT(50), SMTB(50)
  COMMON /EVAP5/ WSEA, WTUBE, FRSEA, FRTB, SMPMS, TPSUMP, TSEA,
  1 TMAX, DEAIR, SSUMP, CSMP, CPSMP, CSEA
  COMMON /EVAP6/ SD(24), FD(54), CNG(8), TI(4), TITLE(20), EXPL(20)
  1, ST, WMIN
  COMMON /EVAP7/ NK, NNK, NTMSP, NE, LCT, NSTGS, ISECT, NARROW, NEWW
  1, ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)
  COMMON /BEAT1/ TBHV, TBHO, CBHO, WBHC, PBHV, WBHI, WBHO, SQUAL
  COMMON /BEAT2/ TBHC, HBI, BHT(6)
  DIMENSION PS(2), PS(2), W(2,200), T(2,200), MTM(2)
  DIMENSION BZT(6), TS(6), SX(6), SCON(2,200)
  C   CALCULATES THE CONDITIONS IN THE BRINE HEATER
  IF(I.GT.0) GO TO 120
  IN1 = IND(1)
  IN4 = IND(4)
  A1 = 1.000/(3.603*FD(14))
  CN1 = 0.000
  CN2 = 1.000
  PI = 3.141592700
  ATO = PI*FD(4)*FD(37)*FD(18)/(FD(14)*12.00)
  BTMASS = FD(21)*A1*FD(37)/FD(25)
  C   FNT IS THE NO. OF TRAINS PER BRINE HEATER
  FNT = FD(47)
  TBHO = FD(23)
  TBHI = FD(24)
  HFBH = HVAP(TBHV,PBHV) - HLIQ(TBHV)
  IF(NEWW.EQ.1) GO TO 1
  TBHA = FD(32)
  TBHT = FD(33)
  AMV = FD(34)
  AMC = FD(38)
  TBHC = FD(39)
  WBHC = FD(40)
  GO TO 2
  1 CONTINUE
  TBHA = 0.500*(TBHI + TBHO)
  SDEN = 1.000/VVAP(TBHV,PBHV)
  AMV = FD(46)*SDEN
  AMC = WBHI*5.000
  TBHC = TBHV
  WBHC = WBHI*SQUAL
  2 CONTINUE
  HI = HIN(TBHA,WTUBE,CN2)
  FUD = FD(15)/(HI*3.603)
  UBAR = HFBH*WBHC/(ATO*FNT*(TBHV - TBHA))
  HBI = HIN(TBHA,WTUBE,FUD)
  CBH = FD(29)

```

BCON


```

CPBH = CP(TBHA,CBH)
IF(NEWW.EQ.0) GO TO 3
BHB5 = DVPSD(SDEN)/FD(46)
FD(50) = BHB5
TBHT = TBHA + WTUBE*CPBH*(TBHO - TBHI)/(ATO*HBI)
3 CONTINUE
HO = HOUT(TBHT,TBHV,CN1,CN2)
TUBER = (FD(4) - FD(1))/(2.401*FD(7))
FOULBH = (FD(10) + TUBER)*3.6D3
FUDBH = 1.00/((1.00/UBAR - 1.00/HBI - FOULBH)*HO)
HO = HOUT(TBHT,TBHV,FOULBH,FUDBH)
CTBMS = FD(41)*ATO
IF(NEWW.EQ.0) BHB5 = FD(50)
RHPDC = PRHV - BHB5*AMV
WBHO = WBHT
BHT(1) = TBHO
BHT(2) = TBHA
BHT(3) = TBHT
BHT(4) = AMV
BHT(5) = AMC
RHT(6) = TBHC
SX(1) = TBHO
SX(2) = TBHA
SX(3) = TBHT
SX(4) = AMV
SX(5) = AMC
SX(6) = TBHC
IF(NEWW.EQ.0) GO TO 4
HTIN = WBHC*HFBH
FNT = HTIN/(ATO*HO*(TBHV - TBHT))
C FNT INCLUDES HFAT BALANCE CORRECTION, IF ANY
FD(47) = FNT
4 CONTINUE
HTIN = ATO*HO*(TBHV - TBHT)
401 WRITE (6,401) UBAR,HBI,HO,FUDBH,FOULBH,ATO,CTBMS,CPBH, FNT
FORMAT(1P9D12.5)
WRITE(6,406) HTIN,TBHV, WBHC
TIM = TBHI
DO 26 L = 1,6
DO 28 N = 1,6
AB(L,N) = 0.000
28 CONTINUE
BZT(L) = 0.000
26 CONTINUE
LBH = 0
C SET UP INITIAL CONDITIONS FOR TIME DELAY
YT = TBHI
DO 20 N=1,2
PSI(N) = WTUBE*TI(N)
NTM = TI(N)/DT + 1.000
MTM(N) = PSI(N)/(WMIN*DT) + 1.000
DO 21 K= 1, NTM
SCON(N,K) = CBH
W(N,K) = WTUBE
T(N,K) = YT
21 CONTINUE
YT = TBHO
20 CONTINUE

```

BCON

BCON

BCON

BCON

BCON

BCON

BCON

```

      GO TO 128
C     NEXT CALCULATE OUTLET TEMPERATURE FROM THE BRINE HEATER LOOP      BCON
C     DURING TRANSIENT OPERATIONS                                       BCON
120  CONTINUE
      FNT = FD(47)
      DO 121 N= 1,2
      KK=0
      PS(N) =0.0D0
      KTM = MTM(N)
      DO 122 K=1, KTM
      QS = DT*W(N,K)
      PS(N) = QS + PS(N)
      KK= KK+1
      IF(PS(N).GE.PSI(N)) GO TO 123
122  CONTINUE
      GO TO 400
123  IF(N-1)124,124,125
124  TIN = T(N,KK) -(T(N,KK) -T(N,KK-1))*(PS(N)-PSI(N))/QS
      CBH= SCON(N,KK) - (SCON(N,KK) - SCON(N,KK-1))*(PS(N) - PSI(N))/QS
      KGM = KTM -1
      DO 126 LK = 1,KGM
      JJ = KTM - LK
      T(N,JJ+1) = T(N,JJ)
      W(N,JJ+1) = W(N,JJ)
      SCON(N,JJ+1) = SCON(N,JJ)
126  CONTINUE
      T(N,1) = X(1,1)
      W(N,1) = WTUBE
      SCON(N,1) = CNCT(1)
      GO TO 121
125  TBHO = T(N,KK)-(T(N,KK)-T(N,KK-1))*(PS(N) - PSI(N))/QS
      CBHO= SCON(N,KK) - (SCON(N,KK) - SCON(N,KK-1))*(PS(N) - PSI(N))/QS
      KGM = KTM -1
      DO 127 LK = 1, KGM
      JJ = KTM - LK
      T(N,JJ+1) = T(N,JJ)
      W(N,JJ+1) = W(N,JJ)
      SCON(N,JJ+1) = SCON(N,JJ)
127  CONTINUE
      W(N,1) = WTUBE
      SCON(N,1) = CBH
128  CONTINUE
      BHTO = BHT(1)
      TBHA = BHT(2)
      TBHT = BHT(3)
      AMV = BHT(4)
      AMC = BHT(5)
      TBHC = BHT(6)
      PBHV = BHB5*AMV + BHPDC
      DTDP = DSTVP(PBHV)
      TBHV = TSAT(PBHV)
      HFBH = HVAP(TBHV,PBHV) - HLIQ(TBHV)
      CPV = CP(TBHV,CN1)
      CPC = CP(TBHC,CN1)
      HBI = HIN(TBHA,WTUBE,FUD)
      CPBH = CP(TBHA,CBH)
      HIO = HOUT(TBHT,TBHV,FOULBH,FUDBH)
      BO = 2.0D0*WTUBE/BTMASS

```

