NewYorkPower Authority

James A. FitzPatrick Nuclear Power Plant

Calculation No.: JAF-CALC-MISC-03116

QA Category: I ☒ M ☐ II/III ☐

Design Basis: YES ☐ NO ☒

Prepared by: JAF ☒ Vendor ☐

Modification or Task No: N/A

Title: EOP/SAOG Support Calculations

Discipline: Mech.

System (number and name): N/A

DBD No. N/A

REV.	Description & Status (F,P,I,V or S)	Prepared by: Name Signature/Date	Checked by: Name Signature/Date	Approved by: Name Signature/Date	D.V. Y/N
0	Final	David E. Burch (David E. Burch) 9DEC98	*	Sheldon K. Kohs (Sheldon K. Kohs) 12/13/98	N

FOR INFORMATION ONLY

JAN 26 1999

FOR INFORMATION ONLY

SYSTEM: 00MISC

QA CAT: CAT I

FILE#: 77-C-0374

COMP. PRINTOUT LOC:

* ~~Independent~~ 12/13/98 Independently verified by alternate calculation method as described in General Physics Corporation Report GP-R-NYPA-98-001, R0, dated December 12, 1998

DISTRIBUTION:

C= CONTROLLED I= INFO

NAME	DEPT	LOC	C	I
Dave Burch	DE-Mech	JAF	X	
Richard Sullivan	Operations	JAF		X

RELATED COMPONENTS

None: ~~These~~ 12/9/02 This calculation provides limits used in plant Emergency Operating Procedures and Severe Accident Operating Guidelines. Due to the comprehsive nature of those documents, it is not practical to list all related components and structures.

RELATED STRUCTURES

None: See discussion under Related Components

RELATED ENGINEERING DOCUMENTS

JAF Emergency Procedure and Severe Accident Guidelines, Revision 0

RELATED PROCEDURES

JAF Emergency Operating Procedures, Rev 5

JAF Severe Accident Operating Guidelines, Rev. 0

SECURITY:	YES	
	NO	X

COMPUTER PRINTOUT:	YES	
	NO	X

No. of Attachments: 31

No. of Calculation pages: 52

GENERAL PHYSICS CORPORATION

120 East First Street
Oswego, NY 13126

(315) 342-5127
Fax (315) 342-5133

GP-L-615982462

December 10, 1998

Mr. Nick Avrakotos
SAM Implementation Project Manager
New York Power Authority
James A. FitzPatrick Nuclear Power Plant
268 Lake Road
P.O. Box 41
Lycoming, NY 13093

Subject: Independent Verification of Appendix C Calculation in Support of Development of Plant Specific Emergency Operating Procedures and Severe Accident Operating Guidelines for the James A. FitzPatrick Nuclear Power Plant

Dear Mr. Avrakotos:

General Physics Corporation, under Agreement C 98 J1089 between General Physics and the Authority, has completed an independent design review of calculation JAF-CALC-MISC-03116, revision 0, in accordance with the Authority's Design Control Manual DCM-4A and the General Physics Nuclear Quality Assurance Program. The method of verification was by Alternate Calculation per the guidelines of DCM-4A, Attachment 4.

In summary, the plant specific input data was verified by direct comparison to the references identified in materials provided by the Authority. The calculation was verified using EOPCalc™, a custom Mocrosoft® Excel application developed by KLR Services, Inc., specifically for performing EPG/SAG Appendix C calculations. Input data provided by the Authority was entered in EOPCalc and used to generate a set of plant specific calculations. The results of these calculations were compared to those obtained by the Authority. The calculation results obtained by the Authority were generally consistent with those obtained by EOPCalc. Small differences, judged to be insignificant, resulted from the two methodologies using different sources for values of the properties of water; either source used is considered an acceptable reference.

The results of our independent review will be forwarded to the Authority as General Physics Report GP-R-NYPA-98-001.

Should you have any questions concerning this letter, or if General Physics can be of service to you in any other matter, please feel free to contact me directly in my Oswego, NY, office at (315) 342-5127 or leave a voice message with my pager at (888) 363-1989. Alternatively, I can be reached by electronic mail at jrfromm@genphysics.com.

Sincerely,

GENERAL PHYSICS CORPORATION

James R. Fromm
Manager, Corrective Action Projects

cc: D. Burch, NYPA
C. Shaefer, GPC QA
Report GP-R-NYPA-98-001 file

Project: JAFNPP
Subject: EOP/SAOG Support Calculations

Computed by DGB Date 12/9/98
Checked by * Date

OBJECTIVE....2
DESIGN INPUTS/BASES....2
ASSUMPTIONS....2
REFERENCES....2
REFERENCE DRAWINGS....2
METHOD....2
Steam.exe....3
Eop_calcs.exe....3
General description....4
'do eop calculations.cpp'....5
'plant data.h'....7
'thermo.h'....12
'eop calculation input.h'....12
'shared data.h'....12
'shared functions.h'....13
COMPUTATIONS....14
Maximum Pressure Suppression Primary Containment Water Level: MPSPCWL....14
Primary Containment Pressure Limits: PCPL-A, PCPL-C....19
SRV Tail Pipe Level Limit: STPLL....20
Heat Capacity Temperature Limit, Low-Pressure Endpoint Temperature: tsp_hctl_2....22
RPV Variables....22
Minimum alternate rpv flooding pressure: marfp....23
Minimum number of srvs required for emergency depressurization: mnsred....24
Minimum srv re-opening pressure: msrp....25
Minimum rpv flooding pressure: mrfp....25
Minimum core flooding interval: mcfi....26
Maximum core uncovery time limit: mcutl....27
Minimum steam cooling rpv water level: mscrwl....27
Minimum zero-injection rpv water level: mzirwl....28
Heat Capacity Temperature Limit: HCTL....35
Boron Injection Variables....36
Cold shutdown boron weight: csbw....36
Hot shutdown boron weight: hsbw....37
Boron injection initiation temperature: biit....39
Pressure Suppression Pressure: PSP....41
Drywell Spray Related Variables....41
Drywell spray initiation limit: dwsil....48
Minimum drywell spray flow: mdsf....48
Minimum Debris Retention Injection Rate: MDRIR....49
Suppression Chamber Spray Initiation Pressure: SCSIP....50
SUMMARY/CONCLUSIONS....50
ATTACHMENTS....52
FUNCTION INDEX....

OBJECTIVE

This calculation determines the values of plant specific variables and curves required by the JAF Emergency Procedure and Severe Accident Guidelines. These guidelines are in turn used to generate Emergency Operating Procedures and Severe Accident Operating Guidelines for JAF.

DESIGN INPUTS/BASES

Values of generically determined parameters required as input to this calculation are provided to the program through the input file, 'generic data.' The sources of each of the generic values are provided in attachments 1 through 3.

Values of plant-specific parameters required as input to this calculation are provided to the program through the input file, 'plant data.' The sources of each of the plant-specific values are provided in attachments 4 through 10.

Generally, inputs to this calculation are nominal values in accordance with the philosophy of using best-estimate rather than licensing basis models for generation of limit curves and point values (reference 1). The limits established by this calculation do not represent safety-related or licensing basis (other than the commitment to the NRC to upgrade Emergency Operating Procedures for Severe Accident Management) design constraints on system (or integrated plant) operation.

ASSUMPTIONS

1) For establishing the maximum and minimum normal operating temperatures in the torus airspace, the gas temperature is assumed to be in equilibrium with the water temperature. Maximum normal operating temperature is based on the Technical Specification water temperature requiring reactor scram, minimum temperature is based on the lower temperature limit given in plant operating procedures.

REFERENCES

1. BWROG Emergency Procedure and Severe Accident Guidelines, Revision 1, July, 1997
2. ASME Steam Tables, Sixth Edition, 1997 printing, American Society of Mechanical Engineers
3. BWROG Emergency Procedure and Severe Accident Guidelines, Appendix C, Revision 1, July, 1997

REFERENCE DRAWINGS

None

METHOD

This calculation was performed as follows. Steam.exe is executed to create and populate the file, 'thermo data,' with values of the properties of water (and steam). 'EOP calculations.exe' is then executed to obtain the values required to support generation of the plant specific EPGs and SAGs. General descriptions of Steam.exe and 'EOP calculations.exe' are provided in the following subsections. Detailed descriptions of calculation processes are provided in the 'COMPUTATIONS' section.

Steam.exe

Steam table values are obtained using the Steam.exe application (version 1.30) obtained from the American Society of Mechanical Engineers. This program determines the properties of steam and water using the 1967 IFC Formulation for Industrial Use and other LAPWS releases (reference 2).

To obtain the steam and water properties required by 'EOP calculations.exe', a file named config.dat with the contents shown in attachment 11 must be present in the same directory as Steam.exe. Config.dat specifies the file containing parameter ranges for which thermodynamic properties will be calculated (thermin.dat, attachment 12), the file containing program output ('thermo data') and options used during program execution. Options specified in config.dat are:

- use of US customary units for input and output;
- calculation of table(s);
- the initial table being at constant pressure;
- no screen or printed output (output to file only), and;
- calculation/editing of specific volume and specific heat in addition to pressure, temperature, specific enthalpy and specific entropy.

Thermin.dat specifies:

- the initial table will be produced for a pressure of 15 psia with temperature ranging from 32°F to 600°F at 1°F intervals;
- production of tables at constant pressures of 500 psia and 1000 psia with temperature ranging from 32°F to 600°F at 1°F intervals;
- production of a saturation properties table for temperatures ranging from 32°F to 600°F at 1°F intervals, and;
- production of a saturation properties table for pressures ranging from 1 psia to 1500 psia at 1 psia intervals.

As previously stated, output from Steam.exe is stored in 'thermo data' which is used as input by 'EOP calculations.exe'.

EOP calculations.exe

GENERAL DESCRIPTION

Note: file names of source code as well as function names are written in normal text while specific variable names are italicized.

The source code specific to 'EOP calculations.exe' was written in the C++ language and compiled using Microsoft Visual C++, version 6.0, operating within version 6.00.8168.0 of the Microsoft Developer's Studio environment. The source code is divided among the following files:

- do eop calculations.cpp (attachment 13)
- plant data.h (attachment 14)
- thermo.h (attachment 15)
- eop calculation input.h (attachment 16)
- shared data.h (attachment 17)
- shared functions.h (attachment 18)
- mpspcwl.h (attachment 19)
- pcpl.h (attachment 20)

- stpll.h (attachment 21)
- hctl.h (attachment 22)
- rpv.h (attachment 23)
- boron.h (attachment 24)
- psp.h (attachment 25)
- drywell spray.h (attachment 26)
- mdrir.h (attachment 27)
- scsip.h (attachment 28)

'do eop calculations.cpp' provides the main function for the program, which utilizes modular functions defined within the header files (*.h files, listed above) to complete the calculation. A listing of all of the available functions is provided in attachment 29, which also gives the header file of the function source code, a brief statement of purpose, and a cross-reference of associated functions. Descriptions of 'do eop calculations.cpp,' 'plant data.h,' thermo.h, 'eop calculation input.h,' 'shared data.h' and 'shared functions.h' are as follows. The other header files contain functions which perform the detailed calculations are discussed in the 'COMPUTATIONS' section.

DO EOP CALCULATIONS.CPP

This file contains the definition of the main function of the program which calculates values required to support generation of the plant specific EPGs and SAGs. This program element sequences the order of execution of functions which complete the actual calculation. The order of execution of main program statements is:

1. load_plant_data(): loads the plant specific and generic inputs required by the calculation.

2. load_thermo(): loads functions for calculation of thermodynamic properties of water and steam.

3. files are created (and headers written) for storage of intermediate and final results of the calculation.

4. write_input_values() writes plant-specific and generic input values to disk.

5. calc_mpspcwl(): calculates the Maximum Pressure Suppression Primary Containment Water Level. This value is required as an input for calculation of the Heat Capacity Temperature Limit (HCTL), SRV Tail Pipe Level Limit (STPLL) and Pressure Suppression Pressure (PSP).

6. sequence_pcpl(): controls the order of execution of functions which ultimately result in calculation of Primary Containment Pressure Limit (PCPL) A and PCPL C. PCPL A is an input to calculation of the HCTL and PSP.

7. calc_stpll(): calculates the STPLL.

8. calc_tsp_hctl_2(): calculates *tsp_hctl_2*, the HCTL low-pressure endpoint, as a function of torus water level. The results of this calculation are inputs for calculation of the Minimum Debris Retention Injection Rate (MDRIR), Minimum Number of SRVs Required for Emergency Depressurization (MNSRED), Minimum RPV Flooding Pressure (MRFP), Minimum Core Flooding Interval (MCFI), HCTL and PSP.

9. sequence_rpv(): controls the order of execution of functions which ultimately calculate the Minimum Alternate RPV Flooding Pressure (MARFP), MNSRED, MRFP, MCFI, Maximum Core Uncovery Time Limit (MCUTL), Minimum Steam Cooling RPV Water Level (MSCRWL) and Minimum Zero-Injection

RPV Water Level (MZIRWL). MRFP is required as input into calculation of the MDRIR and HCTL. MNSRED is required as input into calculation of the HCTL.

10. calc_hctl(): calculates the HCTL. The HCTL is used as input for calculation of the Boron Injection Initiation Temperature (BIIT) and PSP.

11. sequence_boron(): controls the order of execution of functions which ultimately result in calculation of the Cold Shutdown Boron Weight (CSBW), Hot Shutdown Boron Weight (HSBW) and the BIIT.

12. calc_psp(): calculates the PSP.

13. calc_dwsil(): calculates the Drywell Spray Initiation Limit (DWSIL).

14. calc_mdsf(): calculates the Minimum Drywell Spray Flow (MDSF).

15. calc_mdrir(): calculates the Minimum Debris Retention Injection Rate (MDRIR).

16. calc_scsip(): calculates the Suppression Chamber Spray Initiation Pressure (SCSIP).

17. free_thermo(): releases memory used for storage of thermodynamic properties of water and steam.

In C++, program execution begins with the main function, which is called automatically. Descriptions of functions called by main are provided in the remainder of this section, or in the 'COMPUTATIONS' section, as appropriate.

PLANT DATA.H

'plant data.h' declares the plant specific and generic input variables required to perform EOP support calculations and defines functions to load values of these variables into memory from file. Definitions of the functions included within 'plant data.h' are as follows:

assign_int():

Operates on values of type 'char' to return a value of type 'int.'

The algorithm implemented is:

The library function istrstream is used to declare an input buffer named *data_string* (in this case 80 characters long) which is loaded with the contents of the character type variable used in the function call (*line_as_string*). *Data_string* is then available for use with the extraction operator, '>>,' as a source of input. In the case of this function, the contents of *data_string* to the first delimiting character (blank space) are placed in *return_value*, which is given back to the statement that called the function.

The function is called with an assignment statement of form:

a = assign_int(*line_as_string*)

assign_float():

Operates on values of type 'char' to return a (floating point) value of type 'double.'

The algorithm implemented is:

The library function istrstream is used to declare an input buffer named *data_string* (in this case 80 characters long) which is loaded with the contents of the character type variable used in the function call (*line_as_string*). *Data_string* is then available for use with the extraction operator, '>>,' as a source of input. In the case of this function, the contents of *data_string* to the first delimiting character (blank space) are placed in *return_value*, which is given back to the statement that called the function.

The function is called with an assignment statement of form:

a = assign_float(*line_as_string*)

assign_n_floats():

Operates on values of type 'char' to return an array of (floating point) values of type 'double.'

The algorithm implemented is:

The library function istrstream is used to declare an input buffer named *data_string* (in this case 80 characters long) which is loaded with the contents of the character type variable used in the function call (*line_as_string*). *Data_string* is then available for use with the extraction operator, '>>,' as a source of input. In the case of this function, a loop is executed *n_floats* times, with each execution reading input off of *data_string* to the next delimiting character (blank space) and assigning it to the next element of the array variable (*return_floats*) used in the function call.

The function is called with a statement of form:

assign_n_floats(*line_as_string*, *return_floats*, *n_floats*)

load_plant_data():

Requires no input value from its calling statement and returns an integer value indicating the status of execution of the function. The function operates on variables defined in its associated header file, 'plant data.h' and requires input files named 'plant data' and 'generic data' to successfully execute.

The process implemented is:

The strcpy library function is used to assign the file name 'plant data' to a character variable, *input_file*, then the ifstream library function is used to associate this variable with an input stream (*data_file1*). A block of memory sufficient to hold the contents of the input file is then allocated and the file is read into memory using an assignment statement of format *data_file1*.getline(). After the file has been read to memory, it is closed. Individual lines of input are read from memory using 'strcpy' and variable assignments are made using 'strcpy' or the user defined functions 'assign_int,' 'assign_float,' or 'assign_n_floats' as appropriate to the variable type. In the case of structured variables, the individual elements will be obtained by assignation from the array returned by assign_n_floats. Theses assignments will be contained within for() loops, as appropriate.

When all of the plant specific variable values have been assigned, the memory containing the input file contents will be deallocated.

Generic data is then loaded into the appropriate variables using the same process used for loading the plant specific data, with the exception that the input file is 'generic data' (rather than 'plant data'), and the input stream is called *data_file2* (rather than *data_file1*).

The function is called with an assignment statement of form:

a = load_plant_data(). The value assigned to *a* indicates whether load_plant_data was successful in opening the input files, 'plant data' and 'generic data.' A value other than 0, indicates the function did not execute successfully.

THERMO.H

Thermo.h declares variables to describe the thermodynamic properties of water and steam for use within the EOP support calculations. Functions required to initialize these variables using values obtained from the execution of steam.exe and that allow assignment of steam table values to variables used in other program modules are provided. Definitions of the functions included within thermo.h are as follows:

load_thermo():

Requires no input value from its calling statement and returns an integer value indicating the status of execution of the function. The function operates on variables defined in its associated header file, thermo.h and requires an input file named 'thermo data,' created by execution of Steam.exe with input files config.dat and thermin.dat (attachments 11 and 12) to successfully execute.

The process implemented is:

The strcpy library function is used to assign the file name 'thermo data' to a character variable, *input_file*, then the ifstream library function is used to associate this variable with an input stream (*data_file*). A block of memory sufficient to hold the contents of the input file is then allocated and the file is read into memory using an assignment statement of format *data_file*.getline(). After the file has been read to memory, it is closed. Individual lines of input are read from memory into a variable, *line_as_string* (which is 80 characters long) using the 'strcpy' library function. The contents of this variable are then loaded into an input stream buffer, *data_string* using the istrstream library function. The structured variables (pointed to by *sub15_ptr*, *sub500_ptr*, *sub1000_ptr*, *satbytemp_ptr* and *satbypress_ptr*) are then loaded using the extraction operator, '>>,' acting on *data_string*. These assignment processes are embedded in loops, since the input data file has a repetitive structure based on temperature and pressure. Lines of the input file not containing thermodynamic data are jumped over by resetting the pointer into memory.

When all of the thermodynamic property variable values have been assigned, the memory containing the input file contents will be deallocated.

The function is called with an assignment statement of form:

a = load_thermo(). The value assigned to *a* indicates whether load_thermo was successful in opening the input file, 'thermo data.' A value other than 0, indicates the function did not execute successfully.

Functions which return thermodynamic properties of water and steam:

Saturated water and steam properties as a function of pressure

Functions are provided to determine the properties of water and steam at saturation as a function of pressure (in psia). The process used is the same for each property, with the exception that each function accesses elements of the array pointed to by *satbypress_ptr* holding values of the relevant parameter. Parameter values are available for pressures within the range 1 psia to 1500 psia, inclusive.

The first step in this process is to determine the location in the array of the next lower pressure than the value input to the function call. This is done using the int() library function to convert the input value from a floating point value (of type 'double') to an integer and subtracting 1 (since index 0 of the *satbypress_ptr* array contains values for 1 psia). This value is assigned to *lo_index*. This will be used as an offset into the array pointed at by *satbypress_ptr*. The following steps in the process are dependent on the input pressure.

For input pressure lower than 1500 psia (the largest value for which thermodynamic data is stored), the offset to the next location in the array is determined by adding 1 to *lo_index* and assigning the result to *hi_index*.

The value of the desired parameter at the lower index is then assigned to *base_output_value*. Next, the difference between the value of the parameter at the high index and at the low index is assigned to *delta_output_value*. The difference between the input pressure and the next lower integer value of pressure is determined by subtracting *hi_index* from *pressure_in*, with the result assigned to *delta_pressure*. Finally, the value to be returned by the function is determined by evaluation of the expression:

base_output_value + (*delta_pressure* * *delta_output_value*).

Functionally, the process (which does not determine the return value by an assignment statement at the high end of the range of stored values) is represented by the following set of equations:

For $y = f(x)$,
$x_{lo} = \text{int}(x)$
$x_{hi} = x_{lo} + 1$
$y_{lo} = f(x_{lo})$
$\Delta y = f(x_{hi}) - y_{lo} = y_{hi} - y_{lo}$
$\Delta x = x - x_{lo}$
$y = y_{lo} + (\Delta x * \Delta y)$

note: int(x) returns the portion of x left of the decimal point; the expression, $\Delta x * \Delta y$, makes use of the fact that $x_{hi} - x_{lo} = 1$ to eliminate the denominator normally associated with the product of Δx and Δy; when x_{lo} and x_{hi} are converted to the indices, *lo_index* and *hi_index*, the respective values are 1 smaller since the first location in the array (zero offset) has a value of 1.

For input pressure exactly equal to 1500 psia, the value returned by the function is determined by assignment from the location pointed at by (*satbypress_ptr* + *lo_index*).

Available functions, listed by their calling convention, are (more functions are available than are required by the EOP support calculations in anticipation of reuse of the code in the future):

tsat(*pressure_in*): returns saturation temperature [°F], given pressure [psia];
hfsat_by_p(*pressure_in*): returns saturated fluid enthalpy [Btu/lbm], given pressure [psia];
hgsat_by_p(*pressure_in*): returns saturated vapor enthalpy [Btu/lbm], given pressure [psia];
sfsat_by_p(*pressure_in*): returns saturated fluid entropy [Btu/lbm], given pressure [psia];
sgsat_by_p(*pressure_in*): returns saturated vapor entropy [Btu/lbm], given pressure [psia];
vfsat_by_p(*pressure_in*): returns saturated fluid specific volume [ft^3/lbm], given pressure [psia];
vgsat_by_p(*pressure_in*): returns saturated vapor specific volume [ft^3/lbm], given pressure [psia];
cpfsat_by_p(*pressure_in*): returns saturated fluid specific heat [Btu/°F-lbm], given pressure [psia];
cpgsat_by_p(*pressure_in*): returns saturated vapor specific heat [Btu/°F-lbm], given pressure [psia];

Saturated water and steam properties as a function of temperature

Functions are provided to determine the properties of water and steam at saturation as a function of temperature (in °F). The process used is the same for each property, with the exception that each function accesses elements of the array pointed to by *satbytemp_ptr* holding values of the relevant parameter. Parameter values are available for temperatures within the range 32°F to 600°F, inclusive.

The first step in this process is to determine the location in the array of the next lower temperature than the value input to the function call. This is done using the int() library function to convert the input value from a floating point value (of type 'double') to an integer and subtracting 32 (since index 0 of the *satbytemp_ptr* array contains values for 32°F). This value is assigned to *lo_index*. This will be used as an offset into the array pointed at by *satbytemp_ptr*. The following steps in the process are dependent on the input temperature.

For input temperature lower than 600°F (the largest values for which thermodynamic data is stored), the offset to the next location in the array is determined by adding 1 to *lo_index* and assigning the result to *hi_index*.

The value of the desired parameter at the lower index is then assigned to *base_output_value*. Next, the difference between the value of the parameter at the high index and at the low index is assigned to *delta_output_value*. The difference between the input temperature and the next lower integer value of temperature is determined by subtracting (*lo_index* + 32) from *temperature_in* with the result assigned to *delta_temperature*. Finally, the value to be returned by the function is determined by evaluation of the expression:

base_output_value + (*delta_temperature* * *delta_output_value*).

Functionally, the process (which does not determine the return value by an assignment statement at the high end of the range of stored values) is represented by the following set of equations:

For $y = f(x)$,
$x_{lo} = \text{int}(x)$
$x_{hi} = x_{lo} + 1$
$y_{lo} = f(x_{lo})$
$\Delta y = f(x_{hi}) - y_{lo} = y_{hi} - y_{lo}$
$\Delta x = x - x_{lo}$
$y = y_{lo} + (\Delta x * \Delta y)$

note: int(x) returns the portion of x left of the decimal point; the expression, $\Delta x * \Delta y$, makes use of the fact that $x_{hi} - x_{lo} = 1$ to eliminate the denominator normally associated with the product of Δx and Δy; when x_{lo} and x_{hi} are converted to the indices, *lo_index* and *hi_index*, the respective values are 32 smaller since the first location in the array (zero offset) has a value of 32.

For input temperature exactly equal to 600°F, the value returned by the function is determined by assignment from the location pointed at by (*satbytemp_ptr* + *lo_index*).

Available functions, listed by their calling convention, are (more functions are available than are required by the EOP support calculations in anticipation of reuse of the code in the future):

psat(*temperature_in*): returns saturation pressure [psia], given temperature [°F];
hfsat_by_t(*temperature _in*): returns saturated fluid enthalpy [Btu/lbm], given temperature [°F];
hgsat_by_t(*temperature _in*): returns saturated vapor enthalpy [Btu/lbm], given temperature [°F];
sfsat_by_t(*temperature _in*): returns saturated fluid entropy [Btu/lbm], given temperature [°F];
sgsat_by_t(*temperature _in*): returns saturated vapor entropy [Btu/lbm], given temperature [°F];
vfsat_by_t(*temperature _in*): returns saturated fluid specific volume [ft^3/lbm], given temperature [°F];
vgsat_by_t(*temperature _in*): returns saturated vapor specific volume [ft^3/lbm], given temperature [°F];
cpfsat_by_t(*temperature _in*): returns saturated fluid specific heat [Btu/°F-lbm], given temperature [°F];
cpgsat_by_t(*temperature _in*): returns saturated vapor specific heat [Btu/°F-lbm], given temperature [°F];

Subcooled water and steam properties

Functions are provided to determine the properties of subcooled or superheated water and steam as a function of pressure (in psia) and temperature (in °F). The process used is the same for each property, with the exception that each function points to a different element in the arrays containing thermodynamic data (the arrays each contain structured variables). Parameter values are available for pressures of 15 psia, 500 psia and 1000 psia, with temperatures within the range of 32°F to 600°F, inclusive.

The first step in this process is to determine the location in the array of the next lower temperature than the value input to the function call. This is done using the int() library function to convert the input value from a floating point value (of type 'double') to an integer and subtracting 32 for temperature (since index 0 of the *sub15_ptr*, *sub500_ptr* and *sub1000_ptr* arrays contain values for 32°F). This value is assigned to *lo_index*. This will be used as an offset into the array pointed at by *sub15_ptr*, *sub500_ptr* or *sub1000_ptr*, as appropriate. The offset to the next location in the array is then determined by adding 1 to *lo_index* and the value is assigned to *hi_index*.

The next step in the process is to evaluate the pressure entered in the function call. This is done using an 'if' statement to test the value of pressure. The intermediate steps of the process (which are the same independent of pressure) are applied to different arrays depending on the result of this test. Intermediate evaluations are performed on the array pointed at by *sub15_ptr* for an input pressure of 15 psia, *sub500_ptr* for an input pressure of 500 psia and *sub1000_ptr* for an input pressure of 1000 psia.

If the input temperature is below 600°F, the value of the desired parameter at the lower index is then assigned to *base_output_value*. Next, the difference between the value of the parameter at the high index and at the low index is assigned to *delta_output_value*. The function next determines the value of *delta_temperature* by subtracting (*lo_index* + 32) from the input temperature. Finally, the value to be returned by the function is determined by evaluation of the expression:

base_output_value + (*delta_temperature* * *delta_output_value*).

Functionally, the process (which does not determine the return value by an assignment statement at the high end of the range of stored values) is represented by the following set of equations:

For $y = f(x)$,
$x_{lo} = \text{int}(x)$
$x_{hi} = x_{lo} + 1$
$y_{lo} = f(x_{lo})$
$\Delta y = f(x_{hi}) - y_{lo} = y_{hi} - y_{lo}$
$\Delta x = x - x_{lo}$
$y = y_{lo} + (\Delta x * \Delta y)$

note: int(x) returns the portion of x left of the decimal point; the expression, $\Delta x * \Delta y$, makes use of the fact that $x_{hi} - x_{lo} = 1$ to eliminate the denominator normally associated with the product of Δx and Δy; when x_{lo} and x_{hi} are converted to the indices, *lo_index* and i, the respective values are 32 smaller since the first location in the array (zero offset) has a value of 32.

If the input temperature is exactly equal to 600°F, the function returns the value of the parameter at the input pressure and 600°F.

Available functions, listed by their calling convention, are (more functions are available than are required by the EOP support calculations in anticipation of reuse of the code in the future):

hpt(*pressure_in*, *temperature_in*): returns enthalpy [Btu/lbm], given pressure [psia] and temperature [°F];
spt(*pressure_in*, *temperature_in*): returns entropy [Btu/lbm], given pressure [psia] and temperature [°F];
vpt(*pressure_in*, *temperature_in*): returns specific volume [ft^3/lbm], given pressure [psia] and temperature [°F];
cppt(*pressure_in*, *temperature_in*): returns specific heat [Btu/°F-lbm], given pressure [psia] and temperature [°F];

free_thermo():

Requires no input value from its calling statement and returns no value. The function frees the memory pointed at by *sub15_ptr*, *sub500_ptr*, *sub1000_ptr*, *satbytemp_ptr* and *satbypress_ptr*.

The process implemented is:

The delete[] library function is used to deallocate the memory used to store thermodynamic data.

The function is called with a statement of form:

free_thermo()

EOP CALCULATION INPUT.H

This file defines the function, write_input_values(), used to write values of inputs used in the calculation to a disk file (which also contains the final results of the calculation). A brief description of this function is:

write_input_values():

Requires the name of the file used to store the final results of the calculation on disk. This function returns no value.

The process implemented is:

The library function ofstream is used to declare a file buffer named *final_out_01*, attached to the physical disk file named in the string passed to the function through the variable, *final_01*. Since it is desired to add to a pre-existing file, the file is opened in the append mode, by inclusion of the instruction 'ios::app' in the function call. Strings describing plant-specific and generic inputs, with associated values, are placed on the file stream using the insertion operator, '<<.' When all values have been output, the file stream is closed to assure that the associated file buffer is flushed so that all data are output.

The function is called with a statement of form:

write_input_values(*final*)

SHARED DATA.H

This file declares variables used to carry data between functions defined in different header files (i.e., the value of the variable is determined by a function defined in one header file and used as an input by a function defined in another header file).

SHARED FUNCTIONS.H

This file defines functions that are used within more than one other header file. The functions defined accomplish simple mathematical tasks (determining maximum and minimum values, interpolating between values). The following functions are defined:

interpolate():

Operates on values (floating point) of type 'double' to return a value of type 'double.'

The algorithm implemented is:

$y_{mid} = y_{lo} + \{(y_{hi} - y_{lo}) * [(x_{mid} - x_{lo}) / (x_{hi} - x_{lo})]\}$, where
$y = f(x)$

The function is called with an assignment statement of form:

y_{mid} = interpolate(x_{lo}, x_{hi}, x_{mid}, y_{lo}, y_{hi})

min_floats():

Operates on values (floating point) of type 'double' to return a value of type 'double.'

The algorithm implemented is:

min(x, y) = x, if x < y
min(x, y) = y, if x ≥ y

The function is called with an assignment statement of form:

a = min(b, c)

max_int():

Operates on values of type 'integer' to return a value of type 'integer.'

The algorithm implemented is:

max(x, y) = x, if x > y
max(x, y) = y, if x ≤ y

The function is called with an assignment statement of form:

a = max(b, c)

max_floats():

Operates on values (floating point) of type 'double' to return a value of type 'double.'

The algorithm implemented is:

max(x, y) = x, if x > y
max(x, y) = y, if x ≤ y

The function is called with an assignment statement of form:

a = max(b, c)

COMPUTATIONS

The order of execution of functions that calculate values that support generation of the JAF EPGs and SAGs was discussed in the previous section. This section describes the calculations performed by these functions.

Many of the functions discussed in this section also contain code to write results of the computation to the file containing intermediate or final results as appropriate. In each of these cases, the ofstream library function is used to declare an output file buffer and attach it to a previously created disk file (passed to the function through the calling statement). Results are then placed in the file using the standard insertion

operator, '<<, acting on the specified file stream. When all results have been written to file from a given function, the file stream is closed to assure that the file buffer is emptied (and not leave objects hanging). File streams are renamed in each function to avoid compiler errors due to redefinition of variables. No further description of the process of writing output will be provided in descriptions of individual functions.

Intermediate results of the computations are provided in attachment 30.

Maximum Pressure Suppression Primary Containment Water Level: MPSPCWL

The MPSPCWL is the highest torus water level [ft] at which the pressure suppression function of the primary containment is assured (reference 3). The value of this parameter is assigned to the variable *mpspcwl* by the function, calc_mpspcwl(). The equation used for the calculation (which implements the reference 3 calculation methodology) is:

$$mpspcwl = hring - hsp_ref$$

This is implemented in the function as a simple assignment statement.

Primary Containment Pressure Limits: PCPL-A, PCPL-C

Reference 3 provides the following bases for the PCPLs. The PCPLs are the lesser of:

- The pressure capability of the primary containment.
- The maximum primary containment pressure at which vent valves sized to remove all decay heat from the containment can be opened and closed.
- For PCPL-A, the maximum primary containment pressure at which SRVs can be opened and will remain open.

sequence_pcpl()

Calculation of PCPL-A and PCPL-C is controlled by the function, sequence_pcpl(). The sequence of execution of functions called by sequence_pcpl() is:

1. get_component_strength()
2. calc_dppc_dwlpc()
3. pressure_for_srvs()
4. pressure_for_component(*index1*), called iteratively for each of the defined primary containment members.
5. calculate_pcpl()
6. pcpl_for_calcs()
7. do_pcpl_output(*interim_03*, *final_03*)

The process used by each of these functions is described below.

sequence_pcpl() is called with a statement of form:

sequence_pcpl(*interim*, *final*) (*interim* and *final* bring the output file names into the function so that they may be subsequently passed to do_pcpl_output()).

get_component_strength()

This function calculates the material strength (at 'calculation' and 'limiting' temperatures) of primary containment components for which pressure-retaining capability was determined. The calculation is performed iteratively for each component defined in the plant specific data (i.e., nbr_pc_members).

The temperature [°F] at which the pressure retaining capability of a component was calculated is obtained by assignment from the plant specific data:

The limiting temperature [°F] to be used for a given component is assigned to *t_lim* from generic data depending on the value of a flag indicating the location of the component (1 signifies drywell, 2 torus).

A flag indicating whether yield (1) or tensile (2) strength is to be used when evaluating a given component is assigned to *index2* from plant specific data.

A flag indicating the material a component is constructed from is assigned to *index3* from plant specific data. Values of the flag are the same as used in reference 3, i.e.:

1. SS304, SA340, SA320 and A312
2. S21800, A193 and A194
3. A36
4. A160 Gr. B and A105
5. A201 Gr. B, A212 and SA516

The next segment of code determines the temperatures [°F] for which generic material properties are stored that bracket the limiting temperature applicable to the component and assigns the values to *hi_temp* and *lo_temp*. The same information is combined with the flag indicating component strength basis to determine the generic material strength at *lo_temp* and *hi_temp*, with the values assigned to *lo_strength* and *hi_strength*. The normalized material strength at the temperature for which the component pressure retaining capability was calculated is obtained by passing *lo_temp*, *hi_temp*, *t_calc*, *lo_strength* and *hi_strength* to interpolate(), with the result assigned to *fy_at_calc*. The normalized material strength at the limiting temperature for the component location is obtained by passing *lo_temp*, *hi_temp*, *t_lim*, *lo_strength* and *hi_strength* to interpolate(), with the result assigned to *fy_at_lim*.

The fractional strength of the material at the component limiting temperature relative to the strength of the material at the temperature used to determine the pressure retaining capability of the component is determined by dividing *fy_at_lim* by *fy_at_calc* and the result is assigned to *fyield_lim*.

get_component_strength() is called with a statement of form:

get_component_strength()

calc_dppc_dwlpc()

This function evaluates the rate of change of hydrostatic pressure applied to submerged portions of the containment as a function of water level. The algorithm used is:

dppc_dwlpc = 1 / (144 * *vf_flood*), where
vf_flood = vpt(15, *tsp_flood*)

The units of *dppc_dwlpc* are psig/ft and *vf_flood* is in ft^3/lbm. Contrary to the Appendix C methodology, subcooled water properties are used to determine *vf_flood* rather than saturated water properties. This is more appropriate since EOP action levels are formulated to preclude taking the containment subatmospheric (as would be required to cause water at *tsp_flood* to reach saturation).

calc_dppc_dwlpc() is called with a statement of form:

```
calc_dppc_dwlpc()
```

pressure_for_srvs()

This function determines the limit on primary containment airspace pressure (in psig) such that SRVs can be opened. These values are assigned to the 120 x 1 array, *ppc_srv*[].

To open the SRVs, the airspace pressure must be at or below the pressure of the pneumatic supply to the valves (*psup_srv*) less the differential pressure required to open the valves (*dpsrv*). This value is compensated for hydrostatic head of water in containment as applicable.

The function initializes containment water level at 0 ft. For water level at or below the height of the SRV solenoids, the limiting pressure is determined by subtracting *dpsrv* from *psup_srv*, i.e.,

$$ppc_srv[] = psup_srv - dpsrv$$

This calculation is performed iteratively at intervals of 1 ft in water level.

When water level is above the SRV solenoids, but below the elevation of the (highest) containment vent (*hvent_pc*), the hydrostatic head of water above the solenoids must be accounted for by subtraction from the pressure required to open the valves. This is performed by solution of the following equation:

$$ppc_srv[] = (dppc_dwlpc * (hsrv - wlpc)) + psup_srv - dpsrv$$

Again, the calculation is performed iteratively at intervals of 1 ft in water level.

By definition, the PCPL is 0 for water levels above the elevation of the containment vent. This definition is coded in the final section of the function. In this section, *ppc_srv*[] array locations corresponding to water levels above *hvent_pc* are assigned the value 0.

pressure_for_srvs() is called with a statement of form:

```
pressure_for_srvs()
```

pressure_for_components()

This function calculates the limit on primary containment airspace pressure (in psig) to ensure that the load imposed by that pressure does not compromise the structural integrity of the component. These values are assigned to the 10 x 120 arrays, *ppc_calc* and *ppc_lim*. The component for which calculations are to be performed is passed to the function through the parameter, *index1* which is referred to as *component_no* internally to the structure.

The function initially assigns the elevation of the component from the structure defining the component properties to *h_n*, i.e.,

h_n = *pc_data*[*component_no*].*h_n*

Next, containment water level is initialized at 0 ft:

wlpc = 0

Following this, the pressure limits for the component are determined for water levels up to the elevation of the component. For these levels, no correction is required to account for hydrostatic head of water above the component. The pressure for which component integrity is calculated is directly assigned to *ppc_calc*[][]:

ppc_calc[][] = *pc_data*[].*pcalc_n*

The pressure at which component integrity is assured at the limiting temperature for the component location is determined by multiplying the calculated pressure by the ratio of the material strength at the limiting temperature to the material strength at the calculation temperature (*fyield_lim*[]) and the result is assigned to *ppc_lim*[][]:

ppc_lim[][] = *pc_data*[].*pcalc_n* * *fyield_lim*[]

These calculations are iterated for 1 ft increments in water level until water level is above the elevation of the component.

For water levels above the component and below the elevation of the containment vent, the pressure limits for the component are corrected for the hydrostatic head of water above the component. The pressure for which component integrity is calculated is corrected as follows:

ppc_calc[][] = (*dppc_dwlpc* * (*h_n* – *wlpc*)) + *pc_data*[].*pcalc_n*

And the pressure at which component integrity is assured at the limiting temperature for the component location is:

ppc_lim[][] = (*dppc_dwlpc* * (*h_n* – *wlpc*)) + *pc_data*[].*pcalc_n* * *fyield_lim*[]

These calculations are iterated for 1 ft increments in water level until water level is above the elevation of the containment vent.

By definition, the PCPL is 0 for water levels above the elevation of the containment vent. This is coded in the final section of this function by assigning 0 to *ppc_calc*[][] and *ppc_lim*[][] array locations corresponding to water levels above *hvent_pc*.

pressure_for_component() is called with a statement of form:

pressure_for_component(*index1*)

calculate_pcpl()

This function calculates PCPL-A and PCPL-C.

Upon entry, this function initializes the 120 x 1 array *p_component*[] which will store the limiting component pressure to an arbitrarily high value (1500).

Next the limiting pressure (*p_component*[]) to assure component integrity is determined as follows. *Ppc_calc*[][] is assigned to *p_component*[] if *p_component* is larger through the function min_floats (acting on *p_component*[] and *ppc_calc*[][]). *Ppc_lim*[]is assigned to *p_component*[] if *p_component* is larger through the function min_floats (acting on *p_component*[] and *ppc_lim*[][]). This process is repeated iteratively for each defined component (so the lowest value of *ppc_calc*[][] and *ppc_lim*[][] for all components at a given water level (index)) is obtained. The process is repeated iteratively for all indices from 0 to 119 (i.e., for all water levels).

PCPL-C, uncorrected for hydrostatic head of water above the torus pressure instrument tap is obtained by iteratively determining the lower of *p_component*[] (the containment structural pressure limit) and *ppc_vent* (the limiting pressure for containment vent valve operability) as a function of water level. This is done by applying the min_floats function to *ppc_vent* and *p_component*[] and assigning the result to *pcpl_c*[].

PCPL-A, uncorrected for hydrostatic head of water above the torus pressure instrument tap is obtained by iteratively determining the lower of *ppc_srv*[] and *pcpl_c*[] (i.e., consideration of SRV capability to operate is added to maintaining containment structural integrity and the ability to operate containment vent valves) as a function of water level. This is accomplished by applying min_floats to *ppc_srv*[] and *pcpl_c*[] and assigning the result to *pcpl_a*[].

These pressure limits are then corrected for the hydrostatic head of water above the torus pressure instrument tap as follows. First, *wlpc* and a corresponding index (*index1*) are reset to 0. Then, while *wlpc* is less than *hsc_tap* (the elevation of instrument tap), *wlpc* and *index1* are iteratively raised by 1. Finally, the correction for hydrostatic head is applied by iterative solution of the following equations (incrementing *wlpc* and *index1* by 1 at each step):

$$pcpl_a[] = pcpl_a[] + (dppc_dwlpc * (wlpc - hsc_tap))$$
$$pcpl_c[] = pcpl_c[] + (dppc_dwlpc * (wlpc - hsc_tap))$$

calculate_pcpl() is called with a statement of form:

```
calculate_pcpl()
```

pcpl_for_calcs()

This function generates the input required for calculation of the HCTL and PSP and stores the results in the 1 x 11 array, *pcpl_to_calcs*[]. Values are required for elevations ranging from the downcomer openings to the Maximum Pressure Suppression Primary Containment Water Level. Elevations considered are: 9.58, 10, 11, 12, 13, 13.88, 14, 15, 16, 17 and 17.85 ft. Since the index into *pcpl_a*[] corresponds to containment water level (in feet), *pcpl_to_calcs*[] for levels not directly stored are obtained by interpolation using the previously described interpolate function.

pcpl_for_calcs() is called with a statement of form:

```
pcpl_for_calcs()
```

do_pcpl_output()

Since interim and final results of the PCPL calculation were stored in arrays, rather than printing the results during execution of the functions that generated them, output was grouped in this single function. The process used is that which was already described.

do_pcpl_output() is called with a statement of form:

do_pcpl_output(*interim*, *final*)

SRV Tail Pipe Level Limit: STPLL

The STPLL is the lesser of the MPSPCWL and the highest torus water level at which opening an SRV will not result in exceeding the code allowable stresses in the SRV tail pipe, tail pipe supports, quencher or quencher supports. The values defining the curve satisfying these criterion, stored in the 1 x 5 array, *stpll_eop*[], are determined by the function, calc_stpll().

The first operation carried out by calc_stpll() is initialization of values which remain fixed throughout function execution. Design margins for SRV quenchers, quencher supports, tail pipes and tail pipe supports are obtained using the following relations, respectively:

fdes_q = (100 * *pq_code* / *pq_des*) - 100
fdes_qs = (100 * *pqs_code* / *pqs_des*) - 100
fdes_tp = (100 * *ptp_code* / *ptp_des*) - 100
fdes_tps = (100 * *ptps_code* / *ptps_des*) - 100

Then *prpv_1*, the RPV pressure for which actuation of an SRV would not result in exceeding design allowable stresses is determined. First the limiting component is found by successive application of the min_floats() function to the pairs (*fdes_q*, *fdes_qs*), (*prpv_1*, *fdes_tp*) and (*prpv_1*, *fdes_tps*) with the result in each case assigned to *prpv_1*. *Prpv_1* is then assigned the value:

prpv_1 = (*prpv_tp* * *prpv_1* / 100) + *prpv_tp*, where
prpv_tp is the pressure used in the SRV tail design stress analysis.

Reference 3 defines the maximum allowable SRV tail pipe water leg length (*wllmax*) as a function of RPV pressure (*prpv*) by:

wllmax(*prpv*) = (*dptp_dprpv* * (*prpv_1* – *prpv*)) / *dptp_dwll*

Since JAF has only one SRV tail pipe configuration, I have modified the reference 3 process to more directly obtain the values defining the STPLL. Rather than solving the above equation, then using the reference 3 'inversion' procedure to find the torus water levels corresponding to *wllmax*, I have manipulated the relation to obtain the maximum allowable reactor pressure as a function of water leg length:

pressure = *prpv_1* – (*wll* * (*dptp_dwll* / *dptp_dprpv*))

This is then solved for each of the values of *wll* defined in the plant specific input, *srvdl*[]. *Srvdl*[] is a structured variable with elements *wll* and *spwl* corresponding to water leg length and torus water level, respectively. The corresponding torus level is obtained by direct assignment from the input values, *srvdl*[].*spwl*. The respective assignments are to the elements pressure and level of another 1 x 5 array, *prpv_break*[].

Since the pressures determined above may be outside the range of pressure to be plotted on the EOP limit curve (0 psig to the maximum available indicated RPV pressure, *prpv_maxind*), another 1 x 5 array, *stpll*[], truncated on RPV pressure is defined. After initializing *stpll*[] to zero, indices pointing at locations with pressure less than *prpv_maxind* and 0 are found and the number of locations required to store water levels with non-zero pressure is determined.

If no pressures were above *prpv_maxind*, *prpv_break*[] elements with pressure at or above 0 are transferred directly to *stpll*[]. Otherwise, an interpolation is performed to determine the water level corresponding to *prpv_maxind* and this level and *prpv_maxind* are placed in the first element of *stpll*[]. This is followed by assignment of subsequent elements of *prpv_break*[] with non-zero values of pressure to *stpll*[].

If there were pressures below 0 in *prpv_break*[].pressure, an interpolation is performed to find the water level at which pressure is 0. The next element in *stpll*[] is then assigned pressure of 0 and this water level.

Finally, the points defining the STPLL are determined by truncating the curve defined by *stpll*[] at *mpspcwl*, with the values stored in the 1 x 5 array, *stpll_eop*. Initially, *stpll_eop*[] is assigned the values of *stpll*[]. The value of the index of the first element of *stpll_eop* for which level is above *mpspcwl* is then assigned to *hi_index* (if such an element exists). An interpolation is then performed to determine the pressure corresponding to *mpspcwl*. If another element of *stpll_eop*[] filled with (0, 0) exists, the contents are replaced by (0, *mpspcwl*).

If no elements of *stpll*[] contain levels above *mpspcwl*, then program execution falls through with *stpll_eop*[] the same as *stpll*[].

calc_stpll() is called with a statement of form:

calc_stpll(*interim*, *final*), where *interim* and *final* are the names of the output files to which results are to be written.

Heat Capacity Temperature Limit, Low-Pressure Endpoint Temperature: tsp_hctl_2

The HCTL low-pressure endpoint temperature, *tsp_hctl_2* is calculated as a function of torus water level for values between *hdco* (elevation of the downcomer openings) and *mpspcwl* (maximum pressure suppression primary containment water level). This temperature is the more limiting of *tsc_max*, the temperature capability of the torus (a plant-specific input) and *tsc_pcpl*[] (the temperature at which torus pressure will reach PCPL-A with all of the air or nitrogen in the containment in the torus airspace).

The function, calc_tsp_hctl_2() calculates the HCTL low-pressure endpoint temperature as a function of torus water level and assigns the results to the 1 x 11 array, *tsp_hctl_2*[]. Execution of the function begins with determination of the values *pg_sc*, *pnsc_sc* and *wlsp_lim*. *Pg_sc* is the partial pressure of water vapor in the torus airspace at *tsc_minop*, the minimum normal operating torus airspace temperature. *Pg_sc* is determined by calling the function psat(), with the value, *tsc_minop*. *Pnsc_sc* is the partial pressure of dry nitrogen or air in the torus airspace normalized to 100°F using the ideal gas law. It is determined by evaluation of the following relationship:

$pnsc_sc = ((psc_maxop + 14.7 - pg_sc) * 559.69) / (tsc_minop + 459.69)$, where
psc_maxop is the maximum normal operating torus pressure.

The lower limit for torus water level, *wlsp_lim* is determined by subtracting the elevation of the torus level instrument zero (reference value), *hsp_ref* from the elevation of the downcomer openings, *hdco*.

The following parameters are all evaluated as a function of water level by execution of the 'for (*index1* =0; *index1* < 11; ++ *index1*)' loop:

The partial pressure of dry air or nitrogen originating in the drywell, when in the torus airspace is determined next and assigned to the 1 x 11 array, *pnsc_dw*[]. The equation evaluated is:

pnsc_dw[] = ((*pdw_maxop* + 14.7) * 559.69 * *vdw*) / ((*tdw_minop* + 459.69) * *vtorus*[].*vsc_air*), where
pdw_maxop is the maximum normal operating drywell temperature,
vdw is the drywell free volume,
tdw_minop is the minimum normal operating drywell temperature and
vtorus[].*vsc_air* is the torus air volume as a function of water level.

The rate of change of pressure with temperature of all gas in the torus airspace is assigned to the 1 x 11 array, *dpsc_dtsc*[] by evaluation of the equation:

dpsc_dtsc[] = (*pnsc_sc* + *pnsc_dw*[]) / 559.69

The temperature at which torus airspace pressure will reach PCPL-A, *tsc_pcpl*[] (an 1 x 11 array) is determined as follows:

tsc_pcpl[] is initialized at *tsc_minop*.

The gas pressure (in psia) corresponding to this temperature, *p_gas*, is determined by evaluation of the equation:

p_gas = (*dpsc_dtsc*[] * (*tsc_pcpl*[] + 459.69)) + psat(*tsc_pcpl*[])

This pressure is then compared to PCPL-A (converted to an absolute pressure by adding 14.7 to *pcpl_to_calcs*[]), and if *p_gas* is smaller, *tsc_pcpl*[] is increased by 0.1 and *p_gas* at the higher temperature is determined. This process is repeated (by the 'while' loop) until *p_gas* is above (*pcpl_to_calcs*[] + 14.7). 0.1 is then subtacted from *tsc_pcpl*[] to obtain the highest temperature at which the gas in the torus will not exceed PCPL-A.

Next, *tsp_hctl_2*[] is determined by using the min_floats() function to find the lower of *tsc_pcpl*[] and *tsc_max*.

The value of *tsp_hctl_2*[] corresponding to the torus minimum normal operating water level is assigned to the variable, *tsp_hctl_2_lco* by this function. The method used is to test whether the input water level, *vtorus*[].*wlsp* is less than 13.9 ft. If it is, *tsp_hctl_2*[] is assigned to *tsp_hctl_2_lco* (i.e., the assignment will not be performed if the input water level is at or above 13.9 ft). This strategy makes use of the fact that successive elements of *vtorus*[] correspond to the minimum and maximum normal operating torus water levels, 13.88 ft and 14 ft, and no entries exist for levels between these values.

calc_tsp_hctl_2() is called with a statement of form:

calc_tsp_hctl_2(*interim*), where *interim* is the name of the output file to which results are to be written.

RPV Variables

Calculation of values of variables and curves associated with the RPV is controlled by the function, sequence_rpv(). The sequence of execution of functions (and a brief description) called by sequence_rpv() is:

1. init_rpv_values(): determines the enthalpy of water pumped from the torus by calling the function, hpt() with pressure equal to 500 psia and temperature equal to *t_sp_hctl_2_lco*. This variable is available for all subsequent functions called by sequence_rpv(). init_rpv_values() is called with a statement of form, init_rpv_value(*interim*), where *interim* is the name of the output file to which results are to be written
2. calc_marfp(): calculates the MARFP
3. calc_mnsred(): calculates the MNSRED
4. calc_msrp(): calculates the MSRP
5. calc_mrfp(): calculates the MRFP
6. calc_mcfi(): calculates the MCFI
7. calc_mcutl(): calculates the MCUTL
8. calc_mscrwl(): calculates the MSCRWL
9. calc_mzirwl(): calculates the MZIRWL

The method used to calculate each of the parameters associated with the RPV (based on reference 3) is described below.

sequence_rpv() is called with a statement of form:

sequence_rpv(*interim*, *final*), *interim* and *final* bring the output file names into the function for use in printing section headers in the files, and so they may be subsequently passed to other functions called by sequence_rpv().

MINIMUM ALTERNATE RPV FLOODING PRESSURE: MARFP

The MARFP is the lowest RPV pressure at which steam flow through open SRVs is sufficient to preclude clad temperature from exceeding 1500°F irrespective of whether the core is covered with water. MARFP is a function of the number of open SRVs, and its values are calculated using the function, calc_marfp().

Best-estimate SRV flowrate, *wsrv_be*, defined as 110% of the nameplate flow (*wsrv_name*, a plant-specific input) is determined by evaluation of the expression:

$$wsrv_be = 1.10 * wsrv_name$$

The reference pressure, *psrv_ref*, for best-estimate SRV flowrate is 103% of nameplate pressure, *psrv_name* (a plant-specific input) is determined by evaluation of the following expression:

$$psrv_ref = 1.03 * psrv_name$$

The specific SRV flowrate, *dwsrv_dprpv*, given critical flow through the valve is:

$$dwsrv_dprpv = wsrv_be / (psrv_ref + 14.7)$$

The maximum bundle steam flowrate, *wg_1500_max*, required to maintain peak clad temperature below 1500°F is determined as follows. *Wg_1500_max* is initialized at a value of 0. Then for each fuel type described in the plant-specific inputs, *wg_1500_max* is set to the larger of (the previously stored) *wg_1500_max* and *wg_1500*[] (plant-specific input as a function of fuel type stored in an 1 x 10 array) using the max_floats()

function, operating on *wg_1500_max* and *wg_1500*[]. This is done by iteration beginning with an index of 0 and ending with an index of 1 less than the number of fuel types, *nbr_fuel_types* (plant-specific input).

The minimum core flow required to maintain peak clad temperature below 1500°F, *wg_core_1*, is determined by multiplying *wg_1500_max* by the number of bundles in the core, *nbuns* (plant-specific input).

MARFP is then calculated and stored in the 1 x 7 array of structured elements, *marfp*[].*n_open* (number of open SRVs) and *marfp*[].*pressure*. A loop assigning (*index1* + 1) to *marfp*[].*n_open* and assigning *marfp*[].*pressure*,

marfp[].*pressure* = (*wg_core_1* / ((*index1* + 1) * *dwsrv_dprpv*)) – 14.7

is executed for indices ranging from 0 to 1 less than *nsrv_ads* (the number of SRVs assigned to ADS, plant-specific input).

calc_marfp() is called with a statement of form:

calc_marfp(*interim*, *final*)

MINIMUM NUMBER OF SRVS REQUIRED FOR EMERGENCY DEPRESSURIZATION: MNSRED

MNSRED is the larger of:

- The least number of SRVs which, if opened, will remove all decay heat from the core at a pressure sufficiently low that the ECCS with the lowest head will be capable of making up the SRV steam flow.
- The least number of SRVs which correspond to a MARFP sufficiently low that the ECCS with the lowest head will be capable of making up the SRV steam flow at the corresponding MARFP.

MNSRED is calculated by the function, calc_mnsred(), as described below.

Low pressure ECCS take suction on the torus, therefore the density of water pumped is:

rhof_sp = 1 / (*k3* * vpt(500, *tsp_hctl_2_lco*), where,
k3 is a constant loaded with generic data, and
500 psia is the pressure specified for this parameter by reference 3.

This density is then used to calculate the volumetric flow rate equivalent to *wg_core_1* (the value of which was determined by calc_marfp()) as follows:

wf_core_1 = *wg_core_1* / (60 * *rhof_sp*)

The RPV pressure at which LPCI is able to provide at least *wf_core_1*, *prpv_lpci* is determined as follows. Starting with the last element in the array specifying LPCI flow as a function of reactor pressure, *lpci_flow*[], *lpci_flow*[].*flow* is compared to *wf_core_1*. If *lpci_flow*[].*flow* is smaller than *wf_core_1*, the index into the array is decremented and the test of *lpci_flow*[].*flow* against *wf_core_1* repeated. Since the plant-specific *lpci_flow*[] values are ordered from high flow to low flow, this loop is exited with the index variable pointing at the first element in *lpci_flow*[] with *lpci_flow*[].*flow* at least as large as *wf_core_1*. This index is then used to retrieve the value of pressure, *lpci_flow*[].*pressure* which is assigned to *prpv_lpci*.

The code which determines the RPV at which LPCS is able to provide at least *wf_core_1*, *prpv_cs* is determined using the same process as was used for LPCI, with the exception that the plant-specific input is contained in the array, *cs_flow*[] and the result is assigned to *prpv_cs*.

The lowest RPV pressure at which a low head ECCS pump can inject flow sufficient to maintain MARFP is the lower of *prpv_lpci* and *prpv_cs*. The value is determined using the min_floats() function with the result assigned to *prpv_1*.

The number of open SRVs required to maintain MARFP is obtained by finding the first element of the *marfp*[] array for which the value of the pressure element is lower than *prpv_1* (*marfp*[] is ordered from high to low RPV pressure). The number of open SRVs corresponding to this pressure, stored in *marfp*[].*n_open* is then assigned to *nsrv_1*.

The enthalpy of steam discharged through SRVs, *hg_srv*, is obtained by passing the hgsat_by_p() function the absolute pressure corresponding to the lowest SRV lift setpoint, *psrv_lift* (plant-specific input).

Steam flow required to remove decay heat two minutes following reactor shutdown, *wg_core_2*, is determined by evaluating the expression:

wg_core_2 = (*k1* * *fqdh_2* * *qrx_rated*) / (*hg_srv* – *hf_sp*), where
k1 is a conversion constant input with generic data,
fqdh_2, the decay heat fraction two-minutes following shutdown is input with generic data, and
qrx_rated is rated thermal power, input with plant-specific data.

The corresponding volumetric injection flow is:

wf_core_2 = *wg_core_2* / (60 * *rhof_sp*)

The RPV pressure at which low-head ECCS can inject *wf_core_2* is determined using the same process as was followed with *wf_core_1*, with *wf_core_1* replaced by *wf_core_2*. The results for LPCI and core spray are again placed in *prpv_lpci* and *prpv_cs*, respectively. The lowest pressure at which a low-head ECCS can remove two-minute decay heat, *prpv_2*, is then determined using the function, min_floats() operating on *prpv_lpci* and *prpv_cs*.

The number of open SRVs required for a low-head ECCS pump to remove two-minute decay heat, *nsrv_2*, is:

nsrv_2 = *wg_core_2* / (*dwsrv_dprpv* * (*prpv_2* + 14.7))

MNSRED is then obtained by assignment from the result of passing *nsrv_1* and *nsrv_2* to max_int().

calc_mnsred() is called with a statement of form:

calc_mnsred(*interim*, *final*)

MINIMUM SRV RE-OPENING PRESSURE: MSRP

MSRP is the differential pressure between the RPV and containment required for a SRV to open. It is a function of SRV type, and the value, *msrp*, is assigned the value of SRV re-opening pressure, *srv_open*, input with generic data. The value is obtained from the function, calc_msrp() which is called with a statement of form:

calc_msrp(*interim*)

MINIMUM RPV FLOODING PRESSURE: MRFP

MRFP is the larger of the MSRP or the lowest differential pressure between the RPV and torus at which flow through MNSRED SRVs will remove ten-minute decay heat from the reactor. The value of this parameter is assigned to the variable, *mrfp* by the function calc_mrfp().

The pressure at which MNSRED SRVs will remove ten-minute decay heat, *prpv_10*, is determined by initializing *prpv_10* at 0 (which assures that left_side is initially smaller than right_side), and evaluating the quantities:

left_side = *mnsred* * *dwsrv_dprpv* * (*prpv_10* + 14.7)
right_side = (*k1* * *fqdh_10* * *qrx_rated*) / hgsat_by_p(*prpv_10* + 14.7) – *hf_sp*), where
fqdh_10, the ten-minute decay heat fraction is input with generic data.

Then, while the *left_side* is smaller than the *right_side*, this process is repeated with *prpv_10* raised by 0.1 at each step. When *left_side* is no longer smaller than *right_side*, the loop is exited and 0.1 is subtracted from *prpv_10* to obtain the lowest pressure at which MNSRED are capable of removing ten-minute decay heat.

Mrfp is now obtained by passing max_floats() the values of *msrp* and *prpv_10*, to find which is larger.

calc_mrfp() is called with a statement of form:

calc_mrfp(*interim*, *final*)

MINIMUM CORE FLOODING INTERVAL: MCFI

MCFI is the time required to flood the RPV to the top of active fuel (beginning with an empty vessel) with RPV pressure at MRFP and at least MNSRED SRVs open. MCFI is a function of the number of SRVs open, and is evaluated by the function, calc_mcfi(), for values ranging from *mnsred* to *nsrv_ads* with the results assigned to the 1 x 7 array, *mcfi*[].

The heat capacity of the steel in the RPV and Reactor Water Recirculation loops below the top of active fuel (TAF) is:

crpv_taf = *c_steel* * *mrpv_taf*, where
c_steel, the specific heat of steel is input from generic data, and
mrpv_taf, the mass of RWR loops and reactor internals below TAF is input from plant-specific data.

The heat capacity of the core is:

ccore = (*c_clad* * *mclad*) + (*c_fuel* * *mfuel*), where
c_clad, the specific heat of the fuel clad, *c_fuel*, the specific heat of the fuel, *mclad*, the mass of clad and *mfuel*, the mass of fuel are input from plant-specific data.

RPV temperature at the lowest SRV setpoint, *tsrv*, and MRFP, *tmrfp* are obtained by calling the tsat() function with *psrv_lift* and *mrfp* (converted to absolute pressure), respectively.

Next, the quantities, *z1* and *z3*, defined in reference 3, are evaluated as follows:

z1 = (*k1* * *qrx_rated* * (*fqdh_10* – *fqdh_100*)) / (3 * (hgsat_by_p(*mrfp* + 14.7) – *hf_sp*))
z3 = (*ccore* * (1500 – *tmrfp*) + *crpv_taf* * (*tsrv* – *tmrfp*)) / (*hf_sp* – hgsat_by_p(*mrfp* + 14.7))
z3 = *z3* – (*vrpv_taf* / vgsat_by_p(*mrfp* + 14.7))

Finally, *z2* (as defined in reference 3) and *mcfi*[] are determined by execution of a loop (*nsrv_ads* – *mnsred*) times. Initially, *mcfi*[].*n_open* is assigned the value obtained by converting *mnsred* to an integer and adding the value of the index into the array. *Z2* is then calculated as follows:

z2 = *mcfi*[].*n_open* * *dwsrv_dprpv* * (*mrfp* + 14.7)
z2 = *z2* – ((*k1* * *fqdh_10* * *qrx_rated*) / (hgsat_by_p(*mrfp* + 14.7) – *hf_sp*))

The time element of *mcfi*[] is determined as follows:

mcfi[].*time* = 30 * (sqrt((*z2* * *z2*) – (4 * *z1* * *z3*)) – *z2*) / *z1*

The *mcfi*[].*time* element applicable to *nsrv_ads* open SRVs is assigned to the variable, *mcfi_nads*. This variable is used for calculation of the MCUTL. This is done using an assignment statement which is only executed when *mcfi*[].*n_open* is equal to *nsrv_ads*.

calc_mcfi() is called with a statement of form:

calc_mcfi(*interim*, *final*)

MAXIMUM CORE UNCOVERY TIME LIMIT: MCUTL

MCUTL is the greatest amount of time that the clad temperature of the hottest fuel rod will not exceed 1500°F with the core completely uncovered and uncooled. MCUTL is evaluated by the function, calc_mcutl(), as a function of time after shutdown, with results assigned to the 1 x 9 array, *mcutl*[].

Since time that fuel can be uncovered without peak clad temperature exceeding 1500°F is a function of fuel type, the limiting values for the core must be determined. To do this, a variable which will hold the minimum uncovery times for the core, *mcutl_min*[] is initialized with values obtained from *t_to_1500*[] for the first fuel type defined in plant-specific inputs. Subsequent to this, *mcutl_min*[] is assigned the value returned by the min_floats() function operating on the previously stored *mcutl_min*[] and *t_to_1500*[] for the rest of the defined fuel types. When this iteration is complete, *mcutl_min*[] contains uncovery time limits as a function of time after shutdown for the most-limiting fuel type in the core.

Mcutl[] is next initialized with values of individual elements, *t_sd* and *t_cu_15* set equal to zero.

Since MCUTL = 0 for *t_sd* less than *mcfi_nads* (reference 3), indices into *mcutl_min*[] for which *mcutl_min*[*n*].*t_sd* < *mcfi_nads* < *mcutl_min*[*n* + 1].*t_sd* are determined as follows:

An index into *mcutl_min*[], *index1* is initialized at 0, then *mcutl_min*[].*t_sd* is compared to *mcfi_ads*. While *mctul*[].*t_sd* is smaller than *mcfi_ads*, *index1* is raised by 1 and the test repeated. This loop is exited with *index1* pointing at the first element in *mcutl_min*[] with *t_sd* at or above *mcfi_nads*. A pointer to the immediately preceding element of *mcutl_min*[] is defined by assigning *index2* the value, *index1* – 1.

Mcutl[index2].t_sd is set to *mcfi_nads* and the interpolate() function is called with parameters, *mcutl_min[index2].t_sd*, *mcutl_min[index1].t_sd*, *mcfi_ads*, *mcutl_min[index2].t_cu_15* and *mcutl_min[index1].t_cu_15* to obtain *t_cu_15* corresponding to *mcfi_nads*. This value is assigned to *mcutl[index2].t_cu_15*.

The remainder of the *mcutl*[] array is filled by direct assignment from the corresponding elements of the *mcutl_min*[] array.

calc_mcutl() is called with a statement of form:

calc_mcutl(*final*)

MINIMUM STEAM COOLING RPV WATER LEVEL: MSCRWL

The MSCRWL is the lowest RPV water level at which the covered portion of the core will generate sufficient steam to prevent any clad temperature in the uncovered part of the core from exceeding 1500°F with injection of 100°F water into the core. MSCRWL is evaluated using calc_mscrwl() as a function of fuel type and the highest value is assigned to *mscrwl_max*.

Initially, *mscrwl_max* is assigned the level of the bottom of active fuel, *wlrpv_baf*, a plant-specific data input. Then a loop is executed which evaluates the MSCRWL for each fuel type assigning the result of the following expression to *mscrwl*:

mscrwl = *wlrpv_baf* + ((*fafl_15*[] * *lfuel*[]) / 100)

The result of evaluating *mscrwl* and *mscrwl_max* using the max_floats() function is then assigned to *mscrwl_max*. When this loop completes execution, *mscrwl_max* contains the MSCRWL for the core.

calc_mscrwl() is called with a statement of form:

calc_mscrwl(*final*)

MINIMUM ZERO-INJECTION RPV WATER LEVEL: MZIRWL

The MZIRWL is the lowest RPV water level at which the covered portion of the core will generate sufficient steam to prevent any clad temperature in the uncovered part of the core from exceeding 1800°F with no RPV injection. MZIRWL is evaluated using calc_mzirwl() as a function of fuel type and the highest value is assigned to *mzirwl_max*.

Initially, *mzirwl_max* is assigned the level of the bottom of active fuel, *wlrpv_baf*, a plant-specific data input. Then a loop is executed which evaluates the MZIRWL for each fuel type assigning the result of the following expression to *mzirwl*:

mzirwl = *wlrpv_baf* + ((*fafl_18*[] * *lfuel*[]) / 100)

The result of evaluating *mzirwl* and *mzirwl_max* using the max_floats() function is then assigned to *mzirwl_max*. When this loop completes execution, *mzirwl_max* contains the MZIRWL for the core.

calc_mzirwl() is called with a statement of form:

calc_mzirwl(*final*)

Heat Capacity Temperature Limit: HCTL

The HCTL is the highest suppression pool temperature from which emergency RPV depressurization will not raise:

- suppression chamber temperature above the maximum temperature capability of the structure or contained equipment which may be required to operate with the reactor pressurized, or,
- suppression chamber pressure above PCPL-A,

while the rate of energy transfer from the RPV to the containment is greater than the capacity of the containment vent.

The HCTL is a function of RPV pressure and torus water level.

HCTL is calculated using the function calc_hctl(), which is supported by the functions calc_n_values() and calc_qdh(), as well as previously described functions which return thermodynamic properties of water and steam. Calc_n_values() and calc_qdh() will be described first, followed by a description of calc_hctl().

calc_n_values()

This function returns the number of points to be included in the HCTL calculation when provided the high and low endpoint pressures, *psrv_lift* and *mrfp*, respectively. Increments of pressure chosen for the calculation are 10 psi, for pressure above 200 psig and 5 psi, for pressure at or below 200 psig.

The function initially sets *midpoint* equal to 200, then determines the number of points needed between the high-pressure endpoint (called *big* in the function) and the midpoint as follows:

The difference between *big* and *midpoint* is assigned to *difference*. The result of dividing *difference* by 10 is then assigned to *quotient*. The portion of *quotient* to the left of the decimal point is determined by passing *quotient* to the library function int() and assigning the result to *modulus*. The portion of *quotient* to the right of the decimal point is obtained by subtracting *modulus* from *quotient* and assigning the result to *remainder*.

The function then uses the assumption that *psrv_lift* is a multiple of 5, so that *remainder* can be tested to determine whether *psrv_lift* is also a multiple of 10. If *psrv_lift* is not a multiple of 10 (i.e., *remainder* is not equal to 0) 1 is added to *modulus* to account for the pressure point which is not a multiple of 10 and the result is assigned to *answer*. Otherwise, *psrv_lift* is a multiple of 10 and *modulus* is assigned directly to *answer*.

The function next sets *big* to 200, then determines the number of points between *big* (the original midpoint) and the low-pressure endpoint (called *little* in the function), inclusive of the endpoints as follows:

The difference between *big* and *little* is assigned to *difference*. The result of dividing *difference* by 5 is then assigned to *quotient*. Finally, 1 is added to *quotient* (to account for both endpoints) which is added to *answer*, with the result returned to *answer*.

The value of *answer* is returned by the function call, and is the number of values of pressure to be used in calculation of the HCTL.

calc_n_values() is called with an assignment statement of form:

a = calc_n_values(*big*, *little*)

calc_qdh()

This function returns the percentile decay heat [Btu/min.-%] given time after shutdown, referred to internally as *time*.

The function initializes an index, *index1* at 0, and compares *time* with the time element of *sd_pwr*[]. Since *sd_pwr*[] is ordered from low to high, *index1* is incremented and the test repeated until *sd_pwr*[*index1*].*time* is larger than *time*.

If *time* is exactly equal to a stored value, the corresponding power (obtained from *sd_pwr*[*index1* - 1].*power*, since *index1* was incremented when *sd_pwr*[*index1*].*time* = *time* in the preceding loop) is assigned to *pwr_now*. Otherwise, time must be less than *sd_pwr*[*index1*].*time* (which is tested), and *pwr_now* is found by interpolation where the parameters passed to interpolate() are:

t_bfr = *sd_pwr*[*index1* - 1].*time* and
pwr_bfr = *sd_pwr*[*index1* – 1].*power* are time and power at the stored time just before the input time,
t_aftr = *sd_pwr*[*index1*].*time* and
pwr_aftr = *sd_pwr*[*index1*].*power* are time and power at the stored time just after the input time, and
time is the time for which a value of power is to be returned.

The value returned by the interpolate() function is assigned to *pwr_now*.

Percentile power is next determined by evaluating:

qrx_1 = *qrx_rated* * *k2* / 100

to find the number of Btu/min. corresponding to 1% of rated power, and multiplying by *pwr_now* to obtain power in Btu/min:

pwr_now = *pwr_now* * *qrx_1*.

This value of *pwr_now* is returned the function.

calc_qdh() is called with an assignment statement of form:

a = calc_qdh(*time*)

calc_hctl()

After declaring variables local to calc_hctl(), a block of memory of sufficient size to store intermediate results for pressures ranging from 1500 psig to 200 psig in 10 psig increments and 200 psig to 50 psig in 5 psig increments is allocated (and initialized with values of 0). This memory is accessed through a pointer, *res_ptr*. The memory stored is for the *hctl_int* structure defined in the function.

Since storage was allocated for up to 160 intermediate values, the actual number to be used is obtained by assigning the result of calling calc_n_values() with (*psrv_lift*, *mrfp*) to *n_values*.

Best-estimate SRV flowrate, *wsrv_be* is obtained by evaluating:

wsrv_be = (1.10 * *wsrv_name*) / 60

The reference pressure for this flowrate, *psrv_ref*, is:

psrv_ref = 1.03 * *psrv_name*

The SRV specific flowrate, *dwsrv_dprpv*, is then (given critical flow):

dwsrv_dprpv = *wsrv_be* / (*psrv_ref* + 14.7)

RPV heat capacity, *crpv*, is obtained by evaluating the expression:

crpv = (*c_steel* * *mrpv*) + (*c_clad* * *mclad*) + (*c_fuel* * *mfuel*)

RPV volume, *vrpv*, is:

vrpv = (vfsat_by_p(*psrv_lift* + 14.7) * *mf_rpv_hot*) + (vgsat_by_p(*psrv_lift* + 14.7) * *mg_rpv_hot*)

A loop is now entered, in which HCTL is calculated as a function of torus water level. The calculation is performed for those water levels for which air and water volumes were entered in plant-specific inputs, therefore the loop is executed 11 times, with iteration number tracked by *index1*.

The first action performed within the loop is to set an offset, *to_ptr* into the array used to store intermediate results. The value assigned is *index1* * *n_values*.

Now, initial RPV state variables are assigned as follows:

prpv_a = *psrv_lift* + 14.7 (SRV lift pressure in psia)
prpv_g = *psrv_lift* (SRV lift pressure in psig)
(*res_ptr* + *to_ptr*)->*prpv* = *prpv_g*
(*res_ptr* + *to_ptr*)->*trpv* = tsat(*prpv_a*)
(*res_ptr* + *to_ptr*)->*hf_rpv* = hfsat_by_p(*prpv_a*)
(*res_ptr* + *to_ptr*)->*hg_rpv* = hgsat_by_p(*prpv_a*)
(*res_ptr* + *to_ptr*)->*vf_rpv* = vfsat_by_p(*prpv_a*)
(*res_ptr* + *to_ptr*)->*vg_rpv* = vgsat_by_p(*rpv_a*)
(*res_ptr* + *to_ptr*)->*wsrv* = *mnsred* * *dwsrv_dprpv* * *prpv_a*
(*res_ptr* + *to_ptr*)->*mf_rpv* = *mf_rpv_hot*
(*res_ptr* + *to_ptr*)->*mg_rpv* = *mg_rpv_hot*
(*res_ptr* + *to_ptr*)->*erpv* = ((*res_ptr* + *to_ptr*)->*trpv* * *crpv*)
+ ((*res_ptr* + *to_ptr*)->*hf_rpv* * (*res_ptr* + *to_ptr*)->*mf_rpv*)
+ ((*res_ptr* + *to_ptr*)->*hg_rpv* * (*res_ptr* + *to_ptr*)->*mg_rpv*)
(*res_ptr* + *to_ptr*)->*t* = 2
(*res_ptr* + *to_ptr*)->*qdh* = calc_qdh((*res_ptr* + *to_ptr*)->*t*)

Reference 3 specifies calculation of torus water mass using the assumption torus water is at saturation at *tsp_scram*. Since the EOPs are structured to maintain torus pressure at or above atmospheric (nominally 14.67 psia, recognizing that torus pressure is referenced to reactor building pressure which is slightly subatmospheric), this calculation will use subcooled water properties at 15 psia to obtain torus water mass.

(res_ptr + to_ptr)->mf_sp = vtorus[index1].vsp / vpt(15, *tsp_scram*)

Now, the effect of depressurizing the RPV (along the HCTL) on RPV parameters is evaluated (in a loop controlled by *index2* over the range from 1 to *n_values*).

First the offset, *to_ptr*, for the pointer to storage for results of a step is obtained from:

*to_ptr = (index1 * n_values) + index2*

The pressures (gage and absolute) applying to the state point are obtained by subtracting 10 from the previous values if *prv_g* was previously at or above 210, or 5 otherwise.

Now, fluid properties at the new pressure (*psrv_g* or *psrv_a*, as appropriate) may be determined (relations evaluated are the same as before):

(res_ptr + to_ptr)->prpv = prpv_g
(res_ptr + to_ptr)->trpv = tsat(*prpv_a*)
(res_ptr + to_ptr)->hf_rpv = hfsat_by_p(*prpv_a*)
(res_ptr + to_ptr)->hg_rpv = hgsat_by_p(*prpv_a*)
(res_ptr + to_ptr)->vf_rpv = vfsat_by_p(*prpv_a*)
(res_ptr + to_ptr)->vg_rpv = vgsat_by_p(*rpv_a*)

SRV specific flowrate is also determined as before, evaluating at the new pressure, *psrv_a*

*(res_ptr + to_ptr)->wsrv = mnsred * dwsrv_dprpv * prpv_a*

Next, parameters *z1*, *z2*, *z3*, *z4*, *z5* and *z6* as specified in reference 3 are evaluated:

(res_ptr + to_ptr)->z1 = vrpv / (res_ptr + to_ptr)->vg_rpv
(res_ptr + to_ptr)->z2 = (res_ptr + to_ptr)->vf_rpv / (res_ptr + to_ptr)->vg_rpv
(res_ptr + to_ptr)->z3 = (res_ptr + to_ptr - 1)->mg_rpv
+ (res_ptr + to_ptr - 1)->mf_rpv - (res_ptr + to_ptr)->z1
(res_ptr + to_ptr)->z4 = (res_ptr + to_ptr - 1)->qdh
/ (res_ptr + to_ptr - 1)->wsrv - (res_ptr + to_ptr - 1)->hg_rpv
*(res_ptr + to_ptr)->z5 = (res_ptr + to_ptr - 1)->erpv - (crpv * (res_ptr + to_ptr)->trpv)*
*- ((res_ptr + to_ptr)->z1 * (res_ptr + to_ptr)->hg_rpv)*
(res_ptr + to_ptr)->z6 = (res_ptr + to_ptr)->hf_rpv - ((res_ptr + to_ptr)->z2
** (res_ptr + to_ptr)->hg_rpv) + ((res_ptr + to_ptr)->z4*
** (1 - (res_ptr + to_ptr)->z2))*

With these relations, RPV state parameters and torus water mass are determined as follows:

(res_ptr + to_ptr)->mf_rpv = ((res_ptr + to_ptr)->z5 + ((res_ptr + to_ptr)->z3
** (res_ptr + to_ptr)->z4)) / (res_ptr + to_ptr)->z6*
(res_ptr + to_ptr)->mg_rpv = (res_ptr + to_ptr)->z1 - ((res_ptr + to_ptr)->z2

```
    * (res_ptr + to_ptr)->mf_rpv)
(res_ptr + to_ptr)->erpv = ((res_ptr + to_ptr)->trpv * crpv) + ((res_ptr + to_ptr)->hf_rpv
    * (res_ptr + to_ptr)->mf_rpv) + ((res_ptr + to_ptr)->hg_rpv
    * (res_ptr + to_ptr)->mg_rpv)
(res_ptr + to_ptr)->dmg_srv = (((res_ptr + to_ptr)->z2 - 1) * (res_ptr + to_ptr)->mf_rpv
    + (res_ptr + to_ptr)->z3
(res_ptr + to_ptr)->dt = (res_ptr + to_ptr)->dmg_srv / (res_ptr + to_ptr)->wsrv
(res_ptr + to_ptr)->t = (res_ptr + to_ptr - 1)->t + (res_ptr + to_ptr)->dt
(res_ptr + to_ptr)->qdh = calc_qdh((res_ptr + to_ptr)->t)
(res_ptr + to_ptr)->mf_sp = (res_ptr + to_ptr - 1)->mf_sp + (res_ptr + to_ptr)->dmg_srv
```

Derivation of the above relationships is as follows, where (*res_ptr + to_ptr*)->*z1* notation has been replaced by $z1_2$ notation:

(*res_ptr + to_ptr*)->*z1* = $z1_2$
(*res_ptr + to_ptr*)->*z2* = $z2_2$
(*res_ptr + to_ptr*)->*z3* = $z3_2$
(*res_ptr + to_ptr*)->*z4* = $z4_2$
(*res_ptr + to_ptr*)->*z5* = $z5_2$
(*res_ptr + to_ptr*)->*z6* = $z6_2$
(*res_ptr + to_ptr*)-> *vf_rpv* = vf_2
(*res_ptr + to_ptr*)-> *vg_rpv* = vg_2
(*res_ptr + to_ptr* - 1)->*mf_rpv* = mf_1
(*res_ptr + to_ptr* - 1)->*mg_rpv* = mg_1
(*res_ptr + to_ptr*)->*mf_rpv* = mf_2
(*res_ptr + to_ptr*)->*mg_rpv* = mg_2
(*res_ptr + to_ptr* - 1)->*hg_rpv* = hg_1
(*res_ptr + to_ptr*)->*hg_rpv* = hg_2
(*res_ptr + to_ptr* - 1)->*erpv* = $erpv_1$
(*res_ptr + to_ptr*)->*trpv* = $trpv_2$
(*res_ptr + to_ptr*)->*hf_rpv* = hf_2
(*res_ptr + to_ptr*)->*erpv* = $erpv_2$
(*res_ptr + to_ptr*)->*dmg_srv* = dmg_srv_2
(*res_ptr + to_ptr* - 1)->*qdh* = qdh_1
(*res_ptr + to_ptr* - 1)->*wsrv* = $wsrv_1$

Given:

vrpv = (vg * mg) + (vf * mf) at any RPV pressure (i.e., total RPV volume is the sum of the (vapor phase specific volume times the mass of vapor) plus (liquid phase specific volume times the mass of liquid))

permits equating the sums of gas and liquid volumes at two successive pressures:

$$(vg_1 * mg_1) + (vf_1 * mf_1) = (vg_2 * mg_2) + (vf_2 * mf_2) = vrpv$$

Solving for mg_2:

$$mg_2 = [vrpv - (vf_2 * mf_2)] / vg_2 = (vrpv / vg_2) - [(vf_2 / vg_2) * mf_2], \text{ therefore,}$$

$$mg_2 = z1_2 - (z2_2 * mf_2)$$

The mass of fluid passed through the SRVs between state 1 and state 2 is:

$dmg_srv_2 = (mf_1 + mg_1) - (mf_2 + mg_2)$, so substituting mg_2 from before,

$$dmg_srv_2 = (mf_1 + mg_1) - \{mf_2 + (vrpv / vg_2) - [(vf_2 / vg_2) * mf_2]\}$$
$$dmg_srv_2 = mf_1 + mg_1 + \{mf_2 * [(vf_2 / vg_2) - 1]\} - (vrpv / vg_2)$$
$$dmg_srv_2 = \{[(vf_2 / vg_2) - 1] * mf_2\} + [mf_1 + mg_1 - (vrpv / vg_2)], \text{ therefore,}$$

$$dmg_srv_2 = [(z2_2 - 1) * mf_2] + z3_2$$

An energy balance on the RPV allows determination of mf_2. From state parameters evaluated at state 2

$$erpv_2 = (trpv_2 * crpv) + (hf_2 * mf_2) + (hg_2 * mg_2)$$

Energy at state 2 can also be related to energy at state 1 as follows:

$$erpv_2 = erpv_1 - \{[(mf_1 + mg_1) - (mf_2 + mg_2)] * hg_1\} + (qdh_1 * \{[(mf_1 + mg_1) - (mf_2 + mg_2)] / wsrv_1\})$$

where steam discharged through open SRVs is assumed to be at constant enthalpy, hg_1, SRV flowrate is assumed constant at $wsrv_1$, and decay heat is assumed constant at qdh_1. Therefore,

$$(trpv_2 * crpv) + (hf_2 * mf_2) + (hg_2 * mg_2) = erpv_1 - \{[(mf_1 + mg_1) - (mf_2 + mg_2)] * hg_1\} + (qdh_1 * \{[(mf_1 + mg_1) - (mf_2 + mg_2)] / wsrv_1\})$$

Rearranging terms and substituting $\{(vrpv / vg_2) - [(vf_2 / vg_2) * mf_2]\}$ for mg_2:

$$(hf_2 * mf_2) + (hg_2 * \{(vrpv / vg_2) - [(vf_2 / vg_2) * mf_2]\}) = erpv_1 - (trpv_2 * crpv) + \{[(qdh_1 / wsrv_1) - hg_1] * (mf_1 + mg_1 - mf_2 - \{(vrpv / vg_2) - [(vf_2 / vg_2) * mf_2]\})\}$$

$$(mf_2 * \{hf_2 - [(vf_2 / vg_2) * hg_2]\}) + [(vrpv / vg_2) * hg_2] = erpv_1 - (trpv_2 * crpv) + \{[(qdh_1 / wsrv_1) - hg_1] * [mf_1 + mg_1 + \{[(vf_2 / vg_2) - 1] * mf_2\} - (vrpv / vg_2)]\}$$

$$(mf_2 * \{hf_2 - [hg_2 * (vf_2 / vg_2)]\}) - \{[(qdh_1 / wsrv_1) - hg_1] * [(vf_2 / vg_2) - 1] * mf_2\} = erpv_1 - [(vrpv / vg_2) * hg_2] - (trpv_2 * crpv) + \{[(qdh_1 / wsrv_1) - hg_1] * [mf_1 + mg_1 - (vrpv / vg_2)]\}$$

Expressing this in terms of $z1_2$, $z2_2$ and $z4_2$:

$$mf_2 * \{hf_2 - (hg_2 * z2_2) - [z4_2 * (z2_2 - 1)]\} = erpv_1 - (crpv * trpv_2) - (z1_2 * hg_2) + [z4_2 * (mg_1 + mf_1 - z1_2)]$$

And in terms of $z3_2$, $z4_2$, $z5_2$ and $z6_2$:

$$mf_2 * z6_2 = z5_2 + (z3_2 * z4_2)$$

Solving for mf_2:

$$mf_2 = [z5_2 + (z3_2 * z4_2)] / z6_2$$

The time interval between consecutive pressure steps, dt_2, is determined by dividing mass discharge through the SRV(s), dmg_srv_2, by SRV flowrate, $wsrv_2$. Time of the statepoint, t_2, is then time at the previous

statepoint, $t_1 + dt_2$. Decay heat is determined by calling calc_qdh() with the present time, t_2. Torus water mass is calculated by adding the mass of steam discharged through the open SRV(s), dmg_srv_2, to torus water mass at the previous time step, mf_sp_1.

When the loop evaluating RPV state parameters and torus water mass completes, the pointer into memory is reset to point at the last stored statepoint with the statement:

to_ptr = ((*index1* + 1) * *n_values*) -1

Final torus temperature is obtained by assignment from the previously calculated array, *tsp_hctl_2*[], and energy is determined by multiplying torus water mass by enthalpy (estimated using the relation h = T – 32, reference 3) as follows:

esp = *mf_sp* * (*tsp_hctl_2*[] – 32)

Next the value which will point at the first element of the array storing the statepoints for the current torus water level is determined by evaluation of:

torus_calc_base = *index1* * *n_values*

Now it is possible to determine torus energy and temperature for each successive step as RPV pressure is raised. This is done through a loop beginning at the memory location just before that which contains the torus final state (for a given water level) and proceeding backwards through the array. Values are determined by accessing the location in memory pointed to by *to_ptr*. This pointer is set equal to *index2*, which is the variable controlled by the loop.

Torus energy at a given RPV pressure, is obtained by subtracting the energy added by the steam discharged through an open SRV(s) from torus energy at the statepoint corresponding to the next higher RPV pressure:

(*res_ptr* + *to_ptr*)->*esp* = (*res_ptr* + *to_ptr* + 1)->*esp*
– ((*res_ptr* + *to_ptr* + 1)->*hg_rpv* * (*res_ptr* + *to_ptr* + 1)->*dmg_srv*)

Torus temperature at a given pressure is obtained by dividing torus energy for that statepoint by the torus water mass and adding 32 (the estimation that h = T –32 from reference 3 is used):

(*res_ptr* + *to_ptr*)->*tsp* = (*res_ptr* + *to_ptr*)->*esp* / (*res_ptr* + *to_ptr*)->*mf_sp* + 32

After exiting the preceding loop, the HCTL high-pressure endpoint temperature is stored for use in subsequent calculations. The offset to be used to determine the pointer to the appropriate element in the storage array is obtained by evaluating:

to_ptr = *index1* * *n_values*

and completing the assignment:

tsp_hctl_1[*index1*] = (*res_ptr* + *to_ptr*)->*tsp*

The high-pressure endpoint temperatures corresponding to minimum and maximum normal operating torus water level, *tsp_hctl_1_lo* and *tsp_hctl_1_hi*, are assigned by evaluation of conditional statements. These

evaluations depend on the characteristic of the input torus water volumes that no data are stored for torus water levels between minimum and maximum normal operating water level.

These calculations are then repeated for each input torus water level.

calc_hctl() is called with a statement of form:

calc_hctl(*interim*, *final*), *interim* and *final* bring the output file names into the function.

Boron Injection Variables

Calculation of values of variables and curves associated with injection of boron into the RPV is controlled by the function, sequence_boron(). The sequence of execution of functions (and a brief description) called by sequence_boron() is:

1. init_boron_values(): determines the value of a number of parameters which remain fixed through the remainder of the calculation.
2. calc_csbw(): calculates the CSBW
3. calc_hsbw(): calculates the HSBW
4. calc_biit: calculates the BIIT

sequence_boron() is called with a statement of form:

sequence_boron(*interim*, *final*), *interim* and *final* bring the output file names into the function.

The method used to calculate each of the parameters associated with boron injection (based on reference 3) is described below. These methods result in calculated SLC tank level changes which are somewhat larger than actually required to inject the CSBW and HSBW, as well as somewhat longer time to inject the HSBW than actually required. This is due to use of subcooled water properties (at 1000 psia and *tsk*) to obtain the specific volume of the fluid in the SLC tank without correcting for specific gravity of the solution.

init_boron_value()

Init_boron_values() calculates values of the following parameters for use by calc_csbw(), calc_hsbw() and calc_biit().

The weight fraction of the B_{10} isotope in naturally occurring boron, *fb10_nat_wt*, is:

$$fb10_nat_wt = (fb10_nat * mb10) / ((fb10_nat * mb10) + (fb11_nat * mb11))$$

The nuclidic abundance of B_{11} as a fraction of all boron in the SLC tank, *fb11_slc*, is:

$$fb11_slc = 1 - fb10_slc$$

The weight fraction of the B_{10} isotope as a fraction of all boron in the SLC tank, *fb10_slc_wt*, is:

$$fb10_slc_wt = (fb10_slc * mb10) / ((fb10_slc * mb10) + (fb11_slc * mb11))$$

The surface area of the SLC tank [ft^2] is:

slc_tk_area = (π / 4) * (*slc_tk_dia*)^2

The change in indicated tank level per change in solution level [%/in.] is:

delta_tk_lvl = *slc_tk_lvl_out_span* / *slc_tk_lvl_in_span*

The specific volume of water in the SLC tank, *vf_slc* is obtained by passing the vpt() function the pair of values (1000, *tslc*).

init_boron_values() is called with a statement of form:

init_boron_values(*interim*), *interim* brings the output file name into the function.

COLD SHUTDOWN BORON WEIGHT: CSBW

The CSBW is that amount of soluble boron which, if injected into the RPV and mixed uniformly, will maintain the reactor shutdown under all conditions. CSBW is calculated by the function, calc_csbw().

The concentration of boron in the SLC tank, *xb_cld*, (ppm) required to obtain the CSBW is:

xb_cld = *xb_cld_nat* * *fb10_nat_wt* / *fb10_slc_wt*

This corresponds to a mass, *csbw*, (lbm) of boron of:

csbw = *xb_cld* * *mf_rpv_cld* / 1e6

The change in tank level corresponding to this mass of boron [in.] is:

delta_tk_cld = (*csbw* * 1e6 * *vf_slc* * 12 in./ft) / (*xb_slc* * *slc_tk_area*)

The change in indicated tank level corresponding to csbw [%] is:

delta_slc_cld = *delta_tk_cld* * *delta_tk_lvl*

calc_csbw() is called with a statement of form:

calc_csbw(*interim*, *final*), *interim* and *final* bring the output file names into the function.

HOT SHUTDOWN BORON WEIGHT: HSBW

The HSBW is least weight of soluble boron which, if injected into the RPV and mixed uniformly, will maintain the reactor shutdown under hot standby conditions. HSBW is calculated by the function, calc_hsbw().

The concentration of boron in the SLC tank, *xb_hot*, (ppm) required to obtain the HSBW is:

xb_hot = *xb_hot_nat* * *fb10_nat_wt* / *fb10_slc_wt*

This corresponds to a mass, *hsbw*, (lbm) of boron of:

hsbw = *xb_hot* * *mf_rpv_hot* / 1e6

The change in tank level corresponding to this mass of boron [in.] is:

delta_tk_hot = (*hsbw* * 1e6 * *vf_slc* * 12 in./ft) / (*xb_slc* * *slc_tk_area*)

The change in indicated tank level corresponding to hsbw [% o] is:

delta_slc_hot = *delta_tk_hot* * *delta_tk_lvl*

calc_hsbw() is called with a statement of form:

calc_hsbw(*interim*, *final*), *interim* and *final* bring the output file names into the function.

BORON INJECTION INITIATION TEMPERATURE: BIIT

The BIIT (calculated as a function of reactor power) is the greater of:

- The highest torus temperature at which initiation of boron injection will permit injection of the HSBW of boron before torus temperature exceeds the HCTL.
- The torus temperature at which a reactor scram is required by Technical Specifications.

BIIT is determined by execution of the function, calc_biit(). The boron injection rate, *wb*, is determined by evaluating the following:

wb = (*wslc* * *xb_slc*) / (1e6 * *k3* * *vf_slc*)

Knowing the injection rate, it is now possible to determine the time required to inject the HSBW, *t_hsbw*:

t_hsbw = *hsbw* / (60 * *wb*)

The heat balance used to evaluate the BIIT assumes the source of water injected into the RPV is the Condensate Storage Tank (CST), at its maximum normal operating temperature, *tcst*. The enthalpy of this water is then obtained by passing the hpt() function the pair of values (1000, *tcst*). The enthalpy of steam discharged from the SRVs is obtained by passing the hgsat_by_p() function the absolute pressure corresponding to the SRV lifting setpoint, *psrv_lift*.

The lower of HCTL high-pressure endpoint temperature at minimum and maximum normal operating torus water level, *tsp_hctl_lim1* is used to determine the BIIT. The value assigned to this variable is returned when *tsp_hctl_1_lo* and *tsp_hctl_1_hi* are passed to the min_floats() function.

The next section of code determines the torus water volume at the minimum normal operating water level, *wlsp_minlco*. An index, *index1* is initialized at 0, then incremented provided that the *wlsp* element of the *vtorus*[] array is smaller than *wlsp_minlco*. If *vtorus*[].*wlsp* is exactly equal to *wlsp_minlco*, the corresponding element *vtorus*[].*vsp* is placed in *vsp*. This is expected to be the case, since minimum normal operating water level is one of the values for which torus volumes are provided. However, to ensure the function is provided *vsp* if the test of exact equality fails, an else statement is provided to obtain *vsp* by interpolation of the *vtorus*[] elements with *wlsp* on either side of *wlsp_minlco*.

The mass of water in the torus, *mf_sp_lco* is obtained by dividing *tsp* by specific volume obtained from the vpt() function with input values (15, *tsp_scram*). Reference 3 uses saturation properties for this specific volume, however, as discussed earlier in this calculation, subcooled properties are more consistent with EOP usage. Enthalpy of torus water at *tsp_hctl_lim1*, *hsp_hctl_lim1*, is obtained from the hpt() function acting on (15, *tsp_hctl_lim1*). Reference 3 uses (*tsp_hctl_lim1* – 32) to approximate this value but since steam table values are available, this calculation will use them instead.

The quantity, *z1*, defined in reference 3 is now evaluated:

z1 = (*t_hsbw* * (*hg_srv* – *hsp_hctl_lim1*) * *qrx_rated* * *k1*)
z1 = *z1* / (100 * *mf_sp_lco* * (*hg_srv* – *hf_cst*))

The temperature to be assigned to the array defining the BIIT for reactor power less than ten-minute decay heat, *biit*[], is now determined by evaluating:

biit[0].*tsp* = *tsp_hctl_lim1* – (100 * *z1* * *fqdh_10*)

The same temperature is placed in *biit*[1].*tsp* by the assignment:

biit[1].*tsp* = *biit*[0].*tsp*

The power assigned to *biit*[0].*qrx* is by definition, 0:

biit[0].*qrx* = 0;

The next point defining the curve is the percent power corresponding to ten-minute decay heat, i.e.,

biit[1].*qrx* = 100 * *fqdh_10*

The limit on reactor power, *biit*[2].*qrx*, for which sufficient time is available to inject HSBW before torus temperature reaches *tsp_hctl_lim1* is:

biit[2].*qrx* = (*tsp_hctl_lim1* – *tsp_scram*) / *z1*

The pool temperature corresponding to this temperature, *biit*[2].*tsp*, is by definition *tsp_scram*, i.e.,

biit[2].*tsp* = *tsp_scram*

Finally, to define a final segment for the BIIT curve, a last point is arbitrarily defined with coordinates:

biit[3].*tsp* = *tsp_scram*, and
biit[3].*qrx* = 20

calc_biit() is called with a statement of form:

calc_biit(*interim*, *final*), *interim* and *final* bring the output file names into the function.

Pressure Suppression Pressure: PSP

The PSP is the smaller of:

- The highest torus pressure which can occur without steam in the torus airspace.
- The highest torus pressure at which initiation of RPV depressurization will not result in exceeding PCPL-A before RPV pressure drops to the MRFP.
- The highest torus pressure which can be maintained without exceeding the torus boundary design load if SRVs are opened.

The PSP is a function of containment water level with level bounded by the elevation of the downcomer openings, *hdco*, and *mpspcwl* (i.e., The PSP is by definition 0 psig for level less than *hdco*, or greater than or equal to *mpspscwl*). PSP is calculated using the function, calc_psp().

After declaring variables used internally, calc_psp() determines the lower limit of torus level used in the calculation:

$$wlsp_lim = hdco - hsp_ref$$

The partial pressures of water vapor in the drywell and torus atmosphere, given the reference 3 assumption of 100% relative humidity, are obtained by passing the psat() function the respective maximum normal operating temperatures for the space:

$$pg_dw = \text{psat}(tdw_maxop), \text{ and}$$
$$pg_sc = \text{psat}(tsc_maxop)$$

The partial pressures of nitrogen (or air) in the respective air volumes, when pressure in the airspace is at the minimum normal operating value and temperature is normalized to 100°F, are obtained using the ideal gas law. For the drywell, the equation solved is:

$$pndw_dw = ((pdw_minop - pg_dw + 14.7) * 559.69) / (tdw_maxop + 459.69)$$

For the torus, the equation solved is:

$$pnsc_sc = ((psc_minop - pg_sc + 14.7) * 559.69) / (tsc_maxop + 459.69)$$

The specific volume of water in the torus immediately following containment flooding, *vf_flood*, is obtained by calling the vpt() function with pressure set at 15 psia and temperature at *tsp_flood*. Reference 3 uses saturated fluid properties for this parameter, however execution of EOP required actions precludes containment pressure from lowering (significantly) below 15 psia, therefore subcooled water properties are used in this calculation.

Knowing the specific volume of water in the torus following flooding, the rate of change of pressure with water level of submerged portions of the containment is:

$$dpsp_dwlsp = 1 / (144 * vf_flood)$$

Values for an 1 x 11 array of a structured variable containing parameters that vary as a function of torus water level, *to_psp*[], are now assigned using a loop controlled by index1. Parameters evaluated are:

The partial pressure of drywell nitrogen or air in the torus airspace, normalized to 100°F is:

$$to_psp[].pnsc_dw = (pndw_dw * vdw) / vtorus[].vsc_air$$

The rate of change of pressure with temperature of all nitrogen (or air) in the torus is obtained using the ideal gas law:

to_psp[].*dpsc_dtsc* = (*pnsc_sc* + *to_psp*[].*pnsc_dw*) / 559.69

The partial pressure of water vapor at the high and low endpoint temperatures of the HCTL are obtained by passing the appropriate temperature to the psat() function:

to_psp[].*pg_hp* = psat(*tsp_hctl_1*[])
to_psp[].*pg_lp* = psat(*tsp_hctl_2*[])

The maximum pressure of the torus with no steam in the airspace, with the airspace in thermal equilibrium with the pool at the HCTL high-pressure endpoint temperature, *tsp_hctl_1*[], is obtained from the ideal gas law as follows:

to_psp[].*psc_hp_1* = ((*tsp_hctl_1*[] + 459.69) * *to_psp*[].*dpsc_dtsc*) + *to_psp*[].*pg_hp* – 14.7

The ratio of the of the HCTL high-pressure endpoint temperature to the HCTL low-pressure endpoint temperature (expressed as absolute values) is:

to_psp[].*z1* = (*tsp_hctl_1*[] + 459.69) / (*tsp_hctl_2*[] + 459.69)

This ratio is used to determine the initial torus airspace pressure with pool temperature at the HCTL high-pressure endpoint temperature required to assure that pool heat-up along the HCTL does not result in an airspace pressure above PCPL-A. The partial pressure of the gas at the low-pressure HCTL endpoint, when pressure is at PCPL-A is:

pcpl_to_calcs[] + 14.7 – *to_psp*[].*pg_lp*

Using the ideal gas law, the corresponding partial pressure when in thermal equilibrium with the pool at the HCTL high-pressure endpoint is:

to_psp[].*z1* * (*pcpl_to_calcs*[] – *to_psp*[].*pg_lp* + 14.7)

Adding this to the vapor pressure of water at *tsp_hctl_1*, the allowable airspace pressure at the HCTL high-pressure endpoint required to assure that depressurization along the HCTL does not result in airspace pressure rising above the PCPL is:

to_psp[].*psc_hp_2* = *to_psp*[].*z1* * (*pcpl_to_calcs*[] – *to_psp*[].*pg_lp* + 14.7) + *to_psp*[].*pg_hp* + 14.7

The limit on torus pressure to assure that torus boundary loads will not be exceeded if an SRV is opened is obtained as follows. The actual boundary load from SRV actuation is subtracted from the maximum allowable load and the result is compensated for hydrostatic head of water above the level for which the boundary load was calculated:

to_psp[].*psc_srv* = *psc_des* – *psp_srv* – (*dpsp_dwlsp* * (*vtorus*[].*wlsp* – *wlsp_srv*))

The minimum of *to_psp[].psc_hp_1*, *to_psp[].psc_hp_2* and *to_psp[].psc_srv* is assigned to *to_psp[].psc_lim*. Its value is determined by successively passing the min_float() function the parameter pairs (*to_psp[].psc_hp_1*, *to_psp[].psc_hp_2*) and (*to_psp[].psc_lim*, *to_psp.psc_srv*).

The code next uses knowledge of plant-specific values of *hdco* and *mpspcwl* to set-up a pair of indices to control determination of the final value of PSP as a function of water level, (*psp[].pressure* and *psp[].level*, respectively).

For water level below *hdco*, *psp[].pressure* is 0 psig, adjusted for the hydrostatic head of water above the torus pressure instrument tap. The adjustment for hydrostatic head is determined by using max_floats() to determine the larger of *level_float* (set equal to 0) and *h_tap[].hf_tap*, where *h_tap[].hf_tap* is:

$$h_tap[].hf_tap = psp[].level - hsc_tap + hsp_ref$$

The PSP is then:

$$psp[].pressure = dpsp_dwlsp * h_tap[].hf_tap$$

For water level between *hdco* and *mpspcwl*, PSP is the limiting value previously determined, adjusted for hydrostatic head of water above the instrument tap,

$$psp[].pressure = to_psp[].psc_lim + dpsp_dwlsp * h_tap[].hf_tap$$

For water level at or above *mpspcwl*, PSP is, by definition, 0.

calc_psp() is called with a statement of form:

calc_psp(*interim*, *final*), *interim* and *final* bring the output file names into the function.

Drywell Spray Related Variables

Calculation of values of variables and curves associated with operation of drywell sprays is done by calc_dwisl() and calc_mdsf(). The method used by each of these functions is described below.

DRYWELL SPRAY INITIATION LIMIT: DWSIL

Calc_dwsil() calculates the DWSIL using the methodology provided in reference 3. Reference 3 defines the DWISL as the highest drywell temperature at which initiation of drywell sprays will not result in an evaporative cooling pressure drop to below either:

- The drywell-below-wetwell differential pressure capability, or,
- The high drywell pressure scram setpoint.

Additionally, as implemented in JAF procedures, the DWSIL is restricted to those temperatures for which calibrated instrumentation is available. This temperature has been added to the plant-specific inputs, with the name, *tdw_maxind*.

The first value determined in the process of calculating the DWSIL is *pndw_dw*, the partial pressure of dry nitrogen in the drywell normalized to a temperature of 100°F is:

$pndw_dw = ((pdw_maxop + 14.7) * 559.69) / (tdw_minop + 459.69)$

At 100% relative humidity, the partial pressure of water vapor in the torus atmosphere is obtained by calling the psat() function with the minimum normal operating torus airspace temperature:

$pg_sc = \text{psat}(tsc_minop)$

The partial pressure of torus dry nitrogen in the drywell, normalized to 100°F using the ideal gas law is:

$pndw_sc = ((psc_minop - pg_sc + 14.7) * 559.69 * vsc_dco) / ((tsc_minop + 459.69) * vdw)$

The rate of change of temperature with pressure of all dry nitrogen in the drywell obtained using the ideal gas law is:

$dtdw_dpdw = 559.69 / (pndw_dw + pndw_sc)$

The minimum drywell temperature as a function of pressure for all dry nitrogen in the drywell is next determined as follows (for pressure from 0 psig to 59 psig, inclusive):

$tdw_min[].temperature = (dtdw_dpdw * (pdw + 14.7)) - 459.69$

The corresponding drywell pressures were simply assigned to *tdw_min[].pressure*.

Next, the initial drywell temperature required to assure that the evaporative cooling pressure drop will not lower drywell pressure below the high drywell pressure scram setpoint, as a function of drywell pressure is determined. The method used is to call the function, calc_evapdrop() (described below) with the parameter pair (*dpevap*, *pdw*). The value of *dpevap* is the difference between *pdw* and *pdw_scram*.

In the code, the initial point is determined by executing one of two conditional statements, depending on whether or not *pdw_scram* is an integer. If *pdw_scram* is not an integer, *tdw_scram[].pressure* is set equal to *pdw_scram*, *tdw_scram[].temperature* is set equal to the value returned by executing calc_evapdrop() with the parameter pair (0, *pdw_scram*), since in this case *dpevap* = 0. These pressure and temperature values are also assigned to *dwsil[]*, since for *pdw* at or below the sum, (*pdw_scram* + *dpdw_ww*), that is the definition of DWISL. Next, the temperature, *dwsil[].temperature* is compared to the stored maximum temperature, *max_temp*, and assigned to *max_temp* if it is larger, with the index of the stored maximum temperature, *n_max_temp* assigned the corresponding *n_dwsil*. Finally, indices into the *tdw_scram[]* and *dwsil[]* arrays are incremented and the first integer value of pressure above *pdw_scram* obtained by evaluating the expression, int(*pdw_scram*) + 1.

If *pdw_scram* is an integer value, the same process is used to determine, *tdw_scram{}*, *dwsil[]* and *max_temp*. However, in this case, the next integral value of pressure is obtained by evaluating the expression, *pdw_scram* + 1.

For subsequent pressures, up to the sum, (*pdw_scram* + *dpdw_ww*), the same process is used, except calc_evapdrop() is passed *dpevap*, the difference between *pdw* and *pdw_scram* along with *pdw* (i.e., (*dpevap*, *pdw*)).

Finally, *pdw* is set equal to the sum, (*pdw_scram* + *dpdw_ww*). This value is then evaluated with a conditional statement to determine if it is an integer value (in which case the corresponding *tdw_scram* would have been determined in the preceding section of code). If *pdw* is not an integer, one last *tdw_scram[]* value is determined using the same process as has been used previously.

The next section of code determines the limit on drywell temperature required to assure that the evaporative cooling pressure drop will not lower drywell pressure to a lower than the drywell-below-wetwell pressure capability limit, *tdw_cap*[]. The first step in this process is to find the first integer value of pressure above the sum of *pdw_scram* and *dpdw_ww*. This is done with the assignment:

pdw = int(*pdw*) + 1

The number of points for which *tdw_cap*[] has been calculated, *n_tdw_cap* is also initialized at 0.

Then, *tdw_cap*[].temperature is determined by passing calc_evapdrop() the values (*dpdw_ww*, *pdw*). As before, *max_temp* is tracked and *tdw_cap*[] is also assigned to *dwsil*[]. This process is repeated with 1 psi increments in pressure, *pdw*, until a total of 60 (including those previously assigned) values have been assigned to *dwsil*[]. The total number of *tdw_cap*[] values calculated is tracked by incrementing *n_tdw_cap* on each passage through the loop.

The point at which the lines defining *tdw_min*[] and *dwsil*[] intersect is the next set of values determined. The first step in this process is to synchronize the indices into the *tdw_min*[] and *dwsil*[] arrays (*tdw_min*[] was calculated for integer values of pressure, *dwsil*[] may include non-integer pressures at *pdw_scram* and (*pdw_scram* + *dpdw_ww*)). Since *tdw_min*[] is calculated for *pdw* beginning at 0, and *pdw_scram* is greater than 0, the index into *tdw_min*[] is incremented as long as *tdw_min*[].pressure is less than *dwsil*[].pressure. Once *tdw_min*[].*pressure* is greater than *dwsil*[].*pressure*, the index into *dwsil*[], *index_dwsil*, is incremented to bring the pressure values into alignment. As long as the pressure members of *tdw_min*[] and *dwsil*[] are equal (and *tdw_min*[].*temperature* is less than *dwsil*[].*temperature*), the indices into the arrays are both incremented. If the pressure values become unequal (due to the *dwsil*[].*pressure* at (*pdw_scram* + *dpdw_ww*) being a non-integer value), *index_dwsil* is incremented to get past this point. If a non-integer value of *dwsil*[].*pressure* existed, the following while() construct continues to increment *index_tdw* and *index_dwsil* until *tdw_min*[].*temperature* is equal to or greater than *dwsil*[].*temperature*.

The following set of assignments set indices into the *tdw_min*[] and *dwsil*[] arrays such that the pressure elements bracket the pressure at which the curves intersect. These values are assigned to the variables, *lo_press_tdw*, *hi_press_tdw*, *lo_press_dwsil* and *hi_press_dwsil*. These indices are then used to find the equations of the line segments (f(p) = t = (m * p) + b) from *tdw_min*[*lo_press_tdw*].*pressure* to *tdw_min*[*hi_press_tdw*].*pressure* and from *dwsil*[*lo_press_dwsil*].*pressure* to *dwsil*[*hi_press_dwsil*].*pressure*. The slope, m, and intercept, b, of each line is determined by evaluating the relations:

$$m = (f(p_{hi}) - f(p_{lo})) / (p_{hi} - p_{lo})$$

$$b = f(p_{hi}) - (m * p_{hi})$$

The code performing this for the *tdw_min*[] array is:

```
m_tdw_min = (tdw_min[hi_press_tdw].temperature - tdw_min[lo_press_tdw].temperature);
m_tdw_min = m_tdw_min / (tdw_min[hi_press_tdw].pressure - tdw_min[lo_press_tdw].pressure);
b_tdw_min = tdw_min[hi_press_tdw].temperature - (m_tdw_min * tdw_min[hi_press_tdw].pressure);
```

A similar set of equations is used with the *dwsil*[] array.
Knowing the equations of the lines (for convenience, call the functions f(p) and g(p)), at the point of intersection, p is the same and f(p) = g(p). Therefore, the following relation may be solved for p:

$$f(p) = (m_1 * p) + b_1,$$

$g(p) = (m_2 * p) + b_2$

$(m_1 * p) + b_1 = (m_2 * p) + b_2$

$(m_1 * p) - (m_2 * p) = b_2 - b_1$

$p * (m_1 - m_2) = b_2 - b_1$

$p = (b_2 - b_1) / (m_1 - m_2)$

In the code this is implemented as:

intersect.pressure = (*b_dwsil* – *b_tdw_min*) / (*m_tdw_min* – *m_dwsil*)

and the equation of the line for *dwsil*[] is used to find the corresponding temperature:

intersect.temperature = (*m_dwsil* * *intersect.pressure*) + *b_dwsil*

These values are then stored as the next to the last member of the *dwsil*[] array. The last member of *dwsil*[] is assigned an arbitrarily high pressure value (*ppc_maxind*, the maximum indicated containment pressure) and the same temperature (*intersect.temperature*).

The next significant section of code truncates the DWSIL curve (completely defined at the completion of the previous step) to the maximum temperature for which calibrated indication is available, *tdw_maxind*. This code only executes if the previously stored maximum temperature, *max_temp*, is greater than *tdw_maxind*.

If *tdw_maxind* is less than *max_temp*, an index to carry the first *dwsil*[] element with temperature less than *tdw_maxind* is set equal to the element of *dwsil*[] with temperature equal to *max_temp* (i.e., *n_low_cut* = *n_max_temp*). An iteration is then performed in which *n_low_cut* is decremented until *dwsil*[].*temperature* is less than *tdw_maxind*. After this value is determined, an index to carry the first element of *dwsil*[] with temperature below *tdw_maxind*, as pressure (i.e., the index into the array) is increased is set equal to *n_max_temp* (i.e., *n_hi_cut* = *n_max_temp*). This index is then incremented until *dwsil*[*n_hi_cut*].*temperature* is at or below *tdw_maxind*.

When *n_low_cut* and *n_hi_cut* are known, values of *dwsil*[] with indices below *n_lo_cut* are copied to *dwsil_eop*[] (the DWSIL for EOP use).

Following this, the equation of the line defined by the *dwsil*[] points at *n_low_cut* and (*n_low_cut* + 1), the line which passes through *tdw_maxind* is determined using the process described above. Using the coefficients of this equation, and knowing *tdw_maxind*, the equation of the line may be solved for pressure as follows:

$f(p) = (m * p) + b$

$p = (f(p) - b) / m$

In the code this is implemented as:

intersect.pressure = (*tdw_maxind* – *b_dwsil*) / *m_dwsil*

The temperature corresponding to this pressure, *tdw_maxind*, is then assigned to *intersect.temperature*. These values of pressure and temperature are then assigned to the next element of *dwsil_eop*[]. Following this, the same process is used with *n_hi_cut* and the resulting pressure and temperature are assigned as the next elements of *dwsil_eop*[].

Once these values have been assigned to *dwsil_eop*[], the remaining *dwsil*[] values are copied to *dwsil_eop*[].

If *max_temp* is lower than *tdw_maxind*, all values of the *dwsil*[] array are copied directly to the *dwsil_eop*[] array.

calc_dwsil() is called with a statement of form:

calc_dwsil(*interim*, *final*), *interim* and *final* bring the output file names into the function.

calc_evapdrop()

Calc_evapdrop() calculates the initial temperature required to obtain a given evaporative cooling pressure drop when also given the initial pressure. The method used is derived from the equations given in attachment 2 of reference 3. Since DWSIL is predicated on spray into a space with an initial relative humidity of 0%, the initial drywell atmosphere conditions are:

$pdw_1 = pn_1 - 14.7$, where
pdw_1 is drywell pressure [psig] and
pn_1 is the partial pressure of nitrogen [psia],

$pg_1 = 0$, is the partial pressure of water vapor in the airspace [psia], and

t_1 is the temperature [°F].

The spray water is initially at temperature, tspray (set as a constant, 32°F). Reference 3 uses a pressure of 500 psia to determine enthalpy, i.e.,

hfspray = hf(500, tspray)

After evaporative cooling has occurred (given immediate uniform mixing of the spray water such that the atmosphere is completely saturated), the final atmospheric conditions are:

pdw_2 is the final pressure [psig],

pn_2 is the partial pressure of nitrogen [psia],

pg_2 is the vapor pressure of water [psia], and

t_2 is the final temperature [°F].

By definition, the evaporative cooling pressure drop is:

$dpevap = p_1 - p_2$.

Since it is given that the drywell atmosphere is completely saturated following spray operation,

$pg_2 = psat(t_2)$, and

$p_2 = pn_2 + pg_2 - 14.7$.

The partial pressure of nitrogen at the final state is then,

$pn_2 = p_2 - pg_2 + 14.7$

Substituting (p_1 – dpevap) for p_2, and $psat(t_2)$ for pg_2,

$pn_2 = p_1 - dpevap - psat(t_2) + 14.7$

Again, since it is given that the drywell atmosphere is saturated at the completion of the evaporative cooling pressure drop, the enthalpy and specific volume of the water vapor in the drywell are:

$hg_2 = hgsat(t_2)$, and,

$vg_2 = vgsat(t_2)$, respectively.

The energy absorbed to vaporize the spray water is then:

$Espray = (vdw / vg_2) * (hg_2 - hfspray)$

The energy liberated by nitrogen of mass mn is:

$En = mn * cp_n * (t_1 - t_2)$, where, cp_n is the specific heat of nitrogen, and mn may be obtained using the ideal gas law:

$mn = (vdw * pn_2 * 144) / (RN * (t_2 + 459.69))$, where RN is the ideal gas constant for nitrogen.

Since the energy required for vaporization of the spray water is obtained by cooling nitrogen, these energy terms may be equated:

$(vdw / vg_2) * (hg_2 - hfspray) = mn * cp_n * (t_1 - t_2)$

Substituting previous expressions for hg_2, vg_2 and mn into this equation results in the relation:

$(vdw / vgsat(t_2)) * (hgsat(t_2) - hfspray) = [(vdw * pn_2 * 144) / (RN * (t_2 + 459.69))] * cp_n * (t_1 - t_2)$

This is equivalent to:

$(hgsat(t_2) - hfspray) / vgsat(t_2) = [(pn_2 * 144 * cp_n) / (RN * (t_2 + 459.69))] * (t_1 - t_2)$

Rearranging terms, this yields:

$(t_1 - t_2) = [(hgsat(t_2) - hfspray) / vgsat(t_2)] * [RN * (t_2 + 459.69) / (pn_2 * 144 * cp_n)\}$

Inserting the previously derived expression for pn_2, rearranging terms and solving for t_1 yields:

$t_1 = t_2 + [RN * (hgsat(t_2) - hfspray) * (t_2 + 459.69)] / [vgsat(t_2) * cp_n * (p_1 - dpevap - psat(t_2) + 14.7) * 144]$

From the ideal gas law, it is known that:

$pn_1 / (t_1 + 459.69) = pn_2 / (t_2 + 459.69)$

Substituting previously shown expressions for pn_1 and pn_2:

$(p_1 + 14.7) / (t_1 + 459.69) = (p_1 - dpevap - psat(t_2) + 14.7) / (t_2 + 459.69)$

Rearranging terms yields:

$(p_1 + 14.7) * (t_2 + 459.69) / (p_1 - dpevap - psat(t_2) + 14.7) = t_1 + 459.69$, or

$t_1 = [(p_1 + 14.7) * (t_2 + 459.69) / (p_1 - dpevap - psat(t_2) + 14.7)] - 459.69$

Equating the two expressions for t_1, the following is obtained:

$t_2 + [RN * (hgsat(t_2) - hfspray) * (t_2 + 459.69)] / [vgsat(t_2) * cp_n * (p_1 - dpevap - psat(t_2) + 14.7) * 144] = [(p_1 + 14.7) * (t_2 + 459.69) / (p_1 - dpevap - psat(t_2) + 14.7)] - 459.69$

Rearranging terms:

$(t_2 + 459.69) + [RN * (hgsat(t_2) - hfspray) / vgsat(t_2) * cp_n * (p_1 - dpevap - psat(t_2) + 14.7) * 144)] * (t_2 + 459.69) = [(p_1 + 14.7) * (t_2 + 459.69) / (p_1 - dpevap - psat(t_2) + 14.7)]$

This is equivalent to:

$1 + [RN * (hgsat(t_2) - hfspray) / vgsat(t_2) * cp_n * (p_1 - dpevap - psat(t_2) + 14.7) * 144)] = [(p_1 + 14.7) / (p_1 - dpevap - psat(t_2) + 14.7)]$

Multiplying by both sides by $(p_1 - dpevap - psat(t_2) + 14.7)$:

$(p_1 - dpevap - psat(t_2) + 14.7) + [RN * (hgsat(t_2) - hfspray) / (vgsat(t_2) * cp_n * 144)] = p_1 + 14.7$

Subtracting $(p_1 + 14.7)$ from both sides:

$[RN * (hgsat(t_2) - hfspray) / (vgsat(t_2) * cp_n * 144)] - (dpevap + psat(t_2)) = 0$, or,

$[RN * (hgsat(t_2) - hfspray) / (vgsat(t_2) * cp_n * 144)] = dpevap + psat(t_2)$

The code in calc_evapdrop() works as follows:

Values of initial drywell pressure, *p1*, and evaporative cooling pressure drop, *dpevap*, are passed to the function. The function contains definitions for the following parameters:

RN, the ideal gas constant for nitrogen is set constant at 55.1 ft-lbm/lbm-°R,

CPN, the specific heat of nitrogen is set constant at 0.248 Btu/lbm-°R,

tspray, the temperature of the spray water is set to 32°F.

