CRYOTRAN USER'S MANUAL

VERSION 1.0

Glenn R. Cowgill
Analex Corporation
NASA Lewis Research Center
Cleveland, Ohio  44135

David J. Chato
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio  44135

Ehab Saad
Analex Corporation
NASA Lewis Research Center
Cleveland, Ohio  44135

December 14, 1989
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>GENERAL</td>
<td>2</td>
</tr>
<tr>
<td>THREE MAIN PARTS OF CRYOTRAN</td>
<td>3</td>
</tr>
<tr>
<td>Preprocessor Part</td>
<td>3</td>
</tr>
<tr>
<td>Execution Part</td>
<td>4</td>
</tr>
<tr>
<td>Postprocessor Part</td>
<td>4</td>
</tr>
<tr>
<td>USE OF HEAT TRANSFER NETWORK GENERATOR - SINDA</td>
<td>5</td>
</tr>
<tr>
<td>GEOMETRIC MODEL</td>
<td>5</td>
</tr>
<tr>
<td>Modeling</td>
<td>5</td>
</tr>
<tr>
<td>Spherical Wedge Nodes</td>
<td>5</td>
</tr>
<tr>
<td>Cylindrical Nodes</td>
<td>6</td>
</tr>
<tr>
<td>Regions</td>
<td>6</td>
</tr>
<tr>
<td>Constant Heat Source (Q)</td>
<td>7</td>
</tr>
<tr>
<td>Outside Constant Temperature</td>
<td>7</td>
</tr>
<tr>
<td>Heat Exchangers</td>
<td>7</td>
</tr>
<tr>
<td>SINDA Specific Input Data</td>
<td>8</td>
</tr>
<tr>
<td>SINDA Model File</td>
<td>8</td>
</tr>
<tr>
<td>USING CRYOTRAN</td>
<td>8</td>
</tr>
<tr>
<td>Accessing CryoTran</td>
<td>9</td>
</tr>
<tr>
<td>Aborting CryoTran</td>
<td>10</td>
</tr>
<tr>
<td>Input Error Checking</td>
<td>10</td>
</tr>
<tr>
<td>Getting Started</td>
<td>10</td>
</tr>
<tr>
<td>Program Output</td>
<td>11</td>
</tr>
<tr>
<td>INTEGRATING NEW PROGRAMS INTO CRYOTRAN</td>
<td>12</td>
</tr>
<tr>
<td>New Programs for Type 1 or Type 2 Problems</td>
<td>13</td>
</tr>
<tr>
<td>New Programs for Type 3 Problems</td>
<td>14</td>
</tr>
<tr>
<td>Modifications to Run a New Program on the VM Computer</td>
<td>14</td>
</tr>
<tr>
<td>Modifications to Run a New Program on the Cray Computer</td>
<td>15</td>
</tr>
<tr>
<td>PROGRAM INFORMATION</td>
<td>15</td>
</tr>
<tr>
<td>Numbering Conventions</td>
<td>16</td>
</tr>
<tr>
<td>Node Data Numbering Conventions</td>
<td>17</td>
</tr>
<tr>
<td>Conductor Numbering Convention</td>
<td>17</td>
</tr>
<tr>
<td>Material Numbering Conventions</td>
<td>18</td>
</tr>
<tr>
<td>CryoTran Specification Statements</td>
<td>19</td>
</tr>
<tr>
<td>Files Used to Run CryoTran</td>
<td>20</td>
</tr>
<tr>
<td>Miscellaneous Information</td>
<td>21</td>
</tr>
<tr>
<td>CONCLUDING REMARKS</td>
<td>21</td>
</tr>
</tbody>
</table>
**APPENDICES**

Appendix A - NVFILL, TARGET and CRYOCHIL User Guides .............................................. 22
Appendix B - Input Screens for Sample Problems .................................................................. 45
Appendix C - Sample Problems
  Part 1 - Sample Sphere Model with 5 Regions ................................................................. 69
  Part 2 - Sample Sphere Model, No Nodes in Regions 4 & 5 ........................................... 98
  Part 3 - Sample Case of SOLA-ECLIPSE ..................................................................... 120
  Part 4 - Output File (Model) to Run CSAM ................................................................. 136
Appendix D - Subroutine Descriptions .............................................................................. 140
Appendix E - CryoTran Program Listings
  Part 1 - CryoTran FORTRAN .................................................................................. 147
  Part 2 - CRYOSHPR FORTRAN ............................................................................. 199
  Part 3 - CRYOCYL FORTRAN .................................................................................. 222
  Part 4 - CRYOPLT FORTRAN .................................................................................. 263
  Part 5 - VM Exec Files .............................................................................................. 278
  Part 6 - Cray Script File to Execute SOLA-ECLIPSE .................................................. 299
  Part 7 - Cray Script File to Execute CSAM .................................................................. 302

**REFERENCES** ................................................................................................................. 305

**LIST OF TABLES**

Table 1 - Units Used in CryoTran ..................................................................................... 306
Table 2 - Fortran Files and Units ...................................................................................... 307

**LIST OF FIGURES**

Figure 1 - CryoTran - An Integrated Cryogenic Fluid System Model .......................... 308
Figure 2 - Flow Chart of CryoTran ................................................................................... 309
Figure 3 - Sketches Showing Sperical Wedge ................................................................. 310
Figure 4 - Sketches Showing Cylindrical Wedge ............................................................. 311
Figure 5 - Sketches Showing Possible and Configurations of Cylindrical Wedge ........... 312
Figure 6 - Node Numbering Convention ...................................................................... 313
Figure 7 - Sample Model Sphere 1 .................................................................................. 314
Figure 8 - Plot Sample of Cylinder ................................................................................. 315
Figure 9 - Flow Diagram of CryoTran .......................................................................... 316
SUMMARY

The development of cryogenic fluid management systems for space operation is a major portion of the efforts of the Cryogenic Fluids Technology Office (CFTO) at the NASA Lewis Research Center. Analytical models are a necessary part of experimental programs which are used to verify the results of experiments and are also used as a predictor for parametric studies. The CryoTran computer program is a bridge to obtain analytical results.

The object of CryoTran is to coordinate these separate analyses into an integrated framework with a user-friendly interface and a common cryogenic property database. CryoTran is an integrated software system designed to help solve a diverse set of problems involving cryogenic fluid storage and transfer in both ground and low-g environments.

CryoTran is designed to do the following here at NASA - Lewis Research Center (LeRC):
1. Generate models for the SINDA thermal analyzer.
2. Call on programs to be executed interactively on the front end computer, an IBM mainframe computer running the VM operating system, in line with CryoTran.
3. Generate files containing the Cray runstreams to be submitted to the large scale high speed computer, a Cray computer.
4. Execute analysis programs residing on the Cray.

CryoTran prompts the user for all the information necessary to accomplish the desired task. The input responses are tested for validity or feasibility whenever possible.

INTRODUCTION

As part of its effort to develop cryogenic fluid management systems for space operations, the Cryogenic Fluids Technology Office (CFTO) has been developing analytical models of cryogenic systems. Separate analyses have been conducted in the past by the CFTO and others (refs. 1 through 7).

CryoTran is a software system designed to solve a diverse set of problems involving cryogenic fluid storage, supply, and transfer. CryoTran is not constructed as one comprehensive general purpose program, but is instead divided into a set of modular programs for specific analyses. It is constructed with an open architecture which allows new modules to be added easily. User input is menu driven to facilitate usage. This approach makes CryoTran very versatile.

This report presents the general description of CryoTran, describes the types of problems that may be solved using the system, describes how to access and use the program, describes the output and the steps necessary to incorporate new analyses into the system.

GENERAL

CryoTran is designed to solve several types of problems. This initial release of CryoTran (Version 1.0) has the capability to run routines interactively to analyze tank chilldown fluid usage, select a chilldown wall temperature at which no-vent fill is feasible, or transiently analyze the no-vent fill process (refs. 5 and 7). It will set up heat transfer models to be solved using Systems Improved Numerical Differencing Analyzer (SINDA) (ref. 8) for two-dimensional (2-d) problems involving cryogenic storage in spherical tanks or cylindrical tanks with flat, spherical, or elliptical end caps. CryoTran also provides access to two large scale programs: CSAM (ref. 1) and SOLA-ECLIPSE (refs. 2 and 3).
THREE MAIN PARTS OF CRYOTRAN

CryoTran is divided into the following three main parts: preprocessor, execution and postprocessor. Its construction is modular to allow for expansion. A library of several solution routines is included with the system. Figures 1 and 2 show diagrams of CryoTran.

Preprocessor Part

The preprocessor is the driver of the CryoTran system. It accepts user input to define a specific problem. The preprocessor consists of the main program and subroutines, graphics routines and a database of material and thermal properties which are all maintained on the LeRC VM computer.

The preprocessor is written in the IBM FORTRAN 77 (version 2) programming language (ref. 9) and runs on the LeRC computer system running the IBM VM operating system, herein referred to as "the VM" or the "VM computer." The use of extensions and system-dependent code is kept to a minimum to make the code as transportable as possible. The FORTRAN code is generously commented to make it easy to follow. There are a few features (special purpose routines); however, that have been referenced from subroutine libraries at LeRC. These are noted in the code listings so that substitutions may be inserted at other installations.

The user is prompted for input via menus. This system checks whether or not the input is correct and feasible wherever possible and then the user input is put into a file which is saved for future recall.

The output of the preprocessor is a file which contains either the output from the execution of an interactive analysis or an input model to be submitted to the Cray for execution. If option 1 or 2 is chosen from the main menu, the file, called a "model file," will be a thermal model for SINDA. If option 3 is chosen from the main menu, followed by the option to access one of the large analysis codes resident on the Cray, the model file will contain the information to access the program on the Cray and, input data or a reference to the input data.
This model file has English text comments so that the user may make modifications to it prior to submitting it to the Cray for solution. Geometry plots of the SINDA input models may be obtained as part of the preprocessor.

**Execution Part**

The execution part of CryoTran executes one of the interactive programs within the system or submits the model file that was generated by the preprocessor to the Cray for execution. This model file contains job control language (JCL) and input data to an analysis program or a SINDA thermal model. This file is then submitted to the high-speed computer (Cray) for execution. The output of this execution is saved in a file and disposed to where the user may print it or save it for further processing in the postprocessor section.

**Postprocessor Part**

The postprocessor part of the system produces graphical results or analyzes the results obtained by the execution section. The postprocessor section consists of a plotting routine to plot the output of SOLA-ECLIPSE.

The no-vent fill and chilldown modules, which are the programs that run interactively, have been previously documented (refs. 4 to 7). Code details for these modules are not reproduced here. Portions of refs. 4 to 7 are included for convenience in Appendix A which may serve as user's guides for the no-vent fill and chilldown modules. Additionally, the large-scale programs, CSAM and SOLA-ECLIPSE, which are resident on the Cray computer and may be accessed through this system, are documented in refs. 1 through 3. See these references for the limitations of these programs and for input details.
USE OF HEAT TRANSFER NETWORK GENERATOR - SINDA

One of the types of analyses available through CryoTran is thermal/thermo analyses of spherical or cylindrical tanks using SINDA (ref. 8). Some SINDA analyses use models where both the tank and the inside of the tank are nodalized. Other analyses involve models of tanks where the inside of the tank is not nodalized and the thermo analysis problem inside the tank is solved by special purpose subroutines written to solve a particular method of cooling or tank fill procedure. These subroutines are called from the SINDA blocks.

GEOMETRIC MODEL

Modeling

CryoTran uses a two-dimensional analysis on spheres and cylinders. The two dimensions are radial (from the center of the tank outward to the outside surface) and circumferential (along the circumference of the tank from the south pole to the north pole). The tanks are nodalized using wedges.

Spherical Wedge Nodes

Figure 3 shows sketches depicting the spherical wedge nodes. The general model is a wedge with the vertex of the wedge at the center and radiating toward the outer surface at an angle of 1 radian. The nodes are formed by radial lines radiating from the vertex toward the outer surface and then by horizontal arcs at various distances from the vertex. This model looks like a wedge of an orange.

Since all of the radial lines radiate from the center, the thickness of the nodes on the sphere is variable and gets thicker as the distance of the nodes gets farther from the center of the sphere. An input value "n" specifies the number of these wedges along the circumference from the south pole to the north pole, each with angle theta, where theta=180/n.
Cylindrical Nodes

Figure 4 shows sketches indicating the geometry of cylinder nodes. In the case of the cylinder, the nodes are flat, four-sided nodes with two straight sides along the two radii describing the one radian angle and two sides which are circular arcs. On the cylinder the thickness of the nodes is determined by slices through the cylinder from the bottom to the top.

For the cylindrical tanks the ends may be open or closed with flat, spherical, or elliptical ends, or any combination of these. Figure 5 shows possible end configurations.

Regions

When the user is using the options that generate a SINDA model, the model will be somewhat tailored for a specific application. The geometric model is generated using five regions. As specified above, the geometric shapes are spherical wedges or cylindrical wedges. The five regions are as follows:

1. Tank Wall (shell) Required for all models.
2. Outside Layer 1 Region or shell covering the outside surface of the tank. It could be another material (e.g., insulation) or some type of coating layer. (Optional)
3. Outside Layer 2 Second outside layer on top of Region 2. (Optional)
4. Region 4 Portion of the tank interior that is adjacent to the Tank Wall. (Usually liquid or vapor.)
5. Region 5 Inside of the tank at the center. (Usually liquid or vapor.)

The inside of the tank may be made up of either one or two regions. If the user defines the inside of the tank to be a single region, this will be Region 4. If the user defines two regions for the inside of the tank, they will be Regions 4 and 5. The inside of the tank may be modeled using two regions in order to have two different mesh spacings in the radial direction or to have two different materials, etc. The concept of the regions can be seen in Figure 6 and in Figure 7, which shows a plot of a sample sphere model showing the 5 regions.
If Regions 4 and 5 are both liquid and the tank is not full, then a vapor ullage is assumed for the remainder of the volume. This ullage may be on top and flat if a 1-g case is designated or at the center and spherical for a low-g case. If Regions 4 and 5 are not both liquids, for example if Region 4 is a liquid and Region 5 is a vapor or solid, then it will be assumed that the tank is full with no ullage.

If Regions 4 and 5 are both liquids, the program checks to see if both are the same liquid. If the specified liquids in Regions 4 and 5 are different and the user determines that this is in error, the program then gives the user the option to change either or both of the materials.

**Constant Heat Source (Q)**

A constant heat source (Q) may be imposed on the outside surface of the model. The Q may be input as a constant Q per unit area BTU/(hr-ft²) or based on BTU/hr over the entire surface of the sphere.

**Outside Constant Temperature**

Two constant temperature boundary nodes may be defined outside the top layer of the model to simulate an outside temperature. The conductor paths of these nodes to the outer surface may either be convection (BTU/hr-ft²-°F) or radiation (°F).

**Heat Exchangers**

Up to ten heat exchangers may be inserted into the model. In these generated SINDA models, heat exchangers are simulated by constant temperature boundary nodes. These nodes are connected to the adjacent wall or fluid by conduction connectors. These heat exchangers may be placed anywhere in any of the five regions of the SINDA model. See Figure 7 which shows an example of the placement of three such heat exchangers. The system will ask the user for the temperature and position for each heat exchanger to be defined. To specify the position of a heat exchanger, the program asks for the following information:

**A.** The region number.

**B.** Which layer, of the region, the heat exchanger is on top of (counting from the outside, right to left, toward the center of the sphere or cylinder).
C. The theta angle (counting from the south pole where the heat exchanger begins).
D. The number of theta angles that the heat exchanger covers.

Figure 7 also shows the existence of constant temperature boundary nodes, 3 heat exchangers and a Q input into the outside surface.

**SINDA Specific Input Data**

The next input is the name of the SINDA execution routine to be used for the analysis. The SINDA analysis will be specified as transient or steady state. The choice of the execution routine also determines the subsequent input data that is required. The user has a choice of three SINDA execution routines. The STDSTL routine is used for steady state analyses. For the transient cases, the user has a choice of the CNFRDL or FWDBCK routines (see ref. 8 for the descriptions of these SINDA routines). The next input required are SINDA and user constants. The user is prompted for the required values.

**SINDA Model File**

When the model is generated a file named CRYOTRAN MODEL will be left on the user's disk. This file will consist of the Cray JCL followed by the SINDA input model. This file is available to the user to immediately submit it to the Cray, to edit it, to make modifications to the model and then submit it to the Cray, or to save it for future use.

**USING CRYOTRAN**

All of the CryoTran programs, the property database, and the procedure files are owned by the CRYOLIB "userid" on the VM computer at LeRC. (A "userid" is a code name to allow a user to log onto a computer system.) In order to use CryoTran, the user must have an active userid on both the VM and the Cray computers.
Accessing Cryotran

The user must first sign on to the VM computer and then may access CryoTran, by typing the following two commands *:

```
LINK CRYOLIB 200 222 RR
ACCESS 222 M
```

* Note: These two commands must be as typed as shown except for the user disk id (222). This user disk id may be any number not presently being used for any of the user's minidisks.

The user has now accessed the CRYOLIB disk containing the CryoTran program. Now type one of the following two commands *:

```
RUNCryo
RUNCry
```

* Note: Either will run CryoTran; however, the RUNCRY command will not load the graphics routines, thus, the program will be available for use earlier. Keep in mind that the geometry plot options are not available if you use the RUNCRY command.

Once the system is running, the user will be prompted by the system for all the input required to build the desired model or to execute the desired analysis program. The proper physical units will be specified as part of the prompt for each input where units are appropriate. The units used in the program are shown in Table 1.

Note!! When the program prompts the user for input, a response is mandatory.

Do not respond to any prompt by just pressing [Enter]. The user must type in a response to the prompt before pressing [Enter]. If [Enter] is pressed as the response to a prompt, the program will exit with many error messages. After clearing all of the error messages from the screen, the program will then automatically exit CryoTran and the user must restart CryoTran.
Aborting CryoTran

If the user becomes caught in an input loop or would like to abort the system, CryoTran may be exited from almost any input prompt by typing a "q" (quit).

Input Error Checking

CryoTran verifies user input in the following ways:

1. Questions which require replies of Y or N, etc. are checked for correct responses.
2. If an improper response is given to a question the program repeats the question to the user.
3. Upper and lower bound tests are made on integer input.
4. A test is made for the correct number of characters for some character data input.
5. A check is done on real number input to verify that extraneous alphabetic characters do not exist.

Getting Started

The user must first specify the type of problem to be performed by CryoTran. There are 3 types of problems. The first screen of CryoTran asks the user to:

| ENTER THE NUMBER FOR THE DESIRED PROBLEM TYPE |
|____________________________________________|
| THE PROBLEM TYPES ARE AS FOLLOWS:           |

1  -  THERMO/ THERMAL SINDA ANALYSIS ON A SPHERE.
2  -  THERMO/ THERMAL SINDA ANALYSIS ON A CYLINDER.
3  -  RUN A PRESTORED ANALYSIS PROGRAM.

The 3 options from this initial menu correspond to the 3 problem types respectively. They are:

1. type 1  Generates a SINDA model for a 2-d analysis of a sphere. The model generated will be a wedge of 1 radian, using the center axis of the sphere from the north pole to the south pole (see Figure 3).

2. type 2  Generates a 2-d model of a cylinder similar to the sphere wedge. The cylinder model may have the ends open, closed and flat, spherical or elliptical. The two end geometries may differ if so desired (see Figure 5).
3. type 3  Runs an analysis program that is already stored. Some of these programs are stored on the Cray computer. In this case the user will be prompted to make the input data available.

Another option under problem type 3 is the option to run a program that is in the CryoTran library on the front end computer (the VM). This will be run interactively on the VM computer within the CryoTran system.

**Note:** The user will see on the screens generated by the CryoTran program the variable name "ntyp," which will have the value 1, 2 or 3. This number designates the type of the problem (e.g., a type 1 problem is referred to as ntyp = 1).

As noted above, the user is prompted for all input data. The initial responses to the prompts tell the program which type of analysis the user wants, then prompts the user for the particular analysis program. After this has been established, the program then prompts the user for the proper information to build the SINDA model, if any, or to generate the proper file to execute the desired analysis program. CryoTran also prompts the user for any additional input data required by some programs. A sample of input screens for sample problems is shown in Appendix B.

**Program Output**

After a normal exit from CryoTran, some files will be left on the user's A disk and a geometry plot may be produced if the user has selected plot option 3.

The output of the CryoTran program consists of the following files that will be left on the user's A disk on the VM computer.

1. One file contains the input model named CRYOTRAN MODEL. This file contains the Cray JCL and input data.

   In the case of options 1 and 2 (SINDA model on a sphere or cylinder), the data consists of the SINDA model.

   If option 3 is chosen and the user picks an analysis program that is already stored on the Cray, the system will prompt the user for the necessary information to generate the CRYOTRAN MODEL file (consisting of the JCL and the input data for the program). The input data to the program may be stored on the Cray, stored on the VM system or
typed in at the terminal. If the input data is stored on the VM system, it must be on
the user’s A disk, set to LRECL 80 and RECFM F. See ref. 10 for definitions of
these two terms.

If the user wishes to retain the CRYOTRAN MODEL file for further use, the name
must be changed in the VM disk prior to running CryoTran again or the file will be
overwritten. Sample CRYOTRAN MODEL files are shown in Appendix C.

2. PROGRAM OUTPUT is an output file of an analysis routine executed interactively on
VM.

3. CRYOTRAN INPUTEKO A, is an echo of all the user input responses to the system
prompts as the model is being built. In order to use this file as input to the system at
a later time, this file must be renamed by the user prior to any further running of the
system to avoid it from being overwritten as it is being used.

This file, which has been renamed, may then be used as input to the system on a
subsequent run or it may be modified using the XEDIT command on the VM system.
To use this input echo file as input to the system (unit 5 input), type the following
instruction prior to typing the RUNCRYO command:

    FILEDEF FT05F001 DISK PREVIOUS INEKO

where the file name PREVIOUS INEKO is the name to which the user changed the
file CRYOTRAN INPUTEKO from a previous run.