```

B1 = ATO*HBI/(B-TMASS*CPBH)
B3 = ATO*HBI/CTBMS
B4 = ATO*HO/CTBMS
B8 = -FNT*ATO*HO/HFBH
R7 = -B8*BHB5*DTDP
WBHC = -B8*(TBHV - TBHT)
C  ASSUME CONTROL SU FLOW OF CONDENSATE OUT = FLOW OF STEAM IN
    WBHO = WBHI
    HTIN = ATO*HO*(TBHV - TBHT)
    BZT(1) = (B0 - B1)*TRHA - B0*BHTO + B1*TBHT
    BZT(2) = B0*TIN - (B0 + B1)*TBHA + B1*TBHT
    BZT(3) = B4*TBHV - (B3 + B4)*TBHT + B3*TBHA
    BZT(4) = WBHI*SQUAL - WBHC
    BZT(5) = WBHC - WBHO + WBHI*(CN2 - SQUAL)
    BZT(6) = (BZT(5) + WBHO)*(TBHV - TBHC)/AMC
    IF(LBH.GT.0) GO TO 23
    AB(1,2) = (B0 - B1)
    AB(1,3) = B1
    AB(2,2) = -(B0 + B1)
    AB(2,3) = B1
    AB(3,2) = B3
    AB(3,3) = -(B3 + B4)
    AB(5,3) = B8
    AB(6,3) = B8*CPV*(TBHV - TBHC)/(AMC*CPC)
    AB(6,5) = -BZT(6)/AMC
    AB(6,6) = -BZT(5)/AMC
    AB(3,4) = BHB5*DTDP*B4
    AB(4,4) = -B7
    AB(4,3) = -B8
    AB(5,4) = B7
    AB(6,4) = CPV*(B7*(TBHV - TBHC) + WBHC*BHB5*DTDP)/(AMC*CPC)
    IBH = 0
    CALL EMATS(IBH,IBH)
    LBH = 1
    NCT = 0
    IF(I.EQ.0) GO TO 400
23  CONTINUE
    DO 24 L = 1,6
    SUM1 = 0.000
    DO 22 M = 1,6
    SUM1 = SUM1 + HHP(L,M)*B7T(M)
22  CONTINUE
    IS(L) = SUM1 + BHT(L)
    IF(NCT.EQ.0) GO TO 25
    DIF = DABS((TS(L) - SX(L))/SX(L))
    IF(DIF - CNG(L)) 24,27,27
27  LBH = 0
    NCT = 0
    GO TO 24
25  SX(L) = TS(L)
24  CONTINUE
    T(N,1) = TS(1)
121 CONTINUE
    NCT = NCT + 1
    IF(NCT.GE.LCT) LBH = 0
    BHT(1) = TS(1)
    BHT(2) = TS(2)
    BHT(3) = TS(3)

```

B.CON

```

      BHT(4) = TS(4)
      BHT(5) = TS(5)
      BHT(6) = TS(6)
      TBHC = TS(6)
      IF(IPRT.EQ.0) GO TO 410
408  WRITE(6,402)
      WRITE(6,403)(J, BHT(J), BZT(J), J = 1,6)
      WRITE(6,406) HTIN,TBHV, WBHC
      IF(I.EQ.0) GO TO 420
      IF(INI.EQ.0) GO TO 410
420  WRITE(6,411)
      WRITE(6,407)((AB(L,N),N=1,6),L=1,6)
      WRITE(6,407)((HHP(L,N),N = 1,6),L = 1,6)
      GO TO 410
400  WRITE(6,405)
      IERROR = 1
402  FORMAT(1H0,' STATE VARIABLES IN BRINE HEATER'/1H0,' VARIABLE      VA
      1LUE          ZTOT')
403  FORMAT(15,5X,0PF12.3,1PD12.5)
405  FORMAT(46H ERROR IN BRINE HEATER TIME DELAY, END PROGRAM)
406  FORMAT(1H0,' HTIN = ',1PD12.5,' TBHV = ',0PF8.3,' WBHC = ',F8.3)
407  FORMAT(1P6D13.5)
411  FORMAT(36HOCOEFFICIENTS OF BRINE HEATER MATRIX)
410  RETURN
      END

```

```

SUBROUTINE CHANGE(IQ)
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /GEN1/ TIME, DT, TNXT
  COMMON /EVAP5/ WSEA, WTUBE, FRSEA, FRTB, SMPMS, TPSUMP, TSEA,
1  TMAX, DEAIR, SSUMP, CSMP, CPSMP, CSEA
  COMMON /EVAP6/ SD(24), FD(54), CNG(8), TI(4), TITLE(20), EXPL(20)
1, ST, WMIN
  COMMON /EVAP8/ VENT, IATM
  COMMON /BEAT1/ TBHV, TBHO, CBHO, WBHC, PBHV, WBHI, WBHO, SQUAL
  COMMON /SGEN1/ HSG, TSG, PSG, WSG, WSGI, CSGW, PSGI
  COMMON /TRB1/ WVWO, WVWOBP, PRTBP, CNGBP, PART, CNGEL, VLVH, VLVBP
  COMMON /REAC3/ RHOP,RHO, AP, BP, AMPL, FREQ, TPRT
  DIMENSION KIND(20), CHN(20), TCN(20), BHN(20)

C      CHANGE WILL SET UP PERTURBATIONS TO SYSTEM
      IF(IQ.NE.0) GO TO 200

C.     FIRST READ IN DATA
      NCHANG = 0
      TNXT = TMAX
      DO 20 N =1,20
      READ(5,100) I,TM,AMT,BMT
100    FORMAT(I10,3E10.3)
      IF(I.EQ.0) GO TO 210
      KIND(N) =I
      CHN(N) = AMT
      TCN(N) = TM
      BHN(N) = BMT
      NCHANG = NCHANG +1
      TNXT = DMIN1(TNXT,TCN(N))
      IF(N.EQ.1) WRITE(6,202)
      WRITE(6,201) NCHANG, I, TM, AMT, BMT

```

```

202 FORMAT(1H0,' PERTURBATION ARRAY'//
1' NUMBER TYPE TIME      AMT      BMT')
201 FORMAT(2I5, 3F12.6)
20 CONTINUE
200 CONTINUE
210 IF(NCHANG.EQ.0) GO TO 80
C      NEXT CHECK TO SEE IF IT IS TIME FOR A CHANGE AND IF SO MAKE
C      THE CHANGE AND THE RE-ORDER PERTURBATIONS.
      J = 1
30 K = KIND(J)
      IF(TCN(J)-TIME)40,40,50
40 GO TO (1,2,3,4,5,6,7,8,9,10),K
1 WSEA = CHN(J)*WSEA
      WRITE(6,101) CHN(J),WSEA, TIME
101 FORMAT(36H0THE SEA WATER INTAKE WAS CHANGED BY,F7.5, 2HT0, D10.2,
118HPOUNDS PER HOUR AT, F10.2, 7HSECONDS)
      GO TO 60
2 TSEA = TSEA +CHN(J)
      WRITE(6,102) CHN(J),TSEA,TIME
102 FORMAT(48H0THE SEA WATER INTAKE TEMPERATURE WAS CHANGED BY,F7.2,13
1H DEGREES F TO,F7.2,13H DEGREES F AT,F10.2,7HSECONDS)
      GO TO 60
3 WSG = WSG*CHN(J)
      WRITE(6,103) CHN(J),WSG,TIME
103 FORMAT(1H0,' STEAM FLOW FROM STEAM GENERATOR CHANGED TO',OPF4.2,
1' OF ORIGINAL TO',F10.3,' LBS/SEC. AT',F5.1,' SECONDS')
      GO TO 60
4 FRTB = CHN(J)*FRTB
      WRITE(6,104) FRTB,TIME
104 FORMAT(34H0THE BRINE RECYCLE FLOW CHANGED TO,1PD12.5,3H AT,OPF7.2,
18H SECONDS)
      GO TO 60
5 FRSEA = FRSEA*CHN(J)
      WRITE(6,105) FRSEA,TIME
105 FORMAT(44H0THE SEAWATER NON REJECT FRACTION CHANGED TO,F7.5,3H AT,
1' F7.2,8H SECONDS)
      GO TO 60
6 CONTINUE
      AP = CHN(J)
      BP = BHN(J)
      WRITE(6,106) AP,BP
106 FORMAT(1H0,' STEP REACTIVITY CHANGE =',F10.6// ' RAMP REACTIVITY RA
1TE =',F10.6,' /SEC.')
      GO TO 60
7 CONTINUE
      VENT = CHN(J)
      WRITE(6,107) VENT,TIME
107 FORMAT(1H1,' TROUBLE WITH AIR EJECTORS'// ' FOULING IN STAGES NORMA
1LY BELOW ATMOSPHERIC PRESSURE CHANGED TO ',F8.4,' TIMES NORMAL AT
2',F8.2,' SECONDS')
      GO TO 60
8 CONTINUE
      PRTBP = PRTBP + CHN(J)
      CNGBP = BHN(J)
      WRITE(6,108) PRTBP, CNGBP
108 FORMAT(1H0,'STEP AND/OR RAMP CHANGE IN R-P TURBINE DEMAND LOAD'//
1' NEW LOAD CONDITION =',F8.5,' OF RATING CONDITION'//
2' RATE OF CHANGE OF DEMAND =',F8.5,'/SEC OF RATING VALUE')

```