Upon entry, the function determines the enthalpy of the spray water, *hfspray*, by passing the function hpt() the values (500 psia, *tspray*).

Storage for the result of evaluation of the right and left sides of the above equation, *rside* and *lside* is initialized at 0. The final drywell temperature, *t2*, is then set to *tspray* (since drywell temperature is greater than *tspray*, this is a bounding lower value for *t2*).

The following process is then repeated until *rside* becomes greater than *lside*:

rside and *lside* are assigned the values:

rside = *dpevap* + psat(*t2*), and *lside* is assigned the value:

lside = (*RN* * (hgsat_by_t(*t2*) – *hfspray*)) / (*CPN* * 144 * vgsat_by_t(*t2*)),

then the value of *t2* is raised by 0.01.

When the loop terminates, *t1* is determined by evaluation of the expression:

t1 = ((*p1* + 14.7) * (*t2* + 459.69) / (*p1* – *dpevap* – psat(t2) + 14.7)) – 459.69

t1 is then returned to the routine which called calc_evapdrop().

calc_evaprop() is called with an assignment statement of form:

a = calc_evapdrop(*dpevap*, *pdw*)

MINIMUM DRYWELL SPRAY FLOW: MDSF

The MDSF is the lowest spray flow that assures uniform circumferential spray distribution within the drywell. The value assigned will be the larger of the values calculated for 'A' and 'B' spray headers, since this ensures MDSF is met irrespective of failure of either system to operate.

The first step in determining *mdsf* is to evaluate the minimum required spray flow for each header by calculating the product of the number of spray nozzles on the header, *spray_hdr*[].*num_nzls* by the flow required to establish a full cone spray pattern for that nozzle type, *spray_hdr*[].*wnzl*. The results of these evaluations, *msf_a* and *msf_b* are then passed to the max_floats() function to obtain the larger flow, which is assigned to *mdsf*.

This calculation is done by the function, calc_mdsf(). calc_mdsf() is called with a statement of form:

calc_mdsf(*final*), *final* brings the output file name into the function.

Minimum Debris Retention Injection Rate: MDRIR

The MDRIR is the lowest RPV injection rate at which it is expected that core debris will be retained in the RPV when RPV water level cannot be determined to be above the bottom of active fuel. MDRIR is calculated using the function, calc_mdrir().

Calculation of *mdrir*[][] begins with determination of the specific volume of water injected into the RPV. This is done by calling vpt() with the parameter pair (500, *tsp_hctl_2_lco*) in accordance with reference 3.

Mdrir[][] is calculated for selected RPV pressures, specifically, 0 psig, 500 psig and 1000 psig, by repeating a loop based on pressure. Initially, pressure is set at:

prpv_g = 0 psig, and
prpv_a = *prpv_g* + 14.7 (= 14.7) psia.

An index into the *mdrir*[][] array on pressure, *index_p* is set to 0.

Now while *prpv_g* is less than 1100 psig, the following process is iteratively repeated:

The enthalpy of water reaching core debris, *hf_inj*, and the enthalpy of steam discharged from the RPV, *hg_rpv*, are respectively determined by passing the functions, hfsat_by_p() and hgsat_by_p() the value, *prpv_a*.

MDRIR is now determined as a function of time after shutdown using a loop that is executed 25 times, controlled by *index_t*.

The time element of *mdrir*[][] is assigned the value of time from the generic decay heat data, i.e.,

mdrir[][].*time* = *sd_pwr.time*

The pressure element of *mdrir*[][] is assigned pressure, *prpv_g*.

Finally, the flow required to remove decay heat at a given time after shutdown is calculated as follows, and assigned to the flow element of *mdrir*[][]:

mdrir[][].*flow* = (*qrx_rated* * *sd_pwr*[].*power* * *k1* * *vf_inj* * *k3*) / ((*hg_rpv* – *hf_inj*) * 60 * 100)

Following completion of this loop, *prpv_g* and *prpv_a* are raised by 500, *index_p* is incremented and the process repeated (provided *prpv_g* is less than 1100 psig).

calc_mdrir() is called with a statement of form:

calc_mdrir(*interim*, *final*), *interim* and *final* bring the output file names into the function.

Suppression Chamber Spray Initiation Pressure: SCSIP

The SCSIP is the lowest pressure that can exist in the torus when 95% of the noncondensibles from the drywell have been transferred to the torus airspace. SCSIP is calculated using the function, calc_scsip().

The partial pressures of water vapor in the torus and drywell airspace with 100% relative humidity, *pg_sc* and *pg_dw*, are obtained by passing the function, psat() the maximum normal operating airspace temperatures, *tsc_maxop* and *tdw_maxop*.

The partial pressure of torus dry nitrogen (or air) in the torus airspace, normalized to 100°F determined using the ideal gas law is:

$$pnsc_sc = ((psc_minop - pg_sc + 14.7) * 559.69) / (tsc_maxop + 459.69)$$

The partial pressure of drywell dry nitrogen (or air) in the torus airspace, normalized to 100°F determined using the ideal gas law is:

$$pnsc_dw = ((pdw_minop - pg_dw + 14.7) * 559.69 * vdw) / ((tdw_maxop + 459.69) * vsc_lco)$$

The value of *scsip* is then obtained by solving the following expressions, in order:

$$scsip = (tsp_minop + 459.69) * (pnsc_sc + (0.95 * pnsc_dw)) / 559.69, \text{ and,}$$

$$scsip = scsip + \text{psat}(tsp_minop) - 14.7$$

calc_scsip() is called with a statement of form:

calc_scsip(*interim*, *final*), *interim* and *final* bring the output file names into the function.

SUMMARY/CONCLUSIONS

Input values required for generation of JAF Emergency Procedure and Severe Accident Guidelines, Emergency Operating Procedures and Severe Accident Operating Guidelines have been calculated and are tabulated in attachment 31. The objective of this calculation has been achieved.

ATTACHMENTS

1. Containment material properties
2. Decay heat data
3. Generic data
4. Containment data
5. Contaiment spray header data
6. Eccs system flows
7. Fuel data by type
8. Plant data
9. Srv discharge line length
10. Torus volumes
11. config.dat listing
12. thermin.dat listing
13. do eop calculations.cpp listing
14. plant data.h listing
15. thermo.h listing
16. eop calculation input.h listing
17. shared data.h listing
18. shared functions.h listing
19. mpspcwl.h listing
20. pcpl.h listing
21. stpll.h listing
22. hctl.h listing

23. rpv.h listing
24. boron.h listing
25. psp.h listing
26. drywell spray.h listing
27. mdrir.h listing
28. scsip.h listing
29. function cross-reference (by function)
30. eop calc intermediate results
31. eop calculation final results

FUNCTION INDEX

assign_float	5
assign_int	5
assign_n_floats	6
calc_biit	36
calc_csbw	36
calc_dppc_dwlpc	15
calc_dwsil	41
calc_evapdrop	44
calc_hctl	29
calc_hsbw	36
calc_marfp	22
calc_mcfi	25
calc_mcutl	26
calc_mdrir	48
calc_mdsf	48
calc_mnsred	23
calc_mpspcwl	14
calc_mrfp	25
calc_mscrwl	27
calc_msrp	24
calc_mzirwl	27
calc_n_values	28
calc_pcp	38
calc_qdh	29
calc_scsip	49
calc_stpll	19
calc_tsp_hctl_2	20
calculate_pcpl	17
do_pcpl_output	19
free_thermo	11
get_component_strength	15
init_boron_values	35
init_rpv_values	22
interpolate	12
load_plant_data	6
load_thermo	7
max_floats	13
max_int	13
min_floats	13
pcpl_for_calcs	18
pressure_for_components	16
pressure_for_srvs	16
Saturated water properties	
pressure	8
temperature	9
sequence_boron	35
sequence_pcpl	14
sequence_rpv	22
Subcooled fluid properties	10
Superheated fluid properties	10
write_input_values	12

containment material properties

Type 1

SS304, SA240, SA3

Temperature	70	200	400	600
Yield strength	1	0.925	0.81	0.76
Tensile strength	1	0.89	0.72	0.715

Source

"Engineering Properties of Steel," ASM, 1982; USS Steel Design Manual, May 1974; Engineering Material Science, C.W. Richards, 1961; Elevated Temperature Properties of Carbon Steels, Simmons and Cross, ASTM Special Publication No. 180, Sept. 1955

BWROG EPG/SAG App. C, Rev. 1, p. C-3-3

Type 2

S21800, A193 and

Temperature	70	200	400	600
Yield strength	1	0.79	0.58	0.53
Tensile strength	1	0.87	0.79	0.76

Source

"Engineering Properties of Steel," ASM, 1982; USS Steel Design Manual, May 1974; Engineering Material Science, C.W. Richards, 1961; Elevated Temperature Properties of Carbon Steels, Simmons and Cross, ASTM Special Publication No. 180, Sept. 1955

BWROG EPG/SAG App. C, Rev. 1, p. C-3-3

Type 3 A36

Temperature	70	200	400	600
Yield strength	1	0.92	0.95	0.88
Tensile strength	1	0.95	1	1

Source "Engineering Properties of Steel," ASM, 1982; USS Steel Design Manual, May 1974; Engineering Material Science, C.W. Richards, 1961; Elevated Temperature Properties of Carbon Steels, Simmons and Cross, ASTM Special Publication No. 180, Sept. 1955

BWROG EPG/SAG App. C, Rev. 1, p. C-3-3

Type 4 A160 Gr. B and A1

Temperature	70	200	400	600
Yield strength	1	0.94	0.86	0.77
Tensile strength	1	0.96	1	1

Source "Engineering Properties of Steel," ASM, 1982; USS Steel Design Manual, May 1974; Engineering Material Science, C.W. Richards, 1961; Elevated Temperature Properties of Carbon Steels, Simmons and Cross, ASTM Special Publication No. 180, Sept. 1955

BWROG EPG/SAG App. C, Rev. 1, p. C-3-3

Type 5 A201, Gr. B, A212

Temperature	70	200	400	600
Yield strength	1	0.95	0.86	0.76
Tensile strength	1	0.95	1	1

Source "Engineering Properties of Steel," ASM, 1982; USS Steel Design Manual, May 1974; Engineering Material Science, C.W. Richards, 1961; Elevated Temperature Properties of Carbon Steels, Simmons and Cross, ASTM Special Publication No. 180, Sept. 1955

BWROG EPG/SAG App. C, Rev. 1, p. C-3-3

decay heat data

JAF-CALC-MISC-03116, Rev. 0 Attachment 2

Time after shutdown [min.]	Decay heat [%]	Source
2	3.038	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
4	2.658	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
6	2.458	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
8	2.32	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
10	2.212	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
15	2.009	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
20	1.862	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
25	1.747	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
30	1.653	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
35	1.574	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
40	1.507	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
45	1.45	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
50	1.4	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
60	1.318	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4

Time after shutdown [min.]	Decay heat [%]	Source
90	1.155	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
120	1.057	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
150	0.9901	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
200	0.9661	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
300	0.8673	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
450	0.7863	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
600	0.7255	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
900	0.6509	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
1200	0.6002	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
1500	0.5622	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4
1800	0.5311	BWROG EPG/SAG App. C, Rev. 1, p. C-3-4

generic data

Variable	Value	Source
c_steel	0.128	GE internal document Y1002A070; BWROG EPG/SAG App. C, Rev 1, p. C-3-2
dptp_dprpv	0.1	NEDE-21878-1, March 1981; BWROG EPG/SAG App. C, Rev. 1, p. C-3-2
dptp_dwll	5	GE memo PPS-59-82 (DRF A22-0073, Tab E), October 11, 1982; BWROG EPG/SAG App. C, Rev.1, p. C-3-2
fb10_nat	0.1978	Chart of Nuclides, Knolls Atomic Power Laboratory, 14th Ed., 4/88; BWROG EPG/SAG App. C, Rev. 1, p. C-3-2
fb11_nat	0.8022	Chart of Nuclides, Knolls Atomic Power Laboratory, 14th Ed., 4/88; BWROG EPG/SAG App. C, Rev. 1, p. C-3-2
fqdh_10	0.022115	BWROG EPG/SAG App. C, Rev. 1, p. C-3-2
fqdh_100	0.01118	BWROG EPG/SAG App. C, Rev. 1, p. C-3-2
fqdh_2	0.030381	BWROG EPG/SAG App. C, Rev. 1, p. C-3-2
k1	3412000	Marks' Standard Handbook for Mechanical Engineers, Baumeister, Avallone, et al, 1978; BWROG EPG/SAG App. C, Rev. 1, p. C-3-3
k2	56868	Marks' Standard Handbook for Mechanical Engineers, Baumeister, Avallone, et al, 1978; BWROG EPG/SAG App. C, Rev. 1, p. C-3-3
k3	7.48	Marks' Standard Handbook for Mechanical Engineers, Baumeister, Avallone, et al, 1978; BWROG EPG/SAG App. C, Rev. 1, p. C-3-3
mb10	10.01	Chart of Nuclides, Knolls Atomic Power Laboratory, 14th Ed., 4/88; BWROG EPG/SAG App. C, Rev. 1, p. C-3-3
mb11	11.01	Chart of Nuclides, Knolls Atomic Power Laboratory, 14th Ed., 4/88; BWROG EPG/SAG App. C, Rev. 1, p. C-3-3
srv_open	50	NEDE-24988, October, 1981; BWROG EPG/SAG App. C, Rev. 1, p. C-3-4; Target Rock 2-stage, Model 7567F, Vendor Manual #T020-0041
tlim_dw	545	BWROG EPG/SAG App. C, Rev. 1 assumption for PCPL calculation, page C-2-45

Variable	Value	Source
tlim_sp	350	BWROG EPG/SAG App. C, Rev. 1 assumption for PCPL calculation, page C-2-45

containment data

h_n	Location	Material	Strength	pcalc_n	tcalc_n	Source
0	WW	4	1	111	300	JAF-CALC-PC-03115, R0
29.5	WW	4	1	108	300	JAF-CALC-PC-03115, R0
30.3	WW	1	1	120	350	JAF-CALC-PC-01436
50	DW	4	1	129	300	JAF-CALC-PC-03115, R0
85	DW	4	1	118	300	JAF-CALC-PC-03115, R0

containment spray header data

Spray header	A
num_nzls	263
Source of num_nzls	DBD-010, PCN No. JAF-DBD-010-0-041
wnzl	20
Source of wnzl	DBD-010, PCN No. JAF-DBD-010-0-035 contains a Spray Systems Co. catalogue excerpt which shows a 1-1/2 H 30 nozzle provides 20 gpm at a pressure drop of 3 psi

Spray header	B
num_nzls	221
Source of num_nzls	DBD-010, PCN No. JAF-DBD-010-0-041
wnzl	20
Source of wnzl	DBD-010, PCN No. JAF-DBD-010-0-035 contains a Spray Systems Co. catalogue excerpt which shows a 1-1/2 H 30 nozzle provides 20 gpm at a pressure drop of 3 psi

eccs system flows

JAF-CALC-MISC-03116, Rev. 0 Attachment 6

System	CS									
Flow rate	6781	6000	5500	5000	4500	4000	3000	2000	1000	0
RPV pressure	0	59	93	123	151	176	219	256	279	295
Source	JAF-CALC-RHR-03153, Rev. 0									

System	RHR									
Flow rate	9628	9000	8000	7000	6000	5000	4000	3000	2000	0
RPV pressure	0	26	65	99	126	149	166	182	195	208
Source	JAF-CALC-RHR-03153, Rev. 0									

fuel data by type

Fuel type GE11

lfuel [in.]	146								
Source of lfuel	M1-97-030, Cycle 14 Reload Core								
Time after shutdown [min.]	10	30	50	80	100	500	1000	3000	6000
Time to reach 1500 degf [min.]	2.35	3.15	3.72	4.33	4.65	6.85	8.27	11.57	14.87
Source for times	GE letter JAB-N7018, "Plant Emergency Procedure Parameters Affected by Fuel Design," March 14, 1997; GE letter JAB-N8072, "Plant Emergency Procedure Parameters," October 27, 1998								
fafl_15	83.33								
fafl_18	75								
wg_1500	1260								
Source of steam cooling data	GE letter JAB-N7018, "Plant Emergency Procedure Parameters Affected by Fuel Design," March 14, 1997; GE letter JAB-N8072, "Plant Emergency Procedure Parameters," October 27, 1998								

Fuel type GE12

lfuel [in.]	150								
Source of lfuel	M1-97-030, Cycle 14 Reload Core								
Time after shutdown [min.]	10	30	50	80	100	500	1000	3000	6000
Time to reach 1500 degf [min.]	2.35	3.15	3.72	4.33	4.65	6.85	8.27	11.57	14.87
Source for times	GE letter JAB-N7018, "Plant Emergency Procedure Parameters Affected by Fuel Design," March 14, 1997; GE letter JAB-N8072, "Plant Emergency Procedure Parameters," October 27, 1998								
fafl_15	83.33								
fafl_18	75								
wg_1500	1260								
Source of steam cooling data	GE letter JAB-N7018, "Plant Emergency Procedure Parameters Affected by Fuel Design," March 14, 1997; GE letter JAB-N8072, "Plant Emergency Procedure Parameters," October 27, 1998								

plant data

Variable	Value	Source
BWR_PSC_type	Mark I	UFSAR Section 16.7.3.1
BWR_type	BWR/4	UFSAR Section 1.7
c_clad	0.0761	GE internal document Y1002C130; BWROG EPG, Rev. 4 App. C, page C-2-11
c_fuel	0.0683	GE internal document Y1002C021; BWROG EPG, Rev. 4, App. C, page C-2-11
dpdw_ww	8.5	Teledyne Engineering Services Technical Report TR-6177-1, Rev. A, Determination of Allowable Drywell Negative Pressure JAFNPP, 10/84, Table 3-1
dpsrv	28	JAF Mod M1-90-083;
fb10_slc	0.347	Technical Specification 3.4.C
hdco	9.58	JAF-CALC-PC-03044, R0
hring	17.85	JAF-CALC-PC-03044, R0
hsc_tap	20.98	JAF-CALC-PC-03115, R0
hsp_ref	0	UFSAR Table 7.3-10
hsrv	65.91	FP-27B, SRV discharge flange centerline; Dwg. 6.32-80; Dwg. 6.32-82
hvent_pc	104.25	Elevation of bottom invert of penetration X-26A/B. FV-1A provides centerline elevation of 335 ft and nominal diameter of 18 in.
mclad	109124	Memorandum REF-98-112, EPG Calculations - Cycle 14 Core, George Rorke to David Burch, October 28, 1998 (49498.414 kg * 2.2046 lb/kg, Cameron Hydraulic Data, Eighteenth Edition)
mf_rpv_cld	8.95e5	JAF-CALC-MISC-01966, R1

Variable	Value	Source
mf_rpv_hot	5.32e5	JAF-CALC-MISC-01966, R1
mfuel	249314	Memorandum REF-98-112, EPG Calculations - Cycle 14 Core, George Rorke to David Burch, October 28, 1998, 113088.26 kg (x 2.2046 lb/kg, from Cameron Hydraulic Data, Eighteenth Edition)
mg_rpv_hot	1.53e4	JAF-CALC-MISC-01966, R1
mrpv	2.02e6	JAF-CALC-MISC-01966, R1
mrpv_taf	9.59e5	JAF-CALC-MISC-01966, R1
nbr_fuel_types	2	M1-97-030, Cycle 14 Reload Core
nbr_pc_members	5	Arbitrarily defined, sum of the number of components evaluated in JAF-CALC-PC-03115 and JAF-CALC-PC-01436
nbuns	560	M1-97-030, Cycle 14 Reload Core
nsrv_ads	7	Technical Specification 3.5.D.1
pdw_maxop	1.95	OP-37, Containment Atmosphere Dilution System
pdw_minop	1.2	OP-37, Containment Atmosphere Dilution System; procedure requires drywell to torus dP be maintained >= 1.7 psid, minimum torus pressure required by procedure is -0.5 psig
pdw_scram	2.7	Technical Specification Table 3.1-1
ppc_maxind	85 (torus)	UFSAR Table 7.19-1
ppc_vent	62	SWEC Calculation 14620.9011 US(N) 004-1, Suppression Chamber (20") and Drywell (24") Vent and Purge Butterfly Valves Evaluation Based on Relap5/Mod2 56 psig and 62 psig Results
pq_code	37.44	JAF-CALC-PC-01519, R0

Variable	Value	Source
pq_des	27.66	JAF-CALC-PC-01519, R0
pqs_code	36	JAF-CALC-PC-01519, R0
pqs_des	11.287	JAF-CALC-PC-01519, R0
prpv_maxind	1500	UFSAR Table 7.19-1
prpv_tp	1250	JAF-CALC-MISC-02240, R0, Attachment 5
psc_maxop	0.5	OP-37, Containment Atmosphere Dilution System
psc_minop	-0.5	OP-37, Containment Atmosphere Dilution System
psp_des	56	Dwg. 3.11-44
psp_srv	12.87	JAF-CALC-MISC-02240, R0, Attachment 5
psrv_lift	1145	Technical Specification 2.2.1.B
psrv_name	1080	OPL-3, R0, Transient Protection Parameters Verification for Reload Licensing Analyses - FitzPatrick - Reload 13 - Cycle 14 Final
psup_srv	110	OP-37, Containment Atmosphere Dilution System
ptp_code	42	JAF-CALC-PC-01519, R0
ptp_des	22.5	JAF-CALC-MISC-02240, R0, Attachment 5
ptps_code	1.5	JAF-CALC-PC-01519, R0
ptps_des	1	JAF-CALC-PC-01519, R0
qrx_rated	2536	Technical Specification 1.0.N
slc_tk_dia	8.97	Dwg. 3.37-3; outside diameter 9 ft, wall thickness 0.1875 in.

Variable	Value	Source
slc_tk_lvl_in_span	121.5	Dwg. 16.11-47
slc_tk_lvl_out_span	100	Dwg. 16.11-48
srv_type	Target Rock, 2-s	Model 7567F-010; Vendor Manual T020-0041; Rome Equipment Inquiry
tcst	125	JAF-CALC-MISC-02806, R0
tdw_maxind	440	UFSAR Table 7.19-1; ROME Equipment Inquiry for 16-1TR-107 & 16-1TR-108; ISP 28-1
tdw_maxop	135	UFSAR Table 5.2-3; OP-53, Drywell Ventilation and Cooling
tdw_minop	100	Assumption, substantiated by review of data logged during performance of ST-40D for the period 1/3/98 to 3/21/98 (dates from cover sheet). Weighting factors for individual areas obtained from 1987 PCILRT report, attachments 3.2D and 3.2E.
tsc_max	220	Dwg. 3.11-44
tsc_maxop	110	Technical Specification 3.7.A.1.c.(3) limit on torus water temperature. Assumes airspace temperature is in equilibrium with water temperature
tsc_minop	68	OP-13A, RHR - Low Pressure Coolant Injection & OP-13B, RHR - Containment Control limit on pool water temperature. Assumes airspace temperature is in equilibrium with water temperature.
tslc	104	JAF-CALC-MISC-02806, R0
tsp_flood	42	JAF-CALC-PC-03125, R1
tsp_minop	68	OP-13A, RHR - Low Pressure Coolant Injection; OP-13B, RHR - Containment Control

Variable	Value	Source
tsp_scram	110	Technical Specification 3.7.A.1.c.(3)
vdw	176,397	JAF-CALC-PC-03125, R1
vrpv_taf	1.03e4	JAF-CALC-MISC-01966, R1
vsc_dco	154763	JAF-CALC-PC-03044, R0
vsc_lco	113603	JAF-CALC-PC-03044, R0
wlrpv_baf	-144	UFSAR Table 7.8-1 describes reference point for RPV water level indication
wlsp_maxlco	14	Technical Specification 3.7.A.1.a
wlsp_minlco	13.88	Technical Specification 3.7.A.1.b
wlsp_srv	14	NYPA memorandum NED-CS-94-WP51-VU1242, James A. FitzPatrick Nuclear Power Plant Additional Emergency Procedure Guideline SRV Input Data Parameters, 18 May 1994
wlsp_tp	14	NEDE-21935P, "Mark I Containment Program T-Quencher Load Definition Report," August 1978, p. 5-5
wslc	50	Technical Specification 4.4.A.2
wsrv_name	829000	OPL-3, R0, Transient Protection Parameters Verification for Reload Licensing Analyses - FitzPatrick - Reload 13 - Cycle 14 - Final
xb_cld_nat	660	GE letter JAB-N7018, "Plant Emergency Procedure Parameters Affected by Fuel Design," March 14, 1997; GE Letter JAB-N8072, "Plant Emergency Procedure Parameters," October 27, 1998

Variable	Value	Source
xb_hot_nat	522	GE letter JAB-N7018, "Plant Emergency Procedure Parameters Affected by Fuel Design," March 14, 1997; GE Letter JAB-N8072, "Plant Emergency Procedure Parameters," October 27, 1998
xb_slc	1.832e4	JAF-CALC-SLC-03126, R0

srv discharge line length

Torus water level [ft]	SRV water leg length [ft]
13.88	0
14.66	0.86
15.5	8.12
18.51	11.21
29.5	42.79

Source JAF-CALC-MISC-01521, Set No. 2, Rev. 0

torus volumes

JAF-CALC-MISC-03116, Rev. 0 Attachment 10

Torus water level [ft]	Torus water volume [ft3]	Torus air volume [ft3]
9.58	65339	154763
10	69214	150890
11	78589	141436
12	88145	131960
13	97840	122200
13.88	106442	113603
14	107619	112428
15	117174	102119
16	126432	92820
17	135793	83463
17.85	143927	75334