4. If the user selects problem type 1 or 2 (sphere or cylinder SINDA model) and also
selects plot option 1, a file will be produced by the plotting package to be used later
to produce a geometry plot. The plotting package used in this program is the ISSCO
DISPLA program (ref 11). When the user requests a geometry plot, the DISPLA
package is called which produces a file named STD00001 DATA which is left on the
user’s A disk. The user must then type PLOTQA which is a VM exec to use the file
STD00001 DATA to actually produce the plots. Example geometry plots are shown in
Figures 7 and 8.

INTEGRATING NEW PROGRAMS INTO CRYOTRAN

To add programs to the system, some of the system subroutines must be modified and new
routines must be added to subroutine libraries.
New Programs for Type 1 or Type 2 Problems

When adding new programs for type 1 or 2 problems the following changes are necessary:

1. In subroutine MENU2 the data statements that must be changed are:

   array          ANALTi, where i=1, 2, 3, 4, 5 or 6
   array          REG45(j,i), i same as in 1
   variable       NALTi    i=1, 2, 3, 4, 5 or 6
   array          SPECIN(j,k) k=1,2 j=1,15

2. If there are going to be subroutines called from the execution, variables or output blocks, the names of these subroutines are put into the following DATA statements: EXEC1, EXEC2, VBL1, VBL2 and OUT in the system subroutine MENU2.

3. The source code of these subroutines must be in a file on the D disk of userid CRYOLIB. This file must be named CRYOLIB NAME1 where NAME1 is the name of the first subroutine called in the execution block and is also the same name that is put into EXEC1.

These routines that are put into the system for type 1 or 2 problems will usually contain code to solve the liquid problem inside the tank (Region 4). A SINDA model will be generated for Region 1 (Tank Wall) and possibly for Regions 2 and 3. Usually Region 4 will not be a part of the SINDA model and will not have a nodal mesh as in Regions 1, 2 or 3. The programmer of the code solving the problem in Region 4 must have a way to tie together the Region 4 code and the remainder of the SINDA model. The subroutines for this Region 4 problem will be called from the execution, variables and output blocks of SINDA.

The programmer must compute "inside tank" boundary temperatures and some sort of convection or heat transfer coefficient for use by SINDA for the heat transfer from the liquid or vapor to the wall. Further, the system supplies certain information to the programmer for use in computing these values. The information to and from these analysis subroutines is in COMMON blocks.

The COMMON blocks and ARRAYS that the programmer needs to link the thermo routines to SINDA are listed below. These common blocks are inserted by CryoTran into the variables and output blocks of the generated SINDA model.
The arrays in the above list that are labelled "Input" are values supplied to the programmer from SINDA for use in the thermo calculations. The arrays labelled "Output" are values that must be computed by the thermo routines and put into the indicated common blocks to interface the thermo computations with SINDA.

**New Programs for Type 3 Problems**

When a new analysis problem of type 3 is added to the system, the modifications to the system depend on whether the analysis program will run on the Cray or whether the analysis will run on VM.

**Modifications to Run a New Program on the VM Computer**

If the new program is to run interactively on VM, then the following modifications are necessary:

1. The main program of this new analysis code must be converted into a subroutine. The name of this subroutine may be any standard FORTRAN name (call it "name" for this discussion).

2. In subroutine MENU2 add data to:

   - array NALANS
   - variable NALNS
   - array MAINNM
   - array NSRUNM

   where short description up to 15 characters
   add 1 to this value
   name of main subroutine "name"
   which computer system analysis is to run on

   Where NN is the dimension NTHETA.
3. In subroutine VMINTR add the line:

   IF (NAN .EQ. i) CALL "name"

   where "name" is the name of the main subroutine and i is the position of "name"
   in array MAINNM

4. The source code for this program must be added to the CRYVMSUB FORTRAN file
   in userid CRYOLIB. This file is then recompiled and the file CRYVMSUB TEXT
   replaces the former one on the D disk of userid CRYOLIB.

**Modifications to Run a New Program on the Cray Computer**

If the new analysis program is to run on the Cray then:

1. The compiled program must reside on the Cray in userid CRYOLIB.

2. In subroutine MENU2 add data to:

   ```
   array variable array array
   NALANS NALNS MAINNM NSRUNM
   short description up to 15 characters add 1 to this value name of main subroutine "name"
   which computer system to run on
   ```

3. The main program of this new analysis code must be converted into a subroutine. The
   name of this subroutine may be any standard FORTRAN name (call it "name" for this
   discussion).

**PROGRAM INFORMATION**

The following general information about CryoTran will help systems programmers or users
make modifications to the system, add new programs to the system, and write subroutines
be called from SINDA models generated by the system.

The FORTRAN call, CALL CLEAR, to clear the screen, is used in CryoTran. This
routine is on the Amdahl/VM system at LeRC. The routine (CLEAR) is called from a
subroutine in the program named CLEARS (clear screen). On other systems that do not
have this routine, the user may comment out the call to CLEAR in subroutine CLEARS or
access a substitute routine. When CryoTran is used at LeRC the FTNLIB command is
executed prior to the load to access the routine. An alternate way to access the CLEARS
routine is to the ADDLIB command (a local LeRC command). See VM exec RUNCRYO in Appendix E part v.

The SYSCMD call, which is in the MAIN PROGRAM and in the DOJCL subroutine, is a local LeRC subroutine to perform VM JCL requests from inside a FORTRAN program. On other systems that do not have this routine, the user may comment out the call to SYSCMD in subroutine DOJCL or access a substitute routine.

**Numbering Conventions**

There are numbering conventions used in CryoTran to assist with the identification of node data, conductor data and materials for the various regions. These numbering conventions will also assist the programmer with new analysis programs that are to be integrated into CryoTran.
# Node Data Numbering Conventions

<table>
<thead>
<tr>
<th>Region Name</th>
<th>Description</th>
<th>Node Type</th>
<th>Base Node Number</th>
<th>Node Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Tank Wall</td>
<td>inner surface N1 layers</td>
<td>arithmetic</td>
<td>1000</td>
<td>1001, 1002, 1003, ... 1xxx</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3000</td>
<td>3001, 3002, 3003, ... 3xxx</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4000</td>
<td>4001, 4002, 4003, ... 4xxx</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5000</td>
<td>5001, 5002, 5003, ... 5xxx</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6000</td>
<td>6001, 6002, 6003, ... 6xxx</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7000</td>
<td>7001, 7002, 7003, ... 7xxx</td>
</tr>
<tr>
<td></td>
<td>outer surface N2 layers</td>
<td>arithmetic</td>
<td>8000</td>
<td>8001, 8002, 8003, ... 8xxx</td>
</tr>
<tr>
<td></td>
<td></td>
<td>diffusion</td>
<td>10000</td>
<td>10001, 10002, 10003, ... 10xxx</td>
</tr>
<tr>
<td>2 *</td>
<td>N3 layers</td>
<td>arithmetic</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>outer surface N4 layers</td>
<td>diffusion</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 *</td>
<td>inside tank * N5 layers</td>
<td>arithmetic</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>outer surface</td>
<td>diffusion</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 inside tank</td>
<td>when Region 4 = .false.</td>
<td>boundary</td>
<td>18000</td>
<td>18001, 18002, 18003, ... 18xxx</td>
</tr>
<tr>
<td>5 inside tank *</td>
<td>N4 layers</td>
<td>boundary</td>
<td>20000</td>
<td>20001, 20002, 20003, ... 20xxx</td>
</tr>
<tr>
<td></td>
<td>outside</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Atmosphere</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Where:
1. $xxx \leq 999$
2. * optional region
3. Where Base Node Number is the base and generated Node Numbers are incremented by 1 from the base. See Figure 6 to see the node numbering convention.

---

## Conductor Numbering Convention

Conductors start with number 1 and then are incremented by 1 for each conductor in the model.
Material Numbering Conventions

The material numbers from the materials database are 4 digit numbers with the following format:

knxx

Where:
1. k represents the material property number (shown below)
2. n represents the material type (shown below)
3. xx represents the number assigned to the material (within the material type)
4. [nxx] is the material number from the prompt screen when choosing the materials for each region

The material property numbers (k) are:

<table>
<thead>
<tr>
<th>Material Property Number (k)</th>
<th>Description</th>
<th>Symbol</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specific Heat * Density of material nxx</td>
<td>Cp*Rho</td>
<td>Btu/in³·°F</td>
</tr>
<tr>
<td>2</td>
<td>Specific Heat of material nxx</td>
<td>Cp</td>
<td>Btu/ibm·°F</td>
</tr>
<tr>
<td>3</td>
<td>Density of material nxx</td>
<td>Rho</td>
<td>lb/in³</td>
</tr>
<tr>
<td>4</td>
<td>Viscosity of material nxx</td>
<td>Mu</td>
<td>lb hr/in²</td>
</tr>
<tr>
<td>5</td>
<td>Enthalpy of material nxx</td>
<td>h</td>
<td>Btu/ib</td>
</tr>
<tr>
<td>6</td>
<td>Thermal Conductivity</td>
<td>k</td>
<td>Btu/hr-in·°F</td>
</tr>
</tbody>
</table>

where nxx is the material number from the prompt screen when choosing the materials for each region

The material group types (n) are:

<table>
<thead>
<tr>
<th>Material Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>liquid materials</td>
</tr>
<tr>
<td>2</td>
<td>solid materials</td>
</tr>
<tr>
<td>3</td>
<td>gaseous materials</td>
</tr>
</tbody>
</table>
CryoTran Specification Statements

The following is a list of variables defined in COMMON, LOGICAL and CHARACTER statements that occur in the subroutines of the CryoTran system. Not all of these common blocks, logical or character statements appear in each subroutine. The specific list of statements in each subroutine may be found in the program listing in Appendix E.

| COMMON/UNITS/ | MODU, INPEKO, ISCRCH, SINDA |
| COMMON/TITL/TITLE, TITLE0 |
| COMMON/GEOMTY/NTYP, NAN, GEOM(2) |
| COMMON/DATA/RIN, ROUT, NLAY, NTHETA, TIMEND, OUTPUT, FFLOW, TGAS, TLIQ, TWALL, DTIMEI, DRLXCA, ARLXCA, NLOOP |
| COMMON/REGION/NTHETA, NBETAS, BETA, RIN, TVOL, ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9), THKLAY(9), MATNMS(9), RGNMMS(9) |
| COMMON/SUBRTS/XCUT1, XCUT2, VBLBL1, VBLBL2, OUTBLK |
| COMMON/STUFFINHTT, PI, CONVY, CONVR, THETA0, DTHETA, NBASOS, ROUTSF, BNCOEF(2) |
| COMMON/UllAGE/ NLUL4, NLUL5, NTHU41, RINMHH, PCTFUL, RADULG, TVULFT, CT, LG(3), LIQVAP(3) |
| COMMON/HTXGRS/ NHX, HXTEMP(10), NRHX(10), NLHX(10), NTHHX(10), LNGTHX(10) |
| LOGICAL SPLIPT |
| LOGICAL SINDA |
| LOGICAL RGNS, VPCSHD |
| CHARACTER*6 XCUT1, XCUT2, VBLBL1, VBLBL2, OUTBLK, MAINNM |
| CHARACTER*50 TITLE0 |
| CHARACTER*80 TITLE |
| CHARACTER*8 GEOM |
| CHARACTER*16 MATNM2 |
| CHARACTER*25 RGNMMS |
| CHARACTER*6 LIQVAP |
| CHARACTER*16 MATNM |
| CHARACTER*1 CT, LG |
**Files Used to Run CryoTran**

When running CryoTran, the user links to disk 200 in the userid CRYOLIB and then accesses the M disk. The files necessary to execute CryoTran are on this M disk and are available to the user in read-only mode. They are: TEXT files (binary files resulting from compiling the FORTRAN source code), EXEC files (procedures written in REXX (ref. 12)), material property files and a source code file. The following list contains the files located on the M disk:

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRYOTRAN TEXT</td>
<td>Binary file of the main program and general system routines</td>
</tr>
<tr>
<td>CRYOSPHR TEXT</td>
<td>Binary file of subroutines pertaining to a sphere</td>
</tr>
<tr>
<td>CRYOCYL TEXT</td>
<td>Binary file of subroutines pertaining to a cylinder</td>
</tr>
<tr>
<td>CRYOPLOT TEXT</td>
<td>Binary plot routines for a sphere</td>
</tr>
<tr>
<td>CRYVMSUB TEXT</td>
<td>Binary analysis subroutines</td>
</tr>
<tr>
<td>ECLGRAPH TEXT</td>
<td>Binary plot routines for SOLA-ECLIPSE</td>
</tr>
<tr>
<td>SYSCMD TEXT</td>
<td>A LeRC system subroutine</td>
</tr>
<tr>
<td>RUNCRYO EXEC</td>
<td>VM Exec to put CryoTran into execution</td>
</tr>
<tr>
<td>RUNCRY EXEC</td>
<td>VM Exec to put CryoTran into execution without the plot programs</td>
</tr>
<tr>
<td>PLOTQA EXEC</td>
<td>VM Exec to produce geometry plots</td>
</tr>
<tr>
<td>DOECLPLT EXEC</td>
<td>VM Exec to execute the SOLA-ECLIPSE plot program</td>
</tr>
<tr>
<td>CRYOSUBS THWSE1</td>
<td>Subroutines for DeWitt type SINDA analysis</td>
</tr>
<tr>
<td>MATERIAL DATABASE</td>
<td>Database of material properties</td>
</tr>
<tr>
<td>H₂ TABLE</td>
<td>H₂ property data</td>
</tr>
<tr>
<td>N₂ TABLE</td>
<td>N₂ property data</td>
</tr>
<tr>
<td>O₂ TABLE</td>
<td>O₂ property data</td>
</tr>
</tbody>
</table>
Miscellaneous Information

Other information that may be of interest is:

A table of the FORTRAN files and corresponding unit numbers can be found in Table 2. A detailed flow diagram of CryoTran is given in Figure 9 (9-1 through 9-6). A short description of all the CryoTran subroutines is given in Appendix D and a FORTRAN listing of CryoTran along with some of the VM and the Cray script files is in Appendix E.

CONCLUDING REMARKS

This report presents Version 1.0 of CryoTran. It is a user-friendly modular system expected to be a dynamic and evolving program. It is intended that as new analyses become available they will be incorporated into the system.

Present capabilities include a tank chilldown fluid usage analysis, a transient no-vent fill procedure and a user interface to two large analysis programs, CSAM and SOLA-ECLIPSE. The program also generates SINDA models for 2-dimensional analyses of spherical and cylindrical tanks. These thermal models have the capability of multi-layer geometry and allow the user to include user-written subroutines to modify the analyses or expand them.

It is anticipated that future versions of CryoTran will include additional fill procedures and will be extend to 3-dimensional analyses.
APPENDIX A

NVFILL, TARGET and CRYOCHIL User Guides

[For CryoTran User Guide]
NVFILL is a computer model of the no-vent fill process. NVFILL approximates the no-vent fill by splitting the tank into four control volumes: ullage vapor, bulk liquid, liquid-vapor interface and tank wall. Convective heat and mass transfer relationships are used to control mass and heat transfer between the control volumes as well as apportioning the liquid inflow between the ullage and bulk volumes. The no-vent fill process is divided into two stages: a wall cooling stage where heat transfer to the tank wall is dominant and a fill stage where all the thermal energy has been removed from the wall.

Assumptions for the wall cooling stage are as follows. All liquid inflow is flashed to thermodynamic equilibrium on entering the tank as long as the tank pressure is less than the liquid saturation pressure. All remaining liquid after the initial flashing is vaporized upon striking the tank wall. The energy removed from the tank wall is equal to the energy necessary to vaporize this remaining liquid. Heat transfer from the wall to vapor, vapor to liquid and external environment to tank wall are assumed to be negligible.

Assumptions for the fill stage are as follows. The thermal energy of the wall has been removed in the wall cooling phase and can be neglected. The interface temperature is equal to the saturation temperature at the current tank pressure. The condensation rate at the interface is determined by the convective heat transport between the bulk liquid and the interface. Heat (but not mass) transfer to the gas is assumed negligible. The interface area is that of a sphere equal to the ullage volume.

To solve the differential equations of the no-vent fill process, a finite difference approximation is used. During the initial flashing stage the problem adds the mass inflow during the timestep to the ullage, calculates a new ullage density and internal energy, and then uses the density and enthalpy to determine a new ullage pressure. In the fill stage the problem is calculated in a two-step procedure. For the first process, the ullage is held at constant pressure while a mass transfer rate is calculated. For the second process, a new liquid volume is calculated from the mass transfer and current bulk liquid conditions. The ullage vapor is then compressed adiabatically to fill the remaining tank volume. The time step for both these processes combined is set to $10^3$ hours which is sufficiently small compared to the process rates for most cases of interest to insure a good approximation of the continuous mass transfer and compression processes.
User Interface

The code interactively prompts the user for the following input values:

1. Tank volume (ft³), volume of the tank being filled
2. Tank mass to volume ratio (lbm/ft³), mass of the tank divided by its volume
3. Liquid inflow rate (lbm/hr), the rate at which the filling liquid enters the tank.
4. Heat transfer coefficient (Btu/ft² hr °R), the convective heat transfer coefficient between the bulk liquid and the gas
5. Incoming liquid temperature (R), the temperature of the incoming liquid
6. Chilldown temperature (R), the tank wall temperature at the start of the fill

Once the user inputs have been specified, the code executes without further user interaction. The code terminates on one of the following criteria:

1. Tank 95% full (normal ending)
2. Tank pressure exceeds 60 psia
3. Tank fill time exceeds 8 hours
4. There is no vapor mass in the ullage.

Sample input screens are shown in table A-1.
The resultant output is shown in table A-2.
TARGET Description (excerpted from ref. 7)

The TARGET code is used to determine the maximum temperature from which the filling of a given tank can be initiated and subsequently filled to a specified pressure and fill level without venting. The main process is the transfer of the energy stored in the thermal mass of the tank walls into the inflowing liquid. This process is modeled by examining the end state of the no-vent fill process. This state is assumed to be a thermal equilibrium between the tank and the fluid which is well mixed and saturated at the tank pressure. No specific assumptions are made as to the processes or the intermediate thermodynamic states during the filling. It is only assumed that the maximum tank pressure occurs at the final state. As stated above, this assumption implies that, during the initial phases of the filling, the injected liquid must pass through the bulk vapor in such a way that it absorbs a sufficient amount of its superheat so that moderate tank pressures can be maintained. It is believed that this is an achievable design goal for liquid injection systems.

In reference 6, the mass-to-volume was found to be the key scaling parameter relating the target temperatures of prototype and subscaled model tanks. For a given tank material and identical operating conditions, this ratio is the determining factor of a tank’s target temperature. Therefore, the tank’s mass and volume are variable inputs to the TARGET code. The tank material, in addition to its mass and initial temperature, defines the thermal energy which is stored in the tank walls and must be absorbed by the liquid. Currently only 2219 Aluminum is used for the tank material.

The other main inputs required to run TARGET are the pressure and the tank filling percentage of the receiving tank at the completion of the no-vent fill. Since the liquid and vapor phases are assumed to be in equilibrium, the specific internal energy of each phase can be calculated from the final tank pressure. The final fill level quantifies the energy stored in the fluid by defining the liquid and vapor masses. A fluid mass balance equates the injected liquid to the final total mass of the liquid and vapor since there is no venting.

It was assumed in reference 6 that the heat flux into the tank was negligible during the fill operation. Inclusion of the tank heat flux, however, only requires a minor code modification once the heat flux and fill time are quantified. Thus, the only terms missing in the energy balance of the system (Equation 2 of ref. 6) are the enthalpy of the injected liquid and the initial wall temperature. The solution technique of TARGET is a simple incrementing of the initial tank temperature and solving for the required liquid enthalpy to satisfy the energy balance at each temperature. The initial temperature starts at 10 °R
above the liquid saturation temperature and then is increased by another 10 °R each time through a DO loop. This assists in maintaining small temperature ranges over which the ALCP subroutine must be called, thus reducing the errors. The trapezoidal rule is employed to increment the change in wall energy term, DELU. The enthalpy required to balance this wall energy change is then calculated and the corresponding fluid saturation pressure can be found from the data table. The actual output is the required pressure difference between the saturation conditions in the source and receiving tanks, i.e. the required liquid subcooling, for the given initial temperature.

**Code Specifics:**

1a. TARGET can be run with any fluid for which the user has a properties database. Currently it will only run for hydrogen, oxygen, and nitrogen since the pressure-enthalpy data sets have been created for these fluids only. TARGET will read the data into an array from logical unit 2 at the initiation of execution. Each time the array is scanned it will start at the lowest pressure (2 psia) and continue until the corresponding enthalpy value is greater than or equal to the required enthalpy. It should be noted that if GASP is not used for the other fluid properties these values may be inconsistent with those values obtained from an alternate properties subroutine due to the use of a different reference enthalpy value. Reference A-3 should be consulted in such an instance.

1b. Since the minimum saturation pressure in the data tables for the injected liquid state is 2 psia, the maximum pressure difference available for liquid subcooling is the maximum tank pressure less 2 psia. To avoid unnecessary execution, the code compares the calculated pressure difference to the maximum pressure difference and stops execution if there is an equivalence. The other normal termination of execution occurs if the initial temperature exceeds 540 °R in the incrementing DO loop. This would violate the upper limit for the ALCP subroutine.

2. Final fill level is to be entered in terms of a percentage, e.g., 95.0 not 0.95.

3. All output from the TARGET code goes to logical unit 1 which should be defined as data file, or could be directed to the user terminal.

4. All messages to the user are output to logical unit 1 which should be defined.
as data file, or could be directed to the user terminal.

5. There still exist occasional anomalies in the code execution which lead to erroneous values for the pressure difference at random enthalpy values for liquid hydrogen. It is not yet understood whether there is a problem in the data base or in the code execution. Erroneous data should simply be ignored until the problem can be corrected. Note, however, that negative values for the subcooling pressure difference are not erroneous; they merely indicate that subcooling is not required.