```

GO TO 60
9 CONTINUE
PART = PART + CHN(J)
CMGEL = BHN(J)
WRITE(6,109) PART, CMGEL
109 FORMAT(1H0,' STEP AND/OR RAMP CHANGE IN LOAD DEMAND'//
1' NEW LOAD CONDITION =',F8.5,' OF RATING CONDITION'//
2' RATE OF CHANGE OF LOAD DEMAND =',F8.5,'/SEC OF RATING VALUE')
GO TO 60
10 CONTINUE
FD(47) = FD(47)*CHN(J)
WRITE(6,110) CHN(J), FD(47), TIME
110 FORMAT(1H0,' THE NUMBER OF MSF TRAINS IN OPERATION CHANGED BY A FA
1CTOR OF',F4.2,' TO',F7.5,' AT',F8.1,' SECONDS')
60 CONTINUE
NCHANG = NCHANG - 1
IF(NCHANG)65,65,71
71 IF(NCHANG-J)70,66,66
66 DO 62 L = J, NCHANG
KIND(L) = KIND(L+1)
CHN(L) = CHN(L+1)
TCN(L) = TCN(L+1)
BHN(L) = BHN(L+1)
62 CONTINUE
70 CONTINUE
TNXT = TMAX
DO 64 L=1,NCHANG
TNXT = DMIN1(TNXT,TCN(L))
64 CONTINUE
GO TO 72
50 CONTINUE
J = J+1
72 IF(J-NCHANG)30,30,80
65 TNXT = TMAX
80 RETURN
END

```

```

SUBROUTINE EMATS(IFRT,ISGE)
  CALCULATES C AND HP MATRICES
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /GEN1/ TIME, DTL,TNXT
  COMMON /MAT1/ AA(8,8,50), HPPP(8,8,50), AB(6,6), HHP(6,6)
  COMMON/MAT2/RAB(7,7),HPR(7,7), DTR
  COMMON /MAT3/ AFH(9,9,7), HPHH(9,9,7)
  COMMON /CVAP7/ NK, NMK, NTMSP, NN, LCT, NSTGS, TSECT, NARROW, NEWW
1, NRT, IND(16), IERROR, IUP, ISM(50), ICT(50)
  COMMON /SGEN3/ XSG(10), ZTSG(10), ASG(10,10), HPSG(10,10)
  DIMENSION A(10,10), C(10,10),HP(10,10), OPT(10,10), TQP(10)
  IN1 = IND(1)
  IN2 = IND(2)
  IN3 = IND(3)
  IN4 = IND(4)
  DT = DTL
  IF(IFRT.EQ.0) GO TO 1
  GO TO (7,29,4,15), IFRT
1 CONTINUE

```

```

      NE = 6
      DO 303 I = 1,NE
      DO 303 J = 1,NE
      A(I,J) = AB(I,J)
303 CONTINUE
      GO TO 6
4 NE = 7
      DO 306 I = 1,NE
      DO 306 J = 1,NE
      A(I,J) = RAB(I,J)
306 CONTINUE
      DT = DTR
      GO TO 6
29 NE = 9
      DO 308 I = 1,NE
      DO 308 J = 1,NE
      A(I,J) = AFH(I,J,ISGE)
308 CONTINUE
      GO TO 6
7 CONTINUE
      NE = NN
      DO 300 I = 1,NE
      DO 300 J = 1,NE
      A(I,J) = AA(I,J,ISGE)
300 CONTINUE
      GO TO 6
15 NE = 10
      DO 310 I = 1,NE
      DO 310 J = 1,NE
      A(I,J) = ASG(I,J)
310 CONTINUE
6 CONTINUE
      PE = 0.000
      JFK = 0
      P = 0.10-7
      OPTMP = 10.000
      ITMAX = 32
      IMAX = 1
      JMAX = 1
      AMIN = 1.000
      AMAX = 1.000
      DO 401 I = 1,NE
      DO 401 J = 1,NE
      IF (AMAX - DABS(A(I,J))) 402,401,401
402 AMAX = DABS(A(I,J))
      IMAX = I
      JMAX = J
401 CONTINUE
      IMIN = IMAX
      JMIN = JMAX
      DO 409 I = 1,NE
      DO 409 J = 1,NE
      IF (AMAX.GT.1.0D12*DABS(A(I,J))) A(I,J) = 0.000
      IF (A(I,J)) 407,409,407
407 IF (DABS(A(I,J)) - AMIN) 408,409,409
408 AMIN = DABS(A(I,J))
      IMIN = I
      JMIN = J

```

```

409 CONTINUE
      RATIO=AMAX/AMIN
C     AMIN = MINIMUM NON ZERO ELEMENT
      ISTORE=0
      ADT=AMAX*DT
      DO 403 I=1,13
        IF(1.000 - ADT) 413,404,404
413   ISTORE=ISTORE+1
403   ADT=ADT*0.500
404   T=ADT/AMAX
C     COMPUTATION INTERVAL T IS HALVED ISTORE
C     TIMES(12=MAX.) SO MAX. ELEMENT IN A*T
C     IS LESS THAN 1.0
C
      IF(ISTORE-12)8,410,410
410   WRITE(6,411)
411   FORMAT('0A*DT STILL GREATER THAN ALLOWABLE, AFTER 12 HALVINGS.')
```

RETURN

```

      8 DO 9 I=1,NE
        DO 9 J=1,NE
          9 C(I,J)=0.000
C
      DO 10 I=1,NE
        10 C(I,I)=1.000
C
      48 DO 49 I=1,NE
        DO 49 J=1,NE
          49 HP(I,J)=0.000
C
      DO 50 I=1,NE
        50 HP(I,I)=T
C
      DO 11 I=1,NE
        DO 11 J=1,NE
          11 OPT(I,J)=C(I,J)
C
C     NOW FORM THE MATRIX EXPONENTIAL AND ((M-I)*A INVERSE) THRU STATMNT 16
      AL=1.000
C
      12 DO 16 KL=1,ITMAX
C
        KLM=KL
        ALL=T/AL
        AL=AL+1.000
        TALLL=T/AL
C
C     FORM M AND ((M-I)*A INVERSE) TERMS AND SUM THEM
C
      DO 18 I=1,NE
C
        DO 13 J=1,NE
          TQP(J)=0.000
          DO 13 KX=1,NE
            13 TQP(J)=TQP(J)+OPT(I,KX)*A(KX,J)
C
          DO 18 J=1,NE
            18 OPT(I,J)=TQP(J)*ALL
C

```



```

C      DO 44 I=1,NE
        DO 44 J=1,NE
    44 C(I,J)=C(I,J)+QPT(I,J)
C
    45 IF(ITMAX-KL)47,47,145
    145 DO 46 I=1,NE
        DO 46 J=1,NE
    46 HP(I,J)=HP(I,J)+QPT(I,J)*TALLL
C
C FIND MAX ABS ELEMENT IN QPT AND CALL IT PMK
C
    47 PMK =DABS(QPT(IMAX,JMAX))
      IF(QPTMP-PMK) 83,83,502
    502 IF(PMK-P) 406,406,16
    406 DO 14 I=1,NE
        DO 14 J=1,NE
    14 PMK=DMAX1(PMK,DABS(QPT(I,J)))
      IF(PMK-P)17,17,16
    17 IF(PE - 2.000*PMK) 16,21,21
    16 PE=PMK
    21 IF(ITMAX - 1) 20,20,538
    538 IF(KLM - ITMAX) 414,83,83
C
    83 T = T*0.5D0
      JFK = JFK + 1
      IF(JFK - 7) 8,304,304
    304 WRITE (6,305) PMK
    305 FORMAT(32H07 TRIES AT HALVING T N.G., PMK=,F12.6)
      IERROR = 1
      RETURN
    414 ISTORE = ISTORE + JFK
      IF(IN4.EQ.0) GO TO 32
      IF(TIME.EQ.0.0) GO TO 31
      IF(IN1) 31,32,31
    31 WRITE(6,405) IMAX,JMAX,A(IMAX,JMAX),ADT,T,
      1IMIN,JMIN,A(IMIN,JMIN),RATIO
    405 FORMAT(31HOMAX.COEFF. MATRIX ELEMENT =A(,I2,1H,,I2,3H) =,
      1D15.4/13H MAX. A*DT = ,F12.8,2X,14HWITH DELTA T =,F15.8/
      230HMINIMUM NON-ZERO ELEMENT = A(,I2,1H,,I2,3H) =,D15.4/
      318H RATIO AMAX/AMIN =,D15.4)
      WRITE(6,200) KLM
    200 FORMAT(44HNO. OF TERMS IN SERIES APPROX. OF MATEXP = ,I2)
C
      IF(ISTOR) 20,20,416
    416 WRITE(6,415)ISTOR
    415 FORMAT(26H0TOTAL NO. OF T HALVINGS =,I3)
      GO TO 33
    32 IF(ISTOR) 20,20,33
    33 DO 417 KR=1,ISTOR
    419 DO 420 I=1,NE
        DO 421 J=1,NE
          TOP(J)=0.0D0
          DO 421 KX=1,NE
    421 TOP(J)=TOP(J)+HP(I,KX)*C(KX,J)
          DO 420 J=1,NE
    420 HP(I,J)=TOP(J)+HP(I,J)
C

```