Source JAF-CALC-PC-03044, Rev. 0

```
thermin.dat
thermout.dat
u
u
t
2
n
n
vo,cp
```

```
15
32,600,1
c
500
32,600,1
c
1000
32,600,1
n
3
32,600,1
n
4
1,1500,1
q
```

```
// do eop calculations.cpp:
//      This file contains the instructions required to control performance
//      of an integrated set of EOP support calculations.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

#include<fstream.h>
#include<iomanip.h>
#include<iostream.h>
#include<math.h>
#include<stdlib.h>
#include<string.h>
#include<strstrea.h>

#include "plant data.h"
#include "thermo.h"
#include "eop calculation input.h"
#include "shared data.h"
#include "shared functions.h"
#include "mpspcwl.h"
#include "pcpl.h"
#include "stpll.h"
#include "hctl.h"
#include "rpv.h"
#include "boron.h"
#include "psp.h"
#include "drywell spray.h"
#include "mdrir.h"
#include "scsip.h"

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

char    inter_name[30];     // name of file to store intermediate results
char    final_name[30];     // name of file to store final results

//*************************
//
// FUNCTION PROTOTYPE
//
//*************************

// None: do_eop_calcs only provides 'main()' function for program

int main(void)
{
    int     hold;
    int     test_good;  // variable indicating success of operations performed
                        // by load_plant_data() and load_thermo() functions

    test_good = load_plant_data();

    // non-zero value indicates the function did not complete successfully
    // put a delay in loop to allow looking at results on screen

    if (test_good != 0) {
        cout << '\n' << "type a number to continue" << '\n';
        cin >> hold;
        cout << '\n';
        exit(0);
```

```
}

test_good = load_thermo();

// non-zero value indicates the function did not complete successfully
// put a delay in loop to allow looking at results on screen

if (test_good != 0) {
    cout << '\n' << "type a number to continue" << '\n';
    cin >> hold;
    cout << '\n';
    exit(0);
}

// assign file names, create files (ios::out is specified
// to ensure program starts with a clean file)and write headers
// for storage of intermediate and final results.

// files are closed after headers written to assure
// file buffer is flushed

strcpy(inter_name, "eop calc intermediate results");
ofstream inter_out(inter_name, ios::out);
inter_out << "EOP calculation intermediate results:\n\n";
inter_out.close();

strcpy(final_name, "eop calculation final results");
ofstream final_out(final_name, ios::out);
final_out << "EOP calculation final results:\n\n";
final_out.close();

// call function to write inputs used for calculation
// to the file containing final results

write_input_values(final_name);

// do the calculation

calc_mpspcwl(final_name);

sequence_pcpl(inter_name, final_name);

calc_stpll(inter_name, final_name);

calc_tsp_hctl_2(inter_name);

sequence_rpv(inter_name, final_name);

calc_hctl(inter_name, final_name);

sequence_boron(inter_name, final_name);

calc_psp(inter_name, final_name);

calc_dwisl(inter_name, final_name);

calc_mdsf(final_name);

calc_mdrir(inter_name, final_name);

calc_scsip(inter_name, final_name);

// release memory used for storage of
// thermodynamic properties of water

free_thermo();

return (0); // we be done!!
```

```
} // end main
```

```
// plant_data.h:
//      This file declares structures and variables to contain
//      plant-specific and generic parameters used for calculation
//      of variables and curves required by the JAF Emergency
//      Procedure and Severe Accident Guidelines and their derivative
//      procedures. Functions that initialize these variables are
//      also provided.
//
//      Data required for the EPG Calculations is described
//      in Appendix C of the BWROG EPGs/SAGs, Rev. 1
//      dated July, 1997.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// STRUCTURE DECLARATIONS
//
//*************************

// data describing primary containment strength

struct  containment_data {
    double  h_n;            // elevation of member, ft
    double  location;       // drywell (DW) or wetwell (WW)
    double  material;       // material of construction
    double  pcalc_n;        // pressure capability of member
    double  strength;       // basis for pressure capability
    double  tcalc_n;        // temperature at which capability determined
};

// data required to calculate maximum core uncovery time limit

struct  core_heatup {
    double  t_sd[9];        // time after shutdown [min.]
    double  t_cu_15[9];     // time to reach 1500°F for core uncovery [min.]
};

// generic decay heat data

struct  decay_heat {
    double  time;           // time after shutdown [min.]
    double  power;          // decay heat as percent of rated power [%]
};

// data describing low pressure ECCS pump flow charactristics

struct  eccs_pump_flow {
    double  flow;           // pump flow rate [gpm]
    double  rpv_press;      // corresponding RPV pressure [psig]
};

// generic normalized material strengths

struct  material_strength {
    double  temp[4];        // temperature [°F]
    double  fyield[4];      // normalized yield strength [fraction]
    double  ftens[4];       // normalized tensile strength [fraction]
};

// data describing drywell spray headers

struct  spray_hdr_data {
```

```
    double  num_nzls;        // number of nozzles assigned to spray subsystem
    double  wnzl;            // flow rate required to develop spray pattern
};

// data describing SRV tail pipe length vs. torus water level

struct  srvdl_data {
    double  spwl;            // torus water level [ft]
    double  wll;             // srv water leg length [ft]
};

// data describing torus air and water volumes as a function of height

struct torus_volume {
    double  wlsp;            // torus water level [ft]
    double  vsc_air;         // torus airspace volume [ft3]
    double  vsp;             // torus water volume [ft3]
};

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

// plant specific parameter list

int     nbr_fuel_types;      // number of fuel types used for fafl_15, etc.
int     nbr_pc_members;      // number of PC members for which
                             // structural properties are defined
char    BWR_type[6];         // Reactor system product line
char    BWR_PSC_type[7];     // Containment system product line
double  c_clad;              // specific heat of clad [Btu/lbm-°F]
double  c_fuel;              // specific heat of fuel [Btu/lbm-°F]
double  dpdw_ww;             // drywell below wetwell dP capability [psid]
double  dpsrv;               // minimum pneumatic supply dP to open SRVs [psid]
spray_hdr_data  spray_hdr[2];   // drywell spray header data
double  fafl_15[10];            // active fuel length fraction required to be covered to
                                // maintain PCT < 1500°F as a function of fuel type
double  fafl_18[10];         // active fuel length fraction required to be covered to
                             // maintain PCT < 1800°F as a function of fuel type
double  fb10_slc;            // B-10 isotopic enrichment in SLC tank
double  hdco;                // elevation of downcomer openings [ft]
double  hring;               // elevation of vent header (in torus) [ft]
double  hsc_tap;             // elevation of torus instrument tap [ft]
double  hsp_ref;             // elevation of torus water level instrument zero [ft]
double  hsrv;                // elevation of lowest SRV pneumatic solenoid [ft]
double  hvent_pc;            // elevation of primary containment vent [ft]
double  lfuel[10];           // length of active fuel as a function of fuel type [in.]
double  mclad;               // mass of fuel clad and channels [lbm]
double  mf_rpv_cld;          // mass of water in RWR, SDC and RWCU loops with
                             // RPV water level at L8 trip and water temperature at 68°F [lbm]
double  mf_rpv_hot;          // mass of water in RWR and RWCU loops with RPV
                             // water level at L8 trip and water temperature at saturation
                             // temperature for minimum SRV lift pressure [lbm]
double  mfuel;               // mass of fuel [lbm]
double  mg_rpv_hot;          // mass of saturated steam in RPV and main steam lines inboard
                             // of outboard MSIVs with water level at L8 trip and pressure
                             // at minimum SRV lift pressure [lbm]
double  mrpv;                // mass of RPV, internals, RWR loops and main steam
                             // lines inboard of outboard MSIVs [lbm]
double  mrpv_taf;            // mass of RWR loops and RPV and internals below TAF [lbm]
int     nbuns;               // number of fuel bundles
int     nsrv_ads;            // number of SRVs assigned to ADS
containment_data    pc_data[10];
double  pdw_maxop;           // maximum normal operating drywell pressure [psig]
double  pdw_minop;           // minimum normal operating drywell pressure [psig]
double  pdw_scram;           // high drywell pressure scram setpoint [psig]
```

```
double  ppc_maxind;          // maximum indicated drywell pressure [psig]
double  ppc_vent;            // maximum pressure at which drywell vent can be operated [psig]
double  pq_code;             // code allowable stress for SRV quencher [kpsi]
double  pq_des;              // design basis stress for SRV quencher [kpsi]
double  pqs_code;            // code allowable stress for SRV quencher support [kpsi]
double  pqs_des;             // design basis stress for SRV quencher support [kpsi]
double  prpv_maxind;         // Maximum indicated RPV pressure [psig]
double  prpv_tp;             // RPV pressure used for SRV tail pipe design calc's [psig]
double  psc_maxop;           // maximum normal operating torus pressure [psig]
double  psc_minop;           // minimum normal operating torus pressure [psig]
double  psp_des;             // torus boundary design load [psig]
double  psp_srv;             // maximum load on torus boundary from SRV actuation [psi]
double  psrv_lift;           // minimum SRV lift pressure [psig]
double  psrv_name;           // SRV nameplate pressure [psig]
double  psup_srv;            // minimum normal operating SRV pneumatic supply pressure [psig]
double  ptp_code;            // code allowable stress for SRV tail pipe [kpsi]
double  ptp_des;             // design basis stress for SRV tail pipe [kpsi]
double  ptps_code;           // code allowable stress for SRV tail pipe support [kpsi]
double  ptps_des;            // design basis stress for SRV tail pipe support [kpsi]
double  qrx_rated;           // rated reactor power [MWt]
double  slc_tk_dia;          // SLC tank diameter [ft]
double  slc_tk_lvl_in_span;  // SLC tank level instrument input span [in.]
double  slc_tk_lvl_out_span;  // SLC tank level instrument output span [%]
char    srv_type[30];
core_heatup    t_to_1500[10];  // core uncovery time limit data
double  tcst;                // CST temperature [°F]
double  tdw_maxop;           // maximum normal operating drywell temperature [°F]
double  tdw_maxind;          // maximum indicated drywell temperature [°F]
double  tdw_minop;           // minimum normal operating drywell temperature [°F]
double  tsc_max;             // maximum temperature capability of torus and contained
                             // equipment which may be required to operate when the
                             // RPV is pressurized [°F]
double  tsc_maxop;           // maximum normal operating torus airspace temperature [°F]
double  tsc_minop;           // minimum normal operating torus airspace temperature [°F]
double  tslc;                // maximum normal operating SLC tank temperature [°F]
double  tsp_flood;           // minimum temperature of torus
                             // when containment flooded [°F]
double  tsp_minop;           // minimum normal operating torus water temperature [°F]
double  tsp_scram;           // torus water temperature at which scram is required [F]
double  vdw;                 // free volume of drywell and vent system [ft3]
double  vrpv_taf;            // free volume of RWR, RWCU and SDC loops and
                             // RPV below TAF [ft3]
double  vsc_dco;             // torus free volume above downcomer openings [ft3]
double  vsc_lco;             // torus free volume above minimum torus water level LCO [ft3]
torus_volume vtorus[11];     // torus water and air volume as a function of level
double  wg_1500[10];         // bundle steam flow required to maintain PCT < 1500°F [lbm/hr]
srvdl_data  srvdl[5];        // SRV water leg length (in tail pipe) data
eccs_pump_flow  lpci_flow[10];  // ECCS pump flow versus RPV pressure
eccs_pump_flow  cs_flow[10];
double  wlrpv_baf;           // RPV water level at bottom of active fuel [in.]
double  wlsp_maxlco;         // torus maximum water level LCO [ft]
double  wlsp_minlco;         // torus minimum water level LCO [ft]
double  wlsp_srv;            // torus water level used to establish SRV boundary load [ft]
double  wlsp_tp;             // torus water level used for SRV tail pipe load calc's [ft]
double  wslc;                // SLC flowrate [gpm]
double  wsrv_name;           // SRV nameplate flow rate [lbm/hr]
double  xb_cld_nat;          // cold shutdown boron concentration [ppm]
double  xb_hot_nat;          // hot shutdown boron concentration [ppm]
double  xb_slc;              // minimum normal operating concentration
                             // of boron in SLC tank [ppm]

// generic parameter list

double  c_steel;             // specific heat of steel [Btu/lbm-°F]
double  dptp_dprpv;          // SRV tail pipe load variation with
                             // RPV pressure [%/psig]
double  dptp_dwll;           // SRV tail pipe load variation with
                             // SRVDL water leg length [%/ft]
```

```
double  fb10_nat;           // B10 isotope fraction in natural boron
double  fb11_nat;           // B11 isotope fraction in natural boron
double  fqdh_10;            // 10 minute decay heat fraction
double  fqdh_100;           // 100 minute decay heat fraction
double  fqdh_2;             // 2 minute decay heat fraction
double  k1;                 // conversion constant [Btu/hr-MWt]
double  k2;                 // conversion constant [Btu/min.-MWt]
double  k3;                 // conversion constant [gal/ft3]
double  mb10;               // nuclidic mass of B10 isotope
double  mb11;               // nuclidic mass of B11 isotope
material_strength   pc_mat[5];  // material properties for primary containment
decay_heat  sd_pwr[25];     // decay heat as a function of time after shutdown
double  srv_open;           // SRV re-opening pressure [psid]
double  tlim_dw;            // drywell limiting temperature for pcpl calculation [°F]
double  tlim_sp;            // torus limiting temperature for pcpl calculation [°F]

//*************************
//
// FUNCTION PROTOTYPES
//
//*************************

// prototype of a function which reads a single integer from
// the input stream (through memory) to a variable

int     assign_int(char line_as_string[80]);

// prototype of a function which reads a single double from
// the input stream (through memory) to a variable

double  assign_float(char line_as_string[80]);

// prototype of a function which reads up to eleven floats
// from the input stream (through memory) to a variable

void    assign_n_floats(char line_as_string[80], double return_floats[11], int n_floats);

// prototype of function which initializes plant and generic data

int     load_plant_data(void);

//*************************
// assign_int function
//*************************

int     assign_int(char line_as_string[80])
{
    int     return_value;

    istrstream data_string(line_as_string, 80);
    data_string >> return_value;

    return (return_value);

} // end assign_int

//*************************
// assign_float function
//*************************

double  assign_float(char line_as_string[80])
{
    double  return_value;

    istrstream data_string(line_as_string, 80);
    data_string >> return_value;

    return (return_value);
```

```
} // end assign_float

//*************************
// assign_n_floats function
//*************************

void    assign_n_floats(char line_as_string[80], double return_floats[11], int n_floats)
{
    int     index;

    istrstream data_string(line_as_string, 80);
    for (index = 0; index < n_floats; ++index) {
        data_string >> return_floats[index];
    }

} // end assign_n_floats

//*************************
// load_plant_data function
//*************************

int     load_plant_data(void)
{
    // Build database for plant specific parameters.
    // Define structure to pull input file into with a
    // 'single' read, along with a pointer to the structure.

    struct  whole_file {
        char    input_line[80];
    };
    whole_file  *line_ptr1;
    whole_file  *line_ptr2;

    char    input_file[35];     // name of file to get input from
    char    read_line[80];      // individual lines of input from file
    int     index;              // index used to populate arrays
    int     master_index;       // index used to find place in input structure
    char    line_as_string[80]; // individual lines of input, expressed as strings
    double  floats_in[11];
    int     test_open;          // flag for success in opening specified file

    // get file name for plant-specific parameters

    strcpy(input_file, "plant data");

    // construct an ifstream object without opening a file

    ifstream data_file1;

    // open an existing file (i.e., don't make one if it isn't there)

    data_file1.open(input_file, ios::nocreate);

    // get status of call to open file

    test_open = data_file1.is_open();

    // check for success and act appropriately

    if (test_open == 0) {
        cout << "Error: could not open " << input_file << '\n';
        return (1); // non-zero value tells calling program that
                    // 'plant data' could not be opened
    }

    // get memory to store input file
```

```
line_ptr1 = new whole_file[148];

// populate the array

cout << "reading " << input_file << '\n' << flush;

for (index = 0; index < 148; ++index) {
    data_file1.getline(read_line, sizeof(read_line));
    strcpy((line_ptr1 + index)->input_line, read_line);
}

// be neat, close the input file

data_file1.close();

master_index = 0;        // initialize master pointer into input array
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
nbr_fuel_types = assign_int(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
nbr_pc_members = assign_int(line_as_string);

++master_index;
strcpy(BWR_type, (line_ptr1 + master_index)->input_line);

++master_index;
strcpy(BWR_PSC_type, (line_ptr1 + master_index)->input_line);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
c_clad = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
c_fuel = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
dpdw_ww = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
dpsrv = assign_float(line_as_string);

for (index = 0; index < 2; ++index) {
    ++master_index;
    strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
    assign_n_floats(line_as_string, floats_in, 2);
    spray_hdr[index].num_nzls = floats_in[0];
    spray_hdr[index].wnzl = floats_in[1];
}

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
assign_n_floats(line_as_string, fafl_15, nbr_fuel_types);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
assign_n_floats(line_as_string, fafl_18, nbr_fuel_types);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
fb10_slc = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
hdco = assign_float(line_as_string);
```

```
++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
hring = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
hsc_tap = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
hsp_ref = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
hsrv = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
hvent_pc = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
assign_n_floats(line_as_string, lfuel, nbr_fuel_types);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
mclad = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
mf_rpv_cld = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
mf_rpv_hot = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
mfuel = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
mg_rpv_hot = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
mrpv = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
mrpv_taf = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
nbuns = assign_int(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
nsrv_ads = assign_int(line_as_string);

for (index = 0; index < nbr_pc_members; ++index) {
    ++master_index;
    strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
    assign_n_floats(line_as_string, floats_in, 6);
    pc_data[index].h_n = floats_in[0];
    pc_data[index].location = floats_in[1];
    pc_data[index].material = floats_in[2];
```

```
    pc_data[index].pcalc_n = floats_in[3];
    pc_data[index].strength = floats_in[4];
    pc_data[index].tcalc_n = floats_in[5];
}

// reset reference to memory to skip 'pad lines' in input

master_index = master_index + 11 - nbr_pc_members;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
pdw_maxop = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
pdw_minop = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
pdw_scram = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
ppc_maxind = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
ppc_vent = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
pq_code = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
pq_des = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
pqs_code = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
pqs_des = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
prpv_maxind = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
prpv_tp = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
psc_maxop = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
psc_minop = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
psp_des = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
psp_srv = assign_float(line_as_string);

++master_index;
```

```
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
psrv_lift = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
psrv_name = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
psup_srv = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
ptp_code = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
ptp_des = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
ptps_code = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
ptps_des = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
qrx_rated = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
slc_tk_dia = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
slc_tk_lvl_in_span = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
slc_tk_lvl_out_span = assign_float(line_as_string);

++master_index;
strcpy(srv_type, (line_ptr1 + master_index)->input_line);

for (index = 0; index < nbr_fuel_types; ++index) {
    ++master_index;
    strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
    assign_n_floats(line_as_string, t_to_1500[index].t_sd, 9);
    ++master_index;
    strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
    assign_n_floats(line_as_string, t_to_1500[index].t_cu_15, 9);
}

// reset reference to memory to account for 'pad lines' in input

master_index = master_index + (2 * (10 - nbr_fuel_types)) + 1;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
tcst = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
tdw_maxop = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
tdw_maxind = assign_float(line_as_string);
```

```
++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
tdw_minop = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
tsc_max = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
tsc_maxop = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
tsc_minop = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
tslc = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
tsp_flood = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
tsp_minop = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
tsp_scram = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
vdw = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
vrpv_taf = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
vsc_dco = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
vsc_lco = assign_float(line_as_string);

for (index = 0; index < 11; ++index) {
    ++master_index;
    strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
    assign_n_floats(line_as_string, floats_in, 3);
    vtorus[index].wlsp = floats_in[0];
    vtorus[index].vsp= floats_in[1];
    vtorus[index].vsc_air = floats_in[2];
}

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
assign_n_floats(line_as_string, wg_1500, nbr_fuel_types);

for (index = 0; index < 5; ++index) {
    ++master_index;
    strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
    assign_n_floats(line_as_string, floats_in, 2);
    srvdl[index].wll = floats_in[0];
    srvdl[index].spwl = floats_in[1];
```

```
}

for (index = 0; index < 10; ++index) {
    ++master_index;
    strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
    assign_n_floats(line_as_string, floats_in, 2);
    lpci_flow[index].flow = floats_in[0];
    lpci_flow[index].rpv_press = floats_in[1];
}

for (index = 0; index < 10; ++index) {
    ++master_index;
    strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
    assign_n_floats(line_as_string, floats_in, 2);
    cs_flow[index].flow = floats_in[0];
    cs_flow[index].rpv_press = floats_in[1];
}

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
wlrpv_baf = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
wlsp_maxlco = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
wlsp_minlco = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
wlsp_srv = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
wlsp_tp = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
wslc = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
wsrv_name = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
xb_cld_nat = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
xb_hot_nat = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr1 + master_index)->input_line);
xb_slc = assign_float(line_as_string);

// release memory used for storage of input data file

delete[] line_ptr1;

//*************************
// build database for generic parameters
//*************************

// get file name for generic parameters
```

```
strcpy(input_file, "generic data");

// construct an ifstream object without opening a file

ifstream data_file2;

// open an existing file (i.e., don't make one if it isn't there)

data_file2.open(input_file, ios::nocreate);

// get status of call to open file

test_open = data_file2.is_open();

// check for success and act appropriately

if (test_open == 0) {
    cout << "Error: could not open " << input_file << '\n';
    return (1); // non-zero value tells calling program that
                // 'generic data' could not be opened
}

// get memory to store input file

line_ptr2 = new whole_file[56];

// populate the array

cout << "reading " << input_file << '\n' << '\n' << flush;

for (index = 0; index < 56; ++index) {
    data_file2.getline(read_line, sizeof(read_line));
    strcpy((line_ptr2 + index)->input_line, read_line);
}

// be neat, close the input file

data_file2.close();

master_index = 0;       // initialize master pointer into input array
strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
c_steel = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
dptp_dprpv = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
dptp_dwll = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
fb10_nat = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
fb11_nat = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
fqdh_10 = assign_float(line_as_string);

++master_index;
strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
fqdh_100 = assign_float(line_as_string);

++master_index;
```

```
    strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
    fqdh_2 = assign_float(line_as_string);

    ++master_index;
    strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
    k1 = assign_float(line_as_string);

    ++master_index;
    strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
    k2 = assign_float(line_as_string);

    ++master_index;
    strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
    k3 = assign_float(line_as_string);

    ++master_index;
    strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
    mb10 = assign_float(line_as_string);

    ++master_index;
    strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
    mb11 = assign_float(line_as_string);

    for (index = 0; index < 5; ++index) {
        ++master_index;
        strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
        assign_n_floats(line_as_string, pc_mat[index].temp, 4);
        ++master_index;
        strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
        assign_n_floats(line_as_string, pc_mat[index].fyield, 4);
        ++master_index;
        strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
        assign_n_floats(line_as_string, pc_mat[index].ftens, 4);
    }

    for (index = 0; index < 25; ++index) {
        ++master_index;
        strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
        assign_n_floats(line_as_string, floats_in, 2);
        sd_pwr[index].time = floats_in[0];
        sd_pwr[index].power = floats_in[1];
    }

    ++master_index;
    strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
    srv_open = assign_float(line_as_string);

    ++master_index;
    strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
    tlim_dw = assign_float(line_as_string);

    ++master_index;
    strcpy(line_as_string, (line_ptr2 + master_index)->input_line);
    tlim_sp = assign_float(line_as_string);

    delete[] line_ptr2;

    return (0); // function executed successfully

} // end load_plant_data
```

```
// thermo.h:
//      this file declares structures, variables and functions used to
//      calculate the thermodynamic properties of water for use in
//      other programs.
//
//      The following properties are accessible:
//      Saturated fluid: saturation pressure, saturation temperature,
//      specific enthalpy, specific entropy, specific volume
//      and specific heat. Subcooled or superheated fluid: specific
//      enthalpy, specific entropy, specific volume and specific
//      heat.
//
//      The values are stored in the following increments:
//      Saturated fluid:    By pressure: 1 psia to 1500 psia by 1 psia;
//                          By temperature: 32 degf to 600 degf by 1 degf
//      Subcooled/Superheated fluid: at 15 psia, 500 psia and 1000 psia
//                          for the range from 32 degf to 600 degf
//                          by 1 degf
//
//      Raw data were generated by using the ASME software for
//      Properties of Water and Steam Using the 1967 IFC Formulation
//      for Industrial Use, writing the results to file.
//
//      The protoype of a function defining a user interface to the stored
//      thermodynamic data is also included.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// STRUCTURE DECLARATIONS
//
//*************************

// define the structures used to store subcooled/superheated and saturated water properties

struct  subcooled {
    double  pressure;
    double  temperature;
    double  h;
    double  s;
    double  v;
    double  cp;
};

struct  saturated {
    double  pressure;
    double  temperature;
    double  hf, hg;
    double  sf, sg;
    double  vf, vg;
    double  cpf, cpg;
};

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

// define pointers to arrays used to store subcooled/superheated
// water properties at 15, 500 and 1000 psia
```

```
subcooled   *sub15_ptr;
subcooled   *sub500_ptr;
subcooled   *sub1000_ptr;

// define pointers to arrays used to store saturated
// water properties by temperature and pressure

saturated   *satbytemp_ptr;
saturated   *satbypress_ptr;

//*************************
//
// FUNCTION PROTOTYPES
//
//*************************

// prototype of a function to initialize variables containing thermodynamic data

int     load_thermo(void);

// protoype of a function that frees memory which held thermodynamic data

void    free_thermo(void);

// prototypes of functions which return various thermodynamic properties

// functions which return saturated fluid properties as a function of
// pressure, these require input of pressure [psia] and return the value
// signified by the name of the function

double  tsat(double pressure_in);
double  hfsat_by_p(double pressure_in);
double  hgsat_by_p(double pressure_in);
double  sfsat_by_p(double pressure_in);
double  sgsat_by_p(double pressure_in);
double  vfsat_by_p(double pressure_in);
double  vgsat_by_p(double pressure_in);
double  cpfsat_by_p(double pressure_in);
double  cpgsat_by_p(double pressure_in);

// functions which return saturated fluid properties as a function of
// temperature, these require input of temperature [degf] and return
// the value signified by the name of the function

double  psat(double temperature_in);
double  hfsat_by_t(double temperature_in);
double  hgsat_by_t(double temperature_in);
double  sfsat_by_t(double temperature_in);
double  sgsat_by_t(double temperature_in);
double  vfsat_by_t(double temperature_in);
double  vgsat_by_t(double temperature_in);
double  cpfsat_by_t(double temperature_in);
double  cpgsat_by_t(double temperature_in);

// functions which return subcooled or superheated fluid properties
// as a function of pressure and temperature, these require input of
// pressure [psia] and temperature [degf] and return the value signified
// by the name of the function

double  hpt(double pressure_in, double temperature_in);
double  spt(double pressure_in, double temperature_in);
double  vpt(double pressure_in, double temperature_in);
double  cppt(double pressure_in, double temperature_in);

//*************************
// load_thermo function
//*************************
```

```
int     load_thermo(void)
{

    // define structure to pull input file into with a
    // 'single' read, along with a pointer to the structure

    struct  whole_file {
        char    input_line[80];
    };
    whole_file  *line_ptr;

    char    input_file[35];         // name of file to get input from
    char    read_line[80];          // individual lines of input from file
    int     index;                  // index used to populate arrays
    int     master_index;           // index used to find place in input structure
    char    line_as_string[80];     // individual lines of subcooled/superheated data
    char    line1_as_string[80];    // individual lines of saturated data
    char    line2_as_string[80];    // two lines required since input is on two lines
    int     test_open;              // flag for success in opening specified file

    // get file name for thermodynamic data

    strcpy(input_file, "thermo data");

    // construct an ifstream object without opening a file

    ifstream data_file;

    // open an existing file (i.e., don't make one if it isn't there)

    data_file.open(input_file, ios::nocreate);

    // get status of call to open file

    test_open = data_file.is_open();

    // check for success and act appropriately

    if (test_open == 0) {
        cout << "Error: could not open " << input_file << '\n';
        return (1); // non-zero value tells calling program that
                    // 'thermo data' could not be opened
    }

    // get memory to store input file

    line_ptr = new whole_file[5895];

    // populate the array

    cout << "reading " << input_file << '\n' << flush;

    for (index = 0; index < 5895; ++index) {
        data_file.getline(read_line, sizeof(read_line));
        strcpy((line_ptr + index)->input_line, read_line);
    }

    // be neat, close the input file

    data_file.close();

    // get memory for subcooled/superheated data

    sub15_ptr = new subcooled[569];
    sub500_ptr = new subcooled[569];
    sub1000_ptr = new subcooled[569];

    // get memory for saturated data
```

```
satbytemp_ptr = new saturated[569];
satbypress_ptr = new saturated[1500];

cout << "processing 15 psia subcooled data\n" << flush;

// read subcooled 15 psia data into storage array pointed to by sub15_ptr
// subcooled data begins on line 16 of input file, which is stored beginning
// at index 15 of array pointed to by line_ptr

master_index = 15;
for (index = 0; index < 182; ++index) {
    strcpy(line_as_string, (line_ptr + master_index)->input_line);
    istrstream data_string(line_as_string, 80);
    ++master_index;
    data_string >> (sub15_ptr + index)->pressure;
    data_string >> (sub15_ptr + index)->temperature;
    data_string >> (sub15_ptr + index)->h;
    data_string >> (sub15_ptr + index)->s;
    data_string >> (sub15_ptr + index)->v;
    data_string >> (sub15_ptr + index)->cp;
}

// blow past saturation line data for 15 psia by adding 2 to master_index

master_index = master_index + 2;

// read superheated 15 psia data into storage array pointed to by sub15_ptr

cout << "processing 15 psia superheated data\n" << flush;

for (index = 182; index < 569; ++index) {
    strcpy(line_as_string, (line_ptr + master_index)->input_line);
    istrstream data_string(line_as_string, 80);
    ++master_index;
    data_string >> (sub15_ptr + index)->pressure;
    data_string >> (sub15_ptr + index)->temperature;
    data_string >> (sub15_ptr + index)->h;
    data_string >> (sub15_ptr + index)->s;
    data_string >> (sub15_ptr + index)->v;
    data_string >> (sub15_ptr + index)->cp;
}

cout << "processing 500 psia subcooled data\n" << flush;

// read subcooled 500 psia data into storage array pointed to by sub500_ptr
// subcooled data begins on line 594 of input file, which is stored beginning
// at index 593 of array pointed to by line_ptr

master_index = 593;

for (index = 0; index < 436; ++index) {
    strcpy(line_as_string, (line_ptr + master_index)->input_line);
    istrstream data_string(line_as_string, 80);
    ++master_index;
    data_string >> (sub500_ptr + index)->pressure;
    data_string >> (sub500_ptr + index)->temperature;
    data_string >> (sub500_ptr + index)->h;
    data_string >> (sub500_ptr + index)->s;
    data_string >> (sub500_ptr + index)->v;
    data_string >> (sub500_ptr + index)->cp;
}

// blow past saturation line data for 500 psia by adding 2 to master_index

master_index = master_index + 2;

// read superheated 500 psia data into storage array pointed to by sub500_ptr
```

```
cout << "processing 500 psia superheated data\n" << flush;

for (index = 436; index < 569; ++index) {
    strcpy(line_as_string, (line_ptr + master_index)->input_line);
    istrstream data_string(line_as_string, 80);
    ++master_index;
    data_string >> (sub500_ptr + index)->pressure;
    data_string >> (sub500_ptr + index)->temperature;
    data_string >> (sub500_ptr + index)->h;
    data_string >> (sub500_ptr + index)->s;
    data_string >> (sub500_ptr + index)->v;
    data_string >> (sub500_ptr + index)->cp;
}

cout << "processing 1000 psia subcooled data\n" << flush;

// read subcooled 1000 psia data into storage array pointed to by sub1000_ptr
// subcooled data begins on line 1172 of input file, which is stored beginning
// at index 1171 of array pointed to by line_ptr

master_index = 1171;

for (index = 0; index < 513; ++index) {
    strcpy(line_as_string, (line_ptr + master_index)->input_line);
    istrstream data_string(line_as_string, 80);
    ++master_index;
    data_string >> (sub1000_ptr + index)->pressure;
    data_string >> (sub1000_ptr + index)->temperature;
    data_string >> (sub1000_ptr + index)->h;
    data_string >> (sub1000_ptr + index)->s;
    data_string >> (sub1000_ptr + index)->v;
    data_string >> (sub1000_ptr + index)->cp;
}

// blow past saturation line data for 1000 psia by adding 2 to master_index

master_index = master_index + 2;

// read superheated 1000 psia data into storage array pointed to by sub1000_ptr

cout << "processing 1000 psia superheated data\n" << flush;

for (index = 513; index < 569; ++index) {
    strcpy(line_as_string, (line_ptr + master_index)->input_line);
    istrstream data_string(line_as_string, 80);
    ++master_index;
    data_string >> (sub1000_ptr + index)->pressure;
    data_string >> (sub1000_ptr + index)->temperature;
    data_string >> (sub1000_ptr + index)->h;
    data_string >> (sub1000_ptr + index)->s;
    data_string >> (sub1000_ptr + index)->v;
    data_string >> (sub1000_ptr + index)->cp;
}

cout << "processing saturation by temperature data for 32 degf to 200 degf\n" << flush;

// read saturation line data by temperature into storage array pointed
// to by satbytemp_ptr. data begins on line 1750 of input file, which
// is stored beginning at index 1749 of array pointed to by line_ptr

master_index = 1749;

for (index = 0; index < 569; ++index) {
    strcpy(line1_as_string, (line_ptr + master_index)->input_line);
    istrstream data_string1(line1_as_string, 80);
    ++master_index;
    strcpy(line2_as_string, (line_ptr + master_index)->input_line);
```

```
    istrstream data_string2(line2_as_string, 80);
    ++master_index;
    data_string1 >> (satbytemp_ptr + index)->pressure;
    data_string1 >> (satbytemp_ptr + index)->temperature;
    data_string1 >> (satbytemp_ptr + index)->hf;
    data_string1 >> (satbytemp_ptr + index)->sf;
    data_string1 >> (satbytemp_ptr + index)->vf;
    data_string1 >> (satbytemp_ptr + index)->cpf;
    data_string2 >> (satbytemp_ptr + index)->hg;
    data_string2 >> (satbytemp_ptr + index)->sg;
    data_string2 >> (satbytemp_ptr + index)->vg;
    data_string2 >> (satbytemp_ptr + index)->cpg;
    if (index == 168) {
        cout << "processing saturation by temperature data for 201 degf";
        cout << " to 400 degf\n" << flush;
    }
    if (index == 368) {
        cout << "processing saturation by temperature data for 401 degf";
        cout << " to 600 degf\n" << flush;
    }
}

cout << "processing saturation by pressure data for 1 psia to 200 psia\n" << flush;

// read saturation line data by pressure into storage array pointed
// to by satbypress_ptr. data begins on line 2895 of input file, which
// is stored beginning at index 2894 of array pointed to by line_ptr

master_index = 2894;

for (index = 0; index < 1500; ++index) {
    strcpy(line1_as_string, (line_ptr + master_index)->input_line);
    istrstream data_string1(line1_as_string, 80);
    ++master_index;
    strcpy(line2_as_string, (line_ptr + master_index)->input_line);
    istrstream data_string2(line2_as_string, 80);
    ++master_index;
    data_string1 >> (satbypress_ptr + index)->pressure;
    data_string1 >> (satbypress_ptr + index)->temperature;
    data_string1 >> (satbypress_ptr + index)->hf;
    data_string1 >> (satbypress_ptr + index)->sf;
    data_string1 >> (satbypress_ptr + index)->vf;
    data_string1 >> (satbypress_ptr + index)->cpf;
    data_string2>> (satbypress_ptr + index)->hg;
    data_string2 >> (satbypress_ptr + index)->sg;
    data_string2 >> (satbypress_ptr + index)->vg;
    data_string2 >> (satbypress_ptr + index)->cpg;
    if (index == 200) {
        cout << "processing saturation by pressure data for 201 psia";
        cout << " to 400 psia\n" << flush;
    }
    if (index == 400) {
        cout << "processing saturation by pressure data for 401 psia";
        cout << " to 600 psia\n" << flush;
    }
    if (index == 600) {
        cout << "processing saturation by pressure data for 601 psia";
        cout << " to 800 psia\n" << flush;
    }
    if (index == 800) {
        cout << "processing saturation by pressure data for 801 psia";
        cout << " to 1000 psia\n" << flush;
    }
    if (index == 1000) {
        cout << "processing saturation by pressure data for 1001 psia";
        cout << " to 1200 psia\n" << flush;
    }
    if (index == 1200) {
```

```
            cout << "processing saturation by pressure data for 1201 psia";
            cout << " to 1400 psia\n" << flush;
        }
        if (index == 1400) {
            cout << "processing saturation by pressure data for 1401 psia";
            cout << " to 1500 psia\n" << flush;
        }
    }

    cout << '\n' << flush;

    // release memory used to hold input file

    delete[] line_ptr;

    return (0); // function executed successfully

} // end load_thermo

//*************************
// free_thermo function
//*************************

void    free_thermo(void)
{
    // release memory used for storage of thermodynamic properties

    delete[] sub15_ptr;
    delete[]sub500_ptr;
    delete[] sub1000_ptr;
    delete[] satbytemp_ptr;
    delete[] satbypress_ptr;

} // free_thermo

//*************************
// tsat function
//*************************

double  tsat(double pressure_in)
{
    // function which returns saturation temperature,
    // given an absolute pressure

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_pressure;
    double  output_value;

    lo_index = int(pressure_in) - 1;    // index 0 corresponds to 1 psia
    hi_index = lo_index + 1;
    base_output_value = (satbypress_ptr + lo_index)->temperature;
    delta_output_value =  (satbypress_ptr + hi_index)->temperature - base_output_value;
    delta_pressure = pressure_in - hi_index;
    output_value = base_output_value + (delta_pressure * delta_output_value);
    if (pressure_in == 1500) output_value = (satbypress_ptr + lo_index)->temperature;
    return (output_value);

} // end tsat

//*************************
// hfsat_by_p function
//*************************

double  hfsat_by_p(double pressure_in)
{
```

```
    // function which returns fluid enthalpy at saturation,
    // given an absolute pressure

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_pressure;
    double  output_value;

    lo_index = int(pressure_in) - 1;    // index 0 corresponds to 1 psia
    hi_index = lo_index + 1;
    base_output_value = (satbypress_ptr + lo_index)->hf;
    delta_output_value =  (satbypress_ptr + hi_index)->hf - base_output_value;
    delta_pressure = pressure_in - hi_index;
    output_value = base_output_value + (delta_pressure * delta_output_value);
    if (pressure_in == 1500) output_value = (satbypress_ptr + lo_index)->hf;
    return (output_value);

} // end hfsat_by_p

//*************************
// hgsat_by_p function
//*************************

double  hgsat_by_p(double pressure_in)
{
    // function which returns vapor enthalpy at saturation,
    // given an absolute pressure

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_pressure;
    double  output_value;

    lo_index = int(pressure_in) - 1;    // index 0 corresponds to 1 psia
    hi_index = lo_index + 1;
    base_output_value = (satbypress_ptr + lo_index)->hg;
    delta_output_value =  (satbypress_ptr + hi_index)->hg - base_output_value;
    delta_pressure = pressure_in - hi_index;
    output_value = base_output_value + (delta_pressure * delta_output_value);
    if (pressure_in == 1500) output_value = (satbypress_ptr + lo_index)->hg;
    return (output_value);

} // end hgsat_by_p

//*************************
// sfsat_by_p function
//*************************

double  sfsat_by_p(double pressure_in)
{
    // function which returns fluid entropy at saturation,
    // given an absolute pressure

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_pressure;
    double  output_value;

    lo_index = int(pressure_in) - 1;    // index 0 corresponds to 1 psia
    hi_index = lo_index + 1;
    base_output_value = (satbypress_ptr + lo_index)->sf;
    delta_output_value =  (satbypress_ptr + hi_index)->sf - base_output_value;
```

```
    delta_pressure = pressure_in - hi_index;
    output_value = base_output_value + (delta_pressure * delta_output_value);
    if (pressure_in == 1500) output_value = (satbypress_ptr + lo_index)->sf;
    return (output_value);

} // end sfsat_by_p

//*************************
// sgsat_by_p function
//*************************

double  sgsat_by_p(double pressure_in)
{
    // function which returns vapor entropy at saturation,
    // given an absolute pressure

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_pressure;
    double  output_value;

    lo_index = int(pressure_in) - 1;    // index 0 corresponds to 1 psia
    hi_index = lo_index + 1;
    base_output_value = (satbypress_ptr + lo_index)->sg;
    delta_output_value =  (satbypress_ptr + hi_index)->sg - base_output_value;
    delta_pressure = pressure_in - hi_index;
    output_value = base_output_value + (delta_pressure * delta_output_value);
    if (pressure_in == 1500) output_value = (satbypress_ptr + lo_index)->sg;
    return (output_value);

} // end sgsat_by_p

//*************************
// vfsat_by_p function
//*************************

double  vfsat_by_p(double pressure_in)
{
    // function which returns fluid specific volume at saturation,
    // given an absolute pressure

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_pressure;
    double  output_value;

    lo_index = int(pressure_in) - 1;    // index 0 corresponds to 1 psia
    hi_index = lo_index + 1;
    base_output_value = (satbypress_ptr + lo_index)->vf;
    delta_output_value =  (satbypress_ptr + hi_index)->vf - base_output_value;
    delta_pressure = pressure_in - hi_index;
    output_value = base_output_value + (delta_pressure * delta_output_value);
    if (pressure_in == 1500) output_value = (satbypress_ptr + lo_index)->vf;
    return (output_value);

} // end vfsat_by_p

//*************************
// vgsat_by_p function
//*************************

double  vgsat_by_p(double pressure_in)
{
    // function which returns vapor specific volume at saturation,
```

```
    // given an absolute pressure

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_pressure;
    double  output_value;

    lo_index = int(pressure_in) - 1;    // index 0 corresponds to 1 psia
    hi_index = lo_index + 1;
    base_output_value = (satbypress_ptr + lo_index)->vg;
    delta_output_value =  (satbypress_ptr + hi_index)->vg - base_output_value;
    delta_pressure = pressure_in - hi_index;
    output_value = base_output_value + (delta_pressure * delta_output_value);
    if (pressure_in == 1500) output_value = (satbypress_ptr + lo_index)->vg;
    return (output_value);

} // end vgsat_by_p

//*************************
// cpfsat_by_p function
//*************************

double  cpfsat_by_p(double pressure_in)
{
    // function which returns fluid specific heat at saturation,
    // given an absolute pressure

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_pressure;
    double  output_value;

    lo_index = int(pressure_in) - 1;    // index 0 corresponds to 1 psia
    hi_index = lo_index + 1;
    base_output_value = (satbypress_ptr + lo_index)->cpf;
    delta_output_value =  (satbypress_ptr + hi_index)->cpf - base_output_value;
    delta_pressure = pressure_in - hi_index;
    output_value = base_output_value + (delta_pressure * delta_output_value);
    if (pressure_in == 1500) output_value = (satbypress_ptr + lo_index)->cpf;
    return (output_value);

} // end cpfsat_by_p

//*************************
// cpgsat_by_p function
//*************************

double  cpgsat_by_p(double pressure_in)
{
    // function which returns vapor specific heat at saturation,
    // given an absolute pressure

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_pressure;
    double  output_value;

    lo_index = int(pressure_in) - 1;    // index 0 corresponds to 1 psia
    hi_index = lo_index + 1;
    base_output_value = (satbypress_ptr + lo_index)->cpg;
    delta_output_value =  (satbypress_ptr + hi_index)->cpg - base_output_value;
    delta_pressure = pressure_in - hi_index;
```

```
    output_value = base_output_value + (delta_pressure * delta_output_value);
    if (pressure_in == 1500) output_value = (satbypress_ptr + lo_index)->cpg;
    return (output_value);

} // end cpgsat_by_p

//*************************
// psat function
//*************************

double  psat(double temperature_in)
{
    // function which returns saturation pressure, given a temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;    // index 0 corresponds to 32 degf
    hi_index = lo_index + 1;
    base_output_value = (satbytemp_ptr + lo_index)->pressure;
    delta_output_value =  (satbytemp_ptr + hi_index)->pressure - base_output_value;
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    if (temperature_in == 600) output_value = (satbytemp_ptr + lo_index)->pressure;
    return (output_value);

} // end psat

//*************************
// hfsat_by_t function
//*************************

double  hfsat_by_t(double temperature_in)
{
    // function which returns fluid enthalpy, given a temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;    // index 0 corresponds to 32 degf
    hi_index = lo_index + 1;
    base_output_value = (satbytemp_ptr + lo_index)->hf;
    delta_output_value =  (satbytemp_ptr + hi_index)->hf - base_output_value;
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    if (temperature_in == 600) output_value = (satbytemp_ptr + lo_index)->hf;
    return (output_value);

} // end hfsat_by_t

//*************************
// hgsat_by_t function
//*************************

double  hgsat_by_t(double temperature_in)
{
    // function which returns vapor enthalpy, given a temperature

    int     lo_index;
    int     hi_index;
```

```
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;    // index 0 corresponds to 32 degf
    hi_index = lo_index + 1;
    base_output_value = (satbytemp_ptr + lo_index)->hg;
    delta_output_value =  (satbytemp_ptr + hi_index)->hg - base_output_value;
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    if (temperature_in == 600) output_value = (satbytemp_ptr + lo_index)->hg;
    return (output_value);

} // end hgsat_by_t

//*************************
// sfsat_by_t function
//*************************

double  sfsat_by_t(double temperature_in)
{
    // function which returns fluid entropy, given a temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;    // index 0 corresponds to 32 degf
    hi_index = lo_index + 1;
    base_output_value = (satbytemp_ptr + lo_index)->sf;
    delta_output_value =  (satbytemp_ptr + hi_index)->sf - base_output_value;
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    if (temperature_in == 600) output_value = (satbytemp_ptr + lo_index)->sf;
    return (output_value);

} // end sfsat_by_t

//*************************
// sgsat_by_t function
//*************************

double  sgsat_by_t(double temperature_in)
{
    // function which returns vapor entropy, given a temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;    // index 0 corresponds to 32 degf
    hi_index = lo_index + 1;
    base_output_value = (satbytemp_ptr + lo_index)->sg;
    delta_output_value =  (satbytemp_ptr + hi_index)->sg - base_output_value;
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    if (temperature_in == 600) output_value = (satbytemp_ptr + lo_index)->sg;
    return (output_value);

} // end sgsat_by_t
```

```
//*************************
// vfsat_by_t function
//*************************

double  vfsat_by_t(double temperature_in)
{
    // function which returns fluid specific volume,
    // given a temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;    // index 0 corresponds to 32 degf
    hi_index = lo_index + 1;
    base_output_value = (satbytemp_ptr + lo_index)->vf;
    delta_output_value =  (satbytemp_ptr + hi_index)->vf - base_output_value;
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    if (temperature_in == 600) output_value = (satbytemp_ptr + lo_index)->vf;
    return (output_value);

} // end vfsat_by_t

//*************************
// vgsat_by_t function
//*************************

double  vgsat_by_t(double temperature_in)
{
    // function which returns vapor specific volume,
    // given a temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;    // index 0 corresponds to 32 degf
    hi_index = lo_index + 1;
    base_output_value = (satbytemp_ptr + lo_index)->vg;
    delta_output_value =  (satbytemp_ptr + hi_index)->vg - base_output_value;
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    if (temperature_in == 600) output_value = (satbytemp_ptr + lo_index)->vg;
    return (output_value);

} // end vgsat_by_t

//*************************
// cpfsat_by_t function
//*************************

double  cpfsat_by_t(double temperature_in)
{
    // function which returns fluid specific heat,
    // given a temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
```

```
    double  output_value;

    lo_index = int(temperature_in) - 32;     // index 0 corresponds to 32 degf
    hi_index = lo_index + 1;
    base_output_value = (satbytemp_ptr + lo_index)->cpf;
    delta_output_value =  (satbytemp_ptr + hi_index)->cpf - base_output_value;
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    if (temperature_in == 600) output_value = (satbytemp_ptr + lo_index)->cpf;
    return (output_value);

} // end cpfsat_by_t

//*************************
// cpgsat_by_t function
//*************************

double  cpgsat_by_t(double temperature_in)
{
    // function which returns vapor specific heat,
    // given a temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;     // index 0 corresponds to 32 degf
    hi_index = lo_index + 1;
    base_output_value = (satbytemp_ptr + lo_index)->cpg;
    delta_output_value =  (satbytemp_ptr + hi_index)->cpg - base_output_value;
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    if (temperature_in == 600) output_value = (satbytemp_ptr + lo_index)->cpg;
    return (output_value);

} // end cpgsat_by_t

//*************************
// hpt function
//*************************

double  hpt(double pressure_in, double temperature_in)
{
    // function which returns subcooled/superheated enthalpy,
    // given pressure and temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;
    hi_index = lo_index + 1;
    if (pressure_in == 15) {             // index 0 corresponds to 32 degf
        base_output_value = (sub15_ptr + lo_index)->h;
        delta_output_value =  (sub15_ptr + hi_index)->h - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub15_ptr + lo_index)->h;
            return (output_value);
        }
    }
    if (pressure_in == 500) {            // index 0 corresponds to 32 degf
        base_output_value = (sub500_ptr + lo_index)->h;
```

```
        delta_output_value =  (sub500_ptr + hi_index)->h - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub500_ptr + lo_index)->h;
            return (output_value);
        }
    }
    if (pressure_in == 1000) {          // index 0 corresponds to 32 degf
        base_output_value = (sub1000_ptr + lo_index)->h;
        delta_output_value =  (sub1000_ptr + lo_index)->h - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub1000_ptr + lo_index)->h;
            return (output_value);
        }
    }
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    return (output_value);

} // end hpt

//*************************
// spt function
//*************************

double  spt(double pressure_in, double temperature_in)
{
    // function which returns subcooled/superheated entropy,
    // given pressure and temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;
    hi_index = lo_index + 1;
    if (pressure_in == 15) {            // index 0 corresponds to 32 degf
        base_output_value = (sub15_ptr + lo_index)->s;
        delta_output_value =  (sub15_ptr + hi_index)->s - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub15_ptr + lo_index)->s;
            return (output_value);
        }
    }
    if (pressure_in == 500) {           // index 0 corresponds to 32 degf
        base_output_value = (sub500_ptr + lo_index)->s;
        delta_output_value =  (sub500_ptr + hi_index)->s - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub500_ptr + lo_index)->s;
            return (output_value);
        }
    }
    if (pressure_in == 1000) {          // index 0 corresponds to 32 degf
        base_output_value = (sub1000_ptr + lo_index)->s;
        delta_output_value =  (sub1000_ptr + lo_index)->s - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub1000_ptr + lo_index)->s;
            return (output_value);
        }
    }
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    return (output_value);

} // end spt
```

```
//*************************
// vpt function
//*************************

double  vpt(double pressure_in, double temperature_in)
{
    // function which returns subcooled/superheated specific volume,
    // given pressure and temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;
    hi_index = lo_index + 1;
    if (pressure_in == 15) {            // index 0 corresponds to 32 degf
        base_output_value = (sub15_ptr + lo_index)->v;
        delta_output_value =  (sub15_ptr + hi_index)->v - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub15_ptr + lo_index)->v;
            return (output_value);
        }
    }
    if (pressure_in == 500) {           // index 0 corresponds to 32 degf
        base_output_value = (sub500_ptr + lo_index)->v;
        delta_output_value =  (sub500_ptr + hi_index)->v - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub500_ptr + lo_index)->v;
            return (output_value);
        }
    }
    if (pressure_in == 1000) {          // index 0 corresponds to 32 degf
        base_output_value = (sub1000_ptr + lo_index)->v;
        delta_output_value =  (sub1000_ptr + lo_index)->v - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub1000_ptr + lo_index)->v;
            return (output_value);
        }
    }
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    return (output_value);

} // end vpt

//*************************
// cppt function
//*************************

double  cppt(double pressure_in, double temperature_in)
{
    // function which returns subcooled/superheated specific heat,
    // given pressure and temperature

    int     lo_index;
    int     hi_index;
    double  base_output_value;
    double  delta_output_value;
    double  delta_temperature;
    double  output_value;

    lo_index = int(temperature_in) - 32;
    hi_index = lo_index + 1;
    if (pressure_in == 15) {            // index 0 corresponds to 32 degf
        base_output_value = (sub15_ptr + lo_index)->cp;
```

```
        delta_output_value =  (sub15_ptr + hi_index)->cp - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub15_ptr + lo_index)->cp;
            return (output_value);
        }
    }
    if (pressure_in == 500) {            // index 0 corresponds to 32 degf
        base_output_value = (sub500_ptr + lo_index)->cp;
        delta_output_value =  (sub500_ptr + hi_index)->cp - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub500_ptr + lo_index)->cp;
            return (output_value);
        }
    }
    if (pressure_in == 1000) {           // index 0 corresponds to 32 degf
        base_output_value = (sub1000_ptr + lo_index)->cp;
        delta_output_value =  (sub1000_ptr + lo_index)->cp - base_output_value;
        if (temperature_in == 600) {
            output_value = (sub1000_ptr + lo_index)->cp;
            return (output_value);
        }
    }
    delta_temperature = temperature_in - lo_index  - 32;
    output_value = base_output_value + (delta_temperature * delta_output_value);
    return (output_value);

} // end cppt
```

```
// eop calc input.h:
//      This file contains the code to write the value of
//      inputs used for the EOP calculations to an ASCII
//      file. Values are printed for plant-specific data
//      obtained from 'plant data' and for generic parameters
//      obtained from 'generic data.'
//
//      This code was written to confrom to the syntax
//      required by Microsoft Visual C++ 6.0, using old
//      versions of the standard libraries.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// VARIABLE DECLARATION
//
//*************************

char    final_01[30];   // name of file for output of final results

//*************************
//
// FUNCTION PROTOTYPE
//
//*************************

// prototype of function to write
// calculation inputs to an ASCII file

void    write_input_values(char final_01[30]);

//*************************
// write_input_values function
//*************************

void    write_input_values(char final_01[30])
{
    int     index1, index2;

    cout << "\nwriting input values to final results file\n\n" << flush;

    // begin by writing plant-specific parameters to file

    // open file stream to write input values to the file containing
    // final results of the calculations. while the file stream will
    // be attached to the same physical file (determined by the file
    // name passed in through final_01), a new identifier for the
    // stream is used to preclude a compiler error due to
    // redefinition of a variable

    ofstream final_out_01(final_01, ios::app);

    // write input values to file, generally this is done by directing
    // a string giving the name of variable followed by the value of the
    // variable followed a newline escape sequence to the output file
    // stream. arrays require more complicated coding, which will be
    // called out as it is encountered.

    final_out_01 << "Plant-specific input values:\n\n";

    final_out_01 << "nbr_fuel_types = " << nbr_fuel_types << '\n';
    final_out_01 << "nbr_pc_members = " << nbr_pc_members << '\n';
    final_out_01 << "reactor product line: " << BWR_type << '\n';
    final_out_01 << "containment type: " << BWR_PSC_type << '\n';
    final_out_01 << "c_clad = " << c_clad << " Btu/lbm-°F\n";
```

```
final_out_01 << "c_fuel = " << c_fuel << " Btu/lbm-°F\n";
final_out_01 << "dpdw_ww = " << dpdw_ww << " psid\n";
final_out_01 << "dpsrv = " << dpsrv << " psid\n";

final_out_01 << "number of drywell spray nozzles, header A: "
    << spray_hdr[0].num_nzls << '\n';
final_out_01 << "flow required for developed spray, header A: "
    << spray_hdr[0].wnzl << " gpm\n";
final_out_01 << "number of drywell spray nozzles, header B: "
    << spray_hdr[1].num_nzls << '\n';
final_out_01 << "flow required for developed spray, header B: "
    << spray_hdr[1].wnzl << " gpm\n";

// write header, set formatting and print values of
// fafl_15 and fafl_18 arrays

final_out_01 << '\n';
final_out_01 << "fuel type     fafl_15      fafl_18\n";
final_out_01 << "                [%]          [%]\n";
final_out_01 << setiosflags(ios::left);
for (index1 = 0; index1 < nbr_fuel_types; ++index1) {
    final_out_01 << "    " << setw(11)  << index1 + 1
        << setw(12) << fafl_15[index1] << fafl_18[index1] << '\n';
}
final_out_01 << '\n';

// return formatting to normal (remember setw()
// only works on next item printed)

final_out_01 << resetiosflags(ios::left);

final_out_01 << "fb10_slc = " << fb10_slc << '\n';
final_out_01 << "hdco = " << hdco << " ft\n";
final_out_01 << "hring = " << hring << " ft\n";
final_out_01 << "hsc_tap = " << hsc_tap << " ft\n";
final_out_01 << "hsp_ref = " << hsp_ref << " ft\n";
final_out_01 << "hsrv = " << hsrv << " ft\n";
final_out_01 << "hvent_pc = " << hvent_pc << " ft\n";

// write header, set formatting and print values of lfuel array

final_out_01 << '\n';
final_out_01 << "fuel type     fuel length\n";
final_out_01 << "                [in.]\n";
final_out_01 << setiosflags(ios::left);
for (index1 = 0; index1 < nbr_fuel_types; ++index1) {
    final_out_01 << "    " << setw(14)  << index1 + 1
        << lfuel[index1] << '\n';
}
final_out_01 << '\n';

// return formatting to normal (remember setw()
// only works on next item printed)

final_out_01 << resetiosflags(ios::left);

final_out_01 << "mclad = " << mclad << " lbm\n";
final_out_01 << "mf_rpv_cld = " << mf_rpv_cld << " lbm\n";
final_out_01 << "mf_rpv_hot = " << mf_rpv_hot << " lbm\n";
final_out_01 << "mfuel = " << mfuel << " lbm\n";
final_out_01 << "mg_rpv_hot = " << mg_rpv_hot << " lbm\n";
final_out_01 << "mrpv = " << mrpv << " lbm\n";
final_out_01 << "mrpv_taf = " << mrpv_taf << " lbm\n";
final_out_01 << "nbuns = " << nbuns << '\n';
final_out_01 << "nsrv_ads = " << nsrv_ads << '\n';

// write header, set formatting and print values of pc_data array
```

```
final_out_01 << '\n';
final_out_01 << "structure     elevation     material type";
final_out_01 << "     strength basis     pressure     temperature\n";
final_out_01 << "                  [ft]                        ";
final_out_01 << "                                [psig]          [°F]\n";
final_out_01 << setiosflags(ios::left);
for (index1 = 0; index1 < nbr_pc_members; ++index1) {
    if (pc_data[index1].location == 1) final_out_01 << " "
        << setw(16) << "drywell";
    else final_out_01 << "  " << setw(15) << "torus";
    final_out_01 << setw(17) << pc_data[index1].h_n;
    final_out_01 << setw(15) << pc_data[index1].material;
    if (pc_data[index1].strength == 1) final_out_01 << " yield ";
    else final_out_01 << "tensile";
    final_out_01 << "          ";
    final_out_01 << setw(16) << pc_data[index1].pcalc_n;
    final_out_01 << pc_data[index1].tcalc_n;
    final_out_01 << '\n';
}
final_out_01 << '\n';

// return formatting to normal (remember setw()
// only works on next item printed)

final_out_01 << resetiosflags(ios::left);

final_out_01 << "pdw_maxop = " << pdw_maxop << " psig\n";
final_out_01 << "pdw_minop = " << pdw_minop << " psig\n";
final_out_01 << "pdw_scram = " << pdw_scram << " psig\n";
final_out_01 << "ppc_maxind = " << ppc_maxind << " psig\n";
final_out_01 << "ppc_vent = " << ppc_vent << " psig\n";
final_out_01 << "pq_code = " << pq_code << " kpsi\n";
final_out_01 << "pq_des = " << pq_des << " kpsi\n";
final_out_01 << "pqs_code = " << pqs_code << " kpsi\n";
final_out_01 << "pqs_des = " << pqs_des << " kpsi\n";
final_out_01 << "prpv_maxind = " << prpv_maxind << " psig\n";
final_out_01 << "prpv_tp = " << prpv_tp << " psig\n";
final_out_01 << "psc_maxop = " << psc_maxop << " psig\n";
final_out_01 << "psc_minop = " << psc_minop << " psig\n";
final_out_01 << "psp_des = " << psp_des << " psig\n";
final_out_01 << "psp_srv = " << psp_srv << " psig\n";
final_out_01 << "psrv_lift = " << psrv_lift << " psig\n";
final_out_01 << "psrv_name = " << psrv_name << " psig\n";
final_out_01 << "psup_srv = " << psup_srv << " psig\n";
final_out_01 << "ptp_code = " << ptp_code << " kpsi\n";
final_out_01 << "ptp_des = " << ptp_des << " kpsi\n";
final_out_01 << "ptps_code = " << ptps_code << " kpsi\n";
final_out_01 << "ptps_des = " << ptps_des << " kpsi\n";
final_out_01 << "qrx_rated = " << qrx_rated << " MWt\n";
final_out_01 << "slc_tk_dia = " << slc_tk_dia << " ft\n";
final_out_01 << "slc_tk_lvl_in_span = " << slc_tk_lvl_in_span << " in.\n";
final_out_01 << "slc_tk_lvl_out_span = " << slc_tk_lvl_out_span << " %\n";
final_out_01 << "srv type: " << srv_type << '\n';

// write header, set formatting and print values of t_to_1500 array

final_out_01 << '\n';
for (index1 = 0; index1 < nbr_fuel_types; ++index1) {
    final_out_01 << "fuel type: " << index1 + 1 << '\n';
    final_out_01 << "  t_sd       t_cu_15\n";
    final_out_01 << " [min.]       [min.]\n";
    for (index2 = 0; index2 < 9; ++index2) {
        final_out_01 << setw(6) << t_to_1500[index1].t_sd[index2];
        final_out_01 << setw(12) << t_to_1500[index1].t_cu_15[index2] << '\n';
    }
    final_out_01 << '\n';
}
```

```
final_out_01 << "tcst = " << tcst << "°F\n";
final_out_01 << "tdw_maxop = " << tdw_maxop << "°F\n";
final_out_01 << "tdw_maxind = " << tdw_maxind << "°F\n";
final_out_01 << "tdw_minop = " << tdw_minop << "°F\n";
final_out_01 << "tsc_max = " << tsc_max << "°F\n";
final_out_01 << "tsc_maxop = " << tsc_maxop << "°F\n";
final_out_01 << "tslc = " << tslc << "°F\n";
final_out_01 << "tsp_flood = " << tsp_flood << "°F\n";
final_out_01 << "tsp_minop = " << tsp_minop << "°F\n";
final_out_01 << "tsp_scram = " << tsp_scram << "°F\n";
final_out_01 << "vdw = " << vdw << " ft3\n";
final_out_01 << "vrpv_taf = " << vrpv_taf << " ft3\n";
final_out_01 << "vsc_dco = " << vsc_dco << " ft3\n";
final_out_01 << "vsc_lco = " << vsc_lco << " ft3\n";

// write header, set formatting and print values of wg_1500 array

final_out_01 << '\n';
final_out_01 << "fuel type      wg_1500\n";
final_out_01 << "               [lbm/hr]\n";
final_out_01 << setiosflags(ios::left);
for (index1 = 0; index1 < nbr_fuel_types; ++index1) {
    final_out_01 << "    " << setw(11)  << index1 + 1
        << wg_1500[index1] << '\n';
}
final_out_01 << '\n';

// write header, set formatting and print values of vtorus array

final_out_01 << "wlsp            vsp            vsc_air\n";
final_out_01 << "[ft]            [ft3]           [ft3]\n";
for (index1 = 0; index1 < 11; ++index1) {
    final_out_01 << setw(14) << vtorus[index1].wlsp << setw(14)
        << vtorus[index1].vsp
        << vtorus[index1].vsc_air << '\n';
}
final_out_01 << '\n';

// write header, set formatting and print values of srvdl array

final_out_01 << "spwl            wll\n";
final_out_01 << "[ft]            [ft]\n";
for (index1 = 0; index1 < 5; ++index1) {
    final_out_01 << setw(14) << srvdl[index1].spwl << setw(14)
        << srvdl[index1].wll << '\n';
}
final_out_01 << '\n';

// return formatting to normal (remember setw()
// only works on next item printed)

final_out_01 << resetiosflags(ios::left);

// write header, set formatting and print values of lpci_flow array

final_out_01 << "LPCI flow vs. RPV pressure:\n";
final_out_01 << "flow            rpv_press\n";
final_out_01 << "[gpm]             [psig]\n";
for (index1 = 0; index1 < 10; ++index1) {
    final_out_01 << setw(4) << lpci_flow[index1].flow << setw(16)
        << lpci_flow[index1].rpv_press << '\n';
}
final_out_01 << '\n';

// write header, set formatting and print values of cs_flow array

final_out_01 << "CS flow vs. RPV pressure:\n";
final_out_01 << "flow            rpv_press\n";
```

```
final_out_01 << "[gpm]            [psig]\n";
for (index1 = 0; index1 < 10; ++index1) {
    final_out_01 << setw(4) << cs_flow[index1].flow << setw(16)
        << cs_flow[index1].rpv_press << '\n';
}
final_out_01 << '\n';

final_out_01 << "wlrpv_baf = " << wlrpv_baf << " in.\n";
final_out_01 << "wlsp_maxlco = " << wlsp_maxlco << " ft\n";
final_out_01 << "wlsp_minlco = " << wlsp_minlco << " ft\n";
final_out_01 << "wlsp_srv = " << wlsp_srv << " ft\n";
final_out_01 << "wlsp_tp = " << wlsp_tp << " ft\n";
final_out_01 << "wslc = " << wslc << " gpm\n";
final_out_01 << "wsrv_name = " << wsrv_name << " lbm/hr\n";
final_out_01 << "xb_cld_nat = " << xb_cld_nat << " ppm\n";
final_out_01 << "xb_hot_nat = " << xb_hot_nat << " ppm\n";
final_out_01 << "xb_slc = " << xb_slc << " ppm\n";
final_out_01 << '\n';

// continue by writing generic parameters to file

final_out_01 << "Generic input values:\n\n";

final_out_01 << "c_steel = " << c_steel << " Btu/lbm-°F\n";
final_out_01 << "dptp_dprpv = " << dptp_dprpv << " %/psig\n";
final_out_01 << "dptp_dwll = " << dptp_dwll << " %/ft\n";
final_out_01 << "fb10_nat = " << fb10_nat << '\n';
final_out_01 << "fb11_nat = " << fb11_nat << '\n';
final_out_01 << "fqdh_10 = " << fqdh_10 << '\n';
final_out_01 << "fqdh_100 = " << fqdh_100 << '\n';
final_out_01 << "fqdh_2 = " << fqdh_2 << '\n';
final_out_01 << "k1 = " << k1 << " Btu/hr-MWt\n";
final_out_01 << "k2 = " << k2 << " Btu/min.-MWt\n";
final_out_01 << "k3 = " << k3 << " gal/ft3\n";
final_out_01 << "mb10 = " << mb10 << '\n';
final_out_01 << "mb11 = " << mb11 << '\n';
final_out_01 << '\n';

// write header, set formatting and print values of pc_mat array

for (index1 = 0; index1 < 5; ++index1) {
    final_out_01 << "material type: " << index1 + 1 << '\n';
    final_out_01 << "temp            fyield          ftens\n";
    final_out_01 << "[°F]\n";
    for (index2 = 0; index2 < 4; ++index2) {
        final_out_01 << setw(3) << pc_mat[index1].temp[index2]
            << setw(18) << pc_mat[index1].fyield[index2]
            << setw(15) << pc_mat[index1].ftens[index2] << '\n';
    }
    final_out_01 << '\n';
}

// write header, set formatting and print values of sd_pwr array

final_out_01 << " time      power\n";
final_out_01 << "[min.]      [%]\n";
for (index1 = 0; index1 < 25; ++index1) {
    final_out_01 << setw(5) << sd_pwr[index1].time;
    final_out_01 << setw(10) << sd_pwr[index1].power << '\n';
}
final_out_01 << '\n';

final_out_01 << "srv_open = " << srv_open << " psid\n";
final_out_01 << "tlim_dw = " << tlim_dw << "°F\n";
final_out_01 << "tlim_sp = " << tlim_sp << "°F\n" << '\n';

// close the output file to assure the file buffer is flushed
```

```
    final_out_01.close();

}   // end write_input_values()
```

```
// shared_data.h:
//      This file contains declarations for those variables used to pass data
//      between code modules used in the EOP support calculations.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

double  pcpl_to_calcs[11];  // PCPL-A values required for calculation of the HCTL and PSP
double  mpspcwl;            // Maximum Pressure Suppression Primary Containment
                            // Water Level Limit [ft]
double  mnsred;             // Minimum Number of SRVs Required for Emergency Depressurization
double  mrfp;               // Minimum RPV Flooding Pressure [psig]
int     wrt_intr;           // flag indicating whether or not intermediate
                            // results are to be written to file
```

```
// shared_functions.h:
//      This file contains declarations for those functions used by
//      multiple modules of code used in the EOP support calculations.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// FUNCTION PROTOTYPES
//
//*************************

// function to perform interpolation on floats

double  interpolate(double lo_index, double hi_index, double mid_index,
            double lo_value, double hi_value);

// function to determine minimum of two floating point values

double  min_floats(double value_one, double value_two);

//  function to determine the maximum of two integers

int     max_int(int value_one, int value_two);

//  function to determine the maximum of two floats

double  max_floats(double value_one, double value_two);

//*************************
// interpolate function
//*************************

double  interpolate(double lo_index, double hi_index, double mid_index,
            double lo_value, double hi_value)
{
    double  mid_fraction;       // value to interpolate on as a fraction of input range
    double  value_difference;   // output range
    double  output_value;

    mid_fraction = (mid_index - lo_index) / (hi_index - lo_index);
    value_difference = hi_value - lo_value;
    output_value = lo_value + (mid_fraction * value_difference);
    return (output_value);

} // end interpolate

//*************************
// min_floats function
//*************************

double  min_floats(double value_one, double value_two)
{
    if (value_one < value_two) return(value_one);
    else return(value_two);
}

/********************
* max_int function *
********************/

int     max_int(int value_one, int value_two)
{
```

```
    if (value_one > value_two) return(value_one);
    else return(value_two);

} // end max_int

//*************************
// max_floats function
//*************************

double  max_floats(double value_one, double value_two)
{
    if (value_one > value_two) return(value_one);
    else return(value_two);

} // end max_floats
```

```
// mpspcwl.h:
//      this file contains the instructions required to compute the MPSPCWL, the
//      Maximum Pressure Suppression Primary Containment
//      Water Level.
//
//      The methods used are established in BWROG, Emergency Procedure
//      Guidelines / Severe Accident Guidelines (EPGs/SAGs) Appendix C,
//      Rev. 1, dated July, 1997.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// VARIABLE DECLARATION
//
//*************************

char    final_02[30];   // name of file for output

//*************************
//
// FUNCTION PROTOTYPE
//
//*************************

// function to calculate Maximum Pressure Suppression
// Primary Containment Water Level Limit

void    calc_mpspcwl(char final_02[30]);

//*************************
// calc_mpspcwl function
//*************************

void    calc_mpspcwl(char final_02[30])
{
    // while the file stream will be attached to the same physical
    // file (determined by the file name passed in through final_02),
    // a new identifier for the stream is used to preclude a compiler
    // error due to redefinition of a variable

    ofstream final_out_02(final_02, ios::app);

    // determine upper limit on pool water level
    // for Mark I containments, the Maximum Pressure Suppression Primary
    // Containment Water Level is the elevation of the main vent header,
    // referenced to the torus level instrument zero

    cout << "calculating mpspcwl\n\n" << flush;

    mpspcwl = hring - hsp_ref;

    // write result to file

    final_out_02 << "Maximum Pressure Suppression Primary Containment Water Level:\n\n";
    final_out_02 << "mpspcwl = " << mpspcwl << " ft\n" << '\n';

    // close output file, to assure file buffer is flushed

    final_out_02.close();

} // end calc_mpspcwl
```

```
// pcpl.h:
//      this file contains the instructions required to compute the following:
//      -   Primary Containment Pressure Limit - A
//      -   Primary Containment Pressure Limit - C
//
//      The methods used are established in BWROG, Emergency Procedure
//      Guidelines / Severe Accident Guidelines (EPGs/SAGs) Appendix C,
//      Rev. 1, dated July, 1997.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

double  dppc_dwlpc;         // change in containment pressure with water level [psig/ft]
double  hf_tap;             // height of water column above torus pressure instrument [psig]
double  ppc_calc[10][120];  // array to store limiting airspace pressures to preclude failure
                            // of components at calculated temperatures [psig]
double  ppc_lim[10][120];   // array to store limiting airspace pressures to preclude failure
                            // of components at limiting temperatures [psig]
double  ppc_srv[120];       // maximum containment pressure for SRV operation [psig]
double  pcpl_a[120];        // maximum containment pressure at which SRVs, containment
                            // vent valves and structure will remain functional [psig]
double  pcpl_c[120];        // maximum containment pressure at which containment
                            // vent valves and structure will remain functional [psig]
double  pcpl_a_eop[120];    // pcpl_a for eop use
double  pcpl_c_eop[120];    // pcpl_c for eop use
double  tpc;                // primary containment temperature [°F]
double  vf_flood;           // specific volume of subcooled water in torus at
                            // Tsp_flood and 15 psia
double  wlpc;               // primary containment water level [ft]
double  fyield_lim[10];     // fractional strength of material at limiting temperature
int     component_no;       // index into arrays of component information
char    interim_03[30];     // file name for intermediate result storage
char    final_03[30];       // file name for final result storage

//*************************
//
// FUNCTION PROTOTYPES
//
//*************************

// function to sequence other modules used in PCPL calculation

void    sequence_pcpl(char interim_03[30], char final_03[30]);

// function to calculate material strengths (at 'calculation' and
// limiting temperatures) for primary containment components

void    get_component_strength(void);

// function to calculate rate of change of pressure with primary containment
// water level of submerged portions of the containment

void    calc_dppc_dwlpc(void);

// function to calculate airspace pressure
// at which SRVs can be opened

void    pressure_for_srvs(void);
```

```
// function to calculate airspace pressure for which load carrying
// capability of a given component is not compromised

void    pressure_for_component(int component_no);

// function to determine pcpl_a and pcpl_c

void    calculate_pcpl(void);

// function to generate input for HCTL and PSP calculation

void    pcpl_for_calcs(void);

// function to write results to a file

void    do_pcpl_output(char interim_03[30], char final_03[30]);

//*************************
// sequence_pcpl function
//*************************

void    sequence_pcpl(char interim_03[30], char final_03[30])
{
    int     index1;

    // determine strength of materials analyzed for limiting pressure

    get_component_strength();

    // get affect of hydrostatic head on pressure

    calc_dppc_dwlpc();

    // determine pressure limit for SRV operability

    pressure_for_srvs();

    // determine pressure limit for containment structural members

    for (index1 = 0; index1 < nbr_pc_members; ++index1) {
        pressure_for_component(index1);
    }

    // calculate pcpl_a and pcpl_c

    calculate_pcpl();

    // assign pcpl_a values required by other modules to pcpl_to_calcs

    pcpl_for_calcs();

    do_pcpl_output(interim_03, final_03);

} // end sequence_pcpl

//*************************
// get_component_strength function
//*************************

void    get_component_strength(void)
{
    double  lo_temp;
    double  hi_temp;
    double  t_calc;
    double  t_lim;          // drywell limit is 545, torus 350
    double  lo_strength;
    double  hi_strength;
```

```
    double  fy_at_calc;
    double  fy_at_lim;
    int     index1, index2, index3, index4;

    cout << "determining component material properties" << '\n' << flush;

    // iterate on defined primary containment components

    for (index1 = 0; index1 < nbr_pc_members; ++index1) {

        // get temperature used for calculation

        t_calc = pc_data[index1].tcalc_n;

        // get temperature limit to use

        if (pc_data[index1].location == 1) t_lim = tlim_dw;
        else t_lim = tlim_sp;

        // get material strength to use

        index2 = int(pc_data[index1].strength);

        // get material to use

        index3 = int(pc_data[index1].material);

        index4 = 0;
        while (pc_mat[index3].temp[index4] < t_lim) ++index4;
        hi_temp = pc_mat[index3].temp[index4];
        --index4;
        lo_temp = pc_mat[index3].temp[index4];

        if (index2 == 1) {
            lo_strength = pc_mat[index3].fyield[index4];
            ++index4;
            hi_strength = pc_mat[index3].fyield[index4];
        }
        else {
            lo_strength = pc_mat[index3].ftens[index4];
            ++index4;
            hi_strength = pc_mat[index3].ftens[index4];
        }

        fy_at_calc = interpolate(lo_temp, hi_temp, t_calc,
                                 lo_strength, hi_strength);
        fy_at_lim = interpolate(lo_temp, hi_temp, t_lim,
                                 lo_strength, hi_strength);
        fyield_lim[index1] = fy_at_lim / fy_at_calc;
    }

} // end get_component_strength

//*************************
// calc_dppc_dwlpc function
//*************************

void    calc_dppc_dwlpc(void)
{
    // Contrary to Appendix C procedure, vf_flood will be obtained
    // from subcooled water properties at Tsp_flood and nominal
    // atmospheric pressure (15 psia), since it is unlikely that cold,
    // saturated water will be pumped into the containment.

    vf_flood = vpt(15, tsp_flood);
    dppc_dwlpc = 1 / (144 * vf_flood);

} // end calc_dppc_dwlpc
```

```
//*************************
// pressure_for_srvs function
//*************************

void    pressure_for_srvs(void)
{
    int     index1 = 0;

    cout << "determining limit for srvs" << '\n' << flush;

    wlpc = 0;

    // when containment water level is below the elevation of
    // the srv solenoid, it is not necessary to compensate for
    // the hydrostatic head of water above the solenoid

    while (wlpc <= hsrv) {
        ppc_srv[index1] = psup_srv - dpsrv;
        ++wlpc;
        ++index1;
    }

    // when containment water level is above the elevation of
    // the srv solenoid, hydrostatic head of water above the solenoid
    // is accounted for (ppc_srv decreases linearly)

    while (wlpc <= hvent_pc) {
        ppc_srv[index1] = (dppc_dwlpc * (hsrv - wlpc)) + psup_srv - dpsrv;
        ++wlpc;
        ++index1;
    }

    // by definition, ppc_srv is zero when containment water
    // level is above the elevation of the containment vent

    while (index1 < 120) {
        ppc_srv[index1] = 0;
        ++index1;
    }

} // end pressure_for_srvs

//*************************
// pressure_for_component function
//*************************

void    pressure_for_component(int component_no)
{
    int     index1 = 0;
    double  h_n;

    cout << "determining limits for structure" << '\n' << flush;

    h_n = pc_data[component_no].h_n;
    wlpc = 0;

    // when containment water level is below the elevation of
    // the component, it is not necessary to compensate for
    // the hydrostatic head of water above the component

    while (wlpc <= h_n) {
        ppc_calc[component_no][index1] = pc_data[component_no].pcalc_n;
        ppc_lim[component_no][index1] =
            pc_data[component_no].pcalc_n*fyield_lim[component_no];
        ++wlpc;
        ++index1;
    }
```

```
    // when containment water level is above the elevation of
    // the component, hydrostatic head of water above the component
    // is accounted for (ppc_calc and ppc_lim decrease linearly)

    while (wlpc < hvent_pc) {
        ppc_calc[component_no][index1] = (dppc_dwlpc * (h_n - wlpc))
            + pc_data[component_no].pcalc_n;
        ppc_lim[component_no][index1] = (dppc_dwlpc * (h_n - wlpc))
            + pc_data[component_no].pcalc_n*fyield_lim[component_no];
        ++wlpc;
        ++index1;
    }

    // by definition, ppc_calc and ppc_lim are zero when containment
    // water level is above the elevation of the containment vent

    while (index1 <= 120) {
        ppc_calc[component_no][index1]  = 0;
        ppc_lim[component_no][index1]   = 0;
        ++index1;
    }

} // end pressure_for_component

//*************************
// calculate_pcpl function
//*************************

void    calculate_pcpl(void)
{
    int     index1, index2;
    double  p_component[120];

    cout << "\ncalculating pcpl_a and pcpl_c\n\n" << flush;

    // set initial limit on component pressure arbitrarily high

    for (index1 = 0; index1 < 120; ++index1) {
        p_component[index1] = 1500;
    }

    for (index1 = 0; index1 < 120; ++index1) {

        // determine limiting pressure for components

        for (index2 = 0; index2 < nbr_pc_members; ++index2) {
            p_component[index1] = min_floats(p_component[index1], ppc_calc[index2][index1]);
            p_component[index1] = min_floats(p_component[index1], ppc_lim[index2][index1]);
        }
    }

    // calculate PCPL-C (so far uncorrected for hydrostatic head of water
    // above instrument tap, which adds consideration of vent valve
    // operability to component strength

    for (index1 = 0; index1 < 120; ++index1) {
        pcpl_c[index1] = min_floats(ppc_vent, p_component[index1]);
    }

    // calculate PCPL-A (so far uncorrected for hydrostatic head of water
    // above instrument tap, which adds consideration of srv operability
    // along with other considerations used to determine pcpl_c

    for (index1 = 0; index1 < 120; ++index1) {
        pcpl_a[index1] = min_floats(ppc_srv[index1], pcpl_c[index1]);
    }
```

```
    // correct pressure limits for effect of hydrostatic head on indication
    // when containment level is above the elevation of the instrument tap

    // find first index requiring correction

    wlpc = 0;
    index1 = 0;
    while (wlpc < hsc_tap) {
        ++wlpc;
        ++index1;
    }

    // apply correction, remembering that pcpl_a and pcpl_c are
    // by definition equal to zero for water level above the
    // elevation of the containment vent

    while (wlpc < hvent_pc) {
        pcpl_a[index1] = pcpl_a[index1] + (dppc_dwlpc * (wlpc - hsc_tap));
        pcpl_c[index1] = pcpl_c[index1] + (dppc_dwlpc * (wlpc - hsc_tap));
        ++wlpc;
        ++index1;
    }

    // truncate curves at maximum pressure indicated in control room, ppc_maxind

    for (index1 = 0; index1 < 120; ++index1) {
        pcpl_a_eop[index1] = min_floats(pcpl_a[index1], ppc_maxind);
        pcpl_c_eop[index1] = min_floats(pcpl_c[index1], ppc_maxind);
    }

} // end calculate_pcpl

//*************************
// pcpl_for_calcs function
//*************************

void    pcpl_for_calcs(void)
{
    double  lo_level;
    double  hi_level;
    double  lo_pressure;
    double  hi_pressure;
    double  level_to_find;

    // this function will write the value of PCPL-A at 9.58, 10, 11, 12
    // 13, 13.88, 14, 15, 16, 17, 17.88 ft to pcpl_to_calcs for use in the
    // calculation of the HCTL and PSP

    // since the index of pcpl_a is equivalent to containment water level,
    // interpolate between elements 9 and 10 to get PCPL-A at 9.58 ft

    lo_level = 9;
    hi_level = 10;
    level_to_find = 9.58;
    lo_pressure = pcpl_a[9];
    hi_pressure = pcpl_a[10];
    pcpl_to_calcs[0] = interpolate(lo_level, hi_level, level_to_find,
        lo_pressure, hi_pressure);

    // values for 10, 11, 12 and 13 ft are read in directly

    pcpl_to_calcs[1] = pcpl_a[10];
    pcpl_to_calcs[2] = pcpl_a[11];
    pcpl_to_calcs[3] = pcpl_a[12];
    pcpl_to_calcs[4] = pcpl_a[13];

    // interpolate between elements 13 and 14 to get PCPL-A at 13.88 ft
```

```
    lo_level = 13;
    hi_level = 14;
    level_to_find = 13.88;
    lo_pressure = pcpl_a[13];
    hi_pressure = pcpl_a[14];
    pcpl_to_calcs[5] = interpolate(lo_level, hi_level, level_to_find,
        lo_pressure, hi_pressure);

    // values for 14, 15, 16 and 17 are read in directly

    pcpl_to_calcs[6] = pcpl_a[14];
    pcpl_to_calcs[7] = pcpl_a[15];
    pcpl_to_calcs[8] = pcpl_a[16];
    pcpl_to_calcs[9] = pcpl_a[17];

    // interpolate between elements 17 and 18 to get PCPL-A at 17.88 ft

    lo_level = 17;
    hi_level = 18;
    level_to_find = 17.88;
    lo_pressure = pcpl_a[17];
    hi_pressure = pcpl_a[18];
    pcpl_to_calcs[10] = interpolate(lo_level, hi_level, level_to_find,
        lo_pressure, hi_pressure);

} // end pcpl_to_calcs

//*************************
// do_pcpl_output function
//*************************

void    do_pcpl_output(char interim_03[30], char final_03[30])
{
    int     index1, index2;

    // write intermediate results to file
    // while the file stream will be attached to the same physical
    // file (determined by the file name passed in through interim_03),
    // a new identifier for the stream is used to preclude a compiler
    // error due to redefinition of a variable

    ofstream inter_out_03(interim_03, ios::app);

    inter_out_03 << "PCPL calculation, intermediate results:" << '\n' << '\n';
    inter_out_03 << "vf_flood = " << vf_flood << " ft3/lbm\n";
    inter_out_03 << "dppc_dwlpc = " << dppc_dwlpc << " psig/ft\n" << '\n';
    inter_out_03 << "pressure limit for srv operation and to assure\n";
    inter_out_03 << "structural integrity at temperatures for which\n";
    inter_out_03 << "input pressure capabilities were calculated:\n\n";
    inter_out_03 << "wlpc  ppc_srv   ppc_1   ppc_2   ppc_3   ppc_4";
    inter_out_03 << "   ppc_5   ppc_6   ppc_7   ppc_8   ppc_9   ppc_10\n";
    inter_out_03 << "[ft]  [psig]   [psig]  [psig]  [psig]  [psig]";
    inter_out_03 << "  [psig]  [psig]  [psig]  [psig]  [psig]   [psig]\n";

    // set type and precision of numeric output

    inter_out_03 << setiosflags(ios::fixed) << setprecision(1);

    // write results

    for (index1 = 0; index1 < 120; ++index1) {
        inter_out_03 << setiosflags(ios::left) << setw(6) << index1
            << resetiosflags(ios::left) << setw(5) << ppc_srv[index1] << "  ";
        for (index2 = 0; index2 < nbr_pc_members; ++ index2) {
            inter_out_03 << setw(8) << ppc_calc[index2][index1];
        }
        inter_out_03 << '\n';
    }
```

```
    inter_out_03 << '\n';
    inter_out_03 << "pressure limit to assure structural integrity\n";
    inter_out_03 << "at limiting temperature for structure:\n\n";
    inter_out_03 << "wlpc   ppc_1   ppc_2   ppc_3   ppc_4";
    inter_out_03 << "   ppc_5   ppc_6   ppc_7   ppc_8   ppc_9   ppc_10\n";
    inter_out_03 << "[ft]  [psig]  [psig]  [psig]  [psig]  [psig]";
    inter_out_03 << "  [psig]  [psig]  [psig]  [psig]  [psig]   [psig]\n";
    for (index1 = 0; index1 < 120; ++index1) {
        inter_out_03 << setiosflags(ios::left) << setw(4) << index1
            << resetiosflags(ios::left);
        for (index2 = 0; index2 < nbr_pc_members; ++ index2) {
            inter_out_03 << setw(8) << ppc_lim[index2][index1];
        }
        inter_out_03 << '\n';
    }
    inter_out_03 << '\n';

    // reset type and precision of output

    resetiosflags(ios::fixed);

    // close output file to assure file buffer is flushed

    inter_out_03.close();

    // write final results to file

    // while the file stream will be attached to the same physical
    // file (determined by the file name passed in through final_03),
    // a new identifier for the stream is used to preclude a compiler
    // error due to redefinition of a variable

    ofstream final_out_03(final_03, ios::app);

    // write output

    final_out_03 << "Primary Containment Pressure Limits:\n\n";
    final_out_03 << "wlpc     pcpl_a     pcpl_c     pcpl_a_eop     pcpl_c_eop\n";
    final_out_03 << "[ft]     [psig]     [psig]       [psig]         [psig]\n";

    // set type and precision of output

    final_out_03 << setiosflags(ios::fixed) << setprecision(1);
    for (index1 = 0; index1 < 120; ++index1) {
        final_out_03 << setiosflags(ios::left) << setw(4) << index1
            << resetiosflags(ios::left) << setw(10) << pcpl_a[index1]
            << setw(11) << pcpl_c[index1] << setw(13) << pcpl_a_eop[index1]
            << setw(15) << pcpl_c_eop[index1] << '\n';
    }
    final_out_03 << '\n';

    // reset type and precision of output

    resetiosflags(ios::fixed);

    // close output file to assure file buffer is flushed

    final_out_03.close();

} // end do_pcpl_output
```

```
// stpll.h:
//      this file contains the instructions required to compute the STPLL, the
//      lesser of the Maximum Pressure Suppression Primary Containment
//      Water Level or the highest pool level at which opening an SRV will not
//      result in exceeding the code allowable stress in the SRV tail pipe, tail
//      pipe supports, quencher or quencher supports.
//
//      The methods used are established in BWROG, Emergency Procedure
//      Guidelines / Severe Accident Guidelines (EPGs/SAGs) Appendix C,
//      Rev. 1, dated July, 1997.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// STRUCTURE DECLARATION
//
//*************************

// data describing points on STPLL curve

struct  pressure_level {
    double  pressure;       // RPV pressure [psig]
    double  level;          // torus water level [ft]
};

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

double  fdes_q;     // design margin for tee-quencher [%]
double  fdes_qs;    // design margin for tee-quencher support [%]
double  fdes_tp;    // design margin for SRV tail pipe [%]
double  fdes_tps;   // design margin for SRV tail pipe support [%]
double  prpv_1;     // maximum allowable pressure at torus water level
                    // used for SRV tail pipe design calculations [psig]
pressure_level prpv_break[5];   // pressure breakpoints corresponding to
                                // breakpoints in srvdl[5] (SRV discharge
                                // lines at JAF are modelled as four segments
                                // for torus water levels up to the upper
                                // torus invert [psig]
pressure_level stpll_eop[5];    // points defining STPLL for PSTG/SAG/EOP/SAOG use
char    interim_04[30];         // file name for output of intermediate results
char    final_04[30];           // file name for output of final results

//*************************
//
// FUNCTION PROTOTYPES
//
//*************************

// function to calculate SRV tail pipe level limit

void    calc_stpll(char interim_04[30], char final_04[30]);

//*************************
// calc_stpll function
//*************************

void    calc_stpll(char interim_04[30], char final_04[30])
{
```

```
pressure_level stpll[5];    // prpv_break[] truncated at prpv_maxind and 0 psig
int     index1, index2;
int     lo_index, hi_index;
double  lo_press;
double  hi_press;
double  lo_level;
double  hi_level;

cout << "calculating stpll\n\n" << flush;

// declare a file stream for output of intermediate results.
// while the file stream will be attached to an existing physical
// file (determined by the file name passed in through interim_04),
// a new identifier for the stream is used to preclude a compiler
// error due to redefinition of a variable

ofstream inter_out_04(interim_04, ios::app);

inter_out_04 << "STPLL calculation, intermediate results:\n\n";

// initialize values of parameters used in this calculation

// determine design margin for SRV quencher

fdes_q = (100 * pq_code / pq_des) - 100;

// determine design margin for SRV quencher support

fdes_qs = (100 * pqs_code / pqs_des) - 100;

// determine design margin for SRV tail pipe

fdes_tp = (100 * ptp_code / ptp_des) - 100;

// determine design margin for SRV tail pipe support

fdes_tps = (100 * ptps_code / ptps_des) - 100;

// determine maximum allowable RPV pressure at torus
// water level used for SRV tail pipe design calculations

prpv_1 = min_floats(fdes_q, fdes_qs);
prpv_1 = min_floats(prpv_1, fdes_tp);
prpv_1 = min_floats(prpv_1, fdes_tps);
prpv_1 = (prpv_tp * prpv_1 / 100) + prpv_tp;

// write first batch of intermediate values to file

inter_out_04 << "Component design margins:\n\n";
inter_out_04 << "  component                fdes\n";
inter_out_04 << "                            [%]\n";
inter_out_04 << "   quencher                 " << fdes_q << '\n';
inter_out_04 << "   quencher support         " << fdes_qs << '\n';
inter_out_04 << "   tail pipe                " << fdes_tp << '\n';
inter_out_04 << "   tail pipe support        " << fdes_tps << '\n';
inter_out_04 << '\n';

// find the breakpoints in the graph of torus water level (STPLL) vs. RPV pressure.
// rather than using the BWROG, App. C method of calculating a maximum water
// leg length versus RPV pressure, I have solved the defining equation directly for
// RPV pressure as a function of water leg length. the resulting equation is then
// evaluated for the water leg lengths which describe the SRV tail pipes as a function
// of torus water level in the plant database
//
// successive elements of srvdl have larger values of wll and spwl stored, therefore
// successive elements of prpv_break will have smaller values of pressure for rising
// values of torus water level.
```

```
inter_out_04 << "breakpoints in STPLL vs. RPV pressure curve:\n\n";
inter_out_04 << "spwl      pressure\n";
inter_out_04 << "[ft]        [psig]\n";
inter_out_04 << setiosflags(ios::fixed);
inter_out_04 << setiosflags(ios::left);
for (index1 = 0; index1 < 5; ++index1) {
    prpv_break[index1].pressure =
        prpv_1 - (srvdl[index1].wll * (dptp_dwll / dptp_dprpv));
    prpv_break[index1].level = srvdl[index1].spwl;
    inter_out_04 << setprecision(2);
    inter_out_04 << setw(12) << prpv_break[index1].level;
    inter_out_04 << setprecision(0);
    inter_out_04    << prpv_break[index1].pressure << '\n';
}

// initialize stpll

for (index1 = 0; index1 < 5; ++index1) {
    stpll[index1].pressure = 0;
    stpll[index1].level = 0;
}

// determine the points defining the STPLL for EOP use. first, truncate the
// curve at the maximum available RPV pressure indication and 0 psig

// set lo_index to point at the first location for which pressure < prpv_maxind

lo_index = 0;
while (prpv_break[lo_index].pressure > prpv_maxind) ++lo_index;

// set hi_index to point at the first location for which pressure < 0

hi_index = 0;
while (prpv_break[hi_index].pressure > 0) ++hi_index;

// determine number of points to be
// stored for which pressure is not 0

index2 = hi_index - lo_index + 1;

// no pressures were above prpv_maxind, assign all records with
// pressure above 0 to stpll

if (lo_index == 0) {
    for (index1 = 0; index1 < 5; ++index1) {
        while (prpv_break[index1].pressure >= 0) {
            stpll[index1].pressure = prpv_break[index1].pressure;
            stpll[index1].level = prpv_break[index1].level;
        }
    }
}

// there were pressures above prpv_maxind

else {

    // interpolate to find torus water level corresponding to prpv_maxind

    index1 = lo_index - 1;
    lo_press = prpv_break[index1].pressure;
    lo_level = prpv_break[index1].level;
    hi_press = prpv_break[lo_index].pressure;
    hi_level = prpv_break[lo_index].level;
    stpll[0].pressure = prpv_maxind;
    stpll[0].level = interpolate(lo_press, hi_press, prpv_maxind,
        lo_level, hi_level);
    index1 = 1;
```

```
    // store non-zero values of pressure (and corresponding levels)

    while (index1 < index2) {
        stpll[index1].pressure = prpv_break[lo_index].pressure;
        stpll[index1].level = prpv_break[lo_index].level;
        ++index1;
        ++lo_index;
    }
}

// there were pressures below 0

if (hi_index < 5) {

    // interpolate to find the level at which pressure reaches 0

    index1 = hi_index - 1;
    lo_press = prpv_break[index1].pressure;
    lo_level = prpv_break[index1].level;
    hi_press = prpv_break[hi_index].pressure;
    hi_level = prpv_break[hi_index].level;
    stpll[index2].pressure = 0;
    stpll[index2].level = interpolate(lo_press, hi_press, 0,
        lo_level, hi_level);
}

inter_out_04 << '\n' << "STPLL:\n\n";
inter_out_04 << "spwl      pressure\n";
inter_out_04 << "[ft]        [psig]\n";
for (index1 = 0; index1 < 5; ++index1) {
    inter_out_04 << setprecision(2);
    inter_out_04 << setw(12) << stpll[index1].level;
    inter_out_04 << setprecision(0);
    inter_out_04    << stpll[index1].pressure << '\n';
}
inter_out_04 << '\n';

// return output format to normal

resetiosflags(ios::left);
resetiosflags(ios::fixed);

// close output file to assure file buffer is flushed

inter_out_04.close();

// open file stream to write final results of the calculations.
// while the file stream will be attached to an existing physical
// file (determined by the file name passed in through final_04),
// a new identifier for the stream is used to preclude a compiler
// error due to redefinition of a variable

ofstream final_out_04(final_04, ios::app);

// initialize stpll_eop

for (index1 = 0; index1 < 5; ++index1) {
    stpll_eop[index1].pressure = stpll[index1].pressure;
    stpll_eop[index1].level = stpll[index1].level;
}

// now, truncate the curve at the Maximum Pressure Suppression Primary
// Containment Water Level

// set hi_index to point at the first element
// in the array with level above mpspcwl

hi_index = 0;
```

```
    while ((stpll[hi_index].level < mpspcwl) && (hi_index < 5)) ++hi_index;

    // interpolate to find pressure corresponding to mpspcwl
    // if there are entries with level above the mpspcwl

    if (hi_index < 5) {
        lo_index = hi_index - 1;
        lo_press = stpll[lo_index].pressure;
        lo_level = stpll[lo_index].level;
        hi_press = stpll[hi_index].pressure;
        hi_level = stpll[hi_index].level;
        stpll_eop[hi_index].level = mpspcwl;
        stpll_eop[hi_index].pressure = interpolate(lo_level, hi_level,
            mpspcwl, lo_press, hi_press);
        ++hi_index;
    }

    // set the next empty element to (0, mpspcwl)

    if (hi_index < 5) {
        stpll_eop[hi_index].pressure = 0;
        stpll_eop[hi_index].level = mpspcwl;
    }

    final_out_04 << "SRV Tail Pipe Level Limit (for EOPs):\n\n";
    final_out_04 << "spwl      pressure\n";
    final_out_04 << "[ft]        [psig]\n";
    final_out_04 << setiosflags(ios::left) << setiosflags(ios::fixed);
    for (index1 = 0; index1 < 5; ++index1) {
        final_out_04 << setprecision(2);
        final_out_04 << setw(12) << stpll_eop[index1].level;
        final_out_04 << setprecision(0);
        final_out_04 << stpll_eop[index1].pressure << '\n';
    }
    final_out_04 << '\n';

    // return output format to normal

    resetiosflags(ios::left);
    resetiosflags(ios::fixed);

    // close output file to assure file buffer is flushed

    final_out_04.close();

} // end calc_stpll
```

```
// hctl.h:
//      this file contains the instructions required to compute the following:
//      -   tsp_hctl_2: the heat capacity temperature limit low pressure
//          endpoint temperature
//      -   heat capacity temperature limit
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
//      The methods used are established in BWROG, Emergency Procedure
//      Guidelines / Severe Accident Guidelines (EPGs/SAGs) Appendix C,
//      Rev. 1, dated July, 1997.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

double  pg_sc;          // partial pressure of water vapor in torus airspace [psia]
double  pnsc_sc;        // partial pressure of dry gas (N2 or air) originating in torus airspace
                        // normalized to 100°F using ideal gas law when in torus airspace [psia]
double  qrx_1;          // percentile reactor power [Btu/min.-%]
double  wlsp_lim;       // lower limit on torus water level used in calculation [ft]

// the following parameters are all functions of torus water level:

double  tsc_pcpl[11];   // temperature required for gas pressure to reach pcpl_a [°F]
double  pnsc_dw[11];    // partial pressure of dry gas (N2 or air) originating in drywell normalized
                        // to 100°F using ideal gas law when in torus airspace [psia]
double  dpsc_dtsc[11];  // rate of change of pressure with temperature of all dry gas
                        // in torus airspace, from ideal gas law [°R/psia]
double  tsp_hctl_2[11]; // Heat Capacity Temperature Limit low-pressure endpoint
                        // temperature [°F]
double  tsp_hctl_1[11]; // Heat Capacity Temperature Limit high-pressure endpoint
                        // temperature [°F]
double  tsp_hctl_2_lco; // HCTL low-pressure endpoint with torus water
                        // level at the tech spec minimum [°F]
double  tsp_hctl_1_lo;  // HCTL high-pressure endpoint with torus water
                        // level at the tech spec minimum [°F]
double  tsp_hctl_1_hi;  // HCTL high-pressure endpoint with torus water
                        // level at the tech spec maximum [°F]
char    interim_05[30]; // file name for storage of intermediate results
char    final_05[30];   // file name for storage of final results

//*************************
//
// FUNCTION PROTOTYPES
//
//*************************

// function to determine the number of state
// points to include in HCTL calculation

int     calc_n_values(double big, double little);

// function to calculate percentile decay heat as a function of time

double  calc_qdh(double time);

// function to calculate low-pressure endpoint temperature
// as a function of torus water level

void    calc_tsp_hctl_2(char interim_05[30]);
```

```
// function to calculate the heat capacity temperature limit

void    calc_hctl(char interim_05[30], char final_05[30]);

//*************************
// calc_tsp_hctl_2 function
//*************************

void    calc_tsp_hctl_2(char interim_05[30])
{
    int     index1;
    double  p_gas;      // pressure of gas at a given temperature [psia]

    // declare a file stream for output of intermediate results.
    // while the file stream will be attached to an existing physical
    // file (determined by the file name passed in through interim_05),
    // a new identifier for the stream is used to preclude a compiler
    // error due to redefinition of a variable

    ofstream inter_out_05a(interim_05, ios::app);

    cout << "calculating heat capacity temperature limit";
    cout << " low pressure endpoints\n\n" << flush;

    // initialize data required here (and in later program elements)

    // determine partial pressure of water vapor in torus airspace

    pg_sc = psat(tsc_minop);

    // determine partial pressure of dry gas in torus airspace
    // normalized to 100°F using ideal gas law

    pnsc_sc = ((psc_maxop + 14.7 - pg_sc) * 559.69) / (tsc_minop + 459.69);

    // determine lower limit on pool water level
    // for Mark I containments, this is the elevation of the downcomer
    // opening, referenced to the torus level instrument zero

    wlsp_lim = hdco - hsp_ref;

    inter_out_05a << "HCTL low-pressure endpoint intermediate results:\n\n";
    inter_out_05a << "pg_sc = " << pg_sc << " psia\n";
    inter_out_05a << "pnsc_sc = " << pnsc_sc << " psia\n";
    inter_out_05a << "wlsp_lim = " << wlsp_lim << " ft\n\n";

    for (index1 = 0; index1 < 11; ++index1) {

        // determine partial pressure of dry gas originating in drywell
        // normalized to 100°F using ideal gas law, when in torus airspace

        pnsc_dw[index1] = ((pdw_maxop + 14.7) * 559.69 * vdw)
            / ((tdw_minop + 459.69) * vtorus[index1].vsc_air);

        // detemine rate of change of pressure with temperature of all dry gas
        // in torus airspace, from ideal gas law

        dpsc_dtsc[index1] = (pnsc_sc + pnsc_dw[index1]) / 559.69;

        // determine temperature at which pcpl_a will be reached when all
        // of the gas is in the torus airspace

        tsc_pcpl[index1] = tsc_minop;
        p_gas = (dpsc_dtsc[index1] * (tsc_pcpl[index1] + 459.69)) + psat(tsc_pcpl[index1]);

        while (p_gas < (pcpl_to_calcs[index1] + 14.7)) {
            tsc_pcpl[index1] = tsc_pcpl[index1] + 0.1;
```

```
            p_gas = (dpsc_dtsc[index1] * (tsc_pcpl[index1] + 459.69)) + psat(tsc_pcpl[index1]);
        }

        // the last increment of tsc_pcpl[index1] caused the pcpl_a to be exceeded
        // therefore, take it back out

        tsc_pcpl[index1] = tsc_pcpl[index1] - 0.1;

        // determine HCTL low-pressure endpoint temperature (the lower of
        // tsc_pcpl[] and tsc_max)

        tsp_hctl_2[index1] = min_floats(tsc_pcpl[index1], tsc_max);

        // cheat and use knowledge that no data is stored in
        // vtorus for water levels between 13.88 ft and 14 ft
        // to enable use of inequality to get tsp_hctl_2_lco

        if (vtorus[index1].wlsp < 13.9) tsp_hctl_2_lco = tsp_hctl_2[index1];
    }

    inter_out_05a << "wlsp     vsc_air     pnsc_dw     dpsc_dtsc";
    inter_out_05a << "     pcpl_a     tsc_pcpl     tsp_hctl_2\n";
    inter_out_05a << "[ft]      [ft3]       [psia]     [°R/psia]";
    inter_out_05a << "     [psig]       [°F]          [°F]\n";
    inter_out_05a << setiosflags(ios::fixed) << setiosflags(ios::left);
    for (index1 = 0; index1 < 11; ++index1) {
        inter_out_05a << setprecision(2) << setw(9) << vtorus[index1].wlsp;
        inter_out_05a << setprecision(0) << setw(13) << vtorus[index1].vsc_air;
        inter_out_05a << setprecision(2) << setw(13) << pnsc_dw[index1];
        inter_out_05a << setprecision(4) << setw(13) << dpsc_dtsc[index1];
        inter_out_05a << setprecision(1) << setw(11) << pcpl_to_calcs[index1];
        inter_out_05a << setprecision(1) << setw(14) << tsc_pcpl[index1];
        inter_out_05a << setprecision(1) << tsp_hctl_2[index1];
        inter_out_05a << '\n';
    }

    inter_out_05a << '\n' << "tsp_hctl_2_lco = " << tsp_hctl_2_lco << "°F\n\n";

    // return formatting to default values

    inter_out_05a << resetiosflags(ios::left);
    inter_out_05a << resetiosflags(ios::fixed);

    // close output file to assure file buffer is flushed

    inter_out_05a.close();

} // end calc_tsp_hctl_2

//*************************
// calc_n_values function
//*************************

int     calc_n_values(double big, double little)
{
    double  midpoint;
    double  difference;
    double  quotient;
    double  remainder;
    int     modulus;
    int     answer;

    // pressure increments of 10 psi will be used for pressure above 200 psig
    // and increments of 5 psi will be used at or below 200 psig

    midpoint = 200;
    difference = big - midpoint;
    quotient = difference / 10;
```

```
    modulus = int(quotient);
    remainder = quotient - modulus;

    // make assumption that psrv_lift will be an even multiple of 5 psi
    // so that remainder more than zero (0.1 used to avoid problems
    // with representation of numbers, 0 != 0.0000x) is indicative of
    // a setpoint which is not an even multiple of 10 psi

    if (remainder > 0.1) {
        answer = modulus + 1;
    }
    else answer = modulus;

    big = 200;
    difference = big - little;
    quotient = difference / 5;

    answer = int(answer + quotient + 1);

    return (answer);

} // calc_n_values

//*************************
// calc_qdh function
//*************************

double  calc_qdh(double time)
{
    double  pwr_bfr;    // power before 'time' from array sd_pwr [%]
    double  pwr_aftr;   // power after 'time' from array sd_pwr [%]
    double  pwr_now;    // power at 'time' [%, then Btu/min.]
    double  t_bfr;      // time before 'time' from array sd_pwr [min.]
    double  t_aftr;     // time after 'time' from array sd_pwr [min.]
    int     index1;

    index1 = 0;

    // find first stored time equal to or longer than time passed in

    while (sd_pwr[index1].time <= time) {
        ++index1;
    }

    // if we get lucky, the time passed in was a stored value

    if (time == sd_pwr[index1 - 1].time) pwr_now = sd_pwr[index1 - 1].power;

    // if not, find bracketting times and interpolate to get power

    if (time < sd_pwr[index1].time) {
        t_bfr = sd_pwr[index1 - 1].time;
        pwr_bfr = sd_pwr[index1 - 1].power;
        t_aftr = sd_pwr[index1].time;
        pwr_aftr = sd_pwr[index1].power;
        pwr_now = interpolate(t_bfr, t_aftr, time, pwr_bfr, pwr_aftr);
    }

    // calculate value of percentile reactor power

    qrx_1 = qrx_rated * k2 / 100;

    // convert pwr_now from percent power to Btu/min.

    pwr_now = pwr_now * qrx_1;

    return (pwr_now);
```

```
} // calc_qdh

//*************************
// calc_hctl function
//*************************

void    calc_hctl(char interim_05[30], char final_05[30])
{
    // structure with fields for intermediate results

    struct  hctl_int {
        double  prpv;           // RPV pressure [psia]
        double  trpv;           // RPV temperature [°F]
        double  hf_rpv;         // specific enthalpy of fluid in RPV [Btu/lbm]
        double  hg_rpv;         // specific enthalpy of vapor in RPV [Btu/lbm]
        double  vf_rpv;         // specific volume of fluid in RPV [ft3/lbm]
        double  vg_rpv;         // specific volume of vapor in RPV [ft3/lbm]
        double  wsrv;           // SRV flowrate [lbm/min.]
        double  mf_rpv;         // mass of liquid in RPV [lbm]
        double  mg_rpv;         // mass of vapor in RPV [lbm]
        double  erpv;           // energy of RPV [Btu]
        double  dmg_srv;        // mass passed through SRVs [lbm]
        double  mf_sp;          // mass of liquid in torus [lbm]
        double  dt;             // time step [min.]
        double  t;              // time [min.]
        double  qdh;            // decay heat [Btu/min.]
        double  esp;            // energy in torus [Btu]
        double  tsp;            // temperature of torus [°F]
        double  z1, z2, z3, z4, z5, z6;
    };

    int     torus_calc_base;    // index to stop at when calculating torus state
                                // variables as a function of lowering RPV pressure
    int     index1, index2, to_ptr;
    double  wsrv_be;            // best-estimate SRV flow-rate [lbm/min.]
    double  psrv_ref;           // reference pressure for best-estimate SRV flowrate [psig]
    double  dwsrv_dprpv;        // specific SRV flowrate (critical flow) [lbm/min.-psia]
    double  crpv;               // RPV heat capacity [Btu/°F]
    double  vrpv;               // RPV volume [ft3]
    double  prpv_g;             // RPV pressure for a given statepoint [psig]
    double  prpv_a;             // RPV pressure for a given statepoint [psia]
    int     n_values;           // actual number of intermediate pressure points calculated

    hctl_int    *res_ptr;       // pointer to storage for intermediate results

    cout << "calculating heat capacity temperature limit\n\n" << flush;

    // get memory to store intermediate results. sufficient memory is obtained
    // to store 160 pressure values for each of 11 torus water levels. this corresponds
    // to a calculation proceeding from 1500 psig to 200 psig in 10 psig increments,
    // and from 200 psig to 50 psig in 5 psig increments

    res_ptr = new hctl_int[1760];

    // initialize storage

    for (index1 = 0; index1 < 1760; ++index1) {
        (res_ptr + index1)->prpv = 0;
        (res_ptr + index1)->trpv = 0;
        (res_ptr + index1)->hf_rpv = 0;
        (res_ptr + index1)->hg_rpv = 0;
        (res_ptr + index1)->vf_rpv = 0;
        (res_ptr + index1)->vg_rpv = 0;
        (res_ptr + index1)->wsrv = 0;
        (res_ptr + index1)->mf_rpv = 0;
        (res_ptr + index1)->mg_rpv = 0;
        (res_ptr + index1)->erpv = 0;
        (res_ptr + index1)->dmg_srv = 0;
```

```
    (res_ptr + index1)->mf_sp = 0;
    (res_ptr + index1)->dt = 0;
    (res_ptr + index1)->t = 0;
    (res_ptr + index1)->qdh = 0;
    (res_ptr + index1)->esp = 0;
    (res_ptr + index1)->tsp = 0;
    (res_ptr + index1)->z1 = 0;
    (res_ptr + index1)->z2 = 0;
    (res_ptr + index1)->z3 = 0;
    (res_ptr + index1)->z4 = 0;
    (res_ptr + index1)->z5 = 0;
    (res_ptr + index1)->z6 = 0;
}

// determine number of pressure points which are required

n_values = calc_n_values (psrv_lift, mrfp);

// determine best-estimate SRV flowrate

wsrv_be = (1.10 * wsrv_name) / 60;

// determine reference pressure for best-estimate SRV flowrate

psrv_ref = 1.03 * psrv_name;

// determine specific SRV flowrate (critical flow)

dwsrv_dprpv = wsrv_be / (psrv_ref + 14.7);

// determine RPV heat capacity

crpv = (c_steel * mrpv) + (c_clad * mclad) + (c_fuel * mfuel);

// determine RPV volume

vrpv = (vfsat_by_p(psrv_lift + 14.7) * mf_rpv_hot) + (vgsat_by_p(psrv_lift + 14.7) * mg_rpv_hot ;

// iteratively determine HCTL as a function of torus water level

for (index1 = 0; index1 < 11; ++index1) {

    // set pointer into storage array

    to_ptr = index1 * n_values;

    // set initial equilibrium RPV state

    prpv_a = psrv_lift + 14.7;
    prpv_g = psrv_lift;
    (res_ptr + to_ptr)->prpv = prpv_g;
    (res_ptr + to_ptr)->trpv = tsat(prpv_a);
    (res_ptr + to_ptr)->hf_rpv = hfsat_by_p(prpv_a);
    (res_ptr + to_ptr)->hg_rpv = hgsat_by_p(prpv_a);
    (res_ptr + to_ptr)->vf_rpv = vfsat_by_p(prpv_a);
    (res_ptr + to_ptr)->vg_rpv = vgsat_by_p(prpv_a);
    (res_ptr + to_ptr)->wsrv = mnsred * dwsrv_dprpv * prpv_a;
    (res_ptr + to_ptr)->mf_rpv = mf_rpv_hot;
    (res_ptr + to_ptr)->mg_rpv = mg_rpv_hot;
    (res_ptr + to_ptr)->erpv = ((res_ptr + to_ptr)->trpv * crpv)
        + ((res_ptr + to_ptr)->hf_rpv * (res_ptr + to_ptr)->mf_rpv)
        + ((res_ptr + to_ptr)->hg_rpv * (res_ptr + to_ptr)->mg_rpv);
    (res_ptr + to_ptr)->t = 2;
    (res_ptr + to_ptr)->qdh = calc_qdh((res_ptr + to_ptr)->t);

    // rather than using the App. C equation, initial torus water mass will be
    // set at the volume divided by the specific volume of water (subcooled)
    // at 15 psia and tsp_scram, since EOPs function to maintain torus pressure
```

```
// at or above atmospheric (nominally 14.67 psia)

(res_ptr + to_ptr)->mf_sp = vtorus[index1].vsp / vpt(15, tsp_scram);

// calculate RPV parameters and torus water mass as vessel depressurizes (along HCTL

for (index2 = 1; index2 < n_values; ++index2) {

    // set pointer into storage array

    to_ptr = (index1 * n_values) + index2;

    // calculate RPV pressure for next state point

    if (prpv_g >= 210) {
        prpv_g = prpv_g - 10;
        prpv_a = prpv_a -10;
    }
    else {
        prpv_g = prpv_g - 5;
        prpv_a = prpv_a - 5;
    }

    // determine RPV parameters and torus water
    // mass for successively lower RPV pressures

    (res_ptr + to_ptr)->prpv = prpv_g;
    (res_ptr + to_ptr)->trpv = tsat(prpv_a);
    (res_ptr + to_ptr)->hf_rpv = hfsat_by_p(prpv_a);
    (res_ptr + to_ptr)->hg_rpv = hgsat_by_p(prpv_a);
    (res_ptr + to_ptr)->vf_rpv = vfsat_by_p(prpv_a);
    (res_ptr + to_ptr)->vg_rpv = vgsat_by_p(prpv_a);
    (res_ptr + to_ptr)->wsrv = mnsred * dwsrv_dprpv * prpv_a;
    (res_ptr + to_ptr)->z1 = vrpv / (res_ptr + to_ptr)->vg_rpv;
    (res_ptr + to_ptr)->z2 = (res_ptr + to_ptr)->vf_rpv / (res_ptr + to_ptr)->vg_rpv;
    (res_ptr + to_ptr)->z3 = (res_ptr + to_ptr - 1)->mg_rpv
        + (res_ptr + to_ptr - 1)->mf_rpv - (res_ptr + to_ptr)->z1;
    (res_ptr + to_ptr)->z4 = ((res_ptr + to_ptr - 1)->qdh
        / (res_ptr + to_ptr - 1)->wsrv) - (res_ptr + to_ptr - 1)->hg_rpv;
    (res_ptr + to_ptr)->z5 = (res_ptr + to_ptr - 1)->erpv - (crpv * (res_ptr + to_ptr)->trpv)
        - ((res_ptr + to_ptr)->z1 * (res_ptr + to_ptr)->hg_rpv);
    (res_ptr + to_ptr)->z6 = (res_ptr + to_ptr)->hf_rpv - ((res_ptr + to_ptr)->z2
        * (res_ptr + to_ptr)->hg_rpv) + ((res_ptr + to_ptr)->z4
        * (1 - (res_ptr + to_ptr)->z2));
    (res_ptr + to_ptr)->mf_rpv = ((res_ptr + to_ptr)->z5 + ((res_ptr + to_ptr)->z3
        * (res_ptr + to_ptr)->z4)) / (res_ptr + to_ptr)->z6;
    (res_ptr + to_ptr)->mg_rpv = (res_ptr + to_ptr)->z1 - ((res_ptr + to_ptr)->z2
        * (res_ptr + to_ptr)->mf_rpv);
    (res_ptr + to_ptr)->erpv = ((res_ptr + to_ptr)->trpv * crpv) + ((res_ptr + to_ptr)->hf_rpv
        * (res_ptr + to_ptr)->mf_rpv) + ((res_ptr + to_ptr)->hg_rpv
        * (res_ptr + to_ptr)->mg_rpv);
    (res_ptr + to_ptr)->dmg_srv = (((res_ptr + to_ptr)->z2 - 1) * (res_ptr + to_ptr)->mf_rpv)
        + (res_ptr + to_ptr)->z3;
    (res_ptr + to_ptr)->dt = (res_ptr + to_ptr)->dmg_srv / (res_ptr + to_ptr)->wsrv;
    (res_ptr + to_ptr)->t = (res_ptr + to_ptr - 1)->t + (res_ptr + to_ptr)->dt;
    (res_ptr + to_ptr)->qdh = calc_qdh((res_ptr + to_ptr)->t);
    (res_ptr + to_ptr)->mf_sp = (res_ptr + to_ptr - 1)->mf_sp + (res_ptr + to_ptr)->dmg_srv;
}

// set final torus energy and temperature

to_ptr = ((index1 + 1) * n_values) - 1;
(res_ptr + to_ptr)->tsp = tsp_hctl_2[index1];
(res_ptr + to_ptr)->esp = (res_ptr + to_ptr)->mf_sp * (tsp_hctl_2[index1] - 32);

// set lowest index to calculate torus states for

torus_calc_base = index1 * n_values;
```

```
    // determine torus energy and temperature for successively higher RPV pressures

    for (index2 = to_ptr -1; index2 >= torus_calc_base; --index2) {
        to_ptr =  index2;
        (res_ptr + to_ptr)->esp = (res_ptr + to_ptr + 1)->esp - ((res_ptr + to_ptr + 1)->hg_rpv
            * (res_ptr + to_ptr + 1)->dmg_srv);
        (res_ptr + to_ptr)->tsp = ((res_ptr + to_ptr)->esp / (res_ptr + to_ptr)->mf_sp) + 32;
    }

    // write high-pressure endpoint temperature to tsp_hctl_1[] for later use

    to_ptr = index1 * n_values;
    tsp_hctl_1[index1] = (res_ptr + to_ptr)->tsp;

    // cheat and use knowledge that no data is stored in vtorus
    // for water levels between 13.88 ft and 14 ft to enable use
    // of inequality to get tsp_hctl_1_lo

    if (vtorus[index1].wlsp < 13.9) tsp_hctl_1_lo = tsp_hctl_1[index1];

    // cheat and use knowledge that no data is stored in vtorus
    // for water levels between 14 ft and 15 ft to enable use
    // of inequality to get tsp_hctl_1_hi

    if (vtorus[index1].wlsp < 14.1) tsp_hctl_1_hi = tsp_hctl_1[index1];
}

// declare a file stream for output of intermediate results.
// while the file stream will be attached to an existing physical
// file (determined by the file name passed in through interim_05),
// a new identifier for the stream is used to preclude a compiler
// error due to redefinition of a variable

ofstream inter_out_05b(interim_05, ios::app);

for (index1 = 0; index1 < 11; ++index1) {
    inter_out_05b << '\n';
    inter_out_05b << setiosflags(ios::fixed);
    inter_out_05b << setprecision(2);
    inter_out_05b << "torus water level = " << vtorus[index1].wlsp << " ft\n";
    inter_out_05b << '\n';
    inter_out_05b << " prpv   trpv    hf_rpv    hg_rpv     vf_rpv     vg_rpv ";
    inter_out_05b << "     wsrv    mf_rpv  mg_rpv\n";
    inter_out_05b << "[psig]  [°F]  [Btu/lbm] [Btu/lbm]  [ft3/lbm]  [ft3/lbm]";
    inter_out_05b << "  [lbm/min.]  [lbm]   [lbm]\n";
    for (index2 = 0; index2 < n_values; ++index2) {
        to_ptr = (index1 * n_values) + index2;
        inter_out_05b << setprecision(0) << setw(5);
        inter_out_05b << (res_ptr + to_ptr)->prpv;
        inter_out_05b << setprecision(2) << setw(8);
        inter_out_05b << (res_ptr + to_ptr)->trpv;
        inter_out_05b << setprecision(2) << setw(9);
        inter_out_05b << (res_ptr + to_ptr)->hf_rpv;
        inter_out_05b << setprecision(1) << setw(10);
        inter_out_05b << (res_ptr + to_ptr)->hg_rpv;
        inter_out_05b << setprecision(5) << setw(11);
        inter_out_05b << (res_ptr + to_ptr)->vf_rpv;
        inter_out_05b << setprecision(4) << setw(11);
        inter_out_05b << (res_ptr + to_ptr)->vg_rpv;
        inter_out_05b << setprecision(0) << setw(10);
        inter_out_05b << (res_ptr + to_ptr)->wsrv;
        inter_out_05b << setprecision(0) << setw(10);
        inter_out_05b << (res_ptr + to_ptr)->mf_rpv;
        inter_out_05b << setprecision(0) << setw(8);
        inter_out_05b << (res_ptr + to_ptr)->mg_rpv;
        inter_out_05b << '\n';
    }
```

```
    inter_out_05b << '\n';
    inter_out_05b << " prpv      erpv      dmg_srv   mf_sp      dt        t    ";
    inter_out_05b << "      qdh          esp       tsp\n";
    inter_out_05b << "[psig]    [Btu]        [lbm]     [lbm]    [min.]   [min.]";
    inter_out_05b << "  [Btu/min.]    [Btu]     [°F]\n";
    for (index2 = 0; index2 < n_values; ++index2) {
        to_ptr = (index1 * n_values) + index2;
        inter_out_05b << setprecision(0) << setw(5);
        inter_out_05b << (res_ptr + to_ptr)->prpv;
        inter_out_05b << reseticsflags(ios::fixed);
        inter_out_05b << setiosflags(ios::scientific);
        inter_out_05b << setprecision(2) << setw(11);
        inter_out_05b << (res_ptr + to_ptr)->erpv;
        inter_out_05b << reseticsflags(ios::scientific);
        inter_out_05b << setiosflags(ios::fixed);
        inter_out_05b << setprecision(1) << setw(10);
        inter_out_05b << (res_ptr + to_ptr)->dmg_srv;
        inter_out_05b << setprecision(0) << setw(9);
        inter_out_05b << (res_ptr + to_ptr)->mf_sp;
        inter_out_05b << setprecision(4) << setw(8);
        inter_out_05b << (res_ptr + to_ptr)->dt;
        inter_out_05b << setprecision(3) << setw(8);
        inter_out_05b << (res_ptr + to_ptr)->t;
        inter_out_05b << setprecision(0) << setw(11);
        inter_out_05b << (res_ptr + to_ptr)->qdh;
        inter_out_05b << reseticsflags(ios::fixed);
        inter_out_05b << setiosflags(ios::scientific);
        inter_out_05b << setprecision(2) << setw(11);
        inter_out_05b << (res_ptr + to_ptr)->esp;
        inter_out_05b << reseticsflags(ios::scientific);
        inter_out_05b << setiosflags(ios::fixed);
        inter_out_05b << setprecision(1) << setw(7);
        inter_out_05b << (res_ptr + to_ptr)->tsp;
        inter_out_05b << '\n';
    }
    resetiosflags(ios::fixed);
}

inter_out_05b << '\n';

// close output file to assure file buffer is flushed

inter_out_05b.close();

// write tsp at 50 psi intervals to final results file

// declare a file stream for output of final results. while
// the file stream will be attached to an existing physical
// file (determined by the file name passed in through
// final_05), a new identifier for the stream is used to
// preclude a compiler error due to redefinition of a variable

ofstream final_out_05(final_05, ios::app);

// write header

final_out_05 << "Heat Capacity Temperature Limit:\n\n";
final_out_05 << "      ";
for (index1 = 0; index1 < 11; ++index1) {
    final_out_05 << "   wlsp";
}
final_out_05 << '\n' << "      ";
for (index1 = 0; index1 < 11; ++index1) {
    final_out_05 << "   [ft]";
}
final_out_05 << '\n' << "      ";
final_out_05 << setiosflags(ios::fixed) << setprecision(2);
for (index1 = 0; index1 < 11; ++index1) {
```

```
    final_out_05 << setw(7) << vtorus[index1].wlsp;
}
final_out_05 << '\n' << " prpv ";
for (index1 = 0; index1 < 11; ++ index1) {
    final_out_05 << "   tsp ";
}
final_out_05 << '\n';
final_out_05 << "[psig]";
for (index1 = 0; index1 < 11; ++index1) {
    final_out_05 << "  [°F] ";
}
final_out_05 << '\n';

// write final results at psrv_lift to file

index2 = 0;
final_out_05 << setprecision(0) << setw(5);
final_out_05 << (res_ptr + index2)->prpv;
final_out_05 << " ";
for (index1 = 0; index1 < 11; ++index1) {
    to_ptr = (index1 * n_values) + index2;
    final_out_05 << setprecision(1) << setw(7);
    final_out_05 << (res_ptr + to_ptr)->tsp;
}
final_out_05 << '\n';

// write results at 50 psi increments for pressure
// above 200 psig, and 25 psi increments for
// pressure below 200 psig

while (index2 < (n_values - 5)) {
    index2 = index2 + 5;
    final_out_05 << setprecision(0) << setw(5);
    final_out_05 << (res_ptr + index2)->prpv;
    final_out_05 << " ";
    for (index1 = 0; index1 < 11; ++index1) {
        to_ptr = (index1 * n_values) + index2;
        final_out_05 << setprecision(1) << setw(7);
        final_out_05 << (res_ptr + to_ptr)->tsp;
    }
    final_out_05 << '\n';
}

// if values at mrfp were not written in previous step,
// write them now

if (index2 != (n_values - 1)) {
    index2 = n_values - 1;
    final_out_05 << setprecision(0) << setw(5);
    final_out_05 << (res_ptr + index2)->prpv;
    final_out_05 << " ";
    for (index1 = 0; index1 < 11; ++index1) {
        to_ptr = (index1 * n_values) + index2;
        final_out_05 << setprecision(1) << setw(7);
        final_out_05 << (res_ptr + to_ptr)->tsp;
    }
    final_out_05 << '\n';
}
final_out_05 << '\n';

// return formatting to default values

final_out_05 << resetiosflags(ios::fixed);

// close output file to assure file buffer is flushed

final_out_05.close();
```

```
// rpv.h:
//      this file contains the instructions required to compute the following:
//      -   MARFP:  Minimum Alternate RPV Flooding Pressure
//      -   MNSRED: Minimum Number of SRVs Required for
//                  Emergency Depressurization
//      -   MRFP:   Minimum RPV Flooding Pressure
//      -   MCUTL:  Maximum Core Uncovery Time Limit
//      -   MSCRWL: Minimum Steam Cooling RPV Water Level
//      -   MZIRWL: Minimum Zero-Injection RPV Water Level
//
//      The methods used are established in BWROG, Emergency Procedure
//      Guidelines / Severe Accident Guidelines (EPGs/SAGs) Appendix C,
//      Rev. 1, dated July, 1997.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// STRUCTURE DECLARATIONS
//
//*************************

struct  n_p {
    int     n_open;     // number of open SRVs
    double  pressure;   // RPV pressure [psig]
};

struct  n_t {
    int     n_open;     // number of open SRVs
    double  time;       // time [min.]
};

struct  t_t {
    double  t_sd;       // time after shutdown [min.]
    double  t_cu_15;    // time to reach 1500°F [min.]
};

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

// variables required for calculation

double  dwsrv_dprpv;    // specific SRV flowrate (critical flow) [lbm/hr-psia]
double  hf_sp;          // enthalpy of water pumped from torus [Btu/lbm]
double  wg_core_1;      // minimum core steam flow required to maintain PCT < 1500°F [lbm/hr]
n_p     marfp[7];       // Minimum Alternate RPV Flooding Pressure
double  msrp;           // Minimum SRV Re-opening Pressure [psig]
n_t     mcfi[7];        // Minimum Core Flooding Interval
double  mcfi_nads;      // MCFI for nsrv_ads open SRVs [min.]
t_t     mcutl[9];       // Maximum Core Uncovery Time Limit
double  mscrwl;         // Minimum Steam Cooling RPV Water Level [in.]
double  mzirwl;         // Minimum Zero-Injection RPV Water Level [in.]
char    interim_06[30]; // file name for storage of intermediate results
char    final_06[30];   // file name for storage of final results

//*************************
//
// FUNCTION PROTOTYPES
//
//*************************
```

```
    // release memory used to store results
    delete[] res_ptr;
} // calc_hctl
```

```
// function to initialize value used in multiple modules

void    init_rpv_values(char interim_06[30]);

// function to sequence other modules used in RPV variable calculation

void    sequence_rpv(char interim_06[30], char final_06[30]);

// function to calculate MARFP

void    calc_marfp(char interim_06[30], char final_06[30]);

// function to calculate MNSRED

void    calc_mnsred(char interim_06[30], char final_06[30]);

// function to calculate MSRP

void    calc_msrp(char interim_06[30]);

// function to calculate MRFP

void    calc_mrfp(char interim_06[30], char final_06[30]);

// function to calculate MCFI

void    calc_mcfi(char interim_06[30], char final_06[30]);

// function to calculate MCUTL

void    calc_mcutl(char final_06[30]);

// function to calculate MSCRWL

void    calc_mscrwl(char final_06[30]);

// function to calculate MZIRWL

void    calc_mzirwl(char final_06[30]);

//*************************
// init_values function
//*************************

void    init_rpv_values(char interim_06[30])
{
    // declare a file stream for output of intermediate results.
    // while the file stream will be attached to an existing physical
    // file (determined by the file name passed in through interim_06 ,
    // a new identifier for the stream is used to preclude a compiler
    // error due to redefinition of a variable

    ofstream inter_out_06a(interim_06, ios::app);

    // Determine enthalpy of water pumped from torus.
    // Temperature was calculated by code in hctl.h

    hf_sp = hpt(500, tsp_hctl_2_lco);
    inter_out_06a << "hf_sp = " << hf_sp << " Btu/lbm\n";
    inter_out_06a << '\n';

    // close output file to assure flush of file buffer

    inter_out_06a.close();

} // end init_rpv_values
```

```
//*************************
// sequence_rpv function
//*************************

void    sequence_rpv(char interim_06[30], char final_06[30])
{
    // declare file streams for output of intermediate and final results.
    // while the file streams will be attached to existing physical
    // files (determined by the file names passed in through interim_06
    // and final_06), new identifiers for the streams are used to preclude
    // a compiler error due to redefinition of variables

    ofstream inter_out_06b(interim_06, ios::app);
    ofstream final_out_06b(final_06, ios::app);

    // write descriptive line to front of each file

    inter_out_06b << "RPV variables, intermediate results:\n\n";

    final_out_06b << "RPV variables, final results:\n\n";

    // close output files to assure flush of file buffers

    inter_out_06b.close();
    final_out_06b.close();

    cout << "calculating values associated with the RPV\n\n" << flush;

    init_rpv_values(interim_06);

    calc_marfp(interim_06, final_06);

    calc_mnsred(interim_06, final_06);

    calc_msrp(interim_06);

    calc_mrfp(interim_06, final_06);

    calc_mcfi(interim_06, final_06);

    calc_mcutl(final_06);

    calc_mscrwl(final_06);

    calc_mzirwl(final_06);

} // end sequence_rpv


//*************************
// calc_marfp function
//*************************

void    calc_marfp(char interim_06[30], char final_06[30])
{
    int     index1;
    double  wg_1500_max;
    double  wsrv_be;        // best-estimate SRV flowrate [lbm/hr]
    double  psrv_ref;       // reference pressure for best-estimate SRV flowrate [psig]

    // declare file streams for output of intermediate and final results.
    // while the file streams will be attached to existing physical
    // files (determined by the file names passed in through interim_06
    // and final_06), new identifiers for the streams are used to preclude
    // a compiler error due to redefinition of variables

    ofstream inter_out_06c(interim_06, ios::app);
    ofstream final_out_06c(final_06, ios::app);
```

```
// determine the best-estimate SRV flowrate. this is defined as 110% of the
// nameplate flow

wsrv_be = 1.10 * wsrv_name;
inter_out_06c << "wsrv_be = " << wsrv_be << " lbm/hr\n";

// determine the reference pressure for the best-estimate SRV flowrate. this is
// defined as 103% of the nameplate pressure

psrv_ref = 1.03 * psrv_name;
inter_out_06c << "psrv_ref = " << psrv_ref << " psig\n";

// determine the specific SRV flowrate (assuming critical flow)

dwsrv_dprpv = wsrv_be / (psrv_ref + 14.7);
inter_out_06c << "dwsrv_dprpv = " << dwsrv_dprpv << " lbm/hr-psia\n";

// determine the maximum bundle steam flow rate
// required to maintain PCT < 1500°F

wg_1500_max = 0;
for (index1 = 0; index1 < nbr_fuel_types; ++index1) {
    wg_1500_max = max_floats(wg_1500_max, wg_1500[index1]);
}

// determine minimum core steam flow required to
// maintain PCT < 1500°F

wg_core_1 = nbuns * wg_1500_max;
inter_out_06c << "wg_core_1 = " << wg_core_1 << " lbm/hr\n";
inter_out_06c << '\n';

// calculate the MARFP as a function of number of open SRVs

final_out_06c << "Minimum Alternate RPV Flooding Pressure:\n\n";
final_out_06c << "nsrv     marfp\n";
final_out_06c << "        [psig]\n";
final_out_06c << setiosflags(ios::fixed);
for (index1 = 0; index1 < nsrv_ads; ++index1) {
    marfp[index1].n_open = index1 + 1;
    marfp[index1].pressure = (wg_core_1 / ((index1 + 1) * dwsrv_dprpv)) - 14.7;
    final_out_06c << setprecision(0) << setw(3) << marfp[index1].n_open;
    final_out_06c << setprecision(1) << setw(11) << marfp[index1].pressure << '\n';
}

final_out_06c << '\n';
final_out_06c << resetiosflags(ios::fixed);

// close output files to assure flush of file buffers

inter_out_06c.close();
final_out_06c.close();

} // end calc_marfp

//*************************
// calc_mnsred function
//*************************

void    calc_mnsred(char interim_06[30], char final_06[30])
{
    int     index1;
    double  hg_srv;     // enthalpy of steam discharged from SRVs [Btu/lbm]
    int     nsrv_1;     // minimum number of open SRVs required for a
                        // low-pressure ECCS pump to maintain MARFP
    int     nsrv_2;     // minimum number of open SRVs required for a
                        // low-pressure ECCS pump to remove two-minute decay heat
```

```
double  prpv_1;      // maximum RPV pressure at which a low-pressure ECCS
                     // pump can maintain the MARFP [psig]
double  prpv_2;      // maximum RPV pressure at which a low-pressure ECCS
                     // pump can remove two-minute decay heat [psig]
double  prpv_lpci;   // maximum pressure at which lpci flow can maintain MARFP or
                     // remove two-minute decay heat [psig]
double  prpv_cs;     // maximum pressure at which cs flow can maintain MARFP or
                     // remove two_miute decay heat [psig]
double  rhof_sp;     // density of water pumped from torus [lbm/gal]
double  wf_core_1;   // minimum injection flow required to maintain MARFP [gpm]
double  wf_core_2;   // minimum injection flow required to
                     // remove two-minute decay heat [gpm]
double  wg_core_2;   // minimum core steam flow required
                     // to remove two-minute decay heat [lbm/hr]

// declare file streams for output of intermediate and final results.
// while the file streams will be attached to existing physical
// files (determined by the file names passed in through interim_06
// and final_06), new identifiers for the streams are used to preclude
// a compiler error due to redefinition of variables

ofstream inter_out_06d(interim_06, ios::app);
ofstream final_out_06d(final_06, ios::app);

// determine density of water pumped from the torus

rhof_sp = 1 / (k3 * vpt(500, tsp_hctl_2_lco));
inter_out_06d << "rhof_sp = " << rhof_sp << " lbm/gal\n";

// determine minimum injection flow required to maintain MARFP

wf_core_1 = wg_core_1 / (60 * rhof_sp);
inter_out_06d << "wf_core_1 = " << wf_core_1 << " gpm\n";

// determine maximum pressure at which one LPCI
// pump can inject wf_core_1 into the RPV

// lpci flow versus pressure is ordered from high flow to low flow
// therefore, working from the last (high) index to first (low) index,
// find the index of the first element with pump flow
// greater than that which is required

index1 = 9;
while (lpci_flow[index1].flow < wf_core_1) --index1;
prpv_lpci = lpci_flow[index1].rpv_press;

// repeat above process operating on core spray flow array

// cs flow versus pressure is ordered from high flow to low flow
// therefore, working from the last (high) index to first (low) index,
// find the index of the first element with pump flow
// greater than that which is required

index1 = 9;
while (cs_flow[index1].flow < wf_core_1) --index1;
prpv_cs = cs_flow[index1].rpv_press;

// the minimum RPV pressure at which any low-pressure
// ECCS can deliver enough flow to maintain MARFP is
// the lower of prpv_lpci and prpv_cs

prpv_1 = min_floats(prpv_lpci, prpv_cs);
inter_out_06d << "prpv_1 = " << prpv_1 << " psig\n";

// determine number of open SRVs required to maintain MARFP
// MARFP array is ordered from high to low RPV pressure

// first find the first element of the array for which
```

```
// the pressure (MARFP) is lower than prpv_1

index1 = 0;
while (marfp[index1].pressure > prpv_1) ++index1;

// assign the corresponding number of open SRVs to nsrv_1

nsrv_1 = marfp[index1].n_open;
inter_out_06d << "nsrv_1 = " << nsrv_1 << '\n';

// determine enthalpy of steam discharged through SRVs

hg_srv = hgsat_by_p(psrv_lift + 14.7);
inter_out_06d << "hg_srv = " << hg_srv << " Btu/lbm\n";

// determine core steam flow required to remove
// two-minute decay heat

wg_core_2 = (k1 * fqdh_2 * qrx_rated) / (hg_srv - hf_sp);
inter_out_06d << "wg_core_2 = " << wg_core_2 << " lbm/hr\n";

// determine minimum injection flow required
// to remove two-minute decay heat

wf_core_2 = wg_core_2 / (60 * rhof_sp);
inter_out_06d << "wf_core_2 = " << wf_core_2 << " gpm\n";

// determine maximum pressure at which one LPCI
// pump can inject wf_core_2 into the RPV

// lpci flow versus pressure is ordered from high flow to low flow
// therefore, working from the last (high) index to first (low) index,
// find the index of the first element with pump flow
// greater than that which is required

index1 = 9;
while (lpci_flow[index1].flow < wf_core_2) --index1;
prpv_lpci = lpci_flow[index1].rpv_press;

// repeat above process operating on core spray flow array

// cs flow versus pressure is ordered from high flow to low flow
// therefore, working from the last (high) index to first (low) index,
// find the index of the first element with pump flow
// greater than that which is required

index1 = 9;
while (cs_flow[index1].flow < wf_core_2) --index1;
prpv_cs = cs_flow[index1].rpv_press;

prpv_2 = min_floats(prpv_lpci, prpv_cs);
inter_out_06d << "prpv_2 = " << prpv_2 << " psig\n";

// determine number of open SRVs required for a low-pressure
// ECCS to remove two-minute decay heat

nsrv_2 = int(wg_core_2 / (dwsrv_dprpv * (prpv_2 + 14.7)));
inter_out_06d << "nsrv_2 = " << nsrv_2 << '\n';
inter_out_06d << '\n';

// determine the Minimum Number of SRVs
// Required for Emergency Depressurization

mnsred = max_int(nsrv_1, nsrv_2);

final_out_06d << "MNSRED = " << mnsred << '\n' << '\n';

// close output files to assure flush of file buffers
```

```
    inter_out_06d.close();
    final_out_06d.close();

} // end calc_mnsred

//*************************
// calc_msrp function
//*************************

void    calc_msrp(char interim_06[30])
{
    msrp = srv_open;

    // declare a file stream for output of intermediate results.
    // while the file stream will be attached to an existing physical
    // file (determined by the file name passed in through interim_06),
    // a new identifier for the stream is used to preclude a compiler
    // error due to redefinition of a variable

    ofstream inter_out_06e(interim_06, ios::app);

    inter_out_06e << "msrp = " << msrp << " psig\n";
    inter_out_06e << '\n';

    // close output file to assure flush of file buffer

    inter_out_06e.close();

} // end calc_msrp

//*************************
// calc_mrfp function
//*************************

void    calc_mrfp(char interim_06[30], char final_06[30])
{
    double  left_side;
    double  right_side;
    double  prpv_10;    // minimum RPV pressure at which MNSRED SRVs
                        // can remove 10 minute decay heat [psig]

    // declare file streams for output of intermediate and final results.
    // while the file streams will be attached to existing physical
    // files (determined by the file names passed in through interim_06
    // and final_06), new identifiers for the streams are used to preclude
    // a compiler error due to redefinition of variables

    ofstream inter_out_06f(interim_06, ios::app);
    ofstream final_out_06f(final_06, ios::app);

    // determine the minimum RPV pressure at which MNSRED SRVs
    // can remove ten-minute decay heat, a quick hand-calculation
    // demonstrates that the left-hand side of the relation in Appendix C
    // is less than the right-hand side, therefore initialize the left and right
    // side for prpv_10 = 0 psig and test for the first pressure at
    // which the left-hand side exceeds the right-hand side

    prpv_10 = 0;
    left_side = mnsred * dwsrv_dprpv * (prpv_10 + 14.7);
    right_side = (k1 * fqdh_10 * qrx_rated) / (hgsat_by_p(prpv_10 + 14.7) - hf_sp);
    while (left_side < right_side) {
        prpv_10 = prpv_10 + 0.1;
        left_side = mnsred * dwsrv_dprpv * (prpv_10 + 14.7);
        right_side = (k1 * fqdh_10 * qrx_rated) / (hgsat_by_p(prpv_10 + 14.7) - hf_sp);
    }

    // subtract 0.1 from prpv_10 to bring pressure back to value for
```

```
    // which left side of inequality is just below right side

    prpv_10 = prpv_10 - 0.1;
    inter_out_06f << "prpv_10 = " << prpv_10 << " psig\n";
    inter_out_06f << '\n';

    // determine MRFP

    mrfp = max_floats(msrp, prpv_10);

    final_out_06f << "MRFP = " << mrfp << " psig\n\n";

    // close output files to assure flush of file buffers

    inter_out_06f.close();
    final_out_06f.close();

} // end calc_mrfp

//*************************
// calc_mcfi function
//*************************

void    calc_mcfi(char interim_06[30], char final_06[30])
{
    int     index1;
    double  crpv_taf;   // heat capacity ot RPV below TAF [Btu/°F]
    double  ccore;      // heat capacity of the core [Btu/°F]
    double  tsrv;       // RPV temperature at lowest SRV lifting point [°F]
    double  tmrfp;      // RPV temperature at the MARFP [°F]
    double  z1, z2, z3; // quantities defined in Appendix C

    // declare file streams for output of intermediate and final results.
    // while the file streams will be attached to existing physical
    // files (determined by the file names passed in through interim_06
    // and final_06), new identifiers for the streams are used to preclude
    // a compiler error due to redefinition of variables

    ofstream inter_out_06g(interim_06, ios::app);
    ofstream final_out_06g(final_06, ios::app);

    // determine the heat capacities of the RPV and core

    crpv_taf = c_steel * mrpv_taf;
    inter_out_06g << "crpv_taf = " << crpv_taf << " Btu/°F\n";
    ccore = (c_clad * mclad) + (c_fuel * mfuel);
    inter_out_06g << "ccore = " << ccore << " Btu/°F\n";

    // determine RPV temperature at lowest srv lift point

    tsrv = tsat(psrv_lift + 14.7);
    inter_out_06g << "tsrv = " << tsrv << "°F\n";

    // determine RPV temperature at MARFP

    tmrfp = tsat(mrfp + 14.7);
    inter_out_06g << "tmrfp = " << tmrfp << "°F\n";

    // calculate quantities z1 and z3 as defined in App. C

    z1 = (k1 * qrx_rated * (fqdh_10 - fqdh_100)) / (3 * (hgsat_by_p(mrfp + 14.7) - hf_sp));
    inter_out_06g << "z1 = " << z1 << '\n';
    z3 = (ccore * (1500 - tmrfp) + crpv_taf * (tsrv - tmrfp)) / (hf_sp - hgsat_by_p(mrfp + 14.7));
    z3 = z3 - (vrpv_taf / vgsat_by_p(mrfp + 14.7));
    inter_out_06g << "z3 = " << z3 << '\n';
    inter_out_06g << '\n';

    // calculate z2 and mcfi, which are functions of the
```

```
    // number of open SRVs

    final_out_06g << "Minimum Core Flooding Interval:\n\n";
    final_out_06g << "nsrv       z2        mcfi\n";
    final_out_06g << "                    [min.]\n";
    final_out_06g << setiosflags(ios::fixed);
    for (index1 = 0; index1 < (nsrv_ads - mnsred + 1); ++index1) {
        mcfi[index1].n_open = int(mnsred)+ index1;
        z2 = mcfi[index1].n_open * dwsrv_dprpv * (mrfp + 14.7);
        z2 = z2 - ((k1 * fqdh_10 * qrx_rated) / (hgsat_by_p(mrfp + 14.7) - hf_sp));
        mcfi[index1].time = 30 * (sqrt((z2*z2) - (4 * z1 * z3)) - z2) / z1;
        if (mcfi[index1].n_open == nsrv_ads) {
            mcfi_nads = mcfi[index1].time;
        }
        final_out_06g << setprecision(0) << setw(3) << mcfi[index1].n_open;
        final_out_06g << setw(11) << z2;
        final_out_06g << setprecision(1) << setw(9) << mcfi[index1].time << '\n';
    }
    final_out_06g << resetiosflags(ios::fixed);
    final_out_06g << '\n';

    // close output files to assure flush of file buffers

    inter_out_06g.close();
    final_out_06g.close();

} // end calc_mcfi

//*************************
// calc_mcutl function
//*************************

void    calc_mcutl(char final_06[30])
{
    int index1, index2;
    t_t mcutl_min[9];

    // declare a file stream for output of final results.
    // while the file stream will be attached to an existing physical
    // file (determined by the file name passed in through final_06),
    // a new identifier for the stream is used to preclude a compiler
    // error due to redefinition of a variable

    ofstream final_out_06h(final_06, ios::app);

    // find the limiting values of t_cu_15

    index1 = 0;
    for (index2 = 0; index2 < 9; ++index2) {
        mcutl_min[index2].t_sd = t_to_1500[index1].t_sd[index2];
        mcutl_min[index2].t_cu_15 = t_to_1500[index1].t_cu_15[index2];
    }
    for (index1 = 1; index1 < nbr_fuel_types; ++index1) {
        for (index2 = 0; index2 < 9; ++index2) {
            mcutl_min[index2].t_cu_15 = min_floats(mcutl_min[index2].t_cu_15,
                t_to_1500[index1].t_cu_15[index2]);
        }
    }

    // initialize mctul[] at 0

    for (index1 = 0; index1 < 9; ++index1) {
        mcutl[index1].t_sd = 0;
        mcutl[index1].t_cu_15 = 0;
    }

    // find the first element of mcutl_min for
    // which t_sd is longer than mcfi_nads
```

```
    index1 = 0;
    while (mcutl_min[index1].t_sd < mcfi_nads) ++index1;

    // get the index of the element of mcutl_min
    // immediately before index1, this allows mcfi_ads
    // to be bracketted

    index2 = index1 - 1;

    // set mcutl[index2].t_sd to mcfi_nads

    mcutl[index2].t_sd = mcfi_nads;

    // interpolate on mcutl_min to find t_cu_15 at mcfi_ads

    mcutl[index2].t_cu_15 = interpolate(mcutl_min[index2].t_sd  , mcutl_min[index1].t_sd, mcfi_nads,
                                mcutl_min[index2].t_cu_15, mcutl_min[index1].t_cu_15);

    // fill the rest of the mcutl array with the values obtained from mcutl_min

    for (index2 = index1; index2 < 9; ++index2) {
        mcutl[index2].t_sd = mcutl_min[index2].t_sd;
        mcutl[index2].t_cu_15 = mcutl_min[index2].t_cu_15;
    }

    final_out_06h << "Maximum Core Uncovery Time Limit:\n\n";
    final_out_06h << " t_sd     t_cu_15\n";
    final_out_06h << "[min.]     [min.]\n";
    final_out_06h << setiosflags(ios::fixed);
    for (index1 = 0; index1 < 9; ++index1) {
        final_out_06h << " ";
        final_out_06h << setiosflags(ios::left);
        final_out_06h << setprecision(1) << setw(6) << mcutl[index1].t_sd;
        final_out_06h << resetiosflags(ios::left);
        final_out_06h << setprecision(2) << setw(9) << mcutl[index1].t_cu_15 << '\n';
    }
    final_out_06h << resetiosflags(ios::fixed);
    final_out_06h << '\n';

    // close output file to assure flush of file buffer

    final_out_06h.close();

} // end calc_mcutl

//*************************
// calc_mscrwl function
//*************************

void    calc_mscrwl(char final_06[30])
{
    int     index1;
    double  mscrwl_max; // Maximum Minimum Steam Cooling RPV Water Level
                        // for all fuel types

    // declare a file stream for output of final results.
    // while the file stream will be attached to an existing physical
    // file (determined by the file name passed in through final_06),
    // a new identifier for the stream is used to preclude a compiler
    // error due to redefinition of a variable

    ofstream final_out_06i(final_06, ios::app);

    mscrwl_max = wlrpv_baf;
    for (index1 = 0; index1 < nbr_fuel_types; ++index1) {
        mscrwl = wlrpv_baf + ((fafl_15[index1] * lfuel[index1]) / 100);
        mscrwl_max = max_floats(mscrwl, mscrwl_max);
```

```
    }
    final_out_06i << "MSCRWL = " << mscrwl_max << " in.\n\n";

    // close output file to assure flush of file buffer

    final_out_06i.close();

} // end calc_mscrwl

//*************************
// calc_mzirwl function
//*************************

void    calc_mzirwl(char final_06[30])
{
    int     index1;
    double  mzirwl_max; // Maximum Minimum Zero-Injection RPV Water Level
                        // for all fuel types

    // declare a file stream for output of final results.
    // while the file stream will be attached to an existing physical
    // file (determined by the file name passed in through final_06),
    // a new identifier for the stream is used to preclude a compiler
    // error due to redefinition of a variable

    ofstream final_out_06j(final_06, ios::app);

    mzirwl_max = wlrpv_baf;
    for (index1 = 0; index1 < nbr_fuel_types; ++index1) {
        mzirwl = wlrpv_baf + ((fafl_18[index1] * lfuel[index1]) / 100);
        mzirwl_max = max_floats(mzirwl, mzirwl_max);
    }
    final_out_06j << "MZIRWL = " << mzirwl_max << " in.\n\n";

    // close output file to assure flush of file buffer

    final_out_06j.close();

} // end calc_mzirwl
```

```
// boron.h:
//      This file contains the instructions required to compute the following
//          -   CSBW, Cold Shutdown Boron Weight
//          -   HSBW, Hot Shutdown Boron Weight
//          -   BIIT, Boron Injection Initiation Temperature
//
//      The methods used are established in BWROG, Emergency Procedure
//      Guidelines / Severe Accident Guidelines (EPGs/SAGs) Appendix C,
//      Rev. 1, dated July, 1997.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// STRUCTURE DECLARATION
//
//*************************

struct  t_q {
    double  tsp;    // pool temperature [°F]
    double  qrx;    // reactor power [%]
};

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

double  csbw;           // Cold Shutdown Boron Weight [lbm]
double  fb10_nat_wt;    // weight fraction of naturally occurring B-10 isotope
double  fb11_slc;       // nuclidic abundance of B-11 isotope in SLC tank
double  fb10_slc_wt;    // weight fraction of B-10 in SLC tank
double  vf_slc;         // specific volume of water from the SLC tank [ft3/lbm]
double  hsbw;           // Hot Shutdown Boron Weight [lbm]
double  xb_cld;         // cold shutdown boron requirement for SLC tank [ppm]
double  xb_hot;         // hot shutdown boron requirement for SLC tank [ppm]
double  delta_tk_lvl;   // change in indicated slc tank level per change in
                        // change in solution level [%/in.]
double  slc_tk_area;    // slc tank surface area [ft2]
t_q     biit[4];        // storage for points on the BIIT curve
char    interim_07[30]; // file name for storage of intermediate results
char    final_07[30];   // file name for storage of final results

//*************************
//
// FUNCTION PROTOTYPES
//
//*************************

// function to initialize values which remain fixed throughout calculation

void    init_boron_values(char interim_07[30]);

// function to sequence other modules used in boron variable calculations

void    sequence_boron(char interim_07[30], char final_07[30]);

// function to calculate the Cold Shutdown Boron Weight

void    calc_csbw(char interim_07[30], char final_07[30]);

// function to calculate the Hot Shutdown Boron Weight
```

```
void    calc_hsbw(char interim_07[30], char final_07[30]);

// function to calculate the Suppression Chamber Spray Initiation Pressure

void    calc_biit(char interim_07[30], char final_07[30]);

//*************************
// init_boron_values function
//*************************

void    init_boron_values(char interim_07[30])
{
    const   double  pi = 3.1415926535;  // numeric constant

    // determine weight fraction of B-10 in naturally occurring boron

    fb10_nat_wt = (fb10_nat * mb10) / ((fb10_nat * mb10) + (fb11_nat * mb11));

    // determine nuclidic abundance of B-11 as a fraction of of all boron in the SLC tank

    fb11_slc = 1 - fb10_slc;

    // determine weight fraction of B-10 as a fraction of all boron in the SLC tank

    fb10_slc_wt = (fb10_slc * mb10) / ((fb10_slc * mb10) + (fb11_slc * mb11));

    // determine slc tank surface area [ft2]

    slc_tk_area = pi * slc_tk_dia * slc_tk_dia / 4;

    // determine change in indicated tank level per change in tank level [%/in.]

    delta_tk_lvl = slc_tk_lvl_out_span / slc_tk_lvl_in_span;

    // determine specific volume of water in SLC tank [ft3/lbm]

    vf_slc = vpt(1000, tslc);

    // declare a file stream for output of intermediate results.
    // while the file stream will be attached to an existing physical
    // file (determined by the file name passed in through interim_07),
    // a new identifier for the stream is used to preclude a compiler
    // error due to redefinition of a variable

    ofstream inter_out_07a(interim_07, ios::app);

    inter_out_07a << setiosflags(ios::fixed) << setprecision(4);
    inter_out_07a << "fb10_nat_wt = " << fb10_nat_wt << '\n';
    inter_out_07a << "fb11_slc = " << fb11_slc << '\n';
    inter_out_07a << "fb10_slc_wt = " << fb10_slc_wt << '\n';
    inter_out_07a << "delta_tk_lvl = " << delta_tk_lvl << " %/in.\n";
    inter_out_07a << setprecision(3);
    inter_out_07a << "SLC tank surface area: " << slc_tk_area << " ft2\n";
    inter_out_07a << setprecision(5) << "vf_slc = " << vf_slc << " ft3/lbm\n";
    inter_out_07a << resetiosflags(ios::fixed);

    // close file to assure file buffer is flushed

    inter_out_07a.close();

} // end init_boron_values

//*************************
// sequence_boron function
//*************************

void    sequence_boron(char interim_07[30], char final_07[30])
```

```
{
    cout << "calculating variables related to boron\n\n" << flush;

    // declare file streams for output of intermediate and final results.
    // while the file streams will be attached to existing physical
    // files (determined by the file names passed in through interim_07
    // and final_07), new identifiers for the streams are used to preclude
    // a compiler error due to redefinition of variables

    ofstream inter_out_07b(interim_07, ios::app);
    ofstream final_out_07b(final_07, ios::app);

    // write headers to files

    inter_out_07b << "Boron related parameters, intermediate results:\n\n";

    final_out_07b << "Boron related values, final results:\n\n";

    // close files to assure file buffers are flushed

    inter_out_07b.close();
    final_out_07b.close();

    init_boron_values(interim_07);

    calc_csbw(interim_07, final_07);

    calc_hsbw(interim_07, final_07);

    calc_biit(interim_07, final_07);

} // end sequence_boron

//*************************
// calc_csbw function
//*************************

void    calc_csbw(char interim_07[30], char final_07[30])
{
    double  delta_tk_cld;   // change in SLC tank level corresponding to CSBW
    double  delta_slc_cld;  // change in indicated SLC tank level
                            // corresponding to CSBW

    // determine SLC tank boron concentration required for CSBW

    xb_cld = xb_cld_nat * fb10_nat_wt / fb10_slc_wt;

    // determine CSBW in lbm

    csbw = xb_cld * mf_rpv_cld / 1e6;

    // determine change in tank level [in.] corresponding to CSBW

    delta_tk_cld = (csbw * 1e6 * vf_slc * 12) / (xb_slc * slc_tk_area);

    // determine change in indicated tank level [%] corresponding to CSBW

    delta_slc_cld = delta_tk_cld * delta_tk_lvl;

    // declare file streams for output of intermediate and final results.
    // while the file streams will be attached to existing physical
    // files (determined by the file names passed in through interim_07
    // and final_07), new identifiers for the streams are used to preclude
    // a compiler error due to redefinition of variables

    ofstream inter_out_07c(interim_07, ios::app);
    ofstream final_out_07c(final_07, ios::app);
```

```
    // write results to appropriate files

    inter_out_07c << setiosflags(ios::fixed) << setprecision(4);
    inter_out_07c << "xb_cld = " << xb_cld << " ppm\n";
    inter_out_07c << setprecision(2);
    inter_out_07c << "delta_tk_cld = " << delta_tk_cld << " in.\n";
    inter_out_07c << resetiosflags(ios::fixed);

    final_out_07c << setiosflags(ios::fixed) << setprecision(1);
    final_out_07c << "CSBW = " << csbw << " lbm\n";
    final_out_07c << "change in tank level to inject CSBW = ";
    final_out_07c << delta_slc_cld << " %\n";
    final_out_07c << resetiosflags(ios::fixed);

    // close files to assure file buffers are flushed

    inter_out_07c.close();
    final_out_07c.close();

} // end calc_csbw

//*************************
// calc_hsbw function
//*************************

void    calc_hsbw(char interim_07[30], char final_07[30])
{
    double  delta_tk_hot;   // change in SLC tank level corresponding to HSBW
    double  delta_slc_hot;  // change in indicated SLC tank level
                            // corresponding to HSBW

    // determine SLC tank boron concentration required for HSBW

    xb_hot = xb_hot_nat * fb10_nat_wt / fb10_slc_wt;

    // determine HSBW in lbm

    hsbw = xb_hot * mf_rpv_hot / 1e6;

    // determine change in tank level (in.) corresponding to HSBW

    delta_tk_hot = (hsbw * 1e6 * vf_slc * 12) / (xb_slc * slc_tk_area);

    // determine change in indicated tank level (%) corresponding to HSBW

    delta_slc_hot = delta_tk_hot * delta_tk_lvl;

    // declare file streams for output of intermediate and final results.
    // while the file streams will be attached to existing physical
    // files (determined by the file names passed in through interim_07
    // and final_07), new identifiers for the streams are used to preclude
    // a compiler error due to redefinition of variables

    ofstream inter_out_07d(interim_07, ios::app);
    ofstream final_out_07d(final_07, ios::app);

    // write results to appropriate files

    inter_out_07d << setiosflags(ios::fixed) << setprecision(4);
    inter_out_07d << "xb_hot = " << xb_hot << " ppm\n";
    inter_out_07d << setprecision(2);
    inter_out_07d << "delta_tk_hot = " << delta_tk_hot << " in.\n";
    inter_out_07d << resetiosflags(ios::fixed);

    final_out_07d << setiosflags(ios::fixed) << setprecision(1);
    final_out_07d << "HSBW = " << hsbw << " lbm\n";
    final_out_07d << "change in tank level to inject HSBW = ";
    final_out_07d << delta_slc_hot << " %\n\n";
```

```
    final_out_07d << resetiosflags(ios::fixed);

    // close files to assure file buffers are flushed

    inter_out_07d.close();
    final_out_07d.close();

} // end calc_hsbw

//*************************
// calc_biit function
//*************************

void    calc_biit(char interim_07[30], char final_07[30])
{
    int     index1, index2;
    double  wb;             // boron injection rate [lbm/min.]
    double  t_hsbw;         // time to inject HSBW [hr]
    double  hf_cst;         // enthalpy of CST water [Btu/lbm]
    double  hg_srv;         // enthalpy of steam discharged through SRVs [Btu/lbm]
    double  tsp_hctl_lim1;  // limiting HCTL high pressure endpoint temperature [°F]
    double  hsp_hctl_lim1;  // enthalpy of water at 15 psia and tsp_hctl_lim1 [Btu/lbm]
    double  mf_sp_lco;      // mass of water in torus with water level at TS minimum [lbm]
    double  z1;             // defined quantity from Appendix C
    double  vsp;            // torus water volume at TS minimum LCO water level [ft3]

    // determine boron injection rate

    wb = (wslc * xb_slc) / (1e6 * k3 * vf_slc);

    // determine time to inject boron

    t_hsbw = hsbw / (60 * wb);

    // determine enthalpy of CST water

    hf_cst = hpt(1000, tcst);

    // determine enthalpy of steam discharged from SRVs

    hg_srv = hgsat_by_p(psrv_lift + 14.7);

    // determine limiting HCTL high pressure endpoint temperature

    tsp_hctl_lim1 = min_floats(tsp_hctl_1_lo, tsp_hctl_1_hi);

    // determine mass of water in torus with water level at the minimum
    // torus water level LCO and temperature at TS limit for scram

    index1 = 0;
    while (vtorus[index1].wlsp < wlsp_minlco) ++ index1;
    if (vtorus[index1].wlsp == wlsp_minlco) vsp = vtorus[index1].vsp;
    else {
        index2 = index1 - 1;
        vsp = interpolate(vtorus[index2].wlsp, vtorus[index1].wlsp, wlsp_minlco,
            vtorus[index2].vsp, vtorus[index1].vsp);
    }
    mf_sp_lco = vsp / vpt(15, tsp_scram);

    hsp_hctl_lim1 = hpt(15, tsp_hctl_lim1);

    // determine z1, as specified in Appendix C

    z1 = (t_hsbw * (hg_srv - hsp_hctl_lim1) * qrx_rated * k1);

    z1 = z1 / (100 * mf_sp_lco * (hg_srv - hf_cst));

    // determine temperature defining BIIT at power
```

```
// levels less than 10 minute decay heat

biit[0].tsp = tsp_hctl_lim1 - (100 * z1 * fqdh_10);
biit[1].tsp = biit[0].tsp;

// biit[0].qrx is by definition 0

biit[0].qrx = 0;

// determine percent power corresponding to
// to 10 minute decay heat and assign to biit[1].qrx

biit[1].qrx = 100 * fqdh_10;

// determine power at which HSBW will not lead to shutdown
// before TS requires scram on pool temperature

biit[2].qrx = (tsp_hctl_lim1 - tsp_scram) / z1;

// by definition, pool temperature at this power is tsp_scram

biit[2].tsp = tsp_scram;

// by definition, biit will only be graphed to 20% power, tsp_scram

biit[3].tsp = tsp_scram;
biit[3].qrx = 20;

// declare file streams for output of intermediate and final results.
// while the file streams will be attached to existing physical
// files (determined by the file names passed in through interim_07
// and final_07), new identifiers for the streams are used to preclude
// a compiler error due to redefinition of variables

ofstream inter_out_07e(interim_07, ios::app);
ofstream final_out_07e(final_07, ios::app);

// write intermediate results to file

inter_out_07e << setiosflags(ios::fixed);
inter_out_07e << setprecision(3) << "wb = " << wb << " lbm/min.\n";
inter_out_07e << "t_hsbw = " << t_hsbw << " hr\n";
inter_out_07e << setprecision(2) << "hf_cst = " << hf_cst << " Btu/lbm\n";
inter_out_07e << setprecision(1) << "hg_srv = " << hg_srv << " Btu/lbm\n";
inter_out_07e << setprecision(1) << "tsp_hctl_lim1 = " << tsp_hctl_lim1 << "°F\n";
inter_out_07e << setprecision(0) << "mf_sp_lco = " << mf_sp_lco << " lbm\n";
inter_out_07e << setprecision(3) << "z1 = " << z1 << '\n';
inter_out_07e << resetiosflags(ios::fixed);
inter_out_07e << '\n';

// close file to assure file buffer is flushed

inter_out_07e.close();

// write final results to file

final_out_07e << setiosflags(ios::fixed);
final_out_07e << "Boron Injection Initiation Temperature:\n\n";
final_out_07e << "power     biit\n";
final_out_07e << " [%]      [°F]\n";
for (index1 = 0; index1 < 4; ++index1) {
    final_out_07e << setprecision(2) << setw(5) << biit[index1].qrx;
    final_out_07e << setprecision(1) << setw(9) << biit[index1].tsp;
    final_out_07e << '\n';
}
final_out_07e << '\n';

final_out_07e << resetiosflags(ios::fixed);
```

```
    // close file to assure file buffer is flushed
    final_out_07e.close();
} // end calc_biit
```

```
// psp.h:
//      this file contains the instructions required to compute the PSP, the
//      lesser of:
//      -   The highest torus pressure which can occur without steam in
//          the airspace.
//      -   The highest torus pressure at which initiation of RPV
//          depressurization will not result in exceeding PCPL-A before
//          RPV pressure drops to the Minimum RPV Flooding Pressure.
//      -   The highest torus pressure which can be maintained without
//          exceeding the torus boundary design load if SRVs are opened.
//
//      The methods used are established in BWROG, Emergency Procedure
//      Guidelines / Severe Accident Guidelines (EPGs/SAGs) Appendix C,
//      Rev. 1, dated July, 1997.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// STRUCTURE DECLARATIONS
//
//*************************

// data describing points on PSP curve

struct  p_1 {
    double  pressure;       // RPV pressure [psig]
    double  level;          // torus water level [ft]
};

// Wrapper for hf_tap (allows use of assignment statement in calc_psp()).
// Visual C++ compiler generates errors when a bare array of double is
// used for hf_tap.

struct  water_over_tap {
    double  hf_tap;         // head of water above torus pressure instrument tap [ft]
};

struct  int_res {
    double  pnsc_dw;        // partial pressure of dry gas (N2 or air) originating
                            // in drywell normalized to 100 °F using ideal gas law
                            // when in torus airspace [psia]
    double  dpsc_dtsc;      // rate of change of pressure with temperature of nitrogen
                            // or air in the torus [psia/°R]
    double  pg_hp;          // partial pressure of water at tsp_hctl_1 [psia]
    double  pg_lp;          // partial pressure of water at tsp_hctl_2 [psia]
    double  psc_hp_1;       // pressure (maximum) of torus at tsp_hctl_1 [psig]
    double  psc_hp_2;       // pressure (maximum) of torus at tsp_hctl_1 such
                            // that increase to tsp_hctl_2 will not cause pressure
                            // in excess of ppc_lim_a [psig]
    double  z1;             // quantity calculated in accordance with Appendiz C
    double  psc_srv;        // maximum torus pressure such that SRV actuation will
                            // not result in loads above psp_des [psig]
    double  psc_lim;        // maximum of psc_hp_1, psc_hp_2 or psc_srv [psig]
};

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

int_res to_psp[11];         // storage for intermediate results of PSP calculation
```

```
p_1     psp[21];            // storage for results of PSP calculation
water_over_tap  h_tap[21];  // array to hold hf_tap values
char    interim_08[30];     // file name for storage of intermediate results
char    final_08[30];       // file name for storage of final results

//*************************
//
// FUNCTION PROTOTYPE
//
//*************************

// function to calculate the Pressure Suppression Pressure

void    calc_psp(char interim_08[30], char final_08[30]);

//*************************
// calc_psp function
//*************************

void    calc_psp(char interim_08[30], char final_08[30])
{
    int     index1, index2;
    double  wlsp_lim;       // limiting torus water level [ft]
    double  pg_dw;          // partial pressure of water vapor in drywell [psia]
    double  pg_sc;          // partial pressure of water vapor in torus [psia]
    double  pndw_dw;        // partial pressure of dry gas (N2 or air) originating
                            // in drywell normalized to 100 °F using ideal gas law
                            // when in drywell [psia]
    double  pnsc_sc;        // partial pressure of dry gas (N2 or air) originating in
                            // torus airspace normalized to 100°F using ideal gas law
                            // when in torus airspace [psia]
    double  vf_flood;       // specific volume of water in torus when containment
                            // is flooded [ft3/lbm]
    double  dpsp_dwlsp;     // rate of change of pressure on submerged components
                            // with change of containment water level [psig/ft]
    double  level_float;    // variable to hold value '0' when calculating h_tap[].hf_tap

    cout << "calculating psp\n\n" << flush;

    // declare file streams for output of intermediate and final results.
    // while the file streams will be attached to existing physical
    // files (determined by the file names passed in through interim_08
    // and final_08), new identifiers for the streams are used to preclude
    // a compiler error due to redefinition of variables

    ofstream inter_out_08(interim_08, ios::app);
    ofstream final_out_08(final_08, ios::app);

    inter_out_08 << "PSP calculation, intermediate results:\n\n";

    // determine (lower) limiting torus water level

    wlsp_lim = hdco - hsp_ref;

    // determine pg_dw and pg_sc, assuming 100% relative humidity
    // spaces at maximum temperature for normal operation

    pg_dw = psat(tdw_maxop);
    pg_sc = psat(tsc_maxop);

    // calculate pndw_dw and pnsc_sc using ideal gas law

    pndw_dw = ((pdw_minop - pg_dw + 14.7) * 559.69) / (tdw_maxop + 459.69);
    pnsc_sc = ((psc_minop - pg_sc + 14.7) * 559.69) / (tsc_maxop + 459.69);

    // calculate specific volume of water in torus when flooded.
    // contrary to Appendix C calculation, vf_flood will be obtained
    // from subcooled water properties at tsp_flood and atmospheric
```

```
// pressure [15 psig], since it is unlikely that cold, saturated water
// will be pumped into the containment.

vf_flood = vpt(15, tsp_flood);

// calculate rate of change of pressure with
// level of submerged portions of containment

dpsp_dwlsp = 1 / (144 * vf_flood);

// write intermediate results to file

inter_out_08 << setiosflags(ios::fixed);
inter_out_08 << setprecision(2) << "wlsp_lim = " << wlsp_lim << " ft\n";
inter_out_08 << setprecision(3) << "pg_dw = " << pg_dw << " psia\n";
inter_out_08 << setprecision(3) << "pg_sc = " << pg_sc << " psia\n";
inter_out_08 << setprecision(2) << "pndw_dw = " << pndw_dw << " psia\n";
inter_out_08 << setprecision(2) << "pnsc_sc = " << pnsc_sc << " psia\n";
inter_out_08 << setprecision(4) << "vf_flood = " << vf_flood << " ft3/lbm\n";
inter_out_08 << setprecision(4) << "dpsp_dwlsp = " << dpsp_dwlsp << " psi/ft\n\n";

// calculate parameters which change with torus water level
// for levels between the downcomer opening and the
// elevation of the vent header

for (index1 = 0; index1 < 11; ++index1) {
    to_psp[index1].pnsc_dw = (pndw_dw * vdw) / vtorus[index1].vsc_air;
    to_psp[index1].dpsc_dtsc = (pnsc_sc + to_psp[index1].pnsc_dw) / 559.69;
    to_psp[index1].pg_hp = psat(tsp_hctl_1[index1]);
    to_psp[index1].pg_lp = psat(tsp_hctl_2[index1]);
    to_psp[index1].psc_hp_1 = ((tsp_hctl_1[index1] + 459.69) * to_psp[index1].dpsc_dtsc)
                    + to_psp[index1].pg_hp - 14.7;
    to_psp[index1].z1 = (tsp_hctl_1[index1] + 459.69)
                    / (tsp_hctl_2[index1] + 459.69);
    to_psp[index1].psc_hp_2 = to_psp[index1].z1 * (pcpl_to_calcs[index1]
                    - to_psp[index1].pg_lp + 14.7) + to_psp[index1].pg_hp + 14.7;
    to_psp[index1].psc_srv = psp_des - psp_srv - (dpsp_dwlsp * (vtorus[index1].wlsp
                    - wlsp_srv));
    to_psp[index1].psc_lim = min_floats(to_psp[index1].psc_hp_1, to_psp[index1].psc_hp_2);
    to_psp[index1].psc_lim = min_floats(to_psp[index1].psc_lim, to_psp[index1].psc_srv);
}

// assign values to PSP as a function of torus water level
// this section of code uses a bit of trickery, in that it
// is necessary for hdco to be between 9 and 10 ft (9.58 ft at
// JAF) and for mpspcwl to be between 17 and 18 ft (17.85 ft
// at JAF) for it to function properly. since hdco and mpspcwl
// are not likely to change (at least significantly, further
// effort to generalize the code for this go around is
// not warranted

for (index1 = 0; index1 < 21; ++ index1) {
    index2 = index1 - 9;

    // psp for levels below hdco is 0, corrected for the
    // hydrostatic head of water above the pressure instrument tap

    if (index2 < 0) {
        psp[index1].level = index1 + 1;
        h_tap[index1].hf_tap = psp[index1].level - hsc_tap + hsp_ref;
        level_float = 0;
        h_tap[index1].hf_tap = max_floats(level_float, h_tap[index1].hf_tap);
        psp[index1].pressure = dpsp_dwlsp * h_tap[index1].hf_tap;
    }

    // psp for levels between hdco and mpspcwl is psc_lim, corrected for
    // the hydrostatic head of water above the pressure instrument tap
```

```
    if ((index2 >= 0) && (index2 < 11)) {
        psp[index1].level = vtorus[index2].wlsp;
        h_tap[index1].hf_tap = psp[index1].level - hsc_tap + hsp_ref;
        level_float = 0;
        h_tap[index1].hf_tap = max_floats(level_float, h_tap[index1].hf_tap);
        psp[index1].pressure = to_psp[index2].psc_lim
                    + dpsp_dwlsp * h_tap[index1].hf_tap;
    }

    // psp for level above mpspcwl is 0

    if (index2 == 11) {
        psp[index1].level = 18;
        psp[index1].pressure = 0;
    }
}

// write intermediate results which are a
// function of water level to file

inter_out_08 << " wlsp  vsc_air  pnsc_dw  dpsc_dtsc  tsp_hctl_1";
inter_out_08 << "  tsp_hctl_2  pg_hp   pg_lp  psc_hp_1\n";
inter_out_08 << " [ft]    [ft3]     [psia]  [psia/°R]     [°F]    ";
inter_out_08 << "      [°F]     [psia]  [psia]    [psig]\n";
for (index1 = 0; index1 < 11; ++index1) {
    inter_out_08 << setprecision(2) << setw(5) << vtorus[index1].wlsp;
    inter_out_08 << setprecision(0) << setw(9) << vtorus[index1].vsc_air;
    inter_out_08 << setprecision(2) << setw(8) << to_psp[index1].pnsc_dw;
    inter_out_08 << setprecision(4) << setw(11) << to_psp[index1].dpsc_dtsc;
    inter_out_08 << setprecision(1) << setw(11) << tsp_hctl_1[index1];
    inter_out_08 << setprecision(1) << setw(12) << tsp_hctl_2[index1];
    inter_out_08 << setprecision(2) << setw(9) << to_psp[index1].pg_hp;
    inter_out_08 << setprecision(2) << setw(8) << to_psp[index1].pg_lp;
    inter_out_08 << setprecision(2) << setw(9) << to_psp[index1].psc_hp_1;
    inter_out_08 << '\n';
}

inter_out_08 << '\n';
inter_out_08 << " wlsp   z1   ppc_lim_a  psc_hp_2  psc_srv  psc_lim  hf_tap\n";
inter_out_08 << " [ft]          [psig]    [psig]    [psig]    [psig]    [ft]\n";
for (index1 = 0; index1 < 11; ++index1) {
    inter_out_08 << setprecision(2) << setw(5) << vtorus[index1].wlsp;
    inter_out_08 << setprecision(2) << setw(6) << to_psp[index1].z1;
    inter_out_08 << setprecision(2) << setw(9) << pcpl_to_calcs[index1];
    inter_out_08 << setprecision(2) << setw(11) << to_psp[index1].psc_hp_2;
    inter_out_08 << setprecision(2) << setw(9) << to_psp[index1].psc_srv;
    inter_out_08 << setprecision(2) << setw(9) << to_psp[index1].psc_lim;
    inter_out_08 << setprecision(2) << setw(8) << h_tap[index1].hf_tap;
    inter_out_08 << '\n';
}
inter_out_08 << '\n';

inter_out_08 << resetiosflags(ios::fixed);

// close file to assure file buffer is flushed

inter_out_08.close();

// write final results to file

final_out_08 << "Pressure Suppression Pressure:\n\n";
final_out_08 << setiosflags(ios::fixed);
final_out_08 << " wlsp      psp\n";
final_out_08 << " [ft]     [psig]\n";
for (index1 = 0; index1 < 21; ++index1) {
    final_out_08 << setprecision(2) << setw(5) << psp[index1].level;
    final_out_08 << setprecision(1) << setw(9) << psp[index1].pressure;
    final_out_08 << '\n';
```

```
    }
    final_out_08 << '\n';

    final_out_08 << resetiosflags(ios::fixed);

    // close file to assure file buffer is flushed

    final_out_08.close();

} // end calc_psp
```

```
// dwsil.h:
//      This code calculates the Drywell Spray Initiation Limit
//      and Minimum Drywell Spray Flow. The methodolgy
//      used is that presented in BWROG, Emergency Procedure
//      Guidelines / Severe Accident Guidelines (EPGs/SAGs),
//      Appendix C, Rev. 1, dated July, 1997.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// STRUCTURE DECLARATION
//
//*************************

struct  pt {            // pressure-temperature couples
    double  pressure;
    double  temperature;
};

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

// generic parameters required by this calculation, calculated in function
// evapdrop, defined within subsequent code

double  dpevap;     // evaporative cooling pressure drop [psid]
double  pdw;        // initial pressure input into calc_evapdrop, [psig]
double  tdwevap;    // initial drywell temperature for which spray initiation
                    // at corresponding initial pressure, p1 will produce a given
                    // evaporative cooling pressure drop, dpevap. [°F]

// variables used by multiple modules within this code

pt  tdw_min[60];    // minimum drywell temperature as a function of pressure
                    // for all dry nitrogen in the drywell, from ideal gas law [°F]
pt  tdw_scram[60];  // drywell temperature which will prevent evaporative
                    // cooling pressure drop from lowering pressure below
                    // scram setpoint [°F]
pt  tdw_cap[60];    // initial drywell temperature which will prevent evaporative
                    // cooling pressure drop from lowering pressure below
                    // drywell to wetwell negative pressure capability [°F]
pt  dwsil[60];      // result of the dwsil calculation
pt  dwsil_eop[60];  // dwsil for use in eops

int n_tdw_scram;    // number of tdw_scram values calculated
int n_tdw_cap;      // number of tdw_cap values calculated
int n_dwsil;        // number of dwsil values stored
int n_dwsil_eop;    // number of dwsil_eop values stored
char    interim_09[30]; // name of file to store intermediate results
char    final_09[30];   // name of file to store final results

//*************************
//
// FUNCTION PROTOTYPES
//
//*************************

// function which calculates tdwevap
```

```
double  calc_evapdrop(double dpevap, double pdw);

// function which calculates dwsil

void    calc_dwisl(char interim_09[30], char final_09[30]);

// function which calculates MDSF

void    calc_mdsf(char final_09[30]);

//*************************
// calc_evapdrop function
//*************************

double  calc_evapdrop(double dpevap, double p1)
{
    const double RN = 55.1;      // ideal gas constant for nitrogen [ft-lbm/lbm-°R]
    const double CPN = 0.248;    // specific heat of nitrogen [Btu/lbm-°R]

    double  tspray = 32;     // minimum temperature of spray water [°F]
    double  t1, t2;          // initial and final temperatures [°F]
    double  hfspray;         // enthalpy of sprayed water [Btu/lbm]
    double  rside, lside;    // intermediate results

    // get enthalpy of spray water, given Appendix C pressure of 500 psia

    hfspray = hpt(500, tspray);

    // set initial values of rside and lside to ensure condition evaluated
    // by 'while' is true for the first execution

    rside = 0;
    lside = 0;

    // set final drywell temperature at temperature of the spray,
    // since that is the lowest value physically possible

    t2 = tspray;

    // calculate the initial drywell temperature required to assure
    // spray initiation at p1 (drywell pressure from calling statement)
    // results in a pressure drop of dpevap (allowed evaporative pressure
    // drop from calling statement) by iteratively evaluating relations
    // provided in BWROG EPG/SAG, Rev.1, App. C, Attachment 2

    while (rside >= lside) {
        rside = dpevap + psat(t2);
        lside = (RN * (hgsat_by_t(t2) - hfspray))/(CPN * 144 * vgsat_by_t(t2));
        t2 = t2 + 0.01;
    }
    t1 = ((p1 + 14.7) * (t2 + 459.69) / (p1 - dpevap - psat(t2) + 14.7)) - 459.69;

    return (t1);

} // end calc_evapdrop

//*************************
// calc_dwsil function
//*************************

void    calc_dwisl(char interim_09[30], char final_09[30])
{
    double  pndw_dw;        // partial pressure of drywell dry nitrogen in drywell
                            // normalized to 100°F using the ideal gas law [psia]
    double  pg_sc;          // partial pressure of water vapor in torus at 100%
                            // relative humidity [psia]
    double  pndw_sc;        // partial pressure of torus nitrogen in drywell
                            // normalized to 100°F using ideal gas law [psia]
```

```
double  dtdw_dpdw;       // rate of change of temperature with pressure of
                         // dry nitrogen in drywell from ideal gas law [°R/psia]
double  max_temp;        // calculated maximum dwsil temperature [°F]
int     n_max_temp;      // index (in dwsil array) at which max. temperature is located
int     n_low_cut;       // first index (in dwsil array) below n_max_temp at which
                         // temperature is lower than tdw_maxind
int     n_hi_cut;        // first index (in dwsil array) above n_max_temp at which
                         // temperature is lower than tdw_maxind
int     n_tdw_min;       // number of tdw_min values calculated
int     index_tdw;       // index into tdw_min
int     lo_press_tdw;    // last index into tdw_min at which tdw_min < tdw_cap
int     hi_press_tdw;    // first index into tdw_min for which tdw_min > tdw_cap
int     lo_press_dwsil;  // last index into dwsil at which tdw_min < tdw_cap
int     hi_press_dwsil;  // first index into dwsil for which tdw_min > tdw_cap
double  m_tdw_min;       // slope of tdw_min line
double  b_tdw_min;       // intercept of tdw_min line
double  m_dwsil;         // slope of dwsil line
double  b_dwsil;         // intercept of dwsil line
pt      intersect;
int     index_dwsil;
int     index_eop;
int     index;

cout << "calculating dwisl\n\n" << flush;

// declare file streams for output of intermediate and final results.
// while the file streams will be attached to existing physical
// files (determined by the file names passed in through interim_09
// and final_09), new identifiers for the streams are used to preclude
// a compiler error due to redefinition of variables

ofstream inter_out_09a(interim_09, ios::app);
ofstream final_out_09a(final_09, ios::app);

inter_out_09a << "Drywell spray calculation intermediate results:\n\n";

// find partial pressure of nitrogen in drywell

pndw_dw = ((pdw_maxop + 14.7) * 559.69) / (tdw_minop + 459.69);

// find partial pressure of water vapor in torus

pg_sc = psat(tsc_minop);

// find partial pressure of torus nitrogen in the drywell

pndw_sc = ((psc_maxop - pg_sc + 14.7) * 559.69 * vsc_dco) / ((tsc_minop + 459.69) * vdw);

// find rate of change of drywell temperature with pressure

dtdw_dpdw = 559.69 / (pndw_dw + pndw_sc);

// write values to intermediate results file

inter_out_09a << setiosflags(ios::fixed);
inter_out_09a << setprecision(2) << "pndw_dw = ";
inter_out_09a << setw(5) << pndw_dw << " psia\n";
inter_out_09a << "pg_sc = " << setw(4) << pg_sc << " psia\n";
inter_out_09a << "pndw_sc = " << setw(5) << pndw_sc << " psia\n";
inter_out_09a << setprecision(1) << "dtdw_dpdw = ";
inter_out_09a << setw(5) << dtdw_dpdw << " °R/psia\n";
inter_out_09a << '\n';

// determine the minimum drywell temperature as a function of pressure
// for all dry nitrogen in the drywell using the ideal gas law

pdw = 0;
index_tdw = 0;
```

```
n_tdw_min = 60;
while (pdw < n_tdw_min) {
    tdw_min[index_tdw].pressure = pdw;
    tdw_min[index_tdw].temperature = (dtdw_dpdw * (pdw + 14.7)) - 459.69;
    pdw = pdw + 1.0;
    ++index_tdw;
}

// calculate temperatures such that evaporative cooling pressure drop does not
// lower pressure below high drywell pressure scram setpoint

// first, calculate temperature corresponding to pdw_scram

// initialize number of tdw_scram[] and dwsil[] values calculated,
// n_tdw_scram and n_dwsil are also used as indices for storage arrays
// the maximum tdw value, max_temp is also initialized here

n_tdw_scram = 0;
n_dwsil = 0;
max_temp = 0;

// use this logic if pdw_scram is not an integer (referring
// to value, not variable type). this determines tdw_scram at
// the non-integral pressure, then aligns subsequent pdw's
// on integer values

if ((pdw_scram - int(pdw_scram)) != 0) {
    tdw_scram[n_tdw_scram].pressure = pdw_scram;
    tdw_scram[n_tdw_scram].temperature = calc_evapdrop(0, pdw_scram);

    // initially assign tdw_scram[] values to dwsil[] as well

    dwsil[n_dwsil].pressure = tdw_scram[n_tdw_scram].pressure;
    dwsil[n_dwsil].temperature = tdw_scram[n_tdw_scram].temperature;
    if (max_temp < dwsil[n_dwsil].temperature) {
        max_temp = dwsil[n_dwsil].temperature;
        n_max_temp = n_dwsil;
    }

    // increment indices into arrays

    ++n_tdw_scram;
    ++n_dwsil;

    // find first integral value of pressure above pdw_scram

    pdw = int(pdw_scram) + 1.0;
}

// use this logic if pdw_scram is an integer (referring
// to value, not variable type)

else {
    tdw_scram[n_tdw_scram].pressure = pdw_scram;
    tdw_scram[n_tdw_scram].temperature = calc_evapdrop(0, pdw_scram);

    // initially assign tdw_scram[] values to dwsil[] as well

    dwsil[n_dwsil].pressure = tdw_scram[n_tdw_scram].pressure;
    dwsil[n_dwsil].temperature = tdw_scram[n_tdw_scram].temperature;
    if (max_temp < dwsil[n_dwsil].temperature) {
        max_temp = dwsil[n_dwsil].temperature;
        n_max_temp = n_dwsil;
    }

    // increment indices into arrays

    ++n_tdw_scram;
```

```
    ++n_dwsil;

    // calculate next pressure value

    pdw = pdw_scram + 1.0;
}

// continue with pressure values up to the sum of
// the scram setpoint and the negative pressure capability

while (pdw <= (pdw_scram + dpdw_ww)) {
    dpevap = pdw - pdw_scram;
    tdw_scram[n_tdw_scram].pressure = pdw;
    tdw_scram[n_tdw_scram].temperature = calc_evapdrop(dpevap, pdw);

    // dwsil[] is still determined by tdw_scram[]

    dwsil[n_dwsil].pressure = tdw_scram[n_tdw_scram].pressure;
    dwsil[n_dwsil].temperature = tdw_scram[n_tdw_scram].temperature;

    // keep track of maximum value of dwsil[].temperature

    if (max_temp < dwsil[n_dwsil].temperature) {
        max_temp = dwsil[n_dwsil].temperature;
        n_max_temp = n_dwsil;
    }

    // increment pressure and indices into arrays

    pdw = pdw + 1.0;
    ++n_tdw_scram;
    ++n_dwsil;
}

// make sure tdw_scram is calculated for pdw = (pdw_scram + dpdw_ww)

pdw = pdw_scram + dpdw_ww;

// this loop will not execute if (pdw_scram + dpdw_ww)
// was evaluated in previous loop (i.e., if it is an integral value)

if ((pdw - int(pdw)) != 0) {
    dpevap = dpdw_ww;
    tdw_scram[n_tdw_scram].pressure = pdw;
    tdw_scram[n_tdw_scram].temperature = calc_evapdrop(dpevap, pdw);
    dwsil[n_dwsil].pressure = tdw_scram[n_tdw_scram].pressure;
    dwsil[n_dwsil].temperature = tdw_scram[n_tdw_scram].temperature;
    if (max_temp < dwsil[n_dwsil].temperature) {
        max_temp = dwsil[n_dwsil].temperature;
        n_max_temp = n_dwsil;
    }

    // increment indices into arrays (and number of
    // points which have been calculated

    ++n_tdw_scram;
    ++n_dwsil;
}

// calculate temperatures such that evaporative cooling pressure drop
// does not lower drywell pressure below drywell to wetwell negative
// pressure capability value

// find first integer value of pdw greater than (pdw_scram + dpdw_ww)

pdw = int(pdw) + 1.0;
n_tdw_cap = 0;
```

```
// determine tdw_cap[] values until 60 dwsil[] values have been obtained

while (n_dwsil < 60) {
    tdw_cap[n_tdw_cap].pressure = pdw;
    tdw_cap[n_tdw_cap].temperature = calc_evapdrop(dpdw_ww, pdw);

    // dwsil[] is now initially set equal to tdw_cap

    dwsil[n_dwsil].pressure = tdw_cap[n_tdw_cap].pressure;
    dwsil[n_dwsil].temperature = tdw_cap[n_tdw_cap].temperature;

    // continue checking for maximum temperature value

    if (max_temp < dwsil[n_dwsil].temperature) {
        max_temp = dwsil[n_dwsil].temperature;
        n_max_temp = n_dwsil;
    }

    // increment pdw and indices into arrays

    pdw = pdw + 1.0;
    ++n_tdw_cap;
    ++n_dwsil;
}

// begin process of finding point at which tdw_min[] and dwsil[] intersect

// synchronize indices into tdw_min[] and dwsil[]. they have the potential
// to get out of synch at pdw_scram and (pdw_scram + dpdw_ww), therefore
// this is done in two steps

index_tdw = 0;
index_dwsil = 0;

// since tdw_min[0].pressure = 0 and dwsil[0].pressure = pdw_scram,
// begin by incrementing index_tdw

while (tdw_min[index_tdw].pressure <= dwsil[index_dwsil].pressure) ++index_tdw;

// now tdw_min[].pressure is the first integral value above pdw_scram,
// therefore increment index_dwsil to catch up

++index_dwsil;

// increment index_tdw along with index_dwsil as long as tdw_min[] and
// dwsil[] have the same value of pressure (at their respective indices)

while ((tdw_min[index_tdw].pressure == dwsil[index_dwsil].pressure)
    && (tdw_min[index_tdw].temperature < dwsil[index_dwsil].temperature)) {
    ++index_tdw;
    ++index_dwsil;
}

// since an additional dwsil[] point may exist at pdw = pdw_scram + dpdw_ww
// increment index_dwsil to get past it

while (dwsil[index_dwsil].pressure < tdw_min[index_tdw].pressure) ++index_dwsil;

// the indices into tdw_min[] and dwsil[] are now in synch so we can
// go after the point at which the curves intersect. since tdw_min[].temperature
// is lower than dwsil[].temperature (i.e., tdw_cap[].temperature) at
// low pressure, increment indices into both arrays as long as tdw_min[].temperature
// is smaller than dwsil[].temperature (this results in finding the first set
// of indices with tdw_min[].temperature at or above dwsil[].temperature)

while (tdw_min[index_tdw].temperature < dwsil[index_dwsil].temperature) {
    ++index_tdw;
    ++index_dwsil;
```

```
}
lo_press_tdw = index_tdw - 1;
hi_press_tdw = index_tdw;
lo_press_dwsil = index_dwsil - 1;
hi_press_dwsil = index_dwsil;

// find the equations of the lines between these bracketting points

m_tdw_min = (tdw_min[hi_press_tdw].temperature - tdw_min[lo_press_tdw].temperature);
m_tdw_min = m_tdw_min / (tdw_min[hi_press_tdw].pressure - tdw_min[lo_press_tdw].pressure);
b_tdw_min = tdw_min[hi_press_tdw].temperature - (m_tdw_min * tdw_min[hi_press_tdw].pressure);

m_dwsil = (dwsil[hi_press_dwsil].temperature - dwsil[lo_press_dwsil].temperature);
m_dwsil = m_dwsil / (dwsil[hi_press_dwsil].pressure - dwsil[lo_press_dwsil].pressure);
b_dwsil = dwsil[hi_press_dwsil].temperature - (m_dwsil * dwsil[hi_press_dwsil].pressure);

// find the point of intersection

intersect.pressure = (b_dwsil - b_tdw_min) / (m_tdw_min - m_dwsil);
intersect.temperature = (m_dwsil * intersect.pressure) + b_dwsil;

// write values to intermediate results file

inter_out_09a << setprecision(2) << "pressure at intersection = ";
inter_out_09a << setw(5) << intersect.pressure << " psig\n";
inter_out_09a << setprecision(1) << "temperature at intersection = ";
inter_out_09a << setw(5) << intersect.temperature << "°F\n\n";

// store this temperature and pressure as
// the next to last member of dwsil[]

dwsil[hi_press_dwsil].pressure = intersect.pressure;
dwsil[hi_press_dwsil].temperature = intersect.temperature;

// store this temperature and an arbitrarily high pressure value
// (ppc_maxind was chosen) as the last member of dwsil[]

n_dwsil = hi_press_dwsil + 1;
dwsil[n_dwsil].temperature = intersect.temperature;
dwsil[n_dwsil].pressure = ppc_maxind;

// check if maximum dwsil temperature is greater than
// available indication and truncate curve if required

if (tdw_maxind < max_temp) {

    // find last dwsil array locations for which temperature is less than tdw_maxind

    // start at the location in which the maximum temperature is
    // stored and work backward to find the low pressure cutoff

    n_low_cut = n_max_temp;
    while (dwsil[n_low_cut].temperature > tdw_maxind) --n_low_cut;

    // start at the location in which the maximum temperature is
    // stored and work forward to find the high pressure cutoff

    n_hi_cut = n_max_temp;
    while (dwsil[n_hi_cut].temperature > tdw_maxind) ++n_hi_cut;

    // copy dwsil[] to dwsil_eop[] for pressures and temperatures to n_lo_cut

    index_eop = 0;
    index_dwsil = 0;
    while (index_dwsil <= n_low_cut) {
        dwsil_eop[index_eop].pressure = dwsil[index_dwsil].pressure;
        dwsil_eop[index_eop].temperature = dwsil[index_dwsil].temperature;
        ++index_eop;
```

```
        ++index_dwsil;
    }

    // find point at which rising segment of dwsil curve intersects tdw_maxind

    lo_press_dwsil = n_low_cut;
    hi_press_dwsil = n_low_cut + 1;

    // find the equation of the line between these bracketting points

    m_dwsil = (dwsil[hi_press_dwsil].temperature - dwsil[lo_press_dwsil].temperature);
    m_dwsil = m_dwsil / (dwsil[hi_press_dwsil].pressure - dwsil[lo_press_dwsil].pressure);
    b_dwsil = dwsil[hi_press_dwsil].temperature - (m_dwsil * dwsil[hi_press_dwsil].pressure);

    // find the point of intersection

    intersect.pressure = (tdw_maxind - b_dwsil) / m_dwsil;
    intersect.temperature = tdw_maxind;

    // store this temperature and pressure as the next members of dwsil_eop[]

    dwsil_eop[index_eop].pressure = intersect.pressure;
    dwsil_eop[index_eop].temperature = intersect.temperature;
    ++index_eop;

    // find point at which lowering segment of dwsil curve intersects tdw_maxind

    lo_press_dwsil = n_hi_cut - 1;
    hi_press_dwsil = n_hi_cut;

    // find the equation of the line between these bracketting points

    m_dwsil = (dwsil[hi_press_dwsil].temperature - dwsil[lo_press_dwsil].temperature);
    m_dwsil = m_dwsil / (dwsil[hi_press_dwsil].pressure - dwsil[lo_press_dwsil].pressure);
    b_dwsil = dwsil[hi_press_dwsil].temperature - (m_dwsil * dwsil[hi_press_dwsil].pressure);

    // find the point of intersection

    intersect.pressure = (tdw_maxind - b_dwsil) / m_dwsil;
    intersect.temperature = tdw_maxind;

    // store this temperature and pressure as the next members of dwsil_eop[]

    dwsil_eop[index_eop].pressure = intersect.pressure;
    dwsil_eop[index_eop].temperature = intersect.temperature;

    // now copy the rest of dwsil[] to dwsil_eop[]

    ++index_eop;
    index_dwsil = n_hi_cut;
    while (index_dwsil <= n_dwsil) {
        dwsil_eop[index_eop].pressure = dwsil[index_dwsil].pressure;
        dwsil_eop[index_eop].temperature = dwsil[index_dwsil].temperature;
        ++index_eop;
        ++index_dwsil;
    }

    // decrement index_eop and assign to n_dwsil_eop,
    // since that is number of values stored

    n_dwsil_eop = index_eop - 1;
}

// no truncation is required, therefore copy dwsil into dwsil_eop

else {
    index_dwsil = 0;
    index_eop = 0;
```

```
    n_dwsil_eop = n_dwsil;
    while (index_dwsil <= n_dwsil) {
        dwsil_eop[index_eop].pressure = dwsil[index_dwsil].pressure;
        dwsil_eop[index_eop].temperature = dwsil[index_dwsil].temperature;
        ++index_eop;
        ++index_dwsil;
    }
}

// write intermediate results to file

inter_out_09a << "Tdw_scram values:\n";
inter_out_09a << " pdw     tdw_scram\n";
inter_out_09a << "[psig]       [°F]\n";
inter_out_09a << setprecision(1);
for (index = 0; index < n_tdw_scram; ++index) {
    inter_out_09a << setw(4) << tdw_scram[index].pressure;
    inter_out_09a << setw(12) << tdw_scram[index].temperature << '\n';
}
inter_out_09a << '\n' << '\n';

inter_out_09a << "Tdw_cap values:\n";
inter_out_09a << " pdw      tdw_cap\n";
inter_out_09a << "[psig]      [°F]\n";
inter_out_09a << setprecision(1);
for (index = 0; index < n_tdw_cap; ++index) {
    inter_out_09a << setw(4) << tdw_cap[index].pressure;
    inter_out_09a << setw(12) << tdw_cap[index].temperature << '\n';
}
inter_out_09a << '\n' << '\n';

inter_out_09a << "Tdw_min values:\n";
inter_out_09a << " pdw     tdw_min\n";
inter_out_09a << "[psig]      [°F]\n";
inter_out_09a << setprecision(1);
for (index = 0; index < n_tdw_min; ++index) {
    inter_out_09a << setw(4) << tdw_min[index].pressure;
    inter_out_09a << setw(12) << tdw_min[index].temperature << '\n';
}
inter_out_09a << '\n' << '\n';

inter_out_09a << "DWISL values:\n";
inter_out_09a << " pdw      dwisl\n";
inter_out_09a << "[psig]     [°F]\n";
inter_out_09a << setprecision(1);
for (index = 0; index <= n_dwsil; ++index) {
    inter_out_09a << setw(4) << dwsil[index].pressure;
    inter_out_09a << setw(11) << dwsil[index].temperature << '\n';
}
inter_out_09a << '\n';

// close file to assure file buffer is flushed

inter_out_09a.close();

final_out_09a << "Drywell spray related variable, final results:\n\n";
final_out_09a << "Drywell Spray Initiation Limit:\n\n";
final_out_09a << setiosflags(ios::fixed) << setprecision(1);
final_out_09a << " pdw     dwsil\n";
final_out_09a << "[psig]     [°F]\n";
for (index = 0; index <= n_dwsil_eop; ++index) {
    final_out_09a << setw(4) << dwsil_eop[index].pressure;
    final_out_09a << setw(10) << dwsil_eop[index].temperature << '\n';
}
final_out_09a << '\n';

final_out_09a << resetiosflags(ios::fixed);
```

```
    // close file to assure file buffer is flushed

    final_out_09a.close();

} // end calc_dwisl

//*************************
// calc_mdsf function
//*************************

void    calc_mdsf(char final_09[30])
{
    double  msf_a;              // 'A' loop minimum spray flow [gpm]
    double  msf_b;              // 'B' loop minimum spray flow [gpm]
    double  mdsf;               // Minimum Drywell Spray Flow [gpm]

    cout << "calculating mdsf\n\n" << flush;

    // declare a file stream for output of final results.
    // while the file stream will be attached to an existing physical
    // file (determined by the file name passed in through final_09),
    // a new identifier for the stream is used to preclude a compiler
    // error due to redefinition of a variable

    ofstream final_out_09b(final_09, ios::app);

    // determine minimum spray required for each header

    msf_a = spray_hdr[0].num_nzls * spray_hdr[0].wnzl;
    msf_b = spray_hdr[1].num_nzls * spray_hdr[1].wnzl;

    // pass header flows to max_floats to determine the MDSF

    mdsf = max_floats(msf_a, msf_b);

    final_out_09b << setiosflags(ios::fixed) << setprecision(0);
    final_out_09b << "MDSF = " << mdsf << " gpm\n\n";

    final_out_09b << resetiosflags(ios::fixed);

    // close file to assure file buffer is flushed

    final_out_09b.close();

} // end calc_mdsf
```

```
// mdrir.cpp:
//      this file contains the instructions required to compute the MDRIR, the
//      lowest RPV injection rate at which it is expected that core debris will
//      be retained in the RPV when RPV water level cannot be determined to
//      be above the bottom of active fuel.
//
//      The methods used are established in BWROG, Emergency Procedure
//      Guidelines / Severe Accident Guidelines (EPGs/SAGs) Appendix C,
//      Rev. 1, dated July, 1997.
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// STRUCTURE DECLARATION
//
//*************************

struct      t_p_w {
    double  time;       // time after shutdown [min.]
    double  pressure;   // RPV pressure [psig]
    double  flow;       //
};

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

t_p_w   mdrir[3][25];   // storage for MDRIR results
char    interim_10[30]; // file name for storage of intermediate results
char    final_10[30];   // file name for storage of final results

//*************************
//
// FUNCTION PROTOTYPE
//
//*************************

// function to calculate the Minimum Debris Retention Injection Rate

void    calc_mdrir(char interim_10[30], char final_10[30]);

//*************************
// calc_mdrir function
//*************************

void    calc_mdrir(char interim_10[30], char final_10[30])
{
    int     index_p, index_t;
    double  vf_inj;     // specific volume of injection water [ft3/lbm]
    double  prpv_a;     // RPV pressure [psia]
    double  prpv_g;     // RPV pressure [psig]
    double  hf_inj;     // enthalpy of injected water [Btu/lbm]
    double  hg_rpv;     // enthalpy of steam discharged from RPV [Btu/lbm]

    cout << "calculating mdrir\n\n" << flush;

    // declare file streams for output of intermediate and final results.
    // while the file streams will be attached to existing physical
    // files (determined by the file names passed in through interim_10
    // and final_10), new identifiers for the streams are used to preclude
    // a compiler error due to redefinition of variables

    ofstream inter_out_10(interim_10, ios::app);
```

```
ofstream final_out_10(final_10, ios::app);

// Determine specific volume of water injected into the RPV,
// assuming water is at 500 psia and the HCTL low-pressure endpoint
// temperature for torus water level at the low-level LCO.

vf_inj = vpt(500, tsp_hctl_2_lco);

inter_out_10 << "MDRIR calculation intermediate results:\n\n";
inter_out_10 << setiosflags(ios::fixed) << setprecision(5);
inter_out_10 << "vf_inj = " << vf_inj << " ft3/lbm\n\n";

// close file to assure file buffer is flushed

inter_out_10.close();

// Calculate MDRIR as a function of RPV pressure. The calculation will
// be repeated for 3 pressures (0, 500 and 1000 psig).

// initialize RPV pressure at 0 psig

prpv_g = 0;
prpv_a = prpv_g + 14.7;

// initialize first index into MDRIR array

index_p = 0;

// Test that prpv_g is less than an arbitrarily chosen value which ensures loop
// is repeated 3 times when a pressure increment of 500 psig is used.

while (prpv_g < 1100) {

    // Determine enthalpy of water that reaches core debris.

    hf_inj = hfsat_by_p(prpv_a);

    // Determine enthalpy of steam discharged from the RPV.

    hg_rpv = hgsat_by_p(prpv_a);

    // Calculate MDRIR as a function of time after shutdown.

    for (index_t = 0; index_t < 25; ++index_t) {

        // set MDRIR array values for time and pressure

        mdrir[index_p][index_t].time = sd_pwr[index_t].time;
        mdrir[index_p][index_t].pressure = prpv_g;

        // determine MDRIR flowrate at this time and pressure

        mdrir[index_p][index_t].flow = (qrx_rated * sd_pwr[index_t].power * k1
                * vf_inj * k3) / ((hg_rpv - hf_inj) * 60 * 100);
    }

    // increment RPV pressure and index values

    prpv_g = prpv_g + 500;
    prpv_a = prpv_a + 500;
    ++index_p;
}

// write results to file

final_out_10 << setiosflags(ios::fixed);
final_out_10 << "Minimum Debris Retention Injection Rate:\n\n";
final_out_10 << "                    mdrir [gpm]\n";
```

```
    final_out_10 << " time                    rpv pressure\n";
    final_out_10 << "[min.]      0 psig      500 psig      1000 psig\n";
    for (index_t = 0; index_t < 25; ++index_t) {
        final_out_10 << setprecision(0) << setw(6) << mdrir[0][index_t].time;
        final_out_10 << setprecision(1) << setw(11) << mdrir[0][index_t].flow;
        final_out_10 << setprecision(1) << setw(12) << mdrir[1][index_t].flow;
        final_out_10 << setprecision(1) << setw(13) << mdrir[2][index_t].flow;
        final_out_10 << '\n';
    }
    final_out_10 << '\n';

    final_out_10 << resetiosflags(ios::fixed);

    // close file to assure file buffer is flushed

    final_out_10.close();

} // end calc_mdrir
```

```
// scsip.h:
//      this file contains the instructions required to compute the SCSIP, the
//      lowest pressure which will result when 95% of the noncondensibles
//      in the drywell have been transferred to the torus. The limit precludes
//      chugging in the Mark I containment.
//
//      The methods used are established in BWROG, Emergency Procedure
//      Guidelines / Severe Accident Guidelines (EPGs/SAGs) Appendix C,
//      Rev. 1, dated July, 1997.
//
//      This code was written to conform to the syntax required by Microsoft
//      Visual C++
//
// Author:      Dave Burch
// Copyright:   1998
// Version:     1.0

//*************************
//
// VARIABLE DECLARATIONS
//
//*************************

char    interim_11[30]; // file name for storage of interim results
char    final_11[30];   // file name for storage of final results

//*************************
//
// FUNCTION PROTOTYPE
//
//*************************

// function to calculate the Suppression Chamber Spray Initiation Pressure

void    calc_scsip(char interim_11[30], char final_11[30]);

//*************************
// calc_scsip function
//*************************

void    calc_scsip(char interim_11[30], char final_11[30])
{
    double  pg_sc;      // partial pressure of water vapor in torus [psia]
    double  pnsc_sc;    // partial pressure of dry gas (N2 or air) originating in torus airspace
                        // normalized to 100°F using ideal gas law when in torus airspace [psia]
    double  pg_dw;      // partial pressure of water vapor in drywell [psia]
    double  pnsc_dw;    // partial pressure of dry gas (N2 or air) originating in drywell normalized
                        // to 100 °F using ideal gas law when in torus airspace [psia]
    double  scsip;      // result of calculation (SCSIP) [psig]

    cout << "calculating scsip\n\n" << flush;

    // declare file streams for output of intermediate and final results.
    // while the file streams will be attached to existing physical
    // files (determined by the file names passed in through interim_10
    // and final_10), new identifiers for the streams are used to preclude
    // a compiler error due to redefinition of variables

    ofstream inter_out_11(interim_11, ios::app);
    ofstream final_out_11(final_11, ios::app);

    // determine pg_sc and pg_dw, assuming 100% relative humidity
    // spaces at maximum temperature for normal operation

    pg_sc = psat(tsc_maxop);
    pg_dw = psat(tdw_maxop);

    // calculate pnsc_sc and pnsc_dw using ideal gas law
```

```
    pnsc_sc = ((psc_minop - pg_sc + 14.7) * 559.69) / (tsc_maxop + 459.69);
    pnsc_dw = ((pdw_minop - pg_dw + 14.7) * 559.69 * vdw) / ((tdw_maxop + 459.69) * vsc_lcc ;

    // write intermediate results to file

    inter_out_11 << "SCSIP calculation intermediate results:\n\n";
    inter_out_11 << setiosflags(ios::fixed);
    inter_out_11 << setprecision(3);
    inter_out_11 << "pg_sc = " << pg_sc << " psia\n";
    inter_out_11 << "pg_dw = " << pg_dw << " psia\n";
    inter_out_11 << "pnsc_sc = " << pnsc_sc << " psia\n";
    inter_out_11 << "pnsc_dw = " << pnsc_dw << " psia\n\n";

    // close file to assure file buffer is flushed

    inter_out_11.close();

    // calculate scsip

    scsip = (tsp_minop + 459.69) * (pnsc_sc + (0.95 * pnsc_dw)) / 559.69;
    scsip = scsip + psat(tsp_minop) - 14.7;

    final_out_11 << setiosflags(ios::fixed) << setprecision(2);
    final_out_11 << "Suppression Chamber Spray Initiation Pressure:\n\n";
    final_out_11 << "scsip = " << scsip << " psig\n\n";

    final_out_11 << resetiosflags(ios::fixed);

    // close file to assure file buffer is flushed

    final_out_11.close();

} // end calc_scsip
```

Function: assign_float
Header File: plant data.h
Purpose: To return a single double (float) from a given input string.
Called By: load_plant_data
Uses: no user defined functions

Function: assign_int
Header File: plant data.h
Purpose: To return a single integer from a specified input string.
Called By: load_plant_data
Uses: no user defined functions

Function: assign_n_floats
Header File: plant data.h
Purpose: To return a specified number of floating point values from a specified input string.
Called By: load_plant_data
Uses: no user defined functions

Function: calc_biit
Header File: boron.h
Purpose: To calculate the Boron Injection Initiation Temperature.
Called By: sequence_boron
Uses: vpt
hpt
hgsat_by_p
min_floats
interpolate

Function: calc_csbw

Header File: boron.h

Purpose: To calculate the Cold Shutdown Boron Weight.

Called By: sequence_boron

Uses: no user defined functions

Function: calc_dppc_dwlpc

Header File: pcpl.h

Purpose: To calculate rate of change of pressure with primary containment water level of submerged portions of the containment.

Called By: sequence_pcpl

Uses: vpt

Function: calc_dwsil

Header File: drywell spray.h

Purpose: To calculate the Drywell Spray Initiation Limit.

Called By: do_eop_calcs::main()

Uses: psat
calc_evapdrop

Function: calc_evapdrop

Header File: drywell spray.h

Purpose: To calculate tdwevap, the initial drywell temperature as a function of subsequent evaporative cooling pressure drop and initial drywell pressure.

Called By: calc_dwsil

Uses: hpt
psat
hgsat_by_t
vgsat_by_t

Function: calc_hctl

Header File: hctl.h

Purpose: To calculate the Heat Capacity Temperature Limit.

Called By: do_eop_calcs::main()

Uses: calc_n_values
vfsat_by_p
vgsat_by_p
tsat
hfsat_by_p
hgsat_by_p
calc_qdh
vpt

Function: calc_hsbw

Header File: boron.h

Purpose: To calculate the Hot Shutdown Boron Weight.

Called By: sequence_boron

Uses: no user defined functions

Function: calc_marfp

Header File: rpv.h

Purpose: To calculate the Minimum Alternate RPV Flooding Pressure.

Called By: sequence_rpv

Uses: max_floats

Function: calc_mcfi

Header File: rpv.h

Purpose: To calculate the Minimum Core Flooding Interval.

Called By: sequence_rpv

Uses: tsat
hgsat_by_p
vgsat_by_p

Function: calc_mcutl

Header File: rpv.h

Purpose: To calculate the Maximum Core Uncovery Time Limit.

Called By: sequence_rpv

Uses: min_floats
interpolate

Function: calc_mdrir

Header File: mdrir.h

Purpose: To calculate the Minimum Debris Retention Injection Rate

Called By: do_eop_calcs::main()

Uses: vpt
hfsat_by_p
hgsat_by_p

Function: calc_mdsf

Header File: drywell spray.h

Purpose: To calculate the Minimum Drywell Spray Flow rate

Called By: do_eop_calcs::main()

Uses: no user defined functions

Function: calc_mnsred

Header File: rpv.h

Purpose: To calculate the Minimum Number of SRVs Required for Emergency Depressurization.

Called By: sequence_rpv

Uses: vpt
min_floats
hgsat_by_p
max_int

Function: calc_mpspcwl

Header File: mpspcwl.h

Purpose: To calculate the Maximum Pressure Suppression Primary Containment Water Level.

Called By: do_eop_calcs::main()

Uses: no user defined functions

Function: calc_mrfp

Header File: rpv.h

Purpose: To calculate the Minimum RPV Flooding Pressure.

Called By: sequence_rpv

Uses: hgsat_by_p
max_floats

Function: calc_mscrwl

Header File: rpv.h

Purpose: To calculate the Minimum Steam Cooling RPV Water Level.

Called By: sequence_rpv

Uses: max_floats

Function: calc_msrp

Header File: rpv.h

Purpose: To output the Minimum SRV Re-opening Pressure.

Called By: sequence_rpv

Uses: no user defined functions

Function: calc_mzirwl

Header File: rpv.h

Purpose: To calculate the Minimum Zero-Injection RPV Water Level.

Called By: sequence_rpv

Uses: max_floats

Function: calc_n_values

Header File: hctl.h

Purpose: To calculate number of state points to include in the HCTL calculation.

Called By: calc_hctl

Uses: no user defined functions

Function: calc_psp

Header File: psp.h

Purpose: To calculate the Pressure Suppression Pressure.

Called By: do_eop_calcs::main()

Uses: psat
vpt
min_floats
max_floats

Function: calc_qdh

Header File: hctl.h

Purpose: To calculate percentile decay heat as a function of time.

Called By: calc_hctl

Uses: interpolate

Function: calc_scsip

Header File: scsip.h

Purpose: To calculate the Suppression Chamber Spray Initiation Pressure

Called By: do_eop_calcs::main()

Uses: psat

Function: calc_stpll

Header File: stpll.h

Purpose: To calculate SRV Tail Pipe Level Limit.

Called By: do_eop_calcs::main()

Uses: min_floats
interpolate

Function: calc_tsp_hctl_2

Header File: hctl.h

Purpose: To calculate the Heat Capacity Temperature Limit low -pressure endpoint temperature.

Called By: do_eop_calcs::main()

Uses: psat
min_floats

Function: calculate_pcpl

Header File: pcpl.h

Purpose: To calculate Primary Containment Pressure Limit A and Primary Containment Pressure Limit C.

Called By: sequence_pcpl

Uses: min_floats

Function: cpfsat_by_p

Header File: thermo.h

Purpose: To return specific heat of fluid (Btu/lbm-°F) at saturation given pressure (psia).