Input and output for an example run are shown in tables A-3 and A-4 respectively.
CRYOCHIL Description (excerpted from ref. 7)

The CRYOCHIL (CRYOgenic tank CHILldown) code was developed based on the analyses presented in reference 6. As previously stated, its primary function is to predict the optimum liquid charge to be injected for each of a series of charge-hold-vent chilldown cycles. This information can then be used with specified mass flow rates and valve response times to control a liquid injection system for tank chilldown operations. This will insure that the operations proceed quickly and efficiently.

Realizing that tank chilldown and no-vent fill operations are in essence part of the complete "thermodynamic" fill procedure, it is not surprising to find similar information being required as input for the analyses of each process. Again, the tank mass-to-volume ratio plays an important role; it determines the maximum charge which can be introduced to any tank regardless of its actual mass or volume. The maximum charge is found to decrease with increasing mass-to-volume ratios. Obviously the total mass required over a given temperature range is a direct function of the tank's thermal mass. Note, however, from the discussions of target temperatures that the lower mass-to-volume ratio tanks will have higher target temperatures and, thus, a lesser tank chilldown mass. CRYOCHIL will prompt the user for the input of the tank's mass, volume, initial temperature, and a target temperature.

Since the avoidance of tank overpressurization is a major concern, the tank's maximum pressure is an important input. Likewise, in order to calculate the available thermal capacity of given charge, the injected liquid enthalpy must be known. It is calculated from the input value of the supply tank saturation pressure, thus neglecting energy inputs from the transfer line, pressurization system, or transfer pumps. In an actual system the liquid state would be measured just prior to the liquid injection system.

Lastly, CRYOCHIL will prompt the user for a "vent stage pressure drop." This will cause the VENTDN subroutine to perform multiple vent cycles if any value less than the input maximum pressure is given. This is desirable because the venting can be more efficiently accomplished in small stages down to intermediate pressures at which the vapor can be held once again. This will allow for the isentropic expansion of the remaining vapor to cool the vapor and, consequently, the tank wall. Reference 3 used CRYOCHIL to demonstrate that a 23% fluid mass savings is possible for a quarter-scale model of an OTV tank when one 60 psi vent stage is replaced with many 1 psi vent stages. The limit to decreasing the magnitude of the vent stages becomes an infinite number of infinitely small stages which, of course, would be one slow vent stage at an optimized flow rate. This
optimum flow rate has not yet been calculated at Lewis.

CRYOCHIL prompts the user for each input and then echoes this input to an output file. The calculations begin with an evaluation of the liquid enthalpy of the injected liquid based on the specified saturation pressure in the supply tank prior to its being pressurized: this value remains constant for the entire test case.

The code next uses the ALCP subroutine to determine the specific heat of aluminum at the initial wall temperature for the given cycle.

At this point CRYOCHIL must make a guess at the tank wall temperature prior to the initiation of the venting process. This temperature and the temperature at the completion of venting are both unknown. Rather than making an arbitrary guess at this temperature, an educated guess is made based on a parametric evaluation of tank chilldowns over a range of tank mass-to-volume ratios using liquid hydrogen. Guesses for the beginning of subsequent chilldown cycles are made based on information retained from the preceding cycle. The use of these educated guesses has significantly reduced the execution time of the code from the time required with the use of arbitrary guesses.

Since the tank temperature prior to venting is unknown, an iterative solution technique can be used by solving equation 32 from reference 6 based on the previously guessed value. The algorithm is iterative because the properties of the vapor are evaluated at the tank's maximum pressure and a temperature which is 95% of the tank temperature. (Again, this assumption is working toward a prediction of fluid masses for an optimum chilldown cycle; actual spray systems should be designed to achieve this goal.) Recently, however, the code has been modified to more accurately account for the compressibility of the vapor by eliminating the ideal gas law (equation 30) from the analysis. What results is a less complex algorithm using equation 31 instead of equation 32, (ref. 6). Once this calculated value is within the user specified error band (entered via terminal input), the code proceeds to calculate the mass injected by multiplying the vapor density by the tank volume.

Having calculated the wall temperature prior to venting, CRYOCHIL next calls the VENTDN subroutine to model the venting according to the description of reference 6. Since the temperature of the wall at the conclusion of the venting is unknown, an iterative process is once again used. When finished, the VENTDN subroutine returns the wall temperature and the number of vent cycles for the specified vent stage magnitude. The last venting stage will always be down to 2 psia, regardless of its magnitude. Any effects
associated with venting the tank back to space vacuum are considered negligible. Note that VENTDN also uses the ALCP subroutine to evaluate wall specific heats and GASP to evaluate the fluid properties.

This charge-hold-vent procedure repeats for each chilldown cycle until the tank temperature, before or after venting, drops below the tank's target temperature. Since the objective of the modeling is to minimize the fluid consumption, this is undesirable. When this occurs, the CRYOCHIL code will call a subroutine, CALC, designed to chilldown the tank to within the user specified error band of the target temperature.

The CALC subroutine is not as straight forward as one might expect, even though the final temperature of this final cycle is known. This is because the cooling during the tank venting must be taken into account. Therefore, an iterative bisection algorithm, also known as "halving the interval," is used to calculate the tank temperature prior to venting until it is such that the resultant venting cools the tank to the target temperature. To accomplish this, CALC will call both the VENTDN and ALCP subroutines in addition to GASP. Since CALC should only be called when a complete "optimum" chilldown cycle is not possible or desirable, the maximum pressure due to the liquid evaporation should be below the tank's maximum pressure. For this reason, CALC will return the actual tank pressure to CRYOCHIL and print the value out with the number of vent cycles returned from VENTDN. If, however, the mass injected during this final cycle does not raise the tank pressure above 2 psia, VENTDN will not be called, and the number of vent cycles will be zero. This extra effort to hit the target temperature is made to assist in the conductance of trade studies with specified target temperatures. When CALC has found the proper injection mass to reach the target temperature, CRYOCHIL calculates the total mass injected and the total number of chilldown cycles.

Abnormal endings to CRYOCHIL can occur by several different ways: (1) One of the iterative solution techniques, in either the CRYOCHIL, VENTDN or CALC, exceeds the specified number of iterations, usually fifty; (2) the number of tank chilldown cycles exceeds fifty; (3) GASP returns to VENTDN or CALC with a thermodynamic state of the fluid different than expected; and (4) Any temperature in CRYOCHIL, VENTDN, or CALC is out of range for ALCP. If any of these failures occur, the execution will terminate and an error message will be given, usually specifying which failure stopped the execution. Abnormal endings 3 and 4 are most likely to occur if the specified target temperature is excessively low for a given tank and operation conditions.
Code Specifics:

CRYOCHIL also can be run for any fluid for which the property data exists. It is currently configured to run liquids hydrogen, oxygen, and nitrogen. This will be the first input made by the user. The user should not try to chill down a tank too close to the fluid's boiling point since the original assumption of all the liquid evaporating will be violated.

Input data is shown in table A-5.
Output data is shown in table A-6.
THIS PROGRAM DETERMINES THERMODYNAMIC PROPERTIES FOR PARAHYDROGEN FROM THE SUBROUTINE GASP

ENTER TANK VOLUME (FT**3) 
? 
23.6

ENTER TANK MASS TO VOLUME RATIO (LBM/FT**2) 
? 
3.0

ENTER LIQUID INFLOW RATE (LBM/HR) 
? 
500.

ENTER HEAT TRANSFER COEFFICIENT (BTU/FT**2 HR R) 
? 
40.0

ENTER INCOMING LIQUID TEMPERATURE (R) 
? 
36.6

ENTER CHILDDOWN TEMPERATURE (R) 
? 
102.5

TANK 95% FULL

FINAL PRESSURE = 29.04
FINAL GAS TEMPERATURE = 62.36
FINAL LIQUID TEMPERATURE = 38.81

Table A-1
NO VENT FILL TWO STEP MODEL
ADIABATIC COMPRESSION FOLLOWED BY ISOBARIC MASS TRANSFER

TANK VOLUME = 23.60 CU FT
LIQUID INFLOW RATE = 500.00 LB/HR
LIQUID TEMPERATURE = 36.60 R
INTERFACE LIQUID HEAT TRANSFER COEFFICIENT = 40.0000
CHILDOWN TEMP = 102.50 R
MASS TO VOLUME RATIO = 3.0000 LB/CU FT

<p>| TIME PRESS | GAS T | LIQ T | M GAS | V GAS | M LIQ | V LIQ |</p>
<table>
<thead>
<tr>
<th>HR</th>
<th>PSIA</th>
<th>%</th>
<th>R</th>
<th>R</th>
<th>LBM</th>
<th>CU FT</th>
<th>LBM</th>
<th>CU FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>2.00</td>
<td>0.00</td>
<td>102.50</td>
<td>36.60</td>
<td>0.09</td>
<td>23.60</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.001</td>
<td>9.56</td>
<td>0.00</td>
<td>72.95</td>
<td>36.60</td>
<td>0.59</td>
<td>23.60</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.002</td>
<td>14.79</td>
<td>0.00</td>
<td>61.81</td>
<td>36.60</td>
<td>1.09</td>
<td>23.60</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.003</td>
<td>20.09</td>
<td>0.00</td>
<td>58.34</td>
<td>36.60</td>
<td>1.59</td>
<td>23.60</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.004</td>
<td>22.68</td>
<td>0.00</td>
<td>57.33</td>
<td>36.60</td>
<td>1.84</td>
<td>23.60</td>
<td>0.25</td>
<td>0.00</td>
</tr>
</tbody>
</table>

WARNING BULK BOILING X=0.0050

<p>| TIME PRESS | GAS T | LIQ T | M GAS | V GAS | M LIQ | V LIQ |</p>
<table>
<thead>
<tr>
<th>HR</th>
<th>PSIA</th>
<th>%</th>
<th>R</th>
<th>R</th>
<th>LBM</th>
<th>CU FT</th>
<th>LBM</th>
<th>CU FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.005</td>
<td>22.40</td>
<td>0.76</td>
<td>56.74</td>
<td>39.30</td>
<td>1.82</td>
<td>23.42</td>
<td>0.77</td>
<td>0.18</td>
</tr>
<tr>
<td>0.006</td>
<td>22.42</td>
<td>1.23</td>
<td>56.48</td>
<td>38.41</td>
<td>1.82</td>
<td>23.31</td>
<td>1.26</td>
<td>0.29</td>
</tr>
</tbody>
</table>

WARNING BULK BOILING X=0.0020

<p>| TIME PRESS | GAS T | LIQ T | M GAS | V GAS | M LIQ | V LIQ |</p>
<table>
<thead>
<tr>
<th>HR</th>
<th>PSIA</th>
<th>%</th>
<th>R</th>
<th>R</th>
<th>LBM</th>
<th>CU FT</th>
<th>LBM</th>
<th>CU FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.007</td>
<td>22.27</td>
<td>1.76</td>
<td>56.45</td>
<td>39.22</td>
<td>1.80</td>
<td>23.19</td>
<td>1.79</td>
<td>0.61</td>
</tr>
<tr>
<td>0.008</td>
<td>22.34</td>
<td>2.74</td>
<td>56.33</td>
<td>38.79</td>
<td>1.80</td>
<td>23.07</td>
<td>2.28</td>
<td>0.53</td>
</tr>
<tr>
<td>0.009</td>
<td>22.33</td>
<td>3.24</td>
<td>56.38</td>
<td>38.87</td>
<td>1.79</td>
<td>22.95</td>
<td>2.80</td>
<td>0.65</td>
</tr>
<tr>
<td>0.010</td>
<td>22.33</td>
<td>3.74</td>
<td>56.37</td>
<td>38.87</td>
<td>1.78</td>
<td>22.83</td>
<td>3.30</td>
<td>0.77</td>
</tr>
<tr>
<td>0.011</td>
<td>22.32</td>
<td>3.74</td>
<td>56.37</td>
<td>38.87</td>
<td>1.77</td>
<td>22.72</td>
<td>3.81</td>
<td>0.88</td>
</tr>
<tr>
<td>0.012</td>
<td>22.31</td>
<td>4.24</td>
<td>56.37</td>
<td>38.87</td>
<td>1.76</td>
<td>22.60</td>
<td>4.32</td>
<td>1.00</td>
</tr>
<tr>
<td>0.013</td>
<td>22.31</td>
<td>4.74</td>
<td>56.36</td>
<td>38.86</td>
<td>1.75</td>
<td>22.48</td>
<td>4.83</td>
<td>1.12</td>
</tr>
<tr>
<td>0.014</td>
<td>22.31</td>
<td>5.24</td>
<td>56.35</td>
<td>38.86</td>
<td>1.74</td>
<td>22.36</td>
<td>5.34</td>
<td>1.24</td>
</tr>
<tr>
<td>0.015</td>
<td>22.29</td>
<td>5.74</td>
<td>56.34</td>
<td>38.86</td>
<td>1.73</td>
<td>22.25</td>
<td>5.85</td>
<td>1.35</td>
</tr>
<tr>
<td>0.016</td>
<td>22.28</td>
<td>6.24</td>
<td>56.34</td>
<td>38.86</td>
<td>1.72</td>
<td>22.13</td>
<td>6.36</td>
<td>1.47</td>
</tr>
<tr>
<td>0.017</td>
<td>22.28</td>
<td>6.74</td>
<td>56.33</td>
<td>38.85</td>
<td>1.71</td>
<td>22.01</td>
<td>6.87</td>
<td>1.59</td>
</tr>
<tr>
<td>0.018</td>
<td>22.27</td>
<td>7.24</td>
<td>56.32</td>
<td>38.85</td>
<td>1.71</td>
<td>21.89</td>
<td>7.38</td>
<td>1.71</td>
</tr>
<tr>
<td>0.019</td>
<td>22.26</td>
<td>7.74</td>
<td>56.32</td>
<td>38.85</td>
<td>1.70</td>
<td>21.77</td>
<td>7.89</td>
<td>1.83</td>
</tr>
<tr>
<td>0.020</td>
<td>22.26</td>
<td>8.24</td>
<td>56.31</td>
<td>38.85</td>
<td>1.69</td>
<td>21.66</td>
<td>8.40</td>
<td>1.94</td>
</tr>
<tr>
<td>0.021</td>
<td>22.25</td>
<td>8.74</td>
<td>56.30</td>
<td>38.84</td>
<td>1.68</td>
<td>21.54</td>
<td>8.91</td>
<td>2.06</td>
</tr>
<tr>
<td>0.022</td>
<td>22.25</td>
<td>9.24</td>
<td>56.30</td>
<td>38.84</td>
<td>1.67</td>
<td>21.42</td>
<td>9.42</td>
<td>2.18</td>
</tr>
<tr>
<td>0.023</td>
<td>22.24</td>
<td>9.74</td>
<td>56.29</td>
<td>38.84</td>
<td>1.66</td>
<td>21.30</td>
<td>9.93</td>
<td>2.30</td>
</tr>
<tr>
<td>0.024</td>
<td>22.24</td>
<td>10.24</td>
<td>56.29</td>
<td>38.84</td>
<td>1.65</td>
<td>21.18</td>
<td>10.44</td>
<td>2.42</td>
</tr>
<tr>
<td>0.025</td>
<td>22.24</td>
<td>10.73</td>
<td>56.28</td>
<td>38.84</td>
<td>1.64</td>
<td>21.07</td>
<td>10.95</td>
<td>2.53</td>
</tr>
<tr>
<td>0.026</td>
<td>22.24</td>
<td>11.23</td>
<td>56.28</td>
<td>38.83</td>
<td>1.63</td>
<td>20.95</td>
<td>11.46</td>
<td>2.65</td>
</tr>
<tr>
<td>0.027</td>
<td>22.23</td>
<td>11.73</td>
<td>56.28</td>
<td>38.83</td>
<td>1.62</td>
<td>20.83</td>
<td>11.97</td>
<td>2.77</td>
</tr>
<tr>
<td>0.028</td>
<td>22.22</td>
<td>12.23</td>
<td>56.27</td>
<td>38.83</td>
<td>1.61</td>
<td>20.71</td>
<td>12.48</td>
<td>2.89</td>
</tr>
<tr>
<td>0.029</td>
<td>22.22</td>
<td>12.73</td>
<td>56.27</td>
<td>38.83</td>
<td>1.60</td>
<td>20.60</td>
<td>12.98</td>
<td>3.00</td>
</tr>
<tr>
<td>0.030</td>
<td>22.22</td>
<td>13.23</td>
<td>56.27</td>
<td>38.83</td>
<td>1.59</td>
<td>20.48</td>
<td>13.49</td>
<td>3.12</td>
</tr>
<tr>
<td>0.031</td>
<td>22.22</td>
<td>13.73</td>
<td>56.26</td>
<td>38.83</td>
<td>1.58</td>
<td>20.36</td>
<td>14.00</td>
<td>3.24</td>
</tr>
<tr>
<td>0.032</td>
<td>22.21</td>
<td>14.23</td>
<td>56.26</td>
<td>38.82</td>
<td>1.57</td>
<td>20.24</td>
<td>14.51</td>
<td>3.36</td>
</tr>
<tr>
<td>0.033</td>
<td>22.21</td>
<td>14.73</td>
<td>56.26</td>
<td>38.82</td>
<td>1.57</td>
<td>20.12</td>
<td>15.02</td>
<td>3.48</td>
</tr>
<tr>
<td>0.034</td>
<td>22.21</td>
<td>15.23</td>
<td>56.26</td>
<td>38.82</td>
<td>1.56</td>
<td>20.01</td>
<td>15.53</td>
<td>3.59</td>
</tr>
<tr>
<td>0.035</td>
<td>22.21</td>
<td>15.73</td>
<td>56.25</td>
<td>38.82</td>
<td>1.55</td>
<td>19.89</td>
<td>16.04</td>
<td>3.71</td>
</tr>
<tr>
<td>0.036</td>
<td>22.21</td>
<td>16.22</td>
<td>56.25</td>
<td>38.82</td>
<td>1.54</td>
<td>19.77</td>
<td>16.55</td>
<td>3.83</td>
</tr>
<tr>
<td>0.037</td>
<td>22.20</td>
<td>16.72</td>
<td>56.25</td>
<td>38.82</td>
<td>1.53</td>
<td>19.65</td>
<td>17.06</td>
<td>3.95</td>
</tr>
<tr>
<td>0.038</td>
<td>22.20</td>
<td>17.22</td>
<td>56.25</td>
<td>38.82</td>
<td>1.52</td>
<td>19.54</td>
<td>17.57</td>
<td>4.06</td>
</tr>
<tr>
<td>0.039</td>
<td>22.20</td>
<td>17.72</td>
<td>56.25</td>
<td>38.81</td>
<td>1.51</td>
<td>19.42</td>
<td>18.08</td>
<td>4.18</td>
</tr>
<tr>
<td>0.040</td>
<td>22.20</td>
<td>18.22</td>
<td>56.25</td>
<td>38.81</td>
<td>1.50</td>
<td>19.30</td>
<td>18.59</td>
<td>4.30</td>
</tr>
</tbody>
</table>