```

418 DO 430 I=1,NE
      DO 430 J=1,NE
430 QPT(I,J)=0.000
      DO 431 I=1,NE
        DO 431 J=1,NE
          DO 431 KX=1,NE
431 QPT(I,J)=QPT(I,J)+C(I,KX)*C(KX,J)
      DO 432 I=1,NE
        DO 432 J=1,NE
432 C(I,J)=QPT(I,J)
417 T = 2.000*T

```

```

C
C      C(I,J) IS THE MATRIX EXPONENTIAL M=EXP(A*T)
C      AND HP(I,J) IS THE ((M-I)*A INVERSE) MATRIX
C

```

```

20 IF(IFRT.EQ.0) GO TO 5
   GO TO (2,28,3,30), IFRT
2 CONTINUE
   DO 301 I = 1,NE
     DO 301 J=1,NE
       HPPP(I,J,ISGE)=HP(I,J)
301 CONTINUE
   RETURN
28 DO 309 I = 1,NE
     DO 309 J = 1,NE
       HPPH(I,J,ISGE) = HP(I,J)
309 CONTINUE
   RETURN
5 DO 302 I = 1,NE
     DO 302 J = 1,NE
       HHP(I,J) = HP(I,J)
302 CONTINUE
   RETURN
3 DO 307 I = 1,NE
     DO 307 J = 1,NE
       HPR(I,J) = HP(I,J)
307 CONTINUE
   RETURN
30 DO 311 I = 1,NE
     DO 311 J = 1,NE
       HPSG(I,J) = HP(I,J)
311 CONTINUE
   RETURN
END

```

34400217

34400219

```

SUBROUTINE RESTR
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /EVAP1/ X(8,50), XIC(8,50), ZT(8,50), Y(8,50), WMAT(30,50)
  COMMON /EVAP2/ PDC(50), TMASS(50), AO(50), AI(50),
  ICTUBMS(50), B5(50), FPF(50), FB1(50), FB2(50), FB3(50), FTB(50),
  2FP1(50), FP2(50), FWP(50), FOUL(50), FBF(50), FUDHI(50), FUDHO(50)
  COMMON /EVAP3/ WTBO(50), WTBI(50), WTBF(50), WPI(50), WC(50),
  1WPF(50), WPO(50), CPB(50), CPTB(50), CPP(50), CCP(50), CONC(50),
  2CNCT(50), SMTT(50), SMTB(50)
  COMMON /EVAP4/ TTI(50), TTBI(50), TPI(50), BL(50), PL(50), BLI(50)
  1, PLI(50), DMTB(50), DMP(50)

```

```

COMMON /EVAP5/ WSEA, WTUBE, FRSEA, FRTB, SMPMS, TPSUMP, TSEA,
1 TMAX, DEAIR, SSUMP, CSMP, CPSMP, CSEA
COMMON /EVAP6/ SD(24), FD(54), CNG(8), TI(4), TITLE(20), EXPL(20)
1, ST, WMIN
COMMON /EVAP7/ NK, NNK, NTMSP, NE, ICT, NSTGS, ISECT, NARROW, NEWW
1, ISF, IND(16), IERROR, IUP, ISM(50), ICT(50)
COMMON /EVAP8/ VENT, IATM
COMMON /EVAP9/ SSD(50)
COMMON /FEED1/ VFH(8), WFW(8), HFWC(7), WDC(8), HDC(7), HFWC(7),
1 TFHV(8), TFEI(7), TFWO(8), TFWI(8), TDCO(7), FHSO(7), WFHS(8),
2 HFHS(8), TDCI(8), TFEQ(7), PFHV(8)
COMMON /FEED2/ IDC(7), NFH
COMMON /TRB1/ WVWO, WVWOBP, PRTBP, CNGBP, PART, CNGEL, VLVH, VLVBP
COMMON /TRBH/ WHP(12), TH(12), HH(12), PH(12)
COMMON /TRBL/ WLP(12), TL(12), HL(12), PLP(12)
COMMON /TRBB/ TB(6), HB(6), PB(6), WBPT, WFPT, TFP, HFP, PFP
COMMON /RHTR/ HRHO, TRHO, VMAS, HOLD, VOLD, CONM
COMMON /FDAT1/ XFH(9,7), UADL(1), UAEV(7)
COMMON /FDAT2/ B5FH(7), PDCFH(7), FWC(7), FMD(7), FWEM(7)
COMMON /SGEN1/ HSG, TSG, PSG, WSG, WSGI, CSGH, PSGI
COMMON /SGEN2/ HSGI, TSGI, HSGPI, HSGPO
COMMON /BEAT1/ TBHV, TBHO, CBHO, WBHC, PBHV, WBHI, WBHO, SQUAL
COMMON /BEAT2/ TBHC, HBI, BHT(6)
COMMON /REAC2/ WRP, TFA, TCA, TCRI, PWR, FCM, CCM, CLM, CUM, TCRO
COMMON /REAC1/ XR(7), ZRT(7), RFD(30)
COMMON /STEAM1/ TSGPO, TSGPI
COMMON /SGEN3/ XSG(10), ZTSG(10), ASG(10,10), HPSG(10,10)
COMMON /FDAT3/ B5SG, PDCSG, FCON, UAB, UANB
COMMON /FDAT4/ UAFC, UAOF
WRITE(6,201)
201 FORMAT(1H0, ' ENTERING RESTRT, PREPARE FOR RESTART CARDS')
A2 = 3.603*FD(14)
FD(20) = TBHV
FD(21) = A2*WTUBE
FD(23) = TBHO
FD(24) = X(1,1)
FD(32) = BHT(2)
FD(33) = BHT(3)
FD(34) = BHT(4)
FD(38) = BHT(5)
FD(39) = BHT(6)
FD(40) = WBHC
FD(15) = 0.000*HBI
FD(53) = VENT
IF(IND(11).EQ.0) GO TO 8
FD(22) = A2*WSEA
FD(31) = X(5,NSTGS)
FD(35) = X(1,NK+1)
FD(36) = FRTD*A2
FD(45) = SMPMS/WTUBE
8 CONTINUE
PUNCH 101, (FD(I), I = 1,54)
IF(IND(11).EQ.0) GO TO 9
DO 10 I = 1, NSTGS
DO 20 J = 1, 24
SD(J) = 0.0
20 CONTINUE
SD(1) = A2*WTBO(I)

```

```

SD(2) = A2*WPF(I)
SD(3) = X(8,I)
SD(4) = A2*WTBF(I)
SD(5) = A2*WPO(I)
SD(6) = X(5,I)
SD(9) = SSD(I)
SD(10) = A2*WC(I)
SD(13) = B5(I)*X(4,I) + PDC(I)
SD(14) = CNCT(I)
DS13 = SD(13)
SD(7) = TSAT(DS13)
IF(I.EQ.NSTGS) GO TO 3
IF(I-NK) 1,2,1
1 SD(8) = X(1,I+1)
GO TO 4
2 SD(8) = TPSUMP
GO TO 4
3 SD(8) = TSEA
4 CONTINUE
SD(11) = X(6,I)
IF(I-NK) 6,6,7
6 ISECT = 2
WT = WTUBE
GO TO 5
7 ISECT = 3
WT = WSEA
5 SD(12) = TMASS(I)*A2*FD(ISECT + 24)/FD(ISECT + 19)
CN1 = 0.000
CN2 = 1.000
HI = HIN(X(2,I),WT,CN2)
SD(15) = HI*3.603*FUDHI(I)
SD(16) = FD(14)*AD(I)
SD(17) = X(4,I)
SD(18) = CONC(I)
SD(19) = BL(I)
SD(20) = PL(I)
SD(21) = X(7,I)
SD(22) = X(1,I)
SD(23) = X(2,I)
SD(24) = X(3,I)
PUNCH 101,(SD(J), J = 1,24)
10 CONTINUE
9 CONTINUE
PUNCH 108, TRHO, VMAS, HOLD, VOLD, CONM
108 FORMAT(1P5E12.5)
PUNCH 100, VL VH, VL VAP, WVWN, WVWNR
PUNCH 103, PART,PRTBP
PUNCH 101, WHP
PUNCH 101, TH
PUNCH 101, HH
PUNCH 101, PH
PUNCH 101, WLP
PUNCH 101, TL
PUNCH 101, HL
PUNCH 101, PLP
PUNCH 101, TB
PUNCH 101, HB
PUNCH 101, PB

```