Called By: not presently in use

Uses: no user defined functions

Function: cpfsat_by_t

Header File: thermo.h

Purpose: To return saturated fluid specific heat (Btu/lbm-°F) given temperature (°F).

Called By: not presently in use

Uses: no user defined functions

Function: cpgsat_by_p

Header File: thermo.h

Purpose: To return specific heat of vapor (Btu/lbm-°F) at saturation given pressure (psia).

Called By: not presently in use

Uses: no user defined functions

Function: cpgsat_by_t

Header File: thermo.h

Purpose: To return saturated vapor specific heat (Btu/lbm-°F) given temperature (°F).

Called By: not presently in use

Uses: no user defined functions

Function: cppt

Header File: thermo.h

Purpose: To return subcooled/superheated specific heat (Btu/lbm-°F) given temperature (°F) and pressure (psia).

Called By: not presently in use

Uses: no user defined functions

Function: do_pcpl_output

Header File: pcpl.h

Purpose: To write output of calculation to file.

Called By: sequence_pcpl

Uses: no user defined functions

Function: free_thermo

Header File: thermo.h

Purpose: To release memory used to store thermodynamic properties of water (steam).

Called By: do_eop_calcs::main()

Uses: no user defined functions

Function: get_component_strength

Header File: pcpl.h

Purpose: To calculate material strengths (at 'calculation' and limiting temperatures) for primary containment components.

Called By: sequence_pcpl

Uses: interpolate

Function: hfsat_by_p

Header File: thermo.h

Purpose: To return fluid enthalpy at saturation (Btu/lbm) given pressure (psia).

Called By: calc_hctl
calc_mdrir

Uses: no user defined functions

Function: hfsat_by_t

Header File: thermo.h

Purpose: To return saturated fluid enthalpy (Btu/lbm) given temperature (°F).

Called By: not presently in use

Uses: no user defined functions

Function: hgsat_by_p

Header File: thermo.h

Purpose: To return vapor enthalpy at saturation (Btu/lbm) given pressure (psia).

Called By: calc_biit
calc_hctl
calc_mcfi
calc_mnsred
calc_mrfp
calc_mdrir

Uses: no user defined functions

Function: hgsat_by_t

Header File: themo.h

Purpose: To return saturated vapor enthalpy (Btu/lbm) given temperature (°F).

Called By: calc_evapdrop

Uses: no user defined functions

Function: hpt

Header File: thermo.h

Purpose: To return subcooled/superheated enthalpy (Btu/lbm) given temperature (°F) and pressure (psia).

Called By: calc_biit
calc_evapdrop
init_rpv_values

Uses: no user defined functions

Function: init_boron_values

Header File: boron.h

Purpose: To initialize values which remain fixed, and are used by multiple modules within the calculation.

Called By: sequence_boron

Uses: no user defined functions

Function: init_rpv_values

Header File: rpv.h

Purpose: To initialize variables used within multiple modules.

Called By: sequence_rpv

Uses: hpt

Function: interpolate

Header File: shared functions.h

Purpose: To perform interpolation on floating point values.

Called By: calc_biit
calc_mcutl
calc_qdh
calc_stpll
get_component_strength
pcpl_for_calcs

Uses: no user defined function

Function: load_plant_data

Header File: plant data.h

Purpose: Builds a database of plant specific parameters for use by calling routines from input files 'plant_data.dat' and 'gen_data.dat.'

Called By: do_eop_calcs::main

Uses: assign_int
assign_float
assign_n_floats

Function: load_thermo

Header File: thermo.h

Purpose: To initialize variables containing thermodynamic properties of water (steam). Values are obtained from file 'thermo.dat.'

Called By: do_eop_calcs::main()

Uses: no user defined functions

Function: max_floats

Header File: shared function.h

Purpose: To return the maximum of two floating point values.

Called By: calc_marfp
calc_mrfp
calc_mscrwl
calc_mzirwl
calc_psp

Uses: no user defined functions

Function: max_int

Header File: shared functions.h

Purpose: To return the maximum of two integers.

Called By: calc_mnsred

Uses: no user defined functions

Function: min_floats

Header File: shared functions.h

Purpose: To return the minimum of two floating point values.

Called By: calc_blit
calc_mcutl
calc_mnsred
calc_tsp_hctl_2
calculate_pcpl
calc_stpll

Uses: no user defined functions

Function: pcpl_for_calcs

Header File: pcpl.h

Purpose: To generate PCPL data required for calculation of HCTL and PSP.

Called By: sequence_pcpl

Uses: interpolate

Function: pressure_for_component

Header File: pcpl.h

Purpose: To calculate containment airspace pressure for which load carrying capability of a component is not compromised.

Called By: sequence_pcpl

Uses: no user defined functions

Function: pressure_for_srvs

Header File: pcpl.h

Purpose: To calculate containment airspace pressure at which SRVs can be opened.

Called By: sequence_pcpl

Uses: no user defined functions

Function: psat

Header File: thermo.h

Purpose: To return saturation pressure (psia) given temperature (°F).

Called By: calc_dwsil
calc_evapdrop
calc_psp
calc_tsp_hctl_2
calc_scsip

Uses: no user defined functions

Function: sequence_boron

Header File: boron.h

Purpose: To control order in which modules used to calculate boron related parameters execute.

Called By: do_eop_calcs::main()

Uses: init_boron_values
calc_csbw
calc_hsbw
calc_biit

Function: sequence_pcpl

Header File: pcpl.h

Purpose: To control order in which modules used to calculate PCPL execute.

Called By: do_eop_calcs::main()

Uses: get_component_strength
calc_dppc_dwlpc
pressure_for_srvs
pressure_for_component
calculate_pcpl
pcpl_for_calcs
do_pcpl_output

Function: sequence_rpv

Header File: rpv.h

Purpose: To control order in which modules used to calculate parameters associated with RPV execute.

Called By: do_eop_calcs::main()

Uses: init_rpv_values
calc_marfp
calc_mnsred
calc_msrp
calc_mrfp
calc_mcfi
calc_mcutl
calc_mscrwl
calc_mzirwl

Function: sfsat_by_p

Header File: thermo.h

Purpose: To return fluid entropy at saturation (Btu/lbm) given pressure (psia).

Called By: not presently in use

Uses: no user defined functions

Function: sfsat_by_t

Header File: thermo.h

Purpose: To return saturated fluid entropy (Btu/lbm) given temperature (°F).

Called By: not presently in use

Uses: no user defined functions

Function: sgsat_by_p

Header File: thermo.h

Purpose: To return vapor entropy at saturation (Btu/lbm) given pressure (psia).

Called By: not presently in use

Uses: no user defined functions

Function: sgsat_by_t

Header File: thermo.h

Purpose: To return saturated vapor entropy (Btu/lbm) given temperature (°F).

Called By: not presently in use

Uses: no user defined functions

Function: spt

Header File: thermo.h

Purpose: To return subcooled/superheated entropy (Btu/lbm) given temperature (°F) and pressure (psia).

Called By: not presently in use

Uses: no user defined functions

Function: tsat

Header File: thermo.h

Purpose: To return saturation temperature (°F) given pressure (psia).

Called By: calc_hctl
calc_mcfi

Uses: no user defined functions

Function: vfsat_by_p

Header File: thermo.h

Purpose: To return fluid specific volume at saturation (ft3/lbm) given pressure (psia).

Called By: calc_hctl

Uses: no user defined functions

Function: vfsat_by_t

Header File: thermo.h

Purpose: To return saturated fluid specific volume (ft3/lbm) given temperature (°F).

Called By: not presently in use

Uses: no user defined functions

Function: vgsat_by_p

Header File: thermo.h

Purpose: To return vapor specific volume at saturation (ft3/lbm) given pressure (psia).

Called By: calc_hctl
calc_mcfi

Uses: no user defined functions

Function: vgsat_by_t

Header File: thermo.h

Purpose: To return saturated vapor specific volume (ft3/lbm) given temperature (psia).

Called By: calc_evapdrop

Uses: no user defined functions

Function: vpt

Header File: thermo.h

Purpose: To return subcooled/superheated specific volume (ft3/lbm) given temperature (°F) and pressure (psia).

Called By: calc_biit
calc_dppc_dwlpc
calc_hctl
calc_mnsred
calc_psp
calc_mdrir

Uses: no user defined functions

Function: write_input_values

Header File: eop calculation input.h

Purpose: To write plant-specific and generic input values to a named disk file

Called By: do_eop_calcs::main()

Uses: no user defined functions

EOP calculation intermediate results:

PCPL calculation, intermediate results:

vf_flood = 0.0160182 ft3/lbm
dppc_dwlpc = 0.433535 psig/ft

pressure limit for srv operation and to assure
structural integrity at temperatures for which
input pressure capabilities were calculated:

wlpc [ft]	ppc_srv [psig]	ppc_1 [psig]	ppc_2 [psig]	ppc_3 [psig]	ppc_4 [psig]	ppc_5 [psig]	ppc_6 [psig]	ppc_7 [psig]	ppc_8 [psig]	ppc_9 [psig]	ppc_10 [psig]
0	82.0	108.0	111.0	118.0	129.0	120.0					
1	82.0	108.0	110.6	118.0	129.0	120.0					
2	82.0	108.0	110.1	118.0	129.0	120.0					
3	82.0	108.0	109.7	118.0	129.0	120.0					
4	82.0	108.0	109.3	118.0	129.0	120.0					
5	82.0	108.0	108.8	118.0	129.0	120.0					
6	82.0	108.0	108.4	118.0	129.0	120.0					
7	82.0	108.0	108.0	118.0	129.0	120.0					
8	82.0	108.0	107.5	118.0	129.0	120.0					
9	82.0	108.0	107.1	118.0	129.0	120.0					
10	82.0	108.0	106.7	118.0	129.0	120.0					
11	82.0	108.0	106.2	118.0	129.0	120.0					
12	82.0	108.0	105.8	118.0	129.0	120.0					
13	82.0	108.0	105.4	118.0	129.0	120.0					
14	82.0	108.0	104.9	118.0	129.0	120.0					
15	82.0	108.0	104.5	118.0	129.0	120.0					
16	82.0	108.0	104.1	118.0	129.0	120.0					
17	82.0	108.0	103.6	118.0	129.0	120.0					
18	82.0	108.0	103.2	118.0	129.0	120.0					
19	82.0	108.0	102.8	118.0	129.0	120.0					
20	82.0	108.0	102.3	118.0	129.0	120.0					
21	82.0	108.0	101.9	118.0	129.0	120.0					
22	82.0	108.0	101.5	118.0	129.0	120.0					
23	82.0	108.0	101.0	118.0	129.0	120.0					
24	82.0	108.0	100.6	118.0	129.0	120.0					
25	82.0	108.0	100.2	118.0	129.0	120.0					
26	82.0	108.0	99.7	118.0	129.0	120.0					
27	82.0	108.0	99.3	118.0	129.0	120.0					
28	82.0	108.0	98.9	118.0	129.0	120.0					
29	82.0	108.0	98.4	118.0	129.0	120.0					
30	82.0	107.8	98.0	118.0	129.0	120.0					
31	82.0	107.3	97.6	118.0	129.0	119.7					
32	82.0	106.9	97.1	118.0	129.0	119.3					
33	82.0	106.5	96.7	118.0	129.0	118.8					
34	82.0	106.0	96.3	118.0	129.0	118.4					
35	82.0	105.6	95.8	118.0	129.0	118.0					
36	82.0	105.2	95.4	118.0	129.0	117.5					
37	82.0	104.7	95.0	118.0	129.0	117.1					
38	82.0	104.3	94.5	118.0	129.0	116.7					
39	82.0	103.9	94.1	118.0	129.0	116.2					
40	82.0	103.4	93.7	118.0	129.0	115.8					
41	82.0	103.0	93.2	118.0	129.0	115.4					
42	82.0	102.6	92.8	118.0	129.0	114.9					
43	82.0	102.1	92.4	118.0	129.0	114.5					
44	82.0	101.7	91.9	118.0	129.0	114.1					
45	82.0	101.3	91.5	118.0	129.0	113.6					
46	82.0	100.8	91.1	118.0	129.0	113.2					
47	82.0	100.4	90.6	118.0	129.0	112.8					
48	82.0	100.0	90.2	118.0	129.0	112.3					
49	82.0	99.5	89.8	118.0	129.0	111.9					
50	82.0	99.1	89.3	118.0	129.0	111.5					
51	82.0	98.7	88.9	118.0	128.6	111.0					
52	82.0	98.2	88.5	118.0	128.1	110.6					
53	82.0	97.8	88.0	118.0	127.7	110.2					
54	82.0	97.4	87.6	118.0	127.3	109.7					

55	82.0	96.9	87.2	118.0	126.8	109.3
56	82.0	96.5	86.7	118.0	126.4	108.9
57	82.0	96.1	86.3	118.0	126.0	108.4
58	82.0	95.6	85.9	118.0	125.5	108.0
59	82.0	95.2	85.4	118.0	125.1	107.6
60	82.0	94.8	85.0	118.0	124.7	107.1
61	82.0	94.3	84.6	118.0	124.2	106.7
62	82.0	93.9	84.1	118.0	123.8	106.3
63	82.0	93.5	83.7	118.0	123.4	105.8
64	82.0	93.0	83.3	118.0	122.9	105.4
65	82.0	92.6	82.8	118.0	122.5	105.0
66	82.0	92.2	82.4	118.0	122.1	104.5
67	81.5	91.7	82.0	118.0	121.6	104.1
68	81.1	91.3	81.5	118.0	121.2	103.7
69	80.7	90.9	81.1	118.0	120.8	103.2
70	80.2	90.4	80.7	118.0	120.3	102.8
71	79.8	90.0	80.2	118.0	119.9	102.4
72	79.4	89.6	79.8	118.0	119.5	101.9
73	78.9	89.1	79.4	118.0	119.0	101.5
74	78.5	88.7	78.9	118.0	118.6	101.1
75	78.1	88.3	78.5	118.0	118.2	100.6
76	77.6	87.8	78.1	118.0	117.7	100.2
77	77.2	87.4	77.6	118.0	117.3	99.8
78	76.8	87.0	77.2	118.0	116.9	99.3
79	76.3	86.5	76.8	118.0	116.4	98.9
80	75.9	86.1	76.3	118.0	116.0	98.5
81	75.5	85.7	75.9	118.0	115.6	98.0
82	75.0	85.2	75.5	118.0	115.1	97.6
83	74.6	84.8	75.0	118.0	114.7	97.2
84	74.2	84.4	74.6	118.0	114.3	96.7
85	73.7	83.9	74.1	118.0	113.8	96.3
86	73.3	83.5	73.7	117.6	113.4	95.9
87	72.9	83.1	73.3	117.1	113.0	95.4
88	72.4	82.6	72.8	116.7	112.5	95.0
89	72.0	82.2	72.4	116.3	112.1	94.6
90	71.6	81.8	72.0	115.8	111.7	94.1
91	71.1	81.3	71.5	115.4	111.2	93.7
92	70.7	80.9	71.1	115.0	110.8	93.3
93	70.3	80.5	70.7	114.5	110.4	92.8
94	69.8	80.0	70.2	114.1	109.9	92.4
95	69.4	79.6	69.8	113.7	109.5	92.0
96	69.0	79.2	69.4	113.2	109.1	91.5
97	68.5	78.7	68.9	112.8	108.6	91.1
98	68.1	78.3	68.5	112.4	108.2	90.6
99	67.7	77.9	68.1	111.9	107.8	90.2
100	67.2	77.4	67.6	111.5	107.3	89.8
101	66.8	77.0	67.2	111.1	106.9	89.3
102	66.4	76.6	66.8	110.6	106.5	88.9
103	65.9	76.1	66.3	110.2	106.0	88.5
104	65.5	75.7	65.9	109.8	105.6	88.0
105	0.0	0.0	0.0	0.0	0.0	0.0
106	0.0	0.0	0.0	0.0	0.0	0.0
107	0.0	0.0	0.0	0.0	0.0	0.0
108	0.0	0.0	0.0	0.0	0.0	0.0
109	0.0	0.0	0.0	0.0	0.0	0.0
110	0.0	0.0	0.0	0.0	0.0	0.0
111	0.0	0.0	0.0	0.0	0.0	0.0
112	0.0	0.0	0.0	0.0	0.0	0.0
113	0.0	0.0	0.0	0.0	0.0	0.0
114	0.0	0.0	0.0	0.0	0.0	0.0
115	0.0	0.0	0.0	0.0	0.0	0.0
116	0.0	0.0	0.0	0.0	0.0	0.0
117	0.0	0.0	0.0	0.0	0.0	0.0
118	0.0	0.0	0.0	0.0	0.0	0.0
119	0.0	0.0	0.0	0.0	0.0	0.0

pressure limit to assure structural integrity
at limiting temperature for structure:

wlpc [ft]	ppc_1 [psig]	ppc_2 [psig]	ppc_3 [psig]	ppc_4 [psig]	ppc_5 [psig]	ppc_6 [psig]	ppc_7 [psig]	ppc_8 [psig]	ppc_9 [psig]	ppc_10 [psig]	[psig]
0	105.3	108.2	102.1	111.6	120.0						
1	105.3	107.8	102.1	111.6	120.0						
2	105.3	107.4	102.1	111.6	120.0						
3	105.3	106.9	102.1	111.6	120.0						
4	105.3	106.5	102.1	111.6	120.0						
5	105.3	106.1	102.1	111.6	120.0						
6	105.3	105.6	102.1	111.6	120.0						
7	105.3	105.2	102.1	111.6	120.0						
8	105.3	104.8	102.1	111.6	120.0						
9	105.3	104.3	102.1	111.6	120.0						
10	105.3	103.9	102.1	111.6	120.0						
11	105.3	103.5	102.1	111.6	120.0						
12	105.3	103.0	102.1	111.6	120.0						
13	105.3	102.6	102.1	111.6	120.0						
14	105.3	102.2	102.1	111.6	120.0						
15	105.3	101.7	102.1	111.6	120.0						
16	105.3	101.3	102.1	111.6	120.0						
17	105.3	100.9	102.1	111.6	120.0						
18	105.3	100.4	102.1	111.6	120.0						
19	105.3	100.0	102.1	111.6	120.0						
20	105.3	99.6	102.1	111.6	120.0						
21	105.3	99.1	102.1	111.6	120.0						
22	105.3	98.7	102.1	111.6	120.0						
23	105.3	98.3	102.1	111.6	120.0						
24	105.3	97.8	102.1	111.6	120.0						
25	105.3	97.4	102.1	111.6	120.0						
26	105.3	97.0	102.1	111.6	120.0						
27	105.3	96.5	102.1	111.6	120.0						
28	105.3	96.1	102.1	111.6	120.0						
29	105.3	95.7	102.1	111.6	120.0						
30	105.1	95.2	102.1	111.6	120.0						
31	104.7	94.8	102.1	111.6	119.7						
32	104.2	94.4	102.1	111.6	119.3						
33	103.8	93.9	102.1	111.6	118.8						
34	103.4	93.5	102.1	111.6	118.4						
35	102.9	93.1	102.1	111.6	118.0						
36	102.5	92.6	102.1	111.6	117.5						
37	102.1	92.2	102.1	111.6	117.1						
38	101.6	91.8	102.1	111.6	116.7						
39	101.2	91.3	102.1	111.6	116.2						
40	100.8	90.9	102.1	111.6	115.8						
41	100.3	90.5	102.1	111.6	115.4						
42	99.9	90.0	102.1	111.6	114.9						
43	99.5	89.6	102.1	111.6	114.5						
44	99.0	89.2	102.1	111.6	114.1						
45	98.6	88.7	102.1	111.6	113.6						
46	98.2	88.3	102.1	111.6	113.2						
47	97.7	87.9	102.1	111.6	112.8						
48	97.3	87.4	102.1	111.6	112.3						
49	96.9	87.0	102.1	111.6	111.9						
50	96.4	86.6	102.1	111.6	111.5						
51	96.0	86.1	102.1	111.2	111.0						
52	95.6	85.7	102.1	110.8	110.6						
53	95.1	85.3	102.1	110.3	110.2						
54	94.7	84.8	102.1	109.9	109.7						
55	94.3	84.4	102.1	109.5	109.3						
56	93.8	84.0	102.1	109.0	108.9						
57	93.4	83.5	102.1	108.6	108.4						
58	93.0	83.1	102.1	108.2	108.0						
59	92.5	82.7	102.1	107.7	107.6						
60	92.1	82.2	102.1	107.3	107.1						
61	91.7	81.8	102.1	106.9	106.7						
62	91.2	81.4	102.1	106.4	106.3						
63	90.8	80.9	102.1	106.0	105.8						
64	90.4	80.5	102.1	105.6	105.4						

65	89.9	80.1	102.1	105.1	105.0
66	89.5	79.6	102.1	104.7	104.5
67	89.1	79.2	102.1	104.3	104.1
68	88.6	78.8	102.1	103.8	103.7
69	88.2	78.3	102.1	103.4	103.2
70	87.8	77.9	102.1	103.0	102.8
71	87.3	77.5	102.1	102.5	102.4
72	86.9	77.0	102.1	102.1	101.9
73	86.5	76.6	102.1	101.7	101.5
74	86.0	76.2	102.1	101.2	101.1
75	85.6	75.7	102.1	100.8	100.6
76	85.2	75.3	102.1	100.4	100.2
77	84.7	74.9	102.1	99.9	99.8
78	84.3	74.4	102.1	99.5	99.3
79	83.9	74.0	102.1	99.1	98.9
80	83.4	73.6	102.1	98.6	98.5
81	83.0	73.1	102.1	98.2	98.0
82	82.6	72.7	102.1	97.8	97.6
83	82.1	72.3	102.1	97.3	97.2
84	81.7	71.8	102.1	96.9	96.7
85	81.3	71.4	102.1	96.5	96.3
86	80.8	71.0	101.7	96.0	95.9
87	80.4	70.5	101.2	95.6	95.4
88	80.0	70.1	100.8	95.2	95.0
89	79.5	69.7	100.4	94.7	94.6
90	79.1	69.2	99.9	94.3	94.1
91	78.7	68.8	99.5	93.9	93.7
92	78.2	68.4	99.1	93.4	93.3
93	77.8	67.9	98.6	93.0	92.8
94	77.4	67.5	98.2	92.6	92.4
95	76.9	67.1	97.8	92.1	92.0
96	76.5	66.6	97.3	91.7	91.5
97	76.1	66.2	96.9	91.3	91.1
98	75.6	65.8	96.5	90.8	90.6
99	75.2	65.3	96.0	90.4	90.2
100	74.8	64.9	95.6	90.0	89.8
101	74.3	64.5	95.2	89.5	89.3
102	73.9	64.0	94.7	89.1	88.9
103	73.5	63.6	94.3	88.7	88.5
104	73.0	63.2	93.9	88.2	88.0
105	0.0	0.0	0.0	0.0	0.0
106	0.0	0.0	0.0	0.0	0.0
107	0.0	0.0	0.0	0.0	0.0
108	0.0	0.0	0.0	0.0	0.0
109	0.0	0.0	0.0	0.0	0.0
110	0.0	0.0	0.0	0.0	0.0
111	0.0	0.0	0.0	0.0	0.0
112	0.0	0.0	0.0	0.0	0.0
113	0.0	0.0	0.0	0.0	0.0
114	0.0	0.0	0.0	0.0	0.0
115	0.0	0.0	0.0	0.0	0.0
116	0.0	0.0	0.0	0.0	0.0
117	0.0	0.0	0.0	0.0	0.0
118	0.0	0.0	0.0	0.0	0.0
119	0.0	0.0	0.0	0.0	0.0

STPLL calculation, intermediate results:

Component design margins:

component	fdes [%]
quencher	35.3579
quencher support	218.951
tail pipe	86.6667
tail pipe support	50

breakpoints in STPLL vs. RPV pressure curve:

spwl [ft]	pressure [psig]
13.88	1692
14.66	1649
15.50	1286
18.51	1131
29.50	-448

STPLL:

spwl [ft]	pressure [psig]
15.00	1500
15.50	1286
18.51	1131
26.39	0
0.00	0

HCTL low-pressure endpoint intermediate results:

pg_sc = 0.33889 psia
pnsc_sc = 15.7623 psia
wlsp_lim = 9.58 ft

wlsp [ft]	vsc_air [ft3]	pnsc_dw [psia]	dpsc_dtsc [°R/psia]	pcpl_a [psig]	tsc_pcpl [°F]	tsp_hctl_2 [°F]
9.58	154763	18.98	0.0621	62.0	254.6	220.0
10.00	150890	19.46	0.0629	62.0	253.6	220.0
11.00	141436	20.77	0.0653	62.0	250.9	220.0
12.00	131960	22.26	0.0679	62.0	247.7	220.0
13.00	122200	24.03	0.0711	62.0	243.7	220.0
13.88	113603	25.85	0.0744	62.0	239.4	220.0
14.00	112428	26.12	0.0748	62.0	238.7	220.0
15.00	102119	28.76	0.0795	62.0	232.2	220.0
16.00	92820	31.64	0.0847	62.0	224.5	220.0
17.00	83463	35.19	0.0910	62.0	214.2	214.2
17.85	75334	38.99	0.0978	62.0	201.8	201.8

tsp_hctl_2_lco = 220.0°F

RPV variables, intermediate results:

hf_sp = 189.29 Btu/lbm

wsrv_be = 911900 lbm/hr
psrv_ref = 1112.4 psig
dwsrv_dprpv = 809.068 lbm/hr-psia
wg_core_1 = 705600 lbm/hr

rhof_sp = 7.98287 lbm/gal
wf_core_1 = 1473.16 gpm
prpv_1 = 195 psig
nsrv_1 = 5
hg_srv = 1186.57 Btu/lbm
wg_core_2 = 263598 lbm/hr
wf_core_2 = 550.342 gpm
prpv_2 = 195 psig
nsrv_2 = 1

msrp = 50 psig

prpv_10 = 33.3 psig

crpv_taf = 122752 Btu/°F
ccore = 25332.5 Btu/°F
tsrv = 562.877°F
tmrfp = 297.671°F

z1 = 31865.6
z3 = -65205.1

torus water level = 9.58 ft

prpv [psig]	trpv [°F]	hf_rpv [Btu/lbm]	hg_rpv [Btu/lbm]	vf_rpv [ft3/lbm]	vg_rpv [ft3/lbm]	wsrv [lbm/min.]	mf_rpv [lbm]	mg_rpv [lbm]
1145	562.88	566.16	1186.6	0.02217	0.3771	78190	532000	15300
1135	561.79	564.74	1187.0	0.02213	0.3808	77515	530124	15309
1125	560.70	563.31	1187.4	0.02210	0.3847	76841	528251	15314
1115	559.60	561.87	1187.8	0.02206	0.3886	76167	526376	15316
1105	558.49	560.42	1188.3	0.02202	0.3926	75493	524498	15315
1095	557.37	558.97	1188.7	0.02199	0.3966	74819	522619	15312
1085	556.24	557.51	1189.1	0.02195	0.4007	74144	520742	15305
1075	555.11	556.04	1189.5	0.02192	0.4049	73470	518860	15295
1065	553.97	554.57	1189.9	0.02188	0.4091	72796	516980	15283
1055	552.82	553.09	1190.3	0.02184	0.4135	72122	515102	15268
1045	551.66	551.60	1190.7	0.02181	0.4179	71447	513216	15250
1035	550.50	550.11	1191.1	0.02177	0.4224	70773	511335	15229
1025	549.32	548.60	1191.4	0.02173	0.4269	70099	509442	15205
1015	548.14	547.09	1191.8	0.02170	0.4316	69425	507556	15179
1005	546.95	545.57	1192.2	0.02166	0.4363	68751	505664	15150
995	545.75	544.04	1192.6	0.02163	0.4412	68076	503775	15118
985	544.54	542.51	1192.9	0.02159	0.4461	67402	501877	15083
975	543.32	540.96	1193.3	0.02155	0.4511	66728	499979	15046
965	542.10	539.40	1193.7	0.02152	0.4562	66054	498076	15007
955	540.86	537.85	1194.0	0.02148	0.4615	65379	496171	14965
945	539.61	536.27	1194.4	0.02145	0.4668	64705	494264	14920
935	538.35	534.70	1194.7	0.02141	0.4722	64031	492352	14873
925	537.09	533.11	1195.1	0.02137	0.4778	63357	490444	14823
915	535.81	531.51	1195.4	0.02134	0.4834	62683	488524	14771
905	534.52	529.90	1195.7	0.02130	0.4892	62008	486600	14716
895	533.22	528.28	1196.1	0.02127	0.4951	61334	484671	14659
885	531.91	526.65	1196.4	0.02123	0.5011	60660	482740	14600
875	530.59	525.02	1196.7	0.02119	0.5072	59986	480810	14538
865	529.26	523.37	1197.0	0.02116	0.5135	59311	478871	14474
855	527.92	521.71	1197.3	0.02112	0.5199	58637	476927	14407
845	526.56	520.04	1197.7	0.02109	0.5265	57963	474977	14338
835	525.20	518.35	1198.0	0.02105	0.5332	57289	473019	14267
825	523.82	516.66	1198.3	0.02101	0.5401	56614	471061	14194
815	522.42	514.95	1198.5	0.02098	0.5471	55940	469088	14118
805	521.02	513.24	1198.8	0.02094	0.5543	55266	467123	14040
795	519.60	511.50	1199.1	0.02090	0.5617	54592	465138	13960
785	518.16	509.76	1199.4	0.02087	0.5692	53918	463155	13878
775	516.72	508.00	1199.7	0.02083	0.5769	53243	461159	13793
765	515.26	506.24	1199.9	0.02080	0.5848	52569	459165	13706
755	513.78	504.45	1200.2	0.02076	0.5929	51895	457153	13618
745	512.30	502.65	1200.4	0.02072	0.6012	51221	455136	13527
735	510.80	500.84	1200.7	0.02069	0.6097	50546	453117	13433
725	509.28	499.01	1200.9	0.02065	0.6185	49872	451088	13338
715	507.74	497.17	1201.2	0.02061	0.6275	49198	449042	13241
705	506.19	495.31	1201.4	0.02058	0.6367	48524	446997	13141
695	504.62	493.43	1201.6	0.02054	0.6461	47850	444936	13039
685	503.03	491.54	1201.8	0.02050	0.6559	47175	442865	12936
675	501.43	489.64	1202.1	0.02047	0.6658	46501	440790	12830
665	499.81	487.71	1202.3	0.02043	0.6761	45827	438699	12722
655	498.17	485.77	1202.5	0.02039	0.6867	45153	436600	12613
645	496.52	483.81	1202.7	0.02036	0.6976	44478	434488	12501
635	494.84	481.83	1202.8	0.02032	0.7088	43804	432363	12387
625	493.14	479.83	1203.0	0.02028	0.7203	43130	430225	12271
615	491.43	477.82	1203.2	0.02024	0.7322	42456	428084	12153
605	489.69	475.78	1203.4	0.02021	0.7445	41782	425920	12033
595	487.93	473.72	1203.5	0.02017	0.7571	41107	423745	11911
585	486.15	471.64	1203.7	0.02013	0.7701	40433	421556	11788
575	484.35	469.53	1203.8	0.02009	0.7836	39759	419350	11662
565	482.52	467.40	1203.9	0.02005	0.7975	39085	417131	11534
555	480.66	465.25	1204.1	0.02002	0.8119	38410	414894	11404

545	478.78	463.07	1204.2	0.01998	0.8268	37736	412636	11273
535	476.88	460.87	1204.3	0.01994	0.8422	37062	410365	11139
525	474.95	458.64	1204.4	0.01990	0.8582	36388	408072	11004
515	472.99	456.39	1204.5	0.01986	0.8747	35714	405760	10866
505	471.01	454.11	1204.5	0.01982	0.8919	35039	403432	10727
495	468.99	451.80	1204.6	0.01978	0.9097	34365	401076	10585
485	466.95	449.45	1204.7	0.01974	0.9282	33691	398698	10442
475	464.87	447.08	1204.7	0.01970	0.9474	33017	396298	10297
465	462.76	444.68	1204.8	0.01966	0.9674	32342	393876	10150
455	460.61	442.24	1204.8	0.01962	0.9882	31668	391427	10001
445	458.43	439.76	1204.8	0.01958	1.0099	30994	388948	9850
435	456.21	437.25	1204.8	0.01954	1.0325	30320	386443	9697
425	453.96	434.69	1204.8	0.01950	1.0561	29646	383910	9542
415	451.67	432.11	1204.8	0.01946	1.0807	28971	381347	9385
405	449.33	429.48	1204.7	0.01942	1.1065	28297	378748	9226
395	446.95	426.80	1204.7	0.01938	1.1335	27623	376113	9065
385	444.53	424.09	1204.6	0.01934	1.1618	26949	373449	8902
375	442.06	421.32	1204.5	0.01929	1.1915	26274	370742	8738
365	439.53	418.51	1204.4	0.01925	1.2227	25600	367998	8571
355	436.96	415.64	1204.3	0.01921	1.2556	24926	365214	8402
345	434.33	412.72	1204.1	0.01916	1.2902	24252	362377	8231
335	431.65	409.74	1204.0	0.01912	1.3267	23578	359498	8058
325	428.91	406.71	1203.8	0.01908	1.3652	22903	356570	7883
315	426.10	403.60	1203.6	0.01903	1.4061	22229	353581	7706
305	423.22	400.43	1203.4	0.01899	1.4493	21555	350536	7527
295	420.27	397.20	1203.1	0.01894	1.4953	20881	347431	7346
285	417.26	393.89	1202.9	0.01889	1.5443	20206	344259	7162
275	414.16	390.50	1202.6	0.01885	1.5964	19532	341014	6977
265	410.97	387.01	1202.3	0.01880	1.6522	18858	337691	6789
255	407.70	383.45	1201.9	0.01875	1.7120	18184	334290	6598
245	404.33	379.79	1201.5	0.01870	1.7762	17510	330798	6406
235	400.87	376.03	1201.1	0.01865	1.8453	16835	327210	6211
225	397.28	372.15	1200.6	0.01860	1.9200	16161	323515	6014
215	393.59	368.16	1200.1	0.01855	2.0010	15487	319707	5814
205	389.76	364.04	1199.6	0.01849	2.0891	14813	315773	5612
200	387.79	361.93	1199.3	0.01847	2.1360	14476	313741	5510
195	385.79	359.78	1199.0	0.01844	2.1852	14138	311672	5408
190	383.76	357.60	1198.6	0.01841	2.2367	13801	309569	5304
185	381.67	355.38	1198.3	0.01839	2.2906	13464	307420	5200
180	379.56	353.10	1198.0	0.01836	2.3472	13127	305227	5096
175	377.40	350.80	1197.6	0.01833	2.4067	12790	302990	4990
170	375.19	348.44	1197.2	0.01830	2.4693	12453	300701	4884
165	372.94	346.04	1196.8	0.01827	2.5353	12116	298363	4778
160	370.64	343.59	1196.4	0.01824	2.6048	11779	295964	4670
155	368.28	341.09	1196.0	0.01821	2.6784	11442	293504	4562
150	365.87	338.52	1195.5	0.01818	2.7562	11104	290979	4453
145	363.40	335.91	1195.1	0.01815	2.8387	10767	288385	4343
140	360.87	333.23	1194.6	0.01812	2.9264	10430	285711	4233
135	358.27	330.48	1194.0	0.01809	3.0197	10093	282959	4122
130	355.61	327.67	1193.5	0.01806	3.1192	9756	280117	4009
125	352.87	324.78	1192.9	0.01802	3.2256	9419	277182	3896
120	350.06	321.82	1192.3	0.01799	3.3396	9082	274140	3783
115	347.15	318.76	1191.7	0.01796	3.4620	8745	270981	3668
110	344.17	315.63	1191.0	0.01792	3.5940	8408	267703	3552
105	341.08	312.39	1190.3	0.01789	3.7365	8070	264279	3436
100	337.89	309.05	1189.6	0.01785	3.8910	7733	260703	3318
95	334.59	305.59	1188.8	0.01781	4.0590	7396	256959	3200
90	331.16	302.02	1188.0	0.01778	4.2425	7059	253026	3080
85	327.60	298.31	1187.1	0.01774	4.4437	6722	248870	2959
80	323.90	294.46	1186.2	0.01770	4.6654	6385	244459	2837
75	320.04	290.44	1185.2	0.01766	4.9109	6048	239747	2715
70	316.01	286.26	1184.1	0.01761	5.1843	5711	234698	2591
65	311.78	281.88	1183.0	0.01757	5.4907	5374	229238	2465
60	307.33	277.28	1181.8	0.01753	5.8367	5036	223300	2339
55	302.64	272.44	1180.5	0.01748	6.2304	4699	216769	2211
50	297.67	267.31	1179.1	0.01743	6.6827	4362	209478	2082

prpv	erpv	dmg_srv	mf_sp	dt	t	qdh	esp	tsp

[psig]	[Btu]	[lbm]	[lbm]	[min.]	[min.]	[Btu/min.]	[Btu]	[°F]
1145	4.79e+008	0.0	4042129	0.0000	2.000	4381320	4.21e+008	136.2
1135	4.77e+008	1867.9	4043997	0.0241	2.024	4374717	4.24e+008	136.7
1125	4.75e+008	1867.4	4045865	0.0243	2.048	4368058	4.26e+008	137.2
1115	4.73e+008	1872.9	4047738	0.0246	2.073	4361320	4.28e+008	137.7
1105	4.71e+008	1878.3	4049616	0.0249	2.098	4354502	4.30e+008	138.2
1095	4.69e+008	1882.5	4051498	0.0252	2.123	4347608	4.32e+008	138.7
1085	4.66e+008	1884.3	4053383	0.0254	2.148	4340644	4.35e+008	139.2
1075	4.64e+008	1891.2	4055274	0.0257	2.174	4333591	4.37e+008	139.7
1065	4.62e+008	1892.1	4057166	0.0260	2.200	4326469	4.39e+008	140.2
1055	4.60e+008	1893.8	4059060	0.0263	2.226	4319274	4.41e+008	140.7
1045	4.58e+008	1904.5	4060964	0.0267	2.253	4311970	4.44e+008	141.3
1035	4.56e+008	1902.0	4062866	0.0269	2.280	4304606	4.46e+008	141.8
1025	4.54e+008	1915.9	4064782	0.0273	2.307	4297117	4.48e+008	142.3
1015	4.51e+008	1913.2	4066695	0.0276	2.335	4289566	4.51e+008	142.8
1005	4.49e+008	1920.4	4068616	0.0279	2.363	4281912	4.53e+008	143.3
995	4.47e+008	1921.6	4070537	0.0282	2.391	4274178	4.55e+008	143.8
985	4.45e+008	1931.8	4072469	0.0287	2.420	4266324	4.57e+008	144.3
975	4.43e+008	1935.5	4074404	0.0290	2.449	4258376	4.60e+008	144.8
965	4.40e+008	1942.2	4076347	0.0294	2.478	4250319	4.62e+008	145.3
955	4.38e+008	1946.6	4078293	0.0298	2.508	4242161	4.64e+008	145.9
945	4.36e+008	1951.7	4080245	0.0302	2.538	4233896	4.67e+008	146.4
935	4.34e+008	1959.8	4082205	0.0306	2.569	4225509	4.69e+008	146.9
925	4.32e+008	1957.6	4084162	0.0309	2.600	4217043	4.71e+008	147.4
915	4.29e+008	1972.5	4086135	0.0315	2.631	4208420	4.74e+008	147.9
905	4.27e+008	1978.8	4088114	0.0319	2.663	4199676	4.76e+008	148.5
895	4.25e+008	1985.1	4090099	0.0324	2.695	4190807	4.78e+008	149.0
885	4.23e+008	1991.3	4092090	0.0328	2.728	4181812	4.81e+008	149.5
875	4.20e+008	1991.7	4094082	0.0332	2.761	4172714	4.83e+008	150.0
865	4.18e+008	2003.5	4096085	0.0338	2.795	4163458	4.86e+008	150.6
855	4.16e+008	2010.5	4098096	0.0343	2.829	4154063	4.88e+008	151.1
845	4.14e+008	2018.1	4100114	0.0348	2.864	4144523	4.90e+008	151.6
835	4.11e+008	2029.6	4102143	0.0354	2.900	4134815	4.93e+008	152.2
825	4.09e+008	2030.8	4104174	0.0359	2.935	4124986	4.95e+008	152.7
815	4.07e+008	2048.4	4106223	0.0366	2.972	4114952	4.98e+008	153.2
805	4.04e+008	2044.1	4108267	0.0370	3.009	4104817	5.00e+008	153.8
795	4.02e+008	2064.3	4110331	0.0378	3.047	4094456	5.03e+008	154.3
785	4.00e+008	2065.8	4112397	0.0383	3.085	4083957	5.05e+008	154.8
775	3.98e+008	2080.1	4114477	0.0391	3.124	4073252	5.08e+008	155.4
765	3.95e+008	2081.6	4116559	0.0396	3.164	4062402	5.10e+008	155.9
755	3.93e+008	2100.0	4118659	0.0405	3.204	4051314	5.13e+008	156.5
745	3.90e+008	2108.0	4120767	0.0412	3.246	4040037	5.15e+008	157.0
735	3.88e+008	2112.9	4122880	0.0418	3.287	4028582	5.18e+008	157.6
725	3.86e+008	2124.3	4125004	0.0426	3.330	4016911	5.20e+008	158.1
715	3.83e+008	2143.1	4127147	0.0436	3.373	4004975	5.23e+008	158.7
705	3.81e+008	2144.9	4129292	0.0442	3.418	3992863	5.25e+008	159.2
695	3.78e+008	2162.5	4131454	0.0452	3.463	3980479	5.28e+008	159.8
685	3.76e+008	2174.3	4133629	0.0461	3.509	3967850	5.31e+008	160.4
675	3.74e+008	2180.6	4135809	0.0469	3.556	3955000	5.33e+008	160.9
665	3.71e+008	2198.6	4138008	0.0480	3.604	3941854	5.36e+008	161.5
655	3.69e+008	2209.2	4140217	0.0489	3.653	3928448	5.39e+008	162.1
645	3.66e+008	2223.2	4142440	0.0500	3.703	3914751	5.41e+008	162.7
635	3.64e+008	2239.1	4144679	0.0511	3.754	3900745	5.44e+008	163.2
625	3.61e+008	2253.9	4146933	0.0523	3.806	3886426	5.47e+008	163.8
615	3.59e+008	2259.6	4149193	0.0532	3.859	3871842	5.49e+008	164.4
605	3.56e+008	2282.9	4151476	0.0546	3.914	3856870	5.52e+008	165.0
595	3.54e+008	2297.2	4153773	0.0559	3.970	3841557	5.55e+008	165.6
585	3.51e+008	2312.6	4156086	0.0572	4.027	3829395	5.58e+008	166.2
575	3.48e+008	2332.1	4158418	0.0587	4.086	3820936	5.60e+008	166.8
565	3.46e+008	2346.4	4160764	0.0600	4.146	3812278	5.63e+008	167.4
555	3.43e+008	2366.6	4163131	0.0616	4.207	3803392	5.66e+008	168.0
545	3.41e+008	2390.1	4165521	0.0633	4.271	3794258	5.69e+008	168.6
535	3.38e+008	2404.2	4167925	0.0649	4.336	3784903	5.72e+008	169.2
525	3.35e+008	2428.9	4170354	0.0668	4.402	3775276	5.75e+008	169.8
515	3.33e+008	2448.9	4172803	0.0686	4.471	3765387	5.78e+008	170.5
505	3.30e+008	2468.4	4175271	0.0704	4.541	3755227	5.81e+008	171.1
495	3.27e+008	2497.3	4177768	0.0727	4.614	3744747	5.84e+008	171.7
485	3.24e+008	2520.7	4180289	0.0748	4.689	3733957	5.87e+008	172.4

475	3.22e+008	2545.1	4182834	0.0771	4.766	3722840	5.90e+008	173.0
465	3.19e+008	2569.0	4185403	0.0794	4.845	3711384	5.93e+008	173.7
455	3.16e+008	2597.9	4188001	0.0820	4.927	3699553	5.96e+008	174.3
445	3.13e+008	2630.4	4190632	0.0849	5.012	3687314	5.99e+008	175.0
435	3.10e+008	2658.0	4193290	0.0877	5.100	3674671	6.02e+008	175.7
425	3.07e+008	2688.3	4195978	0.0907	5.191	3661593	6.06e+008	176.4
415	3.04e+008	2719.4	4198697	0.0939	5.284	3648057	6.09e+008	177.0
405	3.01e+008	2757.7	4201455	0.0975	5.382	3634002	6.12e+008	177.7
395	2.98e+008	2796.4	4204251	0.1012	5.483	3619402	6.16e+008	178.4
385	2.95e+008	2826.6	4207078	0.1049	5.588	3604275	6.19e+008	179.2
375	2.92e+008	2872.0	4209950	0.1093	5.697	3588511	6.23e+008	179.9
365	2.89e+008	2910.5	4212861	0.1137	5.811	3572115	6.26e+008	180.6
355	2.86e+008	2953.3	4215814	0.1185	5.929	3555028	6.30e+008	181.3
345	2.83e+008	3007.0	4218821	0.1240	6.053	3539537	6.33e+008	182.1
335	2.80e+008	3052.4	4221873	0.1295	6.183	3526654	6.37e+008	182.9
325	2.76e+008	3103.3	4224977	0.1355	6.318	3513171	6.41e+008	183.6
315	2.73e+008	3165.3	4228142	0.1424	6.461	3499002	6.44e+008	184.4
305	2.70e+008	3224.1	4231366	0.1496	6.610	3484117	6.48e+008	185.2
295	2.66e+008	3287.2	4234653	0.1574	6.768	3468452	6.52e+008	186.0
285	2.63e+008	3354.9	4238008	0.1660	6.934	3451930	6.56e+008	186.9
275	2.59e+008	3430.8	4241439	0.1756	7.110	3434452	6.60e+008	187.7
265	2.56e+008	3510.5	4244949	0.1862	7.296	3415927	6.65e+008	188.6
255	2.52e+008	3591.9	4248541	0.1975	7.493	3396271	6.69e+008	189.5
245	2.48e+008	3684.3	4252226	0.2104	7.704	3375332	6.73e+008	190.4
235	2.44e+008	3782.7	4256008	0.2247	7.928	3352973	6.78e+008	191.3
225	2.40e+008	3891.5	4259900	0.2408	8.169	3332670	6.83e+008	192.2
215	2.36e+008	4008.3	4263908	0.2588	8.428	3312514	6.87e+008	193.2
205	2.32e+008	4136.1	4268044	0.2792	8.707	3290768	6.92e+008	194.2
200	2.30e+008	2134.1	4270178	0.1474	8.855	3279287	6.95e+008	194.7
195	2.28e+008	2171.4	4272350	0.1536	9.008	3267327	6.98e+008	195.3
190	2.26e+008	2205.8	4274556	0.1598	9.168	3254880	7.00e+008	195.8
185	2.24e+008	2253.6	4276809	0.1674	9.335	3241845	7.03e+008	196.4
180	2.22e+008	2297.7	4279107	0.1750	9.510	3228214	7.06e+008	196.9
175	2.19e+008	2341.9	4281449	0.1831	9.694	3213955	7.08e+008	197.5
170	2.17e+008	2395.7	4283844	0.1924	9.886	3198972	7.11e+008	198.1
165	2.15e+008	2444.7	4286289	0.2018	10.088	3184953	7.14e+008	198.6
160	2.12e+008	2505.8	4288795	0.2127	10.300	3172496	7.17e+008	199.2
155	2.10e+008	2568.8	4291364	0.2245	10.525	3159350	7.20e+008	199.9
150	2.08e+008	2633.2	4293997	0.2371	10.762	3145466	7.23e+008	200.5
145	2.05e+008	2704.5	4296701	0.2512	11.013	3130759	7.27e+008	201.1
140	2.03e+008	2784.6	4299486	0.2670	11.280	3115127	7.30e+008	201.8
135	2.00e+008	2863.4	4302349	0.2837	11.564	3098516	7.33e+008	202.5
130	1.98e+008	2953.5	4305303	0.3027	11.867	3080790	7.37e+008	203.2
125	1.95e+008	3048.4	4308351	0.3236	12.190	3061840	7.41e+008	203.9
120	1.92e+008	3155.5	4311507	0.3475	12.538	3041496	7.44e+008	204.7
115	1.89e+008	3274.0	4314781	0.3744	12.912	3019574	7.48e+008	205.4
110	1.86e+008	3393.6	4318174	0.4036	13.316	2995940	7.52e+008	206.2
105	1.83e+008	3540.2	4321714	0.4387	13.754	2970256	7.57e+008	207.1
100	1.80e+008	3693.7	4325408	0.4776	14.232	2942289	7.61e+008	207.9
95	1.77e+008	3862.4	4329271	0.5222	14.754	2911712	7.66e+008	208.8
90	1.74e+008	4052.4	4333323	0.5741	15.328	2883403	7.70e+008	209.8
85	1.71e+008	4276.6	4337600	0.6362	15.965	2856427	7.75e+008	210.8
80	1.67e+008	4533.1	4342133	0.7100	16.675	2826324	7.81e+008	211.8
75	1.64e+008	4835.3	4346968	0.7995	17.474	2792425	7.87e+008	212.9
70	1.60e+008	5172.7	4352141	0.9058	18.380	2754019	7.93e+008	214.1
65	1.56e+008	5585.0	4357726	1.0393	19.419	2709951	7.99e+008	215.4
60	1.52e+008	6064.3	4363790	1.2041	20.623	2664651	8.06e+008	216.8
55	1.48e+008	6659.2	4370449	1.4171	22.040	2617647	8.14e+008	218.3
50	1.43e+008	7420.5	4377870	1.7011	23.741	2561222	8.23e+008	220.0

torus water level = 10.00 ft

prpv [psig]	trpv [°F]	hf_rpv [Btu/lbm]	hg_rpv [Btu/lbm]	vf_rpv [ft3/lbm]	vg_rpv [ft3/lbm]	wsrv [lbm/min.]	mf_rpv [lbm]	mg_rpv [lbm]
1145	562.88	566.16	1186.6	0.02217	0.3771	78190	532000	15300
1135	561.79	564.74	1187.0	0.02213	0.3808	77515	530124	15309
1125	560.70	563.31	1187.4	0.02210	0.3847	76841	528251	15314
1115	559.60	561.87	1187.8	0.02206	0.3886	76167	526376	15316

1105	558.49	560.42	1188.3	0.02202	0.3926	75493	524498	15315
1095	557.37	558.97	1188.7	0.02199	0.3966	74819	522619	15312
1085	556.24	557.51	1189.1	0.02195	0.4007	74144	520742	15305
1075	555.11	556.04	1189.5	0.02192	0.4049	73470	518860	15295
1065	553.97	554.57	1189.9	0.02188	0.4091	72796	516980	15283
1055	552.82	553.09	1190.3	0.02184	0.4135	72122	515102	15268
1045	551.66	551.60	1190.7	0.02181	0.4179	71447	513216	15250
1035	550.50	550.11	1191.1	0.02177	0.4224	70773	511335	15229
1025	549.32	548.60	1191.4	0.02173	0.4269	70099	509442	15205
1015	548.14	547.09	1191.8	0.02170	0.4316	69425	507556	15179
1005	546.95	545.57	1192.2	0.02166	0.4363	68751	505664	15150
995	545.75	544.04	1192.6	0.02163	0.4412	68076	503775	15118
985	544.54	542.51	1192.9	0.02159	0.4461	67402	501877	15083
975	543.32	540.96	1193.3	0.02155	0.4511	66728	499979	15046
965	542.10	539.40	1193.7	0.02152	0.4562	66054	498076	15007
955	540.86	537.85	1194.0	0.02148	0.4615	65379	496171	14965
945	539.61	536.27	1194.4	0.02145	0.4668	64705	494264	14920
935	538.35	534.70	1194.7	0.02141	0.4722	64031	492352	14873
925	537.09	533.11	1195.1	0.02137	0.4778	63357	490444	14823
915	535.81	531.51	1195.4	0.02134	0.4834	62683	488524	14771
905	534.52	529.90	1195.7	0.02130	0.4892	62008	486600	14716
895	533.22	528.28	1196.1	0.02127	0.4951	61334	484671	14659
885	531.91	526.65	1196.4	0.02123	0.5011	60660	482740	14600
875	530.59	525.02	1196.7	0.02119	0.5072	59986	480810	14538
865	529.26	523.37	1197.0	0.02116	0.5135	59311	478871	14474
855	527.92	521.71	1197.3	0.02112	0.5199	58637	476927	14407
845	526.56	520.04	1197.7	0.02109	0.5265	57963	474977	14338
835	525.20	518.35	1198.0	0.02105	0.5332	57289	473019	14267
825	523.82	516.66	1198.3	0.02101	0.5401	56614	471061	14194
815	522.42	514.95	1198.5	0.02098	0.5471	55940	469088	14118
805	521.02	513.24	1198.8	0.02094	0.5543	55266	467123	14040
795	519.60	511.50	1199.1	0.02090	0.5617	54592	465138	13960
785	518.16	509.76	1199.4	0.02087	0.5692	53918	463155	13878
775	516.72	508.00	1199.7	0.02083	0.5769	53243	461159	13793
765	515.26	506.24	1199.9	0.02080	0.5848	52569	459165	13706
755	513.78	504.45	1200.2	0.02076	0.5929	51895	457153	13618
745	512.30	502.65	1200.4	0.02072	0.6012	51221	455136	13527
735	510.80	500.84	1200.7	0.02069	0.6097	50546	453117	13433
725	509.28	499.01	1200.9	0.02065	0.6185	49872	451088	13338
715	507.74	497.17	1201.2	0.02061	0.6275	49198	449042	13241
705	506.19	495.31	1201.4	0.02058	0.6367	48524	446997	13141
695	504.62	493.43	1201.6	0.02054	0.6461	47850	444936	13039
685	503.03	491.54	1201.8	0.02050	0.6559	47175	442865	12936
675	501.43	489.64	1202.1	0.02047	0.6658	46501	440790	12830
665	499.81	487.71	1202.3	0.02043	0.6761	45827	438699	12722
655	498.17	485.77	1202.5	0.02039	0.6867	45153	436600	12613
645	496.52	483.81	1202.7	0.02036	0.6976	44478	434488	12501
635	494.84	481.83	1202.8	0.02032	0.7088	43804	432363	12387
625	493.14	479.83	1203.0	0.02028	0.7203	43130	430225	12271
615	491.43	477.82	1203.2	0.02024	0.7322	42456	428084	12153
605	489.69	475.78	1203.4	0.02021	0.7445	41782	425920	12033
595	487.93	473.72	1203.5	0.02017	0.7571	41107	423745	11911
585	486.15	471.64	1203.7	0.02013	0.7701	40433	421556	11788
575	484.35	469.53	1203.8	0.02009	0.7836	39759	419350	11662
565	482.52	467.40	1203.9	0.02005	0.7975	39085	417131	11534
555	480.66	465.25	1204.1	0.02002	0.8119	38410	414894	11404
545	478.78	463.07	1204.2	0.01998	0.8268	37736	412636	11273
535	476.88	460.87	1204.3	0.01994	0.8422	37062	410365	11139
525	474.95	458.64	1204.4	0.01990	0.8582	36388	408072	11004
515	472.99	456.39	1204.5	0.01986	0.8747	35714	405760	10866
505	471.01	454.11	1204.5	0.01982	0.8919	35039	403432	10727
495	468.99	451.80	1204.6	0.01978	0.9097	34365	401076	10585
485	466.95	449.45	1204.7	0.01974	0.9282	33691	398698	10442
475	464.87	447.08	1204.7	0.01970	0.9474	33017	396298	10297
465	462.76	444.68	1204.8	0.01966	0.9674	32342	393876	10150
455	460.61	442.24	1204.8	0.01962	0.9882	31668	391427	10001
445	458.43	439.76	1204.8	0.01958	1.0099	30994	388948	9850
435	456.21	437.25	1204.8	0.01954	1.0325	30320	386443	9697

425	453.96	434.69	1204.8	0.01950	1.0561	29646	383910	9542
415	451.67	432.11	1204.8	0.01946	1.0807	28971	381347	9385
405	449.33	429.48	1204.7	0.01942	1.1065	28297	378748	9226
395	446.95	426.80	1204.7	0.01938	1.1335	27623	376113	9065
385	444.53	424.09	1204.6	0.01934	1.1618	26949	373449	8902
375	442.06	421.32	1204.5	0.01929	1.1915	26274	370742	8738
365	439.53	418.51	1204.4	0.01925	1.2227	25600	367998	8571
355	436.96	415.64	1204.3	0.01921	1.2556	24926	365214	8402
345	434.33	412.72	1204.1	0.01916	1.2902	24252	362377	8231
335	431.65	409.74	1204.0	0.01912	1.3267	23578	359498	8058
325	428.91	406.71	1203.8	0.01908	1.3652	22903	356570	7883
315	426.10	403.60	1203.6	0.01903	1.4061	22229	353581	7706
305	423.22	400.43	1203.4	0.01899	1.4493	21555	350536	7527
295	420.27	397.20	1203.1	0.01894	1.4953	20881	347431	7346
285	417.26	393.89	1202.9	0.01889	1.5443	20206	344259	7162
275	414.16	390.50	1202.6	0.01885	1.5964	19532	341014	6977
265	410.97	387.01	1202.3	0.01880	1.6522	18858	337691	6789
255	407.70	383.45	1201.9	0.01875	1.7120	18184	334290	6598
245	404.33	379.79	1201.5	0.01870	1.7762	17510	330798	6406
235	400.87	376.03	1201.1	0.01865	1.8453	16835	327210	6211
225	397.28	372.15	1200.6	0.01860	1.9200	16161	323515	6014
215	393.59	368.16	1200.1	0.01855	2.0010	15487	319707	5814
205	389.76	364.04	1199.6	0.01849	2.0891	14813	315773	5612
200	387.79	361.93	1199.3	0.01847	2.1360	14476	313741	5510
195	385.79	359.78	1199.0	0.01844	2.1852	14138	311672	5408
190	383.76	357.60	1198.6	0.01841	2.2367	13801	309569	5304
185	381.67	355.38	1198.3	0.01839	2.2906	13464	307420	5200
180	379.56	353.10	1198.0	0.01836	2.3472	13127	305227	5096
175	377.40	350.80	1197.6	0.01833	2.4067	12790	302990	4990
170	375.19	348.44	1197.2	0.01830	2.4693	12453	300701	4884
165	372.94	346.04	1196.8	0.01827	2.5353	12116	298363	4778
160	370.64	343.59	1196.4	0.01824	2.6048	11779	295964	4670
155	368.28	341.09	1196.0	0.01821	2.6784	11442	293504	4562
150	365.87	338.52	1195.5	0.01818	2.7562	11104	290979	4453
145	363.40	335.91	1195.1	0.01815	2.8387	10767	288385	4343
140	360.87	333.23	1194.6	0.01812	2.9264	10430	285711	4233
135	358.27	330.48	1194.0	0.01809	3.0197	10093	282959	4122
130	355.61	327.67	1193.5	0.01806	3.1192	9756	280117	4009
125	352.87	324.78	1192.9	0.01802	3.2256	9419	277182	3896
120	350.06	321.82	1192.3	0.01799	3.3396	9082	274140	3783
115	347.15	318.76	1191.7	0.01796	3.4620	8745	270981	3668
110	344.17	315.63	1191.0	0.01792	3.5940	8408	267703	3552
105	341.08	312.39	1190.3	0.01789	3.7365	8070	264279	3436
100	337.89	309.05	1189.6	0.01785	3.8910	7733	260703	3318
95	334.59	305.59	1188.8	0.01781	4.0590	7396	256959	3200
90	331.16	302.02	1188.0	0.01778	4.2425	7059	253026	3080
85	327.60	298.31	1187.1	0.01774	4.4437	6722	248870	2959
80	323.90	294.46	1186.2	0.01770	4.6654	6385	244459	2837
75	320.04	290.44	1185.2	0.01766	4.9109	6048	239747	2715
70	316.01	286.26	1184.1	0.01761	5.1843	5711	234698	2591
65	311.78	281.88	1183.0	0.01757	5.4907	5374	229238	2465
60	307.33	277.28	1181.8	0.01753	5.8367	5036	223300	2339
55	302.64	272.44	1180.5	0.01748	6.2304	4699	216769	2211
50	297.67	267.31	1179.1	0.01743	6.6827	4362	209478	2082

prpv [psig]	erpv [Btu]	dmg_srv [lbm]	mf_sp [lbm]	dt [min.]	t [min.]	qdh [Btu/min.]	esp [Btu]	tsp [°F]
1145	4.79e+008	0.0	4281852	0.0000	2.000	4381320	4.66e+008	140.9
1135	4.77e+008	1867.9	4283720	0.0241	2.024	4374717	4.69e+008	141.4
1125	4.75e+008	1867.4	4285588	0.0243	2.048	4368058	4.71e+008	141.9
1115	4.73e+008	1872.9	4287460	0.0246	2.073	4361320	4.73e+008	142.3
1105	4.71e+008	1878.3	4289339	0.0249	2.098	4354502	4.75e+008	142.8
1095	4.69e+008	1882.5	4291221	0.0252	2.123	4347608	4.77e+008	143.3
1085	4.66e+008	1884.3	4293106	0.0254	2.148	4340644	4.80e+008	143.7
1075	4.64e+008	1891.2	4294997	0.0257	2.174	4333591	4.82e+008	144.2
1065	4.62e+008	1892.1	4296889	0.0260	2.200	4326469	4.84e+008	144.7
1055	4.60e+008	1893.8	4298783	0.0263	2.226	4319274	4.86e+008	145.2
1045	4.58e+008	1904.5	4300687	0.0267	2.253	4311970	4.89e+008	145.6

1035	4.56e+008	1902.0	4302589	0.0269	2.280	4304606	4.91e+008	146.1
1025	4.54e+008	1915.9	4304505	0.0273	2.307	4297117	4.93e+008	146.6
1015	4.51e+008	1913.2	4306418	0.0276	2.335	4289566	4.96e+008	147.1
1005	4.49e+008	1920.4	4308338	0.0279	2.363	4281912	4.98e+008	147.6
995	4.47e+008	1921.6	4310260	0.0282	2.391	4274178	5.00e+008	148.0
985	4.45e+008	1931.8	4312192	0.0287	2.420	4266324	5.02e+008	148.5
975	4.43e+008	1935.5	4314127	0.0290	2.449	4258376	5.05e+008	149.0
965	4.40e+008	1942.2	4316070	0.0294	2.478	4250319	5.07e+008	149.5
955	4.38e+008	1946.6	4318016	0.0298	2.508	4242161	5.09e+008	150.0
945	4.36e+008	1951.7	4319968	0.0302	2.538	4233896	5.12e+008	150.5
935	4.34e+008	1959.8	4321928	0.0306	2.569	4225509	5.14e+008	150.9
925	4.32e+008	1957.6	4323885	0.0309	2.600	4217043	5.16e+008	151.4
915	4.29e+008	1972.5	4325858	0.0315	2.631	4208420	5.19e+008	151.9
905	4.27e+008	1978.8	4327836	0.0319	2.663	4199676	5.21e+008	152.4
895	4.25e+008	1985.1	4329822	0.0324	2.695	4190807	5.24e+008	152.9
885	4.23e+008	1991.3	4331813	0.0328	2.728	4181812	5.26e+008	153.4
875	4.20e+008	1991.7	4333805	0.0332	2.761	4172714	5.28e+008	153.9
865	4.18e+008	2003.5	4335808	0.0338	2.795	4163458	5.31e+008	154.4
855	4.16e+008	2010.5	4337819	0.0343	2.829	4154063	5.33e+008	154.9
845	4.14e+008	2018.1	4339837	0.0348	2.864	4144523	5.36e+008	155.4
835	4.11e+008	2029.6	4341866	0.0354	2.900	4134815	5.38e+008	155.9
825	4.09e+008	2030.8	4343897	0.0359	2.935	4124986	5.40e+008	156.4
815	4.07e+008	2048.4	4345946	0.0366	2.972	4114952	5.43e+008	156.9
805	4.04e+008	2044.1	4347990	0.0370	3.009	4104817	5.45e+008	157.4
795	4.02e+008	2064.3	4350054	0.0378	3.047	4094456	5.48e+008	157.9
785	4.00e+008	2065.8	4352120	0.0383	3.085	4083957	5.50e+008	158.4
775	3.98e+008	2080.1	4354200	0.0391	3.124	4073252	5.53e+008	158.9
765	3.95e+008	2081.6	4356281	0.0396	3.164	4062402	5.55e+008	159.5
755	3.93e+008	2100.0	4358381	0.0405	3.204	4051314	5.58e+008	160.0
745	3.90e+008	2108.0	4360489	0.0412	3.246	4040037	5.60e+008	160.5
735	3.88e+008	2112.9	4362602	0.0418	3.287	4028582	5.63e+008	161.0
725	3.86e+008	2124.3	4364727	0.0426	3.330	4016911	5.65e+008	161.5
715	3.83e+008	2143.1	4366870	0.0436	3.373	4004975	5.68e+008	162.1
705	3.81e+008	2144.9	4369015	0.0442	3.418	3992863	5.71e+008	162.6
695	3.78e+008	2162.5	4371177	0.0452	3.463	3980479	5.73e+008	163.1
685	3.76e+008	2174.3	4373351	0.0461	3.509	3967850	5.76e+008	163.6
675	3.74e+008	2180.6	4375532	0.0469	3.556	3955000	5.78e+008	164.2
665	3.71e+008	2198.6	4377731	0.0480	3.604	3941854	5.81e+008	164.7
655	3.69e+008	2209.2	4379940	0.0489	3.653	3928448	5.84e+008	165.3
645	3.66e+008	2223.2	4382163	0.0500	3.703	3914751	5.86e+008	165.8
635	3.64e+008	2239.1	4384402	0.0511	3.754	3900745	5.89e+008	166.3
625	3.61e+008	2253.9	4386656	0.0523	3.806	3886426	5.92e+008	166.9
615	3.59e+008	2259.6	4388916	0.0532	3.859	3871842	5.94e+008	167.4
605	3.56e+008	2282.9	4391199	0.0546	3.914	3856870	5.97e+008	168.0
595	3.54e+008	2297.2	4393496	0.0559	3.970	3841557	6.00e+008	168.6
585	3.51e+008	2312.6	4395808	0.0572	4.027	3829395	6.03e+008	169.1
575	3.48e+008	2332.1	4398140	0.0587	4.086	3820936	6.06e+008	169.7
565	3.46e+008	2346.4	4400487	0.0600	4.146	3812278	6.08e+008	170.3
555	3.43e+008	2366.6	4402853	0.0616	4.207	3803392	6.11e+008	170.8
545	3.41e+008	2390.1	4405244	0.0633	4.271	3794258	6.14e+008	171.4
535	3.38e+008	2404.2	4407648	0.0649	4.336	3784903	6.17e+008	172.0
525	3.35e+008	2428.9	4410077	0.0668	4.402	3775276	6.20e+008	172.6
515	3.33e+008	2448.9	4412526	0.0686	4.471	3765387	6.23e+008	173.2
505	3.30e+008	2468.4	4414994	0.0704	4.541	3755227	6.26e+008	173.8
495	3.27e+008	2497.3	4417491	0.0727	4.614	3744747	6.29e+008	174.4
485	3.24e+008	2520.7	4420012	0.0748	4.689	3733957	6.32e+008	175.0
475	3.22e+008	2545.1	4422557	0.0771	4.766	3722840	6.35e+008	175.6
465	3.19e+008	2569.0	4425126	0.0794	4.845	3711384	6.38e+008	176.2
455	3.16e+008	2597.9	4427724	0.0820	4.927	3699553	6.41e+008	176.8
445	3.13e+008	2630.4	4430354	0.0849	5.012	3687314	6.44e+008	177.4
435	3.10e+008	2658.0	4433012	0.0877	5.100	3674671	6.48e+008	178.1
425	3.07e+008	2688.3	4435701	0.0907	5.191	3661593	6.51e+008	178.7
415	3.04e+008	2719.4	4438420	0.0939	5.284	3648057	6.54e+008	179.4
405	3.01e+008	2757.7	4441178	0.0975	5.382	3634002	6.57e+008	180.0
395	2.98e+008	2796.4	4443974	0.1012	5.483	3619402	6.61e+008	180.7
385	2.95e+008	2826.6	4446801	0.1049	5.588	3604275	6.64e+008	181.4
375	2.92e+008	2872.0	4449673	0.1093	5.697	3588511	6.68e+008	182.0
365	2.89e+008	2910.5	4452583	0.1137	5.811	3572115	6.71e+008	182.7

355	2.86e+008	2953.3	4455537	0.1185	5.929	3555028	6.75e+008	183.4
345	2.83e+008	3007.0	4458544	0.1240	6.053	3539537	6.78e+008	184.1
335	2.80e+008	3052.4	4461596	0.1295	6.183	3526654	6.82e+008	184.9
325	2.76e+008	3103.3	4464699	0.1355	6.318	3513171	6.86e+008	185.6
315	2.73e+008	3165.3	4467865	0.1424	6.461	3499002	6.90e+008	186.3
305	2.70e+008	3224.1	4471089	0.1496	6.610	3484117	6.93e+008	187.1
295	2.66e+008	3287.2	4474376	0.1574	6.768	3468452	6.97e+008	187.9
285	2.63e+008	3354.9	4477731	0.1660	6.934	3451930	7.01e+008	188.6
275	2.59e+008	3430.8	4481162	0.1756	7.110	3434452	7.06e+008	189.4
265	2.56e+008	3510.5	4484672	0.1862	7.296	3415927	7.10e+008	190.3
255	2.52e+008	3591.9	4488264	0.1975	7.493	3396271	7.14e+008	191.1
245	2.48e+008	3684.3	4491948	0.2104	7.704	3375332	7.18e+008	192.0
235	2.44e+008	3782.7	4495731	0.2247	7.928	3352973	7.23e+008	192.8
225	2.40e+008	3891.5	4499623	0.2408	8.169	3332670	7.28e+008	193.7
215	2.36e+008	4008.3	4503631	0.2588	8.428	3312514	7.33e+008	194.7
205	2.32e+008	4136.1	4507767	0.2792	8.707	3290768	7.37e+008	195.6
200	2.30e+008	2134.1	4509901	0.1474	8.855	3279287	7.40e+008	196.1
195	2.28e+008	2171.4	4512073	0.1536	9.008	3267327	7.43e+008	196.6
190	2.26e+008	2205.8	4514278	0.1598	9.168	3254880	7.45e+008	197.1
185	2.24e+008	2253.6	4516532	0.1674	9.335	3241845	7.48e+008	197.6
180	2.22e+008	2297.7	4518830	0.1750	9.510	3228214	7.51e+008	198.1
175	2.19e+008	2341.9	4521172	0.1831	9.694	3213955	7.54e+008	198.7
170	2.17e+008	2395.7	4523567	0.1924	9.886	3198972	7.56e+008	199.2
165	2.15e+008	2444.7	4526012	0.2018	10.088	3184953	7.59e+008	199.8
160	2.12e+008	2505.8	4528518	0.2127	10.300	3172496	7.62e+008	200.3
155	2.10e+008	2568.8	4531087	0.2245	10.525	3159350	7.65e+008	200.9
150	2.08e+008	2633.2	4533720	0.2371	10.762	3145466	7.69e+008	201.5
145	2.05e+008	2704.5	4536424	0.2512	11.013	3130759	7.72e+008	202.1
140	2.03e+008	2784.6	4539209	0.2670	11.280	3115127	7.75e+008	202.8
135	2.00e+008	2863.4	4542072	0.2837	11.564	3098516	7.79e+008	203.4
130	1.98e+008	2953.5	4545026	0.3027	11.867	3080790	7.82e+008	204.1
125	1.95e+008	3048.4	4548074	0.3236	12.190	3061840	7.86e+008	204.8
120	1.92e+008	3155.5	4551230	0.3475	12.538	3041496	7.89e+008	205.5
115	1.89e+008	3274.0	4554504	0.3744	12.912	3019574	7.93e+008	206.2
110	1.86e+008	3393.6	4557897	0.4036	13.316	2995940	7.97e+008	206.9
105	1.83e+008	3540.2	4561437	0.4387	13.754	2970256	8.02e+008	207.7
100	1.80e+008	3693.7	4565131	0.4776	14.232	2942289	8.06e+008	208.6
95	1.77e+008	3862.4	4568993	0.5222	14.754	2911712	8.11e+008	209.4
90	1.74e+008	4052.4	4573046	0.5741	15.328	2883403	8.15e+008	210.3
85	1.71e+008	4276.6	4577323	0.6362	15.965	2856427	8.20e+008	211.3
80	1.67e+008	4533.1	4581856	0.7100	16.675	2826324	8.26e+008	212.2
75	1.64e+008	4835.3	4586691	0.7995	17.474	2792425	8.32e+008	213.3
70	1.60e+008	5172.7	4591864	0.9058	18.380	2754019	8.38e+008	214.4
65	1.56e+008	5585.0	4597449	1.0393	19.419	2709951	8.44e+008	215.7
60	1.52e+008	6064.3	4603513	1.2041	20.623	2664651	8.51e+008	217.0
55	1.48e+008	6659.2	4610172	1.4171	22.040	2617647	8.59e+008	218.4
50	1.43e+008	7420.5	4617593	1.7011	23.741	2561222	8.68e+008	220.0

torus water level = 11.00 ft

prpv [psig]	trpv [°F]	hf_rpv [Btu/lbm]	hg_rpv [Btu/lbm]	vf_rpv [ft3/lbm]	vg_rpv [ft3/lbm]	wsrv [lbm/min.]	mf_rpv [lbm]	mg_rpv [lbm]
1145	562.88	566.16	1186.6	0.02217	0.3771	78190	532000	15300
1135	561.79	564.74	1187.0	0.02213	0.3808	77515	530124	15309
1125	560.70	563.31	1187.4	0.02210	0.3847	76841	528251	15314
1115	559.60	561.87	1187.8	0.02206	0.3886	76167	526376	15316
1105	558.49	560.42	1188.3	0.02202	0.3926	75493	524498	15315
1095	557.37	558.97	1188.7	0.02199	0.3966	74819	522619	15312
1085	556.24	557.51	1189.1	0.02195	0.4007	74144	520742	15305
1075	555.11	556.04	1189.5	0.02192	0.4049	73470	518860	15295
1065	553.97	554.57	1189.9	0.02188	0.4091	72796	516980	15283
1055	552.82	553.09	1190.3	0.02184	0.4135	72122	515102	15268
1045	551.66	551.60	1190.7	0.02181	0.4179	71447	513216	15250
1035	550.50	550.11	1191.1	0.02177	0.4224	70773	511335	15229
1025	549.32	548.60	1191.4	0.02173	0.4269	70099	509442	15205
1015	548.14	547.09	1191.8	0.02170	0.4316	69425	507556	15179
1005	546.95	545.57	1192.2	0.02166	0.4363	68751	505664	15150
995	545.75	544.04	1192.6	0.02163	0.4412	68076	503775	15118

985	544.54	542.51	1192.9	0.02159	0.4461	67402	501877	15083
975	543.32	540.96	1193.3	0.02155	0.4511	66728	499979	15046
965	542.10	539.40	1193.7	0.02152	0.4562	66054	498076	15007
955	540.86	537.85	1194.0	0.02148	0.4615	65379	496171	14965
945	539.61	536.27	1194.4	0.02145	0.4668	64705	494264	14920
935	538.35	534.70	1194.7	0.02141	0.4722	64031	492352	14873
925	537.09	533.11	1195.1	0.02137	0.4778	63357	490444	14823
915	535.81	531.51	1195.4	0.02134	0.4834	62683	488524	14771
905	534.52	529.90	1195.7	0.02130	0.4892	62008	486600	14716
895	533.22	528.28	1196.1	0.02127	0.4951	61334	484671	14659
885	531.91	526.65	1196.4	0.02123	0.5011	60660	482740	14600
875	530.59	525.02	1196.7	0.02119	0.5072	59986	480810	14538
865	529.26	523.37	1197.0	0.02116	0.5135	59311	478871	14474
855	527.92	521.71	1197.3	0.02112	0.5199	58637	476927	14407
845	526.56	520.04	1197.7	0.02109	0.5265	57963	474977	14338
835	525.20	518.35	1198.0	0.02105	0.5332	57289	473019	14267
825	523.82	516.66	1198.3	0.02101	0.5401	56614	471061	14194
815	522.42	514.95	1198.5	0.02098	0.5471	55940	469088	14118
805	521.02	513.24	1198.8	0.02094	0.5543	55266	467123	14040
795	519.60	511.50	1199.1	0.02090	0.5617	54592	465138	13960
785	518.16	509.76	1199.4	0.02087	0.5692	53918	463155	13878
775	516.72	508.00	1199.7	0.02083	0.5769	53243	461159	13793
765	515.26	506.24	1199.9	0.02080	0.5848	52569	459165	13706
755	513.78	504.45	1200.2	0.02076	0.5929	51895	457153	13618
745	512.30	502.65	1200.4	0.02072	0.6012	51221	455136	13527
735	510.80	500.84	1200.7	0.02069	0.6097	50546	453117	13433
725	509.28	499.01	1200.9	0.02065	0.6185	49872	451088	13338
715	507.74	497.17	1201.2	0.02061	0.6275	49198	449042	13241
705	506.19	495.31	1201.4	0.02058	0.6367	48524	446997	13141
695	504.62	493.43	1201.6	0.02054	0.6461	47850	444936	13039
685	503.03	491.54	1201.8	0.02050	0.6559	47175	442865	12936
675	501.43	489.64	1202.1	0.02047	0.6658	46501	440790	12830
665	499.81	487.71	1202.3	0.02043	0.6761	45827	438699	12722
655	498.17	485.77	1202.5	0.02039	0.6867	45153	436600	12613
645	496.52	483.81	1202.7	0.02036	0.6976	44478	434488	12501
635	494.84	481.83	1202.8	0.02032	0.7088	43804	432363	12387
625	493.14	479.83	1203.0	0.02028	0.7203	43130	430225	12271
615	491.43	477.82	1203.2	0.02024	0.7322	42456	428084	12153
605	489.69	475.78	1203.4	0.02021	0.7445	41782	425920	12033
595	487.93	473.72	1203.5	0.02017	0.7571	41107	423745	11911
585	486.15	471.64	1203.7	0.02013	0.7701	40433	421556	11788
575	484.35	469.53	1203.8	0.02009	0.7836	39759	419350	11662
565	482.52	467.40	1203.9	0.02005	0.7975	39085	417131	11534
555	480.66	465.25	1204.1	0.02002	0.8119	38410	414894	11404
545	478.78	463.07	1204.2	0.01998	0.8268	37736	412636	11273
535	476.88	460.87	1204.3	0.01994	0.8422	37062	410365	11139
525	474.95	458.64	1204.4	0.01990	0.8582	36388	408072	11004
515	472.99	456.39	1204.5	0.01986	0.8747	35714	405760	10866
505	471.01	454.11	1204.5	0.01982	0.8919	35039	403432	10727
495	468.99	451.80	1204.6	0.01978	0.9097	34365	401076	10585
485	466.95	449.45	1204.7	0.01974	0.9282	33691	398698	10442
475	464.87	447.08	1204.7	0.01970	0.9474	33017	396298	10297
465	462.76	444.68	1204.8	0.01966	0.9674	32342	393876	10150
455	460.61	442.24	1204.8	0.01962	0.9882	31668	391427	10001
445	458.43	439.76	1204.8	0.01958	1.0099	30994	388948	9850
435	456.21	437.25	1204.8	0.01954	1.0325	30320	386443	9697
425	453.96	434.69	1204.8	0.01950	1.0561	29646	383910	9542
415	451.67	432.11	1204.8	0.01946	1.0807	28971	381347	9385
405	449.33	429.48	1204.7	0.01942	1.1065	28297	378748	9226
395	446.95	426.80	1204.7	0.01938	1.1335	27623	376113	9065
385	444.53	424.09	1204.6	0.01934	1.1618	26949	373449	8902
375	442.06	421.32	1204.5	0.01929	1.1915	26274	370742	8738
365	439.53	418.51	1204.4	0.01925	1.2227	25600	367998	8571
355	436.96	415.64	1204.3	0.01921	1.2556	24926	365214	8402
345	434.33	412.72	1204.1	0.01916	1.2902	24252	362377	8231
335	431.65	409.74	1204.0	0.01912	1.3267	23578	359498	8058
325	428.91	406.71	1203.8	0.01908	1.3652	22903	356570	7883
315	426.10	403.60	1203.6	0.01903	1.4061	22229	353581	7706

305	423.22	400.43	1203.4	0.01899	1.4493	21555	350536	7527
295	420.27	397.20	1203.1	0.01894	1.4953	20881	347431	7346
285	417.26	393.89	1202.9	0.01889	1.5443	20206	344259	7162
275	414.16	390.50	1202.6	0.01885	1.5964	19532	341014	6977
265	410.97	387.01	1202.3	0.01880	1.6522	18858	337691	6789
255	407.70	383.45	1201.9	0.01875	1.7120	18184	334290	6598
245	404.33	379.79	1201.5	0.01870	1.7762	17510	330798	6406
235	400.87	376.03	1201.1	0.01865	1.8453	16835	327210	6211
225	397.28	372.15	1200.6	0.01860	1.9200	16161	323515	6014
215	393.59	368.16	1200.1	0.01855	2.0010	15487	319707	5814
205	389.76	364.04	1199.6	0.01849	2.0891	14813	315773	5612
200	387.79	361.93	1199.3	0.01847	2.1360	14476	313741	5510
195	385.79	359.78	1199.0	0.01844	2.1852	14138	311672	5408
190	383.76	357.60	1198.6	0.01841	2.2367	13801	309569	5304
185	381.67	355.38	1198.3	0.01839	2.2906	13464	307420	5200
180	379.56	353.10	1198.0	0.01836	2.3472	13127	305227	5096
175	377.40	350.80	1197.6	0.01833	2.4067	12790	302990	4990
170	375.19	348.44	1197.2	0.01830	2.4693	12453	300701	4884
165	372.94	346.04	1196.8	0.01827	2.5353	12116	298363	4778
160	370.64	343.59	1196.4	0.01824	2.6048	11779	295964	4670
155	368.28	341.09	1196.0	0.01821	2.6784	11442	293504	4562
150	365.87	338.52	1195.5	0.01818	2.7562	11104	290979	4453
145	363.40	335.91	1195.1	0.01815	2.8387	10767	288385	4343
140	360.87	333.23	1194.6	0.01812	2.9264	10430	285711	4233
135	358.27	330.48	1194.0	0.01809	3.0197	10093	282959	4122
130	355.61	327.67	1193.5	0.01806	3.1192	9756	280117	4009
125	352.87	324.78	1192.9	0.01802	3.2256	9419	277182	3896
120	350.06	321.82	1192.3	0.01799	3.3396	9082	274140	3783
115	347.15	318.76	1191.7	0.01796	3.4620	8745	270981	3668
110	344.17	315.63	1191.0	0.01792	3.5940	8408	267703	3552
105	341.08	312.39	1190.3	0.01789	3.7365	8070	264279	3436
100	337.89	309.05	1189.6	0.01785	3.8910	7733	260703	3318
95	334.59	305.59	1188.8	0.01781	4.0590	7396	256959	3200
90	331.16	302.02	1188.0	0.01778	4.2425	7059	253026	3080
85	327.60	298.31	1187.1	0.01774	4.4437	6722	248870	2959
80	323.90	294.46	1186.2	0.01770	4.6654	6385	244459	2837
75	320.04	290.44	1185.2	0.01766	4.9109	6048	239747	2715
70	316.01	286.26	1184.1	0.01761	5.1843	5711	234698	2591
65	311.78	281.88	1183.0	0.01757	5.4907	5374	229238	2465
60	307.33	277.28	1181.8	0.01753	5.8367	5036	223300	2339
55	302.64	272.44	1180.5	0.01748	6.2304	4699	216769	2211
50	297.67	267.31	1179.1	0.01743	6.6827	4362	209478	2082

prpv [psig]	erpv [Btu]	dmg_srv [lbm]	mf_sp [lbm]	dt [min.]	t [min.]	qdh [Btu/min.]	esp [Btu]	tsp [°F]
1145	4.79e+008	0.0	4861827	0.0000	2.000	4381320	5.75e+008	150.3
1135	4.77e+008	1867.9	4863695	0.0241	2.024	4374717	5.78e+008	150.8
1125	4.75e+008	1867.4	4865562	0.0243	2.048	4368058	5.80e+008	151.2
1115	4.73e+008	1872.9	4867435	0.0246	2.073	4361320	5.82e+008	151.6
1105	4.71e+008	1878.3	4869313	0.0249	2.098	4354502	5.84e+008	152.0
1095	4.69e+008	1882.5	4871196	0.0252	2.123	4347608	5.87e+008	152.4
1085	4.66e+008	1884.3	4873080	0.0254	2.148	4340644	5.89e+008	152.8
1075	4.64e+008	1891.2	4874971	0.0257	2.174	4333591	5.91e+008	153.2
1065	4.62e+008	1892.1	4876863	0.0260	2.200	4326469	5.93e+008	153.6
1055	4.60e+008	1893.8	4878757	0.0263	2.226	4319274	5.96e+008	154.1
1045	4.58e+008	1904.5	4880662	0.0267	2.253	4311970	5.98e+008	154.5
1035	4.56e+008	1902.0	4882564	0.0269	2.280	4304606	6.00e+008	154.9
1025	4.54e+008	1915.9	4884479	0.0273	2.307	4297117	6.02e+008	155.3
1015	4.51e+008	1913.2	4886393	0.0276	2.335	4289566	6.05e+008	155.7
1005	4.49e+008	1920.4	4888313	0.0279	2.363	4281912	6.07e+008	156.2
995	4.47e+008	1921.6	4890235	0.0282	2.391	4274178	6.09e+008	156.6
985	4.45e+008	1931.8	4892166	0.0287	2.420	4266324	6.11e+008	157.0
975	4.43e+008	1935.5	4894102	0.0290	2.449	4258376	6.14e+008	157.4
965	4.40e+008	1942.2	4896044	0.0294	2.478	4250319	6.16e+008	157.8
955	4.38e+008	1946.6	4897991	0.0298	2.508	4242161	6.18e+008	158.3
945	4.36e+008	1951.7	4899942	0.0302	2.538	4233896	6.21e+008	158.7
935	4.34e+008	1959.8	4901902	0.0306	2.569	4225509	6.23e+008	159.1
925	4.32e+008	1957.6	4903860	0.0309	2.600	4217043	6.25e+008	159.5

915	4.29e+008	1972.5	4905832	0.0315	2.631	4208420	6.28e+008	160.0
905	4.27e+008	1978.8	4907811	0.0319	2.663	4199676	6.30e+008	160.4
895	4.25e+008	1985.1	4909796	0.0324	2.695	4190807	6.33e+008	160.8
885	4.23e+008	1991.3	4911788	0.0328	2.728	4181812	6.35e+008	161.3
875	4.20e+008	1991.7	4913779	0.0332	2.761	4172714	6.37e+008	161.7
865	4.18e+008	2003.5	4915783	0.0338	2.795	4163458	6.40e+008	162.1
855	4.16e+008	2010.5	4917793	0.0343	2.829	4154063	6.42e+008	162.6
845	4.14e+008	2018.1	4919811	0.0348	2.864	4144523	6.45e+008	163.0
835	4.11e+008	2029.6	4921841	0.0354	2.900	4134815	6.47e+008	163.5
825	4.09e+008	2030.8	4923872	0.0359	2.935	4124986	6.49e+008	163.9
815	4.07e+008	2048.4	4925920	0.0366	2.972	4114952	6.52e+008	164.3
805	4.04e+008	2044.1	4927964	0.0370	3.009	4104817	6.54e+008	164.8
795	4.02e+008	2064.3	4930029	0.0378	3.047	4094456	6.57e+008	165.2
785	4.00e+008	2065.8	4932094	0.0383	3.085	4083957	6.59e+008	165.7
775	3.98e+008	2080.1	4934175	0.0391	3.124	4073252	6.62e+008	166.1
765	3.95e+008	2081.6	4936256	0.0396	3.164	4062402	6.64e+008	166.6
755	3.93e+008	2100.0	4938356	0.0405	3.204	4051314	6.67e+008	167.0
745	3.90e+008	2108.0	4940464	0.0412	3.246	4040037	6.69e+008	167.5
735	3.88e+008	2112.9	4942577	0.0418	3.287	4028582	6.72e+008	167.9
725	3.86e+008	2124.3	4944701	0.0426	3.330	4016911	6.74e+008	168.4
715	3.83e+008	2143.1	4946844	0.0436	3.373	4004975	6.77e+008	168.9
705	3.81e+008	2144.9	4948989	0.0442	3.418	3992863	6.80e+008	169.3
695	3.78e+008	2162.5	4951152	0.0452	3.463	3980479	6.82e+008	169.8
685	3.76e+008	2174.3	4953326	0.0461	3.509	3967850	6.85e+008	170.2
675	3.74e+008	2180.6	4955507	0.0469	3.556	3955000	6.87e+008	170.7
665	3.71e+008	2198.6	4957705	0.0480	3.604	3941854	6.90e+008	171.2
655	3.69e+008	2209.2	4959915	0.0489	3.653	3928448	6.93e+008	171.7
645	3.66e+008	2223.2	4962138	0.0500	3.703	3914751	6.95e+008	172.1
635	3.64e+008	2239.1	4964377	0.0511	3.754	3900745	6.98e+008	172.6
625	3.61e+008	2253.9	4966631	0.0523	3.806	3886426	7.01e+008	173.1
615	3.59e+008	2259.6	4968890	0.0532	3.859	3871842	7.03e+008	173.6
605	3.56e+008	2282.9	4971173	0.0546	3.914	3856870	7.06e+008	174.1
595	3.54e+008	2297.2	4973470	0.0559	3.970	3841557	7.09e+008	174.6
585	3.51e+008	2312.6	4975783	0.0572	4.027	3829395	7.12e+008	175.0
575	3.48e+008	2332.1	4978115	0.0587	4.086	3820936	7.15e+008	175.5
565	3.46e+008	2346.4	4980462	0.0600	4.146	3812278	7.17e+008	176.0
555	3.43e+008	2366.6	4982828	0.0616	4.207	3803392	7.20e+008	176.5
545	3.41e+008	2390.1	4985218	0.0633	4.271	3794258	7.23e+008	177.1
535	3.38e+008	2404.2	4987622	0.0649	4.336	3784903	7.26e+008	177.6
525	3.35e+008	2428.9	4990051	0.0668	4.402	3775276	7.29e+008	178.1
515	3.33e+008	2448.9	4992500	0.0686	4.471	3765387	7.32e+008	178.6
505	3.30e+008	2468.4	4994969	0.0704	4.541	3755227	7.35e+008	179.1
495	3.27e+008	2497.3	4997466	0.0727	4.614	3744747	7.38e+008	179.7
485	3.24e+008	2520.7	4999987	0.0748	4.689	3733957	7.41e+008	180.2
475	3.22e+008	2545.1	5002532	0.0771	4.766	3722840	7.44e+008	180.7
465	3.19e+008	2569.0	5005101	0.0794	4.845	3711384	7.47e+008	181.3
455	3.16e+008	2597.9	5007699	0.0820	4.927	3699553	7.50e+008	181.8
445	3.13e+008	2630.4	5010329	0.0849	5.012	3687314	7.53e+008	182.4
435	3.10e+008	2658.0	5012987	0.0877	5.100	3674671	7.57e+008	182.9
425	3.07e+008	2688.3	5015675	0.0907	5.191	3661593	7.60e+008	183.5
415	3.04e+008	2719.4	5018395	0.0939	5.284	3648057	7.63e+008	184.1
405	3.01e+008	2757.7	5021152	0.0975	5.382	3634002	7.66e+008	184.6
395	2.98e+008	2796.4	5023949	0.1012	5.483	3619402	7.70e+008	185.2
385	2.95e+008	2826.6	5026776	0.1049	5.588	3604275	7.73e+008	185.8
375	2.92e+008	2872.0	5029648	0.1093	5.697	3588511	7.77e+008	186.4
365	2.89e+008	2910.5	5032558	0.1137	5.811	3572115	7.80e+008	187.0
355	2.86e+008	2953.3	5035511	0.1185	5.929	3555028	7.84e+008	187.6
345	2.83e+008	3007.0	5038518	0.1240	6.053	3539537	7.87e+008	188.3
335	2.80e+008	3052.4	5041571	0.1295	6.183	3526654	7.91e+008	188.9
325	2.76e+008	3103.3	5044674	0.1355	6.318	3513171	7.95e+008	189.5
315	2.73e+008	3165.3	5047839	0.1424	6.461	3499002	7.99e+008	190.2
305	2.70e+008	3224.1	5051063	0.1496	6.610	3484117	8.02e+008	190.9
295	2.66e+008	3287.2	5054351	0.1574	6.768	3468452	8.06e+008	191.5
285	2.63e+008	3354.9	5057706	0.1660	6.934	3451930	8.10e+008	192.2
275	2.59e+008	3430.8	5061136	0.1756	7.110	3434452	8.15e+008	192.9
265	2.56e+008	3510.5	5064647	0.1862	7.296	3415927	8.19e+008	193.7
255	2.52e+008	3591.9	5068239	0.1975	7.493	3396271	8.23e+008	194.4
245	2.48e+008	3684.3	5071923	0.2104	7.704	3375332	8.28e+008	195.2

235	2.44e+008	3782.7	5075706	0.2247	7.928	3352973	8.32e+008	195.9
225	2.40e+008	3891.5	5079597	0.2408	8.169	3332670	8.37e+008	196.7
215	2.36e+008	4008.3	5083606	0.2588	8.428	3312514	8.42e+008	197.5
205	2.32e+008	4136.1	5087742	0.2792	8.707	3290768	8.47e+008	198.4
200	2.30e+008	2134.1	5089876	0.1474	8.855	3279287	8.49e+008	198.8
195	2.28e+008	2171.4	5092047	0.1536	9.008	3267327	8.52e+008	199.3
190	2.26e+008	2205.8	5094253	0.1598	9.168	3254880	8.54e+008	199.7
185	2.24e+008	2253.6	5096507	0.1674	9.335	3241845	8.57e+008	200.2
180	2.22e+008	2297.7	5098804	0.1750	9.510	3228214	8.60e+008	200.6
175	2.19e+008	2341.9	5101146	0.1831	9.694	3213955	8.63e+008	201.1
170	2.17e+008	2395.7	5103542	0.1924	9.886	3198972	8.65e+008	201.6
165	2.15e+008	2444.7	5105987	0.2018	10.088	3184953	8.68e+008	202.1
160	2.12e+008	2505.8	5108492	0.2127	10.300	3172496	8.71e+008	202.6
155	2.10e+008	2568.8	5111061	0.2245	10.525	3159350	8.74e+008	203.1
150	2.08e+008	2633.2	5113694	0.2371	10.762	3145466	8.78e+008	203.6
145	2.05e+008	2704.5	5116399	0.2512	11.013	3130759	8.81e+008	204.2
140	2.03e+008	2784.6	5119183	0.2670	11.280	3115127	8.84e+008	204.7
135	2.00e+008	2863.4	5122047	0.2837	11.564	3098516	8.88e+008	205.3
130	1.98e+008	2953.5	5125000	0.3027	11.867	3080790	8.91e+008	205.9
125	1.95e+008	3048.4	5128049	0.3236	12.190	3061840	8.95e+008	206.5
120	1.92e+008	3155.5	5131204	0.3475	12.538	3041496	8.98e+008	207.1
115	1.89e+008	3274.0	5134478	0.3744	12.912	3019574	9.02e+008	207.8
110	1.86e+008	3393.6	5137872	0.4036	13.316	2995940	9.06e+008	208.4
105	1.83e+008	3540.2	5141412	0.4387	13.754	2970256	9.11e+008	209.1
100	1.80e+008	3693.7	5145106	0.4776	14.232	2942289	9.15e+008	209.8
95	1.77e+008	3862.4	5148968	0.5222	14.754	2911712	9.20e+008	210.6
90	1.74e+008	4052.4	5153021	0.5741	15.328	2883403	9.24e+008	211.4
85	1.71e+008	4276.6	5157297	0.6362	15.965	2856427	9.30e+008	212.2
80	1.67e+008	4533.1	5161830	0.7100	16.675	2826324	9.35e+008	213.1
75	1.64e+008	4835.3	5166666	0.7995	17.474	2792425	9.41e+008	214.1
70	1.60e+008	5172.7	5171838	0.9058	18.380	2754019	9.47e+008	215.1
65	1.56e+008	5585.0	5177423	1.0393	19.419	2709951	9.53e+008	216.1
60	1.52e+008	6064.3	5183488	1.2041	20.623	2664651	9.61e+008	217.3
55	1.48e+008	6659.2	5190147	1.4171	22.040	2617647	9.68e+008	218.6
50	1.43e+008	7420.5	5197567	1.7011	23.741	2561222	9.77e+008	220.0

torus water level = 12.00 ft

prpv [psig]	trpv [°F]	hf_rpv [Btu/lbm]	hg_rpv [Btu/lbm]	vf_rpv [ft3/lbm]	vg_rpv [ft3/lbm]	wsrv [lbm/min.]	mf_rpv [lbm]	mg_rpv [lbm]
1145	562.88	566.16	1186.6	0.02217	0.3771	78190	532000	15300
1135	561.79	564.74	1187.0	0.02213	0.3808	77515	530124	15309
1125	560.70	563.31	1187.4	0.02210	0.3847	76841	528251	15314
1115	559.60	561.87	1187.8	0.02206	0.3886	76167	526376	15316
1105	558.49	560.42	1188.3	0.02202	0.3926	75493	524498	15315
1095	557.37	558.97	1188.7	0.02199	0.3966	74819	522619	15312
1085	556.24	557.51	1189.1	0.02195	0.4007	74144	520742	15305
1075	555.11	556.04	1189.5	0.02192	0.4049	73470	518860	15295
1065	553.97	554.57	1189.9	0.02188	0.4091	72796	516980	15283
1055	552.82	553.09	1190.3	0.02184	0.4135	72122	515102	15268
1045	551.66	551.60	1190.7	0.02181	0.4179	71447	513216	15250
1035	550.50	550.11	1191.1	0.02177	0.4224	70773	511335	15229
1025	549.32	548.60	1191.4	0.02173	0.4269	70099	509442	15205
1015	548.14	547.09	1191.8	0.02170	0.4316	69425	507556	15179
1005	546.95	545.57	1192.2	0.02166	0.4363	68751	505664	15150
995	545.75	544.04	1192.6	0.02163	0.4412	68076	503775	15118
985	544.54	542.51	1192.9	0.02159	0.4461	67402	501877	15083
975	543.32	540.96	1193.3	0.02155	0.4511	66728	499979	15046
965	542.10	539.40	1193.7	0.02152	0.4562	66054	498076	15007
955	540.86	537.85	1194.0	0.02148	0.4615	65379	496171	14965
945	539.61	536.27	1194.4	0.02145	0.4668	64705	494264	14920
935	538.35	534.70	1194.7	0.02141	0.4722	64031	492352	14873
925	537.09	533.11	1195.1	0.02137	0.4778	63357	490444	14823
915	535.81	531.51	1195.4	0.02134	0.4834	62683	488524	14771
905	534.52	529.90	1195.7	0.02130	0.4892	62008	486600	14716
895	533.22	528.28	1196.1	0.02127	0.4951	61334	484671	14659
885	531.91	526.65	1196.4	0.02123	0.5011	60660	482740	14600
875	530.59	525.02	1196.7	0.02119	0.5072	59986	480810	14538

865	529.26	523.37	1197.0	0.02116	0.5135	59311	478871	14474
855	527.92	521.71	1197.3	0.02112	0.5199	58637	476927	14407
845	526.56	520.04	1197.7	0.02109	0.5265	57963	474977	14338
835	525.20	518.35	1198.0	0.02105	0.5332	57289	473019	14267
825	523.82	516.66	1198.3	0.02101	0.5401	56614	471061	14194
815	522.42	514.95	1198.5	0.02098	0.5471	55940	469088	14118
805	521.02	513.24	1198.8	0.02094	0.5543	55266	467123	14040
795	519.60	511.50	1199.1	0.02090	0.5617	54592	465138	13960
785	518.16	509.76	1199.4	0.02087	0.5692	53918	463155	13878
775	516.72	508.00	1199.7	0.02083	0.5769	53243	461159	13793
765	515.26	506.24	1199.9	0.02080	0.5848	52569	459165	13706
755	513.78	504.45	1200.2	0.02076	0.5929	51895	457153	13618
745	512.30	502.65	1200.4	0.02072	0.6012	51221	455136	13527
735	510.80	500.84	1200.7	0.02069	0.6097	50546	453117	13433
725	509.28	499.01	1200.9	0.02065	0.6185	49872	451088	13338
715	507.74	497.17	1201.2	0.02061	0.6275	49198	449042	13241
705	506.19	495.31	1201.4	0.02058	0.6367	48524	446997	13141
695	504.62	493.43	1201.6	0.02054	0.6461	47850	444936	13039
685	503.03	491.54	1201.8	0.02050	0.6559	47175	442865	12936
675	501.43	489.64	1202.1	0.02047	0.6658	46501	440790	12830
665	499.81	487.71	1202.3	0.02043	0.6761	45827	438699	12722
655	498.17	485.77	1202.5	0.02039	0.6867	45153	436600	12613
645	496.52	483.81	1202.7	0.02036	0.6976	44478	434488	12501
635	494.84	481.83	1202.8	0.02032	0.7088	43804	432363	12387
625	493.14	479.83	1203.0	0.02028	0.7203	43130	430225	12271
615	491.43	477.82	1203.2	0.02024	0.7322	42456	428084	12153
605	489.69	475.78	1203.4	0.02021	0.7445	41782	425920	12033
595	487.93	473.72	1203.5	0.02017	0.7571	41107	423745	11911
585	486.15	471.64	1203.7	0.02013	0.7701	40433	421556	11788
575	484.35	469.53	1203.8	0.02009	0.7836	39759	419350	11662
565	482.52	467.40	1203.9	0.02005	0.7975	39085	417131	11534
555	480.66	465.25	1204.1	0.02002	0.8119	38410	414894	11404
545	478.78	463.07	1204.2	0.01998	0.8268	37736	412636	11273
535	476.88	460.87	1204.3	0.01994	0.8422	37062	410365	11139
525	474.95	458.64	1204.4	0.01990	0.8582	36388	408072	11004
515	472.99	456.39	1204.5	0.01986	0.8747	35714	405760	10866
505	471.01	454.11	1204.5	0.01982	0.8919	35039	403432	10727
495	468.99	451.80	1204.6	0.01978	0.9097	34365	401076	10585
485	466.95	449.45	1204.7	0.01974	0.9282	33691	398698	10442
475	464.87	447.08	1204.7	0.01970	0.9474	33017	396298	10297
465	462.76	444.68	1204.8	0.01966	0.9674	32342	393876	10150
455	460.61	442.24	1204.8	0.01962	0.9882	31668	391427	10001
445	458.43	439.76	1204.8	0.01958	1.0099	30994	388948	9850
435	456.21	437.25	1204.8	0.01954	1.0325	30320	386443	9697
425	453.96	434.69	1204.8	0.01950	1.0561	29646	383910	9542
415	451.67	432.11	1204.8	0.01946	1.0807	28971	381347	9385
405	449.33	429.48	1204.7	0.01942	1.1065	28297	378748	9226
395	446.95	426.80	1204.7	0.01938	1.1335	27623	376113	9065
385	444.53	424.09	1204.6	0.01934	1.1618	26949	373449	8902
375	442.06	421.32	1204.5	0.01929	1.1915	26274	370742	8738
365	439.53	418.51	1204.4	0.01925	1.2227	25600	367998	8571
355	436.96	415.64	1204.3	0.01921	1.2556	24926	365214	8402
345	434.33	412.72	1204.1	0.01916	1.2902	24252	362377	8231
335	431.65	409.74	1204.0	0.01912	1.3267	23578	359498	8058
325	428.91	406.71	1203.8	0.01908	1.3652	22903	356570	7883
315	426.10	403.60	1203.6	0.01903	1.4061	22229	353581	7706
305	423.22	400.43	1203.4	0.01899	1.4493	21555	350536	7527
295	420.27	397.20	1203.1	0.01894	1.4953	20881	347431	7346
285	417.26	393.89	1202.9	0.01889	1.5443	20206	344259	7162
275	414.16	390.50	1202.6	0.01885	1.5964	19532	341014	6977
265	410.97	387.01	1202.3	0.01880	1.6522	18858	337691	6789
255	407.70	383.45	1201.9	0.01875	1.7120	18184	334290	6598
245	404.33	379.79	1201.5	0.01870	1.7762	17510	330798	6406
235	400.87	376.03	1201.1	0.01865	1.8453	16835	327210	6211
225	397.28	372.15	1200.6	0.01860	1.9200	16161	323515	6014
215	393.59	368.16	1200.1	0.01855	2.0010	15487	319707	5814
205	389.76	364.04	1199.6	0.01849	2.0891	14813	315773	5612
200	387.79	361.93	1199.3	0.01847	2.1360	14476	313741	5510

195	385.79	359.78	1199.0	0.01844	2.1852	14138	311672	5408
190	383.76	357.60	1198.6	0.01841	2.2367	13801	309569	5304
185	381.67	355.38	1198.3	0.01839	2.2906	13464	307420	5200
180	379.56	353.10	1198.0	0.01836	2.3472	13127	305227	5096
175	377.40	350.80	1197.6	0.01833	2.4067	12790	302990	4990
170	375.19	348.44	1197.2	0.01830	2.4693	12453	300701	4884
165	372.94	346.04	1196.8	0.01827	2.5353	12116	298363	4778
160	370.64	343.59	1196.4	0.01824	2.6048	11779	295964	4670
155	368.28	341.09	1196.0	0.01821	2.6784	11442	293504	4562
150	365.87	338.52	1195.5	0.01818	2.7562	11104	290979	4453
145	363.40	335.91	1195.1	0.01815	2.8387	10767	288385	4343
140	360.87	333.23	1194.6	0.01812	2.9264	10430	285711	4233
135	358.27	330.48	1194.0	0.01809	3.0197	10093	282959	4122
130	355.61	327.67	1193.5	0.01806	3.1192	9756	280117	4009
125	352.87	324.78	1192.9	0.01802	3.2256	9419	277182	3896
120	350.06	321.82	1192.3	0.01799	3.3396	9082	274140	3783
115	347.15	318.76	1191.7	0.01796	3.4620	8745	270981	3668
110	344.17	315.63	1191.0	0.01792	3.5940	8408	267703	3552
105	341.08	312.39	1190.3	0.01789	3.7365	8070	264279	3436
100	337.89	309.05	1189.6	0.01785	3.8910	7733	260703	3318
95	334.59	305.59	1188.8	0.01781	4.0590	7396	256959	3200
90	331.16	302.02	1188.0	0.01778	4.2425	7059	253026	3080
85	327.60	298.31	1187.1	0.01774	4.4437	6722	248870	2959
80	323.90	294.46	1186.2	0.01770	4.6654	6385	244459	2837
75	320.04	290.44	1185.2	0.01766	4.9109	6048	239747	2715
70	316.01	286.26	1184.1	0.01761	5.1843	5711	234698	2591
65	311.78	281.88	1183.0	0.01757	5.4907	5374	229238	2465
60	307.33	277.28	1181.8	0.01753	5.8367	5036	223300	2339
55	302.64	272.44	1180.5	0.01748	6.2304	4699	216769	2211
50	297.67	267.31	1179.1	0.01743	6.6827	4362	209478	2082

prpv [psig]	erpv [Btu]	dmg_srv [lbm]	mf_sp [lbm]	dt [min.]	t [min.]	qdh [Btu/min.]	esp [Btu]	tsp [°F]
1145	4.79e+008	0.0	5452999	0.0000	2.000	4381320	6.87e+008	157.9
1135	4.77e+008	1867.9	5454867	0.0241	2.024	4374717	6.89e+008	158.3
1125	4.75e+008	1867.4	5456734	0.0243	2.048	4368058	6.91e+008	158.6
1115	4.73e+008	1872.9	5458607	0.0246	2.073	4361320	6.93e+008	159.0
1105	4.71e+008	1878.3	5460485	0.0249	2.098	4354502	6.95e+008	159.4
1095	4.69e+008	1882.5	5462368	0.0252	2.123	4347608	6.98e+008	159.7
1085	4.66e+008	1884.3	5464252	0.0254	2.148	4340644	7.00e+008	160.1
1075	4.64e+008	1891.2	5466143	0.0257	2.174	4333591	7.02e+008	160.5
1065	4.62e+008	1892.1	5468035	0.0260	2.200	4326469	7.04e+008	160.8
1055	4.60e+008	1893.8	5469929	0.0263	2.226	4319274	7.07e+008	161.2
1045	4.58e+008	1904.5	5471834	0.0267	2.253	4311970	7.09e+008	161.6
1035	4.56e+008	1902.0	5473736	0.0269	2.280	4304606	7.11e+008	161.9
1025	4.54e+008	1915.9	5475652	0.0273	2.307	4297117	7.13e+008	162.3
1015	4.51e+008	1913.2	5477565	0.0276	2.335	4289566	7.16e+008	162.7
1005	4.49e+008	1920.4	5479485	0.0279	2.363	4281912	7.18e+008	163.0
995	4.47e+008	1921.6	5481407	0.0282	2.391	4274178	7.20e+008	163.4
985	4.45e+008	1931.8	5483338	0.0287	2.420	4266324	7.23e+008	163.8
975	4.43e+008	1935.5	5485274	0.0290	2.449	4258376	7.25e+008	164.2
965	4.40e+008	1942.2	5487216	0.0294	2.478	4250319	7.27e+008	164.5
955	4.38e+008	1946.6	5489163	0.0298	2.508	4242161	7.30e+008	164.9
945	4.36e+008	1951.7	5491114	0.0302	2.538	4233896	7.32e+008	165.3
935	4.34e+008	1959.8	5493074	0.0306	2.569	4225509	7.34e+008	165.7
925	4.32e+008	1957.6	5495032	0.0309	2.600	4217043	7.37e+008	166.0
915	4.29e+008	1972.5	5497004	0.0315	2.631	4208420	7.39e+008	166.4
905	4.27e+008	1978.8	5498983	0.0319	2.663	4199676	7.41e+008	166.8
895	4.25e+008	1985.1	5500968	0.0324	2.695	4190807	7.44e+008	167.2
885	4.23e+008	1991.3	5502960	0.0328	2.728	4181812	7.46e+008	167.6
875	4.20e+008	1991.7	5504951	0.0332	2.761	4172714	7.48e+008	168.0
865	4.18e+008	2003.5	5506955	0.0338	2.795	4163458	7.51e+008	168.3
855	4.16e+008	2010.5	5508965	0.0343	2.829	4154063	7.53e+008	168.7
845	4.14e+008	2018.1	5510983	0.0348	2.864	4144523	7.56e+008	169.1
835	4.11e+008	2029.6	5513013	0.0354	2.900	4134815	7.58e+008	169.5
825	4.09e+008	2030.8	5515044	0.0359	2.935	4124986	7.61e+008	169.9
815	4.07e+008	2048.4	5517092	0.0366	2.972	4114952	7.63e+008	170.3
805	4.04e+008	2044.1	5519136	0.0370	3.009	4104817	7.65e+008	170.7

795	4.02e+008	2064.3	5521201	0.0378	3.047	4094456	7.68e+008	171.1
785	4.00e+008	2065.8	5523266	0.0383	3.085	4083957	7.70e+008	171.5
775	3.98e+008	2080.1	5525347	0.0391	3.124	4073252	7.73e+008	171.9
765	3.95e+008	2081.6	5527428	0.0396	3.164	4062402	7.75e+008	172.3
755	3.93e+008	2100.0	5529528	0.0405	3.204	4051314	7.78e+008	172.7
745	3.90e+008	2108.0	5531636	0.0412	3.246	4040037	7.80e+008	173.1
735	3.88e+008	2112.9	5533749	0.0418	3.287	4028582	7.83e+008	173.5
725	3.86e+008	2124.3	5535873	0.0426	3.330	4016911	7.86e+008	173.9
715	3.83e+008	2143.1	5538016	0.0436	3.373	4004975	7.88e+008	174.3
705	3.81e+008	2144.9	5540161	0.0442	3.418	3992863	7.91e+008	174.7
695	3.78e+008	2162.5	5542324	0.0452	3.463	3980479	7.93e+008	175.1
685	3.76e+008	2174.3	5544498	0.0461	3.509	3967850	7.96e+008	175.5
675	3.74e+008	2180.6	5546679	0.0469	3.556	3955000	7.99e+008	176.0
665	3.71e+008	2198.6	5548877	0.0480	3.604	3941854	8.01e+008	176.4
655	3.69e+008	2209.2	5551087	0.0489	3.653	3928448	8.04e+008	176.8
645	3.66e+008	2223.2	5553310	0.0500	3.703	3914751	8.07e+008	177.2
635	3.64e+008	2239.1	5555549	0.0511	3.754	3900745	8.09e+008	177.7
625	3.61e+008	2253.9	5557803	0.0523	3.806	3886426	8.12e+008	178.1
615	3.59e+008	2259.6	5560062	0.0532	3.859	3871842	8.15e+008	178.5
605	3.56e+008	2282.9	5562345	0.0546	3.914	3856870	8.17e+008	178.9
595	3.54e+008	2297.2	5564642	0.0559	3.970	3841557	8.20e+008	179.4
585	3.51e+008	2312.6	5566955	0.0572	4.027	3829395	8.23e+008	179.8
575	3.48e+008	2332.1	5569287	0.0587	4.086	3820936	8.26e+008	180.3
565	3.46e+008	2346.4	5571634	0.0600	4.146	3812278	8.29e+008	180.7
555	3.43e+008	2366.6	5574000	0.0616	4.207	3803392	8.31e+008	181.2
545	3.41e+008	2390.1	5576390	0.0633	4.271	3794258	8.34e+008	181.6
535	3.38e+008	2404.2	5578794	0.0649	4.336	3784903	8.37e+008	182.1
525	3.35e+008	2428.9	5581223	0.0668	4.402	3775276	8.40e+008	182.5
515	3.33e+008	2448.9	5583672	0.0686	4.471	3765387	8.43e+008	183.0
505	3.30e+008	2468.4	5586141	0.0704	4.541	3755227	8.46e+008	183.5
495	3.27e+008	2497.3	5588638	0.0727	4.614	3744747	8.49e+008	183.9
485	3.24e+008	2520.7	5591159	0.0748	4.689	3733957	8.52e+008	184.4
475	3.22e+008	2545.1	5593704	0.0771	4.766	3722840	8.55e+008	184.9
465	3.19e+008	2569.0	5596273	0.0794	4.845	3711384	8.58e+008	185.4
455	3.16e+008	2597.9	5598871	0.0820	4.927	3699553	8.61e+008	185.8
445	3.13e+008	2630.4	5601501	0.0849	5.012	3687314	8.65e+008	186.3
435	3.10e+008	2658.0	5604159	0.0877	5.100	3674671	8.68e+008	186.8
425	3.07e+008	2688.3	5606847	0.0907	5.191	3661593	8.71e+008	187.3
415	3.04e+008	2719.4	5609567	0.0939	5.284	3648057	8.74e+008	187.8
405	3.01e+008	2757.7	5612324	0.0975	5.382	3634002	8.78e+008	188.4
395	2.98e+008	2796.4	5615121	0.1012	5.483	3619402	8.81e+008	188.9
385	2.95e+008	2826.6	5617948	0.1049	5.588	3604275	8.84e+008	189.4
375	2.92e+008	2872.0	5620820	0.1093	5.697	3588511	8.88e+008	189.9
365	2.89e+008	2910.5	5623730	0.1137	5.811	3572115	8.91e+008	190.5
355	2.86e+008	2953.3	5626683	0.1185	5.929	3555028	8.95e+008	191.0
345	2.83e+008	3007.0	5629690	0.1240	6.053	3539537	8.98e+008	191.6
335	2.80e+008	3052.4	5632743	0.1295	6.183	3526654	9.02e+008	192.2
325	2.76e+008	3103.3	5635846	0.1355	6.318	3513171	9.06e+008	192.7
315	2.73e+008	3165.3	5639011	0.1424	6.461	3499002	9.10e+008	193.3
305	2.70e+008	3224.1	5642235	0.1496	6.610	3484117	9.14e+008	193.9
295	2.66e+008	3287.2	5645523	0.1574	6.768	3468452	9.18e+008	194.5
285	2.63e+008	3354.9	5648878	0.1660	6.934	3451930	9.22e+008	195.1
275	2.59e+008	3430.8	5652308	0.1756	7.110	3434452	9.26e+008	195.8
265	2.56e+008	3510.5	5655819	0.1862	7.296	3415927	9.30e+008	196.4
255	2.52e+008	3591.9	5659411	0.1975	7.493	3396271	9.34e+008	197.1
245	2.48e+008	3684.3	5663095	0.2104	7.704	3375332	9.39e+008	197.8
235	2.44e+008	3782.7	5666878	0.2247	7.928	3352973	9.43e+008	198.4
225	2.40e+008	3891.5	5670769	0.2408	8.169	3332670	9.48e+008	199.2
215	2.36e+008	4008.3	5674778	0.2588	8.428	3312514	9.53e+008	199.9
205	2.32e+008	4136.1	5678914	0.2792	8.707	3290768	9.58e+008	200.6
200	2.30e+008	2134.1	5681048	0.1474	8.855	3279287	9.60e+008	201.0
195	2.28e+008	2171.4	5683219	0.1536	9.008	3267327	9.63e+008	201.4
190	2.26e+008	2205.8	5685425	0.1598	9.168	3254880	9.65e+008	201.8
185	2.24e+008	2253.6	5687679	0.1674	9.335	3241845	9.68e+008	202.2
180	2.22e+008	2297.7	5689976	0.1750	9.510	3228214	9.71e+008	202.6
175	2.19e+008	2341.9	5692318	0.1831	9.694	3213955	9.74e+008	203.1
170	2.17e+008	2395.7	5694714	0.1924	9.886	3198972	9.77e+008	203.5
165	2.15e+008	2444.7	5697159	0.2018	10.088	3184953	9.80e+008	203.9

160	2.12e+008	2505.8	5699664	0.2127	10.300	3172496	9.83e+008	204.4
155	2.10e+008	2568.8	5702233	0.2245	10.525	3159350	9.86e+008	204.8
150	2.08e+008	2633.2	5704866	0.2371	10.762	3145466	9.89e+008	205.3
145	2.05e+008	2704.5	5707571	0.2512	11.013	3130759	9.92e+008	205.8
140	2.03e+008	2784.6	5710355	0.2670	11.280	3115127	9.95e+008	206.3
135	2.00e+008	2863.4	5713219	0.2837	11.564	3098516	9.99e+008	206.8
130	1.98e+008	2953.5	5716172	0.3027	11.867	3080790	1.00e+009	207.3
125	1.95e+008	3048.4	5719221	0.3236	12.190	3061840	1.01e+009	207.9
120	1.92e+008	3155.5	5722376	0.3475	12.538	3041496	1.01e+009	208.4
115	1.89e+008	3274.0	5725650	0.3744	12.912	3019574	1.01e+009	209.0
110	1.86e+008	3393.6	5729044	0.4036	13.316	2995940	1.02e+009	209.6
105	1.83e+008	3540.2	5732584	0.4387	13.754	2970256	1.02e+009	210.2
100	1.80e+008	3693.7	5736278	0.4776	14.232	2942289	1.03e+009	210.9
95	1.77e+008	3862.4	5740140	0.5222	14.754	2911712	1.03e+009	211.6
90	1.74e+008	4052.4	5744193	0.5741	15.328	2883403	1.04e+009	212.3
85	1.71e+008	4276.6	5748469	0.6362	15.965	2856427	1.04e+009	213.0
80	1.67e+008	4533.1	5753002	0.7100	16.675	2826324	1.05e+009	213.8
75	1.64e+008	4835.3	5757838	0.7995	17.474	2792425	1.05e+009	214.7
70	1.60e+008	5172.7	5763010	0.9058	18.380	2754019	1.06e+009	215.6
65	1.56e+008	5585.0	5768595	1.0393	19.419	2709951	1.06e+009	216.5
60	1.52e+008	6064.3	5774660	1.2041	20.623	2664651	1.07e+009	217.6
55	1.48e+008	6659.2	5781319	1.4171	22.040	2617647	1.08e+009	218.7
50	1.43e+008	7420.5	5788739	1.7011	23.741	2561222	1.09e+009	220.0

torus water level = 13.00 ft

prpv [psig]	trpv [°F]	hf_rpv [Btu/lbm]	hg_rpv [Btu/lbm]	vf_rpv [ft3/lbm]	vg_rpv [ft3/lbm]	wsrv [lbm/min.]	mf_rpv [lbm]	mg_rpv [lbm]
1145	562.88	566.16	1186.6	0.02217	0.3771	78190	532000	15300
1135	561.79	564.74	1187.0	0.02213	0.3808	77515	530124	15309
1125	560.70	563.31	1187.4	0.02210	0.3847	76841	528251	15314
1115	559.60	561.87	1187.8	0.02206	0.3886	76167	526376	15316
1105	558.49	560.42	1188.3	0.02202	0.3926	75493	524498	15315
1095	557.37	558.97	1188.7	0.02199	0.3966	74819	522619	15312
1085	556.24	557.51	1189.1	0.02195	0.4007	74144	520742	15305
1075	555.11	556.04	1189.5	0.02192	0.4049	73470	518860	15295
1065	553.97	554.57	1189.9	0.02188	0.4091	72796	516980	15283
1055	552.82	553.09	1190.3	0.02184	0.4135	72122	515102	15268
1045	551.66	551.60	1190.7	0.02181	0.4179	71447	513216	15250
1035	550.50	550.11	1191.1	0.02177	0.4224	70773	511335	15229
1025	549.32	548.60	1191.4	0.02173	0.4269	70099	509442	15205
1015	548.14	547.09	1191.8	0.02170	0.4316	69425	507556	15179
1005	546.95	545.57	1192.2	0.02166	0.4363	68751	505664	15150
995	545.75	544.04	1192.6	0.02163	0.4412	68076	503775	15118
985	544.54	542.51	1192.9	0.02159	0.4461	67402	501877	15083
975	543.32	540.96	1193.3	0.02155	0.4511	66728	499979	15046
965	542.10	539.40	1193.7	0.02152	0.4562	66054	498076	15007
955	540.86	537.85	1194.0	0.02148	0.4615	65379	496171	14965
945	539.61	536.27	1194.4	0.02145	0.4668	64705	494264	14920
935	538.35	534.70	1194.7	0.02141	0.4722	64031	492352	14873
925	537.09	533.11	1195.1	0.02137	0.4778	63357	490444	14823
915	535.81	531.51	1195.4	0.02134	0.4834	62683	488524	14771
905	534.52	529.90	1195.7	0.02130	0.4892	62008	486600	14716
895	533.22	528.28	1196.1	0.02127	0.4951	61334	484671	14659
885	531.91	526.65	1196.4	0.02123	0.5011	60660	482740	14600
875	530.59	525.02	1196.7	0.02119	0.5072	59986	480810	14538
865	529.26	523.37	1197.0	0.02116	0.5135	59311	478871	14474
855	527.92	521.71	1197.3	0.02112	0.5199	58637	476927	14407
845	526.56	520.04	1197.7	0.02109	0.5265	57963	474977	14338
835	525.20	518.35	1198.0	0.02105	0.5332	57289	473019	14267
825	523.82	516.66	1198.3	0.02101	0.5401	56614	471061	14194
815	522.42	514.95	1198.5	0.02098	0.5471	55940	469088	14118
805	521.02	513.24	1198.8	0.02094	0.5543	55266	467123	14040
795	519.60	511.50	1199.1	0.02090	0.5617	54592	465138	13960
785	518.16	509.76	1199.4	0.02087	0.5692	53918	463155	13878
775	516.72	508.00	1199.7	0.02083	0.5769	53243	461159	13793
765	515.26	506.24	1199.9	0.02080	0.5848	52569	459165	13706
755	513.78	504.45	1200.2	0.02076	0.5929	51895	457153	13618