Table A-2

(1 of 4)
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.041</td>
<td>22.20</td>
<td>18.72</td>
<td>56.25</td>
<td>38.81</td>
</tr>
<tr>
<td>0.042</td>
<td>22.20</td>
<td>19.22</td>
<td>56.25</td>
<td>38.81</td>
</tr>
<tr>
<td>0.043</td>
<td>22.20</td>
<td>19.72</td>
<td>56.25</td>
<td>38.81</td>
</tr>
<tr>
<td>0.044</td>
<td>22.20</td>
<td>20.22</td>
<td>56.24</td>
<td>38.81</td>
</tr>
<tr>
<td>0.045</td>
<td>22.20</td>
<td>20.72</td>
<td>56.24</td>
<td>38.81</td>
</tr>
<tr>
<td>0.046</td>
<td>22.20</td>
<td>21.21</td>
<td>56.24</td>
<td>38.80</td>
</tr>
<tr>
<td>0.047</td>
<td>22.20</td>
<td>21.71</td>
<td>56.24</td>
<td>38.80</td>
</tr>
<tr>
<td>0.048</td>
<td>22.20</td>
<td>22.21</td>
<td>56.25</td>
<td>38.80</td>
</tr>
<tr>
<td>0.049</td>
<td>22.20</td>
<td>22.71</td>
<td>56.25</td>
<td>38.80</td>
</tr>
<tr>
<td>0.050</td>
<td>22.20</td>
<td>23.21</td>
<td>56.25</td>
<td>38.80</td>
</tr>
<tr>
<td>0.051</td>
<td>22.20</td>
<td>23.71</td>
<td>56.25</td>
<td>38.80</td>
</tr>
<tr>
<td>0.052</td>
<td>22.20</td>
<td>24.21</td>
<td>56.25</td>
<td>38.80</td>
</tr>
<tr>
<td>0.053</td>
<td>22.21</td>
<td>24.70</td>
<td>56.25</td>
<td>38.80</td>
</tr>
<tr>
<td>0.054</td>
<td>22.21</td>
<td>25.20</td>
<td>56.25</td>
<td>38.80</td>
</tr>
<tr>
<td>0.055</td>
<td>22.21</td>
<td>25.70</td>
<td>56.25</td>
<td>38.80</td>
</tr>
<tr>
<td>0.056</td>
<td>22.21</td>
<td>26.20</td>
<td>56.25</td>
<td>38.79</td>
</tr>
<tr>
<td>0.057</td>
<td>22.21</td>
<td>26.70</td>
<td>56.26</td>
<td>38.79</td>
</tr>
<tr>
<td>0.058</td>
<td>22.21</td>
<td>27.20</td>
<td>56.26</td>
<td>38.79</td>
</tr>
<tr>
<td>0.059</td>
<td>22.22</td>
<td>27.70</td>
<td>56.26</td>
<td>38.79</td>
</tr>
<tr>
<td>0.060</td>
<td>22.22</td>
<td>28.20</td>
<td>56.26</td>
<td>38.79</td>
</tr>
<tr>
<td>0.061</td>
<td>22.22</td>
<td>28.70</td>
<td>56.26</td>
<td>38.79</td>
</tr>
<tr>
<td>0.062</td>
<td>22.22</td>
<td>29.19</td>
<td>56.27</td>
<td>38.79</td>
</tr>
<tr>
<td>0.063</td>
<td>22.23</td>
<td>29.69</td>
<td>56.27</td>
<td>38.79</td>
</tr>
<tr>
<td>0.064</td>
<td>22.23</td>
<td>30.19</td>
<td>56.27</td>
<td>38.79</td>
</tr>
<tr>
<td>0.065</td>
<td>22.23</td>
<td>30.69</td>
<td>56.27</td>
<td>38.79</td>
</tr>
<tr>
<td>0.066</td>
<td>22.23</td>
<td>31.19</td>
<td>56.28</td>
<td>38.79</td>
</tr>
<tr>
<td>0.067</td>
<td>22.24</td>
<td>31.69</td>
<td>56.28</td>
<td>38.79</td>
</tr>
<tr>
<td>0.068</td>
<td>22.24</td>
<td>32.18</td>
<td>56.28</td>
<td>38.79</td>
</tr>
<tr>
<td>0.069</td>
<td>22.24</td>
<td>32.68</td>
<td>56.29</td>
<td>38.78</td>
</tr>
<tr>
<td>0.070</td>
<td>22.25</td>
<td>33.18</td>
<td>56.29</td>
<td>38.78</td>
</tr>
<tr>
<td>0.071</td>
<td>22.25</td>
<td>33.68</td>
<td>56.29</td>
<td>38.78</td>
</tr>
<tr>
<td>0.072</td>
<td>22.25</td>
<td>34.18</td>
<td>56.30</td>
<td>38.78</td>
</tr>
<tr>
<td>0.073</td>
<td>22.26</td>
<td>34.68</td>
<td>56.30</td>
<td>38.78</td>
</tr>
<tr>
<td>0.074</td>
<td>22.26</td>
<td>35.18</td>
<td>56.31</td>
<td>38.78</td>
</tr>
<tr>
<td>0.075</td>
<td>22.27</td>
<td>35.67</td>
<td>56.31</td>
<td>38.78</td>
</tr>
<tr>
<td>0.076</td>
<td>22.27</td>
<td>36.17</td>
<td>56.31</td>
<td>38.78</td>
</tr>
<tr>
<td>0.077</td>
<td>22.28</td>
<td>36.67</td>
<td>56.32</td>
<td>38.78</td>
</tr>
<tr>
<td>0.078</td>
<td>22.28</td>
<td>37.17</td>
<td>56.32</td>
<td>38.78</td>
</tr>
<tr>
<td>0.079</td>
<td>22.29</td>
<td>37.67</td>
<td>56.33</td>
<td>38.78</td>
</tr>
<tr>
<td>0.080</td>
<td>22.29</td>
<td>38.17</td>
<td>56.33</td>
<td>38.78</td>
</tr>
<tr>
<td>0.081</td>
<td>22.30</td>
<td>38.67</td>
<td>56.34</td>
<td>38.78</td>
</tr>
<tr>
<td>0.082</td>
<td>22.30</td>
<td>39.17</td>
<td>56.34</td>
<td>38.78</td>
</tr>
<tr>
<td>0.083</td>
<td>22.31</td>
<td>39.67</td>
<td>56.35</td>
<td>38.78</td>
</tr>
<tr>
<td>0.084</td>
<td>22.31</td>
<td>40.16</td>
<td>56.35</td>
<td>38.78</td>
</tr>
<tr>
<td>0.085</td>
<td>22.32</td>
<td>40.66</td>
<td>56.36</td>
<td>38.78</td>
</tr>
<tr>
<td>0.086</td>
<td>22.32</td>
<td>41.16</td>
<td>56.36</td>
<td>38.77</td>
</tr>
<tr>
<td>0.087</td>
<td>22.33</td>
<td>41.66</td>
<td>56.37</td>
<td>38.77</td>
</tr>
<tr>
<td>0.088</td>
<td>22.34</td>
<td>42.16</td>
<td>56.38</td>
<td>38.77</td>
</tr>
<tr>
<td>0.089</td>
<td>22.34</td>
<td>42.66</td>
<td>56.38</td>
<td>38.77</td>
</tr>
<tr>
<td>0.090</td>
<td>22.35</td>
<td>43.16</td>
<td>56.39</td>
<td>38.77</td>
</tr>
<tr>
<td>0.091</td>
<td>22.35</td>
<td>43.65</td>
<td>56.40</td>
<td>38.77</td>
</tr>
<tr>
<td>0.092</td>
<td>22.36</td>
<td>44.15</td>
<td>56.40</td>
<td>38.77</td>
</tr>
<tr>
<td>0.093</td>
<td>22.37</td>
<td>44.65</td>
<td>56.41</td>
<td>38.77</td>
</tr>
<tr>
<td>0.094</td>
<td>22.38</td>
<td>45.15</td>
<td>56.42</td>
<td>38.77</td>
</tr>
<tr>
<td>0.095</td>
<td>22.38</td>
<td>45.65</td>
<td>56.42</td>
<td>38.77</td>
</tr>
</tbody>
</table>

Table A-2
(2 of 4)
0.096
0.097
0.098
0.099
0.300
0.301
0.102
0.103
0.104
0.105
0.106
0.107
0,108
0.109
0.110
0.111
0.112
0.133
0.314
0.135
0.116
0.117
0.118
0.119
0 ]20
0 121
0 122
0 323
O ]24
0 125
0.126
0.127
0.128
0.129
0.130
0.131
0.]32
0.333
0.334
0.135
0.136
0.337
0.138
0.339
0.140
O.laI
0.142
0 143
0 144
O 145
0 146
0 147
0 148
0.149
0.150

22.39
22.40
22.41
22.41
22.42
22.43
22.44
22.45
22.46
22.47
22.47
22.48
22.49
22.50
22,51
22.52
22.54
22.55
22.56
22.57
22.58
22.59
22.61
22.62
22.63
22.64
22.66
22.67
22.69
22.70
22.72
22,73
22.75
22.76
22.78
22.80
22.81
22.83
22,85
22.87
22.89
22.91
22.93
22.95
22.97
22.99
23.02
23.04
23.06
23.09
23.32
23.14
23.17
23.20
23.23

46.15
_i6.64
47.1co
47.64
48.14
48.64
49.14
49.64
50.13
50.63
51.13
51.63
52.13
52.63
53.13
53.62
54.12
54.62
55.12
55.62
56.12
56.61
57.11
57.61
58.11
58.61
59.11
59.61
60.10
60.60
61 .I0
61.60
62.10
62.60
63.10
63.59
64.09
64.59
65
09
65 59
66
09
66
58
67 08
67
58
68
08
68.58
69.08
69.57
70.07
70.57
71.07
71.57
72.07
72.57
73.06

56.43
56.44
56.45
56.45
56.46
56.47
56.48
56.49
56.50
56.51
56.51
56.52
56.53
56.54
56.55
56.56
56.58
56.59
56.60
56.61
56.62
56.63
56.64
56.66
56.67
56.68
56.70
56.71
56.72
56.74
56.75
56.77
56.78
56.80
56.82
56.83
56.85
56.87
56.88
56 90
56 92
56 94
56 96
56 98
57 O0
57 03
57.05
57.07
57.09
57.12
57.]6
57.17
57.20
57.22
57.25

38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.77
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38.76
38 76
38 76
38 76
38 76
38 76
38 76

0.99
0.98
0.98
0.97
0.96
0.95
0.94
0.93
0.92
0.91
0.90
0.89
0.89
0.88
0.87
0.86
0.85
0.84
0.83
0.82
0.81
0.80
0.80
0.79
0.78
0.77
0.76
0.75
0.74
0.73
0.72
O.71
0.71
0.70
0.69
0.68
0.67
0.66
0.65
0.64
0.63
0.62
0.62
0.61
0.60
0.59
0.58
0.57
0.56
0.55
0.54
0.54
0.53
0.52
0.51

Table

A-2

(3

4)

of

35

12.71
12.59
32.47
12.36
12.24
12.12
12.00
11.89
11.77
11.65
11.53
ll .42
ll .30
II .18
11.06
10.94
10.83
10.71
10 59
10 47
]0 36
10 24
30 12
I0 O0
9.89
9.77
9.65
9.53
9.42
9.30
9.18
9.06
8.94
8.83
8.71
8.59
8 47
8 36
8 24
8 12
8 O0
7 E9
7.77
7.65
7.53
7.42
7.30
7.18
7.06
6.95
6.83
6.71
6.59
6 .¢,7
6.36

47.09
47.60
48 .II
48.62
49.33
49.64
50.15
50.66
5] .17
51.67
52.18
52.69
53.20
53.71
54.22
54.73
55.24
55.75
56.25
56.76
57.27
57.78
58.29
58.80
59.31
59.82
60.33
60.84
61 .34
61.85
62.36
62.87
63.38
63.89
64 40
64 91
65 42
65 93
66 43
66 94
67 _5
67 96
68 47
68.98
69.49
70.00
70.51
71.02
71.52
72.03
72.54
73.05
73.56
7(0.07
7_i .58

10.89
11.01
11.13
13.24
11.36
11.48
31.60
]l .71
11.83
11,95
12.07
12.18
12.30
12.42
12.54
12.66
12.77
12.89
]3.01
13.13
13.24
13.36
]3.48
13.60
13.71
13.83
13.95
34.07
14.18
14.30
14.42
14.54
14.66
14.77
]4.89
15.01
15.13
15.24
15.36
15.48
15.60
35.71
15 83
15 95
16 07
16 18
26 30
16 42
16 54
16 65
36.77
16.89
37.01
17.13
17.24


| 0.151 | 23.26 | 73.56 | 57.28 | 38.76 | 0.50 | 6.24 | 75.09 | 17.36 |
| 0.152 | 23.29 | 74.06 | 57.31 | 38.76 | 0.49 | 6.12 | 75.60 | 17.48 |
| 0.153 | 23.32 | 74.56 | 57.35 | 38.76 | 0.48 | 6.00 | 76.11 | 17.60 |
| 0.154 | 23.36 | 75.06 | 57.38 | 38.76 | 0.47 | 5.89 | 76.61 | 17.71 |
| 0.155 | 23.39 | 75.56 | 57.41 | 38.76 | 0.46 | 5.77 | 77.12 | 17.83 |
| 0.156 | 23.43 | 76.05 | 57.45 | 38.76 | 0.45 | 5.65 | 77.63 | 17.95 |
| 0.157 | 23.47 | 76.55 | 57.48 | 38.77 | 0.44 | 5.53 | 78.14 | 18.07 |
| 0.158 | 23.51 | 77.05 | 57.52 | 38.77 | 0.43 | 5.42 | 78.65 | 18.18 |
| 0.159 | 23.55 | 77.55 | 57.56 | 38.77 | 0.42 | 5.30 | 79.16 | 18.30 |
| 0.160 | 23.59 | 78.05 | 57.60 | 38.77 | 0.41 | 5.18 | 79.67 | 18.42 |
| 0.161 | 23.63 | 78.55 | 57.64 | 38.77 | 0.40 | 5.06 | 80.18 | 18.54 |
| 0.162 | 23.68 | 79.04 | 57.69 | 38.77 | 0.39 | 4.95 | 80.69 | 18.65 |
| 0.163 | 23.73 | 79.54 | 57.73 | 38.77 | 0.38 | 4.83 | 81.20 | 18.77 |
| 0.164 | 23.78 | 80.04 | 57.78 | 38.77 | 0.37 | 4.71 | 81.70 | 18.89 |
| 0.165 | 23.83 | 80.54 | 57.83 | 38.77 | 0.36 | 4.59 | 82.21 | 19.01 |
| 0.166 | 23.88 | 81.04 | 57.88 | 38.77 | 0.35 | 4.48 | 82.72 | 19.12 |
| 0.167 | 23.93 | 81.54 | 57.94 | 38.77 | 0.34 | 4.36 | 83.23 | 19.24 |
| 0.168 | 24.00 | 82.03 | 58.00 | 38.77 | 0.33 | 4.24 | 83.74 | 19.36 |
| 0.169 | 24.07 | 82.53 | 58.06 | 38.77 | 0.32 | 4.12 | 84.25 | 19.48 |
| 0.170 | 24.14 | 83.03 | 58.12 | 38.77 | 0.31 | 4.00 | 84.76 | 19.60 |
| 0.171 | 24.21 | 83.53 | 58.19 | 38.77 | 0.30 | 3.89 | 85.27 | 19.71 |
| 0.172 | 24.28 | 84.03 | 58.26 | 38.77 | 0.29 | 3.77 | 85.78 | 19.83 |
| 0.173 | 24.36 | 84.53 | 58.33 | 38.77 | 0.28 | 3.65 | 86.29 | 19.95 |
| 0.174 | 24.45 | 85.03 | 58.40 | 38.77 | 0.27 | 3.53 | 86.80 | 20.07 |
| 0.175 | 24.53 | 85.53 | 58.47 | 38.77 | 0.26 | 3.42 | 87.30 | 20.18 |
| 0.176 | 24.63 | 86.03 | 58.54 | 38.77 | 0.25 | 3.30 | 87.81 | 20.30 |
| 0.177 | 24.73 | 86.53 | 58.62 | 38.77 | 0.24 | 3.18 | 88.32 | 20.42 |
| 0.178 | 24.84 | 87.03 | 58.69 | 38.77 | 0.23 | 3.06 | 88.83 | 20.54 |
| 0.179 | 24.95 | 87.53 | 58.76 | 38.77 | 0.22 | 2.95 | 89.34 | 20.65 |
| 0.180 | 25.06 | 88.03 | 58.83 | 38.77 | 0.21 | 2.83 | 89.85 | 20.77 |
| 0.181 | 25.16 | 88.53 | 58.90 | 38.77 | 0.20 | 2.71 | 90.36 | 20.89 |
| 0.182 | 25.28 | 89.03 | 58.98 | 38.78 | 0.19 | 2.59 | 90.87 | 21.01 |
| 0.183 | 25.40 | 89.53 | 59.06 | 38.78 | 0.18 | 2.48 | 91.38 | 21.12 |
| 0.184 | 25.60 | 90.00 | 59.14 | 38.78 | 0.17 | 2.36 | 91.89 | 21.24 |
| 0.185 | 25.78 | 90.50 | 59.21 | 38.78 | 0.16 | 2.24 | 92.40 | 21.36 |
| 0.186 | 26.08 | 91.00 | 59.29 | 38.78 | 0.15 | 2.12 | 92.90 | 21.49 |
| 0.187 | 26.31 | 91.50 | 59.37 | 38.79 | 0.14 | 2.01 | 93.41 | 21.59 |
| 0.188 | 26.57 | 92.00 | 59.45 | 38.79 | 0.13 | 1.89 | 93.92 | 21.71 |
| 0.189 | 26.85 | 92.50 | 59.53 | 38.79 | 0.12 | 1.77 | 94.43 | 21.83 |
| 0.190 | 27.17 | 93.00 | 59.61 | 38.79 | 0.11 | 1.66 | 94.94 | 21.94 |
| 0.191 | 27.54 | 93.50 | 59.69 | 38.80 | 0.10 | 1.54 | 95.45 | 22.06 |
| 0.192 | 27.96 | 94.00 | 59.77 | 38.80 | 0.99 | 1.42 | 95.96 | 22.18 |
| 0.193 | 28.45 | 94.50 | 59.85 | 38.80 | 0.98 | 1.30 | 96.47 | 22.30 |
| 0.194 | 29.04 | 95.00 | 60.16 | 38.81 | 0.97 | 1.19 | 96.98 | 22.43 |
| 0.195 | 29.04 | 95.46 | 62.36 | 38.81 | 0.96 | 1.19 | 97.49 | 22.53 |

Table A-2

(4 of 4)
Example of TARGET Terminal Input Session

DMSLI0740I EXECUTION BEGINS...

Enter the appropriate number to select a fluid
0.....Hydrogen
1.....Oxygen
2.....Nitrogen
0

Enter the final receiver tank pressure in psia
30.0

Enter the final receiver tank filling in %
95.0

Enter the receiver tank mass in Lbm.
150.0

Enter the receiver tank volume in cu. ft.
50.0

AFB002I STOP Normal ending - TI > 540

Table A-3
Example of TARGET Output

******************************************
****** LIQUID HYDROGEN TEST FLUID ******
******************************************

The final tank condition is $P = 30.0$ psia

with a percent filling of 95.00 %

The receiver tank mass is 150.00 Lbm.

The receiver tank volume is 50.00 cu. ft.

The receiver tank $m/V = 3.000$ Lbm./cu. ft.

The total mass injected is 200.37 Lbm.

The final fluid temperature is 41.30 Deg. R

******************************************
<table>
<thead>
<tr>
<th>T (°F)</th>
<th>H (Btu/lbm)</th>
<th>P (psid)</th>
</tr>
</thead>
<tbody>
<tr>
<td>51.00</td>
<td>-98.163</td>
<td>1.51</td>
</tr>
<tr>
<td>61.00</td>
<td>-98.227</td>
<td>1.60</td>
</tr>
<tr>
<td>71.00</td>
<td>-98.333</td>
<td>1.75</td>
</tr>
<tr>
<td>81.00</td>
<td>-98.491</td>
<td>1.98</td>
</tr>
<tr>
<td>91.00</td>
<td>-98.712</td>
<td>2.29</td>
</tr>
<tr>
<td>101.00</td>
<td>-99.004</td>
<td>2.70</td>
</tr>
<tr>
<td>111.00</td>
<td>-99.372</td>
<td>3.22</td>
</tr>
<tr>
<td>121.00</td>
<td>-99.816</td>
<td>3.83</td>
</tr>
<tr>
<td>131.00</td>
<td>-100.329</td>
<td>4.53</td>
</tr>
<tr>
<td>141.00</td>
<td>-100.909</td>
<td>5.31</td>
</tr>
<tr>
<td>151.00</td>
<td>-101.553</td>
<td>6.17</td>
</tr>
<tr>
<td>161.00</td>
<td>-102.258</td>
<td>7.09</td>
</tr>
<tr>
<td>171.00</td>
<td>-103.020</td>
<td>8.07</td>
</tr>
<tr>
<td>181.00</td>
<td>-103.838</td>
<td>9.10</td>
</tr>
<tr>
<td>191.00</td>
<td>-104.708</td>
<td>10.16</td>
</tr>
<tr>
<td>201.00</td>
<td>-105.627</td>
<td>11.25</td>
</tr>
<tr>
<td>211.00</td>
<td>-106.593</td>
<td>12.37</td>
</tr>
<tr>
<td>221.00</td>
<td>-107.602</td>
<td>13.49</td>
</tr>
<tr>
<td>231.00</td>
<td>-108.651</td>
<td>14.62</td>
</tr>
<tr>
<td>241.00</td>
<td>-109.739</td>
<td>15.75</td>
</tr>
<tr>
<td>251.00</td>
<td>-110.862</td>
<td>16.85</td>
</tr>
<tr>
<td>261.00</td>
<td>-112.018</td>
<td>17.94</td>
</tr>
<tr>
<td>271.00</td>
<td>-113.205</td>
<td>19.00</td>
</tr>
<tr>
<td>281.00</td>
<td>-114.419</td>
<td>20.02</td>
</tr>
<tr>
<td>291.00</td>
<td>-115.659</td>
<td>21.00</td>
</tr>
<tr>
<td>301.00</td>
<td>-116.923</td>
<td>21.93</td>
</tr>
<tr>
<td>311.00</td>
<td>-118.208</td>
<td>22.82</td>
</tr>
<tr>
<td>321.00</td>
<td>-119.512</td>
<td>23.65</td>
</tr>
<tr>
<td>331.00</td>
<td>-120.834</td>
<td>24.43</td>
</tr>
<tr>
<td>341.00</td>
<td>-122.172</td>
<td>25.15</td>
</tr>
<tr>
<td>351.00</td>
<td>-123.524</td>
<td>25.82</td>
</tr>
<tr>
<td>361.00</td>
<td>-124.889</td>
<td>26.42</td>
</tr>
<tr>
<td>371.00</td>
<td>-126.265</td>
<td>26.98</td>
</tr>
<tr>
<td>381.00</td>
<td>-127.651</td>
<td>27.48</td>
</tr>
<tr>
<td>391.00</td>
<td>-129.047</td>
<td>27.92</td>
</tr>
<tr>
<td>401.00</td>
<td>-130.452</td>
<td>2.70</td>
</tr>
<tr>
<td>411.00</td>
<td>-131.864</td>
<td>2.70</td>
</tr>
<tr>
<td>421.00</td>
<td>-133.283</td>
<td>2.70</td>
</tr>
<tr>
<td>431.00</td>
<td>-134.710</td>
<td>2.70</td>
</tr>
<tr>
<td>441.00</td>
<td>-136.144</td>
<td>2.70</td>
</tr>
<tr>
<td>451.00</td>
<td>-137.585</td>
<td>2.70</td>
</tr>
<tr>
<td>461.00</td>
<td>-139.033</td>
<td>2.70</td>
</tr>
<tr>
<td>471.00</td>
<td>-140.490</td>
<td>2.70</td>
</tr>
<tr>
<td>481.00</td>
<td>-141.956</td>
<td>2.70</td>
</tr>
<tr>
<td>491.00</td>
<td>-143.431</td>
<td>2.70</td>
</tr>
<tr>
<td>501.00</td>
<td>-144.918</td>
<td>2.70</td>
</tr>
<tr>
<td>511.00</td>
<td>-146.417</td>
<td>2.70</td>
</tr>
<tr>
<td>521.00</td>
<td>-147.931</td>
<td>2.70</td>
</tr>
<tr>
<td>531.00</td>
<td>-149.462</td>
<td>2.70</td>
</tr>
</tbody>
</table>

Table A-4

(2 of 2)
Example of CRYOCHIL Terminal Input Session

DMSLI0740I EXECUTION BEGINS...