```

PUNCH 101, WRPT, WRHO, WFPT, TFP, HFP, WFP
PUNCH 104, (IDC(I), I = 1,7)
DO 11 I = 1,7
PUNCH 102, ((XFH(J,I), J = 1,9), WFW(I), WDC(I), UADC(I), UAFV(I),
1 TFWI(I), TDCI(I), PFHV(I))
PUNCH 105, (FWEM(I), FWCM(I), FHDIM(I), B5FH(I), PDCFH(I))
11 CONTINUE
RFD(18) = WRP
RFD(19) = TCRI
RFD(17) = TSGPI
PUNCH 101, (RFD(I), I = 1,30)
PUNCH 108, ((XR(I), I = 1,7), UAFG, UANI)
PUNCH 101, HSG, HSGI, WSG, WSGI, TSG, PSG
PUNCH 108, (XSG(I), I = 1,10)
PUNCH 108, B5SG, PDCSG, FCON, UAB, UANB
100 CONTINUE
101 FORMAT(1P4E12.5)
102 FORMAT (8F10.4)
103 FORMAT(1P2E12.5)
104 FORMAT(7I5,4X)
105 FORMAT(1P5E12.5,20X)
106 FORMAT(5E12.5)
107 FORMAT(7I5)
109 FORMAT(1P4E12.5)
RETURN
END

```

```

FUNCTION CP(T,C)
IMPLICIT REAL*8(A-H,O-Z)
IF(C.LT.0.1D-6) GO TO 1
CP = 1.0057733D0 - .13280442D-3*T - 1.5344907D0*C
2 + .3909715D-2*T*C + .65092605D-6*T**2 + 2.495546D0
3 *C**2 - .11156771D-1*T*C**2 - .88941836D-5*T**2
4 *C + .3598702D-4*T**2*C**2
GO TO 2
1 CP = 1.0057733D0 - 0.13280442D-3*T + 0.65092605D-6*T**2
2 CONTINUE
RETURN
END

```

```

FUNCTION HFG(T)
IMPLICIT REAL*8(A-H,O-Z)
HG = 1.0632968D3 + 0.38934495D0*T + 0.51632953D-3*T**2
1 - .26817564D-5*T**3 + .42653473D-8*T**4
2 - .37338556D-11*T**5
HF = -32.804438D0 + 1.0302232D0*T - 0.34513579D-3*T**2
1 + .15228375D-5*T**3 - .27812209D-8*T**4
2 + .24067265D-11*T**5
HFG = HG - HF
RETURN
END

```

40250203

40250103

40250110

40250207

40250208

```

FUNCTION DEN(T,C)
  IMPLICIT REAL*8(A-H,O-Z)
  DEN = 62.71775300-.32152986D-2*T+.44315006D2*C
1-.11647394D-1*C*T-.48932777D-4*T**2+.16449945D2*C**2
2+.13761984D-1*C**2*T+.34231326D-5*C*T**2
3-.92275301D-4*C**2*T**2
  RETURN
END

```

```

FUNCTION DVPSD(D)
  IMPLICIT REAL*8(A-H,O-Z)
  ALD=DL OG(D)
  DVPSD = (1.00/D)*(0.98934D0 - 0.078413D0*ALD -0.0275069D0*ALD*ALD
1 -0.004072245D0*ALD**3 - 2.21621D-4*ALD**4)
2 *DEXP(6.140092D0+.98934D0*ALD-.039267D0*ALD*ALD
3-.916895D-2*ALD**3-.101806D-2*ALD**4
4-.443242D-4*ALD**5)
  RETURN
END

```

```

FUNCTION DSTVP(P)
  IMPLICIT REAL*8(A-H,O-Z)
  X = DL OG(P)/2.3026D0
  T1 = 77.052D0 + 23.904D0*X + 6.1686D0*X*X + 1.68284D0*X**3
1 - 0.342D0*X**4 + 0.37518D0*X**5 - 0.04613D0*X**6
  DSTVP = (1.00/(P*2.3026D0))*T1
  RETURN
END

```

```

FUNCTION HIN(T,W,FUD)
  IMPLICIT REAL*8(A-H,O-Z)
  C CALCULATES TUBE INSIDE H, BTU/(SEC.FT**2.DEG F)
  C T=WATER TEMP, DEG F, W=MASS FLOW RATE, LB/SEC, FUD=FUDGE FACTOR
  HIN = FUD*(1.000 + 0.011D0*T)*W**0.8D0
  RETURN
END

```

```

FUNCTION HOUT(TT,TS,FL,FUD)
  IMPLICIT REAL*8(A-H,O-Z)
  C CALCULATES TUBE OUTSIDE H, BTU/(SEC.FT**2.DEG F) WITH FOULING
  C TT=TUBE TEMP, TS= STEAM TEMP, DEG F, FL=FOULING R, SEC.FT2.F/BTU
  X9 = 2.8026D3 + 23.688D0*TS - 0.03182D0*TS*TS - 8.93D-6*TS**3
  DELT=TS-TT
  IF(DELT.LT.1.00) DELT = 1.000
  S1 = DELT**.25D0
  S2 = FUD*X9
  HOUT = S2/(S1 + FL*S2)
  RETURN
END

```

```

FUNCTION WEQ(P1,P2,V),CK)
IMPLICIT REAL*8(A-H,O-Z)
THIS FUNCTION CONTAINS THE VARIABLE PART OF THE FLOW EQUATION
CHECK FOR CRITICAL FLOW
K1 = CK/(CK - 1.00)
RCRIT = (2.00/(CK + 1.00))**K1
R = DMAX1(P2/P1,RCRIT)
WEQ = DSQRT(K1*(P1/V1)*(R**((2.00/CK) - R**((CK + 1.00)/CK)))
RETURN
END

```

```

SUBROUTINE PFVH(H,VS,P)
IMPLICIT REAL*8(A-H,O-Z)
DO 10 J = 1,10
TP = TVAPH(P,H)
VC = VVAP(TP,P)
IF(DABS(VC - VS).LE.0.1D-4) GO TO 20
P = P*VC/VS
10 CONTINUE
WRITE(6,101) VC,VS
101 FORMAT(1H1,' NON-CONVERGE IN PFVH, VC =',F8.5,'VS =',F8.5)
20 CONTINUE
RETURN
END

```

```

FUNCTION CPW(TF)
IMPLICIT REAL*8(A-H,O-Z)
USES FORMULATION IN HLIQ(TF) TO GET CPW = DH/DT
T = TF
IF(T-360.00) 20,20,10
10 CPW = 10.673802 - 8.5507672D-2*T + 2.823732D-4*T*T - 4.1261428D-7*
1T**3 + 2.280123D-10*T**4
RETURN
20 CPW = (((4.81751575D-12*T - 2.9447512D-9)*T + 1.45661508D-6)*T -
1 2.3033992D-4)*T + 1.0088084
RETURN
END

```