745	512.30	502.65	1200.4	0.02072	0.6012	51221	455136	13527
735	510.80	500.84	1200.7	0.02069	0.6097	50546	453117	13433
725	509.28	499.01	1200.9	0.02065	0.6185	49872	451088	13338
715	507.74	497.17	1201.2	0.02061	0.6275	49198	449042	13241
705	506.19	495.31	1201.4	0.02058	0.6367	48524	446997	13141
695	504.62	493.43	1201.6	0.02054	0.6461	47850	444936	13039
685	503.03	491.54	1201.8	0.02050	0.6559	47175	442865	12936
675	501.43	489.64	1202.1	0.02047	0.6658	46501	440790	12830
665	499.81	487.71	1202.3	0.02043	0.6761	45827	438699	12722
655	498.17	485.77	1202.5	0.02039	0.6867	45153	436600	12613
645	496.52	483.81	1202.7	0.02036	0.6976	44478	434488	12501
635	494.84	481.83	1202.8	0.02032	0.7088	43804	432363	12387
625	493.14	479.83	1203.0	0.02028	0.7203	43130	430225	12271
615	491.43	477.82	1203.2	0.02024	0.7322	42456	428084	12153
605	489.69	475.78	1203.4	0.02021	0.7445	41782	425920	12033
595	487.93	473.72	1203.5	0.02017	0.7571	41107	423745	11911
585	486.15	471.64	1203.7	0.02013	0.7701	40433	421556	11788
575	484.35	469.53	1203.8	0.02009	0.7836	39759	419350	11662
565	482.52	467.40	1203.9	0.02005	0.7975	39085	417131	11534
555	480.66	465.25	1204.1	0.02002	0.8119	38410	414894	11404
545	478.78	463.07	1204.2	0.01998	0.8268	37736	412636	11273
535	476.88	460.87	1204.3	0.01994	0.8422	37062	410365	11139
525	474.95	458.64	1204.4	0.01990	0.8582	36388	408072	11004
515	472.99	456.39	1204.5	0.01986	0.8747	35714	405760	10866
505	471.01	454.11	1204.5	0.01982	0.8919	35039	403432	10727
495	468.99	451.80	1204.6	0.01978	0.9097	34365	401076	10585
485	466.95	449.45	1204.7	0.01974	0.9282	33691	398698	10442
475	464.87	447.08	1204.7	0.01970	0.9474	33017	396298	10297
465	462.76	444.68	1204.8	0.01966	0.9674	32342	393876	10150
455	460.61	442.24	1204.8	0.01962	0.9882	31668	391427	10001
445	458.43	439.76	1204.8	0.01958	1.0099	30994	388948	9850
435	456.21	437.25	1204.8	0.01954	1.0325	30320	386443	9697
425	453.96	434.69	1204.8	0.01950	1.0561	29646	383910	9542
415	451.67	432.11	1204.8	0.01946	1.0807	28971	381347	9385
405	449.33	429.48	1204.7	0.01942	1.1065	28297	378748	9226
395	446.95	426.80	1204.7	0.01938	1.1335	27623	376113	9065
385	444.53	424.09	1204.6	0.01934	1.1618	26949	373449	8902
375	442.06	421.32	1204.5	0.01929	1.1915	26274	370742	8738
365	439.53	418.51	1204.4	0.01925	1.2227	25600	367998	8571
355	436.96	415.64	1204.3	0.01921	1.2556	24926	365214	8402
345	434.33	412.72	1204.1	0.01916	1.2902	24252	362377	8231
335	431.65	409.74	1204.0	0.01912	1.3267	23578	359498	8058
325	428.91	406.71	1203.8	0.01908	1.3652	22903	356570	7883
315	426.10	403.60	1203.6	0.01903	1.4061	22229	353581	7706
305	423.22	400.43	1203.4	0.01899	1.4493	21555	350536	7527
295	420.27	397.20	1203.1	0.01894	1.4953	20881	347431	7346
285	417.26	393.89	1202.9	0.01889	1.5443	20206	344259	7162
275	414.16	390.50	1202.6	0.01885	1.5964	19532	341014	6977
265	410.97	387.01	1202.3	0.01880	1.6522	18858	337691	6789
255	407.70	383.45	1201.9	0.01875	1.7120	18184	334290	6598
245	404.33	379.79	1201.5	0.01870	1.7762	17510	330798	6406
235	400.87	376.03	1201.1	0.01865	1.8453	16835	327210	6211
225	397.28	372.15	1200.6	0.01860	1.9200	16161	323515	6014
215	393.59	368.16	1200.1	0.01855	2.0010	15487	319707	5814
205	389.76	364.04	1199.6	0.01849	2.0891	14813	315773	5612
200	387.79	361.93	1199.3	0.01847	2.1360	14476	313741	5510
195	385.79	359.78	1199.0	0.01844	2.1852	14138	311672	5408
190	383.76	357.60	1198.6	0.01841	2.2367	13801	309569	5304
185	381.67	355.38	1198.3	0.01839	2.2906	13464	307420	5200
180	379.56	353.10	1198.0	0.01836	2.3472	13127	305227	5096
175	377.40	350.80	1197.6	0.01833	2.4067	12790	302990	4990
170	375.19	348.44	1197.2	0.01830	2.4693	12453	300701	4884
165	372.94	346.04	1196.8	0.01827	2.5353	12116	298363	4778
160	370.64	343.59	1196.4	0.01824	2.6048	11779	295964	4670
155	368.28	341.09	1196.0	0.01821	2.6784	11442	293504	4562
150	365.87	338.52	1195.5	0.01818	2.7562	11104	290979	4453
145	363.40	335.91	1195.1	0.01815	2.8387	10767	288385	4343
140	360.87	333.23	1194.6	0.01812	2.9264	10430	285711	4233

135	358.27	330.48	1194.0	0.01809	3.0197	10093	282959	4122
130	355.61	327.67	1193.5	0.01806	3.1192	9756	280117	4009
125	352.87	324.78	1192.9	0.01802	3.2256	9419	277182	3896
120	350.06	321.82	1192.3	0.01799	3.3396	9082	274140	3783
115	347.15	318.76	1191.7	0.01796	3.4620	8745	270981	3668
110	344.17	315.63	1191.0	0.01792	3.5940	8408	267703	3552
105	341.08	312.39	1190.3	0.01789	3.7365	8070	264279	3436
100	337.89	309.05	1189.6	0.01785	3.8910	7733	260703	3318
95	334.59	305.59	1188.8	0.01781	4.0590	7396	256959	3200
90	331.16	302.02	1188.0	0.01778	4.2425	7059	253026	3080
85	327.60	298.31	1187.1	0.01774	4.4437	6722	248870	2959
80	323.90	294.46	1186.2	0.01770	4.6654	6385	244459	2837
75	320.04	290.44	1185.2	0.01766	4.9109	6048	239747	2715
70	316.01	286.26	1184.1	0.01761	5.1843	5711	234698	2591
65	311.78	281.88	1183.0	0.01757	5.4907	5374	229238	2465
60	307.33	277.28	1181.8	0.01753	5.8367	5036	223300	2339
55	302.64	272.44	1180.5	0.01748	6.2304	4699	216769	2211
50	297.67	267.31	1179.1	0.01743	6.6827	4362	209478	2082

prpv [psig]	erpv [Btu]	dmg_srv [lbm]	mf_sp [lbm]	dt [min.]	t [min.]	qdh [Btu/min.]	esp [Btu]	tsp [°F]
1145	4.79e+008	0.0	6052770	0.0000	2.000	4381320	7.99e+008	164.1
1135	4.77e+008	1867.9	6054638	0.0241	2.024	4374717	8.02e+008	164.4
1125	4.75e+008	1867.4	6056505	0.0243	2.048	4368058	8.04e+008	164.7
1115	4.73e+008	1872.9	6058378	0.0246	2.073	4361320	8.06e+008	165.0
1105	4.71e+008	1878.3	6060256	0.0249	2.098	4354502	8.08e+008	165.4
1095	4.69e+008	1882.5	6062139	0.0252	2.123	4347608	8.10e+008	165.7
1085	4.66e+008	1884.3	6064023	0.0254	2.148	4340644	8.13e+008	166.0
1075	4.64e+008	1891.2	6065914	0.0257	2.174	4333591	8.15e+008	166.3
1065	4.62e+008	1892.1	6067806	0.0260	2.200	4326469	8.17e+008	166.7
1055	4.60e+008	1893.8	6069700	0.0263	2.226	4319274	8.19e+008	167.0
1045	4.58e+008	1904.5	6071605	0.0267	2.253	4311970	8.22e+008	167.3
1035	4.56e+008	1902.0	6073507	0.0269	2.280	4304606	8.24e+008	167.7
1025	4.54e+008	1915.9	6075423	0.0273	2.307	4297117	8.26e+008	168.0
1015	4.51e+008	1913.2	6077336	0.0276	2.335	4289566	8.29e+008	168.3
1005	4.49e+008	1920.4	6079256	0.0279	2.363	4281912	8.31e+008	168.7
995	4.47e+008	1921.6	6081178	0.0282	2.391	4274178	8.33e+008	169.0
985	4.45e+008	1931.8	6083110	0.0287	2.420	4266324	8.35e+008	169.3
975	4.43e+008	1935.5	6085045	0.0290	2.449	4258376	8.38e+008	169.7
965	4.40e+008	1942.2	6086987	0.0294	2.478	4250319	8.40e+008	170.0
955	4.38e+008	1946.6	6088934	0.0298	2.508	4242161	8.42e+008	170.3
945	4.36e+008	1951.7	6090886	0.0302	2.538	4233896	8.45e+008	170.7
935	4.34e+008	1959.8	6092845	0.0306	2.569	4225509	8.47e+008	171.0
925	4.32e+008	1957.6	6094803	0.0309	2.600	4217043	8.49e+008	171.4
915	4.29e+008	1972.5	6096775	0.0315	2.631	4208420	8.52e+008	171.7
905	4.27e+008	1978.8	6098754	0.0319	2.663	4199676	8.54e+008	172.0
895	4.25e+008	1985.1	6100739	0.0324	2.695	4190807	8.56e+008	172.4
885	4.23e+008	1991.3	6102731	0.0328	2.728	4181812	8.59e+008	172.7
875	4.20e+008	1991.7	6104722	0.0332	2.761	4172714	8.61e+008	173.1
865	4.18e+008	2003.5	6106726	0.0338	2.795	4163458	8.64e+008	173.4
855	4.16e+008	2010.5	6108736	0.0343	2.829	4154063	8.66e+008	173.8
845	4.14e+008	2018.1	6110754	0.0348	2.864	4144523	8.68e+008	174.1
835	4.11e+008	2029.6	6112784	0.0354	2.900	4134815	8.71e+008	174.5
825	4.09e+008	2030.8	6114815	0.0359	2.935	4124986	8.73e+008	174.8
815	4.07e+008	2048.4	6116863	0.0366	2.972	4114952	8.76e+008	175.2
805	4.04e+008	2044.1	6118907	0.0370	3.009	4104817	8.78e+008	175.5
795	4.02e+008	2064.3	6120972	0.0378	3.047	4094456	8.81e+008	175.9
785	4.00e+008	2065.8	6123038	0.0383	3.085	4083957	8.83e+008	176.2
775	3.98e+008	2080.1	6125118	0.0391	3.124	4073252	8.86e+008	176.6
765	3.95e+008	2081.6	6127199	0.0396	3.164	4062402	8.88e+008	177.0
755	3.93e+008	2100.0	6129299	0.0405	3.204	4051314	8.91e+008	177.3
745	3.90e+008	2108.0	6131407	0.0412	3.246	4040037	8.93e+008	177.7
735	3.88e+008	2112.9	6133520	0.0418	3.287	4028582	8.96e+008	178.0
725	3.86e+008	2124.3	6135644	0.0426	3.330	4016911	8.98e+008	178.4
715	3.83e+008	2143.1	6137788	0.0436	3.373	4004975	9.01e+008	178.8
705	3.81e+008	2144.9	6139932	0.0442	3.418	3992863	9.03e+008	179.1
695	3.78e+008	2162.5	6142095	0.0452	3.463	3980479	9.06e+008	179.5
685	3.76e+008	2174.3	6144269	0.0461	3.509	3967850	9.09e+008	179.9

675	3.74e+008	2180.6	6146450	0.0469	3.556	3955000	9.11e+008	180.3
665	3.71e+008	2198.6	6148648	0.0480	3.604	3941854	9.14e+008	180.6
655	3.69e+008	2209.2	6150858	0.0489	3.653	3928448	9.17e+008	181.0
645	3.66e+008	2223.2	6153081	0.0500	3.703	3914751	9.19e+008	181.4
635	3.64e+008	2239.1	6155320	0.0511	3.754	3900745	9.22e+008	181.8
625	3.61e+008	2253.9	6157574	0.0523	3.806	3886426	9.25e+008	182.2
615	3.59e+008	2259.6	6159833	0.0532	3.859	3871842	9.27e+008	182.6
605	3.56e+008	2282.9	6162116	0.0546	3.914	3856870	9.30e+008	182.9
595	3.54e+008	2297.2	6164414	0.0559	3.970	3841557	9.33e+008	183.3
585	3.51e+008	2312.6	6166726	0.0572	4.027	3829395	9.36e+008	183.7
575	3.48e+008	2332.1	6169058	0.0587	4.086	3820936	9.38e+008	184.1
565	3.46e+008	2346.4	6171405	0.0600	4.146	3812278	9.41e+008	184.5
555	3.43e+008	2366.6	6173771	0.0616	4.207	3803392	9.44e+008	184.9
545	3.41e+008	2390.1	6176161	0.0633	4.271	3794258	9.47e+008	185.3
535	3.38e+008	2404.2	6178565	0.0649	4.336	3784903	9.50e+008	185.7
525	3.35e+008	2428.9	6180994	0.0668	4.402	3775276	9.53e+008	186.2
515	3.33e+008	2448.9	6183443	0.0686	4.471	3765387	9.56e+008	186.6
505	3.30e+008	2468.4	6185912	0.0704	4.541	3755227	9.59e+008	187.0
495	3.27e+008	2497.3	6188409	0.0727	4.614	3744747	9.62e+008	187.4
485	3.24e+008	2520.7	6190930	0.0748	4.689	3733957	9.65e+008	187.8
475	3.22e+008	2545.1	6193475	0.0771	4.766	3722840	9.68e+008	188.3
465	3.19e+008	2569.0	6196044	0.0794	4.845	3711384	9.71e+008	188.7
455	3.16e+008	2597.9	6198642	0.0820	4.927	3699553	9.74e+008	189.2
445	3.13e+008	2630.4	6201272	0.0849	5.012	3687314	9.77e+008	189.6
435	3.10e+008	2658.0	6203930	0.0877	5.100	3674671	9.80e+008	190.0
425	3.07e+008	2688.3	6206618	0.0907	5.191	3661593	9.84e+008	190.5
415	3.04e+008	2719.4	6209338	0.0939	5.284	3648057	9.87e+008	191.0
405	3.01e+008	2757.7	6212096	0.0975	5.382	3634002	9.90e+008	191.4
395	2.98e+008	2796.4	6214892	0.1012	5.483	3619402	9.94e+008	191.9
385	2.95e+008	2826.6	6217719	0.1049	5.588	3604275	9.97e+008	192.4
375	2.92e+008	2872.0	6220591	0.1093	5.697	3588511	1.00e+009	192.8
365	2.89e+008	2910.5	6223501	0.1137	5.811	3572115	1.00e+009	193.3
355	2.86e+008	2953.3	6226454	0.1185	5.929	3555028	1.01e+009	193.8
345	2.83e+008	3007.0	6229461	0.1240	6.053	3539537	1.01e+009	194.3
335	2.80e+008	3052.4	6232514	0.1295	6.183	3526654	1.01e+009	194.8
325	2.76e+008	3103.3	6235617	0.1355	6.318	3513171	1.02e+009	195.4
315	2.73e+008	3165.3	6238782	0.1424	6.461	3499002	1.02e+009	195.9
305	2.70e+008	3224.1	6242007	0.1496	6.610	3484117	1.03e+009	196.4
295	2.66e+008	3287.2	6245294	0.1574	6.768	3468452	1.03e+009	197.0
285	2.63e+008	3354.9	6248649	0.1660	6.934	3451930	1.03e+009	197.5
275	2.59e+008	3430.8	6252079	0.1756	7.110	3434452	1.04e+009	198.1
265	2.56e+008	3510.5	6255590	0.1862	7.296	3415927	1.04e+009	198.7
255	2.52e+008	3591.9	6259182	0.1975	7.493	3396271	1.05e+009	199.3
245	2.48e+008	3684.3	6262866	0.2104	7.704	3375332	1.05e+009	199.9
235	2.44e+008	3782.7	6266649	0.2247	7.928	3352973	1.06e+009	200.5
225	2.40e+008	3891.5	6270540	0.2408	8.169	3332670	1.06e+009	201.1
215	2.36e+008	4008.3	6274549	0.2588	8.428	3312514	1.07e+009	201.8
205	2.32e+008	4136.1	6278685	0.2792	8.707	3290768	1.07e+009	202.5
200	2.30e+008	2134.1	6280819	0.1474	8.855	3279287	1.07e+009	202.8
195	2.28e+008	2171.4	6282990	0.1536	9.008	3267327	1.08e+009	203.2
190	2.26e+008	2205.8	6285196	0.1598	9.168	3254880	1.08e+009	203.5
185	2.24e+008	2253.6	6287450	0.1674	9.335	3241845	1.08e+009	203.9
180	2.22e+008	2297.7	6289747	0.1750	9.510	3228214	1.08e+009	204.3
175	2.19e+008	2341.9	6292089	0.1831	9.694	3213955	1.09e+009	204.7
170	2.17e+008	2395.7	6294485	0.1924	9.886	3198972	1.09e+009	205.1
165	2.15e+008	2444.7	6296930	0.2018	10.088	3184953	1.09e+009	205.5
160	2.12e+008	2505.8	6299435	0.2127	10.300	3172496	1.10e+009	205.9
155	2.10e+008	2568.8	6302004	0.2245	10.525	3159350	1.10e+009	206.3
150	2.08e+008	2633.2	6304637	0.2371	10.762	3145466	1.10e+009	206.7
145	2.05e+008	2704.5	6307342	0.2512	11.013	3130759	1.10e+009	207.1
140	2.03e+008	2784.6	6310127	0.2670	11.280	3115127	1.11e+009	207.6
135	2.00e+008	2863.4	6312990	0.2837	11.564	3098516	1.11e+009	208.1
130	1.98e+008	2953.5	6315943	0.3027	11.867	3080790	1.11e+009	208.5
125	1.95e+008	3048.4	6318992	0.3236	12.190	3061840	1.12e+009	209.0
120	1.92e+008	3155.5	6322147	0.3475	12.538	3041496	1.12e+009	209.5
115	1.89e+008	3274.0	6325421	0.3744	12.912	3019574	1.13e+009	210.1
110	1.86e+008	3393.6	6328815	0.4036	13.316	2995940	1.13e+009	210.6
105	1.83e+008	3540.2	6332355	0.4387	13.754	2970256	1.13e+009	211.2

100	1.80e+008	3693.7	6336049	0.4776	14.232	2942289	1.14e+009	211.8
95	1.77e+008	3862.4	6339911	0.5222	14.754	2911712	1.14e+009	212.4
90	1.74e+008	4052.4	6343964	0.5741	15.328	2883403	1.15e+009	213.0
85	1.71e+008	4276.6	6348240	0.6362	15.965	2856427	1.15e+009	213.7
80	1.67e+008	4533.1	6352773	0.7100	16.675	2826324	1.16e+009	214.4
75	1.64e+008	4835.3	6357609	0.7995	17.474	2792425	1.16e+009	215.2
70	1.60e+008	5172.7	6362781	0.9058	18.380	2754019	1.17e+009	216.0
65	1.56e+008	5585.0	6368366	1.0393	19.419	2709951	1.18e+009	216.9
60	1.52e+008	6064.3	6374431	1.2041	20.623	2664651	1.18e+009	217.8
55	1.48e+008	6659.2	6381090	1.4171	22.040	2617647	1.19e+009	218.8
50	1.43e+008	7420.5	6388510	1.7011	23.741	2561222	1.20e+009	220.0

torus water level = 13.88 ft

prpv [psig]	trpv [°F]	hf_rpv [Btu/lbm]	hg_rpv [Btu/lbm]	vf_rpv [ft3/lbm]	vg_rpv [ft3/lbm]	wsrv [lbm/min.]	mf_rpv [lbm]	mg_rpv [lbm]
1145	562.88	566.16	1186.6	0.02217	0.3771	78190	532000	15300
1135	561.79	564.74	1187.0	0.02213	0.3808	77515	530124	15309
1125	560.70	563.31	1187.4	0.02210	0.3847	76841	528251	15314
1115	559.60	561.87	1187.8	0.02206	0.3886	76167	526376	15316
1105	558.49	560.42	1188.3	0.02202	0.3926	75493	524498	15315
1095	557.37	558.97	1188.7	0.02199	0.3966	74819	522619	15312
1085	556.24	557.51	1189.1	0.02195	0.4007	74144	520742	15305
1075	555.11	556.04	1189.5	0.02192	0.4049	73470	518860	15295
1065	553.97	554.57	1189.9	0.02188	0.4091	72796	516980	15283
1055	552.82	553.09	1190.3	0.02184	0.4135	72122	515102	15268
1045	551.66	551.60	1190.7	0.02181	0.4179	71447	513216	15250
1035	550.50	550.11	1191.1	0.02177	0.4224	70773	511335	15229
1025	549.32	548.60	1191.4	0.02173	0.4269	70099	509442	15205
1015	548.14	547.09	1191.8	0.02170	0.4316	69425	507556	15179
1005	546.95	545.57	1192.2	0.02166	0.4363	68751	505664	15150
995	545.75	544.04	1192.6	0.02163	0.4412	68076	503775	15118
985	544.54	542.51	1192.9	0.02159	0.4461	67402	501877	15083
975	543.32	540.96	1193.3	0.02155	0.4511	66728	499979	15046
965	542.10	539.40	1193.7	0.02152	0.4562	66054	498076	15007
955	540.86	537.85	1194.0	0.02148	0.4615	65379	496171	14965
945	539.61	536.27	1194.4	0.02145	0.4668	64705	494264	14920
935	538.35	534.70	1194.7	0.02141	0.4722	64031	492352	14873
925	537.09	533.11	1195.1	0.02137	0.4778	63357	490444	14823
915	535.81	531.51	1195.4	0.02134	0.4834	62683	488524	14771
905	534.52	529.90	1195.7	0.02130	0.4892	62008	486600	14716
895	533.22	528.28	1196.1	0.02127	0.4951	61334	484671	14659
885	531.91	526.65	1196.4	0.02123	0.5011	60660	482740	14600
875	530.59	525.02	1196.7	0.02119	0.5072	59986	480810	14538
865	529.26	523.37	1197.0	0.02116	0.5135	59311	478871	14474
855	527.92	521.71	1197.3	0.02112	0.5199	58637	476927	14407
845	526.56	520.04	1197.7	0.02109	0.5265	57963	474977	14338
835	525.20	518.35	1198.0	0.02105	0.5332	57289	473019	14267
825	523.82	516.66	1198.3	0.02101	0.5401	56614	471061	14194
815	522.42	514.95	1198.5	0.02098	0.5471	55940	469088	14118
805	521.02	513.24	1198.8	0.02094	0.5543	55266	467123	14040
795	519.60	511.50	1199.1	0.02090	0.5617	54592	465138	13960
785	518.16	509.76	1199.4	0.02087	0.5692	53918	463155	13878
775	516.72	508.00	1199.7	0.02083	0.5769	53243	461159	13793
765	515.26	506.24	1199.9	0.02080	0.5848	52569	459165	13706
755	513.78	504.45	1200.2	0.02076	0.5929	51895	457153	13618
745	512.30	502.65	1200.4	0.02072	0.6012	51221	455136	13527
735	510.80	500.84	1200.7	0.02069	0.6097	50546	453117	13433
725	509.28	499.01	1200.9	0.02065	0.6185	49872	451088	13338
715	507.74	497.17	1201.2	0.02061	0.6275	49198	449042	13241
705	506.19	495.31	1201.4	0.02058	0.6367	48524	446997	13141
695	504.62	493.43	1201.6	0.02054	0.6461	47850	444936	13039
685	503.03	491.54	1201.8	0.02050	0.6559	47175	442865	12936
675	501.43	489.64	1202.1	0.02047	0.6658	46501	440790	12830
665	499.81	487.71	1202.3	0.02043	0.6761	45827	438699	12722
655	498.17	485.77	1202.5	0.02039	0.6867	45153	436600	12613
645	496.52	483.81	1202.7	0.02036	0.6976	44478	434488	12501
635	494.84	481.83	1202.8	0.02032	0.7088	43804	432363	12387

625	493.14	479.83	1203.0	0.02028	0.7203	43130	430225	12271
615	491.43	477.82	1203.2	0.02024	0.7322	42456	428084	12153
605	489.69	475.78	1203.4	0.02021	0.7445	41782	425920	12033
595	487.93	473.72	1203.5	0.02017	0.7571	41107	423745	11911
585	486.15	471.64	1203.7	0.02013	0.7701	40433	421556	11788
575	484.35	469.53	1203.8	0.02009	0.7836	39759	419350	11662
565	482.52	467.40	1203.9	0.02005	0.7975	39085	417131	11534
555	480.66	465.25	1204.1	0.02002	0.8119	38410	414894	11404
545	478.78	463.07	1204.2	0.01998	0.8268	37736	412636	11273
535	476.88	460.87	1204.3	0.01994	0.8422	37062	410365	11139
525	474.95	458.64	1204.4	0.01990	0.8582	36388	408072	11004
515	472.99	456.39	1204.5	0.01986	0.8747	35714	405760	10866
505	471.01	454.11	1204.5	0.01982	0.8919	35039	403432	10727
495	468.99	451.80	1204.6	0.01978	0.9097	34365	401076	10585
485	466.95	449.45	1204.7	0.01974	0.9282	33691	398698	10442
475	464.87	447.08	1204.7	0.01970	0.9474	33017	396298	10297
465	462.76	444.68	1204.8	0.01966	0.9674	32342	393876	10150
455	460.61	442.24	1204.8	0.01962	0.9882	31668	391427	10001
445	458.43	439.76	1204.8	0.01958	1.0099	30994	388948	9850
435	456.21	437.25	1204.8	0.01954	1.0325	30320	386443	9697
425	453.96	434.69	1204.8	0.01950	1.0561	29646	383910	9542
415	451.67	432.11	1204.8	0.01946	1.0807	28971	381347	9385
405	449.33	429.48	1204.7	0.01942	1.1065	28297	378748	9226
395	446.95	426.80	1204.7	0.01938	1.1335	27623	376113	9065
385	444.53	424.09	1204.6	0.01934	1.1618	26949	373449	8902
375	442.06	421.32	1204.5	0.01929	1.1915	26274	370742	8738
365	439.53	418.51	1204.4	0.01925	1.2227	25600	367998	8571
355	436.96	415.64	1204.3	0.01921	1.2556	24926	365214	8402
345	434.33	412.72	1204.1	0.01916	1.2902	24252	362377	8231
335	431.65	409.74	1204.0	0.01912	1.3267	23578	359498	8058
325	428.91	406.71	1203.8	0.01908	1.3652	22903	356570	7883
315	426.10	403.60	1203.6	0.01903	1.4061	22229	353581	7706
305	423.22	400.43	1203.4	0.01899	1.4493	21555	350536	7527
295	420.27	397.20	1203.1	0.01894	1.4953	20881	347431	7346
285	417.26	393.89	1202.9	0.01889	1.5443	20206	344259	7162
275	414.16	390.50	1202.6	0.01885	1.5964	19532	341014	6977
265	410.97	387.01	1202.3	0.01880	1.6522	18858	337691	6789
255	407.70	383.45	1201.9	0.01875	1.7120	18184	334290	6598
245	404.33	379.79	1201.5	0.01870	1.7762	17510	330798	6406
235	400.87	376.03	1201.1	0.01865	1.8453	16835	327210	6211
225	397.28	372.15	1200.6	0.01860	1.9200	16161	323515	6014
215	393.59	368.16	1200.1	0.01855	2.0010	15487	319707	5814
205	389.76	364.04	1199.6	0.01849	2.0891	14813	315773	5612
200	387.79	361.93	1199.3	0.01847	2.1360	14476	313741	5510
195	385.79	359.78	1199.0	0.01844	2.1852	14138	311672	5408
190	383.76	357.60	1198.6	0.01841	2.2367	13801	309569	5304
185	381.67	355.38	1198.3	0.01839	2.2906	13464	307420	5200
180	379.56	353.10	1198.0	0.01836	2.3472	13127	305227	5096
175	377.40	350.80	1197.6	0.01833	2.4067	12790	302990	4990
170	375.19	348.44	1197.2	0.01830	2.4693	12453	300701	4884
165	372.94	346.04	1196.8	0.01827	2.5353	12116	298363	4778
160	370.64	343.59	1196.4	0.01824	2.6048	11779	295964	4670
155	368.28	341.09	1196.0	0.01821	2.6784	11442	293504	4562
150	365.87	338.52	1195.5	0.01818	2.7562	11104	290979	4453
145	363.40	335.91	1195.1	0.01815	2.8387	10767	288385	4343
140	360.87	333.23	1194.6	0.01812	2.9264	10430	285711	4233
135	358.27	330.48	1194.0	0.01809	3.0197	10093	282959	4122
130	355.61	327.67	1193.5	0.01806	3.1192	9756	280117	4009
125	352.87	324.78	1192.9	0.01802	3.2256	9419	277182	3896
120	350.06	321.82	1192.3	0.01799	3.3396	9082	274140	3783
115	347.15	318.76	1191.7	0.01796	3.4620	8745	270981	3668
110	344.17	315.63	1191.0	0.01792	3.5940	8408	267703	3552
105	341.08	312.39	1190.3	0.01789	3.7365	8070	264279	3436
100	337.89	309.05	1189.6	0.01785	3.8910	7733	260703	3318
95	334.59	305.59	1188.8	0.01781	4.0590	7396	256959	3200
90	331.16	302.02	1188.0	0.01778	4.2425	7059	253026	3080
85	327.60	298.31	1187.1	0.01774	4.4437	6722	248870	2959
80	323.90	294.46	1186.2	0.01770	4.6654	6385	244459	2837

75	320.04	290.44	1185.2	0.01766	4.9109	6048	239747	2715
70	316.01	286.26	1184.1	0.01761	5.1843	5711	234698	2591
65	311.78	281.88	1183.0	0.01757	5.4907	5374	229238	2465
60	307.33	277.28	1181.8	0.01753	5.8367	5036	223300	2339
55	302.64	272.44	1180.5	0.01748	6.2304	4699	216769	2211
50	297.67	267.31	1179.1	0.01743	6.6827	4362	209478	2082

prpv [psig]	erpv [Btu]	dmg_srv [lbm]	mf_sp [lbm]	dt [min.]	t [min.]	qdh [Btu/min.]	esp [Btu]	tsp [°F]
1145	4.79e+008	0.0	6584924	0.0000	2.000	4381320	8.99e+008	168.6
1135	4.77e+008	1867.9	6586792	0.0241	2.024	4374717	9.02e+008	168.9
1125	4.75e+008	1867.4	6588659	0.0243	2.048	4368058	9.04e+008	169.2
1115	4.73e+008	1872.9	6590532	0.0246	2.073	4361320	9.06e+008	169.5
1105	4.71e+008	1878.3	6592410	0.0249	2.098	4354502	9.08e+008	169.8
1095	4.69e+008	1882.5	6594293	0.0252	2.123	4347608	9.10e+008	170.1
1085	4.66e+008	1884.3	6596177	0.0254	2.148	4340644	9.13e+008	170.4
1075	4.64e+008	1891.2	6598068	0.0257	2.174	4333591	9.15e+008	170.7
1065	4.62e+008	1892.1	6599960	0.0260	2.200	4326469	9.17e+008	171.0
1055	4.60e+008	1893.8	6601854	0.0263	2.226	4319274	9.19e+008	171.3
1045	4.58e+008	1904.5	6603759	0.0267	2.253	4311970	9.22e+008	171.6
1035	4.56e+008	1902.0	6605661	0.0269	2.280	4304606	9.24e+008	171.9
1025	4.54e+008	1915.9	6607576	0.0273	2.307	4297117	9.26e+008	172.2
1015	4.51e+008	1913.2	6609490	0.0276	2.335	4289566	9.29e+008	172.5
1005	4.49e+008	1920.4	6611410	0.0279	2.363	4281912	9.31e+008	172.8
995	4.47e+008	1921.6	6613332	0.0282	2.391	4274178	9.33e+008	173.1
985	4.45e+008	1931.8	6615263	0.0287	2.420	4266324	9.35e+008	173.4
975	4.43e+008	1935.5	6617199	0.0290	2.449	4258376	9.38e+008	173.7
965	4.40e+008	1942.2	6619141	0.0294	2.478	4250319	9.40e+008	174.0
955	4.38e+008	1946.6	6621088	0.0298	2.508	4242161	9.42e+008	174.3
945	4.36e+008	1951.7	6623039	0.0302	2.538	4233896	9.45e+008	174.6
935	4.34e+008	1959.8	6624999	0.0306	2.569	4225509	9.47e+008	175.0
925	4.32e+008	1957.6	6626957	0.0309	2.600	4217043	9.49e+008	175.3
915	4.29e+008	1972.5	6628929	0.0315	2.631	4208420	9.52e+008	175.6
905	4.27e+008	1978.8	6630908	0.0319	2.663	4199676	9.54e+008	175.9
895	4.25e+008	1985.1	6632893	0.0324	2.695	4190807	9.57e+008	176.2
885	4.23e+008	1991.3	6634884	0.0328	2.728	4181812	9.59e+008	176.5
875	4.20e+008	1991.7	6636876	0.0332	2.761	4172714	9.61e+008	176.8
865	4.18e+008	2003.5	6638880	0.0338	2.795	4163458	9.64e+008	177.2
855	4.16e+008	2010.5	6640890	0.0343	2.829	4154063	9.66e+008	177.5
845	4.14e+008	2018.1	6642908	0.0348	2.864	4144523	9.68e+008	177.8
835	4.11e+008	2029.6	6644938	0.0354	2.900	4134815	9.71e+008	178.1
825	4.09e+008	2030.8	6646969	0.0359	2.935	4124986	9.73e+008	178.4
815	4.07e+008	2048.4	6649017	0.0366	2.972	4114952	9.76e+008	178.8
805	4.04e+008	2044.1	6651061	0.0370	3.009	4104817	9.78e+008	179.1
795	4.02e+008	2064.3	6653126	0.0378	3.047	4094456	9.81e+008	179.4
785	4.00e+008	2065.8	6655191	0.0383	3.085	4083957	9.83e+008	179.7
775	3.98e+008	2080.1	6657271	0.0391	3.124	4073252	9.86e+008	180.1
765	3.95e+008	2081.6	6659353	0.0396	3.164	4062402	9.88e+008	180.4
755	3.93e+008	2100.0	6661453	0.0405	3.204	4051314	9.91e+008	180.7
745	3.90e+008	2108.0	6663561	0.0412	3.246	4040037	9.93e+008	181.1
735	3.88e+008	2112.9	6665674	0.0418	3.287	4028582	9.96e+008	181.4
725	3.86e+008	2124.3	6667798	0.0426	3.330	4016911	9.98e+008	181.7
715	3.83e+008	2143.1	6669941	0.0436	3.373	4004975	1.00e+009	182.1
705	3.81e+008	2144.9	6672086	0.0442	3.418	3992863	1.00e+009	182.4
695	3.78e+008	2162.5	6674249	0.0452	3.463	3980479	1.01e+009	182.7
685	3.76e+008	2174.3	6676423	0.0461	3.509	3967850	1.01e+009	183.1
675	3.74e+008	2180.6	6678604	0.0469	3.556	3955000	1.01e+009	183.4
665	3.71e+008	2198.6	6680802	0.0480	3.604	3941854	1.01e+009	183.8
655	3.69e+008	2209.2	6683011	0.0489	3.653	3928448	1.02e+009	184.1
645	3.66e+008	2223.2	6685235	0.0500	3.703	3914751	1.02e+009	184.5
635	3.64e+008	2239.1	6687474	0.0511	3.754	3900745	1.02e+009	184.8
625	3.61e+008	2253.9	6689728	0.0523	3.806	3886426	1.02e+009	185.2
615	3.59e+008	2259.6	6691987	0.0532	3.859	3871842	1.03e+009	185.5
605	3.56e+008	2282.9	6694270	0.0546	3.914	3856870	1.03e+009	185.9
595	3.54e+008	2297.2	6696567	0.0559	3.970	3841557	1.03e+009	186.2
585	3.51e+008	2312.6	6698880	0.0572	4.027	3829395	1.04e+009	186.6
575	3.48e+008	2332.1	6701212	0.0587	4.086	3820936	1.04e+009	187.0
565	3.46e+008	2346.4	6703558	0.0600	4.146	3812278	1.04e+009	187.3

555	3.43e+008	2366.6	6705925	0.0616	4.207	3803392	1.04e+009	187.7
545	3.41e+008	2390.1	6708315	0.0633	4.271	3794258	1.05e+009	188.1
535	3.38e+008	2404.2	6710719	0.0649	4.336	3784903	1.05e+009	188.5
525	3.35e+008	2428.9	6713148	0.0668	4.402	3775276	1.05e+009	188.8
515	3.33e+008	2448.9	6715597	0.0686	4.471	3765387	1.06e+009	189.2
505	3.30e+008	2468.4	6718066	0.0704	4.541	3755227	1.06e+009	189.6
495	3.27e+008	2497.3	6720563	0.0727	4.614	3744747	1.06e+009	190.0
485	3.24e+008	2520.7	6723084	0.0748	4.689	3733957	1.06e+009	190.4
475	3.22e+008	2545.1	6725629	0.0771	4.766	3722840	1.07e+009	190.8
465	3.19e+008	2569.0	6728198	0.0794	4.845	3711384	1.07e+009	191.2
455	3.16e+008	2597.9	6730796	0.0820	4.927	3699553	1.07e+009	191.6
445	3.13e+008	2630.4	6733426	0.0849	5.012	3687314	1.08e+009	192.0
435	3.10e+008	2658.0	6736084	0.0877	5.100	3674671	1.08e+009	192.4
425	3.07e+008	2688.3	6738772	0.0907	5.191	3661593	1.08e+009	192.8
415	3.04e+008	2719.4	6741492	0.0939	5.284	3648057	1.09e+009	193.2
405	3.01e+008	2757.7	6744249	0.0975	5.382	3634002	1.09e+009	193.7
395	2.98e+008	2796.4	6747046	0.1012	5.483	3619402	1.09e+009	194.1
385	2.95e+008	2826.6	6749872	0.1049	5.588	3604275	1.10e+009	194.5
375	2.92e+008	2872.0	6752744	0.1093	5.697	3588511	1.10e+009	195.0
365	2.89e+008	2910.5	6755655	0.1137	5.811	3572115	1.10e+009	195.4
355	2.86e+008	2953.3	6758608	0.1185	5.929	3555028	1.11e+009	195.9
345	2.83e+008	3007.0	6761615	0.1240	6.053	3539537	1.11e+009	196.4
335	2.80e+008	3052.4	6764668	0.1295	6.183	3526654	1.11e+009	196.8
325	2.76e+008	3103.3	6767771	0.1355	6.318	3513171	1.12e+009	197.3
315	2.73e+008	3165.3	6770936	0.1424	6.461	3499002	1.12e+009	197.8
305	2.70e+008	3224.1	6774160	0.1496	6.610	3484117	1.13e+009	198.3
295	2.66e+008	3287.2	6777448	0.1574	6.768	3468452	1.13e+009	198.8
285	2.63e+008	3354.9	6780802	0.1660	6.934	3451930	1.13e+009	199.3
275	2.59e+008	3430.8	6784233	0.1756	7.110	3434452	1.14e+009	199.8
265	2.56e+008	3510.5	6787744	0.1862	7.296	3415927	1.14e+009	200.4
255	2.52e+008	3591.9	6791336	0.1975	7.493	3396271	1.15e+009	200.9
245	2.48e+008	3684.3	6795020	0.2104	7.704	3375332	1.15e+009	201.5
235	2.44e+008	3782.7	6798803	0.2247	7.928	3352973	1.16e+009	202.0
225	2.40e+008	3891.5	6802694	0.2408	8.169	3332670	1.16e+009	202.6
215	2.36e+008	4008.3	6806703	0.2588	8.428	3312514	1.17e+009	203.2
205	2.32e+008	4136.1	6810839	0.2792	8.707	3290768	1.17e+009	203.9
200	2.30e+008	2134.1	6812973	0.1474	8.855	3279287	1.17e+009	204.2
195	2.28e+008	2171.4	6815144	0.1536	9.008	3267327	1.18e+009	204.5
190	2.26e+008	2205.8	6817350	0.1598	9.168	3254880	1.18e+009	204.8
185	2.24e+008	2253.6	6819604	0.1674	9.335	3241845	1.18e+009	205.2
180	2.22e+008	2297.7	6821901	0.1750	9.510	3228214	1.18e+009	205.5
175	2.19e+008	2341.9	6824243	0.1831	9.694	3213955	1.19e+009	205.9
170	2.17e+008	2395.7	6826639	0.1924	9.886	3198972	1.19e+009	206.2
165	2.15e+008	2444.7	6829083	0.2018	10.088	3184953	1.19e+009	206.6
160	2.12e+008	2505.8	6831589	0.2127	10.300	3172496	1.20e+009	207.0
155	2.10e+008	2568.8	6834158	0.2245	10.525	3159350	1.20e+009	207.4
150	2.08e+008	2633.2	6836791	0.2371	10.762	3145466	1.20e+009	207.7
145	2.05e+008	2704.5	6839496	0.2512	11.013	3130759	1.20e+009	208.1
140	2.03e+008	2784.6	6842280	0.2670	11.280	3115127	1.21e+009	208.6
135	2.00e+008	2863.4	6845144	0.2837	11.564	3098516	1.21e+009	209.0
130	1.98e+008	2953.5	6848097	0.3027	11.867	3080790	1.22e+009	209.4
125	1.95e+008	3048.4	6851146	0.3236	12.190	3061840	1.22e+009	209.9
120	1.92e+008	3155.5	6854301	0.3475	12.538	3041496	1.22e+009	210.3
115	1.89e+008	3274.0	6857575	0.3744	12.912	3019574	1.23e+009	210.8
110	1.86e+008	3393.6	6860969	0.4036	13.316	2995940	1.23e+009	211.3
105	1.83e+008	3540.2	6864509	0.4387	13.754	2970256	1.23e+009	211.9
100	1.80e+008	3693.7	6868203	0.4776	14.232	2942289	1.24e+009	212.4
95	1.77e+008	3862.4	6872065	0.5222	14.754	2911712	1.24e+009	213.0
90	1.74e+008	4052.4	6876117	0.5741	15.328	2883403	1.25e+009	213.6
85	1.71e+008	4276.6	6880394	0.6362	15.965	2856427	1.25e+009	214.2
80	1.67e+008	4533.1	6884927	0.7100	16.675	2826324	1.26e+009	214.8
75	1.64e+008	4835.3	6889762	0.7995	17.474	2792425	1.26e+009	215.5
70	1.60e+008	5172.7	6894935	0.9058	18.380	2754019	1.27e+009	216.3
65	1.56e+008	5585.0	6900520	1.0393	19.419	2709951	1.28e+009	217.1
60	1.52e+008	6064.3	6906585	1.2041	20.623	2664651	1.28e+009	218.0
55	1.48e+008	6659.2	6913244	1.4171	22.040	2617647	1.29e+009	218.9
50	1.43e+008	7420.5	6920664	1.7011	23.741	2561222	1.30e+009	220.0

torus water level = 14.00 ft

prpv [psig]	trpv [°F]	hf_rpv [Btu/lbm]	hg_rpv [Btu/lbm]	vf_rpv [ft3/lbm]	vg_rpv [ft3/lbm]	wsrv [lbm/min.]	mf_rpv [lbm]	mg_rpv [lbm]
1145	562.88	566.16	1186.6	0.02217	0.3771	78190	532000	15300
1135	561.79	564.74	1187.0	0.02213	0.3808	77515	530124	15309
1125	560.70	563.31	1187.4	0.02210	0.3847	76841	528251	15314
1115	559.60	561.87	1187.8	0.02206	0.3886	76167	526376	15316
1105	558.49	560.42	1188.3	0.02202	0.3926	75493	524498	15315
1095	557.37	558.97	1188.7	0.02199	0.3966	74819	522619	15312
1085	556.24	557.51	1189.1	0.02195	0.4007	74144	520742	15305
1075	555.11	556.04	1189.5	0.02192	0.4049	73470	518860	15295
1065	553.97	554.57	1189.9	0.02188	0.4091	72796	516980	15283
1055	552.82	553.09	1190.3	0.02184	0.4135	72122	515102	15268
1045	551.66	551.60	1190.7	0.02181	0.4179	71447	513216	15250
1035	550.50	550.11	1191.1	0.02177	0.4224	70773	511335	15229
1025	549.32	548.60	1191.4	0.02173	0.4269	70099	509442	15205
1015	548.14	547.09	1191.8	0.02170	0.4316	69425	507556	15179
1005	546.95	545.57	1192.2	0.02166	0.4363	68751	505664	15150
995	545.75	544.04	1192.6	0.02163	0.4412	68076	503775	15118
985	544.54	542.51	1192.9	0.02159	0.4461	67402	501877	15083
975	543.32	540.96	1193.3	0.02155	0.4511	66728	499979	15046
965	542.10	539.40	1193.7	0.02152	0.4562	66054	498076	15007
955	540.86	537.85	1194.0	0.02148	0.4615	65379	496171	14965
945	539.61	536.27	1194.4	0.02145	0.4668	64705	494264	14920
935	538.35	534.70	1194.7	0.02141	0.4722	64031	492352	14873
925	537.09	533.11	1195.1	0.02137	0.4778	63357	490444	14823
915	535.81	531.51	1195.4	0.02134	0.4834	62683	488524	14771
905	534.52	529.90	1195.7	0.02130	0.4892	62008	486600	14716
895	533.22	528.28	1196.1	0.02127	0.4951	61334	484671	14659
885	531.91	526.65	1196.4	0.02123	0.5011	60660	482740	14600
875	530.59	525.02	1196.7	0.02119	0.5072	59986	480810	14538
865	529.26	523.37	1197.0	0.02116	0.5135	59311	478871	14474
855	527.92	521.71	1197.3	0.02112	0.5199	58637	476927	14407
845	526.56	520.04	1197.7	0.02109	0.5265	57963	474977	14338
835	525.20	518.35	1198.0	0.02105	0.5332	57289	473019	14267
825	523.82	516.66	1198.3	0.02101	0.5401	56614	471061	14194
815	522.42	514.95	1198.5	0.02098	0.5471	55940	469088	14118
805	521.02	513.24	1198.8	0.02094	0.5543	55266	467123	14040
795	519.60	511.50	1199.1	0.02090	0.5617	54592	465138	13960
785	518.16	509.76	1199.4	0.02087	0.5692	53918	463155	13878
775	516.72	508.00	1199.7	0.02083	0.5769	53243	461159	13793
765	515.26	506.24	1199.9	0.02080	0.5848	52569	459165	13706
755	513.78	504.45	1200.2	0.02076	0.5929	51895	457153	13618
745	512.30	502.65	1200.4	0.02072	0.6012	51221	455136	13527
735	510.80	500.84	1200.7	0.02069	0.6097	50546	453117	13433
725	509.28	499.01	1200.9	0.02065	0.6185	49872	451088	13338
715	507.74	497.17	1201.2	0.02061	0.6275	49198	449042	13241
705	506.19	495.31	1201.4	0.02058	0.6367	48524	446997	13141
695	504.62	493.43	1201.6	0.02054	0.6461	47850	444936	13039
685	503.03	491.54	1201.8	0.02050	0.6559	47175	442865	12936
675	501.43	489.64	1202.1	0.02047	0.6658	46501	440790	12830
665	499.81	487.71	1202.3	0.02043	0.6761	45827	438699	12722
655	498.17	485.77	1202.5	0.02039	0.6867	45153	436600	12613
645	496.52	483.81	1202.7	0.02036	0.6976	44478	434488	12501
635	494.84	481.83	1202.8	0.02032	0.7088	43804	432363	12387
625	493.14	479.83	1203.0	0.02028	0.7203	43130	430225	12271
615	491.43	477.82	1203.2	0.02024	0.7322	42456	428084	12153
605	489.69	475.78	1203.4	0.02021	0.7445	41782	425920	12033
595	487.93	473.72	1203.5	0.02017	0.7571	41107	423745	11911
585	486.15	471.64	1203.7	0.02013	0.7701	40433	421556	11788
575	484.35	469.53	1203.8	0.02009	0.7836	39759	419350	11662
565	482.52	467.40	1203.9	0.02005	0.7975	39085	417131	11534
555	480.66	465.25	1204.1	0.02002	0.8119	38410	414894	11404
545	478.78	463.07	1204.2	0.01998	0.8268	37736	412636	11273
535	476.88	460.87	1204.3	0.01994	0.8422	37062	410365	11139
525	474.95	458.64	1204.4	0.01990	0.8582	36388	408072	11004
515	472.99	456.39	1204.5	0.01986	0.8747	35714	405760	10866

505	471.01	454.11	1204.5	0.01982	0.8919	35039	403432	10727
495	468.99	451.80	1204.6	0.01978	0.9097	34365	401076	10585
485	466.95	449.45	1204.7	0.01974	0.9282	33691	398698	10442
475	464.87	447.08	1204.7	0.01970	0.9474	33017	396298	10297
465	462.76	444.68	1204.8	0.01966	0.9674	32342	393876	10150
455	460.61	442.24	1204.8	0.01962	0.9882	31668	391427	10001
445	458.43	439.76	1204.8	0.01958	1.0099	30994	388948	9850
435	456.21	437.25	1204.8	0.01954	1.0325	30320	386443	9697
425	453.96	434.69	1204.8	0.01950	1.0561	29646	383910	9542
415	451.67	432.11	1204.8	0.01946	1.0807	28971	381347	9385
405	449.33	429.48	1204.7	0.01942	1.1065	28297	378748	9226
395	446.95	426.80	1204.7	0.01938	1.1335	27623	376113	9065
385	444.53	424.09	1204.6	0.01934	1.1618	26949	373449	8902
375	442.06	421.32	1204.5	0.01929	1.1915	26274	370742	8738
365	439.53	418.51	1204.4	0.01925	1.2227	25600	367998	8571
355	436.96	415.64	1204.3	0.01921	1.2556	24926	365214	8402
345	434.33	412.72	1204.1	0.01916	1.2902	24252	362377	8231
335	431.65	409.74	1204.0	0.01912	1.3267	23578	359498	8058
325	428.91	406.71	1203.8	0.01908	1.3652	22903	356570	7883
315	426.10	403.60	1203.6	0.01903	1.4061	22229	353581	7706
305	423.22	400.43	1203.4	0.01899	1.4493	21555	350536	7527
295	420.27	397.20	1203.1	0.01894	1.4953	20881	347431	7346
285	417.26	393.89	1202.9	0.01889	1.5443	20206	344259	7162
275	414.16	390.50	1202.6	0.01885	1.5964	19532	341014	6977
265	410.97	387.01	1202.3	0.01880	1.6522	18858	337691	6789
255	407.70	383.45	1201.9	0.01875	1.7120	18184	334290	6598
245	404.33	379.79	1201.5	0.01870	1.7762	17510	330798	6406
235	400.87	376.03	1201.1	0.01865	1.8453	16835	327210	6211
225	397.28	372.15	1200.6	0.01860	1.9200	16161	323515	6014
215	393.59	368.16	1200.1	0.01855	2.0010	15487	319707	5814
205	389.76	364.04	1199.6	0.01849	2.0891	14813	315773	5612
200	387.79	361.93	1199.3	0.01847	2.1360	14476	313741	5510
195	385.79	359.78	1199.0	0.01844	2.1852	14138	311672	5408
190	383.76	357.60	1198.6	0.01841	2.2367	13801	309569	5304
185	381.67	355.38	1198.3	0.01839	2.2906	13464	307420	5200
180	379.56	353.10	1198.0	0.01836	2.3472	13127	305227	5096
175	377.40	350.80	1197.6	0.01833	2.4067	12790	302990	4990
170	375.19	348.44	1197.2	0.01830	2.4693	12453	300701	4884
165	372.94	346.04	1196.8	0.01827	2.5353	12116	298363	4778
160	370.64	343.59	1196.4	0.01824	2.6048	11779	295964	4670
155	368.28	341.09	1196.0	0.01821	2.6784	11442	293504	4562
150	365.87	338.52	1195.5	0.01818	2.7562	11104	290979	4453
145	363.40	335.91	1195.1	0.01815	2.8387	10767	288385	4343
140	360.87	333.23	1194.6	0.01812	2.9264	10430	285711	4233
135	358.27	330.48	1194.0	0.01809	3.0197	10093	282959	4122
130	355.61	327.67	1193.5	0.01806	3.1192	9756	280117	4009
125	352.87	324.78	1192.9	0.01802	3.2256	9419	277182	3896
120	350.06	321.82	1192.3	0.01799	3.3396	9082	274140	3783
115	347.15	318.76	1191.7	0.01796	3.4620	8745	270981	3668
110	344.17	315.63	1191.0	0.01792	3.5940	8408	267703	3552
105	341.08	312.39	1190.3	0.01789	3.7365	8070	264279	3436
100	337.89	309.05	1189.6	0.01785	3.8910	7733	260703	3318
95	334.59	305.59	1188.8	0.01781	4.0590	7396	256959	3200
90	331.16	302.02	1188.0	0.01778	4.2425	7059	253026	3080
85	327.60	298.31	1187.1	0.01774	4.4437	6722	248870	2959
80	323.90	294.46	1186.2	0.01770	4.6654	6385	244459	2837
75	320.04	290.44	1185.2	0.01766	4.9109	6048	239747	2715
70	316.01	286.26	1184.1	0.01761	5.1843	5711	234698	2591
65	311.78	281.88	1183.0	0.01757	5.4907	5374	229238	2465
60	307.33	277.28	1181.8	0.01753	5.8367	5036	223300	2339
55	302.64	272.44	1180.5	0.01748	6.2304	4699	216769	2211
50	297.67	267.31	1179.1	0.01743	6.6827	4362	209478	2082

prpv [psig]	erpv [Btu]	dmg_srv [lbm]	mf_sp [lbm]	dt [min.]	t [min.]	qdh [Btu/min.]	esp [Btu]	tsp [°F]
1145	4.79e+008	0.0	6657738	0.0000	2.000	4381320	9.13e+008	169.1
1135	4.77e+008	1867.9	6659606	0.0241	2.024	4374717	9.15e+008	169.4
1125	4.75e+008	1867.4	6661473	0.0243	2.048	4368058	9.17e+008	169.7

1115	4.73e+008	1872.9	6663346	0.0246	2.073	4361320	9.20e+008	170.0
1105	4.71e+008	1878.3	6665224	0.0249	2.098	4354502	9.22e+008	170.3
1095	4.69e+008	1882.5	6667107	0.0252	2.123	4347608	9.24e+008	170.6
1085	4.66e+008	1884.3	6668991	0.0254	2.148	4340644	9.26e+008	170.9
1075	4.64e+008	1891.2	6670882	0.0257	2.174	4333591	9.29e+008	171.2
1065	4.62e+008	1892.1	6672774	0.0260	2.200	4326469	9.31e+008	171.5
1055	4.60e+008	1893.8	6674668	0.0263	2.226	4319274	9.33e+008	171.8
1045	4.58e+008	1904.5	6676572	0.0267	2.253	4311970	9.35e+008	172.1
1035	4.56e+008	1902.0	6678474	0.0269	2.280	4304606	9.38e+008	172.4
1025	4.54e+008	1915.9	6680390	0.0273	2.307	4297117	9.40e+008	172.7
1015	4.51e+008	1913.2	6682303	0.0276	2.335	4289566	9.42e+008	173.0
1005	4.49e+008	1920.4	6684224	0.0279	2.363	4281912	9.45e+008	173.3
995	4.47e+008	1921.6	6686145	0.0282	2.391	4274178	9.47e+008	173.6
985	4.45e+008	1931.8	6688077	0.0287	2.420	4266324	9.49e+008	173.9
975	4.43e+008	1935.5	6690013	0.0290	2.449	4258376	9.51e+008	174.2
965	4.40e+008	1942.2	6691955	0.0294	2.478	4250319	9.54e+008	174.5
955	4.38e+008	1946.6	6693902	0.0298	2.508	4242161	9.56e+008	174.8
945	4.36e+008	1951.7	6695853	0.0302	2.538	4233896	9.58e+008	175.1
935	4.34e+008	1959.8	6697813	0.0306	2.569	4225509	9.61e+008	175.4
925	4.32e+008	1957.6	6699771	0.0309	2.600	4217043	9.63e+008	175.8
915	4.29e+008	1972.5	6701743	0.0315	2.631	4208420	9.65e+008	176.1
905	4.27e+008	1978.8	6703722	0.0319	2.663	4199676	9.68e+008	176.4
895	4.25e+008	1985.1	6705707	0.0324	2.695	4190807	9.70e+008	176.7
885	4.23e+008	1991.3	6707698	0.0328	2.728	4181812	9.73e+008	177.0
875	4.20e+008	1991.7	6709690	0.0332	2.761	4172714	9.75e+008	177.3
865	4.18e+008	2003.5	6711694	0.0338	2.795	4163458	9.77e+008	177.6
855	4.16e+008	2010.5	6713704	0.0343	2.829	4154063	9.80e+008	177.9
845	4.14e+008	2018.1	6715722	0.0348	2.864	4144523	9.82e+008	178.3
835	4.11e+008	2029.6	6717752	0.0354	2.900	4134815	9.85e+008	178.6
825	4.09e+008	2030.8	6719783	0.0359	2.935	4124986	9.87e+008	178.9
815	4.07e+008	2048.4	6721831	0.0366	2.972	4114952	9.89e+008	179.2
805	4.04e+008	2044.1	6723875	0.0370	3.009	4104817	9.92e+008	179.5
795	4.02e+008	2064.3	6725939	0.0378	3.047	4094456	9.94e+008	179.8
785	4.00e+008	2065.8	6728005	0.0383	3.085	4083957	9.97e+008	180.2
775	3.98e+008	2080.1	6730085	0.0391	3.124	4073252	9.99e+008	180.5
765	3.95e+008	2081.6	6732167	0.0396	3.164	4062402	1.00e+009	180.8
755	3.93e+008	2100.0	6734267	0.0405	3.204	4051314	1.00e+009	181.2
745	3.90e+008	2108.0	6736375	0.0412	3.246	4040037	1.01e+009	181.5
735	3.88e+008	2112.9	6738488	0.0418	3.287	4028582	1.01e+009	181.8
725	3.86e+008	2124.3	6740612	0.0426	3.330	4016911	1.01e+009	182.1
715	3.83e+008	2143.1	6742755	0.0436	3.373	4004975	1.01e+009	182.5
705	3.81e+008	2144.9	6744900	0.0442	3.418	3992863	1.02e+009	182.8
695	3.78e+008	2162.5	6747063	0.0452	3.463	3980479	1.02e+009	183.1
685	3.76e+008	2174.3	6749237	0.0461	3.509	3967850	1.02e+009	183.5
675	3.74e+008	2180.6	6751418	0.0469	3.556	3955000	1.03e+009	183.8
665	3.71e+008	2198.6	6753616	0.0480	3.604	3941854	1.03e+009	184.2
655	3.69e+008	2209.2	6755825	0.0489	3.653	3928448	1.03e+009	184.5
645	3.66e+008	2223.2	6758049	0.0500	3.703	3914751	1.03e+009	184.9
635	3.64e+008	2239.1	6760288	0.0511	3.754	3900745	1.04e+009	185.2
625	3.61e+008	2253.9	6762542	0.0523	3.806	3886426	1.04e+009	185.6
615	3.59e+008	2259.6	6764801	0.0532	3.859	3871842	1.04e+009	185.9
605	3.56e+008	2282.9	6767084	0.0546	3.914	3856870	1.04e+009	186.3
595	3.54e+008	2297.2	6769381	0.0559	3.970	3841557	1.05e+009	186.6
585	3.51e+008	2312.6	6771694	0.0572	4.027	3829395	1.05e+009	187.0
575	3.48e+008	2332.1	6774026	0.0587	4.086	3820936	1.05e+009	187.3
565	3.46e+008	2346.4	6776372	0.0600	4.146	3812278	1.06e+009	187.7
555	3.43e+008	2366.6	6778739	0.0616	4.207	3803392	1.06e+009	188.1
545	3.41e+008	2390.1	6781129	0.0633	4.271	3794258	1.06e+009	188.4
535	3.38e+008	2404.2	6783533	0.0649	4.336	3784903	1.06e+009	188.8
525	3.35e+008	2428.9	6785962	0.0668	4.402	3775276	1.07e+009	189.2
515	3.33e+008	2448.9	6788411	0.0686	4.471	3765387	1.07e+009	189.6
505	3.30e+008	2468.4	6790879	0.0704	4.541	3755227	1.07e+009	189.9
495	3.27e+008	2497.3	6793377	0.0727	4.614	3744747	1.08e+009	190.3
485	3.24e+008	2520.7	6795897	0.0748	4.689	3733957	1.08e+009	190.7
475	3.22e+008	2545.1	6798443	0.0771	4.766	3722840	1.08e+009	191.1
465	3.19e+008	2569.0	6801012	0.0794	4.845	3711384	1.08e+009	191.5
455	3.16e+008	2597.9	6803609	0.0820	4.927	3699553	1.09e+009	191.9
445	3.13e+008	2630.4	6806240	0.0849	5.012	3687314	1.09e+009	192.3

435	3.10e+008	2658.0	6808898	0.0877	5.100	3674671	1.09e+009	192.7
425	3.07e+008	2688.3	6811586	0.0907	5.191	3661593	1.10e+009	193.1
415	3.04e+008	2719.4	6814306	0.0939	5.284	3648057	1.10e+009	193.5
405	3.01e+008	2757.7	6817063	0.0975	5.382	3634002	1.10e+009	194.0
395	2.98e+008	2796.4	6819860	0.1012	5.483	3619402	1.11e+009	194.4
385	2.95e+008	2826.6	6822686	0.1049	5.588	3604275	1.11e+009	194.8
375	2.92e+008	2872.0	6825558	0.1093	5.697	3588511	1.11e+009	195.3
365	2.89e+008	2910.5	6828469	0.1137	5.811	3572115	1.12e+009	195.7
355	2.86e+008	2953.3	6831422	0.1185	5.929	3555028	1.12e+009	196.1
345	2.83e+008	3007.0	6834429	0.1240	6.053	3539537	1.12e+009	196.6
335	2.80e+008	3052.4	6837482	0.1295	6.183	3526654	1.13e+009	197.1
325	2.76e+008	3103.3	6840585	0.1355	6.318	3513171	1.13e+009	197.5
315	2.73e+008	3165.3	6843750	0.1424	6.461	3499002	1.14e+009	198.0
305	2.70e+008	3224.1	6846974	0.1496	6.610	3484117	1.14e+009	198.5
295	2.66e+008	3287.2	6850261	0.1574	6.768	3468452	1.14e+009	199.0
285	2.63e+008	3354.9	6853616	0.1660	6.934	3451930	1.15e+009	199.5
275	2.59e+008	3430.8	6857047	0.1756	7.110	3434452	1.15e+009	200.0
265	2.56e+008	3510.5	6860558	0.1862	7.296	3415927	1.16e+009	200.6
255	2.52e+008	3591.9	6864150	0.1975	7.493	3396271	1.16e+009	201.1
245	2.48e+008	3684.3	6867834	0.2104	7.704	3375332	1.17e+009	201.7
235	2.44e+008	3782.7	6871617	0.2247	7.928	3352973	1.17e+009	202.2
225	2.40e+008	3891.5	6875508	0.2408	8.169	3332670	1.17e+009	202.8
215	2.36e+008	4008.3	6879516	0.2588	8.428	3312514	1.18e+009	203.4
205	2.32e+008	4136.1	6883653	0.2792	8.707	3290768	1.18e+009	204.0
200	2.30e+008	2134.1	6885787	0.1474	8.855	3279287	1.19e+009	204.3
195	2.28e+008	2171.4	6887958	0.1536	9.008	3267327	1.19e+009	204.7
190	2.26e+008	2205.8	6890164	0.1598	9.168	3254880	1.19e+009	205.0
185	2.24e+008	2253.6	6892417	0.1674	9.335	3241845	1.19e+009	205.3
180	2.22e+008	2297.7	6894715	0.1750	9.510	3228214	1.20e+009	205.7
175	2.19e+008	2341.9	6897057	0.1831	9.694	3213955	1.20e+009	206.0
170	2.17e+008	2395.7	6899453	0.1924	9.886	3198972	1.20e+009	206.4
165	2.15e+008	2444.7	6901897	0.2018	10.088	3184953	1.21e+009	206.7
160	2.12e+008	2505.8	6904403	0.2127	10.300	3172496	1.21e+009	207.1
155	2.10e+008	2568.8	6906972	0.2245	10.525	3159350	1.21e+009	207.5
150	2.08e+008	2633.2	6909605	0.2371	10.762	3145466	1.22e+009	207.9
145	2.05e+008	2704.5	6912310	0.2512	11.013	3130759	1.22e+009	208.3
140	2.03e+008	2784.6	6915094	0.2670	11.280	3115127	1.22e+009	208.7
135	2.00e+008	2863.4	6917958	0.2837	11.564	3098516	1.23e+009	209.1
130	1.98e+008	2953.5	6920911	0.3027	11.867	3080790	1.23e+009	209.5
125	1.95e+008	3048.4	6923959	0.3236	12.190	3061840	1.23e+009	210.0
120	1.92e+008	3155.5	6927115	0.3475	12.538	3041496	1.24e+009	210.4
115	1.89e+008	3274.0	6930389	0.3744	12.912	3019574	1.24e+009	210.9
110	1.86e+008	3393.6	6933783	0.4036	13.316	2995940	1.24e+009	211.4
105	1.83e+008	3540.2	6937323	0.4387	13.754	2970256	1.25e+009	211.9
100	1.80e+008	3693.7	6941016	0.4776	14.232	2942289	1.25e+009	212.5
95	1.77e+008	3862.4	6944879	0.5222	14.754	2911712	1.26e+009	213.0
90	1.74e+008	4052.4	6948931	0.5741	15.328	2883403	1.26e+009	213.6
85	1.71e+008	4276.6	6953208	0.6362	15.965	2856427	1.27e+009	214.2
80	1.67e+008	4533.1	6957741	0.7100	16.675	2826324	1.27e+009	214.9
75	1.64e+008	4835.3	6962576	0.7995	17.474	2792425	1.28e+009	215.6
70	1.60e+008	5172.7	6967749	0.9058	18.380	2754019	1.28e+009	216.3
65	1.56e+008	5585.0	6973334	1.0393	19.419	2709951	1.29e+009	217.1
60	1.52e+008	6064.3	6979398	1.2041	20.623	2664651	1.30e+009	218.0
55	1.48e+008	6659.2	6986058	1.4171	22.040	2617647	1.31e+009	218.9
50	1.43e+008	7420.5	6993478	1.7011	23.741	2561222	1.31e+009	220.0

torus water level = 15.00 ft

prpv [psig]	trpv [°F]	hf_rpv [Btu/lbm]	hg_rpv [Btu/lbm]	vf_rpv [ft3/lbm]	vg_rpv [ft3/lbm]	wsrv [lbm/min.]	mf_rpv [lbm]	mg_rpv [lbm]
1145	562.88	566.16	1186.6	0.02217	0.3771	78190	532000	15300
1135	561.79	564.74	1187.0	0.02213	0.3808	77515	530124	15309
1125	560.70	563.31	1187.4	0.02210	0.3847	76841	528251	15314
1115	559.60	561.87	1187.8	0.02206	0.3886	76167	526376	15316
1105	558.49	560.42	1188.3	0.02202	0.3926	75493	524498	15315
1095	557.37	558.97	1188.7	0.02199	0.3966	74819	522619	15312
1085	556.24	557.51	1189.1	0.02195	0.4007	74144	520742	15305
1075	555.11	556.04	1189.5	0.02192	0.4049	73470	518860	15295

1065	553.97	554.57	1189.9	0.02188	0.4091	72796	516980	15283
1055	552.82	553.09	1190.3	0.02184	0.4135	72122	515102	15268
1045	551.66	551.60	1190.7	0.02181	0.4179	71447	513216	15250
1035	550.50	550.11	1191.1	0.02177	0.4224	70773	511335	15229
1025	549.32	548.60	1191.4	0.02173	0.4269	70099	509442	15205
1015	548.14	547.09	1191.8	0.02170	0.4316	69425	507556	15179
1005	546.95	545.57	1192.2	0.02166	0.4363	68751	505664	15150
995	545.75	544.04	1192.6	0.02163	0.4412	68076	503775	15118
985	544.54	542.51	1192.9	0.02159	0.4461	67402	501877	15083
975	543.32	540.96	1193.3	0.02155	0.4511	66728	499979	15046
965	542.10	539.40	1193.7	0.02152	0.4562	66054	498076	15007
955	540.86	537.85	1194.0	0.02148	0.4615	65379	496171	14965
945	539.61	536.27	1194.4	0.02145	0.4668	64705	494264	14920
935	538.35	534.70	1194.7	0.02141	0.4722	64031	492352	14873
925	537.09	533.11	1195.1	0.02137	0.4778	63357	490444	14823
915	535.81	531.51	1195.4	0.02134	0.4834	62683	488524	14771
905	534.52	529.90	1195.7	0.02130	0.4892	62008	486600	14716
895	533.22	528.28	1196.1	0.02127	0.4951	61334	484671	14659
885	531.91	526.65	1196.4	0.02123	0.5011	60660	482740	14600
875	530.59	525.02	1196.7	0.02119	0.5072	59986	480810	14538
865	529.26	523.37	1197.0	0.02116	0.5135	59311	478871	14474
855	527.92	521.71	1197.3	0.02112	0.5199	58637	476927	14407
845	526.56	520.04	1197.7	0.02109	0.5265	57963	474977	14338
835	525.20	518.35	1198.0	0.02105	0.5332	57289	473019	14267
825	523.82	516.66	1198.3	0.02101	0.5401	56614	471061	14194
815	522.42	514.95	1198.5	0.02098	0.5471	55940	469088	14118
805	521.02	513.24	1198.8	0.02094	0.5543	55266	467123	14040
795	519.60	511.50	1199.1	0.02090	0.5617	54592	465138	13960
785	518.16	509.76	1199.4	0.02087	0.5692	53918	463155	13878
775	516.72	508.00	1199.7	0.02083	0.5769	53243	461159	13793
765	515.26	506.24	1199.9	0.02080	0.5848	52569	459165	13706
755	513.78	504.45	1200.2	0.02076	0.5929	51895	457153	13618
745	512.30	502.65	1200.4	0.02072	0.6012	51221	455136	13527
735	510.80	500.84	1200.7	0.02069	0.6097	50546	453117	13433
725	509.28	499.01	1200.9	0.02065	0.6185	49872	451088	13338
715	507.74	497.17	1201.2	0.02061	0.6275	49198	449042	13241
705	506.19	495.31	1201.4	0.02058	0.6367	48524	446997	13141
695	504.62	493.43	1201.6	0.02054	0.6461	47850	444936	13039
685	503.03	491.54	1201.8	0.02050	0.6559	47175	442865	12936
675	501.43	489.64	1202.1	0.02047	0.6658	46501	440790	12830
665	499.81	487.71	1202.3	0.02043	0.6761	45827	438699	12722
655	498.17	485.77	1202.5	0.02039	0.6867	45153	436600	12613
645	496.52	483.81	1202.7	0.02036	0.6976	44478	434488	12501
635	494.84	481.83	1202.8	0.02032	0.7088	43804	432363	12387
625	493.14	479.83	1203.0	0.02028	0.7203	43130	430225	12271
615	491.43	477.82	1203.2	0.02024	0.7322	42456	428084	12153
605	489.69	475.78	1203.4	0.02021	0.7445	41782	425920	12033
595	487.93	473.72	1203.5	0.02017	0.7571	41107	423745	11911
585	486.15	471.64	1203.7	0.02013	0.7701	40433	421556	11788
575	484.35	469.53	1203.8	0.02009	0.7836	39759	419350	11662
565	482.52	467.40	1203.9	0.02005	0.7975	39085	417131	11534
555	480.66	465.25	1204.1	0.02002	0.8119	38410	414894	11404
545	478.78	463.07	1204.2	0.01998	0.8268	37736	412636	11273
535	476.88	460.87	1204.3	0.01994	0.8422	37062	410365	11139
525	474.95	458.64	1204.4	0.01990	0.8582	36388	408072	11004
515	472.99	456.39	1204.5	0.01986	0.8747	35714	405760	10866
505	471.01	454.11	1204.5	0.01982	0.8919	35039	403432	10727
495	468.99	451.80	1204.6	0.01978	0.9097	34365	401076	10585
485	466.95	449.45	1204.7	0.01974	0.9282	33691	398698	10442
475	464.87	447.08	1204.7	0.01970	0.9474	33017	396298	10297
465	462.76	444.68	1204.8	0.01966	0.9674	32342	393876	10150
455	460.61	442.24	1204.8	0.01962	0.9882	31668	391427	10001
445	458.43	439.76	1204.8	0.01958	1.0099	30994	388948	9850
435	456.21	437.25	1204.8	0.01954	1.0325	30320	386443	9697
425	453.96	434.69	1204.8	0.01950	1.0561	29646	383910	9542
415	451.67	432.11	1204.8	0.01946	1.0807	28971	381347	9385
405	449.33	429.48	1204.7	0.01942	1.1065	28297	378748	9226
395	446.95	426.80	1204.7	0.01938	1.1335	27623	376113	9065

385	444.53	424.09	1204.6	0.01934	1.1618	26949	373449	8902
375	442.06	421.32	1204.5	0.01929	1.1915	26274	370742	8738
365	439.53	418.51	1204.4	0.01925	1.2227	25600	367998	8571
355	436.96	415.64	1204.3	0.01921	1.2556	24926	365214	8402
345	434.33	412.72	1204.1	0.01916	1.2902	24252	362377	8231
335	431.65	409.74	1204.0	0.01912	1.3267	23578	359498	8058
325	428.91	406.71	1203.8	0.01908	1.3652	22903	356570	7883
315	426.10	403.60	1203.6	0.01903	1.4061	22229	353581	7706
305	423.22	400.43	1203.4	0.01899	1.4493	21555	350536	7527
295	420.27	397.20	1203.1	0.01894	1.4953	20881	347431	7346
285	417.26	393.89	1202.9	0.01889	1.5443	20206	344259	7162
275	414.16	390.50	1202.6	0.01885	1.5964	19532	341014	6977
265	410.97	387.01	1202.3	0.01880	1.6522	18858	337691	6789
255	407.70	383.45	1201.9	0.01875	1.7120	18184	334290	6598
245	404.33	379.79	1201.5	0.01870	1.7762	17510	330798	6406
235	400.87	376.03	1201.1	0.01865	1.8453	16835	327210	6211
225	397.28	372.15	1200.6	0.01860	1.9200	16161	323515	6014
215	393.59	368.16	1200.1	0.01855	2.0010	15487	319707	5814
205	389.76	364.04	1199.6	0.01849	2.0891	14813	315773	5612
200	387.79	361.93	1199.3	0.01847	2.1360	14476	313741	5510
195	385.79	359.78	1199.0	0.01844	2.1852	14138	311672	5408
190	383.76	357.60	1198.6	0.01841	2.2367	13801	309569	5304
185	381.67	355.38	1198.3	0.01839	2.2906	13464	307420	5200
180	379.56	353.10	1198.0	0.01836	2.3472	13127	305227	5096
175	377.40	350.80	1197.6	0.01833	2.4067	12790	302990	4990
170	375.19	348.44	1197.2	0.01830	2.4693	12453	300701	4884
165	372.94	346.04	1196.8	0.01827	2.5353	12116	298363	4778
160	370.64	343.59	1196.4	0.01824	2.6048	11779	295964	4670
155	368.28	341.09	1196.0	0.01821	2.6784	11442	293504	4562
150	365.87	338.52	1195.5	0.01818	2.7562	11104	290979	4453
145	363.40	335.91	1195.1	0.01815	2.8387	10767	288385	4343
140	360.87	333.23	1194.6	0.01812	2.9264	10430	285711	4233
135	358.27	330.48	1194.0	0.01809	3.0197	10093	282959	4122
130	355.61	327.67	1193.5	0.01806	3.1192	9756	280117	4009
125	352.87	324.78	1192.9	0.01802	3.2256	9419	277182	3896
120	350.06	321.82	1192.3	0.01799	3.3396	9082	274140	3783
115	347.15	318.76	1191.7	0.01796	3.4620	8745	270981	3668
110	344.17	315.63	1191.0	0.01792	3.5940	8408	267703	3552
105	341.08	312.39	1190.3	0.01789	3.7365	8070	264279	3436
100	337.89	309.05	1189.6	0.01785	3.8910	7733	260703	3318
95	334.59	305.59	1188.8	0.01781	4.0590	7396	256959	3200
90	331.16	302.02	1188.0	0.01778	4.2425	7059	253026	3080
85	327.60	298.31	1187.1	0.01774	4.4437	6722	248870	2959
80	323.90	294.46	1186.2	0.01770	4.6654	6385	244459	2837
75	320.04	290.44	1185.2	0.01766	4.9109	6048	239747	2715
70	316.01	286.26	1184.1	0.01761	5.1843	5711	234698	2591
65	311.78	281.88	1183.0	0.01757	5.4907	5374	229238	2465
60	307.33	277.28	1181.8	0.01753	5.8367	5036	223300	2339
55	302.64	272.44	1180.5	0.01748	6.2304	4699	216769	2211
50	297.67	267.31	1179.1	0.01743	6.6827	4362	209478	2082

prpv [psig]	erpv [Btu]	dmg_srv [lbm]	mf_sp [lbm]	dt [min.]	t [min.]	qdh [Btu/min.]	esp [Btu]	tsp [°F]
1145	4.79e+008	0.0	7248848	0.0000	2.000	4381320	1.02e+009	173.3
1135	4.77e+008	1867.9	7250716	0.0241	2.024	4374717	1.03e+009	173.6
1125	4.75e+008	1867.4	7252583	0.0243	2.048	4368058	1.03e+009	173.8
1115	4.73e+008	1872.9	7254456	0.0246	2.073	4361320	1.03e+009	174.1
1105	4.71e+008	1878.3	7256334	0.0249	2.098	4354502	1.03e+009	174.4
1095	4.69e+008	1882.5	7258217	0.0252	2.123	4347608	1.04e+009	174.6
1085	4.66e+008	1884.3	7260101	0.0254	2.148	4340644	1.04e+009	174.9
1075	4.64e+008	1891.2	7261992	0.0257	2.174	4333591	1.04e+009	175.2
1065	4.62e+008	1892.1	7263884	0.0260	2.200	4326469	1.04e+009	175.5
1055	4.60e+008	1893.8	7265778	0.0263	2.226	4319274	1.04e+009	175.7
1045	4.58e+008	1904.5	7267683	0.0267	2.253	4311970	1.05e+009	176.0
1035	4.56e+008	1902.0	7269585	0.0269	2.280	4304606	1.05e+009	176.3
1025	4.54e+008	1915.9	7271500	0.0273	2.307	4297117	1.05e+009	176.5
1015	4.51e+008	1913.2	7273414	0.0276	2.335	4289566	1.05e+009	176.8
1005	4.49e+008	1920.4	7275334	0.0279	2.363	4281912	1.06e+009	177.1

995	4.47e+008	1921.6	7277256	0.0282	2.391	4274178	1.06e+009	177.4
985	4.45e+008	1931.8	7279187	0.0287	2.420	4266324	1.06e+009	177.7
975	4.43e+008	1935.5	7281123	0.0290	2.449	4258376	1.06e+009	177.9
965	4.40e+008	1942.2	7283065	0.0294	2.478	4250319	1.06e+009	178.2
955	4.38e+008	1946.6	7285012	0.0298	2.508	4242161	1.07e+009	178.5
945	4.36e+008	1951.7	7286963	0.0302	2.538	4233896	1.07e+009	178.8
935	4.34e+008	1959.8	7288923	0.0306	2.569	4225509	1.07e+009	179.1
925	4.32e+008	1957.6	7290881	0.0309	2.600	4217043	1.07e+009	179.3
915	4.29e+008	1972.5	7292853	0.0315	2.631	4208420	1.08e+009	179.6
905	4.27e+008	1978.8	7294832	0.0319	2.663	4199676	1.08e+009	179.9
895	4.25e+008	1985.1	7296817	0.0324	2.695	4190807	1.08e+009	180.2
885	4.23e+008	1991.3	7298808	0.0328	2.728	4181812	1.08e+009	180.5
875	4.20e+008	1991.7	7300800	0.0332	2.761	4172714	1.09e+009	180.8
865	4.18e+008	2003.5	7302804	0.0338	2.795	4163458	1.09e+009	181.1
855	4.16e+008	2010.5	7304814	0.0343	2.829	4154063	1.09e+009	181.3
845	4.14e+008	2018.1	7306832	0.0348	2.864	4144523	1.09e+009	181.6
835	4.11e+008	2029.6	7308862	0.0354	2.900	4134815	1.10e+009	181.9
825	4.09e+008	2030.8	7310893	0.0359	2.935	4124986	1.10e+009	182.2
815	4.07e+008	2048.4	7312941	0.0366	2.972	4114952	1.10e+009	182.5
805	4.04e+008	2044.1	7314985	0.0370	3.009	4104817	1.10e+009	182.8
795	4.02e+008	2064.3	7317050	0.0378	3.047	4094456	1.11e+009	183.1
785	4.00e+008	2065.8	7319115	0.0383	3.085	4083957	1.11e+009	183.4
775	3.98e+008	2080.1	7321195	0.0391	3.124	4073252	1.11e+009	183.7
765	3.95e+008	2081.6	7323277	0.0396	3.164	4062402	1.11e+009	184.0
755	3.93e+008	2100.0	7325377	0.0405	3.204	4051314	1.12e+009	184.3
745	3.90e+008	2108.0	7327485	0.0412	3.246	4040037	1.12e+009	184.6
735	3.88e+008	2112.9	7329598	0.0418	3.287	4028582	1.12e+009	184.9
725	3.86e+008	2124.3	7331722	0.0426	3.330	4016911	1.12e+009	185.2
715	3.83e+008	2143.1	7333865	0.0436	3.373	4004975	1.13e+009	185.5
705	3.81e+008	2144.9	7336010	0.0442	3.418	3992863	1.13e+009	185.8
695	3.78e+008	2162.5	7338173	0.0452	3.463	3980479	1.13e+009	186.1
685	3.76e+008	2174.3	7340347	0.0461	3.509	3967850	1.13e+009	186.4
675	3.74e+008	2180.6	7342528	0.0469	3.556	3955000	1.14e+009	186.7
665	3.71e+008	2198.6	7344726	0.0480	3.604	3941854	1.14e+009	187.0
655	3.69e+008	2209.2	7346935	0.0489	3.653	3928448	1.14e+009	187.4
645	3.66e+008	2223.2	7349159	0.0500	3.703	3914751	1.14e+009	187.7
635	3.64e+008	2239.1	7351398	0.0511	3.754	3900745	1.15e+009	188.0
625	3.61e+008	2253.9	7353652	0.0523	3.806	3886426	1.15e+009	188.3
615	3.59e+008	2259.6	7355911	0.0532	3.859	3871842	1.15e+009	188.6
605	3.56e+008	2282.9	7358194	0.0546	3.914	3856870	1.15e+009	189.0
595	3.54e+008	2297.2	7360491	0.0559	3.970	3841557	1.16e+009	189.3
585	3.51e+008	2312.6	7362804	0.0572	4.027	3829395	1.16e+009	189.6
575	3.48e+008	2332.1	7365136	0.0587	4.086	3820936	1.16e+009	190.0
565	3.46e+008	2346.4	7367482	0.0600	4.146	3812278	1.17e+009	190.3
555	3.43e+008	2366.6	7369849	0.0616	4.207	3803392	1.17e+009	190.6
545	3.41e+008	2390.1	7372239	0.0633	4.271	3794258	1.17e+009	191.0
535	3.38e+008	2404.2	7374643	0.0649	4.336	3784903	1.17e+009	191.3
525	3.35e+008	2428.9	7377072	0.0668	4.402	3775276	1.18e+009	191.6
515	3.33e+008	2448.9	7379521	0.0686	4.471	3765387	1.18e+009	192.0
505	3.30e+008	2468.4	7381990	0.0704	4.541	3755227	1.18e+009	192.3
495	3.27e+008	2497.3	7384487	0.0727	4.614	3744747	1.19e+009	192.7
485	3.24e+008	2520.7	7387008	0.0748	4.689	3733957	1.19e+009	193.1
475	3.22e+008	2545.1	7389553	0.0771	4.766	3722840	1.19e+009	193.4
465	3.19e+008	2569.0	7392122	0.0794	4.845	3711384	1.20e+009	193.8
455	3.16e+008	2597.9	7394720	0.0820	4.927	3699553	1.20e+009	194.1
445	3.13e+008	2630.4	7397350	0.0849	5.012	3687314	1.20e+009	194.5
435	3.10e+008	2658.0	7400008	0.0877	5.100	3674671	1.21e+009	194.9
425	3.07e+008	2688.3	7402696	0.0907	5.191	3661593	1.21e+009	195.3
415	3.04e+008	2719.4	7405416	0.0939	5.284	3648057	1.21e+009	195.6
405	3.01e+008	2757.7	7408173	0.0975	5.382	3634002	1.22e+009	196.0
395	2.98e+008	2796.4	7410970	0.1012	5.483	3619402	1.22e+009	196.4
385	2.95e+008	2826.6	7413796	0.1049	5.588	3604275	1.22e+009	196.8
375	2.92e+008	2872.0	7416668	0.1093	5.697	3588511	1.23e+009	197.2
365	2.89e+008	2910.5	7419579	0.1137	5.811	3572115	1.23e+009	197.6
355	2.86e+008	2953.3	7422532	0.1185	5.929	3555028	1.23e+009	198.0
345	2.83e+008	3007.0	7425539	0.1240	6.053	3539537	1.24e+009	198.5
335	2.80e+008	3052.4	7428592	0.1295	6.183	3526654	1.24e+009	198.9
325	2.76e+008	3103.3	7431695	0.1355	6.318	3513171	1.24e+009	199.3

315	2.73e+008	3165.3	7434860	0.1424	6.461	3499002	1.25e+009	199.8
305	2.70e+008	3224.1	7438084	0.1496	6.610	3484117	1.25e+009	200.2
295	2.66e+008	3287.2	7441372	0.1574	6.768	3468452	1.26e+009	200.7
285	2.63e+008	3354.9	7444726	0.1660	6.934	3451930	1.26e+009	201.1
275	2.59e+008	3430.8	7448157	0.1756	7.110	3434452	1.26e+009	201.6
265	2.56e+008	3510.5	7451668	0.1862	7.296	3415927	1.27e+009	202.1
255	2.52e+008	3591.9	7455260	0.1975	7.493	3396271	1.27e+009	202.6
245	2.48e+008	3684.3	7458944	0.2104	7.704	3375332	1.28e+009	203.1
235	2.44e+008	3782.7	7462727	0.2247	7.928	3352973	1.28e+009	203.6
225	2.40e+008	3891.5	7466618	0.2408	8.169	3332670	1.29e+009	204.2
215	2.36e+008	4008.3	7470627	0.2588	8.428	3312514	1.29e+009	204.7
205	2.32e+008	4136.1	7474763	0.2792	8.707	3290768	1.30e+009	205.3
200	2.30e+008	2134.1	7476897	0.1474	8.855	3279287	1.30e+009	205.6
195	2.28e+008	2171.4	7479068	0.1536	9.008	3267327	1.30e+009	205.9
190	2.26e+008	2205.8	7481274	0.1598	9.168	3254880	1.30e+009	206.2
185	2.24e+008	2253.6	7483528	0.1674	9.335	3241845	1.31e+009	206.5
180	2.22e+008	2297.7	7485825	0.1750	9.510	3228214	1.31e+009	206.8
175	2.19e+008	2341.9	7488167	0.1831	9.694	3213955	1.31e+009	207.1
170	2.17e+008	2395.7	7490563	0.1924	9.886	3198972	1.31e+009	207.4
165	2.15e+008	2444.7	7493008	0.2018	10.088	3184953	1.32e+009	207.8
160	2.12e+008	2505.8	7495513	0.2127	10.300	3172496	1.32e+009	208.1
155	2.10e+008	2568.8	7498082	0.2245	10.525	3159350	1.32e+009	208.5
150	2.08e+008	2633.2	7500715	0.2371	10.762	3145466	1.33e+009	208.8
145	2.05e+008	2704.5	7503420	0.2512	11.013	3130759	1.33e+009	209.2
140	2.03e+008	2784.6	7506204	0.2670	11.280	3115127	1.33e+009	209.6
135	2.00e+008	2863.4	7509068	0.2837	11.564	3098516	1.34e+009	210.0
130	1.98e+008	2953.5	7512021	0.3027	11.867	3080790	1.34e+009	210.4
125	1.95e+008	3048.4	7515070	0.3236	12.190	3061840	1.34e+009	210.8
120	1.92e+008	3155.5	7518225	0.3475	12.538	3041496	1.35e+009	211.2
115	1.89e+008	3274.0	7521499	0.3744	12.912	3019574	1.35e+009	211.6
110	1.86e+008	3393.6	7524893	0.4036	13.316	2995940	1.36e+009	212.1
105	1.83e+008	3540.2	7528433	0.4387	13.754	2970256	1.36e+009	212.6
100	1.80e+008	3693.7	7532127	0.4776	14.232	2942289	1.36e+009	213.1
95	1.77e+008	3862.4	7535989	0.5222	14.754	2911712	1.37e+009	213.6
90	1.74e+008	4052.4	7540041	0.5741	15.328	2883403	1.37e+009	214.1
85	1.71e+008	4276.6	7544318	0.6362	15.965	2856427	1.38e+009	214.7
80	1.67e+008	4533.1	7548851	0.7100	16.675	2826324	1.38e+009	215.3
75	1.64e+008	4835.3	7553686	0.7995	17.474	2792425	1.39e+009	215.9
70	1.60e+008	5172.7	7558859	0.9058	18.380	2754019	1.40e+009	216.6
65	1.56e+008	5585.0	7564444	1.0393	19.419	2709951	1.40e+009	217.4
60	1.52e+008	6064.3	7570509	1.2041	20.623	2664651	1.41e+009	218.2
55	1.48e+008	6659.2	7577168	1.4171	22.040	2617647	1.42e+009	219.0
50	1.43e+008	7420.5	7584588	1.7011	23.741	2561222	1.43e+009	220.0

torus water level = 16.00 ft

prpv [psig]	trpv [°F]	hf_rpv [Btu/lbm]	hg_rpv [Btu/lbm]	vf_rpv [ft3/lbm]	vg_rpv [ft3/lbm]	wsrv [lbm/min.]	mf_rpv [lbm]	mg_rpv [lbm]
1145	562.88	566.16	1186.6	0.02217	0.3771	78190	532000	15300
1135	561.79	564.74	1187.0	0.02213	0.3808	77515	530124	15309
1125	560.70	563.31	1187.4	0.02210	0.3847	76841	528251	15314
1115	559.60	561.87	1187.8	0.02206	0.3886	76167	526376	15316
1105	558.49	560.42	1188.3	0.02202	0.3926	75493	524498	15315
1095	557.37	558.97	1188.7	0.02199	0.3966	74819	522619	15312
1085	556.24	557.51	1189.1	0.02195	0.4007	74144	520742	15305
1075	555.11	556.04	1189.5	0.02192	0.4049	73470	518860	15295
1065	553.97	554.57	1189.9	0.02188	0.4091	72796	516980	15283
1055	552.82	553.09	1190.3	0.02184	0.4135	72122	515102	15268
1045	551.66	551.60	1190.7	0.02181	0.4179	71447	513216	15250
1035	550.50	550.11	1191.1	0.02177	0.4224	70773	511335	15229
1025	549.32	548.60	1191.4	0.02173	0.4269	70099	509442	15205
1015	548.14	547.09	1191.8	0.02170	0.4316	69425	507556	15179
1005	546.95	545.57	1192.2	0.02166	0.4363	68751	505664	15150
995	545.75	544.04	1192.6	0.02163	0.4412	68076	503775	15118
985	544.54	542.51	1192.9	0.02159	0.4461	67402	501877	15083
975	543.32	540.96	1193.3	0.02155	0.4511	66728	499979	15046
965	542.10	539.40	1193.7	0.02152	0.4562	66054	498076	15007
955	540.86	537.85	1194.0	0.02148	0.4615	65379	496171	14965

945	539.61	536.27	1194.4	0.02145	0.4668	64705	494264	14920
935	538.35	534.70	1194.7	0.02141	0.4722	64031	492352	14873
925	537.09	533.11	1195.1	0.02137	0.4778	63357	490444	14823
915	535.81	531.51	1195.4	0.02134	0.4834	62683	488524	14771
905	534.52	529.90	1195.7	0.02130	0.4892	62008	486600	14716
895	533.22	528.28	1196.1	0.02127	0.4951	61334	484671	14659
885	531.91	526.65	1196.4	0.02123	0.5011	60660	482740	14600
875	530.59	525.02	1196.7	0.02119	0.5072	59986	480810	14538
865	529.26	523.37	1197.0	0.02116	0.5135	59311	478871	14474
855	527.92	521.71	1197.3	0.02112	0.5199	58637	476927	14407
845	526.56	520.04	1197.7	0.02109	0.5265	57963	474977	14338
835	525.20	518.35	1198.0	0.02105	0.5332	57289	473019	14267
825	523.82	516.66	1198.3	0.02101	0.5401	56614	471061	14194
815	522.42	514.95	1198.5	0.02098	0.5471	55940	469088	14118
805	521.02	513.24	1198.8	0.02094	0.5543	55266	467123	14040
795	519.60	511.50	1199.1	0.02090	0.5617	54592	465138	13960
785	518.16	509.76	1199.4	0.02087	0.5692	53918	463155	13878
775	516.72	508.00	1199.7	0.02083	0.5769	53243	461159	13793
765	515.26	506.24	1199.9	0.02080	0.5848	52569	459165	13706
755	513.78	504.45	1200.2	0.02076	0.5929	51895	457153	13618
745	512.30	502.65	1200.4	0.02072	0.6012	51221	455136	13527
735	510.80	500.84	1200.7	0.02069	0.6097	50546	453117	13433
725	509.28	499.01	1200.9	0.02065	0.6185	49872	451088	13338
715	507.74	497.17	1201.2	0.02061	0.6275	49198	449042	13241
705	506.19	495.31	1201.4	0.02058	0.6367	48524	446997	13141
695	504.62	493.43	1201.6	0.02054	0.6461	47850	444936	13039
685	503.03	491.54	1201.8	0.02050	0.6559	47175	442865	12936
675	501.43	489.64	1202.1	0.02047	0.6658	46501	440790	12830
665	499.81	487.71	1202.3	0.02043	0.6761	45827	438699	12722
655	498.17	485.77	1202.5	0.02039	0.6867	45153	436600	12613
645	496.52	483.81	1202.7	0.02036	0.6976	44478	434488	12501
635	494.84	481.83	1202.8	0.02032	0.7088	43804	432363	12387
625	493.14	479.83	1203.0	0.02028	0.7203	43130	430225	12271
615	491.43	477.82	1203.2	0.02024	0.7322	42456	428084	12153
605	489.69	475.78	1203.4	0.02021	0.7445	41782	425920	12033
595	487.93	473.72	1203.5	0.02017	0.7571	41107	423745	11911
585	486.15	471.64	1203.7	0.02013	0.7701	40433	421556	11788
575	484.35	469.53	1203.8	0.02009	0.7836	39759	419350	11662
565	482.52	467.40	1203.9	0.02005	0.7975	39085	417131	11534
555	480.66	465.25	1204.1	0.02002	0.8119	38410	414894	11404
545	478.78	463.07	1204.2	0.01998	0.8268	37736	412636	11273
535	476.88	460.87	1204.3	0.01994	0.8422	37062	410365	11139
525	474.95	458.64	1204.4	0.01990	0.8582	36388	408072	11004
515	472.99	456.39	1204.5	0.01986	0.8747	35714	405760	10866
505	471.01	454.11	1204.5	0.01982	0.8919	35039	403432	10727
495	468.99	451.80	1204.6	0.01978	0.9097	34365	401076	10585
485	466.95	449.45	1204.7	0.01974	0.9282	33691	398698	10442
475	464.87	447.08	1204.7	0.01970	0.9474	33017	396298	10297
465	462.76	444.68	1204.8	0.01966	0.9674	32342	393876	10150
455	460.61	442.24	1204.8	0.01962	0.9882	31668	391427	10001
445	458.43	439.76	1204.8	0.01958	1.0099	30994	388948	9850
435	456.21	437.25	1204.8	0.01954	1.0325	30320	386443	9697
425	453.96	434.69	1204.8	0.01950	1.0561	29646	383910	9542
415	451.67	432.11	1204.8	0.01946	1.0807	28971	381347	9385
405	449.33	429.48	1204.7	0.01942	1.1065	28297	378748	9226
395	446.95	426.80	1204.7	0.01938	1.1335	27623	376113	9065
385	444.53	424.09	1204.6	0.01934	1.1618	26949	373449	8902
375	442.06	421.32	1204.5	0.01929	1.1915	26274	370742	8738
365	439.53	418.51	1204.4	0.01925	1.2227	25600	367998	8571
355	436.96	415.64	1204.3	0.01921	1.2556	24926	365214	8402
345	434.33	412.72	1204.1	0.01916	1.2902	24252	362377	8231
335	431.65	409.74	1204.0	0.01912	1.3267	23578	359498	8058
325	428.91	406.71	1203.8	0.01908	1.3652	22903	356570	7883
315	426.10	403.60	1203.6	0.01903	1.4061	22229	353581	7706
305	423.22	400.43	1203.4	0.01899	1.4493	21555	350536	7527
295	420.27	397.20	1203.1	0.01894	1.4953	20881	347431	7346
285	417.26	393.89	1202.9	0.01889	1.5443	20206	344259	7162
275	414.16	390.50	1202.6	0.01885	1.5964	19532	341014	6977

265	410.97	387.01	1202.3	0.01880	1.6522	18858	337691	6789
255	407.70	383.45	1201.9	0.01875	1.7120	18184	334290	6598
245	404.33	379.79	1201.5	0.01870	1.7762	17510	330798	6406
235	400.87	376.03	1201.1	0.01865	1.8453	16835	327210	6211
225	397.28	372.15	1200.6	0.01860	1.9200	16161	323515	6014
215	393.59	368.16	1200.1	0.01855	2.0010	15487	319707	5814
205	389.76	364.04	1199.6	0.01849	2.0891	14813	315773	5612
200	387.79	361.93	1199.3	0.01847	2.1360	14476	313741	5510
195	385.79	359.78	1199.0	0.01844	2.1852	14138	311672	5408
190	383.76	357.60	1198.6	0.01841	2.2367	13801	309569	5304
185	381.67	355.38	1198.3	0.01839	2.2906	13464	307420	5200
180	379.56	353.10	1198.0	0.01836	2.3472	13127	305227	5096
175	377.40	350.80	1197.6	0.01833	2.4067	12790	302990	4990
170	375.19	348.44	1197.2	0.01830	2.4693	12453	300701	4884
165	372.94	346.04	1196.8	0.01827	2.5353	12116	298363	4778
160	370.64	343.59	1196.4	0.01824	2.6048	11779	295964	4670
155	368.28	341.09	1196.0	0.01821	2.6784	11442	293504	4562
150	365.87	338.52	1195.5	0.01818	2.7562	11104	290979	4453
145	363.40	335.91	1195.1	0.01815	2.8387	10767	288385	4343
140	360.87	333.23	1194.6	0.01812	2.9264	10430	285711	4233
135	358.27	330.48	1194.0	0.01809	3.0197	10093	282959	4122
130	355.61	327.67	1193.5	0.01806	3.1192	9756	280117	4009
125	352.87	324.78	1192.9	0.01802	3.2256	9419	277182	3896
120	350.06	321.82	1192.3	0.01799	3.3396	9082	274140	3783
115	347.15	318.76	1191.7	0.01796	3.4620	8745	270981	3668
110	344.17	315.63	1191.0	0.01792	3.5940	8408	267703	3552
105	341.08	312.39	1190.3	0.01789	3.7365	8070	264279	3436
100	337.89	309.05	1189.6	0.01785	3.8910	7733	260703	3318
95	334.59	305.59	1188.8	0.01781	4.0590	7396	256959	3200
90	331.16	302.02	1188.0	0.01778	4.2425	7059	253026	3080
85	327.60	298.31	1187.1	0.01774	4.4437	6722	248870	2959
80	323.90	294.46	1186.2	0.01770	4.6654	6385	244459	2837
75	320.04	290.44	1185.2	0.01766	4.9109	6048	239747	2715
70	316.01	286.26	1184.1	0.01761	5.1843	5711	234698	2591
65	311.78	281.88	1183.0	0.01757	5.4907	5374	229238	2465
60	307.33	277.28	1181.8	0.01753	5.8367	5036	223300	2339
55	302.64	272.44	1180.5	0.01748	6.2304	4699	216769	2211
50	297.67	267.31	1179.1	0.01743	6.6827	4362	209478	2082

prpv [psig]	erpv [Btu]	dmg_srv [lbm]	mf_sp [lbm]	dt [min.]	t [min.]	qdh [Btu/min.]	esp [Btu]	tsp [°F]
1145	4.79e+008	0.0	7821584	0.0000	2.000	4381320	1.13e+009	176.7
1135	4.77e+008	1867.9	7823452	0.0241	2.024	4374717	1.13e+009	177.0
1125	4.75e+008	1867.4	7825320	0.0243	2.048	4368058	1.14e+009	177.2
1115	4.73e+008	1872.9	7827193	0.0246	2.073	4361320	1.14e+009	177.5
1105	4.71e+008	1878.3	7829071	0.0249	2.098	4354502	1.14e+009	177.7
1095	4.69e+008	1882.5	7830953	0.0252	2.123	4347608	1.14e+009	178.0
1085	4.66e+008	1884.3	7832838	0.0254	2.148	4340644	1.15e+009	178.2
1075	4.64e+008	1891.2	7834729	0.0257	2.174	4333591	1.15e+009	178.5
1065	4.62e+008	1892.1	7836621	0.0260	2.200	4326469	1.15e+009	178.7
1055	4.60e+008	1893.8	7838515	0.0263	2.226	4319274	1.15e+009	179.0
1045	4.58e+008	1904.5	7840419	0.0267	2.253	4311970	1.15e+009	179.2
1035	4.56e+008	1902.0	7842321	0.0269	2.280	4304606	1.16e+009	179.5
1025	4.54e+008	1915.9	7844237	0.0273	2.307	4297117	1.16e+009	179.7
1015	4.51e+008	1913.2	7846150	0.0276	2.335	4289566	1.16e+009	180.0
1005	4.49e+008	1920.4	7848071	0.0279	2.363	4281912	1.16e+009	180.2
995	4.47e+008	1921.6	7849992	0.0282	2.391	4274178	1.17e+009	180.5
985	4.45e+008	1931.8	7851924	0.0287	2.420	4266324	1.17e+009	180.7
975	4.43e+008	1935.5	7853859	0.0290	2.449	4258376	1.17e+009	181.0
965	4.40e+008	1942.2	7855802	0.0294	2.478	4250319	1.17e+009	181.3
955	4.38e+008	1946.6	7857748	0.0298	2.508	4242161	1.17e+009	181.5
945	4.36e+008	1951.7	7859700	0.0302	2.538	4233896	1.18e+009	181.8
935	4.34e+008	1959.8	7861660	0.0306	2.569	4225509	1.18e+009	182.0
925	4.32e+008	1957.6	7863617	0.0309	2.600	4217043	1.18e+009	182.3
915	4.29e+008	1972.5	7865590	0.0315	2.631	4208420	1.18e+009	182.6
905	4.27e+008	1978.8	7867569	0.0319	2.663	4199676	1.19e+009	182.8
895	4.25e+008	1985.1	7869554	0.0324	2.695	4190807	1.19e+009	183.1
885	4.23e+008	1991.3	7871545	0.0328	2.728	4181812	1.19e+009	183.4

875	4.20e+008	1991.7	7873537	0.0332	2.761	4172714	1.19e+009	183.6
865	4.18e+008	2003.5	7875540	0.0338	2.795	4163458	1.20e+009	183.9
855	4.16e+008	2010.5	7877551	0.0343	2.829	4154063	1.20e+009	184.1
845	4.14e+008	2018.1	7879569	0.0348	2.864	4144523	1.20e+009	184.4
835	4.11e+008	2029.6	7881598	0.0354	2.900	4134815	1.20e+009	184.7
825	4.09e+008	2030.8	7883629	0.0359	2.935	4124986	1.21e+009	185.0
815	4.07e+008	2048.4	7885678	0.0366	2.972	4114952	1.21e+009	185.2
805	4.04e+008	2044.1	7887722	0.0370	3.009	4104817	1.21e+009	185.5
795	4.02e+008	2064.3	7889786	0.0378	3.047	4094456	1.21e+009	185.8
785	4.00e+008	2065.8	7891852	0.0383	3.085	4083957	1.22e+009	186.0
775	3.98e+008	2080.1	7893932	0.0391	3.124	4073252	1.22e+009	186.3
765	3.95e+008	2081.6	7896014	0.0396	3.164	4062402	1.22e+009	186.6
755	3.93e+008	2100.0	7898114	0.0405	3.204	4051314	1.22e+009	186.9
745	3.90e+008	2108.0	7900222	0.0412	3.246	4040037	1.23e+009	187.2
735	3.88e+008	2112.9	7902335	0.0418	3.287	4028582	1.23e+009	187.4
725	3.86e+008	2124.3	7904459	0.0426	3.330	4016911	1.23e+009	187.7
715	3.83e+008	2143.1	7906602	0.0436	3.373	4004975	1.23e+009	188.0
705	3.81e+008	2144.9	7908747	0.0442	3.418	3992863	1.24e+009	188.3
695	3.78e+008	2162.5	7910909	0.0452	3.463	3980479	1.24e+009	188.6
685	3.76e+008	2174.3	7913084	0.0461	3.509	3967850	1.24e+009	188.9
675	3.74e+008	2180.6	7915264	0.0469	3.556	3955000	1.24e+009	189.1
665	3.71e+008	2198.6	7917463	0.0480	3.604	3941854	1.25e+009	189.4
655	3.69e+008	2209.2	7919672	0.0489	3.653	3928448	1.25e+009	189.7
645	3.66e+008	2223.2	7921895	0.0500	3.703	3914751	1.25e+009	190.0
635	3.64e+008	2239.1	7924134	0.0511	3.754	3900745	1.25e+009	190.3
625	3.61e+008	2253.9	7926388	0.0523	3.806	3886426	1.26e+009	190.6
615	3.59e+008	2259.6	7928648	0.0532	3.859	3871842	1.26e+009	190.9
605	3.56e+008	2282.9	7930931	0.0546	3.914	3856870	1.26e+009	191.2
595	3.54e+008	2297.2	7933228	0.0559	3.970	3841557	1.27e+009	191.5
585	3.51e+008	2312.6	7935541	0.0572	4.027	3829395	1.27e+009	191.8
575	3.48e+008	2332.1	7937873	0.0587	4.086	3820936	1.27e+009	192.1
565	3.46e+008	2346.4	7940219	0.0600	4.146	3812278	1.27e+009	192.4
555	3.43e+008	2366.6	7942586	0.0616	4.207	3803392	1.28e+009	192.7
545	3.41e+008	2390.1	7944976	0.0633	4.271	3794258	1.28e+009	193.1
535	3.38e+008	2404.2	7947380	0.0649	4.336	3784903	1.28e+009	193.4
525	3.35e+008	2428.9	7949809	0.0668	4.402	3775276	1.29e+009	193.7
515	3.33e+008	2448.9	7952258	0.0686	4.471	3765387	1.29e+009	194.0
505	3.30e+008	2468.4	7954726	0.0704	4.541	3755227	1.29e+009	194.3
495	3.27e+008	2497.3	7957223	0.0727	4.614	3744747	1.29e+009	194.7
485	3.24e+008	2520.7	7959744	0.0748	4.689	3733957	1.30e+009	195.0
475	3.22e+008	2545.1	7962289	0.0771	4.766	3722840	1.30e+009	195.3
465	3.19e+008	2569.0	7964858	0.0794	4.845	3711384	1.30e+009	195.7
455	3.16e+008	2597.9	7967456	0.0820	4.927	3699553	1.31e+009	196.0
445	3.13e+008	2630.4	7970087	0.0849	5.012	3687314	1.31e+009	196.3
435	3.10e+008	2658.0	7972745	0.0877	5.100	3674671	1.31e+009	196.7
425	3.07e+008	2688.3	7975433	0.0907	5.191	3661593	1.32e+009	197.0
415	3.04e+008	2719.4	7978152	0.0939	5.284	3648057	1.32e+009	197.4
405	3.01e+008	2757.7	7980910	0.0975	5.382	3634002	1.32e+009	197.8
395	2.98e+008	2796.4	7983706	0.1012	5.483	3619402	1.33e+009	198.1
385	2.95e+008	2826.6	7986533	0.1049	5.588	3604275	1.33e+009	198.5
375	2.92e+008	2872.0	7989405	0.1093	5.697	3588511	1.33e+009	198.9
365	2.89e+008	2910.5	7992316	0.1137	5.811	3572115	1.34e+009	199.2
355	2.86e+008	2953.3	7995269	0.1185	5.929	3555028	1.34e+009	199.6
345	2.83e+008	3007.0	7998276	0.1240	6.053	3539537	1.34e+009	200.0
335	2.80e+008	3052.4	8001328	0.1295	6.183	3526654	1.35e+009	200.4
325	2.76e+008	3103.3	8004432	0.1355	6.318	3513171	1.35e+009	200.8
315	2.73e+008	3165.3	8007597	0.1424	6.461	3499002	1.35e+009	201.2
305	2.70e+008	3224.1	8010821	0.1496	6.610	3484117	1.36e+009	201.6
295	2.66e+008	3287.2	8014108	0.1574	6.768	3468452	1.36e+009	202.1
285	2.63e+008	3354.9	8017463	0.1660	6.934	3451930	1.37e+009	202.5
275	2.59e+008	3430.8	8020894	0.1756	7.110	3434452	1.37e+009	202.9
265	2.56e+008	3510.5	8024406	0.1862	7.296	3415927	1.38e+009	203.4
255	2.52e+008	3591.9	8027996	0.1975	7.493	3396271	1.38e+009	203.8
245	2.48e+008	3684.3	8031681	0.2104	7.704	3375332	1.38e+009	204.3
235	2.44e+008	3782.7	8035463	0.2247	7.928	3352973	1.39e+009	204.8
225	2.40e+008	3891.5	8039355	0.2408	8.169	3332670	1.39e+009	205.3
215	2.36e+008	4008.3	8043363	0.2588	8.428	3312514	1.40e+009	205.8
205	2.32e+008	4136.1	8047499	0.2792	8.707	3290768	1.40e+009	206.3

200	2.30e+008	2134.1	8049633	0.1474	8.855	3279287	1.41e+009	206.6
195	2.28e+008	2171.4	8051805	0.1536	9.008	3267327	1.41e+009	206.9
190	2.26e+008	2205.8	8054011	0.1598	9.168	3254880	1.41e+009	207.2
185	2.24e+008	2253.6	8056264	0.1674	9.335	3241845	1.41e+009	207.4
180	2.22e+008	2297.7	8058562	0.1750	9.510	3228214	1.42e+009	207.7
175	2.19e+008	2341.9	8060904	0.1831	9.694	3213955	1.42e+009	208.0
170	2.17e+008	2395.7	8063299	0.1924	9.886	3198972	1.42e+009	208.3
165	2.15e+008	2444.7	8065744	0.2018	10.088	3184953	1.42e+009	208.6
160	2.12e+008	2505.8	8068250	0.2127	10.300	3172496	1.43e+009	209.0
155	2.10e+008	2568.8	8070819	0.2245	10.525	3159350	1.43e+009	209.3
150	2.08e+008	2633.2	8073452	0.2371	10.762	3145466	1.43e+009	209.6
145	2.05e+008	2704.5	8076156	0.2512	11.013	3130759	1.44e+009	210.0
140	2.03e+008	2784.6	8078941	0.2670	11.280	3115127	1.44e+009	210.3
135	2.00e+008	2863.4	8081804	0.2837	11.564	3098516	1.44e+009	210.7
130	1.98e+008	2953.5	8084758	0.3027	11.867	3080790	1.45e+009	211.0
125	1.95e+008	3048.4	8087806	0.3236	12.190	3061840	1.45e+009	211.4
120	1.92e+008	3155.5	8090962	0.3475	12.538	3041496	1.45e+009	211.8
115	1.89e+008	3274.0	8094236	0.3744	12.912	3019574	1.46e+009	212.2
110	1.86e+008	3393.6	8097629	0.4036	13.316	2995940	1.46e+009	212.7
105	1.83e+008	3540.2	8101169	0.4387	13.754	2970256	1.47e+009	213.1
100	1.80e+008	3693.7	8104863	0.4776	14.232	2942289	1.47e+009	213.6
95	1.77e+008	3862.4	8108726	0.5222	14.754	2911712	1.48e+009	214.0
90	1.74e+008	4052.4	8112778	0.5741	15.328	2883403	1.48e+009	214.5
85	1.71e+008	4276.6	8117055	0.6362	15.965	2856427	1.49e+009	215.1
80	1.67e+008	4533.1	8121588	0.7100	16.675	2826324	1.49e+009	215.6
75	1.64e+008	4835.3	8126423	0.7995	17.474	2792425	1.50e+009	216.2
70	1.60e+008	5172.7	8131596	0.9058	18.380	2754019	1.50e+009	216.9
65	1.56e+008	5585.0	8137181	1.0393	19.419	2709951	1.51e+009	217.5
60	1.52e+008	6064.3	8143245	1.2041	20.623	2664651	1.52e+009	218.3
55	1.48e+008	6659.2	8149904	1.4171	22.040	2617647	1.52e+009	219.1
50	1.43e+008	7420.5	8157325	1.7011	23.741	2561222	1.53e+009	220.0

torus water level = 17.00 ft

prpv [psig]	trpv [°F]	hf_rpv [Btu/lbm]	hg_rpv [Btu/lbm]	vf_rpv [ft3/lbm]	vg_rpv [ft3/lbm]	wsrv [lbm/min.]	mf_rpv [lbm]	mg_rpv [lbm]
1145	562.88	566.16	1186.6	0.02217	0.3771	78190	532000	15300
1135	561.79	564.74	1187.0	0.02213	0.3808	77515	530124	15309
1125	560.70	563.31	1187.4	0.02210	0.3847	76841	528251	15314
1115	559.60	561.87	1187.8	0.02206	0.3886	76167	526376	15316
1105	558.49	560.42	1188.3	0.02202	0.3926	75493	524498	15315
1095	557.37	558.97	1188.7	0.02199	0.3966	74819	522619	15312
1085	556.24	557.51	1189.1	0.02195	0.4007	74144	520742	15305
1075	555.11	556.04	1189.5	0.02192	0.4049	73470	518860	15295
1065	553.97	554.57	1189.9	0.02188	0.4091	72796	516980	15283
1055	552.82	553.09	1190.3	0.02184	0.4135	72122	515102	15268
1045	551.66	551.60	1190.7	0.02181	0.4179	71447	513216	15250
1035	550.50	550.11	1191.1	0.02177	0.4224	70773	511335	15229
1025	549.32	548.60	1191.4	0.02173	0.4269	70099	509442	15205
1015	548.14	547.09	1191.8	0.02170	0.4316	69425	507556	15179
1005	546.95	545.57	1192.2	0.02166	0.4363	68751	505664	15150
995	545.75	544.04	1192.6	0.02163	0.4412	68076	503775	15118
985	544.54	542.51	1192.9	0.02159	0.4461	67402	501877	15083
975	543.32	540.96	1193.3	0.02155	0.4511	66728	499979	15046
965	542.10	539.40	1193.7	0.02152	0.4562	66054	498076	15007
955	540.86	537.85	1194.0	0.02148	0.4615	65379	496171	14965
945	539.61	536.27	1194.4	0.02145	0.4668	64705	494264	14920
935	538.35	534.70	1194.7	0.02141	0.4722	64031	492352	14873
925	537.09	533.11	1195.1	0.02137	0.4778	63357	490444	14823
915	535.81	531.51	1195.4	0.02134	0.4834	62683	488524	14771
905	534.52	529.90	1195.7	0.02130	0.4892	62008	486600	14716
895	533.22	528.28	1196.1	0.02127	0.4951	61334	484671	14659
885	531.91	526.65	1196.4	0.02123	0.5011	60660	482740	14600
875	530.59	525.02	1196.7	0.02119	0.5072	59986	480810	14538
865	529.26	523.37	1197.0	0.02116	0.5135	59311	478871	14474
855	527.92	521.71	1197.3	0.02112	0.5199	58637	476927	14407
845	526.56	520.04	1197.7	0.02109	0.5265	57963	474977	14338
835	525.20	518.35	1198.0	0.02105	0.5332	57289	473019	14267