Enter the appropriate number to select a fluid
0.....Hydrogen
1.....Oxygen
2.....Nitrogen
0

Enter the max. receiver tank pressure in psia
30.0

Enter the supply tank saturation pressure in psia
14.696

Enter the vent stage pressure pressure drop in psia
10.0

Enter the receiver tank mass in Lbm.
150.0

Enter the receiver tank volume in Ft**3
50.0

Enter the initial tank temperature in deg. R
540.0

Enter the TARGET temperature in deg. R
235.0

Enter the TARGET temperature error band
0.5

***** NORMAL ENDING IN CALC; NG = 2**********

AFB002I STOP Normal ending in CRYOCHIL

Table A-5

40
Example of CRYOCHIL Output

*************** LIQUID HYDROGEN TEST FLUID **************

*************** ECHO TERMINAL INPUT**************

Enter the max. receiver tank pressure in psia
30.00000

Enter the supply tank saturation pressure in psia
14.69600

Enter the vent stage pressure pressure drop in psia
10.00000

Enter the receiver tank mass in Lbm.
150.00000

Enter the receiver tank volume in Ft**3
50.00000

The tank mass-to-volume ratio is 3.00 Lbm/Ft**3
Initial temperature for cycle 1 is 540.000 R
Tank temperature before venting is 516.615 R
Mass injected in cycle 1 is 0.57329 Lbm.
Tank vented 4 times
Tank temperature after venting is 511.123 R

Initial temperature for cycle 2 is 511.123 R
Tank temperature before venting is 487.100 R
Mass injected in cycle 2 is 0.60757 Lbm.
Tank vented 4 times
Tank temperature after venting is 481.482 R

Initial temperature for cycle 3 is 481.482 R
Tank temperature before venting is 457.034 R
Mass injected in cycle 3 is 0.64766 Lbm.
Tank vented 4 times
Tank temperature after venting is 451.328 R

Initial temperature for cycle 4 is 451.328 R
Tank temperature before venting is 426.615 R
Mass injected in cycle 3 is 0.69396 Lbm.
Tank vented 4 times
Tank temperature after venting is 420.838 R

Table A-6
(2 of 4)
Initial temperature for cycle 5 is 420.838 R

Tank temperature before venting is 395.940 R

Mass injected in cycle 5 is 0.74780 Lbm.

Tank vented 4 times

Tank temperature after venting is 390.088 R

Initial temperature for cycle 6 is 390.088 R

Tank temperature before venting is 365.002 R

Mass injected in cycle 6 is 0.81123 Lbm.

Tank vented 4 times

Tank temperature after venting is 359.047 R

Initial temperature for cycle 7 is 359.047 R

Tank temperature before venting is 333.677 R

Mass injected in cycle 7 is 0.88737 Lbm.

Tank vented 4 times

Tank temperature after venting is 327.566 R

Initial temperature for cycle 8 is 327.566 R

Tank temperature before venting is 301.715 R

Mass injected in cycle 8 is 0.98125 Lbm.

Tank vented 4 times

Tank temperature after venting is 295.359 R

Table A-6
(3 of 4)
Initial temperature for cycle 9 is 295.359 R
Tank temperature before venting is 268.650 R
Mass injected in cycle 9 is 1.10154 Lbm.
Tank vented 4 times
Tank temperature after venting is 261.905 R
**********************************************
Initial temperature for cycle 10 is 261.905 R
**********************************************
TFNEW < TARGET -- BEGIN NEW CYCLE

Initial temperature for cycle 10 is 261.905 R
Tank temperature before venting is 240.059 R
Mass injected in cycle 10 is 0.95703 Lbm.
Tank vented 3 times
Final tank temperature is 235.000 +/- 0.500 R
Total mass after 10 cycle(s) is 8.00870 Lbm.

Table A-6
(4 of 4)
APPENDIX B

Input Screens for Sample Problems
The following are samples of the screens that the user would see after logging on to the VM computer and beginning execution of CRYOTRAN. The user responses for these sample runs are marked with an "*" to the right of the input line. In most cases the VM system responses are in Capital letters and user responses are in lower case.

Sample 1, Sinda model of a sphere, all 5 regions defined

Ready; T=0.01/0.01 13:07:13
link cryolib 200 222 rr
Ready; T=0.01/0.01 13:07:27
access 222 m
M (222) R/O
Ready; T=0.01/0.01 13:07:35
runcryo
C (301) R/O
D (302) R/O
No filetype specified
CONNECT= 00:10:42 VIRTCPUS 000:00.71 TOTCPU= 000:01.56
CONNECT= 00:00:01 VIRTCPUS 000:00.00 TOTCPU= 000:00.01
Assigning temporary storage destination to disk E

DASD is being cleared
DASD 303 DEFINED 0010 CYL
DASD 304 LINKED R/O; R/W BY VVUSQ; R/O BY 5 USERS
DMSACP7231 F (304) R/O
DMSLIO201W The following names are undefined:
   CYLNDR SPHERE MATMNU CYLINDS SPHND S ULLGET ULLIG AREACYL
   CYLCLS SPHCDS PRPTBL DUNPLT PLTCYL PLTSPH CHILL NVFILL
   TARGET
DMSLIO201W The following names are undefined:
   CYLNDR MATMNU CYLINDS ULLIG AREACYL CYLCLS PRPTBL DUNPLT
   PLTCYL PLTSPH CHILL NVFILL TARGET
DMSLIO201W The following names are undefined:
   DUNPLT PLTCYL PLTSPH CHILL NVFILL TARGET
DMSLIO201W The following names are undefined:
   DUNPLT PLTCYL PLTSPH CHILL NVFILL TARGET
DMSLIO7401 Execution begins...

WELCOME TO CRYOTRAN
YOU WILL BE PROMPTED FOR ALL NECESSARY INPUT.
READ THE INSTRUCTIONS CAREFULLY.
TYPE IN THE INPUT DATA CAREFULLY TO AVOID TROUBLE,
YOU MAY QUIT THE PROGRAM AT ANY INPUT PROMPT BY TYPING A "Q" (QUIT)

ENTER THE NUMBER FOR THE DESIRED PROBLEM TYPE
THE PROBLEM TYPES ARE AS FOLLOWS:

1 - THERMO/ THERMAL SINDA ANALYSIS ON A SPHERE.
2 - THERMO/ THERMAL SINDA ANALYSIS ON A CYLINDER.
3 - RUN A PRESTORED ANALYSIS PROGRAM

CHOOSE THE ANALYSIS PROGRAM YOU WISH TO USE.
TYPE IN THE NUMBER OF THE DESIRED ANALYSIS.
1  2D WEDGE WITH INSIDE OF TANK NODALIZED
2  2D WEDGE SHELL - NO NODES INSIDE OF TANK
3  2D WEDGE SHELL - THICK WALL FILL ANALYSIS

NOW A TITLE FOR THIS PROBLEM.

THE TITLE LINE MAY BE UP TO 80 CHARACTERS LONG.
TYPE IN THE TITLE.
sample model spherel *

THIS TASK IS BEING SET UP FOR THE CRAY,
NOW INPUT NECESSARY CRAY INFO.

WHICH CRAY SYSTEM COS OR UNICOS
TYPE IN C OR U
u *

TYPE IN YOUR CRAY USERID.
userid *

TYPE IN YOUR CRAY PASSWORD.
password *

TYPE IN NO. OF CRAY CPU SECONDS TO BE USED.
IF NUMBER OF SECONDS REQUESTED IS < 10, 60 WILL BE USED.
59 *

TYPE AMOUNT OF CRAY MEMORY TO BE REQUESTED,
IF AMOUNT REQUESTED IS < 1,500,000, 1,500,000 WILL BE USED.
1 *

NOW GIVE YOUR JOB A NAME, TYPE IN THE NAME,
1 - 7 ALPHABETIC CHARACTERS.
spherel *

THE CRAY JCL THAT WAS INPUT IS AS FOLLOWS:
USERID = vvgienn
PASSWORD = password
CPU TIME REQUEST = 59 SECS.
MEMORY REQUEST = 1500000 words
JOB NAME = spherel

ARE THESE ALL CORRECT? TYPE Y OR N OR Q TO QUIT
y *

NOW INPUT SPECIFIC DATA FOR THIS SPHERE.
INPUT DATA TO DEFINE THE SPHERE MAY BE ANY ONE OF:
1 RIN (IN.) AND ROUT (IN.)
2 TKN VOL. (CU.FT.) AND WALL THICKNESS (IN.)
3 TKN VOL. (CU.FT.) AND ROUT (IN.)
4 RIN (IN.) AND WALL THICKNESS (IN.)
5 ROUT (IN.) AND WALL THICKNESS (IN.)

ENTER A NUMBER 1 - 5

ENTER INSIDE TANK RADIUS, RIN(IN.).

ENTER WALL THICKNESS (IN.).

THE GEOMETRY FOR THIS ANALYSIS IS A SPHERE WITH
VOL= 19.393 FT**3, RIN= 20.000 IN., AND
WALL THICKNESS= 0.5000 IN.

TYPE IN NUMBER OF NODES ALONG CIRCUMFERENCE OF THE SPHERE.
SOUTH POLE TO NORTH POLE.
IF VALUE INPUT IS < 10, 20 WILL BE USED AS A DEFAULT.

THIS IS A 2D ANALYSIS, THE WEDGE ANGLE = 1 RAD.

INPUTTING DATA FOR REGION 1, TANKWALL

TYPE IN THE NO. OF LAYERS OF NODES THRU REGION 1

TEMPERATURES MAY BE IN DEGF OR DEGR IF NO RADIATION IS PRESENT.
THE TEMPERATURES WILL BE INPUT IN WHAT UNITS F OR R?

TYPE IN INITIAL TEMPERATURE FOR THIS REGION (DEG R)

ENTER MATERIAL NUMBER FOR REGION

101 LIQUID HYDROGEN
102 LIQUID METHANE
103 LIQUID NITROGEN
104 LIQUID OXYGEN
201 STAINLESS 304/A
202 STAINLESS 347
203 ALUMINUM 6061
204 ALUMINUM 2219
205 ALUMINUM 7075
206 ALUMINUM OXIDE
207 INCONEL X-750

48
IS THERE TO BE A REGION ON THE OUTSIDE OF THE TANKWALL?  
EG. INSULATION.  
TYPE IN Y OR N  
Y  

INPUTTING DATA FOR REGION 2, OUTSIDE LAYER 1  

NOW NEED TO SPECIFY THICKNESS OF REGION 2  
AND THE NUMBER OF LAYERS THRU THE REGION.  
TO DEFINE THE REGION THICKNESS THE INPUT MAY BE:  
1. THE REGION THICKNESS (IN.)  
OR 2. THE THICKNESS OF EACH LAYER IN THE REGION  
TYPE IN 1 OR 2  
1  

TYPE IN THICKNESS (WIDTH) OF REGION 2 (IN.)  
.25  

TYPE IN THE NO. OF LAYERS OF NODES THRU REGION 2  
1  

TYPE IN THE INITIAL TEMPERATURE FOR THIS REGION (DEG R)  
550  

ENTER MATERIAL NUMBER FOR REGION  
101 LIQUID HYDROGEN  
102 LIQUID METHANE  
103 LIQUID NITROGEN  
104 LIQUID OXYGEN  
201 STAINLESS 304A  
202 STAINLESS 347  
203 ALUMINUM 6061  
204 ALUMINUM 2219  
205 ALUMINUM 7075  
206 ALUMINUM OXIDE  
207 INCONEL X-750  
208 NICKEL  
301 GAS HYDROGEN  
302 GAS METHANE  
303 GAS NITROGEN  
304 GAS OXYGEN  
999 USER DEFINED  
204  

49
IS THERE TO BE A 2ND REGION OUTSIDE OF THE TANKWALL? 
EG. MORE OR DIFFERENT INSULATION. 
TYPE IN Y OR N 
  y  

INPUTTING DATA FOR REGION 3, OUTSIDE LAYER 2 

NOW NEED TO SPECIFY THICKNESS OF REGION 3 
AND THE NUMBER OF LAYERS THRU THE REGION. 
TO DEFINE THE REGION THICKNESS THE INPUT MAY BE: 
1. THE REGION THICKNESS (IN.) 
OR 2. THE THICKNESS OF EACH LAYER IN THE REGION 
   TYPE IN 1 OR 2 
  2  
   TYPE IN THICKNESS (WIDTH) OF EACH LAYER OF REGION 3 (IN.) 
  .1  
   TYPE IN THE NO. OF LAYERS OF NODES THRU REGION 3 
  2  
   TYPE IN THE INITIAL TEMPERATURE FOR THIS REGION (DEG R) 
  540  

ENTER MATERIAL NUMBER FOR REGION 
101 LIQUID HYDROGEN 
102 LIQUID METHANE 
103 LIQUID NITROGEN 
104 LIQUID OXYGEN 
201 STAINLESS 304A 
202 STAINLESS 347 
203 ALUMINUM 6061 
204 ALUMINUM 2219 
205 ALUMINUM 7075 
206 ALUMINUM OXIDE 
207 INCONEL X-750 
208 NICKEL 
301 GAS HYDROGEN 
302 GAS METHANE 
303 GAS NITROGEN 
304 GAS OXYGEN 
999 USER DEFINED 

FOR THIS ANALYSIS THE INSIDE OF THE TANK WILL BE NODALIZED 
HOW MANY REGIONS INSIDE OF THE TANK? 1 OR 2 
  2  

INPUTTING DATA FOR REGION 4, INSIDE TANK AT WALL 

REGION 4 IS PART OF THE DISTANCE INSIDE THE SPHERE, ALONG THE RADIUS, 
FROM THE INSIDE TANK WALL TOWARD THE CENTER OF THE SPHERE
WHERE \( r_{in} \), (THE INSIDE SPHERE RADIUS) = 20.000

NOW NEED TO SPECIFY THICKNESS OF REGION 4
AND THE NUMBER OF LAYERS THRU THE REGION.
TO DEFINE THE REGION THICKNESS THE INPUT MAY BE:
1. THE REGION THICKNESS (IN.)
OR 2. THE THICKNESS OF EACH LAYER IN THE REGION

TYPE IN 1 OR 2

* TYPE IN THICKNESS (WIDTH) OF REGION 4 (IN.)
* TYPE IN THE NO. OF LAYERS OF NODES THRU REGION 4
* TYPE IN THE INITIAL TEMPERATURE FOR THIS REGION (DEG R)

ENTER MATERIAL NUMBER FOR REGION
101 LIQUID HYDROGEN
102 LIQUID METHANE
103 LIQUID NITROGEN
104 LIQUID OXYGEN
201 STAINLESS 304A
202 STAINLESS 347
203 ALUMINUM 6061
207 INCONEL X-750
301 GAS HYDROGEN
302 GAS METHANE
303 GAS NITROGEN
304 GAS OXYGEN
999 USER DEFINED
101

INPUTTING DATA FOR REGION 5, INSIDE TANK AT CENTER

TYPE IN THE NO. OF LAYERS OF NODES THRU REGION 5
* TYPE IN THE INITIAL TEMPERATURE FOR THIS REGION (DEG R)

ENTER MATERIAL NUMBER FOR REGION
101 LIQUID HYDROGEN
102 LIQUID METHANE
103 LIQUID NITROGEN
104 LIQUID OXYGEN
201 STAINLESS 304A
202 STAINLESS 347
203 ALUMINUM 6061

51
THE HEAT TRANSFER MECHANISM INSIDE THE TANK, I.E. REGIONS 4 AND 5, IS TO BE:

1. CONDUCTION ONLY
2. CONVECTION ONLY
3. CONDUCTION AND CONVECTION

TYPE IN 1 2 OR 3

TYPE IN % TANK IS FULL OF LIQUID.

75

IS THIS ANALYSIS A LOW-G OR 1-G ANALYSIS?

TYPE IN 0 OR 1

ARE THERE TO BE ANY HEAT EXCHANGERS?

TYPE IN Y OR N

Y

HEAT EXCHANGER INFO, MAX NO. = 10

INPUT FOR HEAT EXCHANGER NO. 1

TYPE IN THE REGION NUMBER WHERE THE HEAT EXCHANGER GOES.

4

THE HEAT EXCHANGER IS ON TOP OF WHICH LAYER OF REGION 4?

TYPE IN THE LAYER NO., COUNT LAYERS FROM OUTSIDE TOWARD THE CENTER.

1

TYPE IN THE THETA ANGLE WHERE THE HEAT EXCHANGER STARTS COUNT UP FROM THE SOUTH POLE.

2

TYPE IN THE NUMBER OF THETAS THAT THE HEAT EXCHANGER COVERS.

4

TYPE IN THE HEAT EXCHANGER TEMPERATURE (DEG R)

36

HEAT EXCHANGER NO. 1 SPECIFIED ON TOP OF LAYER 1 OF REGION 4 STARTING AT THETA ANGLE 2 FOR 4 NODES, WITH TEMPERATURE: 36.00

IS THIS CORRECT?

TYPE IN Y OR N
MORE HEAT EXCHANGERS? TYPE Y OR N

HEAT EXCHANGER INFO, MAX NO. = 10
INPUT FOR HEAT EXCHANGER NO. 2
   TYPE IN THE REGION NUMBER WHERE THE HEAT EXCHANGER GOES.
4
   THE HEAT EXCHANGER IS ON TOP OF WHICH LAYER OF REGION 4?
   TYPE IN THE LAYER NO., COUNT LAYERS FROM OUTSIDE TOWARD THE CENTER.
3
   TYPE IN THE THETA ANGLE WHERE THE HEAT EXCHANGER STARTS
   COUNT UP FROM THE SOUTH POLE.
17
   TYPE IN THE NUMBER OF THETAS THAT THE HEAT EXCHANGER COVERS.
3
   TYPE IN THE HEAT EXCHANGER TEMPERATURE (DEG R)
36
HEAT EXCHANGER NO. 2 SPECIFIED
ON TOP OF LAYER 3 OF REGION 4 STARTING AT THETA ANGLE 17 FOR 3 NODES, WITH TEMPERATURE 36.00
   IS THIS CORRECT?
   TYPE IN Y OR N
Y
   MORE HEAT EXCHANGERS? TYPE Y OR N
Y

HEAT EXCHANGER INFO, MAX NO. = 10
INPUT FOR HEAT EXCHANGER NO. 3
   TYPE IN THE REGION NUMBER WHERE THE HEAT EXCHANGER GOES.
3
   THE HEAT EXCHANGER IS ON TOP OF WHICH LAYER OF REGION 3?
   TYPE IN THE LAYER NO., COUNT LAYERS FROM OUTSIDE TOWARD THE CENTER.
1
   TYPE IN THE THETA ANGLE WHERE THE HEAT EXCHANGER STARTS
   COUNT UP FROM THE SOUTH POLE.
10
   TYPE IN THE NUMBER OF THETAS THAT THE HEAT EXCHANGER COVERS.
3
   TYPE IN THE HEAT EXCHANGER TEMPERATURE (DEG R)
60
HEAT EXCHANGER NO. 3 SPECIFIED
ON TOP OF LAYER 1 OF REGION 3 STARTING AT THETA ANGLE 10 FOR 3 NODES, WITH TEMPERATURE 60.00
   IS THIS CORRECT?
   TYPE IN Y OR N
Y
   MORE HEAT EXCHANGERS? TYPE Y OR N
n
53
THERE MAY BE UP TO TWO BOUNDARY NODES ON THE OUTSIDE OF THE TANKWALL.
E.G. OUTSIDE ATMOSPHERE.
DO YOU WANT ONE OR MORE OF THESE BOUNDARY NODES?
TYPE IN Y OR N

Y

TYPE IN THE OUTSIDE ATMOSPHERE TEMPERATURE (DEG R)
600

THE HEAT TRANSFER TO THIS OUTSIDE TEMPERATURE IS TO BE
CONVECTION OR RADIATION?
TYPE IN C OR R
C

TYPE IN THE CONVECTION COEFFICIENT, (BTU/HR-FT2-DEG)
200

DO YOU WANT A 2ND OUTSIDE BOUNDARY NODE?
TYPE IN Y OR N

Y

TYPE IN THE OUTSIDE ATMOSPHERE TEMPERATURE (DEG R)
900

THE HEAT TRANSFER TO THIS OUTSIDE TEMPERATURE IS TO BE
CONVECTION OR RADIATION?
TYPE IN C OR R
R

TYPE IN THE RADIATION FACTOR (EPS*F)
.2

ONE OR MORE OF THE OUTSIDE BOUNDARY CONDUCTORS IS A RADIATION CONDUCTOR,
ALL TEMPERATURES HAVE BEEN CONVERTED TO DEG F.

IS THERE TO BE A CONSTANT Q INPUT, (SOURCE TERM)
INTO THE OUTSIDE SURFACE OF THE MODEL?
TYPE Y OR N

Y

THE VALUE OF Q MAY BE SPECIFIED IN 3 WAYS:
1 CONSTANT Q PER UNIT AREA, (BTU/(HR-FT2)
2 CONSTANT Q PER UNIT AREA, (BTU/(HR-IN2)
3 Q BASED ON BTU/HR OVER THE ENTIRE SPHERE SURFACE
TYPE 1, 2, OR 3

3

TYPE IN THE VALUE OF Q IN BTU/HR ON SPHERE
12.5

NOW INPUT THE SPECIFIC DATA FOR SINDA
THIS SINDA ANALYSIS MAY BE:
1 A STEADY STATE ANALYSIS
2 A TRANSIENT ANALYSIS
3 STEADY STATE FOLLOWED BY A TRANSIENT
4 A TRANSIENT FOLLOWED BY STEADY STATE
TYPE IN 1, 2, 3, OR 4

2

A TRANSIENT ANALYSIS IS TO BE DONE,
THE EXECUTION SUBROUTINE WILL BE EITHER FWDBCK OR CNPRDL
THIS WILL BE DETERMINED BY THE VALUE OF THE TIME STEP, (DELTIME),
WHICH WILL BE INPUT BELOW.
THE NEXT 4 INPUT VALUES INVOLVE PROBLEM TIME, THESE 4 VALUES MAY BE INPUT IN UNITS OF SECONDS, MINUTES, OR HOURS

NOW TYPE IN S M OR H

NOW TYPE IN THE PROBLEM START TIME (MIN) 0

NOW TYPE IN THE PROBLEM END TIME (MIN) 120

TYPE IN THE TIME STEP, (DELTIME), (MIN) TO BE USED. IF DELTIME IS UNKNOWN, OR IF YOU TYPE ZERO (0), THE SINDA FORWARD DIFFERENCE METHOD, (CNFRDL), WILL BE USED AND DELTIME WILL BE COMPUTED BY THE PROGRAM .001

TYPE IN THE OUTPUT INTERVAL DTOUT (MIN) TEMPERATURES WILL BE PRINTED EVERY DT MIN. IF INPUT VALUE .LE. 0, .25 HRS. WILL BE USED .25

TYPE IN THE CONVERGENCE CRITERIA, DELTA TEMPERATURE SUGGESTED VALUE RANGE .01 TO .001 IF INPUT VALUE .LE. 0 .005 WILL BE USED.