```

FUNCTION FIG8(PX)
IMPLICIT REAL*8(A-H,K-Z)
DIMENSION A(5)
THE FOLLOWING IS THE LIST OF PROGRAM CONSTANTS
DATA A/0.1304D1,0.1973D2,-0.1118D2,-0.9587D2,0.3148D3/,I//
TEST FOR PX .GE.3.0 .AND..LE.100.0 .OR.GE.100.0
IF (PX - 3.0) 30,10,10
10 IF (PX - 100.0) 35,15,15
15 IF (PX - 100.0) 35,25,25
25 FIG8 = 1.5
RETURN
30 PRINT 31,PX
31 FORMAT(5H0PX =,D25.16,38H IS OUTSIDE SPECIFIED LIMITS FOR FIG8)
35 X = 1.0/PX

```

F8	100
F8	105
F8	110
F8	115
F8	120
F8	125
F8	130
F8	135
F8	140
F8	145
F8	150
F8	155
F8	160
F8	165

CALL UNIPOL (A,X,Z,I)	F8	170
FIGR = Z	F8	175
RETURN	F8	180
END	F8	185
FUNCTION FIG10(P)	F10	105
IMPLICIT REAL*8(A-H,K-Z)	F10	110
DIMENSION A(5)	F10	115
C THE FOLLOWING IS THE LIST OF PROGRAM CONSTANTS	F10	120
DATA A/0.1383D1,0.2328D2,0.8501D3,-0.5161D4,0.21358D5/,I/4/	F10	125
C TEST FOR P .LE.5.7 .OR .GT.5.7	F10	130
IF (P - 5.7 - (1.D-5)) 15,15,20	F10	135
15 FIG10 = 24.0	F10	140
RETURN	F10	145
20 X = 1.0/P	F10	150
CALL UNIPOL (A,X,Z,I)	F10	155
FIG10 = Z	F10	160
RETURN	F10	165
END	F10	165
SUBROUTINE UNIPOL (A,X,Z,I)	UNIP	100
IMPLICIT REAL*8(A-H,O-Z)	UNIP	105
DIMENSION A(10)	UNIP	110
Z = 0.0	UNIP	115
I1 = I + 2	UNIP	120
DO 1 N=1,I	UNIP	125
Z = Z*X	UNIP	130
1 Z = Z + A(I1-N)	UNIP	135
Z = Z*X + A(1)	UNIP	140
RETURN	UNIP	145
END	UNIP	150
FUNCTION TSAT(P)	TSAT	100
IMPLICIT REAL*8(A-H,O-Z)	TSAT	101
CALCULATES SATURATION TEMPERATURE (DEG F) OF WATER, .178PSIA TO CRITICAL	TSAT	102
C KEENAN + KEYES, 1951, P.14, EQ.11 + 12	TSAT	105
T=500.	TSAT	110
CORR=3.2	TSAT	115
FOP=DLUG(3206.37858/P)/2.30258509	TSAT	120
GO TO 1	TSAT	125
C 4 TEMP BELOW 140C = 284F	TSAT	130
4 CORR=((1.1702379D-8*TK*TK+5.86826D-3)*TK+3.2437814)	TSAT	135
1 / (2.1878462D-3*TK+1.)	TSAT	136
1 TS=1165.09/(FOP/CORR+1.)	TSAT	140
IF(DABS(TS-T)-.01)10,10,2	TSAT	145
2 T=TS	TSAT	150
TK=(1165.09-T)/1.8	TSAT	155
IF(P-52.418)4,4,3	TSAT	160
C 3 TEMP ABOVE 140C = 284 F	TSAT	165
3 CORR=((16.56444D-11*TK+7.515484D-9)*TK*TK+4.14113D-2)*TK	TSAT	170
1 +3.346313)/(1.+1.3794481D-2*TK)	TSAT	171
GO TO 1	TSAT	175
10 TSAT=TS-459.69	TSAT	180
RETURN	TSAT	181
END	TSAT	185


```

      FUNCTION PSAT(TF)
      IMPLICIT REAL*8(A-H,O-Z)
CALCULATES SATURATION PRESSURE IN PSIA AT TEMPERATURE TF
C   DEGREES FAHRENHEIT      50 TO 705.4
C   KEENAN AND KEYES PAGE 14
      T=(TF+459.69)/1.8
      X=647.27-T
      IF(X-234.11)20,10,10
C 10 TEMP BELOW 140 CENT
      A=((1.1702379D-8**X+5.86826D-3)*X+3.2437814)
      1 / (2.1878462D-3*X+1.)
      GO TO 30
C 20 TEMP ABOVE 140 CENT
      A=((6.56444D-11*X+7.515484D-9)*X+4.14113D-2)*X+3.346313)
      1 / (1.3794481D-2*X+1.)
      30 PSAT=DEXP(-2.30258509*A*X/T)*3206.37858
      RETURN
      END

```

```

PSAT 100
PSAT 101
PSAT 102
PSAT 105
PSAT 110
PSAT 115
PSAT 120
PSAT 125
PSAT 130
PSAT 135
PSAT 136
PSAT 140
PSAT 145
PSAT 150
PSAT 151
PSAT 155
PSAT 160
PSAT 165

```

```

      FUNCTION TLIQ(H)
      IMPLICIT REAL*8(A-H,O-Z)
CALCULATES TEMPERATURE (F) OF SATURATED LIQUID
C   AS FUNCTION OF ENTHALPY (BTU/LB) OF
C   SATURATED LIQUID
C   PROCEDURE SUGGESTED IN WESTINGHOUSE,
C   TRANS ASME V80, NO4, MAY, 1958.
C   CHECK TO SEE IF INPUT ARGUMENTS ARE WITHIN SPECIFIED LIMITS
      IF(H-18.07)21,3,25
      IF DATA OUTSIDE LIMITS PRINT MESSAGE AND
C   CONTINUE EXECUTION
      21 PRINT 13,H
      25 IF(H-870.0)2,4,1
      1 PRINT 13,H
      2 IF(H-400.0)3,3,4
      3 T0=H+35.0
      GO TO 20
      4 T0=-(H*H)/1022.49+1.855*H-172.25
C   BEGIN PROCESS WHEREBY H IS ROUNDED ABOVE
C   BY H0 AND BELOW BY H1
      20 T1=T0-3.0
      H0=HLIQ(T0)
      H1=HLIQ(T1)
      IF(H-H0)5,6,7
      7 T0=T0+5.0
      GO TO 20
      6 TLIQH=T0
      RETURN
      5 IF(H1-H)8,9,10
      9 TLIQH=T1
      RETURN
      10 T0=T0-1
      GO TO 20
C   BEGIN HALVING PROCEDURE TO FIND DESIRED T
      8 DO 11 I=1,50
      TAVG=(T1+T0)/2.0
      HT=HLIQ(TAVG)

```

```

TLIQ 0
TLIQ 1
TLIQ 2
TLIQ 10
TLIQ 20
TLIQ 30
TLIQ 40
TLIQ 50
TLIQ 60
TLIQ 70
TLIQ 80
TLIQ 90
TLIQ 100
TLIQ 110
TLIQ 120
TLIQ 130
TLIQ 140
TLIQ 150
TLIQ 160
TLIQ 170
TLIQ 180
TLIQ 190
TLIQ 200
TLIQ 210
TLIQ 220
TLIQ 230
TLIQ 240
TLIQ 250
TLIQ 260
TLIQ 270
TLIQ 280
TLIQ 290
TLIQ 300
TLIQ 310
TLIQ 320
TLIQ 330
TLIQ 340

```

```

D=H-HT
DT=DARS(D)-1.D-2
IF(D)15,16,17
16 TLIOH=TAVG
RETURN
15 TO=TAVG
GO TO 18
17 T1=TAVG
18 IF(DT)19,19,11
19 TLIOH=TAVG
RETURN
11 CONTINUE
PRINT 14
TLIOH=TAVG
13 FORMAT(5H0H = ,D20.11,5X,24HIS A DATA ERROR IN TLIOH)
14 FORMAT(40H0TLIOH DID NOT CONVERGE IN 50 ITERATIONS)
RETURN
END

```

TLIO 350
TLIO 360
TLIO 370
TLIO 380
TLIO 390
TLIO 400
TLIO 410
TLIO 420
TLIO 430
TLIO 440
TLIO 450
TLIO 460
TLIO 470
TLIO 480
TLIO 500
TLIO 510
TLIO 511
TLIO 520

```

FUNCTION HLIO(TF)
IMPLICIT REAL*8(A-H,N-Z)
CALCULATES ENTHALPY (BTU/LBM) OF SATURATED WATER FROM 50 TO 600 DEG F.
C WESTINGHOUSE ALGORITHM, TRANS ASME V80, NO4, MAY, 1958, P.959
T=TF
IF(T-360.)20,20,10
C 10 TEMP ABOVE 360 F
10 HLIO=(((4.560246D-11*T-1.0315357D-7)*T+9.41244D-5)*T-4.2753836D-2
1 )*T+10.673802)*T-904.11706
RETURN
C 20 TEMP BELOW 360 F
20 HLIO=(((9.6350315D-13*T-7.3618778D-10)*T+4.8553836D-7)*T
1 -1.1516996D-4)*T+1.0088084)*T-32.179105
RETURN
END

```

HLIO 100
HLIO 101
HLIO 102
HLIO 105
HLIO 110
HLIO 115
HLIO 120
HLIO 125
HLIO 126
HLIO 130
HLIO 135
HLIO 140
HLIO 141
HLIO 145
HLIO 150