825	523.82	516.66	1198.3	0.02101	0.5401	56614	471061	14194
815	522.42	514.95	1198.5	0.02098	0.5471	55940	469088	14118
805	521.02	513.24	1198.8	0.02094	0.5543	55266	467123	14040
795	519.60	511.50	1199.1	0.02090	0.5617	54592	465138	13960
785	518.16	509.76	1199.4	0.02087	0.5692	53918	463155	13878
775	516.72	508.00	1199.7	0.02083	0.5769	53243	461159	13793
765	515.26	506.24	1199.9	0.02080	0.5848	52569	459165	13706
755	513.78	504.45	1200.2	0.02076	0.5929	51895	457153	13618
745	512.30	502.65	1200.4	0.02072	0.6012	51221	455136	13527
735	510.80	500.84	1200.7	0.02069	0.6097	50546	453117	13433
725	509.28	499.01	1200.9	0.02065	0.6185	49872	451088	13338
715	507.74	497.17	1201.2	0.02061	0.6275	49198	449042	13241
705	506.19	495.31	1201.4	0.02058	0.6367	48524	446997	13141
695	504.62	493.43	1201.6	0.02054	0.6461	47850	444936	13039
685	503.03	491.54	1201.8	0.02050	0.6559	47175	442865	12936
675	501.43	489.64	1202.1	0.02047	0.6658	46501	440790	12830
665	499.81	487.71	1202.3	0.02043	0.6761	45827	438699	12722
655	498.17	485.77	1202.5	0.02039	0.6867	45153	436600	12613
645	496.52	483.81	1202.7	0.02036	0.6976	44478	434488	12501
635	494.84	481.83	1202.8	0.02032	0.7088	43804	432363	12387
625	493.14	479.83	1203.0	0.02028	0.7203	43130	430225	12271
615	491.43	477.82	1203.2	0.02024	0.7322	42456	428084	12153
605	489.69	475.78	1203.4	0.02021	0.7445	41782	425920	12033
595	487.93	473.72	1203.5	0.02017	0.7571	41107	423745	11911
585	486.15	471.64	1203.7	0.02013	0.7701	40433	421556	11788
575	484.35	469.53	1203.8	0.02009	0.7836	39759	419350	11662
565	482.52	467.40	1203.9	0.02005	0.7975	39085	417131	11534
555	480.66	465.25	1204.1	0.02002	0.8119	38410	414894	11404
545	478.78	463.07	1204.2	0.01998	0.8268	37736	412636	11273
535	476.88	460.87	1204.3	0.01994	0.8422	37062	410365	11139
525	474.95	458.64	1204.4	0.01990	0.8582	36388	408072	11004
515	472.99	456.39	1204.5	0.01986	0.8747	35714	405760	10866
505	471.01	454.11	1204.5	0.01982	0.8919	35039	403432	10727
495	468.99	451.80	1204.6	0.01978	0.9097	34365	401076	10585
485	466.95	449.45	1204.7	0.01974	0.9282	33691	398698	10442
475	464.87	447.08	1204.7	0.01970	0.9474	33017	396298	10297
465	462.76	444.68	1204.8	0.01966	0.9674	32342	393876	10150
455	460.61	442.24	1204.8	0.01962	0.9882	31668	391427	10001
445	458.43	439.76	1204.8	0.01958	1.0099	30994	388948	9850
435	456.21	437.25	1204.8	0.01954	1.0325	30320	386443	9697
425	453.96	434.69	1204.8	0.01950	1.0561	29646	383910	9542
415	451.67	432.11	1204.8	0.01946	1.0807	28971	381347	9385
405	449.33	429.48	1204.7	0.01942	1.1065	28297	378748	9226
395	446.95	426.80	1204.7	0.01938	1.1335	27623	376113	9065
385	444.53	424.09	1204.6	0.01934	1.1618	26949	373449	8902
375	442.06	421.32	1204.5	0.01929	1.1915	26274	370742	8738
365	439.53	418.51	1204.4	0.01925	1.2227	25600	367998	8571
355	436.96	415.64	1204.3	0.01921	1.2556	24926	365214	8402
345	434.33	412.72	1204.1	0.01916	1.2902	24252	362377	8231
335	431.65	409.74	1204.0	0.01912	1.3267	23578	359498	8058
325	428.91	406.71	1203.8	0.01908	1.3652	22903	356570	7883
315	426.10	403.60	1203.6	0.01903	1.4061	22229	353581	7706
305	423.22	400.43	1203.4	0.01899	1.4493	21555	350536	7527
295	420.27	397.20	1203.1	0.01894	1.4953	20881	347431	7346
285	417.26	393.89	1202.9	0.01889	1.5443	20206	344259	7162
275	414.16	390.50	1202.6	0.01885	1.5964	19532	341014	6977
265	410.97	387.01	1202.3	0.01880	1.6522	18858	337691	6789
255	407.70	383.45	1201.9	0.01875	1.7120	18184	334290	6598
245	404.33	379.79	1201.5	0.01870	1.7762	17510	330798	6406
235	400.87	376.03	1201.1	0.01865	1.8453	16835	327210	6211
225	397.28	372.15	1200.6	0.01860	1.9200	16161	323515	6014
215	393.59	368.16	1200.1	0.01855	2.0010	15487	319707	5814
205	389.76	364.04	1199.6	0.01849	2.0891	14813	315773	5612
200	387.79	361.93	1199.3	0.01847	2.1360	14476	313741	5510
195	385.79	359.78	1199.0	0.01844	2.1852	14138	311672	5408
190	383.76	357.60	1198.6	0.01841	2.2367	13801	309569	5304
185	381.67	355.38	1198.3	0.01839	2.2906	13464	307420	5200
180	379.56	353.10	1198.0	0.01836	2.3472	13127	305227	5096

175	377.40	350.80	1197.6	0.01833	2.4067	12790	302990	4990
170	375.19	348.44	1197.2	0.01830	2.4693	12453	300701	4884
165	372.94	346.04	1196.8	0.01827	2.5353	12116	298363	4778
160	370.64	343.59	1196.4	0.01824	2.6048	11779	295964	4670
155	368.28	341.09	1196.0	0.01821	2.6784	11442	293504	4562
150	365.87	338.52	1195.5	0.01818	2.7562	11104	290979	4453
145	363.40	335.91	1195.1	0.01815	2.8387	10767	288385	4343
140	360.87	333.23	1194.6	0.01812	2.9264	10430	285711	4233
135	358.27	330.48	1194.0	0.01809	3.0197	10093	282959	4122
130	355.61	327.67	1193.5	0.01806	3.1192	9756	280117	4009
125	352.87	324.78	1192.9	0.01802	3.2256	9419	277182	3896
120	350.06	321.82	1192.3	0.01799	3.3396	9082	274140	3783
115	347.15	318.76	1191.7	0.01796	3.4620	8745	270981	3668
110	344.17	315.63	1191.0	0.01792	3.5940	8408	267703	3552
105	341.08	312.39	1190.3	0.01789	3.7365	8070	264279	3436
100	337.89	309.05	1189.6	0.01785	3.8910	7733	260703	3318
95	334.59	305.59	1188.8	0.01781	4.0590	7396	256959	3200
90	331.16	302.02	1188.0	0.01778	4.2425	7059	253026	3080
85	327.60	298.31	1187.1	0.01774	4.4437	6722	248870	2959
80	323.90	294.46	1186.2	0.01770	4.6654	6385	244459	2837
75	320.04	290.44	1185.2	0.01766	4.9109	6048	239747	2715
70	316.01	286.26	1184.1	0.01761	5.1843	5711	234698	2591
65	311.78	281.88	1183.0	0.01757	5.4907	5374	229238	2465
60	307.33	277.28	1181.8	0.01753	5.8367	5036	223300	2339
55	302.64	272.44	1180.5	0.01748	6.2304	4699	216769	2211
50	297.67	267.31	1179.1	0.01743	6.6827	4362	209478	2082

prpv [psig]	erpv [Btu]	dmg_srv [lbm]	mf_sp [lbm]	dt [min.]	t [min.]	qdh [Btu/min.]	esp [Btu]	tsp [°F]
1145	4.79e+008	0.0	8400693	0.0000	2.000	4381320	1.19e+009	173.7
1135	4.77e+008	1867.9	8402561	0.0241	2.024	4374717	1.19e+009	173.9
1125	4.75e+008	1867.4	8404428	0.0243	2.048	4368058	1.19e+009	174.1
1115	4.73e+008	1872.9	8406301	0.0246	2.073	4361320	1.20e+009	174.4
1105	4.71e+008	1878.3	8408179	0.0249	2.098	4354502	1.20e+009	174.6
1095	4.69e+008	1882.5	8410062	0.0252	2.123	4347608	1.20e+009	174.8
1085	4.66e+008	1884.3	8411946	0.0254	2.148	4340644	1.20e+009	175.1
1075	4.64e+008	1891.2	8413837	0.0257	2.174	4333591	1.21e+009	175.3
1065	4.62e+008	1892.1	8415729	0.0260	2.200	4326469	1.21e+009	175.5
1055	4.60e+008	1893.8	8417623	0.0263	2.226	4319274	1.21e+009	175.8
1045	4.58e+008	1904.5	8419528	0.0267	2.253	4311970	1.21e+009	176.0
1035	4.56e+008	1902.0	8421430	0.0269	2.280	4304606	1.21e+009	176.2
1025	4.54e+008	1915.9	8423346	0.0273	2.307	4297117	1.22e+009	176.5
1015	4.51e+008	1913.2	8425259	0.0276	2.335	4289566	1.22e+009	176.7
1005	4.49e+008	1920.4	8427179	0.0279	2.363	4281912	1.22e+009	177.0
995	4.47e+008	1921.6	8429101	0.0282	2.391	4274178	1.22e+009	177.2
985	4.45e+008	1931.8	8431032	0.0287	2.420	4266324	1.23e+009	177.4
975	4.43e+008	1935.5	8432968	0.0290	2.449	4258376	1.23e+009	177.7
965	4.40e+008	1942.2	8434910	0.0294	2.478	4250319	1.23e+009	177.9
955	4.38e+008	1946.6	8436857	0.0298	2.508	4242161	1.23e+009	178.2
945	4.36e+008	1951.7	8438808	0.0302	2.538	4233896	1.24e+009	178.4
935	4.34e+008	1959.8	8440768	0.0306	2.569	4225509	1.24e+009	178.6
925	4.32e+008	1957.6	8442726	0.0309	2.600	4217043	1.24e+009	178.9
915	4.29e+008	1972.5	8444698	0.0315	2.631	4208420	1.24e+009	179.1
905	4.27e+008	1978.8	8446677	0.0319	2.663	4199676	1.24e+009	179.4
895	4.25e+008	1985.1	8448662	0.0324	2.695	4190807	1.25e+009	179.6
885	4.23e+008	1991.3	8450654	0.0328	2.728	4181812	1.25e+009	179.9
875	4.20e+008	1991.7	8452645	0.0332	2.761	4172714	1.25e+009	180.1
865	4.18e+008	2003.5	8454649	0.0338	2.795	4163458	1.25e+009	180.4
855	4.16e+008	2010.5	8456659	0.0343	2.829	4154063	1.26e+009	180.6
845	4.14e+008	2018.1	8458677	0.0348	2.864	4144523	1.26e+009	180.9
835	4.11e+008	2029.6	8460707	0.0354	2.900	4134815	1.26e+009	181.1
825	4.09e+008	2030.8	8462738	0.0359	2.935	4124986	1.26e+009	181.4
815	4.07e+008	2048.4	8464786	0.0366	2.972	4114952	1.27e+009	181.6
805	4.04e+008	2044.1	8466830	0.0370	3.009	4104817	1.27e+009	181.9
795	4.02e+008	2064.3	8468895	0.0378	3.047	4094456	1.27e+009	182.1
785	4.00e+008	2065.8	8470960	0.0383	3.085	4083957	1.27e+009	182.4
775	3.98e+008	2080.1	8473041	0.0391	3.124	4073252	1.28e+009	182.6
765	3.95e+008	2081.6	8475122	0.0396	3.164	4062402	1.28e+009	182.9

755	3.93e+008	2100.0	8477222	0.0405	3.204	4051314	1.28e+009	183.2
745	3.90e+008	2108.0	8479330	0.0412	3.246	4040037	1.28e+009	183.4
735	3.88e+008	2112.9	8481443	0.0418	3.287	4028582	1.29e+009	183.7
725	3.86e+008	2124.3	8483567	0.0426	3.330	4016911	1.29e+009	183.9
715	3.83e+008	2143.1	8485710	0.0436	3.373	4004975	1.29e+009	184.2
705	3.81e+008	2144.9	8487855	0.0442	3.418	3992863	1.29e+009	184.5
695	3.78e+008	2162.5	8490018	0.0452	3.463	3980479	1.30e+009	184.7
685	3.76e+008	2174.3	8492192	0.0461	3.509	3967850	1.30e+009	185.0
675	3.74e+008	2180.6	8494373	0.0469	3.556	3955000	1.30e+009	185.3
665	3.71e+008	2198.6	8496571	0.0480	3.604	3941854	1.30e+009	185.6
655	3.69e+008	2209.2	8498781	0.0489	3.653	3928448	1.31e+009	185.8
645	3.66e+008	2223.2	8501004	0.0500	3.703	3914751	1.31e+009	186.1
635	3.64e+008	2239.1	8503243	0.0511	3.754	3900745	1.31e+009	186.4
625	3.61e+008	2253.9	8505497	0.0523	3.806	3886426	1.32e+009	186.7
615	3.59e+008	2259.6	8507756	0.0532	3.859	3871842	1.32e+009	186.9
605	3.56e+008	2282.9	8510039	0.0546	3.914	3856870	1.32e+009	187.2
595	3.54e+008	2297.2	8512336	0.0559	3.970	3841557	1.32e+009	187.5
585	3.51e+008	2312.6	8514649	0.0572	4.027	3829395	1.33e+009	187.8
575	3.48e+008	2332.1	8516981	0.0587	4.086	3820936	1.33e+009	188.1
565	3.46e+008	2346.4	8519328	0.0600	4.146	3812278	1.33e+009	188.4
555	3.43e+008	2366.6	8521694	0.0616	4.207	3803392	1.33e+009	188.6
545	3.41e+008	2390.1	8524084	0.0633	4.271	3794258	1.34e+009	188.9
535	3.38e+008	2404.2	8526488	0.0649	4.336	3784903	1.34e+009	189.2
525	3.35e+008	2428.9	8528917	0.0668	4.402	3775276	1.34e+009	189.5
515	3.33e+008	2448.9	8531366	0.0686	4.471	3765387	1.35e+009	189.8
505	3.30e+008	2468.4	8533835	0.0704	4.541	3755227	1.35e+009	190.1
495	3.27e+008	2497.3	8536332	0.0727	4.614	3744747	1.35e+009	190.4
485	3.24e+008	2520.7	8538853	0.0748	4.689	3733957	1.36e+009	190.8
475	3.22e+008	2545.1	8541398	0.0771	4.766	3722840	1.36e+009	191.1
465	3.19e+008	2569.0	8543967	0.0794	4.845	3711384	1.36e+009	191.4
455	3.16e+008	2597.9	8546565	0.0820	4.927	3699553	1.36e+009	191.7
445	3.13e+008	2630.4	8549195	0.0849	5.012	3687314	1.37e+009	192.0
435	3.10e+008	2658.0	8551853	0.0877	5.100	3674671	1.37e+009	192.3
425	3.07e+008	2688.3	8554541	0.0907	5.191	3661593	1.37e+009	192.7
415	3.04e+008	2719.4	8557261	0.0939	5.284	3648057	1.38e+009	193.0
405	3.01e+008	2757.7	8560019	0.0975	5.382	3634002	1.38e+009	193.3
395	2.98e+008	2796.4	8562815	0.1012	5.483	3619402	1.38e+009	193.7
385	2.95e+008	2826.6	8565642	0.1049	5.588	3604275	1.39e+009	194.0
375	2.92e+008	2872.0	8568514	0.1093	5.697	3588511	1.39e+009	194.4
365	2.89e+008	2910.5	8571424	0.1137	5.811	3572115	1.39e+009	194.7
355	2.86e+008	2953.3	8574377	0.1185	5.929	3555028	1.40e+009	195.1
345	2.83e+008	3007.0	8577384	0.1240	6.053	3539537	1.40e+009	195.5
335	2.80e+008	3052.4	8580437	0.1295	6.183	3526654	1.41e+009	195.8
325	2.76e+008	3103.3	8583540	0.1355	6.318	3513171	1.41e+009	196.2
315	2.73e+008	3165.3	8586705	0.1424	6.461	3499002	1.41e+009	196.6
305	2.70e+008	3224.1	8589930	0.1496	6.610	3484117	1.42e+009	197.0
295	2.66e+008	3287.2	8593217	0.1574	6.768	3468452	1.42e+009	197.4
285	2.63e+008	3354.9	8596572	0.1660	6.934	3451930	1.43e+009	197.8
275	2.59e+008	3430.8	8600002	0.1756	7.110	3434452	1.43e+009	198.2
265	2.56e+008	3510.5	8603513	0.1862	7.296	3415927	1.43e+009	198.6
255	2.52e+008	3591.9	8607105	0.1975	7.493	3396271	1.44e+009	199.0
245	2.48e+008	3684.3	8610789	0.2104	7.704	3375332	1.44e+009	199.5
235	2.44e+008	3782.7	8614572	0.2247	7.928	3352973	1.45e+009	199.9
225	2.40e+008	3891.5	8618463	0.2408	8.169	3332670	1.45e+009	200.4
215	2.36e+008	4008.3	8622472	0.2588	8.428	3312514	1.46e+009	200.9
205	2.32e+008	4136.1	8626608	0.2792	8.707	3290768	1.46e+009	201.4
200	2.30e+008	2134.1	8628742	0.1474	8.855	3279287	1.46e+009	201.6
195	2.28e+008	2171.4	8630913	0.1536	9.008	3267327	1.47e+009	201.9
190	2.26e+008	2205.8	8633119	0.1598	9.168	3254880	1.47e+009	202.2
185	2.24e+008	2253.6	8635373	0.1674	9.335	3241845	1.47e+009	202.4
180	2.22e+008	2297.7	8637670	0.1750	9.510	3228214	1.47e+009	202.7
175	2.19e+008	2341.9	8640012	0.1831	9.694	3213955	1.48e+009	203.0
170	2.17e+008	2395.7	8642408	0.1924	9.886	3198972	1.48e+009	203.3
165	2.15e+008	2444.7	8644853	0.2018	10.088	3184953	1.48e+009	203.5
160	2.12e+008	2505.8	8647358	0.2127	10.300	3172496	1.49e+009	203.8
155	2.10e+008	2568.8	8649927	0.2245	10.525	3159350	1.49e+009	204.1
150	2.08e+008	2633.2	8652560	0.2371	10.762	3145466	1.49e+009	204.5
145	2.05e+008	2704.5	8655265	0.2512	11.013	3130759	1.50e+009	204.8

140	2.03e+008	2784.6	8658049	0.2670	11.280	3115127	1.50e+009	205.1
135	2.00e+008	2863.4	8660913	0.2837	11.564	3098516	1.50e+009	205.4
130	1.98e+008	2953.5	8663866	0.3027	11.867	3080790	1.51e+009	205.8
125	1.95e+008	3048.4	8666915	0.3236	12.190	3061840	1.51e+009	206.2
120	1.92e+008	3155.5	8670070	0.3475	12.538	3041496	1.51e+009	206.5
115	1.89e+008	3274.0	8673344	0.3744	12.912	3019574	1.52e+009	206.9
110	1.86e+008	3393.6	8676738	0.4036	13.316	2995940	1.52e+009	207.3
105	1.83e+008	3540.2	8680278	0.4387	13.754	2970256	1.53e+009	207.7
100	1.80e+008	3693.7	8683972	0.4776	14.232	2942289	1.53e+009	208.1
95	1.77e+008	3862.4	8687834	0.5222	14.754	2911712	1.53e+009	208.6
90	1.74e+008	4052.4	8691887	0.5741	15.328	2883403	1.54e+009	209.1
85	1.71e+008	4276.6	8696163	0.6362	15.965	2856427	1.54e+009	209.6
80	1.67e+008	4533.1	8700696	0.7100	16.675	2826324	1.55e+009	210.1
75	1.64e+008	4835.3	8705532	0.7995	17.474	2792425	1.56e+009	210.7
70	1.60e+008	5172.7	8710704	0.9058	18.380	2754019	1.56e+009	211.3
65	1.56e+008	5585.0	8716289	1.0393	19.419	2709951	1.57e+009	211.9
60	1.52e+008	6064.3	8722354	1.2041	20.623	2664651	1.58e+009	212.6
55	1.48e+008	6659.2	8729013	1.4171	22.040	2617647	1.58e+009	213.4
50	1.43e+008	7420.5	8736433	1.7011	23.741	2561222	1.59e+009	214.2

torus water level = 17.85 ft

prpv [psig]	trpv [°F]	hf_rpv [Btu/lbm]	hg_rpv [Btu/lbm]	vf_rpv [ft3/lbm]	vg_rpv [ft3/lbm]	wsrv [lbm/min.]	mf_rpv [lbm]	mg_rpv [lbm]
1145	562.88	566.16	1186.6	0.02217	0.3771	78190	532000	15300
1135	561.79	564.74	1187.0	0.02213	0.3808	77515	530124	15309
1125	560.70	563.31	1187.4	0.02210	0.3847	76841	528251	15314
1115	559.60	561.87	1187.8	0.02206	0.3886	76167	526376	15316
1105	558.49	560.42	1188.3	0.02202	0.3926	75493	524498	15315
1095	557.37	558.97	1188.7	0.02199	0.3966	74819	522619	15312
1085	556.24	557.51	1189.1	0.02195	0.4007	74144	520742	15305
1075	555.11	556.04	1189.5	0.02192	0.4049	73470	518860	15295
1065	553.97	554.57	1189.9	0.02188	0.4091	72796	516980	15283
1055	552.82	553.09	1190.3	0.02184	0.4135	72122	515102	15268
1045	551.66	551.60	1190.7	0.02181	0.4179	71447	513216	15250
1035	550.50	550.11	1191.1	0.02177	0.4224	70773	511335	15229
1025	549.32	548.60	1191.4	0.02173	0.4269	70099	509442	15205
1015	548.14	547.09	1191.8	0.02170	0.4316	69425	507556	15179
1005	546.95	545.57	1192.2	0.02166	0.4363	68751	505664	15150
995	545.75	544.04	1192.6	0.02163	0.4412	68076	503775	15118
985	544.54	542.51	1192.9	0.02159	0.4461	67402	501877	15083
975	543.32	540.96	1193.3	0.02155	0.4511	66728	499979	15046
965	542.10	539.40	1193.7	0.02152	0.4562	66054	498076	15007
955	540.86	537.85	1194.0	0.02148	0.4615	65379	496171	14965
945	539.61	536.27	1194.4	0.02145	0.4668	64705	494264	14920
935	538.35	534.70	1194.7	0.02141	0.4722	64031	492352	14873
925	537.09	533.11	1195.1	0.02137	0.4778	63357	490444	14823
915	535.81	531.51	1195.4	0.02134	0.4834	62683	488524	14771
905	534.52	529.90	1195.7	0.02130	0.4892	62008	486600	14716
895	533.22	528.28	1196.1	0.02127	0.4951	61334	484671	14659
885	531.91	526.65	1196.4	0.02123	0.5011	60660	482740	14600
875	530.59	525.02	1196.7	0.02119	0.5072	59986	480810	14538
865	529.26	523.37	1197.0	0.02116	0.5135	59311	478871	14474
855	527.92	521.71	1197.3	0.02112	0.5199	58637	476927	14407
845	526.56	520.04	1197.7	0.02109	0.5265	57963	474977	14338
835	525.20	518.35	1198.0	0.02105	0.5332	57289	473019	14267
825	523.82	516.66	1198.3	0.02101	0.5401	56614	471061	14194
815	522.42	514.95	1198.5	0.02098	0.5471	55940	469088	14118
805	521.02	513.24	1198.8	0.02094	0.5543	55266	467123	14040
795	519.60	511.50	1199.1	0.02090	0.5617	54592	465138	13960
785	518.16	509.76	1199.4	0.02087	0.5692	53918	463155	13878
775	516.72	508.00	1199.7	0.02083	0.5769	53243	461159	13793
765	515.26	506.24	1199.9	0.02080	0.5848	52569	459165	13706
755	513.78	504.45	1200.2	0.02076	0.5929	51895	457153	13618
745	512.30	502.65	1200.4	0.02072	0.6012	51221	455136	13527
735	510.80	500.84	1200.7	0.02069	0.6097	50546	453117	13433
725	509.28	499.01	1200.9	0.02065	0.6185	49872	451088	13338
715	507.74	497.17	1201.2	0.02061	0.6275	49198	449042	13241

705	506.19	495.31	1201.4	0.02058	0.6367	48524	446997	13141
695	504.62	493.43	1201.6	0.02054	0.6461	47850	444936	13039
685	503.03	491.54	1201.8	0.02050	0.6559	47175	442865	12936
675	501.43	489.64	1202.1	0.02047	0.6658	46501	440790	12830
665	499.81	487.71	1202.3	0.02043	0.6761	45827	438699	12722
655	498.17	485.77	1202.5	0.02039	0.6867	45153	436600	12613
645	496.52	483.81	1202.7	0.02036	0.6976	44478	434488	12501
635	494.84	481.83	1202.8	0.02032	0.7088	43804	432363	12387
625	493.14	479.83	1203.0	0.02028	0.7203	43130	430225	12271
615	491.43	477.82	1203.2	0.02024	0.7322	42456	428084	12153
605	489.69	475.78	1203.4	0.02021	0.7445	41782	425920	12033
595	487.93	473.72	1203.5	0.02017	0.7571	41107	423745	11911
585	486.15	471.64	1203.7	0.02013	0.7701	40433	421556	11788
575	484.35	469.53	1203.8	0.02009	0.7836	39759	419350	11662
565	482.52	467.40	1203.9	0.02005	0.7975	39085	417131	11534
555	480.66	465.25	1204.1	0.02002	0.8119	38410	414894	11404
545	478.78	463.07	1204.2	0.01998	0.8268	37736	412636	11273
535	476.88	460.87	1204.3	0.01994	0.8422	37062	410365	11139
525	474.95	458.64	1204.4	0.01990	0.8582	36388	408072	11004
515	472.99	456.39	1204.5	0.01986	0.8747	35714	405760	10866
505	471.01	454.11	1204.5	0.01982	0.8919	35039	403432	10727
495	468.99	451.80	1204.6	0.01978	0.9097	34365	401076	10585
485	466.95	449.45	1204.7	0.01974	0.9282	33691	398698	10442
475	464.87	447.08	1204.7	0.01970	0.9474	33017	396298	10297
465	462.76	444.68	1204.8	0.01966	0.9674	32342	393876	10150
455	460.61	442.24	1204.8	0.01962	0.9882	31668	391427	10001
445	458.43	439.76	1204.8	0.01958	1.0099	30994	388948	9850
435	456.21	437.25	1204.8	0.01954	1.0325	30320	386443	9697
425	453.96	434.69	1204.8	0.01950	1.0561	29646	383910	9542
415	451.67	432.11	1204.8	0.01946	1.0807	28971	381347	9385
405	449.33	429.48	1204.7	0.01942	1.1065	28297	378748	9226
395	446.95	426.80	1204.7	0.01938	1.1335	27623	376113	9065
385	444.53	424.09	1204.6	0.01934	1.1618	26949	373449	8902
375	442.06	421.32	1204.5	0.01929	1.1915	26274	370742	8738
365	439.53	418.51	1204.4	0.01925	1.2227	25600	367998	8571
355	436.96	415.64	1204.3	0.01921	1.2556	24926	365214	8402
345	434.33	412.72	1204.1	0.01916	1.2902	24252	362377	8231
335	431.65	409.74	1204.0	0.01912	1.3267	23578	359498	8058
325	428.91	406.71	1203.8	0.01908	1.3652	22903	356570	7883
315	426.10	403.60	1203.6	0.01903	1.4061	22229	353581	7706
305	423.22	400.43	1203.4	0.01899	1.4493	21555	350536	7527
295	420.27	397.20	1203.1	0.01894	1.4953	20881	347431	7346
285	417.26	393.89	1202.9	0.01889	1.5443	20206	344259	7162
275	414.16	390.50	1202.6	0.01885	1.5964	19532	341014	6977
265	410.97	387.01	1202.3	0.01880	1.6522	18858	337691	6789
255	407.70	383.45	1201.9	0.01875	1.7120	18184	334290	6598
245	404.33	379.79	1201.5	0.01870	1.7762	17510	330798	6406
235	400.87	376.03	1201.1	0.01865	1.8453	16835	327210	6211
225	397.28	372.15	1200.6	0.01860	1.9200	16161	323515	6014
215	393.59	368.16	1200.1	0.01855	2.0010	15487	319707	5814
205	389.76	364.04	1199.6	0.01849	2.0891	14813	315773	5612
200	387.79	361.93	1199.3	0.01847	2.1360	14476	313741	5510
195	385.79	359.78	1199.0	0.01844	2.1852	14138	311672	5408
190	383.76	357.60	1198.6	0.01841	2.2367	13801	309569	5304
185	381.67	355.38	1198.3	0.01839	2.2906	13464	307420	5200
180	379.56	353.10	1198.0	0.01836	2.3472	13127	305227	5096
175	377.40	350.80	1197.6	0.01833	2.4067	12790	302990	4990
170	375.19	348.44	1197.2	0.01830	2.4693	12453	300701	4884
165	372.94	346.04	1196.8	0.01827	2.5353	12116	298363	4778
160	370.64	343.59	1196.4	0.01824	2.6048	11779	295964	4670
155	368.28	341.09	1196.0	0.01821	2.6784	11442	293504	4562
150	365.87	338.52	1195.5	0.01818	2.7562	11104	290979	4453
145	363.40	335.91	1195.1	0.01815	2.8387	10767	288385	4343
140	360.87	333.23	1194.6	0.01812	2.9264	10430	285711	4233
135	358.27	330.48	1194.0	0.01809	3.0197	10093	282959	4122
130	355.61	327.67	1193.5	0.01806	3.1192	9756	280117	4009
125	352.87	324.78	1192.9	0.01802	3.2256	9419	277182	3896
120	350.06	321.82	1192.3	0.01799	3.3396	9082	274140	3783

115	347.15	318.76	1191.7	0.01796	3.4620	8745	270981	3668
110	344.17	315.63	1191.0	0.01792	3.5940	8408	267703	3552
105	341.08	312.39	1190.3	0.01789	3.7365	8070	264279	3436
100	337.89	309.05	1189.6	0.01785	3.8910	7733	260703	3318
95	334.59	305.59	1188.8	0.01781	4.0590	7396	256959	3200
90	331.16	302.02	1188.0	0.01778	4.2425	7059	253026	3080
85	327.60	298.31	1187.1	0.01774	4.4437	6722	248870	2959
80	323.90	294.46	1186.2	0.01770	4.6654	6385	244459	2837
75	320.04	290.44	1185.2	0.01766	4.9109	6048	239747	2715
70	316.01	286.26	1184.1	0.01761	5.1843	5711	234698	2591
65	311.78	281.88	1183.0	0.01757	5.4907	5374	229238	2465
60	307.33	277.28	1181.8	0.01753	5.8367	5036	223300	2339
55	302.64	272.44	1180.5	0.01748	6.2304	4699	216769	2211
50	297.67	267.31	1179.1	0.01743	6.6827	4362	209478	2082

prpv [psig]	erpv [Btu]	dmg_srv [lbm]	mf_sp [lbm]	dt [min.]	t [min.]	qdh [Btu/min.]	esp [Btu]	tsp [°F]
1145	4.79e+008	0.0	8903894	0.0000	2.000	4381320	1.17e+009	163.1
1135	4.77e+008	1867.9	8905762	0.0241	2.024	4374717	1.17e+009	163.3
1125	4.75e+008	1867.4	8907630	0.0243	2.048	4368058	1.17e+009	163.5
1115	4.73e+008	1872.9	8909503	0.0246	2.073	4361320	1.17e+009	163.7
1105	4.71e+008	1878.3	8911381	0.0249	2.098	4354502	1.18e+009	164.0
1095	4.69e+008	1882.5	8913263	0.0252	2.123	4347608	1.18e+009	164.2
1085	4.66e+008	1884.3	8915148	0.0254	2.148	4340644	1.18e+009	164.4
1075	4.64e+008	1891.2	8917039	0.0257	2.174	4333591	1.18e+009	164.6
1065	4.62e+008	1892.1	8918931	0.0260	2.200	4326469	1.19e+009	164.9
1055	4.60e+008	1893.8	8920825	0.0263	2.226	4319274	1.19e+009	165.1
1045	4.58e+008	1904.5	8922729	0.0267	2.253	4311970	1.19e+009	165.3
1035	4.56e+008	1902.0	8924631	0.0269	2.280	4304606	1.19e+009	165.5
1025	4.54e+008	1915.9	8926547	0.0273	2.307	4297117	1.19e+009	165.8
1015	4.51e+008	1913.2	8928460	0.0276	2.335	4289566	1.20e+009	166.0
1005	4.49e+008	1920.4	8930381	0.0279	2.363	4281912	1.20e+009	166.2
995	4.47e+008	1921.6	8932302	0.0282	2.391	4274178	1.20e+009	166.4
985	4.45e+008	1931.8	8934234	0.0287	2.420	4266324	1.20e+009	166.7
975	4.43e+008	1935.5	8936169	0.0290	2.449	4258376	1.21e+009	166.9
965	4.40e+008	1942.2	8938112	0.0294	2.478	4250319	1.21e+009	167.1
955	4.38e+008	1946.6	8940058	0.0298	2.508	4242161	1.21e+009	167.4
945	4.36e+008	1951.7	8942010	0.0302	2.538	4233896	1.21e+009	167.6
935	4.34e+008	1959.8	8943970	0.0306	2.569	4225509	1.21e+009	167.8
925	4.32e+008	1957.6	8945927	0.0309	2.600	4217043	1.22e+009	168.1
915	4.29e+008	1972.5	8947900	0.0315	2.631	4208420	1.22e+009	168.3
905	4.27e+008	1978.8	8949879	0.0319	2.663	4199676	1.22e+009	168.5
895	4.25e+008	1985.1	8951864	0.0324	2.695	4190807	1.22e+009	168.8
885	4.23e+008	1991.3	8953855	0.0328	2.728	4181812	1.23e+009	169.0
875	4.20e+008	1991.7	8955847	0.0332	2.761	4172714	1.23e+009	169.2
865	4.18e+008	2003.5	8957850	0.0338	2.795	4163458	1.23e+009	169.5
855	4.16e+008	2010.5	8959861	0.0343	2.829	4154063	1.23e+009	169.7
845	4.14e+008	2018.1	8961879	0.0348	2.864	4144523	1.24e+009	170.0
835	4.11e+008	2029.6	8963908	0.0354	2.900	4134815	1.24e+009	170.2
825	4.09e+008	2030.8	8965939	0.0359	2.935	4124986	1.24e+009	170.4
815	4.07e+008	2048.4	8967988	0.0366	2.972	4114952	1.24e+009	170.7
805	4.04e+008	2044.1	8970032	0.0370	3.009	4104817	1.25e+009	170.9
795	4.02e+008	2064.3	8972096	0.0378	3.047	4094456	1.25e+009	171.2
785	4.00e+008	2065.8	8974162	0.0383	3.085	4083957	1.25e+009	171.4
775	3.98e+008	2080.1	8976242	0.0391	3.124	4073252	1.25e+009	171.6
765	3.95e+008	2081.6	8978324	0.0396	3.164	4062402	1.26e+009	171.9
755	3.93e+008	2100.0	8980424	0.0405	3.204	4051314	1.26e+009	172.1
745	3.90e+008	2108.0	8982532	0.0412	3.246	4040037	1.26e+009	172.4
735	3.88e+008	2112.9	8984645	0.0418	3.287	4028582	1.26e+009	172.6
725	3.86e+008	2124.3	8986769	0.0426	3.330	4016911	1.27e+009	172.9
715	3.83e+008	2143.1	8988912	0.0436	3.373	4004975	1.27e+009	173.1
705	3.81e+008	2144.9	8991057	0.0442	3.418	3992863	1.27e+009	173.4
695	3.78e+008	2162.5	8993219	0.0452	3.463	3980479	1.27e+009	173.7
685	3.76e+008	2174.3	8995394	0.0461	3.509	3967850	1.28e+009	173.9
675	3.74e+008	2180.6	8997574	0.0469	3.556	3955000	1.28e+009	174.2
665	3.71e+008	2198.6	8999773	0.0480	3.604	3941854	1.28e+009	174.4
655	3.69e+008	2209.2	9001982	0.0489	3.653	3928448	1.28e+009	174.7
645	3.66e+008	2223.2	9004205	0.0500	3.703	3914751	1.29e+009	174.9

635	3.64e+008	2239.1	9006444	0.0511	3.754	3900745	1.29e+009	175.2
625	3.61e+008	2253.9	9008698	0.0523	3.806	3886426	1.29e+009	175.5
615	3.59e+008	2259.6	9010958	0.0532	3.859	3871842	1.30e+009	175.7
605	3.56e+008	2282.9	9013241	0.0546	3.914	3856870	1.30e+009	176.0
595	3.54e+008	2297.2	9015538	0.0559	3.970	3841557	1.30e+009	176.3
585	3.51e+008	2312.6	9017851	0.0572	4.027	3829395	1.30e+009	176.5
575	3.48e+008	2332.1	9020183	0.0587	4.086	3820936	1.31e+009	176.8
565	3.46e+008	2346.4	9022529	0.0600	4.146	3812278	1.31e+009	177.1
555	3.43e+008	2366.6	9024896	0.0616	4.207	3803392	1.31e+009	177.4
545	3.41e+008	2390.1	9027286	0.0633	4.271	3794258	1.31e+009	177.7
535	3.38e+008	2404.2	9029690	0.0649	4.336	3784903	1.32e+009	177.9
525	3.35e+008	2428.9	9032119	0.0668	4.402	3775276	1.32e+009	178.2
515	3.33e+008	2448.9	9034568	0.0686	4.471	3765387	1.32e+009	178.5
505	3.30e+008	2468.4	9037036	0.0704	4.541	3755227	1.33e+009	178.8
495	3.27e+008	2497.3	9039533	0.0727	4.614	3744747	1.33e+009	179.1
485	3.24e+008	2520.7	9042054	0.0748	4.689	3733957	1.33e+009	179.4
475	3.22e+008	2545.1	9044599	0.0771	4.766	3722840	1.34e+009	179.7
465	3.19e+008	2569.0	9047168	0.0794	4.845	3711384	1.34e+009	180.0
455	3.16e+008	2597.9	9049766	0.0820	4.927	3699553	1.34e+009	180.3
445	3.13e+008	2630.4	9052397	0.0849	5.012	3687314	1.35e+009	180.6
435	3.10e+008	2658.0	9055055	0.0877	5.100	3674671	1.35e+009	180.9
425	3.07e+008	2688.3	9057743	0.0907	5.191	3661593	1.35e+009	181.2
415	3.04e+008	2719.4	9060462	0.0939	5.284	3648057	1.35e+009	181.5
405	3.01e+008	2757.7	9063220	0.0975	5.382	3634002	1.36e+009	181.9
395	2.98e+008	2796.4	9066016	0.1012	5.483	3619402	1.36e+009	182.2
385	2.95e+008	2826.6	9068843	0.1049	5.588	3604275	1.36e+009	182.5
375	2.92e+008	2872.0	9071715	0.1093	5.697	3588511	1.37e+009	182.8
365	2.89e+008	2910.5	9074626	0.1137	5.811	3572115	1.37e+009	183.2
355	2.86e+008	2953.3	9077579	0.1185	5.929	3555028	1.38e+009	183.5
345	2.83e+008	3007.0	9080586	0.1240	6.053	3539537	1.38e+009	183.9
335	2.80e+008	3052.4	9083638	0.1295	6.183	3526654	1.38e+009	184.2
325	2.76e+008	3103.3	9086742	0.1355	6.318	3513171	1.39e+009	184.6
315	2.73e+008	3165.3	9089907	0.1424	6.461	3499002	1.39e+009	185.0
305	2.70e+008	3224.1	9093131	0.1496	6.610	3484117	1.39e+009	185.3
295	2.66e+008	3287.2	9096418	0.1574	6.768	3468452	1.40e+009	185.7
285	2.63e+008	3354.9	9099773	0.1660	6.934	3451930	1.40e+009	186.1
275	2.59e+008	3430.8	9103204	0.1756	7.110	3434452	1.41e+009	186.5
265	2.56e+008	3510.5	9106714	0.1862	7.296	3415927	1.41e+009	186.9
255	2.52e+008	3591.9	9110306	0.1975	7.493	3396271	1.41e+009	187.3
245	2.48e+008	3684.3	9113991	0.2104	7.704	3375332	1.42e+009	187.7
235	2.44e+008	3782.7	9117773	0.2247	7.928	3352973	1.42e+009	188.2
225	2.40e+008	3891.5	9121665	0.2408	8.169	3332670	1.43e+009	188.6
215	2.36e+008	4008.3	9125673	0.2588	8.428	3312514	1.43e+009	189.1
205	2.32e+008	4136.1	9129809	0.2792	8.707	3290768	1.44e+009	189.5
200	2.30e+008	2134.1	9131943	0.1474	8.855	3279287	1.44e+009	189.8
195	2.28e+008	2171.4	9134115	0.1536	9.008	3267327	1.44e+009	190.0
190	2.26e+008	2205.8	9136321	0.1598	9.168	3254880	1.45e+009	190.3
185	2.24e+008	2253.6	9138574	0.1674	9.335	3241845	1.45e+009	190.5
180	2.22e+008	2297.7	9140872	0.1750	9.510	3228214	1.45e+009	190.8
175	2.19e+008	2341.9	9143214	0.1831	9.694	3213955	1.45e+009	191.1
170	2.17e+008	2395.7	9145609	0.1924	9.886	3198972	1.46e+009	191.3
165	2.15e+008	2444.7	9148054	0.2018	10.088	3184953	1.46e+009	191.6
160	2.12e+008	2505.8	9150560	0.2127	10.300	3172496	1.46e+009	191.9
155	2.10e+008	2568.8	9153129	0.2245	10.525	3159350	1.47e+009	192.2
150	2.08e+008	2633.2	9155762	0.2371	10.762	3145466	1.47e+009	192.5
145	2.05e+008	2704.5	9158466	0.2512	11.013	3130759	1.47e+009	192.8
140	2.03e+008	2784.6	9161251	0.2670	11.280	3115127	1.48e+009	193.1
135	2.00e+008	2863.4	9164114	0.2837	11.564	3098516	1.48e+009	193.4
130	1.98e+008	2953.5	9167068	0.3027	11.867	3080790	1.48e+009	193.8
125	1.95e+008	3048.4	9170116	0.3236	12.190	3061840	1.49e+009	194.1
120	1.92e+008	3155.5	9173272	0.3475	12.538	3041496	1.49e+009	194.5
115	1.89e+008	3274.0	9176546	0.3744	12.912	3019574	1.49e+009	194.8
110	1.86e+008	3393.6	9179939	0.4036	13.316	2995940	1.50e+009	195.2
105	1.83e+008	3540.2	9183479	0.4387	13.754	2970256	1.50e+009	195.6
100	1.80e+008	3693.7	9187173	0.4776	14.232	2942289	1.51e+009	196.0
95	1.77e+008	3862.4	9191036	0.5222	14.754	2911712	1.51e+009	196.4
90	1.74e+008	4052.4	9195088	0.5741	15.328	2883403	1.52e+009	196.9
85	1.71e+008	4276.6	9199365	0.6362	15.965	2856427	1.52e+009	197.4

80	1.67e+008	4533.1	9203898	0.7100	16.675	2826324	1.53e+009	197.9
75	1.64e+008	4835.3	9208733	0.7995	17.474	2792425	1.53e+009	198.4
70	1.60e+008	5172.7	9213906	0.9058	18.380	2754019	1.54e+009	199.0
65	1.56e+008	5585.0	9219491	1.0393	19.419	2709951	1.55e+009	199.6
60	1.52e+008	6064.3	9225555	1.2041	20.623	2664651	1.55e+009	200.3
55	1.48e+008	6659.2	9232214	1.4171	22.040	2617647	1.56e+009	201.0
50	1.43e+008	7420.5	9239635	1.7011	23.741	2561222	1.57e+009	201.8

Boron related parameters, intermediate results:

fb10_nat_wt = 0.1831
fb11_slc = 0.6530
fb10_slc_wt = 0.3257
delta_tk_lvl = 0.8230 %/in.
SLC tank surface area: 63.194 ft2
vf_slc = 0.01609 ft3/lbm
xb_cld = 371.0275 ppm
delta_tk_cld = 55.40 in.
xb_hot = 293.4490 ppm
delta_tk_hot = 26.04 in.
wb = 7.609 lbm/min.
t_hsbw = 0.342 hr
hf_cst = 95.49 Btu/lbm
hg_srv = 1186.6 Btu/lbm
tsp_hctl_lim1 = 168.6°F
mf_sp_lco = 6584924 lbm
z1 = 4.324

PSP calculation, intermediate results:

wlsp_lim = 9.58 ft
pg_dw = 2.537 psia
pg_sc = 1.275 psia
pndw_dw = 12.58 psia
pnsc_sc = 12.70 psia
vf_flood = 0.0160 ft3/lbm
dpsp_dwlsp = 0.4335 psi/ft

wlsp [ft]	vsc_air [ft3]	pnsc_dw [psia]	dpsc_dtsc [psia/°R]	tsp_hctl_1 [°F]	tsp_hctl_2 [°F]	pg_hp [psia]	pg_lp [psia]	psc_hp_1 [psig]
9.58	154763	14.33	0.0483	136.2	220.0	2.62	17.19	16.70
10.00	150890	14.70	0.0490	140.9	220.0	2.96	17.19	17.66
11.00	141436	15.68	0.0507	150.3	220.0	3.75	17.19	19.99
12.00	131960	16.81	0.0527	157.9	220.0	4.51	17.19	22.37
13.00	122200	18.15	0.0551	164.1	220.0	5.22	17.19	24.90
13.88	113603	19.53	0.0576	168.6	220.0	5.80	17.19	27.27
14.00	112428	19.73	0.0579	169.1	220.0	5.87	17.19	27.61
15.00	102119	21.72	0.0615	173.3	220.0	6.46	17.19	30.69
16.00	92820	23.90	0.0654	176.7	220.0	6.98	17.19	33.89
17.00	83463	26.58	0.0702	173.7	214.2	6.52	15.35	36.26
17.85	75334	29.45	0.0753	163.1	201.8	5.10	11.96	37.30

wlsp [ft]	z1	ppc_lim_a [psig]	psc_hp_2 [psig]	psc_srv [psig]	psc_lim [psig]	hf_tap [ft]
9.58	0.88	62.00	69.50	45.05	16.70	0.00
10.00	0.88	62.00	70.25	44.86	17.66	0.00
11.00	0.90	62.00	71.87	44.43	19.99	0.00
12.00	0.91	62.00	73.29	44.00	22.37	0.00
13.00	0.92	62.00	74.53	43.56	24.90	0.00
13.88	0.92	62.00	75.51	43.18	27.27	0.00
14.00	0.93	62.00	75.64	43.13	27.61	0.00
15.00	0.93	62.00	76.58	42.70	30.69	0.00
16.00	0.94	62.00	77.40	42.26	33.89	0.00
17.00	0.94	62.00	78.88	41.83	36.26	0.00
17.85	0.94	62.00	80.75	41.46	37.30	0.00

Drywell spray calculation intermediate results:

pndw_dw = 16.65 psia
pg_sc = 0.34 psia
pndw_sc = 13.83 psia
dtdw_dpdw = 18.4 °R/psia

pressure at intersection = 28.02 psig
temperature at intersection = 324.8°F

Tdw_scram values:

pdw [psig]	tdw_scram [°F]
2.7	34.5
3.0	43.0
4.0	113.1
5.0	173.4
6.0	230.6
7.0	287.7
8.0	345.4
9.0	404.6
10.0	465.4
11.0	528.1
11.2	540.9

Tdw_cap values:

pdw [psig]	tdw_cap [°F]
12.0	520.4
13.0	497.4
14.0	477.0
15.0	458.7
16.0	442.3
17.0	427.4
18.0	413.8
19.0	401.5
20.0	390.1
21.0	379.7
22.0	370.1
23.0	361.1
24.0	352.9
25.0	345.2
26.0	338.0
27.0	331.2
28.0	324.9
29.0	319.0
30.0	313.4
31.0	308.1
32.0	303.2
33.0	298.5
34.0	294.0
35.0	289.8
36.0	285.8
37.0	281.9
38.0	278.3
39.0	274.8
40.0	271.5
41.0	268.4
42.0	265.3
43.0	262.4
44.0	259.7
45.0	257.0
46.0	254.4
47.0	252.0
48.0	249.6
49.0	247.3
50.0	245.1
51.0	243.0
52.0	241.0

53.0	239.0
54.0	237.1
55.0	235.3
56.0	233.5
57.0	231.8
58.0	230.2
59.0	228.6
60.0	227.0

Tdw_min values:

pdw [psig]	tdw_min [°F]
0.0	-189.8
1.0	-171.4
2.0	-153.0
3.0	-134.7
4.0	-116.3
5.0	-97.9
6.0	-79.6
7.0	-61.2
8.0	-42.8
9.0	-24.5
10.0	-6.1
11.0	12.2
12.0	30.6
13.0	49.0
14.0	67.3
15.0	85.7
16.0	104.1
17.0	122.4
18.0	140.8
19.0	159.1
20.0	177.5
21.0	195.9
22.0	214.2
23.0	232.6
24.0	251.0
25.0	269.3
26.0	287.7
27.0	306.0
28.0	324.4
29.0	342.8
30.0	361.1
31.0	379.5
32.0	397.9
33.0	416.2
34.0	434.6
35.0	453.0
36.0	471.3
37.0	489.7
38.0	508.0
39.0	526.4
40.0	544.8
41.0	563.1
42.0	581.5
43.0	599.9
44.0	618.2
45.0	636.6
46.0	654.9
47.0	673.3
48.0	691.7
49.0	710.0
50.0	728.4
51.0	746.8
52.0	765.1
53.0	783.5
54.0	801.9

55.0	820.2
56.0	838.6
57.0	856.9
58.0	875.3
59.0	893.7

DWISL values:

pdw [psig]	dwisl [°F]
2.7	34.5
3.0	43.0
4.0	113.1
5.0	173.4
6.0	230.6
7.0	287.7
8.0	345.4
9.0	404.6
10.0	465.4
11.0	528.1
11.2	540.9
12.0	520.4
13.0	497.4
14.0	477.0
15.0	458.7
16.0	442.3
17.0	427.4
18.0	413.8
19.0	401.5
20.0	390.1
21.0	379.7
22.0	370.1
23.0	361.1
24.0	352.9
25.0	345.2
26.0	338.0
27.0	331.2
28.0	324.9
28.0	324.8
85.0	324.8

MDRIR calculation intermediate results:

vf_inj = 0.01675 ft3/lbm

SCSIP calculation intermediate results:

pg_sc = 1.275 psia
pg_dw = 2.537 psia
pnsc_sc = 12.698 psia
pnsc_dw = 19.528 psia

EOP calculation final results:

Plant-specific input values:

nbr_fuel_types = 2
nbr_pc_members = 5
reactor product line: BWR/4
containment type: Mark I
c_clad = 0.0761 Btu/lbm-°F
c_fuel = 0.0683 Btu/lbm-°F
dpdw_ww = 8.5 psid
dpsrv = 28 psid
number of drywell spray nozzles, header A: 263
flow required for developed spray, header A: 20 gpm
number of drywell spray nozzles, header B: 221
flow required for developed spray, header B: 20 gpm

fuel type	fafl_15 [%]	fafl_18 [%]
1	83.33	75
2	83.33	75

fb10_slc = 0.347
hdco = 9.58 ft
hring = 17.85 ft
hsc_tap = 20.98 ft
hsp_ref = 0 ft
hsrv = 65.91 ft
hvent_pc = 104.25 ft

fuel type	fuel length [in.]
1	146
2	150

mclad = 109124 lbm
mf_rpv_cld = 895000 lbm
mf_rpv_hot = 532000 lbm
mfuel = 249314 lbm
mg_rpv_hot = 15300 lbm
mrpv = 2.02e+006 lbm
mrpv_taf = 959000 lbm
nbuns = 560
nsrv_ads = 7

structure	elevation [ft]	material type	strength basis	pressure [psig]	temperature [°F]
torus	29.5	4	yield	108	300
torus	0	4	yield	111	300
drywell	85	4	yield	118	300
drywell	50	4	yield	129	300
torus	30.3	1	yield	120	350

pdw_maxop = 1.95 psig
pdw_minop = 1.2 psig
pdw_scram = 2.7 psig
ppc_maxind = 85 psig
ppc_vent = 62 psig
pq_code = 37.44 kpsi
pq_des = 27.66 kpsi
pqs_code = 36 kpsi
pqs_des = 11.287 kpsi
prpv_maxind = 1500 psig
prpv_tp = 1250 psig
psc_maxop = 0.5 psig
psc_minop = -0.5 psig
psp_des = 56 psig
psp_srv = 12.87 psig

psrv_lift = 1145 psig
psrv_name = 1080 psig
psup_srv = 110 psig
ptp_code = 42 kpsi
ptp_des = 22.5 kpsi
ptps_code = 1.5 kpsi
ptps_des = 1 kpsi
qrx_rated = 2536 MWt
slc_tk_dia = 8.97 ft
slc_tk_lvl_in_span = 121.5 in.
slc_tk_lvl_out_span = 100 %
srv type: Target Rock, 2-Stage

fuel type: 1

t_sd [min.]	t_cu_15 [min.]
10	2.35
30	3.15
50	3.72
80	4.33
100	4.65
500	6.85
1000	8.27
3000	11.57
6000	14.87

fuel type: 2

t_sd [min.]	t_cu_15 [min.]
10	2.35
30	3.15
50	3.72
80	4.33
100	4.65
500	6.85
1000	8.27
3000	11.57
6000	14.87

tcst = 125°F
tdw_maxop = 135°F
tdw_maxind = 440°F
tdw_minop = 100°F
tsc_max = 220°F
tsc_maxop = 110°F
tslc = 104°F
tsp_flood = 42°F
tsp_minop = 68°F
tsp_scram = 110°F
vdw = 176397 ft3
vrpv_taf = 10300 ft3
vsc_dco = 154763 ft3
vsc_lco = 113603 ft3

fuel type	wg_1500 [lbm/hr]
1	1260
2	1260

wlsp [ft]	vsp [ft3]	vsc_air [ft3]
9.58	65339	154763
10	69214	150890
11	78589	141436
12	88145	131960
13	97840	122200
13.88	106442	113603
14	107619	112428

15	117174	102119
16	126432	92820
17	135793	83463
17.85	143927	75334

spwl [ft]	wll [ft]
13.88	0
14.66	0.86
15.5	8.12
18.51	11.21
29.5	42.79

LPCI flow vs. RPV pressure:

flow [gpm]	rpv_press [psig]
9628	0
9000	26
8000	65
7000	99
6000	126
5000	149
4000	166
3000	182
2000	195
0	208

CS flow vs. RPV pressure:

flow [gpm]	rpv_press [psig]
6781	0
6000	59
5500	93
5000	123
4500	151
4000	176
3000	219
2000	256
1000	279
0	295

wlrpv_baf = -144 in.
wlsp_maxlco = 14 ft
wlsp_minlco = 13.88 ft
wlsp_srv = 14 ft
wlsp_tp = 14 ft
wslc = 50 gpm
wsrv_name = 829000 lbm/hr
xb_cld_nat = 660 ppm
xb_hot_nat = 522 ppm
xb_slc = 18320 ppm

Generic input values:

c_steel = 0.128 Btu/lbm-°F
dptp_dprpv = 0.1 %/psig
dptp_dwll = 5 %/ft
fb10_nat = 0.1978
fb11_nat = 0.8022
fqdh_10 = 0.022115
fqdh_100 = 0.01118
fqdh_2 = 0.030381
k1 = 3.412e+006 Btu/hr-MWt
k2 = 56868 Btu/min.-MWt
k3 = 7.48 gal/ft3
mb10 = 10.01
mb11 = 11.01

material type: 1

temp [°F]	fyield	ftens
70	1	1
200	0.925	0.89
400	0.81	0.72
600	0.76	0.715

material type: 2

temp [°F]	fyield	ftens
70	1	1
200	0.79	0.87
400	0.58	0.79
600	0.53	0.76

material type: 3

temp [°F]	fyield	ftens
70	1	1
200	0.92	0.95
400	0.95	1
600	0.88	1

material type: 4

temp [°F]	fyield	ftens
70	1	1
200	0.94	0.96
400	0.86	1
600	0.77	1

material type: 5

temp [°F]	fyield	ftens
70	1	1
200	0.95	0.95
400	0.86	1
600	0.76	1

time [min.]	power [%]
2	3.038
4	2.658
6	2.458
8	2.32
10	2.212
15	2.009
20	1.862
25	1.747
30	1.653
35	1.574
40	1.507
45	1.45
50	1.4
60	1.318
90	1.155
120	1.057
150	0.9901
200	0.9661
300	0.8673
450	0.7863
600	0.7255
900	0.6509
1200	0.6002
1500	0.5622
1800	0.5311

srv_open = 50 psid
tlim_dw = 545°F
tlim_sp = 350°F

Maximum Pressure Suppression Primary Containment Water Level:

mpspcwl = 17.85 ft

Primary Containment Pressure Limits:

wlpc [ft]	pcpl_a [psig]	pcpl_c [psig]	pcpl_a_eop [psig]	pcpl_c_eop [psig]
0	62.0	62.0	62.0	62.0
1	62.0	62.0	62.0	62.0
2	62.0	62.0	62.0	62.0
3	62.0	62.0	62.0	62.0
4	62.0	62.0	62.0	62.0
5	62.0	62.0	62.0	62.0
6	62.0	62.0	62.0	62.0
7	62.0	62.0	62.0	62.0
8	62.0	62.0	62.0	62.0
9	62.0	62.0	62.0	62.0
10	62.0	62.0	62.0	62.0
11	62.0	62.0	62.0	62.0
12	62.0	62.0	62.0	62.0
13	62.0	62.0	62.0	62.0
14	62.0	62.0	62.0	62.0
15	62.0	62.0	62.0	62.0
16	62.0	62.0	62.0	62.0
17	62.0	62.0	62.0	62.0
18	62.0	62.0	62.0	62.0
19	62.0	62.0	62.0	62.0
20	62.0	62.0	62.0	62.0
21	62.0	62.0	62.0	62.0
22	62.4	62.4	62.4	62.4
23	62.9	62.9	62.9	62.9
24	63.3	63.3	63.3	63.3
25	63.7	63.7	63.7	63.7
26	64.2	64.2	64.2	64.2
27	64.6	64.6	64.6	64.6
28	65.0	65.0	65.0	65.0
29	65.5	65.5	65.5	65.5
30	65.9	65.9	65.9	65.9
31	66.3	66.3	66.3	66.3
32	66.8	66.8	66.8	66.8
33	67.2	67.2	67.2	67.2
34	67.6	67.6	67.6	67.6
35	68.1	68.1	68.1	68.1
36	68.5	68.5	68.5	68.5
37	68.9	68.9	68.9	68.9
38	69.4	69.4	69.4	69.4
39	69.8	69.8	69.8	69.8
40	70.2	70.2	70.2	70.2
41	70.7	70.7	70.7	70.7
42	71.1	71.1	71.1	71.1
43	71.5	71.5	71.5	71.5
44	72.0	72.0	72.0	72.0
45	72.4	72.4	72.4	72.4
46	72.8	72.8	72.8	72.8
47	73.3	73.3	73.3	73.3
48	73.7	73.7	73.7	73.7
49	74.1	74.1	74.1	74.1
50	74.6	74.6	74.6	74.6
51	75.0	75.0	75.0	75.0
52	75.4	75.4	75.4	75.4
53	75.9	75.9	75.9	75.9
54	76.3	76.3	76.3	76.3
55	76.7	76.7	76.7	76.7

56	77.2	77.2	77.2	77.2
57	77.6	77.6	77.6	77.6
58	78.0	78.0	78.0	78.0
59	78.5	78.5	78.5	78.5
60	78.9	78.9	78.9	78.9
61	79.4	79.4	79.4	79.4
62	79.8	79.8	79.8	79.8
63	80.2	80.2	80.2	80.2
64	80.7	80.7	80.7	80.7
65	81.1	81.1	81.1	81.1
66	81.5	81.5	81.5	81.5
67	82.0	82.0	82.0	82.0
68	82.4	82.4	82.4	82.4
69	82.8	82.8	82.8	82.8
70	83.3	83.3	83.3	83.3
71	83.7	83.7	83.7	83.7
72	84.1	84.1	84.1	84.1
73	84.6	84.6	84.6	84.6
74	85.0	85.0	85.0	85.0
75	85.4	85.4	85.0	85.0
76	85.9	85.9	85.0	85.0
77	86.3	86.3	85.0	85.0
78	86.7	86.7	85.0	85.0
79	87.2	87.2	85.0	85.0
80	87.6	87.6	85.0	85.0
81	88.0	88.0	85.0	85.0
82	88.5	88.5	85.0	85.0
83	88.9	88.9	85.0	85.0
84	89.3	89.3	85.0	85.0
85	89.8	89.8	85.0	85.0
86	90.2	90.2	85.0	85.0
87	90.6	90.6	85.0	85.0
88	91.1	91.1	85.0	85.0
89	91.5	91.5	85.0	85.0
90	91.9	91.9	85.0	85.0
91	92.4	92.4	85.0	85.0
92	92.8	92.8	85.0	85.0
93	93.2	93.2	85.0	85.0
94	93.7	93.7	85.0	85.0
95	94.1	94.1	85.0	85.0
96	94.5	94.5	85.0	85.0
97	95.0	95.0	85.0	85.0
98	95.4	95.4	85.0	85.0
99	95.8	95.8	85.0	85.0
100	96.3	96.3	85.0	85.0
101	96.7	96.7	85.0	85.0
102	97.1	97.1	85.0	85.0
103	97.6	97.6	85.0	85.0
104	98.0	98.0	85.0	85.0
105	0.0	0.0	0.0	0.0
106	0.0	0.0	0.0	0.0
107	0.0	0.0	0.0	0.0
108	0.0	0.0	0.0	0.0
109	0.0	0.0	0.0	0.0
110	0.0	0.0	0.0	0.0
111	0.0	0.0	0.0	0.0
112	0.0	0.0	0.0	0.0
113	0.0	0.0	0.0	0.0
114	0.0	0.0	0.0	0.0
115	0.0	0.0	0.0	0.0
116	0.0	0.0	0.0	0.0
117	0.0	0.0	0.0	0.0
118	0.0	0.0	0.0	0.0
119	0.0	0.0	0.0	0.0

SRV Tail Pipe Level Limit (for EOPs):

spwl	pressure

[ft]	[psig]
15.00	1500
15.50	1286
17.85	1165
17.85	0
0.00	0

RPV variables, final results:

Minimum Alternate RPV Flooding Pressure:

nsrv	marfp [psig]
1	857.4
2	421.4
3	276.0
4	203.3
5	159.7
6	130.7
7	109.9

MNSRED = 5

MRFP = 50 psig

Minimum Core Flooding Interval:

nsrv	z2	mcfi [min.]
5	68398	42.9
6	120745	28.8
7	173091	21.2

Maximum Core Uncovery Time Limit:

t_sd [min.]	t_cu_15 [min.]
21.2	2.80
30.0	3.15
50.0	3.72
80.0	4.33
100.0	4.65
500.0	6.85
1000.0	8.27
3000.0	11.57
6000.0	14.87

MSCRWL = -19.005 in.

MZIRWL = -31.5 in.

Heat Capacity Temperature Limit:

prpv [psig]	wlsp [ft] 9.58 tsp [°F]	wlsp [ft] 10.00 tsp [°F]	wlsp [ft] 11.00 tsp [°F]	wlsp [ft] 12.00 tsp [°F]	wlsp [ft] 13.00 tsp [°F]	wlsp [ft] 13.88 tsp [°F]	wlsp [ft] 14.00 tsp [°F]	wlsp [ft] 15.00 tsp [°F]	wlsp [ft] 16.00 tsp [°F]	wlsp [ft] 17.00 tsp [°F]	wlsp [ft] 17.85 tsp [°F]
1145	136.2	140.9	150.3	157.9	164.1	168.6	169.1	173.3	176.7	173.7	163.1
1095	138.7	143.3	152.4	159.7	165.7	170.1	170.6	174.6	178.0	174.8	164.2
1045	141.3	145.6	154.5	161.6	167.3	171.6	172.1	176.0	179.2	176.0	165.3
995	143.8	148.0	156.6	163.4	169.0	173.1	173.6	177.4	180.5	177.2	166.4
945	146.4	150.5	158.7	165.3	170.7	174.6	175.1	178.8	181.8	178.4	167.6
895	149.0	152.9	160.8	167.2	172.4	176.2	176.7	180.2	183.1	179.6	168.8
845	151.6	155.4	163.0	169.1	174.1	177.8	178.3	181.6	184.4	180.9	170.0
795	154.3	157.9	165.2	171.1	175.9	179.4	179.8	183.1	185.8	182.1	171.2
745	157.0	160.5	167.5	173.1	177.7	181.1	181.5	184.6	187.2	183.4	172.4
695	159.8	163.1	169.8	175.1	179.5	182.7	183.1	186.1	188.6	184.7	173.7

645	162.7	165.8	172.1	177.2	181.4	184.5	184.9	187.7	190.0	186.1	174.9
595	165.6	168.6	174.6	179.4	183.3	186.2	186.6	189.3	191.5	187.5	176.3
545	168.6	171.4	177.1	181.6	185.3	188.1	188.4	191.0	193.1	188.9	177.7
495	171.7	174.4	179.7	183.9	187.4	190.0	190.3	192.7	194.7	190.4	179.1
445	175.0	177.4	182.4	186.3	189.6	192.0	192.3	194.5	196.3	192.0	180.6
395	178.4	180.7	185.2	188.9	191.9	194.1	194.4	196.4	198.1	193.7	182.2
345	182.1	184.1	188.3	191.6	194.3	196.4	196.6	198.5	200.0	195.5	183.9
295	186.0	187.9	191.5	194.5	197.0	198.8	199.0	200.7	202.1	197.4	185.7
245	190.4	192.0	195.2	197.8	199.9	201.5	201.7	203.1	204.3	199.5	187.7
200	194.7	196.1	198.8	201.0	202.8	204.2	204.3	205.6	206.6	201.6	189.8
175	197.5	198.7	201.1	203.1	204.7	205.9	206.0	207.1	208.0	203.0	191.1
150	200.5	201.5	203.6	205.3	206.7	207.7	207.9	208.8	209.6	204.5	192.5
125	203.9	204.8	206.5	207.9	209.0	209.9	210.0	210.8	211.4	206.2	194.1
100	207.9	208.6	209.8	210.9	211.8	212.4	212.5	213.1	213.6	208.1	196.0
75	212.9	213.3	214.1	214.7	215.2	215.5	215.6	215.9	216.2	210.7	198.4
50	220.0	220.0	220.0	220.0	220.0	220.0	220.0	220.0	220.0	214.2	201.8

Boron related values, final results:

CSBW = 332.1 lbm
change in tank level to inject CSBW = 45.6 %
HSBW = 156.1 lbm
change in tank level to inject HSBW = 21.4 %

Boron Injection Initiation Temperature:

power [%]	biit [°F]
0.00	159.0
2.21	159.0
13.55	110.0
20.00	110.0

Pressure Suppression Pressure:

wlsp [ft]	psp [psig]
1.00	0.0
2.00	0.0
3.00	0.0
4.00	0.0
5.00	0.0
6.00	0.0
7.00	0.0
8.00	0.0
9.00	0.0
9.58	16.7
10.00	17.7
11.00	20.0
12.00	22.4
13.00	24.9
13.88	27.3
14.00	27.6
15.00	30.7
16.00	33.9
17.00	36.3
17.85	37.3
18.00	0.0

Drywell spray related variable, final results:

Drywell Spray Initiation Limit:

pdw [psig]	dwsil [°F]
2.7	34.5
3.0	43.0
4.0	113.1

5.0	173.4
6.0	230.6
7.0	287.7
8.0	345.4
9.0	404.6
9.6	440.0
16.2	440.0
17.0	427.4
18.0	413.8
19.0	401.5
20.0	390.1
21.0	379.7
22.0	370.1
23.0	361.1
24.0	352.9
25.0	345.2
26.0	338.0
27.0	331.2
28.0	324.9
28.0	324.8
85.0	324.8

MDSF = 5260 gpm

Minimum Debris Retention Injection Rate:

time [min.]	mdrir [gpm] rpv pressure 0 psig	500 psig	1000 psig
2	565.6	730.2	847.5
4	494.9	638.9	741.5
6	457.6	590.8	685.7
8	431.9	557.6	647.2
10	411.8	531.7	617.1
15	374.0	482.9	560.4
20	346.7	447.5	519.4
25	325.3	419.9	487.4
30	307.8	397.3	461.1
35	293.0	378.3	439.1
40	280.6	362.2	420.4
45	270.0	348.5	404.5
50	260.7	336.5	390.6
60	245.4	316.8	367.7
90	215.0	277.6	322.2
120	196.8	254.1	294.9
150	184.3	238.0	276.2
200	179.9	232.2	269.5
300	161.5	208.5	241.9
450	146.4	189.0	219.4
600	135.1	174.4	202.4
900	121.2	156.4	181.6
1200	111.7	144.3	167.4
1500	104.7	135.1	156.8
1800	98.9	127.7	148.2

Suppression Chamber Spray Initiation Pressure:

scsip = 15.10 psig

NEW YORK POWER AUTHORITY
JAMES A. FITZPATRICK NUCLEAR POWER PLANT
OPERATING PROCEDURE

RESIDUAL HEAT REMOVAL SYSTEM*
OP-13
REVISION 87

APPROVED BY: _____ DATE 9/2/99
RESPONSIBLE PROCEDURE OWNER

EFFECTIVE DATE: 9-3-99

FIRST ISSUE ☐ FULL REVISION ☐ LIMITED REVISION ☒

REFERENCE USE	TSR
TECHNICAL	

PERIODIC REVIEW DUE DATE MAR 2003

FOR INFORMATION ONLY

REVISION SUMMARY SHEET

REV. NO.	CHANGE AND REASON FOR CHANGE
87	Changed position of the following RHR and RHRSW pump motor bearing oil drain and sample valves from "locked closed and capped" to "closed and capped" on valve lineup. This change reflects JAF-SE-89-042, Rev. 1: • 10RHR-723A • 10RHR-723B • 10RHR-724A • 10RHR-724B • 10RHR-723C • 10RHR-723D • 10RHR-724C • 10RHR-724D • 10RHR-778A • 10RHR-778B • 10RHR-779A • 10RHR-779B • 10RHR-778C • 10RHR-778D • 10RHR-779C • 10RHR-779D
86	Revised procedure to reflect Design Change JD-99-099 and Safety Evaluation JAF-SE-99-014. These documents implement administrative controls that prohibit use of the steam condensing mode unless directed by the EOPs. These documents also allow re-installation of the motor heaters for the subject valves. Changes affect Commitment A.2.2, and Attachments 1, 3A, and 3B. (PCR #17 dated 7/21/99)
85	Added new Subsections G.3 and G.4 for continuous venting of RHR headers via sample sink line. Pressurization due to leakage past 10RHR-68A/B requires method to prevent lifting of header relief valves. Also changed normal position of 10RHR-780B to locked throttled on valve line-up to limit maximum flow to less than 0.5 gpm. (pcr dated 5/21/99)
84	Revised procedure to reflect installation of Temp Mod 98-120 and JAF-SE-98-052. Overload heaters are removed from 10MOV-21A and 21B. Affects Attachment 1, 3A and 3B. Incorporated TC dated 11/6/98 which added valves 10RHR-440 and 441 per Modification M1-97-021.

TABLE OF CONTENTS

SECTION		PAGE
A.	**REQUIREMENTS**	4
B.	**SYSTEM DESCRIPTION**	5
C.	**PLANT OPERATING REQUIREMENTS**	10
D.	**STARTUP**	13
E.	**NORMAL OPERATION**	14
F.	**SHUTDOWN**	15
G.	**SPECIAL PROCEDURES**	16
H.	**ATTACHMENTS**	22
	1. REFERENCES	23
	2A. TABLE 1 - VALVE LINEUP - RHR LOOP A	26
	2B. TABLE 1 - VALVE LINEUP - RHR LOOP B	45
	2C. TABLE 1 - VALVE LINEUP - COMMON TO RHR LOOPS A AND B	66
	3A. TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP A	71
	3B. TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP B	75
	3C. TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - COMMON TO RHR LOOP A AND B	79

A. REQUIREMENTS

A.1 Technical Specifications

Sections 3.5.A, 3.5.B, 3.5.F, and 3.5.G

A.2 Commitments

A.2.1 JAFP-94-0228, Response to NRC Bulletin No. 93-02, Supplement 1, Debris Plugging of Emergency Core Cooling Suction Strainers. Added precaution to alert operators of the potential for ECCS suction strainer clogging and to adjust flow consistent with required needs to mitigate the clogging.

A.2.2 JD-99-099, JAF-SE-99-014, Deselection of 10MOV-21A&B from the JAF GL 89-10 Program. Deselecting valves from GL 89-10 requires that 10MOV-21A & B can only be open when directed by EOPs or when Primary Containment is not required. Added precaution, note, and commitments to alert operators.

A.3 Validation

Revision 74 validated per ODSO-35.

B. **SYSTEM DESCRIPTION**

Low Pressure Coolant Injection (LPCI) Mode

LPCI is an integral part of the Residual Heat Removal (RHR) System. LPCI restores and maintains reactor pressure vessel (RPV) coolant inventory after a loss of coolant accident (LOCA) to prevent fuel clad temperatures from exceeding 2200°F. Maintaining fuel clad temperatures less than 2200°F limits the energy release caused by a metal-water reaction. To achieve this goal, LPCI operates in conjunction with the following systems:

- High Pressure Coolant Injection System (HPCI)
- Automatic Depressurization System (ADS)
- Core Spray System

LPCI auto-initiates upon low-low-low RPV water level (59.5 inches) or high drywell pressure (2.7 psig).

During LPCI operation, RHR pumps take suction from the torus and discharge through the associated reactor water recirculation loop, and into the RPV. Coolant leaking from a loss of coolant break is contained by the drywell and returned to the torus through suppression vent lines and downcomers.

RHR pump minimum flow protection is provided by a minimum flow valve in a minimum flow line from the RHR pump discharge header to the torus.

RHR pump motors for 10P-3A and 10P-3B are powered from emergency AC Bus 10500. RHR pump motors for 10P-3C and 10P-3D are powered from emergency AC Bus 10600. Logic and control power for each LPCI loop is from a redundant vital DC bus, with the exception of logic and control power for RHR Pumps 10P-3A and 10P-3B, which is from 125 VDC System A, and logic and control power for RHR Pumps 10P-3C and 10P-3D, which is from 125 VDC System B.

RHR service water flow through RHR heat exchangers is not required immediately after a LOCA because heat rejection from the primary containment is unnecessary during the time it takes to flood the reactor core.

When the normal source of water from the torus is unavailable, RHR service water (RHRSW) can be pumped through a crosstie line into the RPV through LPCI injection lines.

When torus water and RHR service water are unavailable, water from Fire Protection System pumps can be pumped through a hose into the RHRSW Loop A header, and into the RPV through LPCI A injection line.

Containment Spray Mode

The containment spray mode is an integral part of the RHR System. Containment sprays reduce drywell and torus pressure and mix primary containment atmosphere following a LOCA. Containment spray is manually initiated when directed by the Emergency Operating Procedures (EOPs). Interlocks prevent initiating containment spray before LPCI requirements are satisfied.

During containment spray operation, RHR pumps take suction from the torus and discharge through RHR heat exchangers, where heat is transferred to RHR service water. The cooled water is then diverted to one or both of the following spray headers:

- Drywell spray header
- Torus spray header

The spray in the drywell condenses steam to lower drywell pressure. The water collects on the bottom of the drywell until water level reaches the suppression vent lines, it then overflows and drains back to the torus.

The spray in the torus air space cools non-condensable gases.

Reactor Isolation (Steam Condensing Mode)

Steam Condensing Mode shall only be used when directed by Emergency Operating Procedures (EOPs). The reactor isolation (steam condensing) mode of RHR is a manually initiated mode of RHR used for controlling RPV pressure and level when the RPV is isolated with a subsequent loss of feedwater.

When the RPV is isolated after power operation, main steam is relieved through main steam safety relief valves to the torus where it is condensed and subcooled. There is a limit to the amount of heat that can be safely added to the torus. Torus water temperature is normally maintained less than 95°F. During testing specified in Technical Specifications, torus water temperature is allowed to reach 105°F. Torus water temperature greater than 95°F is an entry condition for EOP-4, Primary Containment Control.

When in the steam condensing mode, heat is transferred to RHR service water using an RHR heat exchanger as a steam condenser. The steam is condensed and the condensate is drained to the torus or pumped into the RPV using RCIC.

Reactor steam from the HPCI turbine steam supply line is reduced to operating pressure and then admitted to the shell side of an RHR heat exchanger. The steam is condensed and subcooled by RHR service water. The heat transfer rate across the tubes is determined by the condensate level in the shell side of the heat exchanger. Condensate level in the heat exchanger is maintained by a level controller that controls the flow of condensate to one of the following:

- Torus (by throttling a level control valve)
- RCIC pump suction (by controlling RCIC turbine speed)

Operation of the RCIC turbine in the steam condensing mode is expected to raise torus water temperature at a rate of 3°F per hour. One to two hours after reactor shutdown, one RHR heat exchanger will have sufficient heat transfer capacity to remove decay heat. The other RHR loop will be placed in the torus cooling mode to remove the heat input from RCIC turbine operation.

Shutdown Cooling

Shutdown cooling is a manually initiated mode of the RHR System that is placed in service during normal shutdown and cooldown.

The initial phase of RPV cooldown is accomplished by discharging main steam to the main condenser through main turbine bypass valves.

RPV cooldown is completed by aligning the RHR pump suction to the reactor water recirculation loop B suction line. Reactor coolant is pumped through an RHR heat exchanger and back to the RPV through a reactor water recirculation loop. Heat from the reactor coolant is transferred to RHR service water.

Shutdown cooling is capable of reducing reactor coolant temperature to 125°F to permit reactor refueling and servicing. Based on power uprate, the time required to reach cold shutdown is 21 hours.

Fuel Pool Cooling Assist Mode

Fuel pool cooling assist is a manually initiated mode of RHR. The RHR System is used to remove heat from the spent fuel pool and reactor cavity when the Fuel Pool Cooling System has failed or the heat load is beyond its heat removal capacity.

When in fuel pool cooling assist, the RHR pumps take suction from the fuel pool skimmer surge tanks through a suction crosstie, and discharge through the RHR heat exchanger and back to the spent fuel pool. Heat from the fuel pool water is transferred to RHR service water.

Fuel pool cooling assist can only be placed in operation when the reactor is shutdown, as it prevents using RHR for LPCI or containment cooling.

RHRSW and RHR Keep-Full Systems

RHRSW and RHR piping is maintained full by keep-full systems.

RHRSW piping is maintained full using service water as the keep-full water supply. The service water is piped into RHRSW pump discharge piping upstream of the associated RHRSW pump discharge strainer. Keep-full sonic level switches and associated annunciators and EPIC alarms are provided for the RHRSW Loop A and RHRSW Loop B piping upstream of RHR heat exchangers.

RHR piping is maintained full using one or both of the following keep-full systems:

- Condensate transfer: There are four condensate transfer fill connections for RHR piping. Two of these connections can be used to maintain the RHR System piping upstream of 10MOV-27A and 10MOV-27B full. These two connections are referred to as RHR Loop A and RHR Loop B condensate transfer keep-full.

- RHR Keep-Full Pumps 10P-2A and 10P-2B: Each RHR loop has a keep-full pump. RHR keep-full pumps take suction from the RHR pump torus suction piping and discharge to the RHR pump discharge headers, upstream of the associated 10MOV-66A or 10MOV-66B, and through a minimum flow line to the torus.

Keep-full sonic level switches and associated annunciators and EPIC alarms are provided for the following RHR System piping locations:

- LPCI A and B between 10MOV-27A(B) and 10MOV-25A(B)

- Containment Spray A and B upstream of 10MOV-26A(B)

- Head spray piping (head spray is retired, however the head spray keep-full is operational).

C. PLANT OPERATING REQUIREMENTS

C.1 Prerequisites

C.1.1 The following systems are lined up per the respective operating procedure for unrestricted operation of RHR:

- Condensate Transfer System per OP-5
- Reactor Core Isolation Cooling System per OP-19
- Emergency Service Water System per OP-21
- Recirculation System per OP-27
- Instrument Air per OP-39
- Reactor Building Closed Loop Cooling System per OP-40
- Service Water System per OP-42
- 125 VDC Power System per OP-43A
- LPCI Independent Power Supply per OP-43C
- 4160 V and 600 V Normal AC Power Distribution per OP-46A
- 120 VAC Power System per OP-46B
- Liquid Radioactive Waste System per OP-49
- Equipment and Floor Drainage System per OP-50

NOTE: Attachments 2A and 3A are for RHR Loop A. Attachments 2B and 3B are for RHR Loop B. Attachments 2C and 3C are for components that are common to RHR Loop A and B, such as shutdown cooling suction piping, fuel pool cooling assist piping, and RHR to RCIC piping.

C.1.2 System valves are lined up per Attachment 2A, B, C, with exceptions approved by Shift Manager.

C.1.3 System power supplies are lined up per Attachment 3A, B, C with exceptions approved by Shift Manager.

C.2 **Precautions**

↓COMA.2.1
C.2.1 Blockage of ECCS pump suction strainers could occur due to debris in the Torus. Within the latitude provided by EOPs to restore and maintain parameters within specified limits, potential mitigative actions may include:

- Minimizing ECCS flow or removing affected ECCS pumps from service
- Alternating ECCS pumps from one division to another, if available
- Shifting ECCS pump suction to another source, if available
- Operation of alternate injection sources

↓COMA.2.2
C.2.2 10MOV-21A and 10MOV-21B shall only be opened when directed by EOPs or when primary containment <u>is not</u> required.

C.2.3 RES Radiation Protection shall be notified before:

- RHR System piping is drained
- RHR System is started up in any mode of operation, unless an emergency exists

C.2.4 Racking out an RHRSW pump breaker de-energizes the pump motor bearing cooling water SOV and will drain the RHRSW header. To prevent draining the RHRSW header, the bearing cooling water isolation valve shall be closed before racking out pump breaker.

C.2.5 The RHR System <u>shall not</u> be filled by opening Torus Suction Valve 10MOV-13A, B, C, or D or Shutdown Cooling Suction Valve 10MOV-15A, B, C, or D.

C.2.6 Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

C.2.7 Running an RHR pump for greater than 10 minutes with only its minimum flow valve open could damage the pump.

C.2.8 An extremely large drain path exists from the RPV to the torus if either of the following conditions occur:

- 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the associated 10MOV-13A, B, C, or D are open at the same time.
- 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D are open and the associated RHR loop torus spray or torus cooling motor operated isolation valves are open.

The following valves are interlocked to prevent establishing this drain path:

- 10MOV-15A, B, C, and D
- 10MOV-13A, B, C, and D
- 10MOV-39A and B

These interlocks shall not be defeated unless system boundaries are isolated.

C.2.9 Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.

D. **STARTUP**

Refer to the applicable procedure listed below:

PROCEDURE NUMBER	PROCEDURE TITLE
OP-13A	RHR - Low Pressure Coolant Injection*
OP-13B	RHR - Containment Control*
OP-13C	RHR Service Water*
OP-13D	RHR - Shutdown Cooling*
OP-13E	RHR - Keep-Full*
OP-13F	RHR - Fuel Pool Cooling Assist*

NOTE: Entry into OP-13G will be directed by EOPs.

OP-13G	RHR - Steam Condensing*

E. **NORMAL OPERATION**

During normal plant operation, maintain RHR System in the following standby LPCI lineup:

NOTE: Attachments 2A and 3A are for RHR Loop A. Attachments 2B and 3B are for RHR Loop B. Attachments 2C and 3C are for components that are common to RHR Loop A and B, such as shutdown cooling suction piping, fuel pool cooling assist piping, and RHR to RCIC piping.

- System valves lined up per Attachment 2A, B, C.
- System component power supplies lined up per Attachment 3A, B, C.
- Control Room indications, controls, and switches lined up per ODSO-4.
- Switches in North Cable Room positioned as follows:
  - IS-1PCIA07 (control circuit isolation switch for 10MOV-18) . . . . . In REMOTE
  - DS-1PCIA07 (power disconnect switch for 10MOV-18) . . . . . Locked in OFF

F. **SHUTDOWN**

Refer to the applicable procedure listed below:

PROCEDURE NUMBER	PROCEDURE TITLE
OP-13A	RHR - Low Pressure Coolant Injection*
OP-13B	RHR - Containment Control*
OP-13C	RHR Service Water*
OP-13D	RHR - Shutdown Cooling*
OP-13E	RHR - Keep-Full*
OP-13F	RHR - Fuel Pool Cooling Assist*
OP-13G	RHR - Steam Condensing*

G. SPECIAL PROCEDURES

TABLE OF CONTENTS

SUBSECTION		PAGE
G.1	**Venting RHR Loop A to Lower Header Pressure**	17
G.2	**Venting RHR Loop B to Lower Header Pressure**	17
G.3	**Continuous Venting of RHR Header A**	18
G.4	**Continuous Venting of RHR Header B**	20

NOTE: For other special procedures, refer to the applicable procedure listed below:

PROCEDURE NUMBER	PROCEDURE TITLE
OP-13A	RHR - Low Pressure Coolant Injection*
OP-13B	RHR - Containment Control*
OP-13C	RHR Service Water*
OP-13D	RHR - Shutdown Cooling*
OP-13E	RHR - Keep-Full*
OP-13F	RHR - Fuel Pool Cooling Assist*
OP-13G	RHR - Steam Condensing*

NOTE: Subsection G.1 is used when header is pressurizing for reasons such as leakage by 10RHR-68A or 10MOV-70A.

G.1 **Venting RHR Loop A to Lower Header Pressure**

G.1.1 Verify RHR Loop A is in the standby LPCI lineup.

G.1.2 **WHILE** venting RHR Loop A, monitor pressure on RHR HDR PRESS LOOP A 10PI-120A.

G.1.3 Open HX A INBD VENT VLV 10MOV-166A.

G.1.4 Open HX A OUTBD VENT VLV 10MOV-167A.

G.1.5 **WHEN** header pressure lowers to approximately 60 psig, close the following valves:

- HX A INBD VENT VLV 10MOV-166A
- HX A OUTBD VENT VLV 10MOV-167A

G.1.6 Verify annunciator 09-3-1-18 RHR A OR B DISCH LINE NOT FULL is clear.

NOTE: Subsection G.2 is used when header is pressurizing for reasons such as leakage by 10RHR-68B or 10MOV-70B.

G.2 **Venting RHR Loop B to Lower Header Pressure**

G.2.1 Verify RHR Loop B is in the standby LPCI lineup.

G.2.2 **WHILE** venting RHR Loop B, monitor pressure on RHR HDR PRESS LOOP B 10PI-120B.

G.2.3 Open HX B INBD VENT VLV 10MOV-166B.

G.2.4 Open HX B OUTBD VENT VLV 10MOV-167B.

G.2.5 **WHEN** header pressure lowers to approximately 60 psig, close the following valves:

- HX B INBD VENT VLV 10MOV-166B
- HX B OUTBD VENT VLV 10MOV-167B

G.2.6 Verify annunciator 09-3-1-18 RHR A OR B DISCH LINE NOT FULL is clear.

G.3 **Continuous Venting of RHR Header A**

G.3.1 Verify RHR A Keepfull System is in service per OP-13E with either 10P-2A or Condensate Transfer.

NOTE: The continuous venting lineup may need to be secured during sampling by Chemistry.

G.3.2 Notify Chemistry and Radwaste Control Room.

G.3.3 Establish communications between 95SP-7 and Control Room during establishment of sample flow.

G.3.4 Ensure open the following at 95SP-7:

- 10SOV-263A RHR HX A OUTLET INNER SAMPLE SOLENOID VALVE
- 10SOV-264A RHR HX A OUTLET OUTER SAMPLE SOLENOID VALVE

CAUTION

Pressure in the RHR header must be maintained **GREATER THAN** 50 psig to prevent flashing of water within piping.

G.3.5 Throttle open 95SS1-714 (RHR heat exchanger A outlet grab sample isol valve) to lower RHR A header pressure to the desired pressure.

NOTE 1: Position of 95SS1-714 may need to be changed to stablize pressure based upon changing plant conditions; i.e. change in RHR lineup or leakage rate.

NOTE 2: Contact Chemistry if necessary for assistance or beaker.

G.3.6 Throttle 95SS1-714 to establish a stable RHR A header pressure **AND** a sample flow rate **LESS THAN** 0.5 gpm.

IF a stable pressure can not be obtained with flow rate less than 0.5 gpm,
THEN initiate a PID for Technical Services to evaluate leakage.

G.3.7 Ensure yellow THROTTLED tag is hung on 95SS1-714.

G.3.8 **WHEN** continuous venting is no longer desired, ensure closed the following:

- 95SS1-714 (RHR heat exchanger A outlet grab sample isol valve)
- 10SOV-263A RHR HX A OUTLET INNER SAMPLE SOLENOID VALVE
- 10SOV-264A RHR HX A OUTLET OUTER SAMPLE SOLENOID VALVE

G.3.9 Ensure yellow THROTTLED tag removed from 95SS1-714.

G.3.10 Ensure RHR A keep-full alarms are clear.

G.3.11 Notify Chemistry and Radwaste Control Room that venting is secured.

G.4 **Continuous Venting of RHR Header B**

G.4.1 Verify RHR B Keepfull System is in service per OP-13E with either 10P-2B or Condensate Transfer.

NOTE: The continuous venting lineup may need to be secured during sampling by Chemistry.

G.4.2 Notify Chemistry and Radwaste Control Room.

G.4.3 Establish communications between 95SP-7 and Control Room during establishment of sample flow.

G.4.4 Ensure open the following at 95SP-7:

- 10SOV-263B RHR HX B OUTLET INNER SAMPLE SOLENOID VALVE
- 10SOV-264B RHR HX B OUTLET OUTER SAMPLE SOLENOID VALVE

CAUTION

Pressure in the RHR header must be maintained **GREATER THAN** 50 psig to prevent flashing of water within piping.

NOTE: 10RHR-780B is throttled and locked to limit maximum flow via sample line to 0.5 gpm with 95SS1-715 full open.

G.4.5 Throttle open 95SS1-715 (RHR heat exchanger B outlet grab sample isol valve) to lower RHR B header pressure to the desired pressure.

NOTE 1: Position of 95SS1-715 may need to be changed to stablize pressure based upon changing plant conditions; i.e. change in RHR lineup or leakage rate.

NOTE 2: Contact Chemistry if necessary for assistance or beaker.

G.4.6 Throttle 95SS1-715 to establish a stable RHR B header pressure **AND** a sample flow rate **LESS THAN** 0.5 gpm.

IF a stable pressure can not be obtained with flow rate less than 0.5 gpm,
THEN initiate a PID for Technical Services to evaluate leakage.

G.4.7 Ensure yellow THROTTLED tag is hung on 95SS1-715.

G.4.8 **WHEN** continuous venting is no longer desired, ensure closed the following:

- 95SS1-715 (RHR heat exchanger B outlet grab sample isol valve)
- 10SOV-263B RHR HX B OUTLET INNER SAMPLE SOLENOID VALVE
- 10SOV-264B RHR HX B OUTLET OUTER SAMPLE SOLENOID VALVE

G.4.9 Ensure yellow THROTTLED tag removed from 95SS1-715.

G.4.10 Ensure RHR B keep-full alarms are clear.

G.4.11 Notify Chemistry and Radwaste Control Room that venting is secured.

H. **ATTACHMENTS**

1. REFERENCES

2A. TABLE 1 - VALVE LINEUP - RHR LOOP A

2B. TABLE 1 - VALVE LINEUP - RHR LOOP B

2C. TABLE 1 - VALVE LINEUP - COMMON TO RHR LOOPS A AND B

3A. TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP A

3B. TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP B

3C. TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - COMMON TO RHR LOOP A AND B

ATTACHMENT 1

REFERENCES

1.0 **Performance References**

1.1 ODSO-4, Shift Turnover and Log Keeping*

1.2 OP-5, Condensate Transfer System*

1.3 OP-19, Reactor Core Isolation Cooling System*

1.4 OP-21, Emergency Service Water (ESW)*

1.5 OP-27, Recirculation System*

1.6 OP-39, Breathing, Instrument, and Service Air System*

1.7 OP-40, Reactor Building Closed Loop Cooling*

1.8 OP-42, Service Water System*

1.9 OP-43A, 125 VDC Power System*

1.10 OP-43C, LPCI Independent Power Supply System*

1.11 OP-46A, 4160 V and 600 V Normal AC Power Distribution*

1.12 OP-46B, 120 VAC Power System*

1.13 OP-49, Liquid Radioactive Waste System*

1.14 OP-50, Equipment and Floor Drain System*

REFERENCES

2.0 **Developmental References**

2.1 FSAR Section 4.8

2.2 FM-20A, 20B

2.3 JSEM-91-061, Outstanding Action Items Pertaining To Valve Misapplication

2.4 JSEM-91-067, Effects Of Void Formation

2.5 JSEM-91-082, QA Classification of 10MOV-57 and 10MOV-67

2.6 Modification F1-92-091 Rx Head Spray Removal

2.7 Evaluation Of 10MOV-16A/B As Normally Open Valve

2.8 JSEM-92-047, RHR Pump Suction Valves

2.9 JPEM-92-025, Special - Appendix J List

2.10 Modification M1-92-334, Disabling of Valves 10MOV-36B, 70A, 70B, 15MOV-101, 102, and 103

2.11 JPEM-92-029, Revise OP-13 to state that operation of any single RHR pump at less than 6500 gpm or operation of any two RHR pumps in parallel at flows less 13,000 gpm should be avoided. ACTS Item 5857.

2.12 ACTS Item 4832, Operations Department revise applicable procedures listed in Attachments 1 and 2 to ensure the proper positioning of test connections, valves, and caps on penetrations exposed to the containment atmosphere.

2.13 JAF-SE-99-014, JD-99-099, RHR - Steam Condensing

2.14 JTS-95-0221, Operability Assessment for DER 95-0740 - 0748; Industry Notification of B-Fill Qualification Limits for Certain ITT-Barton Indicating Switches

2.15 JSED-95-0100, Impact of ITT Barton Industry Advisory on the Environmental Qualification of 10DPIS-125A&B, 14FIS-45A&B, and 27PS-110A&B

2.16 Memo from Jim Van Ben Coten to Peter Abbott dated 6/3/95, Backseating Manual Valves in Primary Containment

2.17 JAF-SE-95-063, Operation of Four RHR Pumps in Torus Cooling Flow Mode

REFERENCES

2.18 Modification M1-89-086, Oil Sampling Valves for RHR, RHRSW, and Core Spray Pump Motors

2.19 Modification M1-97-021, RHR High Point Vent

2.20 JAF-SE-99-015, Modification to Eliminate RHR Header Pressurization

TABLE 1 - VALVE LINEUP - RHR LOOP A
CONTROL ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10MOV-13A	RHR Pump A Suct Torus Isol Valve	Open		
10MOV-13C	RHR Pump C Suct Torus Isol Valve	Open		
10MOV-15A	RHR Pump A Suct Shutdown Cooling Isol Valve	Closed		
10MOV-15C	RHR Pump C Suct Shutdown Cooling Isol Valve	Closed		
10MOV-66A	RHR Heat Exch A Bypass Valve	Open		
10MOV-65A	RHR Heat Exch A Shell Inlet Isol Valve	Open		
10MOV-12A	RHR Heat Exch A Outlet Isol Valve	Open		
10MOV-25A	RHR A LPCI Inbd Inj Valve	Closed		
10MOV-27A	RHR A LPCI Outbd Inj Valve	Open		
10MOV-39A	RHR A Torus Cooling Isol Valve	Closed		
10MOV-38A	RHR A to Torus Spray Isol Valve	Closed		
10MOV-34A	RHR A Torus Cooling Supply Valve	Closed		
10MOV-26A	RHR A Cont Spray Outbd Isol Valve	Closed		
10MOV-31A	RHR A Cont Spray Inbd Isol Valve	Closed		
10MOV-21A	RHR Heat Exch A Disch to Torus Isol Valve	↓COMA.2.2 Closed		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
CONTROL ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10MOV-36A	RHR Heat Exch A Disch to RCIC Isol Valve	Closed		
10MOV-148A	RHRSW A to RHR Cross Tie Upstr Isol Valve	Closed		
10MOV-149A	RHRSW A to RHR Cross Tie Dnstr Isol Valve	Closed		
10PCV-69A	RHR Heat Exchanger A Steam Inlet Press Control Valve	Closed		
10AOV-71A	RHR Heat Exchanger A Outlet to Torus or RCIC Isol Valve	Closed		
10MOV-166A	RHR Heat Exch A Upstr Vent to Torus Isol Valve	Closed		
10MOV-167A	RHR Heat Exch A Dnstr Vent to Torus Isol Valve	Closed		
10MOV-16A	RHR A Min Flow Isol Valve	Open		
10AOV-68A	RHR A LPCI Testable Check Valve	Closed		
10MOV-89A	RHR Heat Exch A Serv Water Outlet Isol Valve	Closed		
10MOV-57	RHR to Radwaste Throttle Valve	Closed		
10MOV-67	RHR to Radwaste Isol Valve	Closed		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
RX BLDG. 300 EL.

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-51A	RHR Loop A Containment Spray Air Test Connection Isol Valve	Closed		
**10RHR-52A	RHR Loop A Containment Spray Air Test Header Isol Valve	Locked Closed		
10RHR-731A1	RHR Loop A Containment Spray Air Test Inner Vent Valve	Closed		
10RHR-731A2	RHR Loop A Containment Spray Air Test Outer Vent Valve	Closed		
**10RHR-748A	RHR Loop A Containment Spray Inner ILRT Connection Isol Valve	Locked Closed		
**10RHR-749A	RHR Loop A Containment Spray Outer ILRT Connection Isol Valve	Locked Closed and Capped		
10RHR-759A	RHR Loop A Containment Spray Line 10LS-100 Root Valve	Open		
10RHR-760A	RHR Loop A Containment Spray Line 10LS-100 Drain Valve	Closed		
10RHR-761A	RHR Loop A Containment Spray Line 10LS-100 Inner Overflow Valve	Closed		
10RHR-762A	RHR Loop A Containment Spray Line 10LS-100 Outer Overflow Valve	Closed		
10RHR-404A	RHR Loop A Containment Spray Inner Vent Valve	Closed		
10RHR-405A	RHR Loop A Containment Spray Outer Vent Valve	Closed		
10RHR-274	RHR Loop A Containment Spray Keep-Full Cond Xfer Connection Valve	*		

* Closed when 10P-2A is running, otherwise throttled.

**Valve position is required to comply with the requirements of 10CFR50, Appendix J. Do not administratively change the required position of this valve.ACTS Item 4832

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
RX BLDG. 300 EL.

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-273	RHR Loop A Containment Spray Keep-Full Cond Xfer Connection Valve	Closed		
10RHR-275	RHR Loop A Containment Spray Keep-Full Drain Valve	Closed		
10SOV-263A	RHR Heat Exchanger A Outlet Inner Sample Solenoid Valve	Closed		
10SOV-264A	RHR Heat Exchanger A Outlet Outer Sample Solenoid Valve	Closed		
*10RHR-988	RHR Loop A Containment Spray 10FT-136A Hi Side Root Valve	Open		
*10RHR-989	RHR Loop A Containment Spray 10FT-136A Lo Side Root Valve	Open		
10RHR-414	RHR Loop A Cond Xfer Supply Line Check Valve Test Conn Isol Valve	Closed and Capped		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
RX BLDG. 272 EL.

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-420A	RHRSW Loop A Supply Line 10LS-105A Root Valve	Open		
10RHR-421A	RHRSW Loop A Supply Line 10LS-105A Inner Overflow Valve	Closed		
10RHR-422A	RHRSW Loop A Supply Line 10LS-105A Outer Overflow Isol Valve	Closed		
10RHR-423A	RHRSW Loop A Supply Line 10LS-105A Drain Valve	Closed		
10SOV-150A	RHRSW Loop A to RHR Cross-Tie Drain Solenoid Valve	Energized		
10RHR-721A	RHRSW Loop A to RHR Cross-Tie Inner Drain Valve	Closed		
10RHR-722A	RHRSW Loop A to RHR Cross-Tie Outer Drain Valve	Closed		
10RHR-205A	RHR Heat Exchanger A RHRSW Rad Element 17RE-457A Inlet Isol Valve	Closed		
10RHR-206A	RHR Heat Exchanger A RHRSW Rad Element 17RE-457A Outlet Isol Valve	Closed		
10RHR-773	RHRSW Loop A Return Header Drain Valve	Closed		
*10RHR-688	RHRSW Loop A 10FT-97A Hi Side Root Valve	Open		
*10RHR-689	RHRSW Loop A 10FT-97A Lo Side Root Valve	Open		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
RX BLDG. 272 EL.

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
*10RHR-53A	RHR Loop A Containment Spray 10PT-101A, 05PS-16 & 05PT-12A Sensing Line Isol Valve	Open		
*10RHR-53C	RHR Loop A Containment Spray 10PT-101C, PS-119A&C & 05PT-12B Sensing Line Isol Valve	Open		
**10RHR-60A	RHR Loop A Containment Spray Instrument Line (capped) Isol Valve	Locked Closed and Capped		
**10RHR-60C	RHR Loop A Containment Spray Instrument Line (capped) Isol Valve	Locked Closed and Capped		
**10RHR-983	RHR Loop A Containment Spray 05PS-16 Sensing Line Inner Drain Valve	Locked Closed		
**10RHR-984	RHR Loop A Containment Spray 05PS-16 Sensing Line Outer Drain Valve	Locked Closed and Capped		
**10RHR-985	RHR Loop A Containment Spray 05PS-16 Sensing Line Inner Drain Valve	Locked Closed		
**10RHR-986	RHR Loop A Containment Spray 05PS-16 Sensing Line Outer Drain Valve	Locked Closed and Capped		

*Valves are listed on ST-40H.

**Valve position is required to comply with the requirements of 10CFR50, Appendix J. Do not administratively change the required position of this valve.ACTS Item 4832

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
A RHR HX ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-11A	RHR Heat Exchanger A RHRSW Inlet Isol Valve	Locked Open		
10RHR-24A	RHR Heat Exchanger A RHRSW Outlet Isol Valve	Locked Open		
10RHR-717A	RHR Heat Exchanger A Tube Side Inner Drain Valve	Closed		
10RHR-718A	RHR Heat Exchanger A Tube Side Outer Drain Valve	Closed		
10RHR-719A	RHR Heat Exchanger A Tube Side Inner Drain Valve	Closed		
10RHR-720A	RHR Heat Exchanger A Tube Side Outer Drain Valve	Closed		
10RHR-727A	RHR Heat Exchanger A Shell Side Inner Drain Valve	Closed		
10RHR-728A	RHR Heat Exchanger A Shell Side Outer Drain Valve	Closed		
10RHR-747A	RHR Heat Exchanger A Tube Side Chem Clean Isol Valve	Closed		
10RHR-750A	RHR Heat Exchanger A Tube Side Chem Clean Isol Valve	Closed		
10RHR-780A	RHR Heat Exchanger A Outlet Sample Isol Valve	Open		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
A RHR HX ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
*10RHR-690	RHR Heat Exchanger A 10LT-98A Lower Root Valve	Open		
*10RHR-691	RHR Heat Exchanger A 10LT-98A Upper Root Valve	Open		
*10RHR-692	RHR Heat Exchanger A Steam Inlet 10PT-114A Root Valve	Open		
10RHR-401A	RHR Heat Exchanger A HPCI Steam Supply Inner Vent Valve	Closed		
10RHR-402A	RHR Heat Exchanger A HPCI Steam Supply Outer Vent Valve	Closed		
10RHR-451A	RHR Heat Exchanger A Shell Side Vent Valve	Closed		
10RHR-715A	RHR Heat Exchanger A Steam Inlet Inner Drain Valve	Closed		
10RHR-716A	RHR Heat Exchanger A Steam Inlet Outer Drain Valve	Closed		
10MOV-70A	RHR Heat Exch A Steam Inlet Isol Valve	Closed		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
WEST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-403A	RHR Pump A Suct Line From Suppression Pool ILRT Connection Isol Valve	Locked Closed and Capped		
10RHR-93A	RHR Pump A Suct From Suppression Pool Spare Tap (welded cap)	Locked Closed		
10RHR-729A	RHR Pumps A & C Suct From Suppression Pool Drain Valve	Locked Closed		
10RHR-412A	RHR Pump A Shutdown Cooling Vent Valve	Closed		
10RHR-91A	RHR Keep-Full Pump A Suct Isol Valve	Open		
10RHR-91C	RHR Keep-Full Pump A Suct Isol Valve	Open		
10RHR-675A	RHR Pump A Suct Shutdown Cooling Vent Valve	Closed		
10RHR-675C	RHR Pump C Suct Shutdown Cooling Vent Valve	Closed		
10RHR-251A	RHR Pump A Suct Drain Valve	Closed		
10RHR-251C	RHR Pump C Suct Drain Valve	Closed		
10RHR-400A	RHR Pumps A & C Suct Relief Line Vent Valve	Locked Closed		
10RHR-450A	RHR Pump A Seal Cavity Vent Valve	Closed		
10RHR-742A	RHR Pump A Casing Drain Valve	Closed		
10RHR-23A	RHR Pump A Seal Leakoff Valve	Open		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
WEST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-743A	RHR Pump A Cooling Water Leakage Combined Drain Valve	Open		
10RHR-450C	RHR Pump C Seal Cavity Vent Valve	Closed		
10RHR-742C	RHR Pump C Casing Drain Valve	Closed		
10RHR-23C	RHR Pump C Seal Leakoff Valve	Open		
10RHR-743C	RHR Pump C Cooling Water Leakage Combined Drain Valve	Open		
10RHR-28A	Locked RHR Pump A Min Flow Isol Valve	Open		
10RHR-28C	Locked RHR Pump C Min Flow Isol Valve	Open		
10RHR-96A	RHR Keep-Full Pump A Min Flow Valve	Open		
10RHR-250A	RHR Pump A Disch Drain Valve	Closed		
10RHR-250C	RHR Pump C Disch Drain Valve	Closed		
10RHR-730	RHR Pumps A & C Combined Drains Isol Valve	Closed		
10RHR-714A	RHR Pumps A & C Disch Drains to Waste Collector Inlet Header Isol Valve	Closed		
10RHR-252A	RHR Pumps A & C Combined Drains to Sump Isol Valve	Closed		
10RHR-113A	RHR Pumps A & C Suct Drain Conduct Element 113A Isol Valve	Open		
10RHR-45A	RHR Pump A Disch Isol Valve	Locked Open		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
WEST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-45C	RHR Pump C Disch Isol Valve	Locked Open		
10RHR-99A	RHR Keep-Full Pump A Disch to RHR Loop A Isol Valve	Open		
10RHR-201	RHR Loop A Post Accident Sample Isol Valve	Open		
10RHR-776	RHR Loop A Supply Line Inner Drain Valve	Closed		
10RHR-777	RHR Loop A Supply Line Outer Drain Valve	Closed		
10RHR-255A	RHR Loop A Cond Xfer Isol Valve	Closed		
10RHR-409A	RHR Loop A Torus Spray Line ILRT Connection Isol Valve	Locked Closed and Capped		
10RHR-732A	RHR Loop A Torus Spray Line Drain Valve	Locked Closed and Capped		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
WEST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-413	RHR Loop A Full Flow Test Line Vent Valve	Locked Closed		
10RHR-696	RHR Pump A Disch Spare Instrument Isol Valve	Closed and Capped		
10RHR-697	RHR Pump C Disch Spare Instrument Isol Valve	Closed and Capped		
*10RHR-676	RHR Pump A Suct 10PI-106A & A1 Root Valve	Open		
*10RHR-677	RHR Pump C Suct 10PI-106C & C1 Root Valve	Open		
*10RHR-695	RHR Pump A Disch 10PI-107A2, 10PS-120A & E Root Valve	Open		
*10RHR-693	RHR Pump C Disch 10PI-107C2, 10PS-120C & G Root Valve	Open		
*10RHR-680	RHR Pump A Disch Drain Isol Valve	Closed and Capped		
*10RHR-681	RHR Pump C Disch Drain Valve	Closed and Capped		
*10RHR-694	RHR Loop A 10PT-120A & 10PS-123 Root Valve	Open		
*10RHR-971	RHR Loop A 10FT-109A & 10DPIS-125A Hi Side Root Valve	Open		
*10RHR-972	RHR Loop A 10FT-109A & 10DPIS-125A Lo Side Root Valve	Open		
*10RHR-973	RHR Loop A 10PS-122A Root Valve	Open		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
WEST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-778A	RHR Pump A Motor Upper Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-779A	RHR Pump A Motor Lower Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-778C	RHR Pump C Motor Upper Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-779C	RHR Pump C Motor Lower Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-425A	RHR Keep-Full Pump A Suct A Suct 10PI-277A Root Valve	Open		
10RHR-426A	RHR Keep-Full Pump A Disch 10PI-278A Root Valve	Open		
10RHR-1101	RHR Keep-Full Pump A Min Flow Check Valve RHR-95A ILRT Connection Isol Valve	Locked Closed and Capped		
10RHR-1102	RHR Keep-Full Pump A Min Flow Check Valve RHR-95A ILRT Connection Isol Valve	Locked Closed and Capped		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
TORUS AREA

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
***10RHR-408A	RHR Loop A Torus Spray Line Vent Valve	Locked Closed and Capped		
10RHR-775A	RHR Pumps A & C Suct From Suppression Pool Drain Valve	Locked Closed and Capped		
*10MOV-151A	RHR Pumps A & C Suct Suppr Pool Isol Valve	Locked Open*		
**10RHR-990	RHR Loop A Torus Spray 10FT-137A Hi Side Root Valve	Open		
**10RHR-991	RHR Loop A Torus Spray 10FT-137A Lo Side Root Valve	Open		

* Electrically Disarmed

**Valves are listed on ST-40H.

*** Valve position is required to comply with the requirements of 10CFR50, Appendix J. Do not administratively change the required position of this valve.ACTS Item 4832

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
DRYWELL ENTRANCE

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-78A	RHR Loop A LPCI Inbd Isol Valve 10MOV-25A Inner Inlet Drain Valve	Locked Closed		
10RHR-79A	RHR Loop A LPCI Inbd Isol Valve 10MOV-25A Outer Inlet Drain Valve	locked Closed		
10RHR-256A	RHR Loop A LPCI Cond Xfer Inner Isol Valve	Closed		
10RHR-257A	RHR Loop A LPCI Cond Xfer Outer Isol Valve	Closed		
10RHR-786A	RHR Loop A LPCI Inbd Isol Valve 10MOV-25A Stem Leakoff Valve	Open		
10RHR-767A	RHR Loop A Keep-Full 10LS-101 Root Valve	Open		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
DRYWELL

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-735A	RHR Loop A LPCI ILRT Conn Outer Isol Valve	Locked Closed and Capped		
10RHR-736A	RHR Loop A LPCI ILRT Conn Inner Isol Valve	Locked Closed		
10RHR-700A	RHR Loop A Testable Check AOV-68A Inlet Drain Inner Isol Valve	Closed		
10RHR-701A	RHR Loop A Testable Check AOV-68A Inlet Drain Outer Isol Valve	Closed		
10RHR-702A	RHR Loop A Testable Check AOV-68A Outlet Drain Inner Isol Valve	Closed		
10RHR-703A	RHR Loop A Testable Check AOV-68A Outlet Drain Outer Isol Valve	Closed		
10RHR-713	RHR Loop A Isol RHR-81A Stem Leakoff Isol Valve	Closed		
10RHR-737A	RHR Loop A Testable Check AOV-68A ILRT Conn Isol Valve	Locked Closed and Capped		
10RHR-81A	RHR A LPCI to RWR A Loop Isol Valve	Locked Open & Backseated		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE NEUP - RHR LOOP A
RHRSW . MP ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-22A	~~Locked~~ RHRSW Pump A Disch Isol Valve	Locked Open		
10RHR-272A	RHRSW Pump A Motor Bearing Cooling Water Supply Isol Valve	Open		
10RHR-744A	RHRSW Pump A Motor Bearing Cooling Water Return Isol Valve	Open		
10RHR-22C	~~Locked~~ RHRSW Pump C Disch Isol Valve	Locked Open		
10RHR-272C	RHRSW Pump C Motor Bearing Cooling Water Supply Isol Valve	Open		
10RHR-744C	RHRSW Pump C Motor Bearing Cooling Water Return Isol Valve	Open		
10RHR-746	RHRSW Pumps A & C Motors Bearing Cooling Water Supply Header Drain Valve	Closed		
10RHR-754A	RHRSW Pumps A & C Disch Header Drain Valve	Closed		
10RHR-430A	RHRSW Loop A Keep-Full SWS Supply Isol Valve	Open		
10RHR-432A	RHRSW Loop A Keep-Full SWS Supply Isol Valve	Open		
10RHR-432	RHRSW to Fire Protection Cross-Tie Isol Valve	Locked Closed and Capped		
10RHR-792	RHRSW to Fire Protection Cross-Tie Bleed Off Valve	Closed		
10RHR-757A	RHRSW Pumps A & C Disch Strainer Basket 2 Drain Valve	Closed		
10RHR-758A	RHRSW Pumps A & C Disch Strainer Basket 1 Drain Valve	Closed		
10RHR-111A	RHRSW Pumps A & C Disch Strainer A Equalizing Valve	Closed		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A

RHRSW PUMP ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-630A	RHRSW Pump A Disch 10PI-117A Root Valve	Open		
10RHR-630C	RHRSW Pump C Disch 10PI-117C Root Valve	Open		
*10RHR-686	RHRSW Pumps A & C Disch Strainer A 10DPIS-277A Hi Side Root Valve	Open		
*10RHR-687	RHRSW Pumps A & C Disch Strainer A 10DPIS-277A Lo Side Root Valve	Open		
10RHR-723A	RHRSW Pump A Motor Upper Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-724A	RHRSW Pump A Motor Lower Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-723C	RHRSW Pump C Motor Upper Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-724C	RHRSW Pump C Motor Lower Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-500A	RHRSW Loop A Keep-Full Check Valve RHR-431A Leak Test Connection Isol Valve	Closed and Capped		
10RHR-501A	RHRSW Loop A Keep-Full Check Valve RHR-431A Leak Test Connection Isol Valve	Closed and Capped		
	TUNNEL			
10RHR-772A	RHRSW Loop A Supply Header Drain Valve	Closed		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP A
REACTOR BUILDING DRYWELL
ENTRANCE MEZZANINE

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-781A	RHR Loop A LPCI Keep-Full 10LS-101 Isol Valve	Open		
10RHR-784A	RHR Loop A LPCI Keep-Full 10LS-101 Drain Valve	Closed		
10RHR-783A	RHR Loop A LPCI Keep-Full 10LS-101 Drain Valve	Closed		
10RHR-782A	RHR Loop A LPCI Keep-Full N2 Connection 10LS-101 Drain Valve	Closed		
10RHR-788A	RHR Loop A LPCI Keep-Full 10PI-131A Drain Valve	Closed		
10RHR-789A	RHR Loop A LPCI LLRT Connection Isol Valve	Closed and Capped		
10RHR-787A	RHR Loop A LPCI Keep-Full 10PI-131A Root Valve	Locked Closed		
10RHR-785A	RHR Loop A LPCI Keep-Full N2 Isol Valve	Closed and Capped		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
CONTROL ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10MOV-13B	RHR Pump B Suct Torus Isol Valve	Open		
10MOV-13D	RHR Pump D Suct Torus Isol Valve	Open		
10MOV-15B	RHR Pump B Suct Shutdown Cooling Isol Valve	Closed		
10MOV-15D	RHR Pump D Suct Shutdown Cooling Isol Valve	Closed		
10MOV-66B	RHR Heat Exch B Bypass Valve	Open		
10MOV-65B	RHR Heat Exch B Shell Inlet Isol Valve	Open		
10MOV-12B	RHR Heat Exch B Outlet Isol Valve	Open		
10MOV-25B	RHR B LPCI Inbd Inj Valve	Closed		
10MOV-27B	RHR B LPCI Outbd Inj Valve	Open		
10MOV-39B	RHR B Torus Cooling Isol Valve	Closed		
10MOV-38B	RHR B to Torus Spray Isol Valve	Closed		
10MOV-34B	RHR B Torus Cooling Supply Valve	Closed		
10MOV-26B	RHR B Cont Spray Outbd Isol Valve	Closed		
10MOV-31B	RHR B Cont Spray Inbd Isol Valve	Closed		
10MOV-21B	RHR Heat Exch B Disch to Torus Isol Valve	↓COMA.2.2 Closed		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
CONTROL ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10MOV-148B	RHRSW B to RHR Cross Tie Upstr Isol Valve	Closed		
10MOV-149B	RHRSW B to RHR RHR Cross Tie Dnstr Isol Valve	Closed		
10PCV-69B	RHR Heat Exchanger B Steam Inlet Press Control Valve	Closed		
10AOV-71B	RHR Heat Exchanger B Outlet to Torus or RCIC Isol Valve	Closed		
10MOV-166B	RHR Heat Exch B Upstr Vent to Torus Isol Valve	Closed		
10MOV-167B	RHR Heat Exch B Dnstr Vent to Torus Isol Valve	Closed		
10MOV-16B	RHR B Min Flow Isol Valve	Open		
10AOV-68B	RHR B LPCI Testable Check Valve	Closed		
10MOV-89B	RHR Heat Exch B Serv Water Outlet Isol Valve	Closed		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
RX BLDG. 326 EL.

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-406	Head Spray Line Inner Vent Valve	Closed		
10RHR-407	Head Spray Line Middle Vent Valve	Closed		
10RHR-411	Reactor Head Spray Line Outer Vent Valve	Closed		
10RHR-763	Reactor Head Spray Line 10LS-102 Root Valve	Open		
10RHR-764	Reactor Head Spray Line 10LS-102 Drain Valve	Closed		
10RHR-765	Reactor Head Spray Line 10LS-102 Inner Drain Valve	Closed		
10RHR-766	Reactor Head Spray Line 10LS-102 Outer Drain Valve	Closed		
**10RHR-708	Reactor Head Spray Line Inner ILRT Connection Isol Valve	Locked Closed		
**10RHR-709	Reactor Head Spray Line Outer ILRT Connection Isol Valve	Locked Closed and Capped		

**Valve position is required to comply with the requirements of 10CFR50, Appendix J. Do not administratively change the required position of this valve.ACTS Item 4832

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
RX BLDG. 300 EL.

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-51B	RHR Loop B Containment Spray Air Test Connection Isol Valve	Closed		
**10RHR-52B	RHR Loop B Containment Spray Air Test Header Isol Valve	Locked Closed		
10RHR-731B	RHR Loop B Containment Spray Air Test Vent Valve	Closed		
**10RHR-748B	RHR Loop B Containment Spray Inner ILRT Connection Isol Valve	Locked Closed		
**10RHR-749B	RHR Loop B Containment Spray Outer ILRT Connection Isol Valve	Locked Closed and Capped		
10RHR-759B	RHR Loop B Containment Spray Line 10LS-103 Root Valve	Open		
10RHR-760B	RHR Loop B Containment Spray Line 10LS-103 Drain Valve	Closed		
10RHR-761B	RHR Loop B Containment Spray Line 10LS-103 Inner Overflow Valve	Closed		
10RHR-762B	RHR Loop B Containment Spray Line 10LS-103 Outer Overflow Valve	Closed		
10RHR-404B	RHR Loop B Containment Spray Inner Vent Valve	Closed		
10RHR-405B	RHR Loop B Containment Spray Outer Vent Valve	Closed		

**Valve position is required to comply with the requirements of 10CFR50, Appendix J. Do not administratively change the required position of this valve.ACTS Item 4832

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
RX BLDG. 300 EL.

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10SOV-263B	RHR Heat Exchanger B Outlet Inner Sample Solenoid Valve	Closed		
10SOV-264B	RHR Heat Exch B Outlet Outer Sample Solenoid Valve	Closed		
*10RHR-616	10FT-136B High Side Root Valve	Open		
*10RHR-617	10FT-136B Low Side Root Valve	Open		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
RX BLDG. 272 EL.

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-260	RHR Loop B Reactor Head Spray Keep-Full 10PCV-266 Bypass Valve	*		
10RHR-268	Reactor Head Spray Keep-Full 10PCV-266 Outlet Isol Valve	Closed		
10RHR-269	Reactor Head Spray Keep-Full 10PCV-266 Inlet Isol Valve	Closed		
10RHR-76	Reactor Head Spray Line Drain Valve	Closed		
10RHR-752	Reactor Head Spray Line Drain Valve	Closed		
10SOV-150B	RHRSW Loop B to RHR Cross-Tie Drain Solenoid Valve	Energized		
10RHR-721B	RHRSW Loop B to RHR Cross-Tie Inner Drain Valve	Closed		
10RHR-722B	RHRSW Loop B to RHR Cross-Tie Outer Drain Valve	Closed		
10RHR-420B	RHRSW Loop B Supply Line 10LS-105B Root Valve	Open		
10RHR-421B	RHRSW Loop B Supply Line 10LS-105B Inner Overflow Valve	Closed		
10RHR-422B	RHRSW Loop B Supply Line 10LS-105B Outer Overflow Isol Valve	Closed		
10RHR-423B	RHRSW Loop B Supply Line 10LS-105B Drain Valve	Closed		
10RHR-609	10FT-103 High Side Isol Valve	Closed		
10RHR-610	10FT-103 Low Side Isol Valve	Closed		

*Closed when 10P-2B is running, otherwise throttled.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
RX BLDG. 272 EL.

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
*10RHR-53B	RHR Loop B Containment Spray 10PT-101B, 05PT-12C Sensing Line Isol Valve	Open		
*10RHR-53D	RHR Loop B Containment Spray 10PT-101D, PS-B&D & 05PT-12D Sensing Line Isol Valve	Open		
* **10RHR-60B	RHR Loop B Containment Spray Instrument Line (capped) Isol Valve	Locked Closed and Capped		
* **10RHR-60D	RHR Loop B Containment Spray Instrument Line (capped) Isol Valve	Locked Closed and Capped		
*10RHR-664	RHRSW Loop B 10FT-97B & 10FT-134 Hi Side Root Valve	Open		
*10RHR-665	RHRSW Loop B 10FT-97B & 10FT-134 Lo Side Root Valve	Open		

*Valves are listed on ST-40H.

**Valve position is required to comply with the requirements of 10CFR50, Appendix J. Do not administratively change the required position of this valve.ACTS Item 4832

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
RX BLDG. 272 EL.