TYPE IN NLOOP, THE NUMBER OF ITERATION LOOPS ALLOWED SUGGESTED RANGE OF VALUES 100 TO 1000 IF INPUT VALUE IS .LE. 0 100 WILL BE USED.

NOTE: SOME STEADY STATE CASES MAY NEED NLOOP > 1000

500

THE FOLLOWING IS THE RANGE OF PRESSURES IN THE MATERIAL BASE FOR HYDROGEN IN REGION #4:

STARTING PRESSURE = 5.00
ENDING PRESSURE = 81.00
INCREMENT = 2.00

ENTER THE DESIRED PRESSURE FOR THAT REGION 49

IN THE PLOTTING ROUTINE, NTYP=1; 2D SPHERE WEDGE DO YOU WANT A PLOT OF THIS GEOMETRY? TYPE Y OR N

y
IN THE SPHERE PLOTTING ROUTINE
SEND THE GRAPH TO
1. THE QMS PRINTER
2. THE TERMINAL SCREEN
3. SOME OTHER DEVICE
TYPE IN 1 2 OR 3

PLOT -- RADMAX, RSTEP = 23.9499969 5.32222080
END OF CRYOTRAN PREPROCESSOR PROGRAM,
ON TO ANALYSIS PROGRAM
THE OUTPUT FILE IS CALLED "CRYOTRAN MODEL".
THIS "CRYOTRAN MODEL" FILE IS A SINDA MODEL.

USER MAY NOW SUBMIT THE FILE "CRYOTRAN MODEL"
TO THE CRAY COMPUTER FOR EXECUTION,
OR MAKE ANY DESIRED MODIFICATIONS WITH AN EDITOR
PRIOR TO SUBMITTING IT TO THE CRAY.

TO SUBMIT THE FILE TO CRAY,
ON THE VM SYSTEM TYPE: CRSUBMIT CRYOTRAN MODEL

DO YOU WANT TO GO TO BEGINNING OF CRYOTRAN OR QUIT?
TYPE Y TO GO BACK TO BEGINNING OF CRYOTRAN,
OR TYPE N TO QUIT CRYOTRAN.

ON TO ANALYSIS PROGRAM
THE OUTPUT FILE IS CALLED "CRYOTRAN MODEL".
THIS "CRYOTRAN MODEL" FILE IS A SINDA MODEL.

USER MAY NOW SUBMIT THE FILE "CRYOTRAN MODEL"
TO THE CRAY COMPUTER FOR EXECUTION,
OR MAKE ANY DESIRED MODIFICATIONS WITH AN EDITOR
PRIOR TO SUBMITTING IT TO THE CRAY.

TO SUBMIT THE FILE TO CRAY,
ON THE VM SYSTEM TYPE: CRSUBMIT CRYOTRAN MODEL

IF USER HAS REQUESTED A GEOMETRY PLOT OF THE SINDA MODEL
THE PLOT DATA IS IN FILE NAMED "QMS PLOTDATA"

USER MAY PLOT THESE RESULTS BY TYPING: PLOTQA
END OF DISPLA 11.0 -- 27876 VECTORS IN 1 PLOTS.
RUN ON 12/8/89 USING SERIAL NUMBER 2312 AT NASA LEWIS RESEARCH CENTER
PROPRIETARY SOFTWARE PRODUCT OF COMPUTER ASSOCIATES, INC.
2729 VIRTUAL STORAGE REFERENCES; 17 READS; 4 WRITES.
AFB2401 VABEX : ABEND OCCURRED IN FORTRAN PROCESSING OF ORIGINAL ABEND.
DMSFRE161T Invalid DMSFRET call from P3570C, error number 6
CMS FILEL
DMSABN150W 75 (HEX 0000F) doublewords of system storage were not recovered
Ready; T=,*/*/.* 07:28:43

56
Which QMS printer would you like to have your output sent to?

1) ANALEX
2) RAC
3) ERB
4) DEB

Enter the number of your choice:

1

PRT FILE 8336 TO RSCS COPY 001 NOHOLD

Ready; T=0.13/0.56 07:28:34

This ends the CryoTran input prompts and the responses to a sample case 1.

Sample 2 and 3

Sinda model of sphere with subroutines called, followed by sphere with inside of tank not nodalized. As in the previous sample screens the lines containing user responses are marked with a *.

Ready; T=0.01/0.01 11:16:52
link cryolib 200 222 rr
DASD 222 LINKED R/O; R/W BY CRYOLIB
Ready; T=0.01/0.01 11:17:09
access 222 m
M (222) R/O
Ready; T=0.01/0.01 11:17:20
runcry
No filetype specified
C (301) R/O
The following names are undefined:
CYLNDR SPHERE MATMNU CYLNDR SPHND S PULLGET ULLIG AREACYL
CYLCS SPHCDS PRPTBL DUNPLT PLTCYL PLTPH CHILL NVFILL
TARGET
The following names are undefined:
CYLNDR MATMNU CYLNDR ULLIG AREACYL CYLCS PRPTBL DUNPLT
PLTCYL PLTPH CHILL NVFILL TARGET
The following names are undefined:
DUNPLT PLTCYL PLTPH CHILL NVFILL TARGET
The following names are undefined:
DUNPLT PLTCYL PLTPH
The following names are undefined:
ANGLE AREA2D BASAF CURVE DONEPL ENDPL HEIGHT IBM52
INTNO MARKER MESSAG PAGE PDEV POLAR QMS2 REALNO
RESET RLINT RLMESS RLVEIC THKCRV TRIFLX VECTOR XPOSN
YPOSN GRAF QMS
Execution begins...

WELCOME TO CRYOTRAN
YOU WILL BE PROMPTED FOR ALL NECESSARY INPUT.
READ THE INSTRUCTIONS CAREFULLY.
TYPE IN THE INPUT DATA CAREFULLY TO AVOID TROUBLE,
YOU MAY QUIT THE PROGRAM AT ANY INPUT PRACTICE BY TYPING A "Q" (QUIT)

ENTER THE NUMBER FOR THE DESIRED PROBLEM TYPE
THE PROBLEM TYPES ARE AS FOLLOWS:

1 - THERMO/THERMAL SINDA ANALYSIS ON A SPHERE.
2 - THERMO/THERMAL SINDA ANALYSIS ON A CYLINDER.
3 - RUN A PRESTORED ANALYSIS PROGRAM

CHOOSE THE ANALYSIS PROGRAM YOU WISH TO USE.
TYPE IN THE NUMBER OF THE DESIRED ANALYSIS.
1 2D WEDGE WITH INSIDE OF TANK NODALIZED
2 2D WEDGE SHELL - NO NODES INSIDE OF TANK
3 2D WEDGE SHELL - THICK WALL FILL ANALYSIS

NOW A TITLE FOR THIS PROBLEM.

THE TITLE LINE MAY BE UP TO 80 CHARACTERS LONG.
TYPE IN THE TITLE.
sample run of no nodes in tank, calling subroutines

THIS TASK IS BEING SET UP FOR THE CRAY,
NOW INPUT NECESSARY CRAY INFO.

WHICH CRAY SYSTEM COS OR UNICOS
TYPE IN C OR U
u

TYPE IN YOUR CRAY USERID.
userid

TYPE IN YOUR CRAY PASSWORD.
password

TYPE IN NO. OF CRAY CPU SECONDS TO BE USED.
IF NUMBER OF SECONDS REQUESTED IS < 10, 60 WILL BE USED.
59

TYPE AMOUNT OF CRAY MEMORY TO BE REQUESTED,
IF AMOUNT REQUESTED IS < 1,500,000, 1,500,000 WILL BE USED.
1

NOW GIVE YOUR JOB A NAME. TYPE IN THE NAME,
1 - 1 ALPHABETIC CHARACTERS.
sphere2

THE CRAY JCL THAT WAS INPUT IS AS FOLLOWS:
USERID = userid
PASSWORD = password
CPU TIME REQUEST = 59 SECS.
MEMORY REQUEST = 1500000 words
JOB NAME = sphere2

ARE THESE ALL CORRECT? TYPE Y OR N OR Q TO QUIT

Y

NOW INPUT SPECIFIC DATA FOR THIS SPHERE.
 INPUT DATA TO DEFINE THE SPHERE MAY BE ANY ONE OF:
1 RIN (IN.) AND ROUT (IN.)
2 TANK VOL. (CU.FT.) AND WALL THICKNESS (IN.)
3 TANK VOL. (CU.FT.) AND ROUT (IN.)
4 RIN (IN.) AND WALL THICKNESS (IN.)
5 ROUT (IN.) AND WALL THICKNESS (IN.)

ENTER A NUMBER 1 - 5

4

ENTER INSIDE TANK RADIUS, RIN(IN.).
24

ENTER WALL THICKNESS (IN.).
2

THE GEOMETRY FOR THIS ANALYSIS IS A SPHERE WITH
VOL = 33.51 FT**3, RIN = 24.00 IN., AND WALL THICKNESS = 2.00 IN.

TYPE IN NUMBER OF NODES ALONG CIRCUMFERENCE OF THE SPHERE,
SOUTH POLE TO NORTH POLE.
IF VALUE INPUT IS < 10, 20 WILL BE USED AS A DEFAULT.
25

THIS IS A 2D ANALYSIS, THE WEDGE ANGLE = 1 RAD.

INPUTTING DATA FOR REGION 1, TANKWALL

TYPE IN THE NO. OF LAYERS OF NODES THRU REGION 1

4

TEMPERATURES MAY BE IN DEGF OR DEGR IF NO RADIATION IS PRESENT.
THE TEMPERATURES WILL BE INPUT IN WHAT UNITS F OR R?
TYPE IN F OR R

TYPE IN THE INITIAL TEMPERATURE FOR THIS REGION (DEG R)

540

ENTER MATERIAL NUMBER FOR REGION
IS THERE TO BE A REGION ON THE OUTSIDE OF THE TANKWALL?
EG. INSULATION.
TYPE IN Y OR N

ARE THERE TO BE ANY HEAT EXCHANGERS?
TYPE IN Y OR N

THERE MAY BE UP TO TWO BOUNDARY NODES ON THE OUTSIDE OF THE TANKWALL.
EG. OUTSIDE ATMOSPHERE.
DO YOU WANT ONE OR MORE OF THESE BOUNDARY NODES?
TYPE IN Y OR N

IS THERE TO BE A CONSTANT Q INPUT, (SOURCE TERM)
INTO THE OUTSIDE SURFACE OF THE MODEL?
TYPE Y OR N

NOW INPUT THE SPECIFIC DATA FOR SINDA
THIS SINDA ANALYSIS MAY BE:
1  A STEADY STATE ANALYSIS
2  A TRANSIENT ANALYSIS
3  STEADY STATE FOLLOWED BY A TRANSIENT
4  A TRANSIENT FOLLOWED BY STEADY STATE
TYPE IN 1, 2, 3, OR 4

A TRANSIENT ANALYSIS IS TO BE DONE,
THE EXECUTION SUBROUTINE WILL BE EITHER FWDRECK OR CFNROL
THIS WILL BE DETERMINED BY THE VALUE OF THE TIME STEP, (DELTIME),
WHICH WILL BE INPUT BELOW.

THE NEXT 4 INPUT VALUES INVOLVE PROBLEM TIME,
THese 4 VALUES MAY BE INPUT IN UNITS OF
SECONDS, MINUTES, OR HOURS
NOW TYPE IN S M OR H

m

NOW TYPE IN THE PROBLEM START TIME (MIN)
0

NOW TYPE IN THE PROBLEM END TIME (MIN)
360

TYPE IN THE TIME STEP, (DELTIME), (MIN) TO BE USED.
IF DELTIME IS UNKNOWN, OR IF YOU TYPE ZERO (0),
THE SINDA FORWARD DIFFERENCE METHOD, (CNFRDL),
WILL BE USED AND DELTIME WILL BE COMPUTED BY THE PROGRAM
.0125

TYPE IN THE OUTPUT INTERVAL DTOUT (MIN) TEMPERATURES WILL BE PRINTED EVERY DT MIN.
IF INPUT VALUE .LE. 0, >>> .25 HRS. WILL BE USED
.25

TYPE IN THE CONVERGENCE CRITERIA, DELTA TEMPERATURE
SUGGESTED VALUE RANGE .01 TO .001
IF INPUT VALUE .LE. 0 >>> .005 WILL BE USED.
.001

TYPE IN NLOOP, THE NUMBER OF ITERATION LOOPS ALLOWED
SUGGESTED RANGE OF VALUES 100 TO 1000
IF INPUT VALUE IS .LE. 0 >>> 100 WILL BE USED.
NOTE: SOME STEADY STATE CASES MAY NEED NLOOP > 1000
300

SPECIAL INPUT FOR TANKFILL PROCEDURES

TYPE IN FLUID FLOW RATE (LB/HR)
.5

TYPE IN FLUID TEMPERATURE (DEG R)
NOTE: TEMPERATURE UNITS MUST BE DEG R.
40

TYPE IN VAPOR TEMPERATURE (DEG R)
60

FILL THE TANK HOW FULL? TYPE IN PERCENT TO FILL
95

DO YOU NEED MATERIAL PROPERTIES FOR THE LIQUID?
TYPE Y OR N
y

ENTER MATERIAL NUMBER FOR REGION
101 LIQUID HYDROGEN
102 LIQUID METHANE
103 LIQUID NITROGEN
104 LIQUID OXYGEN
201 STAINLESS 304A
202 STAINLESS 347
203 ALUMINUM 6061
204 ALUMINUM 2219
205 ALUMINUM 7075
206 ALUMINUM OXIDE
207 INCONEL X-750
208 NICKEL
301 GAS HYDROGEN
THE FOLLOWING IS THE RANGE OF PRESSURES IN THE MATERIAL DATABASE FOR HYDROGEN IN REGION #6:

STARTING PRESSURE = 5.00
ENDING PRESSURE = 81.00
INCREMENT = 2.00

ENTER THE DESIRED PRESSURE FOR THAT REGION

49
JCL COMMAND - IRC=FILEDEF CRYSUBS DISK CRYOSUBS THWSEI M
JCL COMMAND - IRC=FILEDEF CRYSUBS CLEAR 0

IN THE PLOTTING ROUTINE, NTYP=1; 2D SPHERE WEDGE
DO YOU WANT A PLOT OF THIS GEOMETRY?
TYPE Y OR N

n
END OF CRYOTRAN PREPROCESSOR PROGRAM,
ON TO ANALYSIS PROGRAM
THE OUTPUT FILE IS CALLED "CRYOTRAN MODEL".
THIS "CRYOTRAN MODEL" FILE IS A SINDA MODEL.

USER MAY NOW SUBMIT THE FILE "CRYOTRAN MODEL"
TO THE CRAY COMPUTER FOR EXECUTION,
OR MAKE ANY DESIRED MODIFICATIONS WITH AN EDITOR
PRIOR TO SUBMITTING IT TO THE CRAY.

TO SUBMIT THE FILE TO CRAY,
ON THE VM SYSTEM TYPE: CRSUBMIT CRYOTRAN MODEL

DO YOU WANT TO GO TO BEGINNING OF CRYOTRAN OR QUIT?
TYPE Y TO GO BACK TO BEGINNING OF CRYOTRAN,
OR TYPE N TO QUIT CRYOTRAN.

y
BEFORE CONTINUING YOU MAY WANT TO CHANGE THE NAME
OF SOME OF THE OUTPUT FILES. IF YOU DO NOT CHANGE THE NAME
OF THE MODEL FILE, THE NEW MODEL OUTPUT OF THE NEW RUN
WILL OVERWRITE THE MODEL OUTPUT OF THE PREVIOUS RUN.

DO YOU WANT TO CHANGE THE NAME OF ANY OF YOUR OUTPUT FILES FROM THIS RUN BEFORE CONTINUING?

TYPE IN Y OR N

y
CHANGE THE NAME OF THE FILE "CRYOTRAN INPUTEO"?
TYPE IN Y OR N

CHANGE THE NAME OF THE FILE "CRYOTRAN MODEL"?

TYPE IN Y OR N

Y

TYPE IN THE NEW FILE NAME; FILE TYPE; FILE MODE
YOU MUST TYPE IN ALL THREE PARTS OF NAME FN FT FM
sphere2 thkw a

JCL COMMAND - IRC=RENAMe CRYOTRAN MODEL A SPHERE2 THKW A

WELCOME TO CRYOTRAN
YOU WILL BE PROMPTED FOR ALL NECESSARY INPUT.
READ THE INSTRUCTIONS CAREFULLY.
TYPE IN THE INPUT DATA CAREFULLY TO AVOID TROUBLE,
YOU MAY QUIT THE PROGRAM AT ANY INPUT PROMPT BY TYPING A "Q" (QUIT)

ENTER THE NUMBER FOR THE DESIRED PROBLEM TYPE
THE PROBLEM TYPES ARE AS FOLLOWS:
1 - THERMO/THERMAL SINDA ANALYSIS ON A SPHERE.
2 - THERMO/THERMAL SINDA ANALYSIS ON A CYLINDER.
3 - RUN A PRESTORED ANALYSIS PROGRAM
1

CHOOSE THE ANALYSIS PROGRAM YOU WISH TO USE.
TYPE IN THE NUMBER OF THE DESIRED ANALYSIS.
1 2D WEDGE WITH INSIDE OF TANK NODALIZED
2 2D WEDGE SHELL - NO NODES INSIDE OF TANK
3 2D WEDGE SHELL - THICK WALL FILL ANALYSIS
2

NOW A TITLE FOR THIS PROBLEM.
THE TITLE LINE MAY HAVE UP TO 80 CHARACTERS LONG.
TYPE IN THE TITLE.
sample of sphere not nodalized in tank

THIS TASK IS BEING SET UP FOR THE CRAY,
NOW INPUT NECESSARY CRAY INFO.

WHICH CRAY SYSTEM COS OR UNICOS
TYPE IN C OR U
U

TYPE IN YOUR CRAY USERID.
userid

63
TYPE IN YOUR CRAY PASSWORD.

*password*

TYPE IN NO. OF CRAY CPU SECONDS TO BE USED.
IF NUMBER OF SECONDS REQUESTED IS < 10, 60 WILL BE USED.

*59*

TYPE AMOUNT OF CRAY MEMORY TO BE REQUESTED,
IF AMOUNT REQUESTED IS < 1,500,000, 1,500,000 WILL BE USED.

*1*

NOW GIVE YOUR JOB A NAME, TYPE IN THE NAME,
1 - 7 ALPHABETIC CHARACTERS.

*sphere3*

THE CRAY JCL THAT WAS INPUT IS AS FOLLOWS:
USERID = userid
PASSWORD = password
CPU TIME REQUEST = 59 SECS.
MEMORY REQUEST = 1500000 words
JOB NAME = sphere3

ARE THESE ALL CORRECT? TYPE Y OR N OR Q TO QUIT

*Y*

NOW INPUT SPECIFIC DATA FOR THIS SPHERE.
INPUT DATA TO DEFINE THE SPHERE MAY BE ANY ONE OF:

1 RIN (IN.) AND ROUT (IN.)
2 TNK VOL. (CU.FT.) AND WALL THICKNESS (IN.)
3 TNK VOL. (CU.FT.) AND ROUT (IN.)
4 RIN (IN.) AND WALL THICKNESS (IN.)
5 ROUT (IN.) AND WALL THICKNESS (IN.)

ENTER A NUMBER 1 - 5

*2*

ENTER TANK VOLUME (CU.FT.).

*200*

ENTER WALL THICKNESS (IN.).

*.2*

THE GEOMETRY FOR THIS ANALYSIS IS A SPHERE WITH
VOL= 200.000 FT**3, RIN= 43.534 IN., AND WALL THICKNESS= 0.2000 IN.

TYPE IN NUMBER OF NODES ALONG CIRCUMFERENCE OF THE SPHERE.
SOUTH POL. TO NORTH POL.
IF VALUE INPUT IS < 10, 20 WILL BE USED AS A DEFAULT.

*40*
THIS IS A 2D ANALYSIS, THE WEDGE ANGLE = 1 RAD.

INPUTTING DATA FOR REGION 1, TANKWALL

TYPE IN THE NO. OF LAYERS OF NODES THRU REGION 1

2

TEMPERATURES MAY BE IN DEGF OR DEGR IF NO RADIATION IS PRESENT.
THE TEMPERATURES WILL BE INPUT IN WHAT UNITS F OR R?
TYPE IN F OR R

TYPE IN THE INITIAL TEMPERATURE FOR THIS REGION (DEG R)
540

ENTER MATERIAL NUMBER FOR REGION
101 LIQUID HYDROGEN
102 LIQUID METHANE
103 LIQUID NITROGEN
104 LIQUID OXYGEN
201 STAINLESS 304A
202 STAINLESS 347
203 ALUMINUM 6061
204 ALUMINUM 2219
205 ALUMINUM 7075
206 ALUMINUM OXIDE
207 INCONEL X-750
208 NICKEL
301 GAS HYDROGEN
302 GAS METHANE
303 GAS NITROGEN
304 GAS OXYGEN
999 USER DEFINED

IS THERE TO BE A REGION ON THE OUTSIDE OF THE TANKWALL?
EG. INSULATION.
TYPE IN Y OR N

ARE THERE TO BE ANY HEAT EXCHANGERS?
TYPE IN Y OR N

THERE MAY BE UP TO TWO BOUNDARY NODES ON THE OUTSIDE OF THE TANKWALL.
EG. OUTSIDE ATMOSPHERE.
DO YOU WANT ONE OR MORE OF THESE BOUNDARY NODES?
TYPE IN Y OR N
FOR THIS MODEL, REGION 4 (INSIDE OF TANK), IS NOT NODALIZED WITH SINDA NODES;
DO YOU WANT CONSTANT TEMPERATURE BOUNDARY NODES TO CONNECT TO INSIDE OF TANK WALL, OR NOT?
YOU MAY HAVE:
1. NO CONSTANT TEMPERATURE BOUNDARY NODES.
2. A SINGLE SET OF CONSTANT TEMPERATURE NODES.
3. 2 SETS OF CONSTANT TEMPERATURE NODES TO SIMULATE LIQUID AND VAPOR IN 1-G.