```

FUNCTION SLIO(T)
IMPLICIT REAL*8(A-H,N-Z)
CALCULATES ENTROPY (BTU/LBM/F) OF SATURATED WATER FROM 50F TO 670F
C WESTINGHOUSE ALGORITHM, TRANS ASME V80, NO4, MAY, 1958, P.959
COORDINATES SHIFTED FROM ORIGINAL OF TO 360F OR 560F
IF(T-450.)20,20,10
C 10 TEMP ABOVE 450
10 TB=(T-560.)/110.
SLIO=((((3.4265601D-3*TB+3.9726934D-3)*TB-2.1403201D-3)*TB
1 -1.0108801D-3)*TB+5.7828937D-3)*TB+7.5137702D-3)*TB
2 +.13690825)*TB+.76209767
RETURN
C 20 TEMP BELOW 450
20 TB=(T-360.)/310.
SLIO=((((0.01234655*TB+.012035893)*TB-3.670358D-3)*TB-6.072333D-3
1 )*TB+.034251697)*TB-.045979941)*TB+.39679646)*TB+.51575516
RETURN
END

```

SLIO 100
SLIO 101
SLIO 102
SLIO 105
SLIO 106
SLIO 110
SLIO 115
SLIO 120
SLIO 125
SLIO 126
SLIO 127
SLIO 130
SLIO 135
SLIO 140
SLIO 145
SLIO 146
SLIO 150
SLIO 155

```

FUNCTION VLIQ(TF)
  IMPLICIT REAL*8(A-H,O-Z)
  CALCULATES SPECIFIC VOLUME (CU.FT./LBM) OF SATURATED WATER FROM 32F TO
C  KEENAN + KEYES, 1951, P.21, EQ.18
  T=374.11-(TF-32.)/1.8
  CRT=T**.3333333333333
  DEN=CRT*.1342489-3.946263D-3*T+1.
  VLIQ=((T**3*7.48908D-13-1.203374D-3)*T-.3151548*CRT+3.1975)
1  *.0160185/DEN
  RETURN
END

```

VLIQ 100
VLIQ 101
VLIQ 102
VLIQ 105
VLIQ 110
VLIQ 115
VLIQ 120
VLIQ 125
VLIQ 126
VLIQ 130
VLIQ 135

```

FUNCTION HVAL(TF,PF)
  IMPLICIT REAL*8(A-H,O-Z)
  CALCULATES ENTHALPY OF STEAM REFERRED TO WATER AT 32F AND .08854 PSIA
C  KEENAN + KEYES, 1951, PP. 15-16, EQ 13-15
  COMMON /H2OEN/ TOLD,POLD,TAU,TAU2,B0,Q,Q3,Q12,DB0,F0,F1,G1,F2,G2
1  F3,G3
  P=PF/14.6959
  T=(TF+459.69)/1.8
  IF(T-TOLD)33,32,33
32 IF(P-POLD)34,220,34
33 TOLD=T
34 POLD=P
  TAU=1./T
  TAU2=TAU*TAU
  TAU3=TAU2*TAU
  TAU12=TAU3**4
  EXTAU=- (10.*(80870.*TAU2))
  B0=EXTAU*TAU**2 641.62+1.89
  Q=B0*TAU
  Q2=Q*Q
  Q3=Q2*Q
  Q12=Q3**4
  DB0=(372420.11*TAU2+1.)*EXTAU*2641.62
  F0=DB0*TAU+D0
  DG1=-3.2492D5*TAU+82.546
  G1=(-1.6246D5*TAU+82.546)*TAU
  F1=(G1*F0**2.+Q*DG1)*D0
  DG2=-2.5394D5*TAU
  G2=-1.2697D5*TAU2+.21828
  F2=(G2*F0**4.+Q*DG2)*Q3
  DG3=+1.62432D66*TAU12*TAU12*T
  G3=+6.768D64*TAU12*TAU12-3.635D-4
C  SIGNS OF G3 + DG3 REVERSED FROM K+K      AVNTDS - IN F3
  F3=(G3*F0**13.+Q*DG3)*Q12
220 DHCT=((P**Q*F3/3,25+F2)*Q.5*P*P+F1)*.5*P*F0)*P**4.3557D 24.2150633
C  DHCT=CHANGE IN H AT CONSTANT T AS PRESSURE GOES FROM ZERO TO PF PLHVAL 260
C  CHANGE IN H AT 32F AS P=.08854 GOES TO P=0      HVAL 265
  DHZP=((3.7783D-4*T+1.472)*T+DLOG(T)*47.8365)*Q.43-300.420108      HVAL 270
C  DHZP=CHANGE IN H AT ZERO PRESSURE AS TEMP. GOES FROM 32F TO TF      HVAL 275
C  INTEGRAL (CPO*DT) EVALUATED AT 32F = 300.420108      HVAL 280
  HVAL=DHZP+DHCT+1075.8      HVAL 285
C  H OF EVAPORATION AT 32F + .08854PSIA = 1075.8 BTU/LBM      HVAL 290
  RETURN      HVAL 295
END      HVAL 300

```

```

FUNCTION SVAP(TF,PF)                                SVAP 100
IMPLICIT REAL*8(A-H,O-Z)                            SVAP 101
CALCULATES ENTROPY OF STEAM REFERRED TO WATER AT 32F AND .08854 PSIA SVAP 102
C KEENAN + KEYES, 1951, PP. 15-19, EQ. 13, 15, 16A SVAP 105
COMMON /H2OQEN/ TOLD,POLD,TAU,TAU2,B0,Q,Q3,Q12,DB0,F0,F1,G1,F2,G2 SVAP 110
1,F3,G3 SVAP 111
P=PF/14.6959 SVAP 115
T=(TF+459.69)/1.8 SVAP 120
IF(T-TOLD)33,32,33 SVAP 125
32 IF(P-POLD)34,220,34 SVAP 130
33 TOLN=T SVAP 135
34 POLD=P SVAP 140
TAU=1./T SVAP 145
TAU2=TAU*TAU SVAP 150
TAU3=TAU2*TAU SVAP 155
TAU12=TAU3**4 SVAP 160
EXTAU=-(10.*(80870.*TAU2)) SVAP 165
B0=EXTAU*TAU*2641.62+1.89 SVAP 170
Q=B0*TAU SVAP 175
Q2=Q*Q SVAP 180
Q3=Q2*Q SVAP 185
Q12=Q3**4 SVAP 190
DB0=(372420.11*TAU2+1.)*EXTAU*2641.62 SVAP 195
F0=DB0*TAU+R0 SVAP 200
DG1=-3.249205*TAU+82.546 SVAP 205
G1=(-1.624605*TAU+82.546)*TAU SVAP 210
F1=(G1*F0*2.+Q*DG1)*Q SVAP 215
DG2=-2.539405*TAU SVAP 220
G2=-1.269705*TAU2+.21828 SVAP 225
F2=(G2*F0*4.+Q*DG2)*Q3 SVAP 230
DG3=+1.62432066*TAU12*TAU12*T SVAP 235
G3=+6.768064*TAU12*TAU12-3.6350-4 SVAP 240
C SIGNS OF G3 + DG3 REVERSED FROM K+K AVOIDS - IN F3 SVAP 245
F3=(G3*F0*13.+Q*DG3)*Q12 SVAP 250
220 EMO=DB0*TAU2 SVAP 255
EM1=(-B0*Q*G1+F1)*TAU*.5 SVAP 260
EM2=(-B0*Q3*G2+F2)*TAU*.25 SVAP 265
EM3=(-B0*Q12*G3+F3)*TAU/13. SVAP 270
DSC T=((((EM3*P**9+EM2)*P*P+EM1)*P+EMO)*P-DLOG(P)*4.55504)* SVAP 275
1 2.41983330-2 -.56307532 SVAP 276
C DSC T=CHANGE IN ENTROPY AT CONSTANT TF FROM ZERO PRESSURE TO PF PLUS SVAP 280
C CHANGE IN ENTROPY AT 32F FROM .08854 PSIA TO 0.0 PSIA SVAP 285
DSZP=(DLOG(T)*1.472+7.55660-4*T-47.8365*TAU)*.23888889-1.98022309 SVAP 290
C DSZP=CHANGE IN ENTROPY AT ZERO PRESSURE AS TEMP. GOES FROM 32F TO SVAP 295
C INTEGRAL(CPO/T*DT) EVALUATED AT 32F = 1.98022309 SVAP 300
SVAP=DSZP+DSC T+2.1877 SVAP 305
C S OF EVAPORATION AT 32F + .08854 PSIA = 2.1877 SVAP 310
RETURN SVAP 315
END SVAP 320

```