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
**10RHR-611	RHR Loop B Cont Spray 05PT-12 Sensing Line Inner Drain Valve	Locked Closed		
**10RHR-612	RHR Loop B Cont Spray 05PT-12 Sensing Line Outer Drain Valve	Locked Closed and Capped		
**10RHR-613	RHR Loop B Cont Spray 05PT-12 Sensing Line Inner Drain Valve	Locked Closed		
**10RHR-614	RHR Loop B Cont Spray 05PT-12 Sensing Line Outer Drain Valve	Locked Closed and Capped		
10RHR-415	RHR Loop B Cond Xfer Supply Line Check Valve Test Conn Isol Valve	Closed and Capped		
10RHR-797A	RHRSW Loop B to RHR Cross-Tie Inner Drain Valve	Closed		
10RHR-797B	RHRSW Loop B to RHR Cross-Tie Outer Drain Valve	Closed and Capped		
10RHR-992	RHR 10PCV-266 Sensing Line Isolation Valve	Closed		

**Valve position is required to comply with the requirements of 10CFR50, Appendix J. Do not administratively change the required position of this valve.ACTS Item 4832

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
B RHR HX ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-11B	RHR Heat Exchanger B RHRSW Inlet Isol Valve	Locked Open		
10RHR-24B	RHR Heat Exchanger B RHRSW Outlet Isol Valve	Locked Open		
10RHR-717B	RHR Heat Exchanger B Tube Side Inner Drain Valve	Closed		
10RHR-718B	RHR Heat Exchanger B Tube Side Outer Drain Valve	Closed		
10RHR-719B	RHR Heat Exchanger B Tube Side Inner Drain Valve	Closed		
10RHR-720B	RHR Heat Exchanger B Tube Side Outer Drain Valve	Closed		
10RHR-727B	RHR Heat Exchanger B Shell Side Inner Drain Valve	Closed		
10RHR-728B	RHR Heat Exchanger B Shell Side Outer Drain Valve	Closed		
10RHR-747B	RHR Heat Exchanger B Tube Side Chem Clean Isol Valve	Closed		
10RHR-750B	RHR Heat Exchanger B Tube Side Chem Clean Isol Valve	Closed		
10RHR-780B	RHR Heat Exchanger B Outlet Sample Isol Valve	* Locked Throttled		
10RHR-225	RHR Heat Exchanger B RHRSW Aux Steam Inlet Isol Valve	Closed		
10RHR-227	RHR Heat Exchanger B RHRSW Steam Trap Outlet Isol Valve	Closed		
10RHR-768	RHR Heat Exchanger B RHRSW Steam Trap Drain Valve	Closed		
10RHR-769	RHR Heat Exchanger B RHRSW Steam Trap Drain Valve	Closed		

* Throttled and locked to limit flow rate in sample line to 0.5 gpm to comply with JAF-SE-99-015.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
B RHR HX ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-205B	RHR Heat Exchanger B RHRSW Rad Element 17RE-457B Inlet Isol Valve	Closed		
10RHR-206B	RHR Heat Exchanger B RHRSW Rad Element 17RE-457B Outlet Isol Valve	Closed		
10RHR-401B	RHR Heat Exchanger B HPCI Steam Supply Inner Vent Valve	Closed		
10RHR-402B	RHR Heat Exchanger B HPCI Steam Supply Outer Vent Valve	Closed		
10RHR-451B	RHR Heat Exchanger B Shell Side Vent Valve	Closed		
10RHR-715B	RHR Heat Exchanger B Steam Inlet Inner Drain Valve	Closed		
10RHR-716B	RHR Heat Exchanger B Steam Inlet Outer Drain Valve	Closed		
*10RHR-661	RHR Heat Exchanger B Steam Inlet 10PT-114B Root Valve	Open		
*10RHR-662	RHR Heat Exchanger B 10LT-98B Upper Root Valve	Open		
*10RHR-663	RHR Heat Exchanger B 10LT-98B Lower Root Valve	Open		
10MOV-70B	RHR Heat Exch B Steam Inlet Isol Valve	Closed		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
EAST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-403B	RHR Pump Suct Line From Suppression Pool ILRT Connection Isol Valve	Locked Closed and Capped		
10RHR-93B	RHR Pump B Suct From Suppression Pool Spare Tap (Welded Cap)	Locked Closed		
10RHR-729B	RHR Pumps B & D Suct From Suppression Pool Drain Valve	Locked Closed		
10RHR-412B	RHR Pumps B & D Suction Header Vent Valve	Closed		
10RHR-91B	RHR Keep-Full Pump B Suct Isol Valve	Open		
10RHR-91D	RHR Keep-Full Pump B Suct Isol Valve	Open		
10RHR-675B	RHR Pump B Suct Shutdown Cooling Vent Valve	Closed		
10RHR-675D	RHR Pump D Suct Shutdown Cooling Vent Valve	Closed		
10RHR-400B	RHR Pumps B & D Suct Relief Line Vent Valve	Locked Closed		
10RHR-251B	RHR Pump B Suct Drain Valve	Closed		
10RHR-251D	RHR Pump D Suct Drain Valve	Closed		
10RHR-714B	RHR Pumps B & D Disch Drains to Waste Collector Inlet Header Isol Valve	Closed		
10RHR-252B	RHR Pumps B & D Combined Drains to Sump Isol Valve	Closed		
10RHR-253	RHR Pumps B & D Disch Drains to Equip Sump Isol Valve	Closed		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
EAST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-113B	RHR Pumps B & D Suct Drain Conduct Element 113B Isol Valve	Open		
10RHR-450B	RHR Pump B Seal Cavity Vent Valve	Closed		
10RHR-742B	RHR Pump B Casing Drain Valve	Closed		
10RHR-23B	RHR Pump B Seal Leakoff Valve	Open		
10RHR-743B	RHR Pump B Cooling Water Leakage Combined Drain Valve	Open		
10RHR-450D	RHR Pump D Seal Cavity Vent Valve	Closed		
10RHR-742D	RHR Pump D Casing Drain Valve	Closed		
10RHR-23D	RHR Pump D Seal Leakoff Valve	Open		
10RHR-743D	RHR Pump D Cooling Water Leakage Combined Drain Valve	Open		
10RHR-28B	Locked RHR Pump B Min Flow Isol Valve	Open		
10RHR-28D	Locked RHR Pump D Min Flow Isol Valve	Open		
10RHR-250B	RHR Pump B Disch Drain Valve	Closed		
10RHR-250D	RHR Pump D Disch Drain Valve	Closed		
10RHR-45B	RHR Pump B Disch Isol Valve	Locked Open		
10RHR-45D	RHR Pump D Disch Isol Valve	Locked Open		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
EAST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-96B	RHR Keep-Full Pump B Min Flow Valve	Open		
10RHR-99B	RHR Keep-Full Pump B Disch to RHR Loop B Isol Valve	Open		
10RHR-202	RHR Loop B Post Accident Sample Isol Valve	Open		
10RHR-409B	RHR Loop B Torus Spray Line ILRT Connection Isol Valve	Locked Closed and Capped		
10RHR-732B	RHR Loop B Torus Spray Line Drain Valve	Locked Closed and Capped		
10MOV-36B	RHR Heat Exch B Disch to RCIC Isol Valve	Closed		
10RHR-698	RHR Pump B Disch Spare Instrument Isol Valve	Closed and Capped		
10RHR-699	RHR Pump D Disch Spare Instrument Isol Valve	Closed and Capped		
*10RHR-678	RHR Pump B Suct 10PI-106B & B1 Root Valve	Open		
*10RHR-679	RHR Pump D Suct 10PI-106D & D1 Root Valve	Open		
*10RHR-659	RHR Pump B 10PI-107B2, 10PS-120B & 10PS-120F Root Valve	Open		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
EAST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
*10RHR-658	RHR Pump D Disch 10PI-107D2, 10PS-120D & H Root Valve	Open		
*10RHR-682	RHR Pump B Disch Drain Valve	Closed and Capped		
*10RHR-683	RHR Pump D Disch Drain Valve	Closed and Capped		
*10RHR-660	RHR Loop B 10PT-120B & 10PS-123B Root Valve	Open		
*10RHR-602	RHR Loop B 10FT-109B & 10DPIS-125B Hi Side Root Valve	Open		
*10RHR-601	RHR Loop B 10FT-109B & 10DPIS-125B Lo Side Root Valve	Open		
*10RHR-600	RHR Loop B 10PS-122B Root Valve	Open		
10RHR-778B	RHR Pump B Motor Upper Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-779B	RHR Pump B Motor Lower Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-778D	RHR Pump D Motor Upper Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-779D	RHR Pump D Motor Lower Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-425B	RHR Keep-Full Pump B Suct 10PI-277B Root Valve	Open		
10RHR-426B	RHR Keep-Full Pump B Disch 10PI-278B Root Valve	Open		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B

EAST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-1103	RHR Keep-Full Pump B Min Flow Check Valve RHR-95B ILRT Connection Isol Valve	Locked Closed and Capped		
10RHR-1104	RHR Keep-Full Pump B Min Flow Check Valve RHR-95B ILRT Connection Isol Valve	Locked Closed and Capped		
10RHR-39B-1	RHR Loop B Torus Cooling Isol Valve 10MOV-39B Bonnet Press Equalizing Valve	Locked Open		

WEST CRESCENT

10RHR-255B	RHR Loop B Cond Xfer Isol Valve	Closed		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
TORUS AREA

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
***10RHR-408B	RHR Loop B Torus Spray Line Vent Valve	Locked Closed and Capped		
10RHR-775B	RHR Pumps B & D Suct From Suppression Pool Drain Valve	Locked Closed and Capped		
10MOV-151B	RHR Pumps B & D Suct Suppr Pool Isol Valve	Locked Open*		
**10RHR-618	RHR Loop B Torus Spray 10FT-137B Hi Side Root Valve	Open		
**10RHR-619	RHR Loop B Torus Spray 10FT-137B Lo Side Root Valve	Open		

* Electrically Disarmed

**Valves are listed on ST-40H.

*** Valve position is required to comply with the requirements of 10CFR50, Appendix J. Do not administratively change the required position of this valve.ACTS Item 4832

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
DRYWELL ENTRANCE

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-78B	RHR Loop B LPCI Inbd Isol Valve 10MOV-25B Inner Inlet Isol Valve	Locked Closed		
10RHR-79B	RHR Loop B LPCI Inbd Isol Valve 10MOV-25B Outer Inlet Drain Valve	Locked Closed		
10RHR-767B	RHR Loop B Keep-Full 10LS-104 Root Valve	Open		
10RHR-256B	RHR Loop B LPCI Cond Xfer Inner Isol Valve	Closed		
10RHR-257B	RHR Loop B LPCI Cond Xfer Outer Isol Valve	Closed		
10RHR-786B	RHR Loop B LPCI Inbd Isol Valve 10MOV-25B Stem Leakoff Valve	Open		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
DRYWELL

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-706	RHR Loop B Isol RHR-81B Stem Leakoff Isol Valve	Closed		
10RHR-81B	RHR B LPCI to RWR B Loop Isol Valve	Locked Open & Backseated		
10RHR-700B	RHR Loop B Testable Check AOV-68B Inlet Drain Inner Isol Valve	Closed		
10RHR-701B	RHR Loop B Testable Check AOV-68B Inlet Drain Outer Isol Valve	Closed		
10RHR-702B	RHR Loop B Testable Check AOV-68B Outlet Drain Inner Isol Valve	Closed		
10RHR-703B	RHR Loop B Testable Check AOV-68B Outlet Drain Outer Isol Valve	Closed		
10RHR-735B	RHR Loop B LPCI ILRT Conn Outer Isol Valve	Locked Closed and Capped		
10RHR-736B	RHR Loop B LPCI ILRT Conn Inner Isol Valve	Locked Closed		
10RHR-737B	RHR Loop B Testable Check AOV-68B Inner ILRT Isol Valve	Locked Closed		
10RHR-738B	RHR Loop B Testable Check AOV-68B Outer ILRT Isol Valve	Locked Closed and Capped		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE VEUP - RHR LOOP B
RHRSW PUMP ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-22B	~~Locked~~ RHRSW Pump B Disch Isol Valve	Locked Open		
10RHR-272B	RHRSW Pump B Motor Bearing Cooling Water Supply Isol Valve	Open		
10RHR-744B	RHRSW Pump B Motor Bearing Cooling Water Return Isol Valve	Open		
10RHR-22D	~~Locked~~ RHRSW Pump D Disch Isol Valve	Locked Open		
10RHR-272D	RHRSW Pump D Motor Bearing Cooling Water Supply Isol Valve	Open		
10RHR-744D	RHRSW Pump D Motor Bearing Cooling Water Return Isol Valve	Open		
10RHR-430B	RHRSW Loop B Keep-Full SWS Supply Isol Valve	Open		
10RHR-432B	RHRSW Loop B Keep-Full SWS Supply Isol Valve	Open		
10RHR-745	RHRSW Pumps B & D Motors Bearing Cooling Water Supply Header Drain Valve	Closed		
10RHR-754B	RHRSW Pumps B & D Disch Header Drain Valve	Closed		
10RHR-757B	RHRSW Pumps B & D Disch Strainer Basket 2 Drain Valve	Closed		
10RHR-758B	RHRSW Pumps B & D Disch Strainer Basket 1 Drain Valve	Closed		
10RHR-111B	RHRSW Pumps B & D Disch Strainer B Equalizing Valve	Closed		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
	RHRSW PUMP ROOM			
10RHR-630B	RHRSW Pump B Disch 10PI-117B Root Valve	Open		
10RHR-630D	RHRSW Pump D Disch 10PI-117D Root Valve	Open		
*10RHR-666	RHRSW Pumps B & D Disch Strainer B 10DPIS-277B Hi Side Root Valve	Open		
*10RHR-667	RHRSW Pumps B & D Disch Strainer B 10DPIS-277B Lo Side Root Valve	Open		
10RHR-723B	RHRSW Pump B Motor Upper Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-724B	RHRSW Pump B Motor Lower Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-723D	RHRSW Pump D Motor Upper Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-724D	RHRSW Pump D Motor Lower Bearing Oil Drain and Sample Valve	Closed and Capped		
10RHR-500B	RHRSW Loop B Keep-Full Check Valve RHR-431B Leak Test Connection Isol Valve	Closed and Capped		
10RHR-501B	RHRSW Loop B Keep-Full Check Valve RHR-431B Leak Test Connection Isol Valve	Closed and Capped		
	TUNNEL			
10RHR-772B	RHRSW Loop B Supply Header Drain Valve	Closed		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - RHR LOOP B
REACTOR BUILDING DRYWELL
ENTRANCE MEZZANINE

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-781B	RHR Loop B LPCI Keep-Full 10LS-104 Isol Valve	Open		
10RHR-784B	RHR Loop B LPCI Keep-Full 10LS-104 Drain Valve	Closed		
10RHR-783B	RHR Loop B LPCI Keep-Full 10LS-104 Drain Valve	Closed		
10RHR-782B	RHR Loop B LPCI Keep-Full N2 Connection 10LS-104 Drain Valve	Closed		
10RHR-788B	RHR Loop B LPCI Keep-Full 10P1-131B Drain Valve	Closed		
10RHR-789B	RHR Loop B LPCI LLRT Connection Isol Valve	Locked Closed and Capped		
10RHR-787B	RHR Loop B LPCI Keep-Full 10PI-131B Root Valve	Locked Closed		
10RHR-785B	RHR Loop B LPCI Keep-Full N2 Isol Valve	Closed and Capped		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - COMMON TO RHR LOOPS A AND B

CONTROL ROOM

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10MOV-17	RHR Shutdown Cooling Outbd Isol Valve	Closed		
10MOV-18	RHR Shutdown Cooling Inbd Isol Valve	Closed		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - COMMON TO RHR LOOPS A AND B
EAST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-09	RHR Loops A & B Headers Cross-Tie Isol Valve	Locked Closed		
10RHR-770	RHR Loops A & B Cross-Tie Inner Drain Valve	Closed		
10RHR-771	RHR Loops A & B Cross-Tie Outer Drain Valve	Closed		
10RHR-410	RHR to RCIC Line Vent Valve	Closed		
*10RHR-982	RHR Shutdown Cooling 10PS-118 Root Valve	Open		
10RHR-795A	RHR Loop B Cross-Tie to Fuel Pool Outer Drain Valve	Closed		
10RHR-795B	RHR Loop B Cross-Tie to Fuel Pool Inner Drain Valve	Closed and Capped		
10RHR-796A	RHR Pumps Suct From Fuel Pool Outer Drain Valve	Closed		
10RHR-796B	RHR Pumps Suct From Fuel Pool Inner Drain Valve	Closed and Capped		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - COMMON TO RHR LOOPS A AND B
WEST CRESCENT

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-254	RHR Loops A & B Cond Xfer Isol Valve	Closed		
10MOV-20	RHR Disch Hdr Cross Tie Valve	Locked Closed		
10RHR-725	RHR Pumps A & C Shutdown Cooling Inner Drain Valve	Closed		
10RHR-726	RHR Pumps A & C Shutdown Cooling Outer Drain Valve	Closed		
10RHR-741	RHR to RCIC Relief Line Vent Valve	Locked Closed and Capped		
*10RHR-603	RHR Return to RCIC 10PT-146 & 10PI-147 Root Valve	Open		

*Valves are listed on ST-40H.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - COMMON TO RHR LOOPS A AND B
DRYWELL ENTRANCE

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-84	RHR Shutdown Cooling Outbd Isol Valve 10MOV-17 Upstr Inner ILRT Isol Valve	Locked Closed		
10RHR-85	RHR Shutdown Cooling Outbd Isol Valve 10MOV-17 Upstr Outer ILRT Isol Valve	Locked Closed and Capped		
10RHR-86	RHR Shutdown Cooling Outbd Isol Valve 10MOV-17 Dnstr Inner ILRT Isol Valve	Locked Closed		
10RHR-87	RHR Shutdown Cooling Outbd Isol Valve 10MOV-17 Dnstr Outer ILRT Isol Valve	Locked Closed and Capped		
10RHR-258	RHR Shutdown Cooling Outer Cond Xfer Isol Valve	Closed		
10RHR-259	RHR Shutdown Cooling Inner Cond Xfer Isol Valve	Closed		
10RHR-17-1	RHR Shutdown Cooling Outbd Isol Valve 10MOV-17 Bonnet Press Equalizing Valve	Locked Open		
10RHR-440	RHR SDC High Point Vent Valve	Locked Closed		
10RHR-441	RHR SDC High Point Vent Valve	Locked Closed and Capped		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 1 - VALVE LINEUP - COMMON TO RHR LOOPS A AND B
DRYWELL

VALVE NO.	DESCRIPTION	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10RHR-704	RHR Shutdown Cooling Isol RHR-88 Stem Leakoff Isol Valve	Closed		
10RHR-800	Shutdown Cooling Inbd Isol MOV-18 Press Equalizing Valve	Locked Open & Backseated		
10RHR-88	RHR Shutdown Cooling Inbd Isol Valve	Locked Open & Backseated		
10RHR-739	Shutdown Cooling Inner ILRT Isol Valve	Locked Closed		
10RHR-740	Shutdown Cooling Outer ILRT Isol Valve	Locked Closed and Capped		
10RHR-790	Shutdown Cooling Inner Drain Valve	Locked Closed		
10RHR-791	Shutdown Cooling Outer Drain Valve	Locked Closed and Capped		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP A

COMPONENT NUMBER	DESCRIPTION	POWER SUPPLY	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10P-3A	Residual Heat Removal Pump A	71-10550	Racked In		
10P-3C	Residual Heat Removal Pump C	71-10650	Racked In		
10P-1A	RHR Service Water Pump A	71-10520	Racked In		
10P-1C	RHR Service Water Pump C	71-10510	Racked In		
10MOV-166A	RHR Heat Exch A Upstr Vent to Torus Isol Valve	71MCC-151-OC1	ON		
10MOV-167A	RHR Heat Exch A Dnstr Vent to Torus Isol Valve	71MCC-151-OC2	ON		
10MOV-12A	RHR Heat Exch A Outlet Isol Valve	71MCC-151-OD1	ON		
10MOV-65A	RHR Heat Exch A Shell Inlet Isol Valve	71MCC-153-OD2	ON		
10MOV-27A	RHR A LPCI Outbd Inj Valve	71MCC-155-OH2	ON		
10MOV-25A	RHR A LPCI Inbd Inj Valve	71MCC-155-OJ4	ON		
10MOV-13A	RHR Pump A Suct Torus Isol Valve	71MCC-153-OA4	ON		
10MOV-13C	RHR Pump C Suct Torus Isol Valve	71MCC-153-OA5	ON		
10MOV-15A	RHR Pump A Suct Shutdown Cooling Isol Valve	71MCC-153-OC4	ON		
10MOV-15C	RHR Pump C Suct Shutdown Cooling Isol Valve	71MCC-153-OF1	ON		
10MOV-16A	RHR A Min Flow Isol Valve	71MCC-153-OB3	ON		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP A

COMPONENT NUMBER	DESCRIPTION	POWER SUPPLY	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10MOV-89A	RHR Heat Exch A Serv Water Outlet Isol Valve	71MCC-151-OD2	ON		
10MOV-26A	RHR A Cont Spray Outbd Isol Valve	71MCC-152-OF1	ON		
10MOV-31A	RHR A Cont Spray Inbd Isol Valve	71MCC-152-OF4	ON		
10MOV-34A	RHR A Torus Cooling Supply Valve	71MCC-153-OC1	ON		
10MOV-38A	RHR A to Torus Spray Isol Valve	71MCC-153-OC3	ON		
10MOV-39A	RHR A Torus Cooling Isol Valve	71MCC-153-OD1	ON		
10MOV-67	RHR to Radwaste Isol Valve	71BMCC-2-OC2	ON		
10MOV-57	RHR to Radwaste Throttle Valve	71MCC-151-OD4	ON		
10MOV-36A	RHR Heat Exch A Disch to RCIC Isol Valve	71MCC-153-OC2	ON		
10MOV-21A	RHR Heat Exch A Disch to Torus Isol Valve	71MCC-153-OB4	ON		
10MOV-70A	RHR Heat Exch A Steam Inlet Isol Valve	71MCC-151-OE2	OFF *		
10MOV-66A	RHR Heat Exch A Bypass Valve	71MCC-155-OH1	ON		

* Valve power supply normal position is OFF to prevent spurious valve operation during a postulated Appendix R fire per MOD M1-92-334.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP A

COMPONENT NUMBER	DESCRIPTION	POWER SUPPLY	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10MOV-148A	RHRSW A to RHR Cross-Tie Upstr Isol Valve	71MCC-151-OD3	ON		
10MOV-149A	RHRSW A to RHR Cross-Tie Dnstr Isol Valve	71MCC-151-OE3	ON		
10SOV-69A	RHR Heat Exchanger A Steam Inlet 10PCV-69A Solenoid Valve	71ESSA1, Fuse 10A-F20A in 09-32	Fuse In		
10SOV-71A	RHR Heat Exchanger A to Torus or RCIC Isol Valve 10AOV-71A Solenoid Valve	71ESSA1, Fuse 10A-F26A in 09-32	Fuse In		
10SOV-150A	RHRSW Loop A to RHR Cross-Tie Drain Solenoid Valve	71ESSA1, Fuse 10A-F30A in 09-32	Fuse In		
10SOV-263A	RHR Heat Exchanger A Outlet Inner Sample Solenoid Valve	71ESSA1, Fuse 10A-F18A in 09-3	Fuse In		
10SOV-264A	RHR Heat Exchanger A Outlet Outer Sample Solenoid Valve	71ESSA1, Fuse 10A-F31A in 09-3	Fuse In		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP A

COMPONENT NUMBER	DESCRIPTION	POWER SUPPLY	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10SOV-68A	RHR A Testable Check Air Supply Isol Solenoid Valve	71ACUPS, Fuse 10A-F33 in 09-3	Fuse In		
	RHR A Logic	71DC-A2 CKT #4	ON		
Panel 09-32	RHR A Logic	Fuse 10A-F1A & 2A in 09-32	Fuse In		
10RHR-81A Indication	RHR A LPCI to RWR A Loop Isol Valve Inside Drywell	Fuse 10A-F15A in 09-3	Fuse In		
Panel 09-3	RHR Instruments Bus A	Bkrs 3,4,5 and 12 71ESSA1	ON		
10SOV-101A	RHRSW Pump A Motor Cooling Water Return Solenoid Valve	ACB 10520 Trip Coil Circuit	Fuse In		
10SOV-101C	RHRSW Pump C Motor Cooling Water Return Solenoid Valve	ACB 10510 Trip Coil Circuit	Fuse In		
10P-2A	RHR Keep-Full Pump A	71MCC-153-OE1	ON		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP B

COMPONENT NUMBER	DESCRIPTION	POWER SUPPLY	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10P-3B	Residual Heat Removal Pump B	71-10540	Racked In		
10P-3D	Residual Heat Removal Pump D	71-10640	Racked In		
10P-1B	RHR Service Water Pump B	71-10620	Racked In		
10P-1D	RHR Service Water Pump D	71-10610	Racked In		
10MOV-166B	RHR Heat Exch B Upstr Vent to Torus Isol Valve	71MCC-161-OC1	ON		
10MOV-167B	RHR Heat Exch B Dnstr Vent to Torus Isol Valve	71MCC-161-OC2	ON		
10MOV-12B	RHR Heat Exch B Outlet Isol Valve	71MCC-161-OB1	ON		
10MOV-65B	RHR Heat Exch B Shell Inlet Isol Valve	71MCC-163-OF2	ON		
10MOV-27B	RHR B LPCI Outbd Inj Valve	71MCC-165-OB2	ON		
10MOV-25B	RHR B LPCI Inbd Inj Valve	71MCC-165-OA4	ON		
10MOV-13B	RHR Pump B Suct Torus Isol Valve	71MCC-163-OJ4	ON		
10MOV-13D	RHR Pump D Suct Torus Isol Valve	71MCC-163-OJ5	ON		
10MOV-15B	RHR Pump B Suct Shutdown Cooling Isol Valve	71MCC-163-OD1	ON		
10MOV-15D	RHR Pump D Suct Shutdown Cooling Isol Valve	71MCC-163-OG4	ON		
10MOV-16B	RHR B Min Flow Isol Valve	71MCC-163-OH3	ON		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP B

COMPONENT NUMBER	DESCRIPTION	POWER SUPPLY	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10MOV-89B	RHR Heat Exch B Serv Water Outlet Isol Valve	71MCC-161-OB2	ON		
10MOV-26B	RHR B Cont Spray Outbd Isol Valve	71MCC-161-OC3	ON		
10MOV-31B	RHR B Cont Spray Inbd Isol Valve	71MCC-161-OD2	ON		
10MOV-34B	RHR B Torus Cooling Supply Valve	71MCC-163-OG1	ON		
10MOV-38B	RHR B to Torus Spray Isol Valve	71MCC-163-OG3	ON		
10MOV-39B	RHR B Torus Cooling Isol Valve	71MCC-163-OF1	ON		
10MOV-36B	RHR Heat Exch B Disch to RCIC Isol Valve	71MCC-163-OG2	OFF *		
10MOV-21B	RHR Heat Exch B Disch to Torus Isol Valve	71MCC-163-OH4	ON		
10MOV-70B	RHR Heat Exch B Steam Inlet Isol Valve	71MCC-161-OA2	OFF *		
10MOV-66B	RHR Heat Exch B Bypass Valve	71MCC-165-OB1	ON		

* Valve power supply normal position is OFF to prevent spurious valve operation during a postulated Appendix R fire per MOD M1-92-334.

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP B

COMPONENT NUMBER	DESCRIPTION	POWER SUPPLY	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10MOV-148B	RHRSW B to RHR Cross-Tie Upstr Isol Valve	71MCC-161-OB3	ON		
10MOV-149B	RHRSW B to RHR Cross-Tie Dnstr Isol Valve	71MCC-161-OA3	ON		
10SOV-69B	RHR Heat Exchanger B Steam Inlet 10PCV-69B Solenoid Valve	71ESSB1, Fuse 10A-F20B in 09-33	Fuse In		
10SOV-71B	RHR Heat Exchanger B to Torus or RCIC Isol Valve 10AOV-71B Solenoid Valve	71ESSB1, Fuse 10A-F26B in 09-33	Fuse In		
10SOV-150B	RHRSW Loop B to RHR Cross-Tie Drain Solenoid Valve	71ESSB1, Fuse 10A-F30B in 09-33	Fuse In		
10SOV-263B	RHR Heat Exchanger B Outlet Inner Sample Solenoid Valve	71ESSB1, Fuse 10A-F18B in 09-3	Fuse In		
10SOV-264B	RHR Heat Exchanger B Outlet Outer Sample Solenoid Valve	71ESSB1, Fuse 10A-F31B in 09-3	Fuse In		
10SOV-68B	RHR B Testable Check Air Supply Isol Solenoid Valve	71ACUPS, Fuse 10A-F33 in 09-3	Fuse In		
	RHR B Logic	71DC-B2 CKT #1	ON		
Panel 09-33	RHR B Logic	Fuse 10A-F1B & 2B in 09-33	Fuse In		
10RHR-81B Indication	RHR B LPCI to RWR B Loop Isol Valve Inside Drywell	Fuse 10A-F15B in 09-3	Fuse In		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 2 - MAJOR COMPONENT POWER SUPPLIES - RHR LOOP B

COMPONENT NUMBER	DESCRIPTION	POWER SUPPLY	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
Panel 09-3	RHR Instruments Bus B	Bkrs 3,4,5 and 8 71ESSB1	ON		
10SOV-101B	RHRSW Pump B Motor Cooling Water Return Solenoid Valve	ACB 10620 Trip Coil Circuit	Fuse In		
10SOV-101D	RHRSW Pump D Motor Cooling Water Return Solenoid Valve	ACB 10610 Trip Coil Circuit	Fuse In		
10P-2B	RHR Keep-Full Pump B	71MCC-163-OE1	ON		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

TABLE 2 - MAJOR COMPONENT POWER SUPPLIES
COMMON TO RHR LOOP A AND B

COMPONENT NUMBER	DESCRIPTION	POWER SUPPLY	NORMAL POSITION	ACTUAL POSITION	INITIALS & DATE
10MOV-20	RHR Disch Hdr Cross Tie Valve	71MCC-132-OB3	OFF-OVERLOAD HTRS REMOVED		
10MOV-18	RHR Shutdown Cooling Inbd Isol Valve	71MCC-156-OD2	ON		
10DS-1PCIA07	RHR MOV-18 Disconnect Switch		LOCKED OFF		
10IS-1PCIA07	RHR MOV-18 Isol Switch		REMOTE		
10MOV-17	Outbd Isol Valve	71BMCC-4-OD2	ON		

COMPLETED LINEUPS ARE A QUALITY RECORD TO BE RETAINED PER AP-02.08.

NEW YORK POWER AUTHORITY
JAMES A. FITZPATRICK NUCLEAR POWER PLANT
OPERATING PROCEDURE

RHR - CONTAINMENT CONTROL*
OP-13B
REVISION 3

APPROVED BY: _____ DATE 9/15/99
RESPONSIBLE PROCEDURE OWNER

EFFECTIVE DATE: 9-20-99

FIRST ISSUE ☐ FULL REVISION ☐ LIMITED REVISION ☒

REFERENCE USE	TSR
TECHNICAL	

PERIODIC REVIEW DUE DATE 9-2004

FOR INFORMATION ONLY

REVISION SUMMARY SHEET

REV. NO.	CHANGE AND REASON FOR CHANGE
3	Updated cover page to reflect completion of periodic review. Superceeds EC dated 4/23/98. Incorporated operator aids 513, 514, and 517 for torus spray, torus cooling and drywell spray into posted attachments. One common posted attachment was created to be used for both loops to minimize excessive posted attachments. (pcr#6 dated 1/23/98)(posted attachments not rev barred) Included steps to shut down loop in service into Subsections G.1 and G.2 instead of referencing Section F. Minimizes chance of missing steps when utilizing multiple subsections during a frequently used procedure. (pcr#5 dated 11/14/97) Added information to Section E indicating RHR is maintained in standby lineup per ODSO-4. Previous information referenced OP-13 which then referenced ODSO-4. Deleted note in Subsections D.3 through D.6 which stated that the SPRAY CONTROL switch had to be manipulated on each successive spray. The step should be performed regardless of whether there is a note or not. Moved caution boxes to more appropriate locations within Subsections G.1, G.2 and D.1 through D.6 to improve user interface. Added note specifying procedure may continue during isolation of condensate transfer in Subsections D.1 through D.6. The steps following the isolation are not impacted.

TABLE OF CONTENTS

SECTION		PAGE
A.	**REQUIREMENTS**	4
B.	**SYSTEM DESCRIPTION**	4
C.	**PLANT OPERATING REQUIREMENTS**	4
D.	**STARTUP**	6
E.	**NORMAL OPERATION**	23
F.	**SHUTDOWN**	24
G.	**SPECIAL PROCEDURES**	30
H.	**ATTACHMENTS**	41
	1. REFERENCES	42
	2. TORUS COOLING STARTUP (POSTED)	43
	3. TORUS SPRAY STARTUP (POSTED)	44
	4. DRYWELL SPRAY STARTUP (POSTED)	45
	5. RHRSW STARTUP AND TEMPERATURE CONTROL (POSTED)	46
	6. CONTAINMENT COOLING/SPRAY SHUTDOWN (POSTED)	47

A. REQUIREMENTS

A.1 Technical Specifications

Sections 3.5.A, 3.5.B, 3.5.F, and 3.5.G

A.2 Commitments

A.2.1 JAFP-94-0228, Response to NRC Bulletin No. 93-02, Supplement 1, Debris Plugging of Emergency Core Cooling Suction Strainers. Added precaution to alert operators of the potential for ECCS suction strainer clogging and to adjust flow consistent with required needs to mitigate the clogging.

A.3 Validation

Procedure developed from OP-13, Revision 74, which was validated per ODSO-35.

B. SYSTEM DESCRIPTION

Refer to Section B of OP-13.

C. PLANT OPERATING REQUIREMENTS

C.1 Prerequisites

RHR System valves, power supplies, and support systems are lined up per OP-13, with exceptions approved by Shift Manager.

C.2 Precautions

↓COMA.2.1
C.2.1 Blockage of ECCS pump suction strainers could occur due to debris in the Torus. Within the latitude provided by EOPs to restore and maintain parameters within specified limits, potential mitigative actions may include:

- Minimizing ECCS flow or removing affected ECCS pumps from service
- Alternating ECCS pumps from one division to another, if available
- Shifting ECCS pump suction to another source, if available
- Operation of alternate injection sources

C.2.2 RES Radiation Protection shall be notified before:

- RHR System piping is drained
- RHR System is started up in any mode of operation, unless an emergency exists

C.2.3 An extremely large drain path exists from the RPV to the torus if either of the following conditions occur:

- 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the associated 10MOV-13A, B, C, or D are open at the same time.
- 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D are open and the associated RHR loop torus spray or torus cooling motor operated isolation valves are open.

The following valves are interlocked to prevent establishing this drain path:

- 10MOV-15A, B, C, and D
- 10MOV-13A, B, C, and D
- 10MOV-39A and B

These interlocks shall not be defeated unless system boundaries are isolated.

C.2.4 Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

C.2.5 Running an RHR pump for greater than 10 minutes with only its minimum flow valve open could damage the pump.

C.2.6 An RHR loop will drain down if power is lost to RHR pumps operating in a containment control mode. RHR loop keepfull alarms shall be cleared prior to manually restarting an RHR pump in that loop. ARP 09-3-1-18 RHR A OR B DISCH LINE NOT FULL provides guidance for clearing keepfull alarms.

NOTE: The LPCI MOV bus charger/inverters are affected by harsh radiation fields. LPCI MOV Bus A Inverter 71INV-3A could be limited to 80 hours of operation due to radiation induced failure following a LOCA. LPCI MOV Bus B Inverter 71INV-3B could be limited to 6 hours of operation due to radiation induced failure following a LOCA.

D. **STARTUP**

NOTE: Subsections D.1 through D.6 may be performed independently or concurrently.

TABLE OF CONTENTS

SUBSECTION		PAGE
D.1	**RHR Loop A Torus Cooling Startup**	7
D.2	**RHR Loop B Torus Cooling Startup**	10
D.3	**RHR Loop A Torus Spray Startup**	13
D.4	**RHR Loop B Torus Spray Startup**	15
D.5	**RHR Loop A Drywell Spray Startup**	17
D.6	**RHR Loop B Drywell Spray Startup**	20

CAUTION

Operating both RHR loops in torus cooling should be avoided, unless it is required to maintain torus water temperature less than 95°F, or it has been directed by an AOP, EOP, or test procedure.

D.1 **RHR Loop A Torus Cooling Startup**

D.1.1 **IF** a LPCI auto-initiation signal is sealed in, **THEN** perform the following:

a. **IF** RPV water level is **LESS THAN** zero inches on fuel zone water level indication, **AND** the EOPs permit diverting LPCI flow, **THEN** place DW & TORUS SPRAY VLV OVERRIDE OF FUEL ZONE LVL 10A-S18A keylock switch in MANUAL OVERRD.

b. Place SPRAY CNTRL 10A-S17A switch to MANUAL, spring return to normal.

c. Verify white SPRAY PERM 10A-DS67A light is on.

CAUTION

Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

NOTE: Both RHR pumps are required to be running for maximum torus cooling.

D.1.2 Ensure at least one of the following RHR pumps is running:

- RHR PMP 10P-3A
- RHR PMP 10P-3C

NOTE: 10MOV-34A may be throttled when 10MOV-39A has dual position indication.

D.1.3 Open RHR TEST TORUS CLG & SPRAY 10MOV-39A.

D.1.4 Throttle RHR TEST & TORUS CLG 10MOV-34A to establish desired flow.

D.1.5 **WHEN** RHR Loop A flow is **GREATER THAN** 1100 gpm, ensure closed MIN FLOW VLV 10MOV-16A.

D.1.6 Establish RHRSW flow as follows:

a. Verify one of the following alarms is clear:

- Annunciator 09-4-3-4 RHRSW A OR B DISCH LINE NOT FULL
- EPIC-D-134 10LS-105A

CAUTION

Running an RHRSW pump with flow less than 2500 gpm or greater than 4000 gpm per pump could damage pump.

NOTE: Both RHRSW pumps are required to be running for maximum torus cooling.

b. Start at least one of the following RHRSW pumps:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

c. Throttle RHRSW DISCH VLV FROM HX A 10MOV-89A to establish 2500 to 4000 gpm per RHRSW pump.

NOTE: If a LPCI auto-initiation signal is present, 10MOV-66A will have a sealed in open signal for 3 minutes from when LPCI auto-initiated.

D.1.7 Close HX A BYP VLV 10MOV-66A.

NOTE: Procedure may be continued while condensate transfer is being isolated.

D.1.8 **IF** RHR Loop A condensate transfer keep-full is in service,
AND RHR Loop A pressure is **LESS THAN** condensate transfer pressure,
THEN close 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve).

D.1.9 **WHILE** performing the following step, maintain torus water temperature **GREATER THAN** 68°F.

D.1.10 Control containment heat removal rate using one or a combination of the following methods:

- Throttle RHR TEST & TORUS CLG 10MOV-34A.
- Start or stop second RHR pump in RHR Loop A.
- Throttle RHRSW DISCH VLV FROM HX A 10MOV-89A.
- Start or stop second RHRSW pump in RHRSW Loop A.
- Shut down RHRSW Loop A per Section F of OP-13C.

CAUTION

Operating both RHR loops in torus cooling should be avoided, unless it is required to maintain torus water temperature less than 95°F, or it has been directed by an AOP, EOP, or test procedure.

D.2 **RHR Loop B Torus Cooling Startup**

D.2.1 **IF** a LPCI auto-initiation signal is sealed in, **THEN** perform the following:

a. **IF** RPV water level is **LESS THAN** zero inches on fuel zone water level indication, **AND** the EOPs permit diverting LPCI flow, **THEN** place DW & TORUS SPRAY VLV OVERRIDE OF FUEL ZONE LVL 10A-S18B keylock switch in MANUAL OVERRD.

b. Place SPRAY CNTRL 10A-S17B switch to MANUAL, spring return to normal.

c. Verify white SPRAY PERM 10A-DS67B light is on.

CAUTION

Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

NOTE: Both RHR pumps are required to be running for maximum torus cooling.

D.2.2 Ensure at least one of the following RHR pumps is running:

- RHR PMP 10P-3B
- RHR PMP 10P-3D

NOTE: 10MOV-34B may be throttled when 10MOV-39B has dual position indication.

D.2.3 Open RHR TEST TORUS CLG & SPRAY 10MOV-39B.

D.2.4 Throttle RHR TEST & TORUS CLG 10MOV-34B to establish desired flow.

D.2.5 **WHEN** RHR Loop B flow is **GREATER THAN** 1100 gpm, ensure closed MIN FLOW VLV 10MOV-16B.

D.2.6 Establish RHRSW flow as follows:

a. Verify one of the following alarms is clear:

- Annunciator 09-4-3-4 RHRSW A OR B DISCH LINE NOT FULL
- EPIC-D-135 10LS-105B

CAUTION

Running an RHRSW pump with flow less than 2500 gpm or greater than 4000 gpm per pump could damage pump.

NOTE: Both RHRSW pumps are required to be running for maximum torus cooling.

b. Start at least one of the following RHRSW pumps:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

c. Throttle RHRSW DISCH VLV FROM HX B 10MOV-89B to establish 2500 to 4000 gpm per RHRSW pump.

NOTE: If a LPCI auto-initiation signal is present, 10MOV-66B will have a sealed in open signal for 3 minutes from when LPCI auto-initiated.

D.2.7 Close HX B BYP VLV 10MOV-66B.

NOTE: Procedure may be continued while condensate transfer is being isolated.

D.2.8 **IF** RHR Loop B condensate transfer keep-full is in service,
AND RHR Loop B pressure is **LESS THAN** condensate transfer pressure,
THEN close 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve).

D.2.9 **WHILE** performing the following step, maintain torus water temperature **GREATER THAN** 68°F.

D.2.10 Control containment heat removal rate using one or a combination of the following methods:

- Throttle RHR TEST & TORUS CLG 10MOV-34B.
- Start or stop second RHR pump in RHR Loop B.
- Throttle RHRSW DISCH VLV FROM HX B 10MOV-89B.
- Start or stop second RHRSW pump in RHRSW Loop B.
- Shut down RHRSW Loop B per Section F of OP-13C.

CAUTION

Subsection D.3 shall only be used when directed by EOP-4.

D.3 **RHR Loop A Torus Spray Startup**

D.3.1 Verify torus pressure is **GREATER THAN** 2.7 psig.

D.3.2 **IF** RPV water level is **LESS THAN** zero inches on fuel zone water level indication,
AND the EOPs permit diverting LPCI flow,
THEN place DW & TORUS SPRAY VLV OVERRIDE OF FUEL ZONE LVL 10A-S18A keylock switch in MANUAL OVERRD.

D.3.3 Place SPRAY CNTRL 10A-S17A switch to MANUAL, spring return to normal.

D.3.4 Verify white SPRAY PERM 10A-DS67A light is on.

CAUTION

Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

D.3.5 Ensure available RHR pumps in RHR Loop A are running:

- RHR PMP 10P-3A
- RHR PMP 10P-3C

NOTE: 10MOV-38A may be throttled when 10MOV-39A has dual position indication.

D.3.6 Open RHR TEST TORUS CLG & SPRAY 10MOV-39A.

D.3.7 Throttle TORUS SPRAY INBD VLV 10MOV-38A to establish desired torus spray flow rate.

D.3.8 **WHEN** RHR Loop A flow is **GREATER THAN** 1100 gpm, ensure closed MIN FLOW VLV 10MOV-16A.

D.3.9 Establish RHRSW flow as follows:

a. Verify one of the following alarms is clear:

- Annunciator 09-4-3-4 RHRSW A OR B DISCH LINE NOT FULL
- EPIC-D-134 10LS-105A

CAUTION

Running an RHRSW pump with flow less than 2500 gpm or greater than 4000 gpm per pump could damage pump.

b. Ensure available RHRSW pumps in RHRSW Loop A are running:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

c. Throttle RHRSW DISCH VLV FROM HX A 10MOV-89A to establish 4000 gpm per RHRSW pump.

NOTE: 10MOV-66A will have a sealed in open signal for 3 minutes from when LPCI logic auto-initiated.

D.3.10 Close HX A BYP VLV 10MOV-66A.

NOTE: Procedure may be continued while condensate transfer is being isolated.

D.3.11 **IF** RHR Loop A condensate transfer keep-full is in service,
AND RHR Loop A pressure is **LESS THAN** condensate transfer pressure,
THEN close 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve).

D.3.12 **WHILE** performing the following step, maintain torus water temperature **GREATER THAN** 68°F.

D.3.13 Control containment heat removal rate using one or a combination of the following methods:

- Throttle TORUS SPRAY INBD VLV 10MOV-38A.
- Start or stop second RHR pump in RHR Loop A.
- Throttle RHRSW DISCH VLV FROM HX A 10MOV-89A.
- Start or stop second RHRSW pump in RHRSW Loop A.
- Shut down RHRSW Loop A per Section F of OP-13C.

CAUTION

Subsection D.4 shall only be used when directed by EOP-4.

D.4 **RHR Loop B Torus Spray Startup**

D.4.1 Verify torus pressure is **GREATER THAN** 2.7 psig.

D.4.2 **IF** RPV water level is **LESS THAN** zero inches on fuel zone water level indication,
AND the EOPs permit diverting LPCI flow,
THEN place DW & TORUS SPRAY VLV OVERRIDE OF FUEL ZONE LVL 10A-S18B keylock switch in MANUAL OVERRD.

D.4.3 Place SPRAY CNTRL 10A-S17B switch to MANUAL, spring return to normal.

D.4.4 Verify white SPRAY PERM 10A-DS67B light is on.

CAUTION

Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

D.4.5 Ensure available RHR pumps in RHR Loop B are running:

- RHR PMP 10P-3B
- RHR PMP 10P-3D

NOTE: 10MOV-38B may be throttled when 10MOV-39B has dual position indication.

D.4.6 Open RHR TEST TORUS CLG & SPRAY 10MOV-39B.

D.4.7 Throttle TORUS SPRAY INBD VLV 10MOV-38B to establish desired torus spray flow rate.

D.4.8 **WHEN** RHR Loop B flow is **GREATER THAN** 1100 gpm, ensure closed MIN FLOW VLV 10MOV-16B.

D.4.9 Establish RHRSW flow as follows:

a. Verify one of the following alarms is clear:

- Annunciator 09-4-3-4 RHRSW A OR B DISCH LINE NOT FULL
- EPIC-D-135 10LS-105B

CAUTION

Running an RHRSW pump with flow less than 2500 gpm or greater than 4000 gpm per pump could damage pump.

b. Ensure available RHRSW pumps in RHRSW Loop B are running:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

c. Throttle RHRSW DISCH VLV FROM HX B 10MOV-89B to establish 4000 gpm per RHRSW pump.

NOTE: 10MOV-66B will have a sealed in open signal for 3 minutes from when LPCI logic actuated.

D.4.10 Close HX B BYP VLV 10MOV-66B.

NOTE: Procedure may be continued while condensate transfer is being isolated.

D.4.11 **IF** RHR Loop B condensate transfer keep-full is in service,
AND RHR Loop B pressure is **LESS THAN** condensate transfer pressure,
THEN close 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve).

D.4.12 **WHILE** performing the following step, maintain torus water temperature **GREATER THAN** 68°F.

D.4.13 Control containment heat removal rate using one or a combination of the following methods:

- Throttle TORUS SPRAY INBD VLV 10MOV-38B.
- Start or stop second RHR pump in RHR Loop B.
- Throttle RHRSW DISCH VLV FROM HX B 10MOV-89B.
- Start or stop second RHRSW pump in RHRSW Loop B.
- Shut down RHRSW Loop B per Section F of OP-13C.

CAUTION

Subsection D.5 shall only be used when directed by EOP-4.

D.5 **RHR Loop A Drywell Spray Startup**

D.5.1 Ensure the following components are tripped:

RWR Pumps	Drywell Cooling Fans
• 02-2P-1A	• 68FN-2A, B, C, and D
• 02-2P-1B	• 68FN-4A, B, C, and D

D.5.2 Verify drywell temperature and pressure are within the Drywell Spray Initiation Limit.

D.5.3 **IF** RPV water level is **LESS THAN** zero inches on fuel zone water level indication,
AND the EOPs permit diverting LPCI flow,
THEN place DW & TORUS SPRAY VLV OVERRIDE OF FUEL ZONE LVL 10A-S18A keylock switch in MANUAL OVERRD.

D.5.4 Place SPRAY CNTRL 10A-S17A switch to MANUAL, spring return to normal.

D.5.5 Verify white SPRAY PERM 10A-DS67A light is on.

CAUTION

Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

D.5.6 Ensure available RHR pumps in RHR Loop A are running:

- RHR PMP 10P-3A
- RHR PMP 10P-3C

D.5.7 Open DW SPRAY OUTBD VLV 10MOV-26A.

D.5.8 Throttle DW SPRAY INBD VLV 10MOV-31A to establish desired drywell spray flow rate.

D.5.9 **WHEN** RHR Loop A flow is **GREATER THAN** 1100 gpm, ensure closed MIN FLOW VLV 10MOV-16A.

D.5.10 Establish RHRSW flow as follows:

a. Verify one of the following alarms is clear:

- Annunciator 09-4-3-4 RHRSW A OR B DISCH LINE NOT FULL
- EPIC-D-134 10LS-105A

CAUTION

Running an RHRSW pump with flow less than 2500 gpm or greater than 4000 gpm per pump could damage pump.

b. Ensure available RHRSW pumps in RHRSW Loop A are running:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

c. Throttle RHRSW DISCH VLV FROM HX A 10MOV-89A to establish 4000 gpm per RHRSW pump.

NOTE: 10MOV-66A will have a sealed in open signal for 3 minutes from when LPCI auto-initiated.

D.5.11 Close HX A BYP VLV 10MOV-66A.

NOTE: Procedure may be continued while condensate transfer is being isolated.

D.5.12 **IF** RHR Loop A condensate transfer keep-full is in service,
AND RHR Loop A pressure is **LESS THAN** condensate transfer pressure,
THEN close 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve).

D.5.13 **WHILE** performing the following step, maintain torus water temperature **GREATER THAN** 68°F.

D.5.14 Control containment heat removal rate using one or a combination of the following methods:

- Throttle DW SPRAY INBD VLV 10MOV-31A.
- Start or stop second RHR pump in RHR Loop A.
- Throttle RHRSW DISCH VLV FROM HX A 10MOV-89A.
- Start or stop second RHRSW pump in RHRSW Loop A.
- Shut down RHRSW Loop A per Section F of OP-13C.

CAUTION

Subsection D.6 shall only be used when directed by EOP-4.

D.6 **RHR Loop B Drywell Spray Startup**

D.6.1 Ensure the following components are tripped:

RWR Pumps	Drywell Cooling Fans
• 02-2P-1A	• 68FN-2A, B, C, and D
• 02-2P-1B	• 68FN-4A, B, C, and D

D.6.2 Verify drywell temperature and pressure are within the Drywell Spray Initiation Limit.

D.6.3 **IF** RPV water level is **LESS THAN** zero inches on fuel zone water level indication,
AND the EOPs permit diverting LPCI flow,
THEN place DW & TORUS SPRAY VLV OVERRIDE OF FUEL ZONE LVL 10A-S18B keylock switch in MANUAL OVERRD.

D.6.4 Place SPRAY CNTRL 10A-S17B switch to MANUAL, spring return to normal.

D.6.5 Verify white SPRAY PERM 10A-DS67B light is on.

CAUTION

Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

D.6.6 Ensure available RHR pumps in RHR Loop B are running:

- RHR PMP 10P-3B
- RHR PMP 10P-3D

D.6.7 Open DW SPRAY OUTBD VLV 10MOV-26B.

D.6.8 Throttle DW SPRAY INBD VLV 10MOV-31B to establish desired drywell spray flow rate.

D.6.9 **WHEN** RHR Loop B flow is **GREATER THAN** 1100 gpm, ensure closed MIN FLOW VLV 10MOV-16B.

D.6.10 Establish RHRSW flow as follows:

a. Verify one of the following alarms is clear:

- Annunciator 09-4-3-4 RHRSW A OR B DISCH LINE NOT FULL
- EPIC-D-135 10LS-105B

CAUTION

Running an RHRSW pump with flow less than 2500 gpm or greater than 4000 gpm per pump could damage pump.

b. Ensure available RHRSW pumps in RHRSW Loop B are running:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

c. Throttle RHRSW DISCH VLV FROM HX B 10MOV-89B to establish 4000 gpm per RHRSW pump.

NOTE: 10MOV-66B will have a sealed in open signal for 3 minutes from when LPCI auto-initiated.

D.6.11 Close HX B BYP VLV 10MOV-66B.

NOTE: Procedure may be continued while condensate transfer is being isolated.

D.6.12 **IF** RHR Loop B condensate transfer keep-full is in service,
AND RHR Loop B pressure is **LESS THAN** condensate transfer pressure,
THEN close 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve).

D.6.13 **WHILE** performing the following step, maintain torus water temperature **GREATER THAN** 68°F.

D.6.14 Control containment heat removal rate using one or a combination of the following methods:

- Throttle DW SPRAY INBD VLV 10MOV-31B.
- Start or stop second RHR pump in RHR Loop B.
- Throttle RHRSW DISCH VLV FROM HX B 10MOV-89B.
- Start or stop second RHRSW pump in RHRSW Loop B.
- Shut down RHRSW Loop B per Section F of OP-13C.

E. **NORMAL OPERATION**

During normal plant operation, RHR System is maintained in a standby LPCI lineup, Control Room indications, controls, and switches lined up per ODSO-4.

NOTE: The LPCI MOV bus charger/inverters are affected by harsh radiation fields. LPCI MOV Bus A Inverter 71INV-3A could be limited to 80 hours of operation due to radiation induced failure following a LOCA. LPCI MOV Bus B Inverter 71INV-3B could be limited to 6 hours of operation due to radiation induced failure following a LOCA.

F. **SHUTDOWN**

NOTE: Subsections F.1 through F.8 may be performed independently or concurrently.

TABLE OF CONTENTS

SUBSECTION		PAGE
F.1	**RHR Loop A Torus Cooling Shutdown**	25
F.2	**RHR Loop B Torus Cooling Shutdown**	25
F.3	**RHR Loop A Torus Spray Shutdown**	26
F.4	**RHR Loop B Torus Spray Shutdown**	26
F.5	**RHR Loop A Drywell Spray Shutdown**	27
F.6	**RHR Loop B Drywell Spray Shutdown**	27
F.7	**RHR Loop A Shutdown From Torus Cooling or Containment Spray Mode**	28
F.8	**RHR Loop B Shutdown From Torus Cooling or Containment Spray Mode**	29

F.1 **RHR Loop A Torus Cooling Shutdown**

F.1.1 Close RHR TEST & TORUS CLG 10MOV-34A.

F.1.2 **IF** RHR Loop A flow is **LESS THAN** 500 gpm, **THEN** ensure open MIN FLOW VLV 10MOV-16A.

CAUTION

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

F.1.3 **IF** RHR Loop A operation is not required, **THEN** shut down RHR Loop A per Subsection F.7.

F.2 **RHR Loop B Torus Cooling Shutdown**

F.2.1 Close RHR TEST & TORUS CLG 10MOV-34B.

F.2.2 **IF** RHR Loop B flow is **LESS THAN** 500 gpm, **THEN** ensure open MIN FLOW VLV 10MOV-16B.

CAUTION

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

F.2.3 **IF** RHR Loop B operation is not required, **THEN** shut down RHR Loop B per Subsection F.8.

F.3 **RHR Loop A Torus Spray Shutdown**

F.3.1 Close TORUS SPRAY INBD VLV 10MOV-38A.

F.3.2 **IF** RHR Loop A flow is **LESS THAN** 500 gpm, **THEN** ensure open MIN FLOW VLV 10MOV-16A.

CAUTION

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

F.3.3 **IF** RHR Loop A operation is not required, **THEN** shut down RHR Loop A per Subsection F.7.

F.4 **RHR Loop B Torus Spray Shutdown**

F.4.1 Close TORUS SPRAY INBD VLV 10MOV-38B.

F.4.2 **IF** RHR Loop B flow is **LESS THAN** 500 gpm, **THEN** ensure open MIN FLOW VLV 10MOV-16B.

CAUTION

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

F.4.3 **IF** RHR Loop B operation is not required, **THEN** shut down RHR Loop B per Subsection F.8.

F.5 **RHR Loop A Drywell Spray Shutdown**

F.5.1 Close DW SPRAY INBD VLV 10MOV-31A.

F.5.2 **IF** RHR Loop A flow is **LESS THAN** 500 gpm, **THEN** ensure open MIN FLOW VLV 10MOV-16A.

CAUTION

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

F.5.3 **IF** RHR Loop A operation is not required, **THEN** shut down RHR Loop A per Subsection F.7.

F.6 **RHR Loop B Drywell Spray Shutdown**

F.6.1 Close DW SPRAY INBD VLV 10MOV-31B.

F.6.2 **IF** RHR Loop B flow is **LESS THAN** 500 gpm, **THEN** ensure open MIN FLOW VLV 10MOV-16B.

CAUTION

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

F.6.3 **IF** RHR Loop B operation is not required, **THEN** shut down RHR Loop B per Subsection F.8.

F.7 **RHR Loop A Shutdown From Torus Cooling or Containment Spray Mode**

F.7.1 Ensure one of the RHR Loop A keep-full systems is in service as follows:

- RHR KEEP-FULL PMP 10P-2A is running,

OR

- 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve) is throttled open.

F.7.2 Ensure closed the following valves:

- RHR TEST & TORUS CLG 10MOV-34A
- TORUS SPRAY INBD VLV 10MOV-38A
- DW SPRAY INBD VLV 10MOV-31A

F.7.3 Ensure the following RHR pumps are stopped:

- RHR PMP 10P-3A
- RHR PMP 10P-3C

F.7.4 Ensure closed the following valves:

- RHR TEST TORUS CLG & SPRAY 10MOV-39A
- DW SPRAY OUTBD VLV 10MOV-26A

F.7.5 Ensure open MIN FLOW VLV 10MOV-16A.

F.7.6 Ensure open HX A BYP VLV 10MOV-66A.

F.7.7 **IF** RHRSW Loop A operation <u>is not</u> required, **THEN** shut down RHRSW Loop A as follows:

a. Close RHRSW DISCH VLV FROM HX A 10MOV-89A.

b. Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

FOR INFORMATION O...

F.8 **RHR Loop B Shutdown From Torus Cooling or Containment Spray Mode**

F.8.1 Ensure one of the RHR Loop B keep-full systems is in service as follows:

- RHR KEEP-FULL PMP 10P-2B is running,

OR

- 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve) is throttled open.

F.8.2 Ensure closed the following valves:

- RHR TEST & TORUS CLG 10MOV-34B
- TORUS SPRAY INBD VLV 10MOV-38B
- DW SPRAY INBD VLV 10MOV-31B

F.8.3 Ensure the following RHR pumps are stopped:

- RHR PMP 10P-3B
- RHR PMP 10P-3D

F.8.4 Ensure closed the following valves:

- RHR TEST TORUS CLG & SPRAY 10MOV-39B
- DW SPRAY OUTBD VLV 10MOV-26B

F.8.5 Ensure open MIN FLOW VLV 10MOV-16B.

F.8.6 Ensure open HX B BYP VLV 10MOV-66B.

F.8.7 **IF** RHRSW Loop B operation <u>is not</u> required, **THEN** shut down RHRSW Loop B as follows:

a. Close RHRSW DISCH VLV FROM HX B 10MOV-89B.

b. Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

G. SPECIAL PROCEDURES

NOTE: Subsections G.1 through G.4 are independent procedures and may be performed in any order, provided steps within each subsection are performed in the order written.

TABLE OF CONTENTS

SUBSECTION		PAGE
G.1	**Pumping Torus Water to Reactor Building Equipment Drain Discharge Header With RHR Loop A**	31
G.2	**Pumping Torus Water to Reactor Building Equipment Drain Discharge Header With RHR Loop B**	35
G.3	**Raising Torus Water Level With RHR Loop A**	39
G.4	**Raising Torus Water Level With RHR Loop B**	40

CAUTION

Operating both RHR loops in torus cooling should be avoided, unless it is required to maintain torus water temperature less than 95°F, or it has been directed by an AOP, EOP, or test procedure.

G.1 **Pumping Torus Water to Reactor Building Equipment Drain Discharge Header With RHR Loop A**

G.1.1 **IF** water is to be pumped to a radwaste storage tank,
THEN direct radwaste operator to perform the following:

a. Ensure lineup from reactor building equipment drain discharge header to the desired radwaste storage tank is correct.

b. Ensure sufficient radwaste storage tank volume is available.

G.1.2 **IF** water is to be pumped to the main condenser hotwell,
THEN perform the following:

a. Notify RES Radiation Protection.

b. Ensure main condenser vacuum is approximately zero inches Hg (atmospheric pressure).

c. Close 20RDW-101C (reactor building equipment drainage to waste collector isol valve).

d. Open 20RDW-101D (reactor building equipment drainage to condenser hotwell isol valve).

G.1.3 Open RHR pump discharge drain valve:

RHR Pump	RHR Pump Discharge Drain Valve
10P-3A	10RHR-250A
10P-3C	10RHR-250C

CAUTION

Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

G.1.4 Ensure one of the following RHR pumps is running:

- RHR PMP 10P-3A
- RHR PMP 10P-3C

NOTE: 10MOV-34A may be throttled when 10MOV-39A has dual position indication.

G.1.5 Ensure open RHR TEST TORUS CLG & SPRAY 10MOV-39A.

G.1.6 Throttle RHR TEST & TORUS CLG 10MOV-34A to establish desired RHR Loop A flow.

G.1.7 **IF** RHR Loop A condensate transfer keep-full is in service,
AND RHR Loop A pressure is **LESS THAN** condensate transfer pressure,
THEN close 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve).

G.1.8 Open RHR DISCH TO RADW 10MOV-67.

G.1.9 Throttle RHR DISCH TO RADW 10MOV-57 to control flow to reactor building equipment drain discharge header.

G.1.10 **WHILE** pumping torus water, monitor torus water level using multiple indications.

G.1.11 **WHEN** desired torus water level is reached, close RHR DISCH TO RADW 10MOV-57.

G.1.12 Close RHR DISCH TO RADW 10MOV-67.

G.1.13 **IF** RHR Loop A operation is not required, **THEN** shut down RHR Loop A as follows:

a. Ensure one of the RHR Loop A keep-full systems is in service as follows:

- RHR KEEP-FULL PMP 10P-2A is running,

OR

- 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve) is throttled open.

b. Ensure closed RHR TEST & TORUS CLG 10MOV-34A.

c. Ensure the following RHR pumps are stopped:

- RHR PMP 10P-3A
- RHR PMP 10P-3C

d. Ensure closed RHR TEST TORUS CLG & SPRAY 10MOV-39A.

e. Ensure open MIN FLOW VLV 10MOV-16A.

f. Ensure open HX A BYP VLV 10MOV-66A.

g. **IF** RHRSW Loop A operation is not required, **THEN** shut down RHRSW Loop A as follows:

1) Close RHRSW DISCH VLV FROM HX A 10MOV-89A.

2) Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

G.1.14 Close RHR pump discharge drain valve for the RHR pump that was used to pump torus water:

RHR Pump	RHR Pump Discharge Drain Valve
10P-3A	10RHR-250A
10P-3C	10RHR-250C

G.1.15 **IF** the reactor building equipment drain discharge header was lined up to the main condenser hotwell, **THEN** perform the following:

a. Close 20RDW-101D (reactor building equipment drainage to condenser hotwell isol valve).

b. Open 20RDW-101C (reactor building equipment drainage to waste collector isol valve).

CAUTION

Operating both RHR loops in torus cooling should be avoided, unless it is required to maintain torus water temperature less than 95°F, or it has been directed by an AOP, EOP, or test procedure.

G.2 **Pumping Torus Water to Reactor Building Equipment Drain Discharge Header With RHR Loop B**

G.2.1 **IF** water is to be pumped to a radwaste storage tank,
THEN direct radwaste operator to perform the following:

a. Ensure lineup from reactor building equipment drain discharge header to desired radwaste storage tank is correct.

b. Ensure sufficient radwaste storage tank volume is available.

G.2.2 **IF** water is to be pumped to the main condenser hotwell,
THEN perform the following:

a. Notify RES Radiation Protection.

b. Ensure main condenser vacuum is zero inches Hg (atmospheric pressure).

c. Close 20RDW-101C (reactor building equipment drainage to waste collector isol valve).

d. Open 20RDW-101D (reactor building equipment drainage to condenser hotwell isol valve).

G.2.3 Open RHR pump discharge drain valve:

RHR Pump	RHR Pump Discharge Drain Valve
10P-3B	10RHR-250B
10P-3D	10RHR-250D

CAUTION

Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

G.2.4 Ensure one of the following RHR pumps is running:

- RHR PMP 10P-3B
- RHR PMP 10P-3D

NOTE: 10MOV-34B may be throttled when 10MOV-39B has dual position indication.

G.2.5 Ensure open RHR TEST TORUS CLG & SPRAY 10MOV-39B.

G.2.6 Throttle RHR TEST & TORUS CLG 10MOV-34B to establish desired RHR Loop B flow.

G.2.7 **IF** RHR Loop B condensate transfer keep-full is in service,
AND RHR Loop B pressure is **LESS THAN** condensate transfer pressure,
THEN close 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve).

G.2.8 Throttle open 10RHR-253 (RHR pumps B & D disch drains to equip sump isol valve) to control flow to reactor building equipment drain discharge header.

G.2.9 **WHILE** pumping torus water, monitor torus water level using multiple indications.

G.2.10 **WHEN** desired torus water level is reached, close 10RHR-253.

G.2.11 **IF** RHR Loop B operation is not required, **THEN** shut down RHR Loop B as follows:

a. Ensure one of the RHR Loop B keep-full systems is in service as follows:

- RHR KEEP-FULL PMP 10P-2B is running,

OR

- 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve) is throttled open.

b. Ensure closed RHR TEST & TORUS CLG 10MOV-34B.

c. Ensure the following RHR pumps are stopped:

- RHR PMP 10P-3B
- RHR PMP 10P-3D

d. Ensure closed RHR TEST TORUS CLG & SPRAY 10MOV-39B.

e. Ensure open MIN FLOW VLV 10MOV-16B.

f. Ensure open HX B BYP VLV 10MOV-66B.

g. **IF** RHRSW Loop B operation is not required, **THEN** shut down RHRSW Loop B as follows:

1) Close RHRSW DISCH VLV FROM HX B 10MOV-89B.

2) Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

G.2.12 Close RHR pump discharge drain valve for the RHR pump that was used to pump torus water:

RHR Pump	RHR Pump Discharge Drain Valve
10P-3B	10RHR-250B
10P-3D	10RHR-250D

G.2.13 **IF** the reactor building equipment drain discharge header was lined up to the main condenser hotwell, **THEN** perform the following:

a. Close 20RDW-101D (reactor building equipment drainage to condenser hotwell isol valve).

b. Open 20RDW-101C (reactor building equipment drainage to waste collector isol valve).

G.3 **Raising Torus Water Level With RHR Loop A**

G.3.1 **IF** RHR Loop A is in a LPCI standby lineup, **THEN** raise torus water level as follows:

a. Ensure open 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve).

b. Open HX A INBD VENT VLV 10MOV-166A.

c. Open HX A OUTBD VENT VLV 10MOV-167A.

d. **WHILE** raising torus water level, monitor torus water level using multiple indications.

e. **WHEN** desired torus water level is reached, close the following valves:

- HX A INBD VENT VLV 10MOV-166A
- HX A OUTBD VENT VLV 10MOV-167A

f. **IF** RHR KEEP-FULL PMP 10P-2A is running, **THEN** close 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve).

G.3.2 **IF** RHR Loop A flow is being returned to the torus, **THEN** raise torus water level as follows:

a. Open 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve).

b. Raise RHR Loop A flow until RHR Loop A pressure is **LESS THAN** condensate transfer pressure.

c. **WHEN** desired torus water level is reached, perform the following:

1) **IF** RHR KEEP-FULL PMP 10P-2A is running, **THEN** close 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve).

2) **IF** RHR KEEP-FULL PMP 10P-2A is stopped, **AND** RHR Loop A pressure is **LESS THAN** condensate transfer pressure, **THEN** close 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve).

G.4 **Raising Torus Water Level With RHR Loop B**

G.4.1 **IF** RHR Loop B is in a LPCI standby lineup, **THEN** raise torus water level as follows:

a. Open 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve).

b. Open HX B INBD VENT VLV 10MOV-166B.

c. Open HX B OUTBD VENT VLV 10MOV-167B.

d. **WHILE** raising torus water level, monitor torus water level using multiple indications.

e. **WHEN** desired torus water level is reached, close the following valves:

- HX B INBD VENT VLV 10MOV-166B
- HX B OUTBD VENT VLV 10MOV-167B

f. **IF** RHR KEEP-FULL PMP 10P-2B is running, **THEN** close 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve).

G.4.2 **IF** RHR Loop B flow is being returned to the torus, **THEN** raise torus water level as follows:

a. Open 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve).

b. Raise RHR Loop B flow until RHR Loop B pressure is **LESS THAN** condensate transfer pressure.

c. **WHEN** desired torus water level is reached, perform the following:

1) **IF** RHR KEEP-FULL PMP 10P-2B is running, **THEN** close 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve).

2) **IF** RHR KEEP-FULL PMP 10P-2B is stopped, **AND** RHR Loop B pressure is **LESS THAN** condensate transfer pressure, **THEN** close 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve).

H. **ATTACHMENTS**

1. REFERENCES
2. TORUS COOLING STARTUP (POSTED)
3. TORUS SPRAY STARTUP (POSTED)
4. DRYWELL SPRAY STARTUP (POSTED)
5. RHRSW STARTUP AND TEMPERATURE CONTROL (POSTED)
6. CONTAINMENT COOLING/SPRAY SHUTDOWN (POSTED)

ATTACHMENT 1

REFERENCES

1.0 **Performance References**

1.1 OP-13, Residual Heat Removal System*

1.2 OP-13C, RHR Service Water*

2.0 **Developmental References**

2.1 OP-13, Revision 74, Residual Heat Removal System*

2.2 FSAR Section 4.8

2.3 System Flow Diagrams:

- FM-20A Flow Diagram Residual Heat Removal Sheet 1
- FM-20B Flow Diagram Residual Heat Removal Sheet 2

2.4 JSEM-91-082, QA Classification of 10MOV-57 and 10MOV-67

2.5 Evaluation Of 10MOV-16A/B As Normally Open Valve

2.6 JSEM-92-047, RHR Pump Suction Valves

2.7 JAF-CALC-ELEC-00589, Rev 0, Removal of LPCI Inverters From EQ Program

2.8 JPEM-92-029, Revise OP-13 to state that operation of any single RHR pump at less than 6500 gpm or operation of any two RHR pumps in parallel at flows less 13,000 gpm should be avoided. ACTS Item 5857

2.9 IE Information Notice No. 87-10, Potential for Water Hammer During Restart of Residual Heat Removal Pumps

2.10 JAF-SE-95-063, Operation of Four RHR Pumps in Torus Cooling Flow Mode

TORUS COOLING STARTUP

1. **IF** a LPCI auto-initiation signal is sealed in,
   THEN perform the following:

   a. **IF** RPV water level is **LESS THAN** zero inches on fuel zone water level indication,
      AND the EOPs permit diverting LPCI flow,
      THEN place DW & TORUS SPRAY VLV OVERRIDE OF FUEL ZONE LVL 10A-S18A(B) keylock switch in MANUAL OVERRD.

   b. Place SPRAY CNTRL 10A-S17A(B) switch to MANUAL, spring return to normal.

   c. Verify white SPRAY PERM 10A-DS67A(B) light is on.

> **CAUTION**
>
> Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.
>
> Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

NOTE: Both RHR pumps are required to be running for maximum torus cooling.

2. Ensure at least one of the following RHR pumps is running:

   - RHR PMP 10P-3A(B)
   - RHR PMP 10P-3C(D)

NOTE: 10MOV-34A(B) may be throttled when 10MOV-39A(B) has dual position indication.

3. Open RHR TEST TORUS CLG & SPRAY 10MOV-39A(B).

4. Throttle RHR TEST & TORUS CLG 10MOV-34A(B) to establish desired flow.

5. **WHEN** RHR Loop A(B) flow is **GREATER THAN** 1100 gpm, ensure closed MIN FLOW VLV 10MOV-16A(B).

NOTE: Procedure may be continued while condensate transfer is being isolated.

6. **IF** RHR Loop A(B) condensate transfer keep-full is in service,
   AND RHR Loop A(B) pressure is **LESS THAN** condensate transfer pressure,
   THEN close 10RHR-274(10RHR-260).

7. Establish RHRSW flow and temperature control.

TORUS SPRAY STARTUP

1. Verify torus pressure is **GREATER THAN** 2.7 psig.

2. **IF** RPV water level is **LESS THAN** zero inches on fuel zone water level indication,
   AND the EOPs permit diverting LPCI flow,
   THEN place DW & TORUS SPRAY VLV OVERRIDE OF FUEL ZONE LVL 10A-S18A(B) keylock switch in MANUAL OVERRD.

3. Place SPRAY CNTRL 10A-S17A(B) switch to MANUAL, spring return to normal.

4. Verify white SPRAY PERM 10A-DS67A(B) light is on.

CAUTION

Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

5. Ensure available RHR pumps in RHR Loop A(B) are running:

   - RHR PMP 10P-3A(B)
   - RHR PMP 10P-3C(D)

NOTE: 10MOV-38A(B) may be throttled when 10MOV-39A(B) has dual position indication.

6. Open RHR TEST TORUS CLG & SPRAY 10MOV-39A(B).

7. Throttle TORUS SPRAY INBD VLV 10MOV-38A(B) to establish desired torus spray flow rate.

8. **WHEN** RHR Loop A(B) flow is **GREATER THAN** 1100 gpm, ensure closed MIN FLOW VLV 10MOV-16A(B).

NOTE: Procedure may be continued while condensate transfer is being isolated.

9. **IF** RHR Loop A(B) condensate transfer keep-full is in service,
   AND RHR Loop A(B) pressure is **LESS THAN** condensate transfer pressure,
   THEN close 10RHR-274(10RHR-260).

10. Establish RHRSW flow and temperature control.

POSTED ATTACHMENT
(Location: 09-3 panel-Control Room and Simulator)

DRYWELL SPRAY STARTUP

1. Ensure the following components are tripped:

RWR Pumps	Drywell Cooling Fans
• 02-2P-1A	• 68FN-2A, B, C, and D
• 02-2P-1B	• 68FN-4A, B, C, and D

2. Verify drywell temperature and pressure are within the Drywell Spray Initiation Limit.

3. **IF** RPV water level is **LESS THAN** zero inches on fuel zone water level indication,
AND the EOPs permit diverting LPCI flow,
THEN place DW & TORUS SPRAY VLV OVERRIDE OF FUEL ZONE LVL 10A-S18A(B) keylock switch in MANUAL OVERRD.

4. Place SPRAY CNTRL 10A-S17A(B) switch to MANUAL, spring return to normal.

5. Verify white SPRAY PERM 10A-DS67A(B) light is on.

CAUTION

Starting an RHR pump in an RHR loop that is not full could result in severe water hammer and equipment damage. RHR loop piping shall be full prior to manually starting an RHR pump in that loop.

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

6. Ensure available RHR pumps in RHR Loop A(B) are running:
   - RHR PMP 10P-3A(B)
   - RHR PMP 10P-3C(D)

7. Open DW SPRAY OUTBD VLV 10MOV-26A(B).

8. Throttle DW SPRAY INBD VLV 10MOV-31A(B) to establish desired drywell spray flow rate.

9. **WHEN** RHR Loop A(B) flow is **GREATER THAN** 1100 gpm, ensure closed MIN FLOW VLV 10MOV-16A(B).

NOTE: Procedure may be continued while condensate transfer is being isolated.

10. **IF** RHR Loop A(B) condensate transfer keep-full is in service,
AND RHR Loop A(B) pressure is **LESS THAN** condensate transfer pressure,
THEN close 10RHR-274(10RHR-260).

11. Establish RHRSW flow and temperature control.

RHRSW STARTUP AND TEMPERATURE CONTROL

1. Establish RHRSW flow as follows:

   a. Verify one of the following alarms is clear:

      - Annunciator 09-4-3-4 RHRSW A OR B DISCH LINE NOT FULL
      - EPIC-D-134 10LS-105A(EPIC-D-135 10LS-105B)

   NOTE: Both RHRSW pumps are required to be running for maximum torus cooling.

   b. Start at least one of the following RHRSW pumps:

      - RHRSW PMP 10P-1A(B)
      - RHRSW PMP 10P-1C(D)

CAUTION

Running an RHRSW pump with flow less than 2500 gpm or greater than 4000 gpm per pump could damage pump.

   c. Throttle RHRSW DISCH VLV FROM HX A(B) 10MOV-89A(B) to establish 2500 to 4000 gpm per RHRSW pump.

   d. **IF** drywell or torus sprays are in service, **THEN** establish 4000 gpm per RHRSW pump.

NOTE: If a LPCI auto-initiation signal is present, 10MOV-66A(B) will have a sealed in open signal for 3 minutes from when LPCI auto-initiated.

2. Close HX A(B) BYP VLV 10MOV-66A(B).

3. **WHILE** performing the following step, maintain torus water temperature **GREATER THAN** 68°F.

4. Control containment heat removal rate using one or a combination of the following methods:

   - Throttle applicable loop throttle valve:

     - RHR TEST & TORUS CLG 10MOV-34A(B).
     - TORUS SPRAY INBD VLV 10MOV-38A(B)
     - DW SPRAY INBD VLV 10MOV-31A(B)

   - Start or stop second RHR pump in RHR Loop A(B).

   - Throttle RHRSW DISCH VLV FROM HX A(B) 10MOV-89A(B).

   - Start or stop second RHRSW pump in RHRSW Loop A(B).

   - Shut down RHRSW Loop A(B) per Section F of OP-13C.

CONTAINMENT COOLING/SPRAY SHUTDOWN

CAUTION

Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

1. **IF** securing torus cooling,
   THEN close RHR TEST & TORUS CLG 10MOV-34A(B).

2. **IF** securing torus spray,
   THEN close TORUS SPRAY INBD VLV 10MOV-38A(B).

3. **IF** securing drywell spray,
   THEN close DW SPRAY INBD VLV 10MOV-31A(B).

4. **IF** RHR Loop A(B) flow is **LESS THAN** 500 gpm,
   THEN ensure open MIN FLOW VLV 10MOV-16A(B).

5. **IF** RHR Loop A(B) operation is not required,
   THEN shut down RHR Loop A(B) per Subsection F.7(F.8) of OP-13B.

NEW YORK POWER AUTHORITY
JAMES A. FITZPATRICK NUCLEAR POWER PLANT
OPERATING PROCEDURE

RHR SERVICE WATER*
OP-13C
REVISION 3

APPROVED BY: _____ DATE 1/13/00
RESPONSIBLE PROCEDURE OWNER

EFFECTIVE DATE: January 21, 2000

FIRST ISSUE ☐ FULL REVISION ☐ LIMITED REVISION ☒

REFERENCE USE	TSR
TECHNICAL	

FOR INFORMATION ONLY

REVISION SUMMARY SHEET

REV. NO.	CHANGE AND REASON FOR CHANGE
3	Added note before Steps D.1.2 and D.2.2 to indicate that EPIC strainer dP alarms are expected with both RHRSW pumps running. (PCR #6 dated 11/15/99)
	Added note and caution before Steps G.1.5 and G.1.6 to indicate that strainer pointer might not exactly align with basket position, and that use of excessive force could damage strainer. (PCR dated 10/21/99)
2	Updated cover page due to periodic review.
	Added guidance to Subsections G.1 and G.2 for placing strainers in service following maintenance. Ensures proper filling and venting. (pcr#2 dated 6/28/96)
	Added guidance to Subsections G.1 and G.2 for strainer baskets that fail to isolate. The basket that is capable of being isolated is the one to be maintained in service to maintain the ability to clean a clogged basket.
	Changed order of Steps G.1.3/G.2.3 and G.1.4/G.2.4. Improved clarity and human factors.
1	Added Step G.3.2 and G.4.2 to throttle open two turns from the closed seat the discharge valve that was closed in the previous step. This valve is difficult to open from the full closed position when the RHRSW pump is running. Opening the valve several turns will reduce the differential pressure across the valve, making it easier to open the valve. This change incorporates PCR #1 dated 3/18/95.
	Revised note above Step G.4.1 to state that 10P-1D is the preferred pump to be used for filling RHRSW Loop B because it is in better mechanical condition than RHRSW Pump 10P-1B. 10P-1D was changed out recently, and 10P-1B has WRs 95-03776-00 and 95-03776-04 against it for degraded pump performance.

TABLE OF CONTENTS

SECTION		PAGE
A.	**REQUIREMENTS**	4
B.	**SYSTEM DESCRIPTION**	4
C.	**PLANT OPERATING REQUIREMENTS**	4
D.	**STARTUP**	5
E.	**NORMAL OPERATION**	8
F.	**SHUTDOWN**	9
G.	**SPECIAL PROCEDURES**	10
H.	**ATTACHMENTS**	23
	1. REFERENCES	24

A. REQUIREMENTS

A.1 Technical Specifications

Sections 3.5.A, 3.5.B, 3.5.F, and 3.5.G

A.2 Commitments

None

A.3 Validation

Procedure developed from OP-13, Revision 74, which was validated per ODSO-35.

B. SYSTEM DESCRIPTION

Refer to Section B of OP-13.

C. PLANT OPERATING REQUIREMENTS

C.1 Prerequisites

RHR System valves, power supplies, and support systems are lined up per OP-13, with exceptions approved by Shift Manager.

C.2 Precautions

Racking out an RHRSW pump breaker de-energizes the pump motor bearing cooling water SOV and will drain the RHRSW header. To prevent draining the RHRSW header, the bearing cooling water isolation valve shall be closed before racking out pump breaker.

D. **STARTUP**

NOTE: Subsections D.1 and D.2 may be performed independently or concurrently.

TABLE OF CONTENTS

SUBSECTION		PAGE
D.1	**RHR Service Water Loop A Startup**	6
D.2	**RHR Service Water Loop B Startup**	7

D.1 **RHR Service Water Loop A Startup**

D.1.1 Verify one of the following alarms is clear:

- Annunciator 09-4-3-4 RHRSW A OR B DISCH LINE NOT FULL
- EPIC-D-134 10LS-105A

NOTE 1: Both RHRSW pumps are required to be running for maximum cooling.

NOTE 2: EPIC-D-127 RHRSW STRAINER S5A DIFF PRESS is expected to alarm with both RHRSW pumps running. This is acceptable provided that desired RHRSW loop flow rate can be maintained. WR-97-08136-02 tracks an evaluation of the alarm setpoint.

D.1.2 Start at least one of the following RHRSW pumps:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

CAUTION

Running an RHRSW pump with flow less than 2500 gpm or greater than 4000 gpm per pump could damage pump.

D.1.3 Throttle RHRSW DISCH VLV FROM HX A 10MOV-89A to establish 2500 to 4000 gpm per RHRSW pump.

D.2 **RHR Service Water Loop B Startup**

D.2.1 Verify one of the following alarms is clear:

- Annunciator 09-4-3-4 RHRSW A OR B DISCH LINE NOT FULL
- EPIC-D-135 10LS-105B

NOTE 1: Both RHRSW pumps are required to be running for maximum cooling.

NOTE 2: EPIC-D-128 RHRSW STRAINER S5B DIFF PRESS is expected to alarm with both RHRSW pumps running. This is acceptable provided that desired RHRSW loop flow rate can be maintained. WR-97-08136-02 tracks an evaluation of the alarm setpoint.

D.2.2 Start at least one of the following RHRSW pumps:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

CAUTION

Running an RHRSW pump with flow less than 2500 gpm or greater than 4000 gpm per pump could damage pump.

D.2.3 Throttle RHRSW DISCH VLV FROM HX B 10MOV-89B to establish 2500 to 4000 gpm per RHRSW pump.

E. **NORMAL OPERATION**

Refer to Section E of OP-13.

F. **SHUTDOWN**

NOTE: Subsections F.1 and F.2 may be performed independently or concurrently.

F.1 **RHR Service Water Loop A Shutdown**

F.1.1 Close RHRSW DISCH VLV FROM HX A 10MOV-89A.

F.1.2 Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

F.2 **RHR Service Water Loop B Shutdown**

F.2.1 Close RHRSW DISCH VLV FROM HX B 10MOV-89B.

F.2.2 Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

G. SPECIAL PROCEDURES

NOTE: Subsections G.1 through G.4 are independent procedures and may be performed in any order, provided steps within each subsection are performed in the order written.

TABLE OF CONTENTS

SUBSECTION		PAGE
G.1	**Shifting Baskets in Twin Basket Strainer 10S-5A** . . .	11
G.2	**Shifting Baskets in Twin Basket Strainer 10S-5B** . . .	15
G.3	**Filling RHR Service Water Loop A With an RHRSW Pump** .	19
G.4	**Filling RHR Service Water Loop B With an RHRSW Pump** .	21

G.1 **Shifting Baskets in Twin Basket Strainer 10S-5A**

NOTE: Possible strainer basket positions are as follows:

- NO.1 BASKET
- BOTH
- NO.2 BASKET

G.1.1 Determine which basket is in service by observing position indicator plates on 10S-5A.

G.1.2 Following maintenance on strainers, the following steps should be adhered to while performing the remainder of this subsection:

a. **IF** work was performed on only one strainer, **THEN** perform the following:

1) Ensure the non-worked strainer is placed in service.

2) Shift strainers to place the repaired/cleaned strainer in service.

b. **IF** work was performed on both strainers, **THEN** gradually pressurize each strainer simultaneously.

G.1.3 **IF** RHRSW Pump 10P-1A and C are shutdown, **THEN** perform the following:

a. Open 10RHR-111A (RHRSW pumps A & C disch strainer A equalizing valve).

b. Open petcock vent valve on top cover plate for the strainer that is to be placed in service.

c. **WHEN** solid stream of water flows from petcock vent valve, close petcock vent valve.

G.1.4 **IF** RHRSW Pump 10P-1A or C is running, **THEN** perform the following:

a. **IF** only NO.1 Basket is in service, **THEN** flush loose debris from NO.1 Basket as follows:

1) Open 10RHR-758A (RHRSW pumps A & C disch strainer A drain valve).

2) **WHEN** water flowing from drain is free of debris, close 10RHR-758A.

b. **IF** only NO.2 Basket is in service, **THEN** flush loose debris from NO.2 Basket as follows:

1) Open 10RHR-757A (RHRSW pumps A & C disch strainer A drain valve).

2) **WHEN** water flowing from drain is free of debris, close 10RHR-757A.

c. **IF** BOTH baskets are in service, **THEN** flush loose debris from both baskets as follows:

1) Open the following valves:

- 10RHR-757A (RHRSW pumps A & C disch strainer A drain valve)
- 10RHR-758A (RHRSW pumps A & C disch strainer A drain valve)

2) **WHEN** water flowing from drains is free of debris, close the following valves:

- 10RHR-757A
- 10RHR-758A

d. Open 10RHR-111A (RHRSW pumps A & C disch strainer A equalizing valve).

e. Open petcock vent valve on top cover plate for the strainer that is to be placed in service.

f. **WHEN** solid stream of water flows from petcock vent valve, close petcock vent valve.

CAUTION

Placing both baskets in service at the same time could result in clogging both strainer baskets and a subsequent loss of RHRSW flow.

G.1.5 Shift strainer baskets as follows:

NOTE: Pointer is used as a guide; pointer might not exactly align with basket position.

CAUTION

Use of excessive force to align pointer could damage strainer.

a. **IF** shifting from BOTH **OR** NO.2 BASKET to NO.1 BASKET,
THEN rotate handwheel clockwise until pointer is lined up with NO.1 BASKET.

b. **IF** shifting from NO.2 BASKET to BOTH,
THEN rotate handwheel clockwise until pointer is lined up with BOTH.

c. **IF** shifting from BOTH **OR** NO.1 BASKET to NO.2 BASKET,
THEN rotate handwheel counterclockwise until pointer is lined up with NO.2 BASKET.

d. **IF** shifting from NO.1 BASKET to BOTH,
THEN rotate handwheel counterclockwise until pointer is lined up with BOTH.

G.1.6 Ensure closed 10RHR-111A (RHRSW pumps A & C disch strainer A equalizing valve).

G.1.7 **IF** maintenance is required on the out-of-service strainer basket,
THEN perform the following:

a. Open petcock vent valve on top cover plate for the out-of-service strainer basket.

b. Open drain valve on the out-of-service strainer basket.

Basket Number	Drain Valve
NO.1	10RHR-758A
NO.2	10RHR-757A

G.1.8 **IF** a strainer basket fails to isolate,
THEN perform the following:

a. Ensure the basket that is capable of being isolated is maintained in service.

b. Special condition tag the basket selector to indicate, "NO.(1 or 2) Basket is to be maintained in service to maintain the ability to clean a clogged basket strainer."

G.1.9 **IF** strainer baskets were shifted due to high strainer differential pressure (**GREATER THAN** 15 psid),
THEN direct Performance Engineering to determine if high differential pressure is due to high flow rates or strainer basket fouling, before opening basket covers for inspection.

G.2 **Shifting Baskets in Twin Basket Strainer 10S-5B**

NOTE: Possible strainer basket positions are as follows:

- NO.1 BASKET
- BOTH
- NO.2 BASKET

G.2.1 Determine which basket is in service by observing position indicator plates on 10S-5B.

G.2.2 Following maintenance on strainers, the following steps should be adhered to while performing the remainder of this subsection:

a. **IF** work was performed on only one strainer, **THEN** perform the following:

1) Ensure the non-worked strainer is placed in service.

2) Shift strainers to place the repaired/cleaned strainer in service.

b. **IF** work was performed on both strainers, **THEN** gradually pressurize each strainer simultaneously.

G.2.3 **IF** RHRSW Pump 10P-1B and D are shutdown, **THEN** perform the following:

a. Open 10RHR-111B (RHRSW pumps B & D disch strainer B equalizing valve).

b. Open petcock vent valve on top cover plate for the strainer that is to be placed in service.

c. **WHEN** solid stream of water flows from petcock vent valve, close petcock vent valve.

G.2.4 **IF** RHRSW Pump 10P-1B or D is running, **THEN** perform the following:

a. **IF** only NO.1 Basket is in service, **THEN** flush loose debris from NO.1 Basket as follows:

1) Open 10RHR-758B (RHRSW pumps B & D disch strainer B drain valve).

2) **WHEN** water flowing from drain is free of debris, close 10RHR-758B.

b. **IF** only NO.2 Basket is in service, **THEN** flush loose debris from NO.2 Basket as follows:

1) Open 10RHR-757B (RHRSW pumps B & D disch strainer B drain valve).

2) **WHEN** water flowing from drain is free of debris, close 10RHR-757B.

c. **IF** BOTH baskets are in service, **THEN** flush loose debris from both baskets as follows:

1) Open the following valves:

- 10RHR-757B (RHRSW pumps B & D disch strainer B drain valve)
- 10RHR-758B (RHRSW pumps B & D disch strainer B drain valve)

2) **WHEN** water flowing from drains is free of debris, close the following valves:

- 10RHR-757B
- 10RHR-758B

d. Open 10RHR-111B (RHRSW pumps B & D disch strainer B equalizing valve).

e. Open petcock vent valve on top cover plate for the strainer that is to be placed in service.

f. **WHEN** solid stream of water flows from petcock vent valve, close petcock vent valve.

CAUTION

Placing both baskets in service at the same time could clog strainer and reduce RHRSW flow.

G.2.5 Shift strainer baskets as follows:

NOTE: Pointer is used as a guide; pointer might not exactly align with basket position.

CAUTION

Use of excessive force to align pointer could damage strainer.

a. **IF** shifting from BOTH **OR** NO.2 BASKET to NO.1 BASKET,
THEN rotate handwheel clockwise until pointer is lined up with NO.1 BASKET.

b. **IF** shifting from NO.2 BASKET to BOTH,
THEN rotate handwheel clockwise until pointer is lined up with BOTH.

c. **IF** shifting from BOTH **OR** NO.1 BASKET to NO.2 BASKET,
THEN rotate handwheel counterclockwise until pointer is lined up with NO.2 BASKET.

d. **IF** shifting from NO.1 BASKET to BOTH,
THEN rotate handwheel counterclockwise until pointer is lined up with BOTH.

G.2.6 Ensure closed 10RHR-111B (RHRSW pumps B & D disch strainer B equalizing valve).

G.2.7 **IF** maintenance is required on the out-of-service strainer basket,
THEN perform the following:

a. Open petcock vent valve on top cover plate for the out-of-service strainer basket.

b. Open drain valve on the out-of-service strainer basket.

Basket Number	Drain Valve
NO.1	10RHR-758B
NO.2	10RHR-757B

G.2.8 **IF** a strainer basket fails to isolate,
THEN perform the following:

a. Ensure the basket that is capable of being isolated is maintained in service.

b. Special condition tag the basket selector to indicate, "NO.(1 or 2) Basket is to be maintained in service to maintain the ability to clean a clogged basket strainer."

G.2.9 **IF** strainer baskets were shifted due to high strainer differential pressure (**GREATER THAN** 15 psid),
THEN direct Performance Engineering to determine if high differential pressure is due to high flow rates or strainer basket fouling, before opening basket covers for inspection.

G.3 **Filling RHR Service Water Loop A With an RHRSW Pump**

NOTE: RHRSW Pump 10P-1C is the preferred pump to be used for filling RHRSW Loop A because it is in better mechanical condition than RHRSW Pump 10P-1A.

G.3.1 Close the RHRSW pump discharge valve for the pump to be used to fill RHR Service Water Loop A:

- 10RHR-22A (RHRSW pump A disch isol valve)

OR

- 10RHR-22C (RHRSW pump C disch isol valve)

G.3.2 Throttle open two turns off closed seat the RHRSW pump discharge valve that was closed in Step G.3.1.

G.3.3 Perform Steps G.3.4 through G.3.7 without delay between steps to minimize time that the RHRSW pump is operated at low flow.

G.3.4 Start one of the following RHRSW pumps:

- RHRSW PMP 10P-1A

OR

- RHRSW PMP 10P-1C

G.3.5 Throttle open RHRSW DISCH VLV FROM HX A 10MOV-89A to establish dual position indication.

G.3.6 Slowly open the RHRSW pump discharge valve for the pump to be used to fill RHR Service Water Loop A:

- 10RHR-22A (RHRSW pump A disch isol valve)

OR

- 10RHR-22C (RHRSW pump C disch isol valve)

NOTE: RHRSW Loop A flow rate indication could be unavailable until I&C vents the flow transmitter.

G.3.7 Throttle RHRSW DISCH VLV FROM HX A 10MOV-89A to establish one or both of the following conditions:

- RHRSW Loop A flow rate 2500 to 4000 gpm, if flow indication is available
- RHRSW pump motor current slightly **LESS THAN** max normal amps

G.3.8 **IF** operation of both RHRSW pumps is desired, **THEN** perform the following:

a. Start the other RHRSW pump:

- RHRSW PMP 10P-1A

OR

- RHRSW PMP 10P-1C

b. Throttle RHRSW DISCH VLV FROM HX A 10MOV-89A to establish one or both of the following conditions:

- RHRSW Loop A flow rate 5000 to 8000 gpm, if flow indication is available
- RHRSW Pump 10P-1A and RHRSW Pump 10P-1C motor current slightly **LESS THAN** max normal amps

G.3.9 Vent RHRSW A Line Level Switch 10LS-105A per Section G of OP-13E.

G.3.10 **IF** RHRSW Loop A operation <u>is not</u> desired, **THEN** shut down RHR Service Water Loop A per Subsection F.1.

G.4 **Filling RHR Service Water Loop B With an RHRSW Pump**

NOTE: RHRSW Pump 10P-1D is the preferred pump to be used for filling RHRSW Loop B because it is in better mechanical condition than RHRSW Pump 10P-1B.

G.4.1 Close the RHRSW pump discharge valve for the pump to be used to fill RHR Service Water Loop B:

- 10RHR-22B (RHRSW pump B disch isol valve)

OR

- 10RHR-22D (RHRSW pump D disch isol valve)

G.4.2 Throttle open two turns off closed seat the RHRSW pump discharge valve that was closed in Step G.4.1.

G.4.3 Perform Steps G.4.4 through G.4.7 without delay between steps to minimize time that the RHRSW pump is operated at low flow.

G.4.4 Start one of the following RHRSW pumps:

- RHRSW PMP 10P-1B

OR

- RHRSW PMP 10P-1D

G.4.5 Throttle open RHRSW DISCH VLV FROM HX B 10MOV-89B to establish dual position indication.

G.4.6 Slowly open the RHRSW pump discharge valve for the pump to be used to fill RHR Service Water Loop B:

- 10RHR-22B (RHRSW pump B disch isol valve)

OR

- 10RHR-22D (RHRSW pump D disch isol valve)

NOTE: RHRSW Loop B flow rate indication could be unavailable until I&C vents the flow transmitter.

G.4.7 Throttle RHRSW DISCH VLV FROM HX B 10MOV-89B to establish one or both of the following conditions:

- RHRSW Loop B flow rate 2500 to 4000 gpm, if flow indication is available
- RHRSW pump motor current slightly **LESS THAN** max normal amps

G.4.8 **IF** operation of both RHRSW pumps is desired, **THEN** perform the following:

a. Start the other RHRSW pump:

- RHRSW PMP 10P-1B

OR

- RHRSW PMP 10P-1D

b. Throttle RHRSW DISCH VLV FROM HX B 10MOV-89B to establish one or both of the following conditions:

- RHRSW Loop B flow rate 5000 to 8000 gpm, if flow indication is available
- RHRSW Pump 10P-1B and RHRSW Pump 10P-1D motor current slightly **LESS THAN** max normal amps

G.4.9 Vent RHRSW B Line Level Switch 10LS-105B per Section G of OP-13E.

G.4.10 **IF** RHRSW Loop B operation <u>is not</u> desired, **THEN** shut down RHR Service Water Loop B per Subsection F.2.

H. **ATTACHMENTS**

1. REFERENCES

ATTACHMENT 1

REFERENCES

1.0 **Performance References**

1.1 OP-13, Residual Heat Removal System*

1.2 OP-13E, RHR - Keep-Full*

2.0 **Developmental References**

2.1 OP-13, Revision 74, Residual Heat Removal System*

2.2 FSAR Section 4.8

2.3 System Flow Diagrams:

- FM-20A Flow Diagram Residual Heat Removal Sheet 1
- FM-20B Flow Diagram Residual Heat Removal Sheet 2

2.4 JSEM-92-109, Revise annunciator response procedures to have performance engineering monitor RHRSW flow to verify basket fouling when a high strainer dP signal is received. ACTS Item 6256.

2.5 JTS-93-0514, Lack of RHRSW and ESW Keepfull During the Upcoming Service Water Outage

2.6 JTS-93-0279, ESW Strainer 46STR-5A1 Functionality Due to Failure to Meet ST-8R Test Requirements

2.7 JPLN-96-179, A RHR/RHRSW LCO Critique Recommendation Resolutions

NEW YORK POWER AUTHORITY
JAMES A. FITZPATRICK NUCLEAR POWER PLANT
OPERATING PROCEDURE

RHR - SHUTDOWN COOLING*
OP-13D
REVISION 12

APPROVED BY: Thos. A. Plunge Jr. (Acting) DATE 5/30/00
RESPONSIBLE PROCEDURE OWNER

EFFECTIVE DATE: 5-31-00

FIRST ISSUE ☐ FULL REVISION ☐ LIMITED REVISION ■

REFERENCE USE	TSR
TECHNICAL	

FOR INFORMATION ONLY

REVISION SUMMARY SHEET

REV. NO.	CHANGE AND REASON FOR CHANGE
12	Incorporated EC dated 10/18/99 to correct step cross-reference in notes on pages 33 and 40. Also relocated the notes to just before Steps D.5.22 and D.6.22. Added Steps A.3.8, D.1.3, and D.2.3 to ensure continuous vent lineup is secured before establishing shutdown cooling lineup. (DER-00-01306, ACT-00-49763) Revised Steps D.1.7.c, D.1.8.c, D.2.7.c, D.2.8.c, D.5.3.c, D.5.4.c, D.6.3.c, and D.6.4.c to ensure 10MOV-16A/B and 10MOV-39A/B handwheels are protective tagged closed. This change clarifies the intent of these steps which is to tag the valves locally. (PCR #16 dated 5/8/00) Added Steps F.1.11 and F.2.11 to verify DHR System is capable of assuming the spent fuel pool and core decay heat load before rendering RHR shutdown cooling unavailable. (JAF-SE-96-042, ACT-98-36998)
11	Updated cover page due to periodic review. Revised Subsection D.1 and D.2 to allow omitting loop flush and venting and delay PTR hanging when the expected recovery time exceeds the time to boil. (pcr#14 dated 10/21/98) Incorporated TC dated 11/6/98 which added valves 10RHR-440 and 441 per Mod M1-97-021. Affects Subsection G.1. Incorporated EC dated 11/28/98 to correct typographical error. Added local valve as a component required to be closed and tagged for SDC. Affects Steps D.1.6, D.1.7, D.2.6, and D.2.7. Valves could have been omitted on PTR. (pcr#15 dated 12/9/98) Deleted requirement to obtain initial chemistry sample from Subsections D.1, D.2, D.5, and D.6. Initial sample results have normally resulted in the need to flush loop. (pcr#13 dated 10/6/98)

TABLE OF CONTENTS

SECTION		PAGE
A.	**REQUIREMENTS**	4
B.	**SYSTEM DESCRIPTION**	5
C.	**PLANT OPERATING REQUIREMENTS**	5
D.	**STARTUP**	7
E.	**NORMAL OPERATION**	42
F.	**SHUTDOWN**	44
G.	**SPECIAL PROCEDURES**	55
H.	**ATTACHMENTS**	75
	1. REFERENCES	76
	2. RPV WATER LEVEL RESPONSE DUE TO RHR OR RWR PUMP STARTUP	78

A. **REQUIREMENTS**

A.1 **Technical Specifications**

Sections 3.5.A, 3.5.B, 3.5.F, and 3.5.G

A.2 **Commitments**

A.2.1 LER-93-005, ACTS Item 7265, revise OP-13 to fill shutdown cooling suction piping by cycling 10MOV-17 and 18.

A.2.2 LER-93-012, ACTS Item 8142, ACTS Item 8143, revise OP-13 to require shutdown cooling piping be filled and vented using the condensate transfer system prior to opening 10MOV-17 and 10MOV-18 any time shutdown cooling is out of service for greater than 24 hours

A.3 **Expectations**

A.3.1 JPEM-92-029, ACTS Item 5857, revise OP-13 to state that operation of any single RHR pump at less than 6500 gpm or operation of any two RHR pumps in parallel at flows less 13,000 gpm should be avoided.

A.3.2 JSEM-92-047, ACTS Item 4476, update OP-13 to require isolation of 10MOV-15s each time SDC is secured.

A.3.3 SER 87-11, OE/VI 870081, Water Hammer in the Component Cooling System

A.3.4 ACTS Item 13988, revise OP-13 to open 10MOV-16 valves prior to opening 10MOV-13 valves to go into SDC.

A.3.5 JTS-97-0031, Revision 1, ACTS Item 24604, revised procedure to require a minimum vent time of 30 minutes for venting shutdown cooling suction piping.

A.3.6 DER-96-1504, ACTS Item 23524, SIL 406, Rev. 1, Added steps to minimize flow induced vibration of in-core detectors when detector is not fully supported on all sides by a fuel assembly or blade guide.

A.3.7 JAF-SE-96-042, ACT-98-36998, Added steps to verify DHR System is capable of assuming the spent fuel pool and core decay heat load before rendering RHR shutdown cooling unavailable.

A.3.8 DER-00-01306, ACT-00-49763, Revised OP-13D to provide additional link to ensure continuous vent lineup is secured before establishing shutdown cooling lineup.

A.4 **Validation**

Procedure developed from OP-13, Revision 74, which was validated per ODSO-35.

B. **SYSTEM DESCRIPTION**

Refer to Section B of OP-13.

C. **PLANT OPERATING REQUIREMENTS**

C.1 **Prerequisites**

RHR System valves, power supplies, and support systems are lined up per OP-13, with exceptions approved by Shift Manager.

C.2 **Precautions**

C.2.1 RES Radiation Protection shall be notified before:

- RHR System piping is drained
- RHR System is started up in any mode of operation, unless an emergency exists

C.2.2 An extremely large drain path exists from the RPV to the torus if either of the following conditions occur:

- 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the associated 10MOV-13A, B, C, or D are open at the same time.
- 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D are open and the associated RHR loop torus spray or torus cooling motor operated isolation valves are open.

The following valves are interlocked to prevent establishing this drain path:

- 10MOV-15A, B, C, and D
- 10MOV-13A, B, C, and D
- 10MOV-39A and B

These interlocks shall not be defeated unless system boundaries are isolated.

C.2.3 The RHR System shall not be filled by opening Torus Suction Valve 10MOV-13A, B, C, or D or Shutdown Cooling Suction Valve 10MOV-15A, B, C, or D.

EXPA.3.1
C.2.4 Operation of any single RHR pump at flows less than 6500 gpm or operation of any two RHR pumps in parallel at flows less than 13000 gpm should be minimized to prevent high pump vibration.

C.2.5 Running an RHR pump for greater than 10 minutes with only its minimum flow valve open could damage the pump.

C.2.6 As a good operating practice, only one RHR shutdown cooling loop should be placed in the SDC mode until the reactor coolant temperature has been decreased to 212°F.

C.2.7 Before securing Shutdown Cooling, decay heat levels should be evaluated such that they are low enough to allow sufficient time to complete the required evolution(s) without allowing boiling to occur, or that other systems are available with adequate decay heat removal capacity.

NOTE: The LPCI MOV bus charger/inverters are affected by harsh radiation fields. LPCI MOV Bus A Inverter 71INV-3A could be limited to 80 hours of operation due to radiation induced failure following a LOCA. LPCI MOV Bus B Inverter 71INV-3B could be limited to 6 hours of operation due to radiation induced failure following a LOCA.

D. STARTUP

NOTE: Subsections D.1 through D.6 are independent procedures and may be performed in any order, provided steps within each subsection are performed in the order written.

TABLE OF CONTENTS

SUBSECTION		PAGE
D.1	**RHR Loop A Shutdown Cooling Startup/Shifting Shutdown Cooling Loops**	8
D.2	**RHR Loop B Shutdown Cooling Startup/Shifting Shutdown Cooling Loops**	17
D.3	**Shifting RHR Pumps with RHR Loop A in Shutdown Cooling**	26
D.4	**Shifting RHR Pumps with RHR Loop B in Shutdown Cooling**	27
D.5	**RHR Loop A Shutdown Cooling Startup With RHR Loop B Already in Shutdown Cooling (No Interruption in Shutdown Cooling Flow)**	28
D.6	**RHR Loop B Shutdown Cooling Startup With RHR Loop A Already in Shutdown Cooling (No Interruption in Shutdown Cooling Flow)**	35

NOTE: RHR Loop B is preferred for shutdown cooling because RHR Pump 10P-3B or 10P-3D can be operated at maximum flow with lower motor winding temperatures than RHR Pump 10P-3A or 10P-3C.

D.1 **RHR Loop A Shutdown Cooling Startup/Shifting Shutdown Cooling Loops**

EXPA.3.6

D.1.1 **IF** fuel assemblies have been removed from the reactor vessel such that any in-core detector is not fully supported on all sides by a fuel assembly or blade guide,
THEN perform one of the following as determined by the Shift Manager:

a. Minimize the amount of time that shutdown cooling is in operation, unless required for decay heat removal.

OR

b. Arrange blade guides (in accordance with the appropriate RAP) to provide proper protection from flow induced vibration of in-core detectors.

D.1.2 Ensure required LCO's are entered per ODSO-34 for LPCI/Containment Cooling.

EXPA.3.8

D.1.3 Ensure continuous venting of RHR Header A and B is secured per Section G of OP-13.

D.1.4 **IF** rapid recovery of SDC is necessary,
AND expected time for restoration of SDC exceeds time-to-boil,
THEN the following exceptions may be taken at the Shift Manager's discretion:

- Loop flush per Step D.1.5 may be omitted
- Tags for Steps D.1.6 through D.1.8 may be delayed until after restoration complete
- If 10MOV-17 and/or 10MOV-18 isolated, they may be re-opened in Step D.1.19. Fill and vent is not required.

D.1.5 **IF** RHR Loop A suction is lined up to the torus,
THEN flush RHR Loop A per Subsection G.2.

NOTE: Steps D.1.6 through D.1.9 may be performed concurrently.

D.1.6 Ensure RHR Keep-Full Pump 10P-2A is protective tagged/removed from service as follows:

a. Ensure RHR KEEP-FULL PMP 10P-2A control switch is in STOP.

b. Ensure circuit breaker 71MCC-153-OE1 (10P-2A(M) RHR keep-full pump A motor) is in OFF.

c. Ensure closed 10RHR-96A (RHR keep-full pump A min flow valve).

d. Ensure closed 10RHR-99A (RHR keep-full pump A disch to RHR loop A isol valve).

e. Ensure closed the following valves:

- 10RHR-91A (RHR keep-full pump A suct isol valve)
- 10RHR-91C (RHR keep-full pump A suct isol valve)

D.1.7 Ensure 10MOV-16A is protective tagged/closed as follows:

a. Ensure closed MIN FLOW VLV 10MOV-16A.

b. Ensure circuit breaker 71MCC-153-OB3 (10MOV-16A RHR loop A min flow isol valve) is in OFF.

c. Ensure 10MOV-16A (RHR loop A min flow isol valve) handwheel is protective tagged closed.

D.1.8 Ensure 10MOV-39A is protective tagged/closed as follows:

a. Ensure closed RHR TEST TORUS CLG & SPRAY 10MOV-39A.

b. Ensure circuit breaker 71MCC-153-OD1 (10MOV-39A RHR loop A torus cooling isol valve) is in OFF.

c. Ensure 10MOV-39A (RHR A torus cooling isol valve) handwheel is protective tagged closed.

D.1.9 **IF** 10MOV-18 power supply is not lined up, **THEN** line up 10MOV-18 power supply as follows:

a. Place IS-1PCIA07 (control circuit isolation switch for 10MOV-18) in LOCAL.

b. Place IS-1PCIA07 (control circuit isolation switch for 10MOV-18) in REMOTE.

c. Unlock DS-1PCIA07 (power disconnect switch for 10MOV-18).

d. Place DS-1PCIA07 (power disconnect switch for 10MOV-18) in ON.

D.1.10 **IF** RHR Loop B is in a shutdown cooling lineup, **THEN** secure RHR Loop B from the shutdown cooling lineup as follows:

CAUTION

10MOV-16B is protective tagged closed. Running an RHR pump for greater than 1 minute without establishing flow could damage pump.

a. Close LPCI OUTBD INJ VLV 10MOV-27B.

b. Ensure the following RHR pumps are stopped:

- RHR PMP 10P-3B
- RHR PMP 10P-3D

c. Close LPCI INBD INJ VLV 10MOV-25B.

d. Open LPCI OUTBD INJ VLV 10MOV-27B.

e. Ensure open the following valves:

- HX B BYP VLV 10MOV-66B
- HX B OUTLET VLV 10MOV-12B

(Step D.1.10 continued on next page)

D.1.10 (Cont)

f. **IF** RHRSW Loop B operation is not required, **THEN** shut down RHRSW Loop B as follows:

1) Ensure closed RHRSW DISCH VLV FROM HX B 10MOV-89B.

2) Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

CAUTION

If RPV water level is less than 234.5 inches with no forced core recirculation, reactor coolant temperature indications could be invalid due to insufficient natural circulation.

g. **IF** RWR Pumps 02-2P-1A and 02-2P-1B are shutdown, **THEN** raise RPV water level to **ABOVE** 234.5 inches on REFUEL WTR LVL 02-3LI-86 at panel 09-4.

CAUTION

An extremely large drain path exists from the RPV to the torus if 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the associated 10MOV-13A, B, C, or D are open at the same time.

↓EXPA.3.2
h. Ensure closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15B
- SHUTDOWN CLG SUCT VLV 10MOV-15D

CAUTION

If RPV water level is less than 234.5 inches with no forced core recirculation, reactor coolant temperature indications could be invalid due to insufficient natural circulation.

D.1.11 **IF** RWR Pumps 02-2P-1A and 02-2P-1B are shutdown, **THEN** raise RPV water level to **ABOVE** 234.5 inches on REFUEL WTR LVL 02-3LI-86 at panel 09-4.

D.1.12 Ensure RPV pressure is **LESS THAN** 75 psig.

D.1.13 **IF** RHR shutdown cooling isolation logic has actuated,
THEN reset isolation logic as follows:

a. Verify the following relays indicate ON:

- 16A-K28 in panel 09-41
- 16A-K50 in panel 09-42

b. Simultaneously rotate the following PCIS VLV RESET switches to both RESET positions, spring return to NOR:

- 16A-S32
- 16A-S33

c. Verify the following relays indicate ON:

- 16A-K29 in panel 09-41
- 16A-K53 in panel 09-41
- 16A-K30 in panel 09-42
- 16A-K54 in panel 09-42

D.1.14 **IF** white HI DW OR LO LVL CLOSE OF 10MOV-25A IN SHUTDOWN CLG 10A-DS85A light is on,
THEN reset logic as follows:

a. Depress 10MOV-25A CLOSE SIG RESET 10A-S32A pushbutton.

b. Verify white HI DW OR LO LVL CLOSE OF 10MOV-25A IN SHUTDOWN CLG 10A-DS85A light is off.

D.1.15 Ensure closed the following valves:

- TORUS SUCT VLV 10MOV-13A
- TORUS SUCT VLV 10MOV-13C

D.1.16 Ensure open the shutdown cooling suction valve for the RHR pump that is to be used for shutdown cooling:

RHR Pump	Shutdown Cooling Suction Valve
10P-3A	SHUTDOWN CLG SUCT VLV 10MOV-15A
10P-3C	SHUTDOWN CLG SUCT VLV 10MOV-15C

D.1.17 Ensure open the following valves:

- HX A BYP VLV 10MOV-66A
- HX A INLET VLV 10MOV-65A
- HX A OUTLET VLV 10MOV-12A

D.1.18 Ensure RPV pressure is **LESS THAN** 10 psig, if practicable.

↓COMA.2.1
D.1.19 **IF** either of the following valves is closed:

- SHUTDOWN CLG SUCT 10MOV-18
- SHUTDOWN CLG SUCT 10MOV-17

THEN fill shutdown cooling suction piping per Subsection G.1.

D.1.20 Verify open the following valves:

- SHUTDOWN CLG SUCT 10MOV-18
- SHUTDOWN CLG SUCT 10MOV-17

D.1.21 Ensure Recirc Loops A and B are shutdown per OP-27, with the following exceptions:

a. If both suction and discharge valves will be closed at the same time, ensure pump min purge flow is isolated prior to isolating pump.

b. Ensure closed pump suction and/or discharge valve to prevent back flow through loop.

D.1.22 Close HX A OUTLET VLV 10MOV-12A.

D.1.23 **IF** any RHR Loop A keep-full alarms are present, **THEN** perform the following:

a. Ensure 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve) is throttled open.

b. Ensure RHR Loop A keep-full alarms are clear.

NOTE: RPV water level will trend downward when an RHR pump is started. Expected RPV water level response to RHR pump startup is shown on Attachment 2.

D.1.24 Ensure RPV water level is **GREATER THAN** 215 inches.

D.1.25 **IF** 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve) is open, **THEN** station an operator at 10RHR-274.

CAUTION

10MOV-16A is protective tagged closed. Running an RHR pump for greater than 1 minute without establishing flow could damage pump.

D.1.26 Start the RHR pump in RHR Loop A that is lined up for shutdown cooling (RHR PMP 10P-3A or RHR PMP 10P-3C).

D.1.27 Open LPCI INBD INJ VLV 10MOV-25A.

D.1.28 **IF** 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve) is open, **THEN** immediately close 10RHR-274.

D.1.29 Throttle open HX A OUTLET VLV 10MOV-12A for 15 to 20 seconds.

EXPA.3.3

CAUTION

When reactor coolant temperature is greater than 212°F, delay in establishing RHRSW flow could cause water hammer in RHRSW piping due to heatup of the RHR heat exchanger.

D.1.30 Control temperature per Section E.

D.1.31 **IF** RWR Loop B operation is required, **THEN** perform the following:

a. Ensure reactor coolant temperature is **LESS THAN** 140°F.

b. Place RWR Loop B in service per OP-27.

NOTE: If RHR Loop B is to be returned to the shutdown cooling lineup, and is not required to be in the LPCI standby lineup, performance of Step D.1.32 may be delayed or waived at the discretion of the Shift Manager.

D.1.32 **IF** RHR Loop B was in shutdown cooling prior to placing RHR Loop A in shutdown cooling, **THEN** place RHR Loop B in the LPCI standby lineup as follows:

↓EXPA.3.2
a. Verify closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15B
- SHUTDOWN CLG SUCT VLV 10MOV-15D

↓EXPA.3.4
b. Release protective tags and line up the following components in the position/status specified below:

1) Place circuit breaker 71MCC-163-OH3 (10MOV-16B RHR loop B min flow isol valve) in ON.

2) Open MIN FLOW VLV 10MOV-16B.

c. Open the following valves:

- TORUS SUCT VLV 10MOV-13B
- TORUS SUCT VLV 10MOV-13D

(Step D.1.32 continued on next page)

D.1.32 (Cont)

d. Release protective tags and line up the following components in the position/status specified below:

NOTE: The following components are listed in the correct order to restore components to service.

1) Ensure RHR KEEP-FULL PMP 10P-2B control switch is in STOP.

2) Open the following valves:

- 10RHR-91B (RHR keep-full pump B suct isol valve)
- 10RHR-91D (RHR keep-full pump B suct isol valve)

3) Open 10RHR-96B (RHR keep-full pump B min flow valve).

4) Open 10RHR-99B (RHR keep-full pump B disch to RHR loop B isol valve).

5) Place circuit breaker 71MCC-163-OE1 (10P-2B(M) RHR keep-full pump B motor) in ON.

6) Ensure closed 10MOV-39B.

7) Place circuit breaker 71MCC-163-OF1 (10MOV-39B RHR loop B torus cooling isol valve) in ON.

e. Place RHR Loop B keep-full in service per Section G of OP-13E.

NOTE: RHR Loop B is preferred for shutdown cooling because RHR Pump 10P-3B or 10P-3D can be operated at maximum flow with lower motor winding temperatures than RHR Pump 10P-3A or 10P-3C.

D.2 **RHR Loop B Shutdown Cooling Startup/Shifting Shutdown Cooling Loops**

↓EXPA.3.6

D.2.1 **IF** fuel assemblies have been removed from the reactor vessel such that any in-core detector is not fully supported on all sides by a fuel assembly or blade guide,
THEN perform one of the following as determined by the Shift Manager:

a. Minimize the amount of time that shutdown cooling is in operation, unless required for decay heat removal.

OR

b. Arrange blade guides (in accordance with the appropriate RAP) to provide proper protection from flow induced vibration of in-core detectors.

D.2.2 Ensure required LCO's are entered per ODSO-34 for LPCI/Containment Cooling.

↓EXPA.3.8

D.2.3 Ensure continuous venting of RHR Header A and B is secured per Section G of OP-13.

D.2.4 **IF** rapid recovery of SDC is necessary,
AND expected time for restoration of SDC exceeds time-to-boil,
THEN the following exceptions may be taken at the Shift Manager's discretion:

- Loop flush per Step D.2.5 may be omitted
- Tags for Steps D.2.6 through D.2.8 may be delayed until after restoration complete
- If 10MOV-17 and/or 10MOV-18 isolated, they may be re-opened in Step D.2.19. Fill and vent is not required.

D.2.5 **IF** RHR Loop B suction is lined up to the torus,
THEN flush RHR Loop B per Subsection G.3.

NOTE: Steps D.2.6 through D.2.9 may be performed concurrently.

D.2.6 Ensure RHR Keep-Full Pump 10P-2B is protective tagged/removed from service as follows:

a. Ensure RHR KEEP-FULL PMP 10P-2B control switch is in STOP.

b. Ensure circuit breaker 71MCC-163-OE1 (10P-2B(M) RHR keep-full pump B motor) is in OFF.

c. Ensure closed 10RHR-96B (RHR keep-full pump B min flow valve).

d. Ensure closed 10RHR-99B (RHR keep-full pump B disch to RHR loop B isol valve).

e. Ensure closed the following valves:

- 10RHR-91B (RHR keep-full pump B suct isol valve)
- 10RHR-91D (RHR keep-full pump B suct isol valve)

D.2.7 Ensure 10MOV-16B is protective tagged/closed as follows:

a. Ensure closed MIN FLOW VLV 10MOV-16B.

b. Ensure circuit breaker 71MCC-163-OH3 (10MOV-16B RHR loop B min flow isol valve) is in OFF.

c. Ensure 10MOV-16B (RHR loop B min flow isol valve) handwheel is protective tagged closed.

D.2.8 Ensure 10MOV-39B is protective tagged/closed as follows:

a. Ensure closed RHR TEST TORUS CLG & SPRAY 10MOV-39B.

b. Ensure circuit breaker 71MCC-163-OF1 (10MOV-39B RHR loop B torus cooling isol valve) is in OFF.

c. Ensure 10MOV-39B (RHR B torus cooling isol valve) handwheel is protective tagged closed.

D.2.9 **IF** 10MOV-18 power supply is not lined up, **THEN** line up 10MOV-18 power supply as follows:

a. Place IS-1PCIA07 (control circuit isolation switch for 10MOV-18) in LOCAL.

b. Place IS-1PCIA07 (control circuit isolation switch for 10MOV-18) in REMOTE.

c. Unlock DS-1PCIA07 (power disconnect switch for 10MOV-18).

d. Place DS-1PCIA07 (power disconnect switch for 10MOV-18) in ON.