TYPE IN 1 2 OR 3

* TYPE IN THE TEMPERATURE OF THE LIQUID BNDY NODES DEG(R) 36

* TYPE IN THE TEMPERATURE OF THE VAPOR BNDY NODES DEG(R) 45

* TYPE IN % TANK IS FULL OF LIQUID.
100

IS THERE TO BE A CONSTANT Q INPUT, (SOURCE TERM) INTO THE OUTSIDE SURFACE OF THE MODEL?
TYPE Y OR N

NOW INPUT THE SPECIFIC DATA FOR SINDA
THIS SINDA ANALYSIS MAY BE:
1 A STEADY STATE ANALYSIS
2 A TRANSIENT ANALYSIS
3 STEADY STATE FOLLOWED BY A TRANSIENT
4 A TRANSIENT FOLLOWED BY STEADY STATE
TYPE IN 1, 2, 3, OR 4
1

* TYPE IN THE CONVERGENCE CRITERIA, DELTA TEMPERATURE.
SUGGESTED VALUE RANGE .01 TO .001
IF INPUT VALUE .LE. 0 >>>.005 WILL BE USED.
.001

* TYPE IN NLOOP, THE NUMBER OF ITERATION LOOPS ALLOWED
SUGGESTED RANGE OF VALUES 100 TO 1000
IF INPUT VALUE IS .LE. 100 >> 100 WILL BE USED.
NOTE: SOME STEADY STATE CASES MAY NEED NLOOP > 1000
2000

TEMPERATURES INSIDE OF TANK ARE DEFINED TL = 36.00 DEG R AND TV = 45.00 DEG R
WANT TO INPUT HL AND HV TO COMPUTE CONVECTION COEFFICIENTS G-H*A?

TYPE IN Y OR N
Y

* TYPE IN FILM COEFFICIENT HL (BTU/HR-FT2-R)
TYPE IN FILM COEFFICIENT HV (BTU/HR-FT^2-R)

IN THE PLOTTING ROUTINE, NTYPE=1; 2D SPHERE WEDGE
DO YOU WANT A PLOT OF THIS GEOMETRY?
TYPE Y OR N

END OF CRYOTRAN PREPROCESSOR PROGRAM,
ON TO ANALYSIS PROGRAM
THE OUTPUT FILE IS CALLED "CRYOTRAN MODEL".
THIS "CRYOTRAN MODEL" FILE IS A SINDA MODEL.

USER MAY NOW SUBMIT THE FILE "CRYOTRAN MODEL"
TO THE CRAY COMPUTER FOR EXECUTION,
OR MAKE ANY DESIRED MODIFICATIONS WITH AN EDITOR
PRIOR TO SUBMITTING IT TO THE CRAY.

TO SUBMIT THE FILE TO CRAY,
ON THE VM SYSTEM TYPE: CRSUBMIT CRYOTRAN

DO YOU WANT TO GO TO BEGINNING OF CRYOTRAN OR QUIT?
TYPE Y TO GO BACK TO BEGINNING OF CRYOTRAN,
OR TYPE N TO QUIT CRYOTRAN.

ON TO ANALYSIS PROGRAM
THE OUTPUT FILE IS CALLED "CRYOTRAN MODEL".
THIS "CRYOTRAN MODEL" FILE IS A SINDA MODEL.

USER MAY NOW SUBMIT THE FILE "CRYOTRAN MODEL"
TO THE CRAY COMPUTER FOR EXECUTION,
OR MAKE ANY DESIRED MODIFICATIONS WITH AN EDITOR
PRIOR TO SUBMITTING IT TO THE CRAY.

TO SUBMIT THE FILE TO CRAY,
ON THE VM SYSTEM TYPE: CRSUBMIT CRYOTRAN

IF USER HAS REQUESTED A GEOMETRY PLOT OF THE SINDA MODEL
THE PLOT DATA IS IN FILE NAMED "QMS PLOTDATA"

USER MAY PLOT THESE RESULTS BY TYPING: PLOTQA
Ready: T=2.77/4.95 11:28:47

This ends the CryoTran input prompts and the responses to a sample cases 2 and 3.
APPENDIX C

Sample Problems
Sample sphere model with 5 regions, regions 4 and 5 are nodalized, Q on outside surface, 3 heat exchangers and 2 outside boundary nodes. Figure 7 is a plot of this sample model.

BCD 3THERMAL LPCS
C REM THIS SINDA MODEL WAS GENERATED BY CRYOTRAN
C REM SPHERE --- 2D WEDGE WITH INSIDE OF TANK NODALIZED
C REM WEDGE ANGLE-BETA = 1.0 RADIANS
BCD 9
END
BCD 3MODE DATA
REM NODE TEMPERATURES ARE IN (DEG F)
REM DIMENSIONS ARE IN (IN.), TIME IS IN (SECS)
REM SURFACE NODES, INSIDE TANK WALL
1001, 90.0, -1.000000 $ SURFACE NODES
REM HEAT EXCHANGER NO. 1, REPLACES NODES 1002 THRU 1005
GEN 1006, 2, 1, 90.0, -1.000000 $ SURFACE NODES
REM DIFFUSION NODES, REGION 1, TANKWALL
REM REGION 1, LAYER NO. 1
SIM 2001, 2, 24, 90.0, A1204, 0.247099 $ ALUMINUM 2219
SIM 2002, 2, 22, 90.0, A1204, 1.633664 $ ALUMINUM 2219
SIM 2003, 2, 20, 90.0, A1204, 2.692478 $ ALUMINUM 2219
SIM 2004, 2, 18, 90.0, A1204, 3.709835 $ ALUMINUM 2219
SIM 2005, 2, 16, 90.0, A1204, 4.668679 $ ALUMINUM 2219
SIM 2006, 2, 14, 90.0, A1204, 5.553900 $ ALUMINUM 2219
SIM 2007, 2, 12, 90.0, A1204, 6.351530 $ ALUMINUM 2219
SIM 2008, 2, 10, 90.0, A1204, 7.048999 $ ALUMINUM 2219
SIM 2009, 2, 8, 90.0, A1204, 7.635290 $ ALUMINUM 2219
SIM 2010, 2, 6, 90.0, A1204, 8.101176 $ ALUMINUM 2219
SIM 2011, 2, 4, 90.0, A1204, 8.439301 $ ALUMINUM 2219
SIM 2012, 2, 2, 90.0, A1204, 8.644333 $ ALUMINUM 2219
SIM 2013, 90.0, A1204, 8.713038 $ ALUMINUM 2219
REM REGION 1, LAYER NO. 2
SIM 2026, 2, 24, 90.0, A1204, 0.538204 $ ALUMINUM 2219
SIM 2027, 2, 22, 90.0, A1204, 1.606119 $ ALUMINUM 2219
SIM 2028, 2, 20, 90.0, A1204, 2.664705 $ ALUMINUM 2219
SIM 2029, 2, 18, 90.0, A1204, 3.649517 $ ALUMINUM 2219
SIM 2030, 2, 16, 90.0, A1204, 4.592772 $ ALUMINUM 2219
SIM 2031, 2, 14, 90.0, A1204, 5.463598 $ ALUMINUM 2219
SIM 2032, 2, 12, 90.0, A1204, 6.348263 $ ALUMINUM 2219
SIM 2033, 2, 10, 90.0, A1204, 7.084390 $ ALUMINUM 2219
SIM 2034, 2, 8, 90.0, A1204, 7.511152 $ ALUMINUM 2219
SIM 2035, 2, 6, 90.0, A1204, 7.969460 $ ALUMINUM 2219
SIM 2036, 2, 4, 90.0, A1204, 8.302086 $ ALUMINUM 2219
SIM 2037, 2, 2, 90.0, A1204, 8.503784 $ ALUMINUM 2219
SIM 2038, 90.0, A1204, 8.751373 $ ALUMINUM 2219
REM REGION 1, LAYER NO. 3
SIM 2051, 2, 24, 90.0, A1204, 0.529381 $ ALUMINUM 2219
SIM 2052, 2, 22, 90.0, A1204, 1.579787 $ ALUMINUM 2219
SIM 2053, 2, 20, 90.0, A1204, 2.605285 $ ALUMINUM 2219
SIM 2054, 2, 18, 90.0, A1204, 3.596587 $ ALUMINUM 2219
SIM 2055, 2, 16, 90.0, A1204, 4.517477 $ ALUMINUM 2219
SIM 2056, 2, 14, 90.0, A1204, 5.374026 $ ALUMINUM 2219
SIM 2057, 2, 12, 90.0, A1204, 6.145826 $ ALUMINUM 2219
SIM 2058, 2, 10, 90.0, A1204, 6.820707 $ ALUMINUM 2219
SIM 2059, 2, 8, 90.0, A1204, 7.388011 $ ALUMINUM 2219
REM SURFACE NODES, OUTSIDE SURFACE, REGION 1, TANK WALL
GEN 3001, 25, 1, 90.0, -1.000000 $ SURFACE NODES
REM DIFFUSION NODES, REGION 2, OUTSIDE LAYER 1
REM REGION 2, LAYER NO. 1
SIM 4001, 2, 24, 90.0, A1204, 0.837483 $ ALUMINUM 2219
SIM 4002, 2, 22, 90.0, A1204, 2.499235 $ ALUMINUM 2219
SIM 4003, 2, 20, 90.0, A1204, 4.215567 $ ALUMINUM 2219
SIM 4004, 2, 18, 90.0, A1204, 5.678905 $ ALUMINUM 2219
SIM 4005, 2, 16, 90.0, A1204, 7.146675 $ ALUMINUM 2219
SIM 4006, 2, 14, 90.0, A1204, 8.501743 $ ALUMINUM 2219
SIM 4007, 2, 12, 90.0, A1204, 9.722736 $ ALUMINUM 2219
SIM 4008, 2, 10, 90.0, A1204, 10.790398 $ ALUMINUM 2219
SIM 4009, 2, 8, 90.0, A1204, 11.687861 $ ALUMINUM 2219
SIM 4010, 2, 6, 90.0, A1204, 12.401039 $ ALUMINUM 2219
SIM 4011, 2, 4, 90.0, A1204, 12.918633 $ ALUMINUM 2219
SIM 4012, 2, 2, 90.0, A1204, 13.232487 $ ALUMINUM 2219
SIM 4013, 2, 0, 90.0, A1204, 13.337662 $ ALUMINUM 2219
REM SURFACE NODES, OUTSIDE SURFACE, REGION 2, OUTSIDE LAYER 1
GEN 5001, 25, 1, 90.0, -1.000000 $ SURFACE NODES
REM DIFFUSION NODES, REGION 3, OUTSIDE LAYER 2
REM REGION 3, LAYER NO. 1
SIM 6001, 2, 24, 80.0, A1202, 0.343986 $ STAINLESS 347
SIM 6002, 2, 22, 80.0, A1202, 1.028528 $ STAINLESS 347
SIM 6003, 2, 20, 80.0, A1202, 1.692882 $ STAINLESS 347
SIM 6004, 2, 18, 80.0, A1202, 2.332538 $ STAINLESS 347
SIM 6005, 2, 16, 80.0, A1202, 2.935409 $ STAINLESS 347
SIM 6006, 2, 14, 80.0, A1202, 3.491985 $ STAINLESS 347
SIM 6007, 2, 12, 80.0, A1202, 3.953491 $ STAINLESS 347
SIM 6008, 2, 10, 80.0, A1202, 4.432021 $ STAINLESS 347
SIM 6009, 2, 8, 80.0, A1202, 4.800650 $ STAINLESS 347
SIM 6010, 2, 6, 80.0, A1202, 5.093573 $ STAINLESS 347
SIM 6011, 2, 4, 80.0, A1202, 5.306167 $ STAINLESS 347
SIM 6012, 2, 2, 80.0, A1202, 5.435080 $ STAINLESS 347
SIM 6013, 2, 0, 80.0, A1202, 5.476277 $ STAINLESS 347
REM REGION 3, LAYER NO. 2
SIM 6026, 2, 24, 80.0, A1202, 0.340702 $ STAINLESS 347
SIM 6027, 2, 22, 80.0, A1202, 1.016728 $ STAINLESS 347
SIM 6028, 2, 20, 80.0, A1202, 1.676720 $ STAINLESS 347
SIM 6029, 2, 18, 80.0, A1202, 2.310270 $ STAINLESS 347
SIM 6030, 2, 16, 80.0, A1202, 2.907384 $ STAINLESS 347
SIM 6031, 2, 14, 80.0, A1202, 3.458648 $ STAINLESS 347
SIM 6032, 2, 12, 80.0, A1202, 3.953569 $ STAINLESS 347
SIM 6033, 2, 10, 80.0, A1202, 4.399710 $ STAINLESS 347
SIM 6034, 2, 8, 80.0, A1202, 4.754818 $ STAINLESS 347
SIM 6035, 2, 6, 80.0, A1202, 5.044944 $ STAINLESS 347
SIM 6036, 2, 4, 80.0, A1202, 5.255507 $ STAINLESS 347
SIM 6037, 2, 2, 80.0, A1202, 5.383190 $ STAINLESS 347
SIM 6038, 80.0, A1202, 5.425976 $ STAINLESS 347
REM SURFACE NODES, OUTSIDE SURFACE, REGION 3, OUTSIDE LAYER 2
GEN 7001, 9, 1, 80.0, -1.000000 $ SURFACE NODES
REM HEAT EXCHANGER NO. 3, REPLACES NODES 7010 THRU 7012
GEN 7013, 13, 1, 80.0, -1.000000 $ SURFACE NODES
REM DIFFUSION NODES, REGION 4, INSIDE TANK WALL
REM THIS MODEL: TANK IS 75% FULL, A 1-G CASE, ULLAGE AT TOP & FLAT
REM ULLAGE STARTS AT TANK WALL AT THETA POSITION NO. 16
REM (COUNTING FROM SOUTH POLE)
REM REGION 4, LAYER NO. 1
SIV 8001, -420.0, A1101, 3.695521 $ L-HYDROGEN
SIV 8002, -420.0, A1301, 3.695521 $ L-HYDROGEN
SIV 8002, -420.0, A1101, 11.027384 $ L-HYDROGEN
SIV 8004, -420.0, A1301, 11.027384 $ G-HYDROGEN
SIV 8005, -420.0, A1101, 18.185623 $ L-HYDROGEN
SIV 8005, -420.0, A1301, 18.185623 $ G-HYDROGEN
SIV 8006, -420.0, A1101, 25.057068 $ L-HYDROGEN
SIV 8006, -420.0, A1301, 25.057068 $ G-HYDROGEN
SIV 8005, -420.0, A1101, 37.533310 $ L-HYDROGEN
SIV 8021, -420.0, A1301, 31.533310 $ G-HYDROGEN
SIV 8006, -420.0, A1101, 37.512283 $ L-HYDROGEN
SIV 8020, -420.0, A1301, 37.512283 $ G-HYDROGEN
SIV 8007, -420.0, A1101, 42.899673 $ L-HYDROGEN
SIV 8019, -420.0, A1301, 42.899673 $ G-HYDROGEN
SIV 8008, -420.0, A1101, 47.610519 $ L-HYDROGEN
SIV 8018, -420.0, A1101, 47.610519 $ L-HYDROGEN
SIV 8009, -420.0, A1101, 51.570480 $ L-HYDROGEN
SIV 8017, -420.0, A1301, 51.570480 $ G-HYDROGEN
SIV 8010, -420.0, A1101, 54.717178 $ L-HYDROGEN
SIV 8016, -420.0, A1301, 54.717178 $ G-HYDROGEN
SIV 8011, -420.0, A1101, 57.000531 $ L-HYDROGEN
SIV 8015, -420.0, A1101, 57.000531 $ L-HYDROGEN
SIV 8012, -420.0, A1101, 58.385773 $ L-HYDROGEN
SIV 8014, -420.0, A1101, 58.385773 $ L-HYDROGEN
SIV 8013, -420.0, A1101, 58.849808 $ L-HYDROGEN
REM REGION 4, LAYER NO. 2
SIV 8026, -420.0, A1101, 3.233808 $ L-HYDROGEN
SIV 8050, -420.0, A1301, 3.233808 $ G-HYDROGEN
SIV 8027, -420.0, A1101, 9.650397 $ L-HYDROGEN
SIV 8049, -420.0, A1301, 9.650397 $ G-HYDROGEN
SIV 8028, -420.0, A1101, 15.914793 $ L-HYDROGEN
SIV 8048, -420.0, A1301, 15.914793 $ G-HYDROGEN
SIV 8029, -420.0, A1101, 21.928192 $ L-HYDROGEN
SIV 8047, -420.0, A1301, 21.928192 $ G-HYDROGEN
SIV 8030, -420.0, A1101, 27.595749 $ L-HYDROGEN
SIV 8046, -420.0, A1301, 27.595749 $ G-HYDROGEN
SIV 8031, -420.0, A1101, 32.828140 $ L-HYDROGEN
SIV 8045, -420.0, A1301, 32.828140 $ G-HYDROGEN
SIV 8032, -420.0, A1301, 37.542770 $ L-HYDROGEN
SIV 8044, -420.0, A1301, 37.542770 $ G-HYDROGEN
SIV 8033, -420.0, A1101, 41.665390 $ L-HYDROGEN
SIV 8043, -420.0, A1301, 41.665390 $ G-HYDROGEN
SIV 8034, -420.0, A1101, 45.130859 $ L-HYDROGEN
SIV 8042, -420.0, A1301, 45.130859 $ G-HYDROGEN
SIV 8035, -420.0, A1101, 47.884628 $ L-HYDROGEN
SIV 8041, -420.0, A1301, 47.884628 $ G-HYDROGEN
SIV 8036, -420.0, A1101, 49.883224 $ L-HYDROGEN
SIV 8040, -420.0, A1301, 49.883224 $ G-HYDROGEN
SIV 8037, -420.0, A1101, 51.095139 $ L-HYDROGEN
SIV 8039, -420.0, A1301, 51.095139 $ G-HYDROGEN
SIV 8038, -420.0, A1101, 53.501236 $ L-HYDROGEN
REM REGION 4, LAYER NO. 3
SIV 8051, -420.0, A1101, 2.803146 $ L-HYDROGEN
SIV 8075, -420.0, A1301, 2.803146 $ G-HYDROGEN
SIV 8052, -423.0, A1101, 8.365207 $ L-HYDROGEN
SIV 8074, -423.0, A1301, 8.365207 $ G-HYDROGEN
SIV 8053, -423.0, A1101, 13.795338 $ L-HYDROGEN
SIV 8073, -423.0, A1301, 13.795338 $ G-HYDROGEN
SIV 8054, -420.0, A1101, 19.007904 $ L-HYDROGEN
SIV 8072, -420.0, A1301, 19.007904 $ G-HYDROGEN
SIV 8055, -420.0, A1101, 23.920685 $ L-HYDROGEN
SIV 8071, -420.0, A1301, 23.920685 $ G-HYDROGEN
SIV 8056, -420.0, A1101, 28.456238 $ L-HYDROGEN
SIV 8070, -420.0, A1301, 28.456238 $ G-HYDROGEN
SIV 8057, -420.0, A1101, 32.543030 $ L-HYDROGEN
SIV 8059, -420.0, A1301, 32.543030 $ G-HYDROGEN
SIV 8058, -420.0, A1101, 36.116608 $ L-HYDROGEN
SIV 8058, -420.0, A1301, 36.116608 $ G-HYDROGEN
SIV 8059, -420.0, A1101, 39.120575 $ L-HYDROGEN
SIV 8067, -420.0, A1301, 39.120575 $ G-HYDROGEN
SIV 8060, -420.0, A1101, 41.507599 $ L-HYDROGEN
SIV 8066, -420.0, A1301, 41.507599 $ G-HYDROGEN
SIV 8061, -420.0, A1101, 43.240051 $ L-HYDROGEN
SIV 8061, -420.0, A1301, 43.240051 $ G-HYDROGEN
SIV 8062, -420.0, A1101, 44.290558 $ L-HYDROGEN
SIV 8064, -420.0, A1101, 44.290558 $ G-HYDROGEN
SIV 8063, -420.0, A1101, 44.642563 $ L-HYDROGEN
REM REGION 4, LAYER NO. 4
SIV 8076, -420.0, A1101, 2.403246 $ L-HYDROGEN
<table>
<thead>
<tr>
<th>SIV</th>
<th>X, Y, Z</th>
<th>X, Y, Z</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>8100</td>
<td>-420.0, A1101</td>
<td>2.403246</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>8077</td>
<td>-420.0, A1101</td>
<td>7.171817</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8099</td>
<td>-420.0, A1101</td>
<td>7.171817</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>8078</td>
<td>-420.0, A1101</td>
<td>11.827278</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8079</td>
<td>-420.0, A1101</td>
<td>11.827278</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>8097</td>
<td>-420.0, A1101</td>
<td>16.296219</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8080</td>
<td>-420.0, A1101</td>
<td>16.296219</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>8096</td>
<td>-420.0, A1101</td>
<td>20.508133</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8081</td>
<td>-420.0, A1101</td>
<td>20.508133</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>8095</td>
<td>-420.0, A1101</td>
<td>24.396637</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8082</td>
<td>-420.0, A1101</td>
<td>24.396637</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>8094</td>
<td>-420.0, A1101</td>
<td>27.900406</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8083</td>
<td>-420.0, A1101</td>
<td>30.964172</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8093</td>
<td>-420.0, A1101</td>
<td>30.964172</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>8084</td>
<td>-420.0, A1101</td>
<td>33.539597</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8092</td>
<td>-420.0, A1101</td>
<td>33.539597</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>8085</td>
<td>-420.0, A1101</td>
<td>35.586060</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8091</td>
<td>-420.0, A1101</td>
<td>35.586060</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8086</td>
<td>-420.0, A1101</td>
<td>37.071350</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8089</td>
<td>-420.0, A1101</td>
<td>37.071350</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>8090</td>
<td>-420.0, A1101</td>
<td>37.972000</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8087</td>
<td>-420.0, A1101</td>
<td>37.972000</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>8088</td>
<td>-420.0, A1101</td>
<td>38.723804</td>
<td>L-HYDROGEN</td>
</tr>
</tbody>
</table>