```

FUNCTION VVAP(TF,PF)                                VVAP 100
IMPLICIT REAL*8(A-H,O-Z)                            VVAP 101
CALCULATES SPECIFIC VOLUME OF STEAM                VVAP 102
C KEENAN + KEYES, 1951, P. 15, EQ. 13              VVAP 105
P=PF/14.6959 VVAP 110
T=(TF+459.69)/1.8 VVAP 115

```

```

TAU=1./T
TAU2=TAU*TAU
TAU3=TAU2*TAU
TAU12=TAU3**4
EXTAU=- (10.** (80870.*TAU2))
BO=EXTAU*TAU*2641.62+1.89
Q=BO*TAU
QP=Q*P
G1=(-1.624605*TAU+82.546)*TAU
G2=-1.269705*TAU2+.21828
G3=+6.768064*TAU12*TAU12-3.6350-4
C SIGN OF G3 REVERSED FROM K+K
B=(( (G3*QP**9+G2)*QP*QP+G1)*QP+1.)*BO*.0160185
VVAP=.072964908*T/P+R
RETURN
END

```

VVAP 120
VVAP 125
VVAP 130
VVAP 135
VVAP 140
VVAP 145
VVAP 150
VVAP 155
VVAP 160
VVAP 165
VVAP 170
VVAP 175
VVAP 180
VVAP 185
VVAP 190
VVAP 195

```

FUNCTION TVAPH(PI,HI)
IMPLICIT REAL*8(A-H,O-Z)
CALCULATES TEMPERATURE OF STEAM FROM PRESSURE AND ENTHALPY USING HVAP AND
C LINEAR INTERPOLATION
P=PI
H=HI
C CHECK FOR SUPERHEAT (ADDED ON 8-12-71)
T = TSAT(P)
H1 = HVAP(T,P)
IF(H1.GE.H) GO TO 10
C START WITH ABOUT 50 DEGREES SUPERHEAT
DEL T=40.
T=8574./ (15.424-DLOG(P))-410.
H1=HVAP(T,P)
IF(H1-H)2,10,1
C 1 USE LESS SUPERHEAT
1 DEL T=-DEL T
2 T=DEL T+T
5 H0=H1
H1=HVAP(T,P)
DEL T=(H1-H)*DEL T/(H0-H1)
T=DEL T+T
IF(DABS(DEL T)-.01)10,10,5
10 TVAPH=T
RETURN
END

```

TVAPH100
TVAPH101
TVAPH102
TVAPH105
TVAPH110
TVAPH115

TVAPH120
TVAPH125
TVAPH130
TVAPH135
TVAPH140
TVAPH145
TVAPH150
TVAPH155
TVAPH160
TVAPH165
TVAPH170
TVAPH175
TVAPH180
TVAPH185
TVAPH190
TVAPH195

```

FUNCTION TVAPS(PI,S1)
IMPLICIT REAL*8(A-H,O-Z)
CALCULATES TEMPERATURE OF STEAM FROM PRESSURE AND ENTROPY USING SVAP AND
C LINEAR INTERPOLATION
S=S1
P=PI
C START WITH ABOUT 50 DEGREES SUPERHEAT
DEL T=40.
T=8574./ (15.424-DLOG(P))-410.
S1=SVAP(T,P)

```

TVAPS100
TVAPS101
TVAPS102
TVAPS105
TVAPS110
TVAPS115
TVAPS120
TVAPS125
TVAPS130
TVAPS135

```

      IF(S1-S)2,10,1
C 1 USE LFSS SUPERHEAT
      DELT=-DELT
      T=DELT+T
      S0=S1
      S1=SVAP(T,P)
      DELT=(S1-S)*DELT/(S0-S1)
      T=DELT+T
      IF(DABS(DELT)-0.01)10,10,5
C 10 TVAPS=T
      RETURN
      END

```

TVAPS140
TVAPS145
TVAPS150
TVAPS155
TVAPS160
TVAPS165
TVAPS170
TVAPS175
TVAPS180
TVAPS185
TVAPS190
TVAPS195

```

      SUBROUTINE PROPH(AM,S,T,P,H)
      IMPLICIT REAL*8(A-H,O-Z)
C
C  SUBPROGRAM FOR STEAM PROPERTIES WHEN P AND H ARE KNOWN
C
      T=TSAT(P)
      HG=HVAP(T,P)
      IF(H-HG)1,2,3
      1 CONTINUE
C  WET STEAM
      AM=(HG-H)/(HG-HLIQ(T))
      S=(1.00-AM)*SVAP(T,P)+AM*SLIQ(T)
      RETURN
      3 CONTINUE
C  SUPERHEATED STEAM
      T=TVAPH(P,H)
      2 AM = 0.000
C  SATURATED STEAM
      S=SVAP(T,P)
      RETURN
      END

```

```

      SUBROUTINE PROPS(H,P,S)
      IMPLICIT REAL*8(A-H,O-Z)
C
C  SUBPROGRAM FOR STEAM PROPERTIES WHEN P AND S ARE KNOWN
C
      T=TSAT(P)
      SG=SVAP(T,P)
      IF(S-SG)1,2,3
      1 CONTINUE
C  WET STEAM
      AM=(SG-S)/(SG-SLIQ(T))
      H=(1.00-AM)*HVAP(T,P)+AM*HLIQ(T)
      RETURN
      3 CONTINUE
C  SUPERHEATED STEAM
      T=TVAPS(P,S)
      2 CONTINUE
C  SATURATED STEAM
      H=HVAP(T,P)
      RETURN
      END

```

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

INTERNAL DISTRIBUTION

1. T. D. Anderson	51. M. I. Lundin
2. S. J. Ball	52. R. N. Lyon
3. S. E. Beall	53. R. E. MacPherson
4. L. L. Bennett	54. H. C. McCurdy
5. H. I. Bowers	55. J. W. Michel
6. R. H. Chapman	56. A. J. Miller
7. W. B. Cottrell	57. A. M. Perry
8. N. E. Clapp	58. M. W. Rosenthal
9. T. E. Cole	59. M. Sheldon
10. F. L. Culler	60. M. J. Skinner
11-15. J. G. Delene	61. I. Spiewak
16. D. M. Eissenberg	62. D. A. Sundberg
17. A. P. Fraas	63. D. G. Thomas
18. L. C. Fuller	64. D. B. Trauger
19-43. R. P. Hammond	65. J. V. Wilson
44. E. C. Hise	66. G. D. Whitman
45. H. W. Hoffman	67. MIT Practice School
46. J. S. Johnson	68-69. Central Research Library
47. J. E. Jones	70. Y-12 Document Ref. Section
48. P. R. Kasten	71-73. Laboratory Records Dept.
49. T. W. Kerlin (UT)	74. Laboratory Records - RC
50. K. A. Kraus	75-250. Laboratory Records (for
AEC List 2 Distribution)

EXTERNAL DISTRIBUTION

AEC-Washington

- 251-253. Director, Division of Reactor Licensing
- 254-255. Director, Division of Reactor Standards
- 256. Director, RDT
- 257. Assistant Director for Project Management, RDT
- 258-259. Chief, Desalting Branch, RDT
- 260. Assistant Director for Plant Engineering, RDT
- 261. Assistant Director for Reactor Engineering, RDT
- 262. Assistant Director for Nuclear Safety, RDT
- 263. Assistant Director for Engineering Standards, RDT
- 264. Assistant Director for Program Analysis, RDT
- 265. Assistant Director for Army Reactors, RDT

AEC-ORO

- 266. Senior Site Representative, ORNL
- 267. Research and Technical Support Division
- 268. Patent Office
- 269-270. Technical Information Center
- 271-287. Manager, Technical Information Center (for ACRS distribution)

U.S. Department of the Interior - OSW

- 288-290. Director, Office of Saline Water
- 291-294. Director, Research
- 295-296. Chief, Distillation Division
 - 297. Chief, Desalting Feasibility and Economic Studies
 - 298. Chief, Program Analysis

OAK RIDGE NATIONAL LABORATORY

OPERATED BY
UNION CARBIDE CORPORATION
NUCLEAR DIVISION

POST OFFICE BOX X
OAK RIDGE, TENNESSEE 37830

SPECIAL 4TH CLASS RATE BOOKS

If a change of address is needed, indicate the change, include the zip code, and identify the document; tear off this cover and return it to the above address.