D.2.10 **IF** RHR Loop A is in a shutdown cooling lineup, **THEN** secure RHR Loop A from the shutdown cooling lineup as follows:

CAUTION

10MOV-16A is protective tagged closed. Running an RHR pump for greater than 1 minute without establishing flow could damage pump.

a. Close LPCI OUTBD INJ VLV 10MOV-27A.

b. Ensure the following RHR pumps are stopped:

- RHR PMP 10P-3A
- RHR PMP 10P-3C

c. Close LPCI INBD INJ VLV 10MOV-25A.

d. Open LPCI OUTBD INJ VLV 10MOV-27A.

e. Ensure open the following valves:

- HX A BYP VLV 10MOV-66A
- HX A OUTLET VLV 10MOV-12A

(Step D.2.10 continued on next page)

D.2.10 (Cont)

f. **IF** RHRSW Loop A operation is not required, **THEN** shut down RHRSW Loop A as follows:

1) Ensure closed RHRSW DISCH VLV FROM HX A 10MOV-89A.

2) Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

CAUTION

If RPV water level is less than 234.5 inches with no forced core recirculation, reactor coolant temperature indications could be invalid due to insufficient natural circulation.

g. **IF** RWR Pumps 02-2P-1A and 02-2P-1B are shutdown, **THEN** raise RPV water level to **ABOVE** 234.5 inches on REFUEL WTR LVL 02-3LI-86 at panel 09-4.

CAUTION

An extremely large drain path exists from the RPV to the torus if 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the associated 10MOV-13A, B, C, or D are open at the same time.

↓EXPA.3.2

h. Ensure closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15A
- SHUTDOWN CLG SUCT VLV 10MOV-15C

CAUTION

If RPV water level is less than 234.5 inches with no forced core recirculation, reactor coolant temperature indications could be invalid due to insufficient natural circulation.

D.2.11 **IF** RWR Pumps 02-2P-1A and 02-2P-1B are shutdown, **THEN** raise RPV water level to **ABOVE** 234.5 inches on REFUEL WTR LVL 02-3LI-86 at panel 09-4.

D.2.12 Ensure RPV pressure is **LESS THAN** 75 psig.

D.2.13 **IF** RHR shutdown cooling isolation logic has actuated,
THEN reset isolation logic as follows:

a. Verify the following relays indicate ON:

- 16A-K28 in panel 09-41
- 16A-K50 in panel 09-42

b. Simultaneously rotate the following PCIS VLV RESET switches to both RESET positions, spring return to NOR:

- 16A-S32
- 16A-S33

c. Verify the following relays indicate ON:

- 16A-K29 in panel 09-41
- 16A-K53 in panel 09-41
- 16A-K30 in panel 09-42
- 16A-K54 in panel 09-42

D.2.14 **IF** white HI DW OR LO LVL CLOSE OF 10MOV-25B IN SHUTDOWN CLG 10A-DS85B light is on,
THEN reset logic as follows:

a. Depress 10MOV-25B CLOSE SIG RESET 10A-S32B pushbutton.

b. Verify white HI DW OR LO LVL CLOSE OF 10MOV-25B IN SHUTDOWN CLG 10A-DS85B light is off.

D.2.15 Ensure closed the following valves:

- TORUS SUCT VLV 10MOV-13B
- TORUS SUCT VLV 10MOV-13D

D.2.16 Ensure open the shutdown cooling suction valve for the RHR pump that is to be used for shutdown cooling:

RHR Pump	Shutdown Cooling Suction Valve
10P-3B	SHUTDOWN CLG SUCT VLV 10MOV-15B
10P-3D	SHUTDOWN CLG SUCT VLV 10MOV-15D

D.2.17 Ensure open the following valves:

- HX B BYP VLV 10MOV-66B
- HX B INLET VLV 10MOV-65B
- HX B OUTLET VLV 10MOV-12B

D.2.18 Ensure RPV pressure is **LESS THAN** 10 psig, if practicable.

COMA.2.1
D.2.19 **IF** either of the following valves is closed:

- SHUTDOWN CLG SUCT 10MOV-18
- SHUTDOWN CLG SUCT 10MOV-17

THEN fill shutdown cooling suction piping per Subsection G.1.

D.2.20 Verify open the following valves:

- SHUTDOWN CLG SUCT 10MOV-18
- SHUTDOWN CLG SUCT 10MOV-17

D.2.21 Ensure Recirc Loops A and B are shutdown per OP-27, with the following exceptions:

a. If both suction and discharge valves will be closed at the same time, ensure pump min purge flow is isolated prior to isolating pump.

b. Ensure closed pump suction and/or discharge valve to prevent back flow through loop.

D.2.22 Close HX B OUTLET VLV 10MOV-12B.

D.2.23 **IF** any RHR Loop B keep-full alarms are present, **THEN** perform the following:

a. Ensure 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve) is throttled open.

b. Ensure RHR Loop B keep-full alarms are clear.

NOTE: RPV water level will trend downward when an RHR pump is started. Expected RPV water level response to RHR pump startup is shown on Attachment 2.

D.2.24 Ensure RPV water level is **GREATER THAN** 215 inches.

D.2.25 **IF** 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve) is open, **THEN** station an operator at 10RHR-260.

CAUTION

10MOV-16B is protective tagged closed. Running an RHR pump for greater than 1 minute without establishing flow could damage pump.

D.2.26 Start the RHR pump in RHR Loop B that is lined up for shutdown cooling (RHR PMP 10P-3B or RHR PMP 10P-3D).

D.2.27 Open LPCI INBD INJ VLV 10MOV-25B.

D.2.28 **IF** 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve) is open, **THEN** immediately close 10RHR-260.

D.2.29 Throttle open HX B OUTLET VLV 10MOV-12B for 15 to 20 seconds.

⬇EXPA.3.3

CAUTION

When reactor coolant temperature is greater than 212°F, delay in establishing RHRSW flow could cause water hammer in RHRSW piping due to heatup of the RHR heat exchanger.

D.2.30 Control temperature per Section E.

D.2.31 **IF** RWR Loop A operation is required, **THEN** perform the following:

a. Ensure reactor coolant temperature is **LESS THAN** 140°F.

b. Place RWR Loop A in service per OP-27.

NOTE: If RHR Loop A is to be returned to the shutdown cooling lineup, and is not required to be in the LPCI standby lineup, performance of Step D.2.32 may be delayed or waived at the discretion of the Shift Manager.

D.2.32 **IF** RHR Loop A was in shutdown cooling prior to placing RHR Loop B in shutdown cooling, **THEN** place RHR Loop A in the LPCI standby lineup as follows:

↓EXPA.3.2
a. Verify closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15A
- SHUTDOWN CLG SUCT VLV 10MOV-15C

↓EXPA.3.4
b. Release protective tags and line up the following components in the position/status specified below:

1) Place circuit breaker 71MCC-153-OB3 (10MOV-16A RHR loop A min flow isol valve) in ON.

2) Open MIN FLOW VLV 10MOV-16A.

c. Open the following valves:

- TORUS SUCT VLV 10MOV-13A
- TORUS SUCT VLV 10MOV-13C

(Step D.2.32 continued on next page)

D.2.32 (Cont)

d. Release protective tags and line up the following components in the position/status specified below:

NOTE: The following components are listed in the correct order to restore components to service.

1) Ensure RHR KEEP-FULL PMP 10P-2A control switch is in STOP.

2) Open the following valves:

- 10RHR-91A (RHR keep-full pump A suct isol valve)
- 10RHR-91C (RHR keep-full pump A suct isol valve)

3) Open 10RHR-96A (RHR keep-full pump A min flow valve).

4) Open 10RHR-99A (RHR keep-full pump A disch to RHR loop A isol valve).

5) Place circuit breaker 71MCC-153-OE1 (10P-2A(M) RHR keep-full pump A motor) in ON.

6) Ensure closed 10MOV-39A.

7) Place circuit breaker 71MCC-153-OD1 (10MOV-39A RHR loop A torus cooling isol valve) in ON.

e. Place RHR Loop A keep-full in service per Section G of OP-13E.

NOTE: Reactor coolant temperature could rise while shifting RHR pumps due to 10MOV-12A being closed.

D.3 **Shifting RHR Pumps with RHR Loop A in Shutdown Cooling**

D.3.1 Ensure open HX A BYP VLV 10MOV-66A.

D.3.2 Ensure closed HX A OUTLET VLV 10MOV-12A.

CAUTION

Operation of two RHR pumps at a flow rate less than 13,000 gpm could cause high pump vibration.

D.3.3 **IF** shifting from RHR Pump A to RHR Pump C, **THEN** perform the following:

a. Ensure closed TORUS SUCT VLV 10MOV-13C.

b. Open SHUTDOWN CLG SUCT VLV 10MOV-15C.

c. Start RHR PMP 10P-3C.

d. Stop RHR PMP 10P-3A.

e. Close SHUTDOWN CLG SUCT VLV 10MOV-15A.

D.3.4 **IF** shifting from RHR Pump C to RHR Pump A, **THEN** perform the following:

a. Ensure closed TORUS SUCT VLV 10MOV-13A.

b. Open SHUTDOWN CLG SUCT VLV 10MOV-15A.

c. Start RHR PMP 10P-3A.

d. Stop RHR PMP 10P-3C.

e. Close SHUTDOWN CLG SUCT VLV 10MOV-15C.

D.3.5 Control temperature per Section E.

D.3.6 **IF** RWR Loop B operation is required, **THEN** perform the following:

a. Ensure reactor coolant temperature is **LESS THAN** 140°F.

b. Place RWR Loop B in service per OP-27.

NOTE: Reactor coolant temperature could rise while shifting RHR pumps due to 10MOV-12B being closed.

D.4 **Shifting RHR Pumps with RHR Loop B in Shutdown Cooling**

D.4.1 Ensure open HX B BYP VLV 10MOV-66B.

D.4.2 Ensure closed HX B OUTLET VLV 10MOV-12B.

CAUTION

Operation of two RHR pumps at a flow rate less than 13,000 gpm could cause high pump vibration.

D.4.3 **IF** shifting from RHR Pump B to RHR Pump D, **THEN** perform the following:

a. Ensure closed TORUS SUCT VLV 10MOV-13D.

b. Open SHUTDOWN CLG SUCT VLV 10MOV-15D.

c. Start RHR PMP 10P-3D.

d. Stop RHR PMP 10P-3B.

e. Close SHUTDOWN CLG SUCT VLV 10MOV-15B.

D.4.4 **IF** shifting from RHR Pump D to RHR Pump B, **THEN** perform the following:

a. Ensure closed TORUS SUCT VLV 10MOV-13B.

b. Open SHUTDOWN CLG SUCT VLV 10MOV-15B.

c. Start RHR PMP 10P-3B.

d. Stop RHR PMP 10P-3D.

e. Close SHUTDOWN CLG SUCT VLV 10MOV-15D.

D.4.5 Control temperature per Section E.

D.4.6 **IF** RWR Loop A operation is required, **THEN** perform the following:

a. Ensure reactor coolant temperature is **LESS THAN** 140°F.

b. Place RWR Loop A in service per OP-27.

NOTE 1: This procedure is to be used to place RHR Loop A in the Shutdown Cooling mode when RHR Loop B is already in Shutdown Cooling, the reactor is in the cold condition, and it is desired to provide uninterrupted shutdown cooling flow through the reactor core.

NOTE 2: RHR Loop B is preferred for shutdown cooling because RHR Pump 10P-3B or 10P-3D can be operated at maximum flow with lower motor winding temperatures than RHR Pump 10P-3A or 10P-3C.

D.5 **RHR Loop A Shutdown Cooling Startup With RHR Loop B Already in Shutdown Cooling (No Interruption in Shutdown Cooling Flow)**

D.5.1 **IF** RHR Loop A suction is lined up to the torus, **THEN** flush RHR Loop A per Subsection G.2.

NOTE: Steps D.5.2 through D.5.4 may be performed concurrently.

D.5.2 Ensure RHR Keep-Full Pump 10P-2A is protective tagged/removed from service as follows:

a. Ensure RHR KEEP-FULL PMP 10P-2A control switch is in STOP.

b. Ensure circuit breaker 71MCC-153-OE1 (10P-2A(M) RHR keep-full pump A motor) is in OFF.

c. Ensure closed 10RHR-96A (RHR keep-full pump A min flow valve).

d. Ensure closed 10RHR-99A (RHR keep-full pump A disch to RHR loop A isol valve).

e. Ensure closed the following valves:

- 10RHR-91A (RHR keep-full pump A suct isol valve)
- 10RHR-91C (RHR keep-full pump A suct isol valve)

D.5.3 Ensure 10MOV-16A is protective tagged/closed as follows:

a. Ensure closed MIN FLOW VLV 10MOV-16A.

b. Ensure circuit breaker 71MCC-153-OB3 (10MOV-16A RHR loop A min flow isol valve) is in OFF.

c. Ensure 10MOV-16A (RHR loop A min flow isol valve) handwheel is protective tagged closed.

D.5.4 Ensure 10MOV-39A is protective tagged/closed as follows:

a. Ensure closed RHR TEST TORUS CLG & SPRAY 10MOV-39A.

b. Ensure circuit breaker 71MCC-153-OD1 (10MOV-39A RHR loop A torus cooling isol valve) is in OFF.

c. Ensure 10MOV-39A (RHR A torus cooling isol valve) handwheel is protective tagged closed.

D.5.5 **IF** white HI DW OR LO LVL CLOSE OF 10MOV-25A IN SHUTDOWN CLG 10A-DS85A light is on, **THEN** reset logic as follows:

a. Depress 10MOV-25A CLOSE SIG RESET 10A-S32A pushbutton.

b. Verify white HI DW OR LO LVL CLOSE OF 10MOV-25A IN SHUTDOWN CLG 10A-DS85A light is off.

NOTE: In order to minimize the time required to recover from a loss of shutdown cooling, this procedure may be performed up through Step D.5.6 to place RHR Loop A in a standby shutdown cooling lineup. While RHR Loop A is in a standby shutdown cooling lineup, Step D.5.11 may be performed at the discretion of the Shift Manager to maintain RHR Loop A full of water.

D.5.6 Ensure closed the following valves:

- TORUS SUCT VLV 10MOV-13A
- TORUS SUCT VLV 10MOV-13C

D.5.7 Ensure open the shutdown cooling suction valve for the RHR pump that is to be used for shutdown cooling:

RHR Pump	Shutdown Cooling Suction Valve
10P-3A	SHUTDOWN CLG SUCT VLV 10MOV-15A
10P-3C	SHUTDOWN CLG SUCT VLV 10MOV-15C

D.5.8 Ensure open the following valves:

- HX A BYP VLV 10MOV-66A
- HX A INLET VLV 10MOV-65A
- HX A OUTLET VLV 10MOV-12A

D.5.9 Ensure Recirc Loops A and B are shutdown per OP-27, with the following exceptions:

a. If both suction and discharge valves will be closed at the same time, ensure pump min purge flow is isolated prior to isolating pump.

b. Ensure closed pump suction and/or discharge valve to prevent back flow through loop.

D.5.10 Close HX A OUTLET VLV 10MOV-12A.

D.5.11 **IF** any RHR Loop A keep-full alarms are present, **THEN** perform the following:

a. Ensure 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve) is throttled open.

b. Ensure RHR Loop A keep-full alarms are clear.

NOTE: RPV water level will trend downward when an RHR pump is started. Expected RPV water level response to RHR pump startup is shown on Attachment 2.

D.5.12 Ensure RPV water level is **GREATER THAN** 215 inches.

D.5.13 **IF** 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve) is open, **THEN** station an operator at 10RHR-274.

CAUTION

10MOV-16A is protective tagged closed. Running an RHR pump for greater than 1 minute without establishing flow could damage pump.

D.5.14 Start the RHR pump in RHR Loop A that is lined up for shutdown cooling (RHR PMP 10P-3A or RHR PMP 10P-3C).

D.5.15 Open LPCI INBD INJ VLV 10MOV-25A.

D.5.16 **IF** 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve) is open, **THEN** immediately close 10RHR-274.

D.5.17 Throttle open HX A OUTLET VLV 10MOV-12A for 15 to 20 seconds.

D.5.18 Control temperature per Section E.

D.5.19 Secure RHR Loop B from the shutdown cooling lineup as follows:

CAUTION

10MOV-16B is protective tagged closed. Running an RHR pump for greater than 1 minute without establishing flow could damage pump.

a. Close LPCI OUTBD INJ VLV 10MOV-27B.

b. Ensure the following RHR pumps are stopped:

- RHR PMP 10P-3B
- RHR PMP 10P-3D

c. Close LPCI INBD INJ VLV 10MOV-25B.

d. Open LPCI OUTBD INJ VLV 10MOV-27B.

e. Ensure open the following valves:

- HX B BYP VLV 10MOV-66B
- HX B OUTLET VLV 10MOV-12B

(Step D.5.19 continued on next page)

D.5.19 (Cont)

f. **IF** RHRSW Loop B operation is not required, **THEN** shut down RHRSW Loop B as follows:

1) Ensure closed RHRSW DISCH VLV FROM HX B 10MOV-89B.

2) Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

CAUTION

An extremely large drain path exists from the RPV to the torus if 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the associated 10MOV-13A, B, C, or D are open at the same time.

↓EXPA.3.2

g. Ensure closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15B
- SHUTDOWN CLG SUCT VLV 10MOV-15D

D.5.20 **IF** RWR Loop B operation is required, **THEN** perform the following:

a. Ensure reactor coolant temperature is **LESS THAN** 140°F.

b. Place RWR Loop B in service per OP-27.

D.5.21 **IF** RHR Loop B will be maintained in a standby shutdown cooling lineup, **THEN** throttle open 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve) at the discretion of the Shift Manager to maintain RHR Loop B full of water.

NOTE: If RHR Loop B is to be returned to the shutdown cooling lineup, and is not required to be in the LPCI standby lineup, performance of Step D.5.22 may be delayed or waived at the discretion of the Shift Manager.

D.5.22 Place RHR Loop B in the LPCI standby lineup as follows:

↓EXPA.3.2

a. Verify closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15B
- SHUTDOWN CLG SUCT VLV 10MOV-15D

↓EXPA.3.4

b. Release protective tags and line up the following components in the position/status specified below:

1) Place circuit breaker 71MCC-163-OH3 (10MOV-16B RHR loop B min flow isol valve) in ON.

2) Open MIN FLOW VLV 10MOV-16B.

c. Open the following valves:

- TORUS SUCT VLV 10MOV-13B
- TORUS SUCT VLV 10MOV-13D

(Step D.5.22 continued on next page)

D.5.22 (Cont)

d. Release protective tags and line up the following components in the position/status specified below:

NOTE: The following components are listed in the correct order to restore components to service.

1) Ensure RHR KEEP-FULL PMP 10P-2B control switch is in STOP.

2) Open the following valves:

- 10RHR-91B (RHR keep-full pump B suct isol valve)
- 10RHR-91D (RHR keep-full pump B suct isol valve)

3) Open 10RHR-96B (RHR keep-full pump B min flow valve).

4) Open 10RHR-99B (RHR keep-full pump B disch to RHR loop B isol valve).

5) Place circuit breaker 71MCC-163-OE1 (10P-2B(M) RHR keep-full pump B motor) in ON.

6) Ensure closed 10MOV-39B.

7) Place circuit breaker 71MCC-163-OF1 (10MOV-39B RHR loop B torus cooling isol valve) in ON.

e. Place RHR Loop B keep-full in service per Section G of OP-13E.

NOTE 1: This procedure is to be used to place RHR Loop B in the Shutdown Cooling mode when RHR Loop A is already in Shutdown Cooling, the reactor is in the cold condition, and it is desired to provide uninterrupted shutdown cooling flow through the reactor core.

NOTE 2: RHR Loop B is preferred for shutdown cooling because RHR Pump 10P-3B or 10P-3D can be operated at maximum flow with lower motor winding temperatures than RHR Pump 10P-3A or 10P-3C.

D.6 **RHR Loop B Shutdown Cooling Startup With RHR Loop A Already in Shutdown Cooling (No Interruption in Shutdown Cooling Flow)**

D.6.1 **IF** RHR Loop B suction is lined up to the torus, **THEN** flush RHR Loop B per Subsection G.3.

NOTE: Steps D.6.2 through D.6.4 may be performed concurrently.

D.6.2 Ensure RHR Keep-Full Pump 10P-2B is protective tagged/removed from service as follows:

a. Ensure RHR KEEP-FULL PMP 10P-2B control switch is in STOP.

b. Ensure circuit breaker 71MCC-163-OE1 (10P-2B(M) RHR keep-full pump B motor) is in OFF.

c. Ensure closed 10RHR-96B (RHR keep-full pump B min flow valve).

d. Ensure closed 10RHR-99B (RHR keep-full pump B disch to RHR loop B isol valve).

e. Ensure closed the following valves:

- 10RHR-91B (RHR keep-full pump B suct isol valve)
- 10RHR-91D (RHR keep-full pump B suct isol valve)

D.6.3 Ensure 10MOV-16B is protective tagged/closed as follows:

a. Ensure closed MIN FLOW VLV 10MOV-16B.

b. Ensure circuit breaker 71MCC-163-OH3 (10MOV-16B RHR loop B min flow isol valve) is in OFF.

c. Ensure 10MOV-16B (RHR loop B min flow isol valve) handwheel is protective tagged closed.

D.6.4 Ensure 10MOV-39B is protective tagged/closed as follows:

a. Ensure closed RHR TEST TORUS CLG & SPRAY 10MOV-39B.

b. Ensure circuit breaker 71MCC-163-OF1 (10MOV-39B RHR loop B torus cooling isol valve) is in OFF.

c. Ensure 10MOV-39B (RHR B torus cooling isol valve) handwheel is protective tagged closed.

D.6.5 **IF** white HI DW OR LO LVL CLOSE OF 10MOV-25B IN SHUTDOWN CLG 10A-DS85B light is on, **THEN** reset logic as follows:

a. Depress 10MOV-25B CLOSE SIG RESET 10A-S32B pushbutton.

b. Verify white HI DW OR LO LVL CLOSE OF 10MOV-25B IN SHUTDOWN CLG 10A-DS85B light is off.

NOTE: In order to minimize the time required to recover from a loss of shutdown cooling, this procedure may be performed up through Step D.6.6 to place RHR Loop B in a standby shutdown cooling lineup. While RHR Loop B is in a standby shutdown cooling lineup, Step D.6.11 may be performed at the discretion of the Shift Manager to maintain RHR Loop B full of water.

D.6.6 Ensure closed the following valves:

- TORUS SUCT VLV 10MOV-13B
- TORUS SUCT VLV 10MOV-13D

D.6.7 Ensure open the shutdown cooling suction valve for the RHR pump that is to be used for shutdown cooling:

RHR Pump	Shutdown Cooling Suction Valve
10P-3B	SHUTDOWN CLG SUCT VLV 10MOV-15B
10P-3D	SHUTDOWN CLG SUCT VLV 10MOV-15D

D.6.8 Ensure open the following valves:

- HX B BYP VLV 10MOV-66B
- HX B INLET VLV 10MOV-65B
- HX B OUTLET VLV 10MOV-12B

D.6.9 Ensure Recirc Loops A and B are shutdown per OP-27, with the following exceptions:

a. If both suction and discharge valves will be closed at the same time, ensure pump min purge flow is isolated prior to isolating pump.

b. Ensure closed pump suction and/or discharge valve to prevent back flow through loop.

D.6.10 Close HX B OUTLET VLV 10MOV-12B.

D.6.11 **IF** any RHR Loop B keep-full alarms are present, **THEN** perform the following:

a. Ensure 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve) is throttled open.

b. Ensure RHR Loop B keep-full alarms are clear.

NOTE: RPV water level will trend downward when an RHR pump is started. Expected RPV water level response to RHR pump startup is shown on Attachment 2.

D.6.12 Ensure RPV water level is **GREATER THAN** 215 inches.

D.6.13 **IF** 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve) is open, **THEN** station an operator at 10RHR-260.

CAUTION

10MOV-16B is protective tagged closed. Running an RHR pump for greater than 1 minute without establishing flow could damage pump.

D.6.14 Start the RHR pump in RHR Loop B that is lined up for shutdown cooling (RHR PMP 10P-3B or RHR PMP 10P-3D).

D.6.15 Open LPCI INBD INJ VLV 10MOV-25B.

D.6.16 **IF** 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve) is open, **THEN** immediately close 10RHR-260.

D.6.17 Throttle open HX B OUTLET VLV 10MOV-12B for 15 to 20 seconds.

D.6.18 Control temperature per Section E.

D.6.19 Secure RHR Loop A from the shutdown cooling lineup as follows:

CAUTION

10MOV-16A is protective tagged closed. Running an RHR pump for greater than 1 minute without establishing flow could damage pump.

a. Close LPCI OUTBD INJ VLV 10MOV-27A.

b. Ensure the following RHR pumps are stopped:

- RHR PMP 10P-3A
- RHR PMP 10P-3C

c. Close LPCI INBD INJ VLV 10MOV-25A.

d. Open LPCI OUTBD INJ VLV 10MOV-27A.

e. Ensure open the following valves:

- HX A BYP VLV 10MOV-66A
- HX A OUTLET VLV 10MOV-12A

(Step D.6.19 continued on next page)

D.6.19 (Cont)

f. **IF** RHRSW Loop A operation is not required, **THEN** shut down RHRSW Loop A as follows:

1) Ensure closed RHRSW DISCH VLV FROM HX A 10MOV-89A.

2) Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

CAUTION

An extremely large drain path exists from the RPV to the torus if 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the associated 10MOV-13A, B, C, or D are open at the same time.

↓EXPA.3.2

g. Ensure closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15A
- SHUTDOWN CLG SUCT VLV 10MOV-15C

D.6.20 **IF** RWR Loop A operation is required, **THEN** perform the following:

a. Ensure reactor coolant temperature is **LESS THAN** 140°F.

b. Place RWR Loop A in service per OP-27.

D.6.21 **IF** RHR Loop A will be maintained in a standby shutdown cooling lineup, **THEN** throttle open 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve) at the discretion of the Shift Manager to maintain RHR Loop A full of water.

NOTE: If RHR Loop A is to be returned to the shutdown cooling lineup, and is not required to be in the LPCI standby lineup, performance of Step D.6.22 may be delayed or waived at the discretion of the Shift Manager.

D.6.22 Place RHR Loop A in the LPCI standby lineup as follows:

↓EXPA.3.2

a. Verify closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15A
- SHUTDOWN CLG SUCT VLV 10MOV-15C

↓EXPA.3.4

b. Release protective tags and line up the following components in the position/status specified below:

1) Place circuit breaker 71MCC-153-OB3 (10MOV-16A RHR loop A min flow isol valve) in ON.

2) Open MIN FLOW VLV 10MOV-16A.

c. Open the following valves:

- TORUS SUCT VLV 10MOV-13A
- TORUS SUCT VLV 10MOV-13C

(Step D.6.22 continued on next page)

D.6.22 (Cont)

d. Release protective tags and line up the following components in the position/status specified below:

NOTE: The following components are listed in the correct order to restore components to service.

1) Ensure RHR KEEP-FULL PMP 10P-2A control switch is in STOP.

2) Open the following valves:

- 10RHR-91A (RHR keep-full pump A suct isol valve)
- 10RHR-91C (RHR keep-full pump A suct isol valve)

3) Open 10RHR-96A (RHR keep-full pump A min flow valve).

4) Open 10RHR-99A (RHR keep-full pump A disch to RHR loop A isol valve).

5) Place circuit breaker 71MCC-153-OE1 (10P-2A(M) RHR keep-full pump A motor) in ON.

6) Ensure closed 10MOV-39A.

7) Place circuit breaker 71MCC-153-OD1 (10MOV-39A RHR loop A torus cooling isol valve) in ON.

e. Place RHR Loop A keep-full in service per Section G of OP-13E.

NOTE: Section written for A loop, use designations in parentheses for B loop.

E. **NORMAL OPERATION**

E.1 Refer to ODSO-32 for shutdown operation guidance.

NOTE: It is normal for 10FI-136A(B) to indicate up to 1200 gpm when a RHR pump is running with 10MOV-26A(B) closed.

E.2 Control RPV and reactor coolant temperatures using one or a combination of the following methods:

↓EXPA.3.3

CAUTION

When reactor coolant temperature is greater than 212°F, delay in establishing RHRSW flow could cause water hammer in RHRSW piping due to heatup of the RHR heat exchanger.

- Start up RHRSW Loop per Section D of OP-13C.

NOTE: 10MOV-12A(B) or 10MOV-66A(B) is maintained full open to ensure maximum RHR pump flow at all times while in shutdown cooling.

- With 10MOV-66A(B) full open, throttle HX A(B) OUTLET VLV 10MOV-12A(B).
- With 10MOV-12A(B) full open, throttle HX A(B) BYP VLV 10MOV-66A(B).

CAUTION

Running an RHRSW pump with flow less than 2500 gpm or greater than 4000 gpm per pump could damage pump.

- Throttle RHRSW DISCH VLV FROM HX A 10MOV-89A(B).
- Start or stop second RHRSW pump in RHRSW Loop.
- Shut down RHRSW Loop per Section F of OP-13C.

E.3 **IF** RHR pump motor winding temperature reaches 284°F while running in shutdown cooling,
THEN immediately perform Subsection G.6 **OR** G.7.

E.4 Maintain reactor coolant temperature **GREATER THAN** 68°F.

E.5 Maintain cooldown rate of the following parameters **LESS THAN OR EQUAL TO** 80°F per hour:

- RHR Loop A(B) water temperature
- RPV shell temperature
- RPV flange temperature

E.6 Monitor RPV head flange temperature on RX VESSEL TEMP 02-3TR-89 (point 6) VESSEL TOP HEAD FLANGE 02-3TE-66B1 at panel 09-21.

NOTE: The LPCI MOV bus charger/inverters are affected by harsh radiation fields. LPCI MOV Bus A Inverter 71INV-3A could be limited to 80 hours of operation due to radiation induced failure following a LOCA. LPCI MOV Bus B Inverter 71INV-3B could be limited to 6 hours of operation due to radiation induced failure following a LOCA.

F. SHUTDOWN

NOTE: Subsections F.1 and F.2 are independent procedures and may be performed in any order, provided steps within each subsection are performed in the order written.

TABLE OF CONTENTS

SUBSECTION		PAGE
F.1	**RHR Loop A Shutdown Cooling Shutdown**	45
F.2	**RHR Loop B Shutdown Cooling Shutdown**	50

CAUTION

Before securing Shutdown Cooling, decay heat levels should be evaluated such that they are low enough to allow sufficient time to complete the required evolution(s) without allowing boiling to occur, or that other systems are available with adequate decay heat removal capacity.

F.1 **RHR Loop A Shutdown Cooling Shutdown**

CAUTION

10MOV-16A is protective tagged closed. Running an RHR pump for greater than 1 minute without establishing flow could damage pump.

F.1.1 **IF** an RWR loop will be placed in service when shutdown cooling is secured,
THEN ensure reactor coolant temperature is **LESS THAN** 140°F to allow opening the RWR pump discharge valve to warm the loop without performing ST-26K.

F.1.2 Close LPCI OUTBD INJ VLV 10MOV-27A.

F.1.3 Ensure the following RHR pumps are stopped:

- RHR PMP 10P-3A
- RHR PMP 10P-3C

F.1.4 Close LPCI INBD INJ VLV 10MOV-25A.

F.1.5 Open LPCI OUTBD INJ VLV 10MOV-27A.

EXPA.3.2
F.1.6 Ensure closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15A
- SHUTDOWN CLG SUCT VLV 10MOV-15C

F.1.7 **IF** RHRSW Loop A operation <u>is not</u> required,
THEN shut down RHRSW Loop A as follows:

a. Close RHRSW DISCH VLV FROM HX A 10MOV-89A.

b. Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

CAUTION

If RPV water level is less than 234.5 inches with no forced core recirculation, reactor coolant temperature indications could be invalid due to insufficient natural circulation.

F.1.8 **IF** RWR Pumps 02-2P-1A and 02-2P-1B are shutdown,
AND will remain shutdown,
THEN raise RPV water level to **ABOVE** 234.5 inches on REFUEL WTR LVL 02-3LI-86 at panel 09-4.

F.1.9 **IF** RHR Loop A is being shutdown for reasons that will not cause or require further RHR Loop A valve operation,
AND RHR Loop A will be returned to shutdown cooling per Subsection D.1,
THEN return to Subsection D.1 when RHR Loop A shutdown cooling mode operation is desired.

F.1.10 **IF** RHR Loop A is being shutdown for testing that will cause or require RHR Loop A valve operation; for example, surveillance testing that causes system isolation signals,
AND RHR Loop A is to be returned to the shutdown cooling mode per Subsection D.1,
THEN line up RHR Loop A as follows:

CAUTION

An extremely large drain path exists from the RPV to the torus if 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the associated 10MOV-13A, B, C, or D are open at the same time.

a. Close the following valves:

- SHUTDOWN CLG SUCT 10MOV-18
- SHUTDOWN CLG SUCT 10MOV-17

b. Throttle 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve) as necessary to maintain RHR Loop A full.

↓EXPA.3.7

F.1.11 **IF** both trains of RHR shutdown cooling are removed from service for the first time during a refueling outage,
THEN perform the following before rendering RHR shutdown cooling unavailable:

a. Ensure Fuel Pool Cooling System is operating.

b. Ensure only one train of DHR is operating, providing cooling to the core and spent fuel pool (gates removed, cavity flooded).

c. Verify availability of the redundant DHR train and backup emergency power.

d. Close the following valves to isolate RBCLC to fuel pool cooling heat exchangers:

- 15RBC-15A (fuel pool heat exch 19E-1A RBCLC return isol valve)
- 15RBC-15B (fuel pool heat exch 19E-1B RBCLC return isol valve)

NOTE: RHR shutdown cooling is required to be secured for at least 2 hours to ensure equilibrium thermal-hydraulic conditions are established in the core and spent fuel pool.

e. **WHEN** both trains of RHR shutdown cooling have been secured for at least 2 hours, verify DHR System is capable of assuming the spent fuel pool and core decay heat load by observing temperature trend on RHR WTR & HPCI TEMP 10TRS-131. Points 7, 8, and 9 should show a constant or decreasing temperature trend over the next 1 hour period.

NOTE: Temperature data need not be retained. SM log entry to state that DHR capability was demonstrated per OP-13D is sufficient documentation.

f. SM review temperature trend data to verify DHR capability and document in SM Log.

g. Open the following valves:

- 15RBC-15A
- 15RBC-15B

F.1.12 **IF** RHR Loop A shutdown cooling operation is no longer required,
THEN place RHR Loop A in the LPCI standby lineup as follows:

> **CAUTION**
>
> An extremely large drain path exists from the RPV to the torus if 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the associated 10MOV-13A, B, C, or D are open at the same time.

a. Close the following valves:

- SHUTDOWN CLG SUCT 10MOV-18
- SHUTDOWN CLG SUCT 10MOV-17

EXPA.3.2
b. Ensure closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15A
- SHUTDOWN CLG SUCT VLV 10MOV-15C

c. Ensure open the following valves:

- HX A BYP VLV 10MOV-66A
- HX A OUTLET VLV 10MOV-12A

d. Release protective tags and line up the following components in the position/status specified below:

1) Place circuit breaker 71MCC-153-OB3 (10MOV-16A RHR loop A min flow isol valve) in ON.

2) Open MIN FLOW VLV 10MOV-16A.

e. Open the following valves:

- TORUS SUCT VLV 10MOV-13A
- TORUS SUCT VLV 10MOV-13C

(Step F.1.12 continued on next page)

F.1.12 (Cont)

f. Release protective tags and line up the following components in the position/status specified below:

NOTE: The following components are listed in the correct order to restore components to service.

1) Ensure RHR KEEP-FULL PMP 10P-2A control switch is in STOP.

2) Open the following valves:

- 10RHR-91A (RHR keep-full pump A suct isol valve)
- 10RHR-91C (RHR keep-full pump A suct isol valve)

3) Open 10RHR-96A (RHR keep-full pump A min flow valve).

4) Open 10RHR-99A (RHR keep-full pump A disch to RHR loop A isol valve).

5) Place circuit breaker 71MCC-153-OE1 (10P-2A(M) RHR keep-full pump A motor) in ON.

6) Ensure closed 10MOV-39A.

7) Place circuit breaker 71MCC-153-OD1 (10MOV-39A RHR loop A torus cooling isol valve) in ON.

g. Place RHR Loop A keep-full in service per Section G of OP-13E.

h. Place DS-1PCIA07 (power disconnect switch for 10MOV-18) in OFF.

i. Lock DS-1PCIA07 (power disconnect switch for 10MOV-18) in OFF.

j. Verify IS-1PCIA07 (control circuit isolation switch for 10MOV-18) is in REMOTE.

k. Lock IS-1PCIA07 (control circuit isolation switch for 10MOV-18) in REMOTE.

CAUTION

Before securing Shutdown Cooling, decay heat levels should be evaluated such that they are low enough to allow sufficient time to complete the required evolution(s) without allowing boiling to occur, or that other systems are available with adequate decay heat removal capacity.

F.2 **RHR Loop B Shutdown Cooling Shutdown**

CAUTION

10MOV-16B is protective tagged closed. Running an RHR pump for greater than 1 minute without establishing flow could damage pump.

F.2.1 **IF** an RWR loop will be placed in service when shutdown cooling is secured,
THEN ensure reactor coolant temperature is **LESS THAN** 140°F to allow opening the RWR pump discharge valve to warm the loop without performing ST-26K.

F.2.2 Close LPCI OUTBD INJ VLV 10MOV-27B.

F.2.3 Ensure the following RHR pumps are stopped:

- RHR PMP 10P-3B
- RHR PMP 10P-3D

F.2.4 Close LPCI INBD INJ VLV 10MOV-25B.

F.2.5 Open LPCI OUTBD INJ VLV 10MOV-27B.

↓EXPA.3.2
F.2.6 Ensure closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15B
- SHUTDOWN CLG SUCT VLV 10MOV-15D

F.2.7 **IF** RHRSW Loop B operation <u>is not</u> required,
THEN shut down RHRSW Loop B as follows:

a. Close RHRSW DISCH VLV FROM HX B 10MOV-89B.

b. Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

CAUTION

If RPV water level is less than 234.5 inches with no forced core recirculation, reactor coolant temperature indications could be invalid due to insufficient natural circulation.

F.2.8 **IF** RWR Pumps 02-2P-1A and 02-2P-1B are shutdown, **AND** will remain shutdown,
THEN raise RPV water level to **ABOVE** 234.5 inches on REFUEL WTR LVL 02-3LI-86 at panel 09-4.

F.2.9 **IF** RHR Loop B is being shutdown for reasons that will not cause or require further RHR Loop B valve operation,
AND RHR Loop B will be returned to shutdown cooling per Subsection D.2,
THEN return to Subsection D.2 when RHR Loop B shutdown cooling mode operation is desired.

F.2.10 **IF** RHR Loop B is being shutdown for testing that will cause or require RHR Loop B valve operation; for example, surveillance testing that causes system isolation signals,
AND RHR Loop B is to be returned to the shutdown cooling mode per Subsection D.2,
THEN line up RHR Loop B as follows:

CAUTION

An extremely large drain path exists from the RPV to the torus if 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the associated 10MOV-13A, B, C, or D are open at the same time.

a. Close the following valves:

- SHUTDOWN CLG SUCT 10MOV-18
- SHUTDOWN CLG SUCT 10MOV-17

b. Throttle 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve) as necessary to maintain RHR Loop B full.

EXPA.3.7
F.2.11 **IF** both trains of RHR shutdown cooling are removed from service for the first time during a refueling outage,
THEN perform the following before rendering RHR shutdown cooling unavailable:

a. Ensure Fuel Pool Cooling System is operating.

b. Ensure only one train of DHR is operating, providing cooling to the core and spent fuel pool (gates removed, cavity flooded).

c. Verify availability of the redundant DHR train and backup emergency power.

d. Close the following valves to isolate RBCLC to fuel pool cooling heat exchangers:

- 15RBC-15A (fuel pool heat exch 19E-1A RBCLC return isol valve)
- 15RBC-15B (fuel pool heat exch 19E-1B RBCLC return isol valve)

NOTE: RHR shutdown cooling is required to be secured for at least 2 hours to ensure equilibrium thermal-hydraulic conditions are established in the core and spent fuel pool.

e. **WHEN** both trains of RHR shutdown cooling have been secured for at least 2 hours, verify DHR System is capable of assuming the spent fuel pool and core decay heat load by observing temperature trend on RHR WTR & HPCI TEMP 10TRS-131. Points 7, 8, and 9 should show a constant or decreasing temperature trend over the next 1 hour period.

NOTE: Temperature data need not be retained. SM log entry to state that DHR capability was demonstrated per OP-13D is sufficient documentation.

f. SM review temperature trend data to verify DHR capability and document in SM Log.

g. Open the following valves:

- 15RBC-15A
- 15RBC-15B

F.2.12 **IF** RHR Loop B shutdown cooling operation is no longer required,
THEN place RHR Loop B in the LPCI standby lineup as follows:

CAUTION

An extremely large drain path exists from the RPV to the torus if 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the associated 10MOV-13A, B, C, or D are open at the same time.

a. Close the following valves:

- SHUTDOWN CLG SUCT 10MOV-18
- SHUTDOWN CLG SUCT 10MOV-17

↓EXPA.3.2
b. Ensure closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15B
- SHUTDOWN CLG SUCT VLV 10MOV-15D

c. Ensure open the following valves:

- HX B BYP VLV 10MOV-66B
- HX B OUTLET VLV 10MOV-12B

d. Release protective tags and line up the following components in the position/status specified below:

1) Place circuit breaker 71MCC-163-OH3 (10MOV-16B RHR loop B min flow isol valve) in ON.

2) Open MIN FLOW VLV 10MOV-16B.

e. Open the following valves:

- TORUS SUCT VLV 10MOV-13B
- TORUS SUCT VLV 10MOV-13D

(Step F.2.12 continued on next page)

F.2.12 (Cont)

f. Release protective tags and line up the following components in the position/status specified below:

NOTE: The following components are listed in the correct order to restore components to service.

1) Ensure RHR KEEP-FULL PMP 10P-2B control switch is in STOP.

2) Open the following valves:

- 10RHR-91B (RHR keep-full pump B suct isol valve)
- 10RHR-91D (RHR keep-full pump B suct isol valve)

3) Open 10RHR-96B (RHR keep-full pump B min flow valve).

4) Open 10RHR-99B (RHR keep-full pump B disch to RHR loop B isol valve).

5) Place circuit breaker 71MCC-163-OE1 (10P-2B(M) RHR keep-full pump B motor) in ON.

6) Ensure closed 10MOV-39B.

7) Place circuit breaker 71MCC-163-OF1 (10MOV-39B RHR loop B torus cooling isol valve) in ON.

g. Place RHR Loop B keep-full in service per Section G of OP-13E.

h. Place DS-1PCIA07 (power disconnect switch for 10MOV-18) in OFF.

i. Lock DS-1PCIA07 (power disconnect switch for 10MOV-18) in OFF.

j. Verify IS-1PCIA07 (control circuit isolation switch for 10MOV-18) is in REMOTE.

k. Lock IS-1PCIA07 (control circuit isolation switch for 10MOV-18) in REMOTE.

G. SPECIAL PROCEDURES

NOTE: Subsections G.1 through G.8 are independent procedures and may be performed in any order, provided steps within each subsection are performed in the order written.

TABLE OF CONTENTS

SUBSECTION		PAGE
G.1	**Filling Shutdown Cooling Suction Piping**	56
G.2	**RHR Loop A Flush**	60
G.3	**RHR Loop B Flush**	62
G.4	**Pumping Reactor Coolant to Reactor Building Equipment Drain Discharge Header With RHR Loop A in Shutdown Cooling**	64
G.5	**Pumping Reactor Coolant to Reactor Building Equipment Drain Discharge Header With RHR Loop B in Shutdown Cooling**	67
G.6	**Responding to RHR Loop A Pump Motor Winding High Temperature Alarm While in Shutdown Cooling Mode**	70
G.7	**Responding to RHR Loop B Pump Motor Winding High Temperature Alarm While in Shutdown Cooling Mode**	72
G.8	**Shutdown Cooling Operation From Outside Control Room**	74

G.1 **Filling Shutdown Cooling Suction Piping**

G.1.1 **IF** shutdown cooling has been out of service **LESS THAN OR EQUAL TO** 24 hours, **THEN** perform the following:

a. **IF** Shift Manager desires that piping be filled using condensate transfer, **THEN** fill piping per Step G.1.3.

b. Fill piping per Step G.1.4.

COMA.2.2
G.1.2 **IF** shutdown cooling has been out of service **GREATER THAN** 24 hours, **THEN** perform the following:

a. Fill piping per Step G.1.3.

b. Fill piping per Step G.1.4.

G.1.3 Fill piping using condensate transfer as follows:

a. Establish communication between the drywell entrance and the Control Room.

b. Ensure closed SHUTDOWN CLG SUCT 10MOV-18.

c. Ensure open SHUTDOWN CLG SUCT 10MOV-17.

d. Connect a hose to 10RHR-441 (RHR SDC high point vent valve) as follows:

1) Connect hose to 10RHR-441.

2) Secure other end of hose to floor drain or other suitable container.

e. Open 10RHR-440 (RHR SDC high point vent valve).

f. Open 10RHR-258 (RHR shutdown cooling outer cond xfer isol valve).

(Step G.1.3 continued on next page)

G.1.3 (Cont)

CAUTIONS

RHR Pump Suction Relief Valves 10RV-41A and C are set to lift at 115 psig.

RHR Shutdown Cooling Safety Valve 10SV-40 is set to lift at 150 psig.

g. **IF** annunciator 09-4-3-22 SHUTDOWN COOL SUCT HDR PRESS HI alarms while performing Steps G.1.3.h and G.1.3.i,
THEN perform the following:

1) Close 10RHR-259.

2) **WHEN** annunciator 09-4-3-22 SHUTDOWN COOL SUCT HDR PRESS HI clears, crack open 10RHR-259 to establish a small amount of condensate flow.

h. Crack open 10RHR-259 (RHR shutdown cooling inner cond xfer isol valve) to establish a small amount of condensate flow.

i. Crack open 10RHR-441.

EXPA.3.5
j. Continue venting for at least 30 minutes.

k. **WHEN** water flowing from 10RHR-441 is free of air, close the following valves:

- 10RHR-440
- 10RHR-441
- 10RHR-258
- 10RHR-259

l. Disconnect hose from 10RHR-441.

m. Lock closed the following valves:

- 10RHR-440
- 10RHR-441

n. Install cap on 10RHR-441.

G.1.4 Fill piping by cycling 10MOV-17 and 10MOV-18 as follows:

a. **WHILE** performing Steps G.1.4.c through G.1.4.l, continuously monitor RPV water level using multiple indications.

NOTE: As RPV pressure is lowered, dissolved non-condensable gasses in the reference legs of RPV water level instrumentation could come out of solution. Industry experience has shown two types of observable indications of this phenomena:

- Alternating step increases and decreases in indicated RPV water level on one or more instruments (notching). Under notching conditions, the lower indicated level of the notch will be the most accurate indication of actual RPV water level.

- A sustained step increase in indicated RPV water level on one or more instruments. This step increase in indicated level could be of a significant magnitude and could be sustained for several hours.

The reference leg backfill system is designed to eliminate this problem.

b. **IF** an indication of non-condensable gases coming out of solution in reference legs is observed while cycling 10MOV-17 or 10MOV-18, **THEN** perform the following:

1) Ensure closed the following valves:

- SHUTDOWN CLG SUCT 10MOV-18

- SHUTDOWN CLG SUCT 10MOV-17

2) Notify Shift Manager.

c. Ensure closed the following valves:

- SHUTDOWN CLG SUCT 10MOV-18

- SHUTDOWN CLG SUCT 10MOV-17

(Step G.1.4 continued on next page)

G.1.4 (Cont)

d. Open and time SHUTDOWN CLG SUCT 10MOV-18.

e. Subtract 10 seconds from 10MOV-18 opening time. Use this time to mark the opening of 10MOV-17 in Step G.1.4.g.

f. Place control switch for SHUTDOWN CLG SUCT 10MOV-18 to CLOSE and start stopwatch.

g. **WHEN** the time calculated in Step G.1.4.e is reached, open and time SHUTDOWN CLG SUCT 10MOV-17.

h. Subtract 10 seconds from 10MOV-17 opening time. Use this time to mark the opening of 10MOV-18 in Step G.1.4.j.

i. Place control switch for SHUTDOWN CLG SUCT 10MOV-17 to CLOSE and start stopwatch.

j. **WHEN** the time calculated in Step G.1.4.h is reached, open SHUTDOWN CLG SUCT 10MOV-18.

k. **IF** the shutdown cooling suction line is suspected of being partially drained, **THEN** repeat Steps G.1.4.c through G.1.4.j until shutdown cooling line is full.

l. Open SHUTDOWN CLG SUCT 10MOV-17.

G.2 **RHR Loop A Flush**

G.2.1 **IF** RHR Loop A condensate transfer keep-full is in service,
THEN close 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve).

G.2.2 Stop RHR KEEP-FULL PMP 10P-2A.

G.2.3 Close the following valves:

- TORUS SUCT VLV 10MOV-13A
- TORUS SUCT VLV 10MOV-13C

G.2.4 Ensure closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15A
- SHUTDOWN CLG SUCT VLV 10MOV-15C

NOTE: Torus water level will rise when RHR Loop A is drained.

G.2.5 Monitor torus water level using multiple indications.

G.2.6 Open the following valves to drain RHR Loop A:

- RHR TEST TORUS CLG & SPRAY 10MOV-39A
- RHR TEST & TORUS CLG 10MOV-34A

G.2.7 **WHEN** RHR Loop A has drained for at least 5 minutes, close the following valves:

- RHR TEST TORUS CLG & SPRAY 10MOV-39A
- RHR TEST & TORUS CLG 10MOV-34A

G.2.8 Throttle open 10RHR-274 (RHR loop A containment spray keep-full cond xfer connection valve).

G.2.9 Verify pressure on RHR HDR PRESS LOOP A 10PI-120A returns to approximately 150 psig.

G.2.10 Direct RES Chemistry to perform the following:

a. Commence sampling RHR Loop A water.

b. Notify Shift Manager when RHR Loop A water conductivity is **LESS THAN** 3.0 $\mu mho/cm^2$.

G.2.11 Open RHR pump discharge drain valve for the RHR pump that is to be used for shutdown cooling:

RHR Pump	RHR Pump Discharge Drain Valve
10P-3A	10RHR-250A
10P-3C	10RHR-250C

G.2.12 Direct radwaste operator to perform the following:

a. Ensure lineup from reactor building equipment drain discharge header to desired radwaste storage tank is correct.

b. Ensure sufficient radwaste storage tank volume is available.

G.2.13 Open RHR DISCH TO RADW 10MOV-67.

G.2.14 **WHILE** performing Step G.2.15, maintain RHR Loop A pressure **GREATER THAN** 50 psig on RHR HDR PRESS LOOP A 10PI-120A.

G.2.15 Throttle open RHR DISCH TO RADW 10MOV-57.

G.2.16 Close HX A BYP VLV 10MOV-66A.

G.2.17 **WHEN** RES Chemistry reports water conductivity is acceptable, continue at next step.

G.2.18 Ensure closed the following valves:

- RHR DISCH TO RADW 10MOV-67
- RHR DISCH TO RADW 10MOV-57
- 10RHR-250A (RHR pump A disch drain valve)
- 10RHR-250C (RHR pump C disch drain valve)

G.2.19 Open HX A BYP VLV 10MOV-66A.

G.2.20 Ensure RHR Loop A keep-full level alarms are clear.

G.3 **RHR Loop B Flush**

G.3.1 **IF** RHR Loop B condensate transfer keep-full is in service,
THEN close 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve).

G.3.2 Stop RHR KEEP-FULL PMP 10P-2B.

G.3.3 Close the following valves:

- TORUS SUCT VLV 10MOV-13B
- TORUS SUCT VLV 10MOV-13D

G.3.4 Ensure closed the following valves:

- SHUTDOWN CLG SUCT VLV 10MOV-15B
- SHUTDOWN CLG SUCT VLV 10MOV-15D

NOTE: Torus water level will rise when RHR Loop B is drained.

G.3.5 Monitor torus water level using multiple indications.

G.3.6 Open the following valves to drain RHR Loop B:

- RHR TEST TORUS CLG & SPRAY 10MOV-39B
- RHR TEST & TORUS CLG 10MOV-34B

G.3.7 **WHEN** RHR Loop B has drained for at least 5 minutes, close the following valves:

- RHR TEST TORUS CLG & SPRAY 10MOV-39B
- RHR TEST & TORUS CLG 10MOV-34B

G.3.8 Throttle open 10RHR-260 (RHR loop B reactor head spray keep-full 10PCV-266 bypass valve).

G.3.9 Verify pressure on RHR HDR PRESS LOOP B 10PI-120B returns to approximately 150 psig.

G.3.10 Direct RES Chemistry to perform the following:

a. Commence sampling RHR Loop B water.

b. Notify Shift Manager when RHR Loop B water conductivity is **LESS THAN** 3.0 $\mu mho/cm^2$.

G.3.11 Open RHR pump discharge drain valve for the RHR pump that is to be used for shutdown cooling:

RHR Pump	RHR Pump Discharge Drain Valve
10P-3B	10RHR-250B
10P-3D	10RHR-250D

G.3.12 Direct radwaste operator to perform the following:

a. Ensure lineup from reactor building equipment drain discharge header to desired radwaste storage tank is correct.

b. Ensure sufficient radwaste storage tank volume is available.

G.3.13 Throttle open 10RHR-253 (RHR pumps B & D disch drains to equip sump isol valve) to accomplish both of the following:

- Control flow to reactor building equipment drain discharge header.
- Maintain RHR Loop B pressure **GREATER THAN** 50 psig on RHR HDR PRESS LOOP B 10PI-120B.

G.3.14 Close HX B BYP VLV 10MOV-66B.

G.3.15 **WHEN** RES Chemistry reports water conductivity is acceptable, continue at next step.

G.3.16 Ensure closed the following valves:

- 10RHR-253 (RHR pumps B & D disch drains to equip sump isol valve)
- 10RHR-250B (RHR pump B disch drain valve)
- 10RHR-250D (RHR pump D disch drain valve)

G.3.17 Open HX B BYP VLV 10MOV-66B.

G.3.18 Ensure RHR Loop B keep-full level alarms are clear.

G.4 **Pumping Reactor Coolant to Reactor Building Equipment Drain Discharge Header With RHR Loop A in Shutdown Cooling**

G.4.1 **IF** reactor coolant will be pumped to the reactor building equipment drain discharge header for **GREATER THAN** one shift,
THEN special condition tag the following control switches to indicate that RHR Pump Discharge Drain Valves 10RHR-250A and 10RHR-250C shall be verified closed before re-opening 10MOV-17 or 10MOV-18 following a shutdown cooling isolation:

- SHUTDOWN CLG SUCT 10MOV-18
- SHUTDOWN CLG SUCT 10MOV-17

G.4.2 **IF** water is to be pumped to a radwaste storage tank,
THEN direct radwaste operator to perform the following:

a. Ensure lineup from reactor building equipment drain discharge header to desired radwaste storage tank is correct.

b. Ensure sufficient radwaste storage tank volume is available.

G.4.3 **IF** water is to be pumped to the main condenser hotwell,
THEN perform the following:

a. Notify RES Radiation Protection.

b. Ensure main condenser vacuum is approximately zero inches Hg (atmospheric pressure).

c. Close 20RDW-101C (reactor building equipment drainage to waste collector isol valve).

d. Open 20RDW-101D (reactor building equipment drainage to condenser hotwell isol valve).

G.4.4 Open the RHR pump discharge drain valve:

RHR Pump	RHR Pump Discharge Drain Valve
10P-3A	10RHR-250A
10P-3C	10RHR-250C

G.4.5 Open RHR DISCH TO RADW 10MOV-67.

G.4.6 Throttle RHR DISCH TO RADW 10MOV-57 to control flow to reactor building equipment drain discharge header.

G.4.7 **WHILE** pumping reactor coolant, perform the following:

a. Monitor RPV water level using multiple indications.

NOTE: As RPV pressure is lowered, dissolved non-condensable gasses in the reference legs of RPV water level instrumentation could come out of solution. Industry experience has shown two types of observable indications of this phenomena:

- Alternating step increases and decreases in indicated RPV water level on one or more instruments (notching). Under notching conditions, the lower indicated level of the notch will be the most accurate indication of actual RPV water level.

- A sustained step increase in indicated RPV water level on one or more instruments. This step increase in indicated level could be of a significant magnitude and could be sustained for several hours.

The reference leg backfill system is designed to eliminate this problem.

b. **IF** an indication of non-condensable gases coming out of solution in reference legs is observed, **THEN** perform the following:

1) Close the following valves:

- RHR DISCH TO RADW 10MOV-57

- RHR DISCH TO RADW 10MOV-67

2) Notify Shift Manager.

G.4.8 **WHEN** desired RPV water level is reached, close RHR DISCH TO RADW 10MOV-57.

G.4.9 Close RHR DISCH TO RADW 10MOV-67.

G.4.10 Close RHR pump discharge drain valve for the RHR pump that was used to pump reactor coolant:

RHR Pump	RHR Pump Discharge Drain Valve
10P-3A	10RHR-250A
10P-3C	10RHR-250C

G.4.11 **IF** the reactor building equipment drain discharge header was lined up to main condenser hotwell, **THEN** perform the following:

a. Close 20RDW-101D (reactor building equipment drainage to condenser hotwell isol valve).

b. Open 20RDW-101C (reactor building equipment drainage to waste collector isol valve).

G.4.12 **IF** 10MOV-17 and 10MOV-18 control switches were special condition tagged per Step G.4.1, **THEN** release special condition tags.

G.5 **Pumping Reactor Coolant to Reactor Building Equipment Drain Discharge Header With RHR Loop B in Shutdown Cooling**

G.5.1 **IF** reactor coolant will be pumped to the reactor building equipment drain discharge header for **GREATER THAN** one shift,
THEN special condition tag the following control switches to indicate that RHR Pump Discharge Drain Valves 10RHR-250B and 10RHR-250D shall be verified closed before re-opening 10MOV-17 or 10MOV-18 following a shutdown cooling isolation:

- SHUTDOWN CLG SUCT 10MOV-18
- SHUTDOWN CLG SUCT 10MOV-17

G.5.2 **IF** water is to be pumped to a radwaste storage tank,
THEN direct radwaste operator to perform the following:

a. Ensure lineup from reactor building equipment drain discharge header to desired radwaste storage tank is correct.

b. Ensure sufficient radwaste storage tank volume is available.

G.5.3 **IF** water is to be pumped to main condenser hotwell,
THEN perform the following:

a. Notify RES Radiation Protection.

b. Ensure main condenser vacuum is approximately zero inches Hg (atmospheric pressure).

c. Close 20RDW-101C (reactor building equipment drainage to waste collector isol valve).

d. Open 20RDW-101D (reactor building equipment drainage to condenser hotwell isol valve).

G.5.4 Open the RHR pump discharge drain valve:

RHR Pump	RHR Pump Discharge Drain Valve
10P-3B	10RHR-250B
10P-3D	10RHR-250D

G.5.5 Throttle open 10RHR-253 (RHR pumps B & D disch drains to equip sump isol valve) to control flow to reactor building equipment drain discharge header.

G.5.6 **WHILE** pumping reactor coolant, perform the following:

a. Monitor RPV water level using multiple indications.

NOTE: As RPV pressure is lowered, dissolved non-condensable gasses in the reference legs of RPV water level instrumentation could come out of solution. Industry experience has shown two types of observable indications of this phenomena:

- Alternating step increases and decreases in indicated RPV water level on one or more instruments (notching). Under notching conditions, the lower indicated level of the notch will be the most accurate indication of actual RPV water level.

- A sustained step increase in indicated RPV water level on one or more instruments. This step increase in indicated level could be of a significant magnitude and could be sustained for several hours.

The reference leg backfill system is designed to eliminate this problem.

b. **IF** an indication of non-condensable gases coming out of solution in reference legs is observed, **THEN** perform the following:

1) Ensure closed the following valves:

- 10RHR-253 (RHR pumps B & D disch drains to equip sump isol valve)

- 10RHR-250B (RHR pump B discharge drain valve)

- 10RHR-250D (RHR pump D discharge drain valve)

2) Notify Shift Manager.

G.5.7 **WHEN** desired RPV water level is reached, close 10RHR-253.

G.5.8 Close the RHR pump discharge drain valve:

RHR Pump	RHR Pump Discharge Drain Valve
RHR Pump B	10RHR-250B
RHR Pump D	10RHR-250D

G.5.9 **IF** the reactor building equipment drain discharge header was lined up to main condenser hotwell, **THEN** perform the following:

a. Close 20RDW-101D (reactor building equipment drainage to condenser hotwell isol valve).

b. Open 20RDW-101C (reactor building equipment drainage to waste collector isol valve).

G.5.10 **IF** 10MOV-17 and 10MOV-18 control switches were special condition tagged per Step G.5.1, **THEN** release special condition tags.

CAUTION

Running an RHR pump in the shutdown cooling mode with motor winding temperature greater than or equal to 302°F could damage pump motor.

G.6 **Responding to RHR Loop A Pump Motor Winding High Temperature Alarm While in Shutdown Cooling Mode**

G.6.1 **WHILE** performing this subsection, monitor RHR pump motor winding temperature on the associated EPIC point:

RHR Pump	EPIC A
10P-3A	1107
10P-3C	1109

G.6.2 Start each of the following crescent area unit cooler fans by placing the associated control switch to START, spring return to normal (red flagged) at panel 09-75:

- WEST CRESC CLR 66UC-22J
- WEST CRESC CLR 66UC-22G

G.6.3 **IF** motor winding temperature reaches 302°F, **THEN** immediately shut down RHR Loop A per Subsection F.1.

G.6.4 **IF** the other RHR pump in RHR Loop A is available, **THEN** shift RHR pumps per Subsection D.3.

G.6.5 **IF** the other RHR pump in RHR Loop A <u>is not</u> available,
AND RHR Loop B is available for shutdown cooling,
THEN shift shutdown cooling loops from RHR Loop A to RHR Loop B per Subsection D.2.

NOTE: Motor winding temperature indication will respond slowly as flow is throttled. Flow should be throttled slowly to allow time for temperature indication to respond.

G.6.6 **WHILE** performing the following steps, do not throttle RHR Loop A flow to **LESS THAN** 7000 gpm.

G.6.7 **IF** the other RHR pump in RHR Loop A is not available,
AND RHR Loop B is not available for shutdown cooling,
THEN throttle RHR Pump Discharge Isolation Valve (10RHR-45A or 10RHR-45C) until one of the following conditions occurs:

- The running RHR pump motor winding temperature alarm clears,

OR

- RHR Loop A flow lowers to 7000 gpm.

G.6.8 **IF** RHR pump motor winding temperature alarm stays in alarm,
AND RHR Loop A flow is 7000 gpm,
THEN immediately shutdown RHR Loop A per Subsection F.1.

CAUTION

Running an RHR pump in the shutdown cooling mode with motor winding temperature greater than or equal to 302°F could damage pump motor.

G.7 **Responding to RHR Loop B Pump Motor Winding High Temperature Alarm While in Shutdown Cooling Mode**

G.7.1 **WHILE** performing this subsection, monitor RHR pump motor winding temperature on the associated EPIC point:

RHR Pump	EPIC A
10P-3B	1108
10P-3D	1110

G.7.2 Start each of the following crescent area unit cooler fans by placing the associated control switch to START, spring return to normal (red flagged) at panel 09-75:

- EAST CRESC CLR 66UC-22B
- EAST CRESC CLR 66UC-22D

G.7.3 **IF** motor winding temperature reaches 302°F, **THEN** immediately shut down RHR Loop B per Subsection F.2.

G.7.4 **IF** the other RHR pump in RHR Loop B is available, **THEN** shift RHR pumps per Subsection D.4.

G.7.5 **IF** the other RHR pump in RHR Loop B <u>is not</u> available,
AND RHR Loop A is available for shutdown cooling, **THEN** shift shutdown cooling loops from RHR Loop B to RHR Loop A per Subsection D.1.

NOTE: Motor winding temperature indication will respond slowly as flow is throttled. Flow should be throttled slowly to allow time for temperature indication to respond.

G.7.6 **WHILE** performing the following steps, do not throttle RHR Loop B flow to **LESS THAN** 7000 gpm.

G.7.7 **IF** the other RHR pump in RHR Loop B is not available,
AND RHR Loop A is not available for shutdown cooling,
THEN throttle RHR Pump Discharge Isolation Valve (10RHR-45B or 10RHR-45D) until one of the following conditions occurs:

- The running RHR pump motor winding temperature alarm clears,

OR

- RHR Loop B flow lowers to 7000 gpm.

G.7.8 **IF** RHR pump motor winding temperature alarm stays in alarm,
AND RHR Loop B flow is 7000 gpm,
THEN immediately shutdown RHR Loop B per Subsection F.2.

G.8 Shutdown Cooling Operation From Outside Control Room

CAUTION

An extremely large drain path exists from the RPV to the torus if 10MOV-17, 10MOV-18, and 10MOV-15A, B, C, or D and the corresponding 10MOV-13A, B, C, or D are open at the same time.

G.8.1 Refer to Subsection D.1 for RHR Loop A or D.2 for RHR Loop B, substituting the following methods for operating valves and pumps.

G.8.2 Align 10MOV-18 power supply as follows:

a. Place IS-1PCIA07 (control circuit isolation switch for 10MOV-18) in LOCAL.

b. Place IS-1PCIA07 (control circuit isolation switch for 10MOV-18) in REMOTE.

c. Place IS-1PCIA07 (control circuit isolation switch for 10MOV-18) in LOCAL.

d. Unlock DS-1PCIA07 (power disconnect switch for 10MOV-18).

e. Place DS-1PCIA07 (power disconnect switch for 10MOV-18) in ON.

G.8.3 Open or close 10MOV-18 as necessary at 71MCC-156 (Rx Bldg 344 southwest).

G.8.4 Operate other valves by valve handwheel.

G.8.5 Operate RHR and RHRSW pumps at 4160V switchgear.

H. **ATTACHMENTS**

1. REFERENCES
2. RPV WATER LEVEL RESPONSE DUE TO RHR OR RWR PUMP STARTUP

REFERENCES

1.0 **Performance References**

1.1 OP-13, Residual Heat Removal System*

1.2 OP-13C, RHR Service Water*

1.3 OP-13E, RHR Keep-Full*

1.4 OP-27, Recirculation System*

REFERENCES

2.0 **Developmental References**

2.1 FSAR Section 4.8

2.2 System Flow Diagrams:

- FM-20A Flow Diagram Residual Heat Removal Sheet 1
- FM-20B Flow Diagram Residual Heat Removal Sheet 2

2.3 Nuclear Safety Evaluation JAF-SE-91-061, Replacement Of Control Switches For 10MOV-12A/B; Justification For LPCI Throttling With 10MOV-27A/B And Shutdown Cooling Throttling With 10MOV-12A/B and 10MOV-66A/B

2.4 JSEM-91-082, QA Classification of 10MOV-57 and 10MOV-67

2.5 POT-10M, RHR Shutdown Cooling Throttling Valve Pre-op Test

2.6 POT-10P, RHR Shutdown Cooling Valve Throttling

2.7 Evaluation Of 10MOV-16A/B As Normally Open Valve

2.8 JAF-CALC-ELEC-00589, Rev 0, Removal of LPCI Inverters From EQ Program

2.9 Critique JOPS-CRT-93-011, RHR shutdown cooling was secured prior to reaching a reactor coolant temperature of < 140°F which precluded a timely RWR System startup.

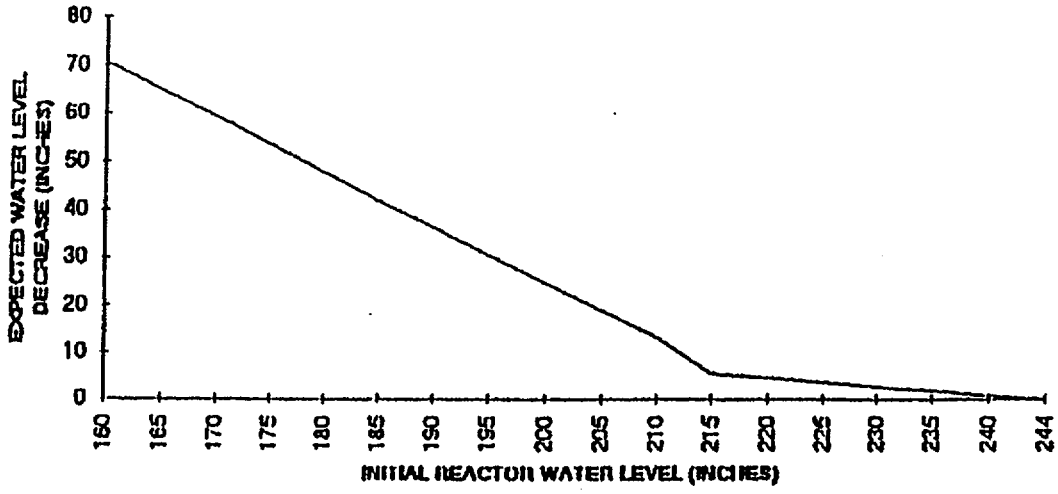
2.10 Modification M1-93-035, Reroute 02PS-128A Sensing Line, WR#097734

2.11 SEN 171, Inadvertent Reactor Coolant Heatup and Mode Change While Realigning Shutdown Cooling

2.12 Modification M1-97-021, Install High Point Vent in SDC Header

ATTACHMENT 2

RPV WATER LEVEL RESPONSE DUE TO RHR OR RWR PUMP STARTUP



THIS PAGE IS AN OVERSIZED DRAWING OR FIGURE,

THAT CAN BE VIEWED AT THE RECORD TITLED:

FM-25A, REV 60: FLOW DIAGRAM HIGH PRESSURE COOLANT INJECTION SYSTEM 23

WITHIN THIS PACKAGE...OR, BY SEARCHING USING THE DRAWING NUMBER:

FM-25A, Rev. 60

NOTE: Because of this page's large file size, it may be more convenient to copy the file to a local drive and use the Imaging (Wang) viewer, which can be accessed from the Programs/Accessories menu.

D-1

THIS PAGE IS AN OVERSIZED DRAWING OR FIGURE,

THAT CAN BE VIEWED AT THE RECORD TITLED:

FV-IJ, REV 12: DRYWELL AND SUPRESSION CHAMBER PENETRATIONS LOCATION & DETAILS SHEET 9

WITHIN THIS PACKAGE...OR, BY SEARCHING USING THE DRAWING NUMBER:

FV-IJ, Rev. 12

NOTE: Because of this page's large file size, it may be more convenient to copy the file to a local drive and use the Imaging (Wang) viewer, which can be accessed from the Programs/Accessories menu.

D-2

THIS PAGE IS AN OVERSIZED DRAWING OR FIGURE,

THAT CAN BE VIEWED AT THE RECORD TITLED:

C-5493, REV 2:

MODIFICATION TO EXISTING 24 INCH 0 HPCI LINE (X-219) FITZPATRICK

WITHIN THIS PACKAGE...OR, BY SEARCHING USING THE DRAWING NUMBER:

C-5493, Rev. 2

NOTE: Because of this page's large file size, it may be more convenient to copy the file to a local drive and use the Imaging (Wang) viewer, which can be accessed from the Programs/Accessories menu.

~~D-5~~

D-3

THIS PAGE IS AN OVERSIZED DRAWING OR FIGURE,

THAT CAN BE VIEWED AT THE RECORD TITLED:

FM-258, REV 29: FLOW DIAGRAM HPCI LUBE OIL SYSTEM SYSTEM 23

WITHIN THIS PACKAGE...OR, BY SEARCHING USING THE DRAWING NUMBER:

FM-258, Rev. 29

NOTE: Because of this page's large file size, it may be more convenient to copy the file to a local drive and use the Imaging (Wang) viewer, which can be accessed from the Programs/Accessories menu.

D-4

THIS PAGE IS AN OVERSIZED DRAWING OR FIGURE,

THAT CAN BE VIEWED AT THE RECORD TITLED:

FM-20A , REV 57:

FLOW DIAGRAM RESIDUAL HEAT REMOVAL SYSTEM 10, SHEET 1 OF 1

WITHIN THIS PACKAGE...OR, BY SEARCHING USING THE DRAWING NUMBER:

FM-20A , Rev. 57

NOTE: Because of this page's large file size, it may be more convenient to copy the file to a local drive and use the Imaging (Wang) viewer, which can be accessed from the Programs/Accessories menu.

D-5

THIS PAGE IS AN OVERSIZED DRAWING OR FIGURE,

THAT CAN BE VIEWED AT THE RECORD TITLED:

FM-20B , REV 55:

FLOW DIAGRAM RESIDUAL HEAT REMOVAL SYSTEM 10,

WITHIN THIS PACKAGE...OR, BY SEARCHING USING THE DRAWING NUMBER:

FM-20B , Rev. 55

NOTE: Because of this page's large file size, it may be more convenient to copy the file to a local drive and use the Imaging (Wang) viewer, which can be accessed from the Programs/Accessories menu.

[redacted]

D-6

THIS PAGE IS AN OVERSIZED DRAWING OR FIGURE,

THAT CAN BE VIEWED AT THE RECORD TITLED:

EOP-2 , REV 5:
RVP CONTROL

WITHIN THIS PACKAGE...OR, BY SEARCHING USING THE DRAWING NUMBER:

EOP-2 , Rev. 5

NOTE: Because of this page's large file size, it may be more convenient to copy the file to a local drive and use the Imaging (Wang) viewer, which can be accessed from the Programs/Accessories menu.

D-7

THIS PAGE IS AN OVERSIZED DRAWING OR FIGURE,

THAT CAN BE VIEWED AT THE RECORD TITLED: EOP-3 , REV 5: FAILURE TO SCRAM

WITHIN THIS PACKAGE...OR, BY SEARCHING USING THE DRAWING NUMBER: EOP-3 , Rev. 5

NOTE: Because of this page's large file size, it may be more convenient to copy the file to a local drive and use the Imaging (Wang) viewer, which can be accessed from the Programs/Accessories menu.

D-8

NEW YORK POWER AUTHORITY
JAMES A. FITZPATRICK NUCLEAR POWER PLANT
EOP AND SAOG SUPPORT PROCEDURE

ALTERNATE DEPRESSURIZATION USING SRVs FROM 02ADS-71*
EP-11
REVISION 0

REVIEWED BY: PLANT OPERATING REVIEW COMMITTEE

MEETING NO. NA DATE NA

APPROVED BY: _____ DATE 12/12/98
RESPONSIBLE PROCEDURE OWNER

APPROVED BY: _____ DATE 12/12/98
GENERAL MANAGER - OPERATIONS

EFFECTIVE DATE: 12-13-98

FIRST ISSUE ☒ FULL REVISION ☐ LIMITED REVISION ☐

REFERENCE USE	TSR
TECHNICAL	

VALIDATED PER AP-02.02

PERIODIC REVIEW DUE DATE 12/2003

FOR INFORMATION ONLY

FOR INFORMATION ONLY

REVISION SUMMARY SHEET

REV. NO.	CHANGE AND REASON FOR CHANGE
0	New procedure to provide instructions for depressurizing the RPV by operating SRVs from ADS Relief Valve Control Panel 02ADS-71. This procedure is used when directed by EOPs or SAOGs.

TABLE OF CONTENTS

SECTION		PAGE
1.0	**PURPOSE**	4
2.0	**PRECAUTIONS**	4
3.0	**PREREQUISITES**	4
4.0	**SPECIAL INSTRUCTIONS**	4
5.0	**PROCEDURE**	5
5.1	**Panel 02ADS-71 SRV Operation**	5
5.2	**Panel 02ADS-71 SRV Restoration**	8
6.0	**REFERENCES**	9
7.0	**REQUIREMENTS**	9
8.0	**ATTACHMENTS**	9

COM7.1.1

1.0 PURPOSE

To provide instructions for depressurizing the RPV by operating SRVs from ADS Relief Valve Control Panel 02ADS-71. This procedure is used when directed by EOPs or SAOGs.

2.0 PRECAUTIONS

Torus water level shall be greater than or equal to 5.5 feet to open an SRV. Opening an SRV with torus water level less than 5.5 feet will cause direct pressurization of the containment and possible containment damage.

3.0 PREREQUISITES

Performance of this procedure has been directed by the EOPs or SAOGs.

4.0 SPECIAL INSTRUCTIONS

4.1 Tools and materials for this procedure are located in the equipment cabinets near Remote Shutdown Panel 25RSP and panel 02ADS-71.

4.2 Notify RES Radiation Protection before entering the Reactor Building due to potentially changing radiological conditions.

5.0 **PROCEDURE** (✓)

5.1 **Panel 02ADS-71 SRV Operation**

5.1.1 Establish communication between the Control Room and ADS RELIEF VALVE CONTROL PANEL 02ADS-71. (_)

5.1.2 Verify the following control switches are in SOL DEENG at panel 02ADS-071:

- ADS & SRV 02RV-71C . . . . . (_)
- SRV 02RV-71F . . . . . (_)
- SRV 02RV-71J . . . . . (_)
- SRV 02RV-71L . . . . . (_)
- ADS & SRV 02RV-71B . . . . . (_)
- ADS & SRV 02RV-71E . . . . . (_)
- ADS & SRV 02RV-71H . . . . . (_)
- SRV 02RV-71K . . . . . (_)
- ADS & SRV 02RV-71A . . . . . (_)
- ADS & SRV 02RV-71D . . . . . (_)
- ADS & SRV 02RV-71G . . . . . (_)

5.1.3 Place CONTROL POWER breaker in ON at panel 02ADS-071. (_)

5.1.4 Verify in the control room annunciator 09-4-3-3 ADS REMOTE CNTRL PNL BKR CLOSED is in alarm. (_)

(✓)

5.1.5 **WHILE** performing Step 5.1.7, monitor the following to verify SRV operation:

- Annunciator 09-4-2-6 SRV SONIC MON ALARM HI (_)
- ADS & SRV ACOUSTIC MONITOR 02VMR-071 at panel 09-4 (_)
- White light at SRV control switch at panel 09-4 (_)
- Red SOL ENGD and green SOL DEENG lights at panel 02ADS-71 (_)

5.1.6 **IF** torus water level is **LESS THAN** 5.5 feet, **THEN** perform the following:

A. Notify SM or CRS that SRVs cannot be opened. (_)

B. Restore Panel 02ADS-71 per Subsection 5.2. (_)

5.1.7 Place one or more of the following control switches in SOL ENGD at panel 02ADS-71, as necessary to establish seven SRVs open:

- ADS & SRV 02RV-71C . . . . . (_)
- SRV 02RV-71F . . . . . (_)
- SRV 02RV-71J . . . . . (_)
- SRV 02RV-71L . . . . . (_)
- ADS & SRV 02RV-71B . . . . . (_)
- ADS & SRV 02RV-71E . . . . . (_)
- ADS & SRV 02RV-71H . . . . . (_)
- SRV 02RV-71K . . . . . (_)
- ADS & SRV 02RV-71A . . . . . (_)
- ADS & SRV 02RV-71D . . . . . (_)
- ADS & SRV 02RV-71G . . . . . (_)

(✓)

5.1.8 **IF** the total number of open SRVs is **GREATER THAN OR EQUAL TO** five, **THEN** notify the SM or CRS of the following:

- Number of open SRVs
- Use of other group 2 depressurization systems <u>is not</u> required.
- **IF** RPV Flooding is in progress, **THEN** the following valves are required to be closed:
  - MSIVs
  - Main steam line drains
  - RCIC steam isolation valves
  - HPCI steam isolation valves
  - Steam supply to RHR heat exchangers

(_)

5.1.9 **IF** any SRV was opened from panel 02ADS-71, **THEN** perform the following:

A. Close and lock panel door. (_)

B. Notify Control Room. (_)

C. <u>Do not</u> perform Subsection 5.2 until directed by the SM or CRS. (_)

5.1.10 **IF** <u>no</u> SRV was opened from panel 02ADS-71, **THEN** restore Panel 02ADS-71 per Subsection 5.2. (_)

5.2 **Panel 02ADS-71 SRV Restoration** (✓)

5.2.1 Ensure each of the following control switches is in SOL DEENG with green light on and red light off at panel 02ADS-071:

- ADS & SRV 02RV-71C . . . . . (_)
- SRV 02RV-71F . . . . . (_)
- SRV 02RV-71J . . . . . (_)
- SRV 02RV-71L . . . . . (_)
- ADS & SRV 02RV-71B . . . . . (_)
- ADS & SRV 02RV-71E . . . . . (_)
- ADS & SRV 02RV-71H . . . . . (_)
- SRV 02RV-71K . . . . . (_)
- ADS & SRV 02RV-71A . . . . . (_)
- ADS & SRV 02RV-71D . . . . . (_)
- ADS & SRV 02RV-71G . . . . . (_)

5.2.2 Place CONTROL POWER breaker in OFF at panel 02ADS-071. (_)

5.2.3 Verify annunciator 09-4-3-3 ADS REMOTE CNTRL PNL BKR CLOSED is clear. (_)

5.2.4 Close and lock door at panel 02ADS-71. (_)

5.2.5 Return equipment to cabinets near Remote Shutdown Panel 25RSP. (_)

5.2.6 Licensed Operator verify completion of Steps 5.2.1 through 5.2.5.

_____ / _____
Signature / Date

6.0 REFERENCES

6.1 Performance References

None

6.2 Developmental References

6.2.1 OP-68, Automatic Depressurization System*

6.2.2 ST-22B, Manual Safety Relief Valve Operation and Valve Monitoring System Functional Test (IST)*

7.0 REQUIREMENTS

7.1 Commitments

7.1.1 NRCI-94-03, JAFP-94-0175, ACTS Item 10946. Created EOP Support Procedures (EPs).

7.2 Validation

Revision 0 validated per AP-02.02.

8.0 ATTACHMENTS

None

NEW YORK POWER AUTHORITY
JAMES A. FITZPATRICK NUCLEAR POWER PLANT
OPERATING PROCEDURE

RHR - STEAM CONDENSING*
OP-13G
REVISION 4

APPROVED BY: Peter Abbott (ACTING) DATE 7/29/99
RESPONSIBLE PROCEDURE OWNER

EFFECTIVE DATE: 8-6-99

FIRST ISSUE ☐ FULL REVISION ☐ LIMITED REVISION ☒

REFERENCE USE	TSR
TECHNICAL	

PERIODIC REVIEW DUE DATE May 2004

FOR INFORMATION ONLY

REVISION SUMMARY SHEET

REV. NO.	CHANGE AND REASON FOR CHANGE
4	Revised procedure to reflect Design Change JD-99-099 and Safety Evaluation JAF-SE-99-014. These documents implement administrative controls that prohibit use of the steam condensing mode unless directed by the EOPs. These documents also allow re-installation of the motor heaters for the subject valves. Changes consist of deleting steps for installation and removal of motor heaters, updating Commitment A.2.1 and adding references. (PCR #2 dated 7/21/99) Updated description for 13A-DS67 in Steps D.1.8.c and D.2.8.c to match panel labels. (EC dated 7/26/99)

TABLE OF CONTENTS

SECTION		PAGE
A.	**REQUIREMENTS**	4
B.	**SYSTEM DESCRIPTION**	4
C.	**PLANT OPERATING REQUIREMENTS**	4
D.	**STARTUP**	5
E.	**NORMAL OPERATION**	18
F.	**SHUTDOWN**	19
G.	**SPECIAL PROCEDURES**	24
H.	**ATTACHMENTS**	24
	1. REFERENCES	25

A. REQUIREMENTS

A.1 Technical Specifications

Sections 3.5.A, 3.5.B, 3.5.F, and 3.5.G

A.2 Commitments

A.2.1 JD-99-099, JAF-SE-99-014, Deselection of 10MOV-21A&B from the JAF GL 89-10 Program. Deselecting valves from GL 89-10 program requires that 10MOV-21A & B can only be open when directed by EOPs or when Primary Containment is not required. Added precaution, steps, and commitments to alert operators.

A.3 Validation

Procedure developed from OP-13, Revision 74, which was validated per ODSO-35.

B. SYSTEM DESCRIPTION

Refer to Section B of OP-13.

C. PLANT OPERATING REQUIREMENTS

C.1 Prerequisites

RHR System valves, power supplies, and support systems are lined up per OP-13, with exceptions approved by Shift Manager.

C.2 Precautions

COMA.2.1

C.2.1 10MOV-21A and 10MOV-21B shall only be opened when directed by EOPs or when primary containment is not required.

C.2.2 RES Radiation Protection shall be notified before:

- RHR System piping is drained
- RHR System is started up in any mode of operation, unless an emergency exists

D. **STARTUP**

NOTE: Subsections D.1 and D.2 are independent procedures and may be performed in any order, provided steps within each subsection are performed in the order written.

TABLE OF CONTENTS

SUBSECTION		PAGE
D.1	**RHR Loop A Reactor Isolation (Steam Condensing Mode) Startup**	6
D.2	**RHR Loop B Reactor Isolation (Steam Condensing Mode) Startup**	12

NOTE: Only one RHR loop may be placed in the steam condensing mode to permit operating the other RHR loop in the torus cooling mode to support RCIC and SRV operation.

CAUTION

Subsection D.1 shall only be used when directed by EOPs.

D.1 **RHR Loop A Reactor Isolation (Steam Condensing Mode) Startup**

↓COMA.2.1

D.1.1 Verify EOPs have directed entry into Steam Condensing Mode.

D.1.2 Close circuit breaker 71MCC-151-OE2(CB) (10MOV-70A RHR heat exchanger A steam inlet isol valve).

D.1.3 Ensure the following controllers are in MAN with minimum output signal:

- RHR HX A PRESS CNTRL 10PIC-137A
- RHR HX A LVL CNTRL 10LIC-135A

D.1.4 Place 10PIC-147 RCIC SUCT PRESS CNTRL in AUTO with setpoint at 75 psig.

D.1.5 Verify closed HX A DRN TO RCIC 10MOV-36A.

D.1.6 Verify RCIC FLOW MODE SEL switch is in RCIC FLOW.

D.1.7 Verify STM COND VLV CNTRL 10PCV-69A & 10AOV-71A 10A-S41A switch is in CLOSE.

NOTE: RCIC TEST SPEED ADJUST potentiometer <u>will not</u> function if RCIC has a sealed-in auto-initiation signal.

D.1.8 **IF** RCIC has auto-initiated,
THEN reset RCIC initiation logic as follows:

a. Verify RPV water level is **GREATER THAN** 126.5 inches.

b. Depress INITIATION SIG RESET 13A-S18 pushbutton.

c. Verify white INITIATION SIGNAL 13A-DS67 light is off.

D.1.9 Ensure RCIC is in RPV pressure control mode per OP-19.

CAUTION

Operating RCIC at less than 2200 rpm could result in improper oil system operation and insufficient exhaust flow resulting in check valve banging or steam flow reversals.

D.1.10 **IF** RCIC FLOW CNTRL 13FIC-91 is in AUTO, **THEN** transfer RCIC flow control to test potentiometer as follows:

a. Ensure TEST SPEED ADJUST potentiometer is fully counterclockwise.

b. Place TURB TEST 13A-S20 switch in TURB TEST.

c. Place TEST PWR 13A-S21 switch in ON.

d. Verify TEST PWR ON 13A-DS56 light is on.

e. Control RCIC discharge flow with TEST SPEED ADJUST potentiometer.

D.1.11 Close HX A OUTLET VLV 10MOV-12A.

D.1.12 Close HX A INLET VLV 10MOV-65A.

D.1.13 Open HX A DRN TO TORUS 10MOV-21A.

D.1.14 Open HX A INBD VENT VLV 10MOV-166A.

D.1.15 Open HX A OUTBD VENT VLV 10MOV-167A.

D.1.16 Establish RHRSW flow as follows:

a. Verify one of the following alarms is clear:

- Annunciator 09-4-3-4 RHRSW A OR B DISCH LINE NOT FULL
- EPIC-D-134 10LS-105A

NOTE: Both RHRSW pumps are required to be running for maximum cooling.

b. Ensure at least one of the following RHRSW pumps is running:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

c. Throttle RHRSW DISCH VLV FROM HX A 10MOV-89A to establish 2500 to 4000 gpm per RHRSW pump.

D.1.17 Place STM COND VLV CNTRL 10PCV-69A & 10AOV-71A switch in OPEN.

D.1.18 Verify open the following valves:

- INBD STM SUPP VLV 23MOV-15
- OUTBD STM SUPP VLV 23MOV-16

D.1.19 Open RHR HX A STM INLET VLV 10MOV-70A.

CAUTIONS

Maximum rated RHR heat exchanger shell side temperature is 400°F.

Maximum rated RHR heat exchanger shell side pressure is 450 psig.

D.1.20 Lower RHR Heat Exchanger A water level to **LESS THAN OR EQUAL TO** 80% as follows:

a. Adjust RHR HX A PRESS CNTRL 10PIC-137A manual control knob to slowly pressurize RHR Heat Exchanger A to **LESS THAN** 50 psig on 10PI-137A.

b. **WHEN** RHR Heat Exchanger A pressure is **LESS THAN** 50 psig on 10PI-137A, adjust RHR HX A PRESS CNTRL 10PIC-137A manual control knob until output signal is zero%.

c. Adjust RHR HX A LVL CNTRL 10LIC-135A manual control knob until output signal is 100%.

d. Verify open RHR HX A DISCH TO TORUS OR RCIC 10AOV-71A.

e. **WHEN** RHR Heat Exchanger A water level is **LESS THAN OR EQUAL TO** 80%, place RHR HX A LVL CNTRL 10LIC-135A in automatic as follows:

1) Balance RHR HX A LVL CNTRL 10LIC-135A.

2) Place RHR HX A LVL CNTRL 10LIC-135A in AUTO.

D.1.21 Throttle HX A INBD VENT VLV 10MOV-166A to 10% open.

D.1.22 Adjust RHR HX A PRESS CNTRL 10PIC-137A manual control knob to slowly pressurize RHR Heat Exchanger A to **LESS THAN** 50 psig on 10PI-137A.

D.1.23 **IF** severe hammering of system components is heard, **THEN** isolate steam as follows:

a. Place STM COND VLV CNTRL 10PCV-69A & 10AOV-71A switch in CLOSE.

b. Close RHR HX A STM INLET VLV 10MOV-70A.

c. Investigate hammering before reopening steam valves.

D.1.24 Place RHR HX A PRESS CNTRL 10PIC-137A in automatic as follows:

a. Balance RHR HX A PRESS CNTRL 10PIC-137A.

b. Place RHR HX A PRESS CNTRL 10PIC-137A in AUTO.

D.1.25 Slowly raise RHR HX A PRESS CNTRL 10PIC-137A setpoint to 200 psig.

D.1.26 **WHILE** performing Steps D.1.28 through D.1.32, monitor heat exchanger pressure on 10PI-137A.

D.1.27 Slowly lower RHR HX A LVL CNTRL 10LIC-135A setpoint to establish desired heat transfer rate.

D.1.28 Monitor RHR Heat Exchanger A RHRSW differential temperature on EPIC display TTC.

CAUTION

Exceeding specified heat transfer rate limits could damage heat exchanger.

D.1.29 With RHRSW flow at 8000 gpm, control RHR Heat Exchanger A RHRSW differential temperature within the following limits:

a. **IF** pressure is 200 psig on 10PI-137A, **THEN** maintain differential temperature **LESS THAN OR EQUAL TO** 26.8°F.

b. **IF** pressure is **LESS THAN** 200 psig on 10PI-137A, **THEN** maintain differential temperature **LESS THAN OR EQUAL TO** 20.1°F.

NOTE: If condensate from RHR heat exchanger is to be returned to the RPV, RCIC turbine speed should be used to control RHR heat exchanger level.

D.1.30 **IF** RHR heat exchanger condensate is to be returned to the RPV using RCIC,
THEN perform the following:

NOTE: Acceptable water quality is normally reached within the first 30 minutes of operation.

a. Verify RHR Heat Exchanger A shell side water conductivity is **LESS THAN** 3µmho/cm² on 10CIS-95A, in West Crescent 242'.

CAUTIONS

Maximum rated RCIC suction pressure is 75 psig.

Maximum rated RCIC suction temperature is 140°F.

b. Open HX A DRN TO RCIC 10MOV-36A.

c. Close HX A DRN TO TORUS 10MOV-21A.

d. Adjust RCIC turbine speed using TEST SPEED ADJUST potentiometer to maintain desired heat exchanger level and heat transfer rate.

D.1.31 Establish heat exchanger level control using RCIC as follows:

a. Place RCIC FLOW MODE SEL switch in HX-A.

b. Adjust RCIC TEST SPEED ADJUST potentiometer to balance signal on RHR HX A LVL CNTRL 10LIC-135A.

c. **WHEN** RHR HX A LVL CNTRL 10LIC-135A is balanced, place TURB TEST 13A-S20 switch in NORMAL.

d. Verify RCIC SUCT PRESS CNTRL 10PIC-147 is controlling RCIC suction pressure at approximately 75 psig.

NOTE: Only one RHR loop may be placed in the steam condensing mode to permit operating the other RHR loop in the torus cooling mode to support RCIC and SRV operation.

CAUTION

Subsection D.2 shall only be used when directed by EOPs.

D.2 **RHR Loop B Reactor Isolation (Steam Condensing Mode) Startup**

↓COMA.2.1

D.2.1 Verify EOPs have directed entry into Steam Condensing Mode.

D.2.2 Close the following circuit breakers:

- 71MCC-161-OA2(CB) (10MOV-70B RHR heat exchanger B steam inlet isol valve)
- 71MCC-163-OG2(CB) (10MOV-36B RHR heat exchanger B return to RCIC isol valve)

D.2.3 Ensure the following controllers are in MAN with minimum output signal:

- RHR HX B PRESS CNTRL 10PIC-137B
- RHR HX B LVL CNTRL 10LIC-135B

D.2.4 Place 10PIC-147 RCIC SUCT PRESS CNTRL in AUTO with setpoint at 75 psig.

D.2.5 Verify closed HX B DRN TO RCIC 10MOV-36B.

D.2.6 Verify RCIC FLOW MODE SEL switch is in RCIC FLOW.

D.2.7 Verify STM COND VLV CNTRL 10PCV-69B & 10AOV-71B 10A-S41B switch is in CLOSE.

NOTE: RCIC TEST SPEED ADJUST potentiometer will not function if RCIC has a sealed-in auto-initiation signal.

D.2.8 **IF** RCIC has auto-initiated, **THEN** reset RCIC initiation logic as follows:

a. Verify RPV water level is **GREATER THAN** 126.5 inches.

b. Depress INITIATION SIG RESET 13A-S18 pushbutton.

c. Verify white INITIATION SIGNAL 13A-DS67 light is off.

D.2.9 Ensure RCIC is in RPV pressure control mode per OP-19.

CAUTION

Operating RCIC at less than 2200 rpm could result in improper oil system operation and insufficient exhaust flow resulting in check valve banging or steam flow reversals.

D.2.10 **IF** RCIC FLOW CNTRL 13FIC-91 is in AUTO, **THEN** transfer RCIC flow control to test potentiometer as follows:

a. Ensure TEST SPEED ADJUST potentiometer is fully counterclockwise.

b. Place TURB TEST 13A-S20 switch in TURB TEST.

c. Place TEST PWR 13A-S21 switch in ON.

d. Verify TEST PWR ON 13A-DS56 light is on.

e. Control RCIC discharge flow with TEST SPEED ADJUST potentiometer.

D.2.11 Close HX B OUTLET VLV 10MOV-12B.

D.2.12 Close HX B INLET VLV 10MOV-65B.

D.2.13 Open HX B DRN TO TORUS 10MOV-21B.

D.2.14 Open HX B INBD VENT VLV 10MOV-166B.

D.2.15 Open HX B OUTBD VENT VLV 10MOV-167B.

D.2.16 Establish RHRSW flow as follows:

a. Verify one of the following alarms is clear:

- Annunciator 09-4-3-4 RHRSW A OR B DISCH LINE NOT FULL
- EPIC-D-135 10LS-105B

NOTE: Both RHRSW pumps are required to be running for maximum cooling.

b. Ensure at least one of the following RHRSW pumps is running:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

c. Throttle RHRSW DISCH VLV FROM HX B 10MOV-89B to establish 2500 to 4000 gpm per RHRSW pump.

D.2.17 Place STM COND VLV CNTRL 10PCV-69B & 10AOV-71B switch in OPEN.

D.2.18 Verify open the following valves:

- INBD STM SUPP VLV 23MOV-15
- OUTBD STM SUPP VLV 23MOV-16

D.2.19 Open RHR HX B STM INLET VLV 10MOV-70B.

CAUTIONS

Maximum rated RHR heat exchanger shell side temperature is 400°F.

Maximum rated RHR heat exchanger shell side pressure is 450 psig.

D.2.20 Lower RHR Heat Exchanger B water level to **LESS THAN OR EQUAL TO** 80% as follows:

a. Adjust RHR HX B PRESS CNTRL 10PIC-137B manual control knob to slowly pressurize RHR Heat Exchanger B to **LESS THAN** 50 psig on 10PI-137B.

b. **WHEN** RHR Heat Exchanger B pressure is **LESS THAN** 50 psig on 10PI-137B, adjust RHR HX B PRESS CNTRL 10PIC-137B manual control knob until output signal is zero%.

c. Adjust RHR HX B LVL CNTRL 10LIC-135B manual control knob until output signal is 100%.

d. Verify open RHR HX B DISCH TO TORUS OR RCIC 10AOV-71B.

e. **WHEN** RHR Heat Exchanger B water level is **LESS THAN OR EQUAL TO** 80%, place RHR HX B LVL CNTRL 10LIC-135B in automatic as follows:

1) Balance RHR HX B LVL CNTRL 10LIC-135B.

2) Place RHR HX B LVL CNTRL 10LIC-135B in AUTO.

D.2.21 Throttle HX B INBD VENT VLV 10MOV-166B to 10% open.

D.2.22 Adjust RHR HX B PRESS CNTRL 10PIC-137B manual control knob to slowly pressurize RHR Heat Exchanger B to **LESS THAN** 50 psig on 10PI-137B.

D.2.23 **IF** severe hammering of system components is heard, **THEN** isolate steam as follows:

a. Place STM COND VLV CNTRL 10PCV-69B & 10AOV-71B switch in CLOSE.

b. Close RHR HX B STM INLET VLV 10MOV-70B.

c. Investigate hammering before reopening steam valves.

D.2.24 Place RHR HX B PRESS CNTRL 10PIC-137B in automatic as follows:

a. Balance RHR HX B PRESS CNTRL 10PIC-137B.

b. Place RHR HX B PRESS CNTRL 10PIC-137B in AUTO.

D.2.25 Slowly raise RHR HX B PRESS CNTRL 10PIC-137B setpoint to 200 psig.

D.2.26 **WHILE** performing Steps D.2.28 through D.2.32, monitor heat exchanger pressure on 10PI-137B.

D.2.27 Slowly lower RHR HX B LVL CNTRL 10LIC-135B setpoint to establish desired heat transfer rate.

D.2.28 Monitor RHR Heat Exchanger B RHRSW differential temperature on EPIC display TTC.

CAUTION

Exceeding specified heat transfer rate limits could damage heat exchanger.

D.2.29 With RHRSW flow at 8000 gpm, control RHR Heat Exchanger B RHRSW differential temperature within the following limits:

a. **IF** pressure is 200 psig on 10PI-137B, **THEN** maintain differential temperature **LESS THAN OR EQUAL TO** 26.8°F.

b. **IF** pressure is **LESS THAN** 200 psig on 10PI-137B, **THEN** maintain differential temperature **LESS THAN OR EQUAL TO** 20.1°F.

NOTE: If condensate from RHR heat exchanger is to be returned to the RPV, RCIC turbine speed should be used to control RHR heat exchanger level.

D.2.30 **IF** RHR heat exchanger condensate is to be returned to the RPV using RCIC,
THEN perform the following:

NOTE: Acceptable water quality is normally reached within the first 30 minutes of operation.

a. Verify RHR Heat Exchanger B shell side water conductivity is **LESS THAN** 3µmho/cm² on 10CIS-95B, in East Crescent 242'.

CAUTIONS

Maximum rated RCIC suction pressure is 75 psig.

Maximum rated RCIC suction temperature is 140°F.

b. Open HX B DRN TO RCIC 10MOV-36B.

c. Close HX B DRN TO TORUS 10MOV-21B.

d. Adjust RCIC turbine speed using TEST SPEED ADJUST potentiometer to maintain desired heat exchanger level and heat transfer rate.

D.2.31 Establish heat exchanger level control using RCIC as follows:

a. Place RCIC FLOW MODE SEL switch in HX-B.

b. Adjust RCIC TEST SPEED ADJUST potentiometer to balance signal on RHR HX B LVL CNTRL 10LIC-135B.

c. **WHEN** RHR HX B LVL CNTRL 10LIC-135B is balanced, place TURB TEST 13A-S20 switch in NORMAL.

d. Verify RCIC SUCT PRESS CNTRL 10PIC-147 is controlling RCIC suction pressure at approximately 75 psig.

E. **NORMAL OPERATION**

Refer to Section E of OP-13.

F. SHUTDOWN

NOTE: Subsections F.1 and F.2 are independent procedures and may be performed in any order, provided steps within each subsection are performed in the order written.

TABLE OF CONTENTS

SUBSECTION		PAGE
F.1	**RHR Loop A Steam Condensing Mode Shutdown**	20
F.2	**RHR Loop B Steam Condensing Mode Shutdown**	22

F.1 **RHR Loop A Steam Condensing Mode Shutdown**

F.1.1 Balance RCIC FLOW CNTRL 13FIC-91.

F.1.2 Place RCIC FLOW MODE SEL switch in RCIC FLOW.

F.1.3 Open HX A DRN TO TORUS 10MOV-21A.

F.1.4 Verify 10AOV-71A (RHR heat exchanger A outlet to torus or RCIC isol valve) controls RHR Heat Exchanger A level.

F.1.5 Close HX A DRN TO RCIC 10MOV-36A.

F.1.6 Raise RHR Heat Exchanger A water level to **GREATER THAN** 90% by adjusting RHR HX A LVL CNTRL 10LIC-135A setpoint to **GREATER THAN** 90%.

F.1.7 Adjust RHR HX A PRESS CNTRL 10PIC-137A setpoint to zero psig.

F.1.8 Close RHR HX A STM INLET VLV 10MOV-70A.

F.1.9 Place STM COND VLV CNTRL 10PCV-69A & 10AOV-71A switch in CLOSE.

F.1.10 Close HX A DRN TO TORUS 10MOV-21A.

F.1.11 Close HX A INBD VENT VLV 10MOV-166A.

F.1.12 Close HX A OUTBD VENT VLV 10MOV-167A.

F.1.13 Open HX A INLET VLV 10MOV-65A.

F.1.14 Verify open HX A BYP VLV 10MOV-66A.

F.1.15 Open HX A INBD VENT VLV 10MOV-166A.

F.1.16 Open HX A OUTBD VENT VLV 10MOV-167A.

F.1.17 **WHEN** RHR Heat Exchanger A has vented for several minutes, close the following valves:

- HX A INBD VENT VLV 10MOV-166A
- HX A OUTBD VENT VLV 10MOV-167A

F.1.18 Open HX A OUTLET VLV 10MOV-12A.

F.1.19 **IF** RHRSW Loop A operation is not required, **THEN** shut down RHRSW Loop A as follows:

a. Close RHRSW DISCH VLV FROM HX A 10MOV-89A.

b. Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1A
- RHRSW PMP 10P-1C

F.1.20 Place RHR Loop A keep-full in service per Section G of OP-13E.

F.1.21 Open circuit breaker 71MCC-151-OE2(CB) (10MOV-70A RHR heat exchanger A steam inlet isol valve).

F.2 **RHR Loop B Steam Condensing Mode Shutdown**

F.2.1 Balance RCIC FLOW CNTRL 13FIC-91.

F.2.2 Place RCIC FLOW MODE SEL switch in RCIC FLOW.

F.2.3 Open HX B DRN TO TORUS 10MOV-21B.

F.2.4 Verify 10AOV-71B (RHR heat exchanger B outlet to torus or RCIC isol valve) controls RHR Heat Exchanger B level.

F.2.5 Close HX B DRN TO RCIC 10MOV-36B.

F.2.6 Raise RHR Heat Exchanger B water level to **GREATER THAN** 90% by adjusting RHR HX B LVL CNTRL 10LIC-135B setpoint to **GREATER THAN** 90%.

F.2.7 Adjust RHR HX B PRESS CNTRL 10PIC-137B setpoint to zero psig.

F.2.8 Close RHR HX B STM INLET VLV 10MOV-70B.

F.2.9 Place STM COND VLV CNTRL 10PCV-69B & 10AOV-71B switch in CLOSE.

F.2.10 Close HX B DRN TO TORUS 10MOV-21B.

F.2.11 Close HX B INBD VENT VLV 10MOV-166B.

F.2.12 Close HX B OUTBD VENT VLV 10MOV-167B.

F.2.13 Open HX B INLET VLV 10MOV-65B.

F.2.14 Verify open HX B BYP VLV 10MOV-66B.

F.2.15 Open HX B INBD VENT VLV 10MOV-166B.

F.2.16 Open HX B OUTBD VENT VLV 10MOV-167B.

F.2.17 **WHEN** RHR Heat Exchanger B has vented for several minutes, close the following valves:

- HX B INBD VENT VLV 10MOV-166B
- HX B OUTBD VENT VLV 10MOV-167B

F.2.18 Open HX B OUTLET VLV 10MOV-12B.

F.2.19 **IF** RHRSW Loop B operation is not required, **THEN** shut down RHRSW Loop B as follows:

a. Close RHRSW DISCH VLV FROM HX B 10MOV-89B.

b. Ensure the following RHRSW pumps are stopped:

- RHRSW PMP 10P-1B
- RHRSW PMP 10P-1D

F.2.20 Place RHR Loop B keep-full in service per Section G of OP-13E.

F.2.21 Open the following circuit breakers:

- 71MCC-161-OA2(CB) (10MOV-70B RHR heat exchanger B steam inlet isol valve).
- 71MCC-163-OG2(CB) (10MOV-36B RHR heat exchanger B return to RCIC isol valve)

G. **SPECIAL PROCEDURES**

None

H. **ATTACHMENTS**

1. REFERENCES

ATTACHMENT 1

REFERENCES

1.0 **Performance References**

1.1 OP-13, Residual Heat Removal System*

1.2 OP-13E, RHR - Keep-Full*

2.0 **Developmental References**

2.1 OP-13, Revision 74, Residual Heat Removal System*

2.2 FSAR Section 4.8

2.3 FM-20A, FM-20B

2.4 Modification M1-92-334, Disabling of Valves 10MOV-36B, 70A, 70B, 15MOV-101, 102, and 103

2.5 JAF-SE-99-014, JD-99-099, RHR - Steam Condensing

James A. FitzPatrick NPP
Docket 50-333

System Component Evaluation Worksheet

EQUIPMENT DESCRIPTION

System: 10 RHR
Component ID: 10MOV-17(OP)
Description: VALVE MOTOR ACTUATOR
Manufacturer: LIMITORQUE Model No: SB-1-60
Function: RHR SDC OUTBOARD SUCTION ISOL

Accuracy Reqd: N/A
Accuracy Dem: N/A
Building: RB
Elevation: 272
Location: 3\R
Above Flood Level: YES
Qual Level: 50.49
Radiation Node: 2A
Orig. Mod No: F1-89-027
Accident Node: 272-2
Orig. PO No: 0210928-89

Parameter	ENVIRONMENT Specification		Qualif.	Documentation Ref. Specif.	Qual.	Qualification Method
Post-Acc Op Time	30 DAYS		180 DAYS	0.4	QDR: 3.01	Combination Test/Analysis
	Normal	Accident				
Temp. Deg F	85 90% 100 10%	260	340	0.1 1D	QDR: 3.01	Combination Test/Analysis
Pressure PSIA	14.70	22.40	134.7	0.2 2D	QDR: 3.01	Test/Analysis
Rel Hum %	40-70	100	100	0.3	QDR: 3.01	Test/Analysis
D/W Spray	N/A	N/A	YES	0.6	QDR: 3.01	Test/Analysis
	T.I.D.	Accident				
Rads	2.38E5	2.2E5	1.0E7	0.7	QDR: 3.01	Test/Analysis
Aging	40 Years		40.0 Yrs	0.5	QDR: 3.01	Test/Analysis

Maint Proc No: MP-56.1 Old File No: 3.20 Normal Rad Dose: 1.8E4

Notes:

Rev 6 Date 11/2/96 Reviewed MO Approved 12/02/96

James A. FitzPatrick NPP
Docket 50-333

System Component Evaluation Worksheet

EQUIPMENT DESCRIPTION

System: 10 RHR
Component ID: 10MOV-18(OP)
Description: VALVE MOTOR ACTUATOR
Manufacturer: LIMITORQUE Model No: SMB-0-40
Function: RHR SDC INBOARD SUCTION ISOL

Accuracy Reqd: N/A Accuracy Dem: N/A
Building: PC Above Flood Level: YES
Elevation: 272
Location: 3\T Qual Level: 50.49
Radiation Node: DW Orig. Mod No: M1-93-101
Accident Node: DRYWELL Orig. PO No: S-94-56807

Parameter	ENVIRONMENT Specification		Qualif.	Documentation Ref. Specif.	Qual.	Qualification Method
Post-Acc Op Time	30 DAYS		180 DAYS	0.4	QDR: 3.01	Combination Test/Analysis
	Normal	Accident				
Temp. Deg F	NOTE 1	335	340	0.1 1BF	QDR: 3.01	Combination Test/Analysis
Pressure PSIA	14.70	60	119.7	0.2 2BB	QDR: 3.01	Test/Analysis
Rel Hum %	40-55	100	100	0.3	QDR: 3.01	Test/Analysis
D/W Spray	N/A	YES	YES	0.6	QDR: 3.01	Test/Analysis
	T.I.D.	Accident				
Rads	9.15E7	8.8E7	2.0E8	0.7	QDR: 3.01	Test/Analysis
Aging	40 Years		40.0 Yrs	0.5	QDR: 3.01	Test/Analysis

Maint Proc No: MP-56.1 Old File No: 247 Normal Rad Dose: 3.5E6

Notes: 1. THE NORMAL TEMPERATURE PROFILE IS AS FOLLOWS:
2.5%-165F, 95%-135F, 2.5%-100F

Rev 5 Date 11/30/96 Reviewed MD Approved 12/03/96

General Electric Company

AFFIDAVIT

I, David J. Robare, being duly sworn, depose and state as follows:

(1) I am Technical Projects Manager, General Electric Company ("GE") and have been delegated the function of reviewing the information described in paragraph (2) which is sought to be withheld, and have been authorized to apply for its withholding.

(2) The information sought to be withheld is contained in a proprietary GE report GE-NE-T23-00737-01, entitled, *James A. Fitzpatrick Nuclear Power Plant Higher RHR Service Water Temperature Analysis,* Class III, dated August 1996.

(3) In making this application for withholding of proprietary information of which it is the owner, GE relies upon the exemption from disclosure set forth in the Freedom of Information Act ("FOIA"), 5 USC Sec. 552(b)(4), and the Trade Secrets Act, 18 USC Sec. 1905, and NRC regulations 10 CFR 9.17(a)(4), 2.790(a)(4), and 2.790(d)(1) for "trade secrets and commercial or financial information obtained from a person and privileged or confidential" (Exemption 4). The material for which exemption from disclosure is here sought is all "confidential commercial information", and some portions also qualify under the narrower definition of "trade secret", within the meanings assigned to those terms for purposes of FOIA Exemption 4 in, respectively, Critical Mass Energy Project v. Nuclear Regulatory Commission, 975F2d871 (DC Cir. 1992), and Public Citizen Health Research Group v. FDA, 704F2d1280 (DC Cir. 1983).

(4) Some examples of categories of information which fit into the definition of proprietary information are:

a. Information that discloses a process, method, or apparatus, including supporting data and analyses, where prevention of its use by General Electric's competitors without license from General Electric constitutes a competitive economic advantage over other companies;

b. Information which, if used by a competitor, would reduce his expenditure of resources or improve his competitive position in the design, manufacture, shipment, installation, assurance of quality, or licensing of a similar product;

c. Information which reveals cost or price information, production capacities, budget levels, or commercial strategies of General Electric, its customers, or its suppliers;

d. Information which reveals aspects of past, present, or future General Electric customer-funded development plans and programs, of potential commercial value to General Electric;

e. Information which discloses patentable subject matter for which it may be desirable to obtain patent protection.

The information sought to be withheld is considered to be proprietary for the reasons set forth in both paragraphs (4)a. and (4)b., above.

(5) The information sought to be withheld is being submitted to NRC in confidence. The information is of a sort customarily held in confidence by GE, and is in fact so held. The information sought to be withheld has, to the best of my knowledge and belief, consistently been held in confidence by GE, no public disclosure has been made, and it is not available in public sources. All disclosures to third parties including any required transmittals to NRC, have been made, or must be made, pursuant to regulatory provisions or proprietary agreements which provide for maintenance of the information in confidence. Its initial designation as proprietary information, and the subsequent steps taken to prevent its unauthorized disclosure, are as set forth in paragraphs (6) and (7) following.

(6) Initial approval of proprietary treatment of a document is made by the manager of the originating component, the person most likely to be acquainted with the value and sensitivity of the information in relation to industry knowledge. Access to such documents within GE is limited on a "need to know" basis.

(7) The procedure for approval of external release of such a document typically requires review by the staff manager, project manager, principal scientist or other equivalent authority, by the manager of the cognizant marketing function (or his delegate), and by the Legal Operation, for technical content, competitive effect, and determination of the accuracy of the proprietary designation. Disclosures outside GE are limited to regulatory bodies, customers, and potential customers, and their agents, suppliers, and licensees, and others with a legitimate need for the information, and then only in accordance with appropriate regulatory provisions or proprietary agreements.

(8) The information identified in paragraph (2), above, is classified as proprietary because it contains detailed results of analytical models, methods and processes, including computer codes, which GE has developed, discussed with the NRC, and applies in the Containment analyses for the BWR.

The development and approval of the containment computer code was achieved at a significant cost, on the order of several million dollars, to GE.

The development of the evaluation process along with the interpretation and application of the analytical results is derived from the extensive experience database that constitutes a major GE asset.

(9) Public disclosure of the information sought to be withheld is likely to cause substantial harm to GE's competitive position and foreclose or reduce the availability of profit-making opportunities. The information is part of GE's comprehensive BWR safety and technology base, and its commercial value extends beyond the original development cost. The value of the technology base goes beyond the extensive physical database and analytical methodology and includes development of the expertise to determine and apply the appropriate evaluation process. In addition, the technology base includes the value derived from providing analyses done with NRC-approved methods.

The research, development, engineering, analytical and NRC review costs comprise a substantial investment of time and money by GE.

The precise value of the expertise to devise an evaluation process and apply the correct analytical methodology is difficult to quantify, but it clearly is substantial.

GE's competitive advantage will be lost if its competitors are able to use the results of the GE experience to normalize or verify their own process or if they are able to claim an equivalent understanding by demonstrating that they can arrive at the same or similar conclusions.

The value of this information to GE would be lost if the information were disclosed to the public. Making such information available to competitors without their having been required to undertake a similar expenditure of resources would unfairly provide competitors with a windfall, and deprive GE of the opportunity to exercise its competitive advantage to seek an adequate return on its large investment in developing these very valuable analytical tools.

STATE OF CALIFORNIA)
) ss:
COUNTY OF SANTA CLARA)

David J. Robare, being duly sworn, deposes and says:

That he has read the foregoing affidavit and the matters stated therein are true and correct to the best of his knowledge, information, and belief.

Executed at San Jose, California, this 28TH day of JUNE 2000.

David J. Robare
General Electric Company

Subscribed and sworn before me this 28th day of June 2000.

VICKY D. SCHROER
Commission # 1224251
Notary Public - California
Santa Clara County
My Comm. Expires Jun 12, 2003

Notary Public, State of California

General Electric Company

AFFIDAVIT

I, David J. Robare, being duly sworn, depose and state as follows:

(1) I am Technical Projects Manager, General Electric Company ("GE") and have been delegated the function of reviewing the information described in paragraph (2) which is sought to be withheld, and have been authorized to apply for its withholding.

(2) The information sought to be withheld is contained in a proprietary GE report GE-NE-187-45-1191, entitled, *Fitzpatrick Power Uprate Impact Study, Engineering Report: Section 4.1, Containment Systems Evaluation for the James A. Fitzpatrick Nuclear Power Plant,* dated November 1991.

(3) In making this application for withholding of proprietary information of which it is the owner, GE relies upon the exemption from disclosure set forth in the Freedom of Information Act ("FOIA"), 5 USC Sec. 552(b)(4), and the Trade Secrets Act, 18 USC Sec. 1905, and NRC regulations 10 CFR 9.17(a)(4), 2.790(a)(4), and 2.790(d)(1) for "trade secrets and commercial or financial information obtained from a person and privileged or confidential" (Exemption 4). The material for which exemption from disclosure is here sought is all "confidential commercial information", and some portions also qualify under the narrower definition of "trade secret", within the meanings assigned to those terms for purposes of FOIA Exemption 4 in, respectively, Critical Mass Energy Project v. Nuclear Regulatory Commission, 975F2d871 (DC Cir. 1992), and Public Citizen Health Research Group v. FDA, 704F2d1280 (DC Cir. 1983).

(4) Some examples of categories of information which fit into the definition of proprietary information are:

a. Information that discloses a process, method, or apparatus, including supporting data and analyses, where prevention of its use by General Electric's competitors without license from General Electric constitutes a competitive economic advantage over other companies;

b. Information which, if used by a competitor, would reduce his expenditure of resources or improve his competitive position in the design, manufacture, shipment, installation, assurance of quality, or licensing of a similar product;

c. Information which reveals cost or price information, production capacities, budget levels, or commercial strategies of General Electric, its customers, or its suppliers;

d. Information which reveals aspects of past, present, or future General Electric customer-funded development plans and programs, of potential commercial value to General Electric;

e. Information which discloses patentable subject matter for which it may be desirable to obtain patent protection.

The information sought to be withheld is considered to be proprietary for the reasons set forth in both paragraphs (4)a. and (4)b., above.

(5) The information sought to be withheld is being submitted to NRC in confidence. The information is of a sort customarily held in confidence by GE, and is in fact so held. The information sought to be withheld has, to the best of my knowledge and belief, consistently been held in confidence by GE, no public disclosure has been made, and it is not available in public sources. All disclosures to third parties including any required transmittals to NRC, have been made, or must be made, pursuant to regulatory provisions or proprietary agreements which provide for maintenance of the information in confidence. Its initial designation as proprietary information, and the subsequent steps taken to prevent its unauthorized disclosure, are as set forth in paragraphs (6) and (7) following.

(6) Initial approval of proprietary treatment of a document is made by the manager of the originating component, the person most likely to be acquainted with the value and sensitivity of the information in relation to industry knowledge. Access to such documents within GE is limited on a "need to know" basis.

(7) The procedure for approval of external release of such a document typically requires review by the staff manager, project manager, principal scientist or other equivalent authority, by the manager of the cognizant marketing function (or his delegate), and by the Legal Operation, for technical content, competitive effect, and determination of the accuracy of the proprietary designation. Disclosures outside GE are limited to regulatory bodies, customers, and potential customers, and their agents, suppliers, and licensees, and others with a legitimate need for the information, and then only in accordance with appropriate regulatory provisions or proprietary agreements.

(8) The information identified in paragraph (2), above, is classified as proprietary because it contains detailed results of analytical models, methods and processes, including computer codes, which GE has developed, discussed with the NRC, and applies in the Containment analyses for the BWR.

The development and approval of the containment computer code was achieved at a significant cost, on the order of several million dollars, to GE.

The development of the evaluation process along with the interpretation and application of the analytical results is derived from the extensive experience database that constitutes a major GE asset.

(9) Public disclosure of the information sought to be withheld is likely to cause substantial harm to GE's competitive position and foreclose or reduce the availability of profit-making opportunities. The information is part of GE's comprehensive BWR safety and technology base, and its commercial value extends beyond the original development cost. The value of the technology base goes beyond the extensive physical database and analytical methodology and includes development of the expertise to determine and apply the appropriate evaluation process. In addition, the technology base includes the value derived from providing analyses done with NRC-approved methods.

The research, development, engineering, analytical and NRC review costs comprise a substantial investment of time and money by GE.

The precise value of the expertise to devise an evaluation process and apply the correct analytical methodology is difficult to quantify, but it clearly is substantial.

GE's competitive advantage will be lost if its competitors are able to use the results of the GE experience to normalize or verify their own process or if they are able to claim an equivalent understanding by demonstrating that they can arrive at the same or similar conclusions.

The value of this information to GE would be lost if the information were disclosed to the public. Making such information available to competitors without their having been required to undertake a similar expenditure of resources would unfairly provide competitors with a windfall, and deprive GE of the opportunity to exercise its competitive advantage to seek an adequate return on its large investment in developing these very valuable analytical tools.

STATE OF CALIFORNIA)
) ss:
COUNTY OF SANTA CLARA)

David J. Robare, being duly sworn, deposes and says:

That he has read the foregoing affidavit and the matters stated therein are true and correct to the best of his knowledge, information, and belief.

Executed at San Jose, California, this 28TH day of JUNE 2000.

David J. Robare
General Electric Company

Subscribed and sworn before me this 28th day of June 2000.

VICKY D. SCHROER
Commission # 1224251
Notary Public - California
Santa Clara County
My Comm. Expires Jun 12, 2003

Notary Public, State of California