**REM**  SURFACE NODES, OUTSIDE SURFACE, REGION 5, INSIDE TANK AT CENTER
**GEN**  9001, 25.1, -420.0, -1.000000, SURFACE NODES
**REM**  DIFFUSION NODES, REGION 5, INSIDE TANK AT CENTER
**REM**  REGION 5, LAYER NO. 1
<table>
<thead>
<tr>
<th>SIV</th>
<th>X, Y, Z</th>
<th>X, Y, Z</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>10001</td>
<td>-420.0, A1101</td>
<td>1.655709</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10025</td>
<td>-420.0, A1101</td>
<td>1.655709</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10002</td>
<td>-420.0, A1101</td>
<td>4.941003</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10024</td>
<td>-420.0, A1101</td>
<td>4.941003</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10003</td>
<td>-420.0, A1101</td>
<td>8.148369</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10023</td>
<td>-420.0, A1101</td>
<td>8.148369</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10004</td>
<td>-420.0, A1101</td>
<td>11.227234</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10022</td>
<td>-420.0, A1101</td>
<td>11.227234</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10005</td>
<td>-420.0, A1101</td>
<td>14.129028</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10021</td>
<td>-420.0, A1101</td>
<td>14.129028</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10006</td>
<td>-420.0, A1101</td>
<td>16.807999</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10007</td>
<td>-420.0, A1101</td>
<td>19.221909</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10019</td>
<td>-420.0, A1101</td>
<td>19.221909</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10008</td>
<td>-420.0, A1101</td>
<td>21.332672</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10018</td>
<td>-420.0, A1101</td>
<td>21.332672</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10009</td>
<td>-420.0, A1101</td>
<td>23.107010</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10017</td>
<td>-420.0, A1101</td>
<td>23.107010</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10010</td>
<td>-420.0, A1101</td>
<td>24.516937</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10016</td>
<td>-420.0, A1101</td>
<td>24.516937</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10011</td>
<td>-420.0, A1101</td>
<td>25.540222</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10015</td>
<td>-420.0, A1101</td>
<td>25.540222</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10012</td>
<td>-420.0, A1101</td>
<td>26.160721</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10014</td>
<td>-420.0, A1101</td>
<td>26.160721</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10013</td>
<td>-420.0, A1101</td>
<td>26.368637</td>
<td>L-HYDROGEN</td>
</tr>
</tbody>
</table>

**REM**  REGION 5, LAYER NO. 2
<table>
<thead>
<tr>
<th>SIV</th>
<th>X, Y, Z</th>
<th>X, Y, Z</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>10026</td>
<td>-420.0, A1101</td>
<td>1.435211</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10050</td>
<td>-420.0, A1301</td>
<td>1.435211</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10027</td>
<td>-420.0, A1101</td>
<td>4.282986</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10049</td>
<td>-420.0, A1301</td>
<td>4.282986</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10028</td>
<td>-420.0, A1101</td>
<td>7.063215</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10048</td>
<td>-420.0, A1301</td>
<td>7.063215</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10029</td>
<td>-420.0, A1101</td>
<td>9.732048</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10047</td>
<td>-420.0, A1301</td>
<td>9.732048</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10030</td>
<td>-420.0, A1101</td>
<td>12.247398</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10046</td>
<td>-420.0, A1101</td>
<td>12.247398</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10031</td>
<td>-420.0, A1101</td>
<td>14.569603</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10045</td>
<td>-420.0, A1301</td>
<td>14.569603</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10032</td>
<td>-420.0, A1101</td>
<td>16.662033</td>
<td>L-HYDROGEN</td>
</tr>
<tr>
<td>10044</td>
<td>-420.0, A1301</td>
<td>16.662033</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10033</td>
<td>-420.0, A1101</td>
<td>18.491699</td>
<td>G-HYDROGEN</td>
</tr>
<tr>
<td>10043</td>
<td>-420.0, A1301</td>
<td>18.491699</td>
<td>G-HYDROGEN</td>
</tr>
</tbody>
</table>
SIV 10034, -420.0, A1101, 20.029724 $\triangle$ L-HYDROGEN
SIV 10042, -420.0, A1301, 20.029724 $\triangle$ G-HYDROGEN
SIV 10035, -420.0, A1101, 21.251877 $\triangle$ L-HYDROGEN
SIV 10041, -420.0, A1101, 21.251877 $\triangle$ L-HYDROGEN
SIV 10036, -420.0, A1101, 22.138885 $\triangle$ L-HYDROGEN
SIV 10040, -420.0, A1101, 22.138885 $\triangle$ L-HYDROGEN
SIV 10037, -420.0, A1101, 22.676758 $\triangle$ L-HYDROGEN
SIV 10039, -420.0, A1101, 22.676758 $\triangle$ L-HYDROGEN
SIV 10038, -420.0, A1101, 22.856995 $\triangle$ L-HYDROGEN

REM REGION 5, LAYER NO. 3
SIV 10051, -420.0, A1101, 1.230462 $\triangle$ L-HYDROGEN
SIV 10075, -420.0, A1301, 1.230462 $\triangle$ G-HYDROGEN
SIV 10052, -420.0, A1301, 3.671969 $\triangle$ L-HYDROGEN
SIV 10074, -420.0, A1301, 3.671969 $\triangle$ G-HYDROGEN
SIV 10053, -420.0, A1301, 6.055569 $\triangle$ L-HYDROGEN
SIV 10073, -420.0, A1301, 6.055569 $\triangle$ G-HYDROGEN
SIV 10054, -420.0, A1101, 8.343666 $\triangle$ L-HYDROGEN
SIV 10072, -420.0, A1301, 8.343666 $\triangle$ G-HYDROGEN
SIV 10055, -420.0, A1101, 10.500168 $\triangle$ L-HYDROGEN
SIV 10071, -420.0, A1301, 10.500168 $\triangle$ G-HYDROGEN
SIV 10058, -420.0, A1301, 12.491089 $\triangle$ L-HYDROGEN
SIV 10070, -420.0, A1301, 12.491089 $\triangle$ G-HYDROGEN
SIV 10057, -420.0, A1101, 14.285011 $\triangle$ L-HYDROGEN
SIV 10069, -420.0, A1301, 14.285011 $\triangle$ G-HYDROGEN
SIV 10058, -420.0, A1101, 15.853661 $\triangle$ L-HYDROGEN
SIV 10068, -420.0, A1301, 15.853661 $\triangle$ G-HYDROGEN
SIV 10059, -420.0, A1101, 17.192266 $\triangle$ L-HYDROGEN
SIV 10067, -420.0, A1101, 17.192266 $\triangle$ G-HYDROGEN
SIV 10060, -420.0, A1101, 18.220062 $\triangle$ L-HYDROGEN
SIV 10066, -420.0, A1101, 18.220062 $\triangle$ G-HYDROGEN
SIV 10061, -420.0, A1101, 18.980530 $\triangle$ L-HYDROGEN
SIV 10065, -420.0, A1101, 18.980530 $\triangle$ G-HYDROGEN
SIV 10062, -420.0, A1101, 19.441666 $\triangle$ L-HYDROGEN
SIV 10064, -420.0, A1101, 19.441666 $\triangle$ G-HYDROGEN
SIV 10063, -420.0, A1101, 19.596176 $\triangle$ L-HYDROGEN

REM REGION 5, LAYER NO. 4
SIV 10076, -420.0, A1101, 1.041463 $\triangle$ L-HYDROGEN
SIV 10100, -420.0, A1301, 1.041463 $\triangle$ G-HYDROGEN
SIV 10077, -420.0, A1301, 3.107956 $\triangle$ L-HYDROGEN
SIV 10099, -420.0, A1301, 3.107956 $\triangle$ G-HYDROGEN
SIV 10078, -420.0, A1101, 5.125433 $\triangle$ L-HYDROGEN
SIV 10098, -420.0, A1301, 5.125433 $\triangle$ G-HYDROGEN
SIV 10079, -420.0, A1101, 7.062079 $\triangle$ L-HYDROGEN
SIV 10097, -420.0, A1301, 7.062079 $\triangle$ G-HYDROGEN
SIV 10080, -420.0, A1301, 8.887344 $\triangle$ L-HYDROGEN
SIV 10096, -420.0, A1301, 8.887344 $\triangle$ G-HYDROGEN
SIV 10081, -420.0, A1301, 10.572456 $\triangle$ L-HYDROGEN
SIV 10095, -420.0, A1301, 10.572456 $\triangle$ G-HYDROGEN
SIV 10082, -420.0, A1101, 12.090836 $\triangle$ L-HYDROGEN
SIV 10094, -420.0, A1301, 12.090836 $\triangle$ G-HYDROGEN
SIV 10083, -420.0, A1301, 13.418541 $\triangle$ L-HYDROGEN
SIV 10093, -420.0, A1301, 13.418541 $\triangle$ G-HYDROGEN
SIV 10084, -420.0, A1101, 14.534615 $\triangle$ L-HYDROGEN
SIV 10092, -420.0, A1101, 14.534615 $\triangle$ G-HYDROGEN
SIV 10085, -420.0, A1101, 15.421478 $\triangle$ L-HYDROGEN
SIV 10091, -420.0, A1101, 15.421478 $\triangle$ G-HYDROGEN
SIV 10086, -420.0, A1101, 16.065125 $\triangle$ L-HYDROGEN
SIV 10090, -420.0, A1101, 16.065125 $\triangle$ G-HYDROGEN
SIV 10087, -420.0, A1101, 16.455414 $\triangle$ L-HYDROGEN
SIV 10089, -420.0, A1101, 16.455414 $\triangle$ G-HYDROGEN
SIV 10088, -420.0, A1101, 16.586212 $\triangle$ L-HYDROGEN

REM REGION 5, LAYER NO. 5
SIV 10101, -420.0, A1101, 0.868214 $\triangle$ L-HYDROGEN
SIV 10125, -420.0, A1301, 0.868214 $\triangle$ G-HYDROGEN
SIV 10102, -420.0, A1301, 2.590941 $\triangle$ L-HYDROGEN
SIV 10124, -420.0, A1301, 2.590941 $\triangle$ G-HYDROGEN
SIV 10103, -420.0, A1101, 4.272809 $\triangle$ L-HYDROGEN
SIV 10123, -420.0, A1301, 4.272809 $\triangle$ G-HYDROGEN
SIV 10104, -420.0, A1101, 5.887291 $\triangle$ L-HYDROGEN
SIV 10122, -420.0, A1301, 5.887291 $\triangle$ G-HYDROGEN
<table>
<thead>
<tr>
<th>No.</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10290</td>
<td>-420.0</td>
<td>1301</td>
<td>3.527408 G-HYDROGEN</td>
</tr>
<tr>
<td>2</td>
<td>10291</td>
<td>-420.0</td>
<td>1301</td>
<td>3.527408 G-HYDROGEN</td>
</tr>
<tr>
<td>3</td>
<td>10292</td>
<td>-420.0</td>
<td>1301</td>
<td>3.674633 G-HYDROGEN</td>
</tr>
<tr>
<td>4</td>
<td>10293</td>
<td>-420.0</td>
<td>1301</td>
<td>3.674633 G-HYDROGEN</td>
</tr>
<tr>
<td>5</td>
<td>10294</td>
<td>-420.0</td>
<td>1301</td>
<td>3.763907 G-HYDROGEN</td>
</tr>
<tr>
<td>6</td>
<td>10295</td>
<td>-420.0</td>
<td>1301</td>
<td>3.763907 G-HYDROGEN</td>
</tr>
<tr>
<td>7</td>
<td>10296</td>
<td>-420.0</td>
<td>1301</td>
<td>3.793823 G-HYDROGEN</td>
</tr>
</tbody>
</table>

**REM REGION 5, LAYER NO. 11**

<table>
<thead>
<tr>
<th>No.</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10297</td>
<td>-420.0</td>
<td>1301</td>
<td>0.159468 G-HYDROGEN</td>
</tr>
<tr>
<td>2</td>
<td>10298</td>
<td>-420.0</td>
<td>1301</td>
<td>0.159468 G-HYDROGEN</td>
</tr>
<tr>
<td>3</td>
<td>10299</td>
<td>-420.0</td>
<td>1301</td>
<td>0.475887 G-HYDROGEN</td>
</tr>
<tr>
<td>4</td>
<td>10300</td>
<td>-420.0</td>
<td>1301</td>
<td>0.475887 G-HYDROGEN</td>
</tr>
<tr>
<td>5</td>
<td>10301</td>
<td>-420.0</td>
<td>1301</td>
<td>0.784801 G-HYDROGEN</td>
</tr>
<tr>
<td>6</td>
<td>10302</td>
<td>-420.0</td>
<td>1301</td>
<td>0.784801 G-HYDROGEN</td>
</tr>
<tr>
<td>7</td>
<td>10303</td>
<td>-420.0</td>
<td>1301</td>
<td>1.081338 G-HYDROGEN</td>
</tr>
<tr>
<td>8</td>
<td>10304</td>
<td>-420.0</td>
<td>1301</td>
<td>1.081338 G-HYDROGEN</td>
</tr>
<tr>
<td>9</td>
<td>10305</td>
<td>-420.0</td>
<td>1301</td>
<td>1.360822 G-HYDROGEN</td>
</tr>
<tr>
<td>10</td>
<td>10306</td>
<td>-420.0</td>
<td>1301</td>
<td>1.360822 G-HYDROGEN</td>
</tr>
<tr>
<td>11</td>
<td>10307</td>
<td>-420.0</td>
<td>1301</td>
<td>1.618865 G-HYDROGEN</td>
</tr>
<tr>
<td>12</td>
<td>10308</td>
<td>-420.0</td>
<td>1301</td>
<td>1.618865 G-HYDROGEN</td>
</tr>
</tbody>
</table>

**REM REGION 5, LAYER NO. 12**

<table>
<thead>
<tr>
<th>No.</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10309</td>
<td>-420.0</td>
<td>1301</td>
<td>0.096468 G-HYDROGEN</td>
</tr>
<tr>
<td>2</td>
<td>10310</td>
<td>-420.0</td>
<td>1301</td>
<td>0.096468 G-HYDROGEN</td>
</tr>
<tr>
<td>3</td>
<td>10311</td>
<td>-420.0</td>
<td>1301</td>
<td>0.287883 G-HYDROGEN</td>
</tr>
<tr>
<td>4</td>
<td>10312</td>
<td>-420.0</td>
<td>1301</td>
<td>0.287883 G-HYDROGEN</td>
</tr>
<tr>
<td>5</td>
<td>10313</td>
<td>-420.0</td>
<td>1301</td>
<td>0.474757 G-HYDROGEN</td>
</tr>
<tr>
<td>6</td>
<td>10314</td>
<td>-420.0</td>
<td>1301</td>
<td>0.474757 G-HYDROGEN</td>
</tr>
<tr>
<td>7</td>
<td>10315</td>
<td>-420.0</td>
<td>1301</td>
<td>0.654143 G-HYDROGEN</td>
</tr>
<tr>
<td>8</td>
<td>10316</td>
<td>-420.0</td>
<td>1301</td>
<td>0.654143 G-HYDROGEN</td>
</tr>
<tr>
<td>9</td>
<td>10317</td>
<td>-420.0</td>
<td>1301</td>
<td>0.823214 G-HYDROGEN</td>
</tr>
<tr>
<td>10</td>
<td>10318</td>
<td>-420.0</td>
<td>1301</td>
<td>0.823214 G-HYDROGEN</td>
</tr>
<tr>
<td>11</td>
<td>10319</td>
<td>-420.0</td>
<td>1301</td>
<td>0.979301 G-HYDROGEN</td>
</tr>
<tr>
<td>12</td>
<td>10320</td>
<td>-420.0</td>
<td>1301</td>
<td>0.979301 G-HYDROGEN</td>
</tr>
<tr>
<td>13</td>
<td>10321</td>
<td>-420.0</td>
<td>1301</td>
<td>1.119945 G-HYDROGEN</td>
</tr>
<tr>
<td>14</td>
<td>10322</td>
<td>-420.0</td>
<td>1301</td>
<td>1.119945 G-HYDROGEN</td>
</tr>
<tr>
<td>15</td>
<td>10323</td>
<td>-420.0</td>
<td>1301</td>
<td>1.242927 G-HYDROGEN</td>
</tr>
<tr>
<td>16</td>
<td>10324</td>
<td>-420.0</td>
<td>1301</td>
<td>1.242927 G-HYDROGEN</td>
</tr>
<tr>
<td>17</td>
<td>10325</td>
<td>-420.0</td>
<td>1301</td>
<td>1.346305 G-HYDROGEN</td>
</tr>
<tr>
<td>18</td>
<td>10326</td>
<td>-420.0</td>
<td>1301</td>
<td>1.346305 G-HYDROGEN</td>
</tr>
<tr>
<td>19</td>
<td>10327</td>
<td>-420.0</td>
<td>1301</td>
<td>1.488074 G-HYDROGEN</td>
</tr>
<tr>
<td>20</td>
<td>10328</td>
<td>-420.0</td>
<td>1301</td>
<td>1.488074 G-HYDROGEN</td>
</tr>
<tr>
<td>21</td>
<td>10329</td>
<td>-420.0</td>
<td>1301</td>
<td>1.524227 G-HYDROGEN</td>
</tr>
<tr>
<td>22</td>
<td>10330</td>
<td>-420.0</td>
<td>1301</td>
<td>1.524227 G-HYDROGEN</td>
</tr>
<tr>
<td>23</td>
<td>10331</td>
<td>-420.0</td>
<td>1301</td>
<td>1.536342 G-HYDROGEN</td>
</tr>
<tr>
<td>SIV 10306</td>
<td>-420.0, A1301</td>
<td>0.499644 $ G-HYDROGEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIV 10307</td>
<td>-420.0, A1301</td>
<td>0.571401 $ G-HYDROGEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIV 10308</td>
<td>-420.0, A1301</td>
<td>0.63147 $ G-HYDROGEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIV 10309</td>
<td>-420.0, A1301</td>
<td>0.686891 $ G-HYDROGEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIV 10310</td>
<td>-420.0, A1301</td>
<td>0.728803 $ G-HYDROGEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIV 10311</td>
<td>-420.0, A1301</td>
<td>0.759222 $ G-HYDROGEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIV 10312</td>
<td>-420.0, A1301</td>
<td>0.777667 $ G-HYDROGEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIV 10313</td>
<td>-420.0, A1301</td>
<td>0.783848 $ G-HYDROGEN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REM REGION 5, LAYER NO. 14
| SIV 10326 | -420.0, A1301 | 0.017719 $ G-HYDROGEN |
| SIV 10327 | -420.0, A1301 | 0.052876 $ G-HYDROGEN |
| SIV 10328 | -420.0, A1301 | 0.08200 $ G-HYDROGEN |
| SIV 10329 | -420.0, A1301 | 0.120149 $ G-HYDROGEN |
| SIV 10330 | -420.0, A1301 | 0.151202 $ G-HYDROGEN |
| SIV 10331 | -420.0, A1301 | 0.179872 $ G-HYDROGEN |
| SIV 10332 | -420.0, A1301 | 0.205704 $ G-HYDROGEN |
| SIV 10333 | -420.0, A1301 | 0.228293 $ G-HYDROGEN |
| SIV 10334 | -420.0, A1301 | 0.247281 $ G-HYDROGEN |
| SIV 10335 | -420.0, A1301 | 0.262639 $ G-HYDROGEN |
| SIV 10336 | -420.0, A1301 | 0.273320 $ G-HYDROGEN |
| SIV 10337 | -420.0, A1301 | 0.279960 $ G-HYDROGEN |
| SIV 10338 | -420.0, A1301 | 0.282185 $ G-HYDROGEN |

REM REGION 5, LAYER NO. 15
| SIV 10351 | -420.0, A1301 | 0.001969 $ G-HYDROGEN |
| SIV 10352 | -420.0, A1301 | 0.005875 $ G-HYDROGEN |
| SIV 10353 | -420.0, A1301 | 0.009689 $ G-HYDROGEN |
| SIV 10354 | -420.0, A1301 | 0.013530 $ G-HYDROGEN |
| SIV 10355 | -420.0, A1301 | 0.026800 $ G-HYDROGEN |
| SIV 10356 | -420.0, A1301 | 0.056800 $ G-HYDROGEN |
| SIV 10357 | -420.0, A1301 | 0.019886 $ G-HYDROGEN |
| SIV 10358 | -420.0, A1301 | 0.022856 $ G-HYDROGEN |
| SIV 10359 | -420.0, A1301 | 0.025366 $ G-HYDROGEN |
| SIV 10360 | -420.0, A1301 | 0.028536 $ G-HYDROGEN |
| SIV 10361 | -420.0, A1301 | 0.030369 $ G-HYDROGEN |
| SIV 10362 | -420.0, A1301 | 0.031107 $ G-HYDROGEN |
| SIV 10363 | -420.0, A1301 | 0.031354 $ G-HYDROGEN |

-20002, -424.0, 1.000000 $ HEAT EXCHANGER 2
-20003, -400.0, 1.000000 $ HEAT EXCHANGER 3

77
<table>
<thead>
<tr>
<th>Layer 1 Field</th>
<th>Layer 2 Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>140.0, 1.000000</td>
<td>$ OUTSIDE ATMOS 1</td>
</tr>
<tr>
<td>440.0, 1.000000</td>
<td>$ OUTSIDE ATMOS 2</td>
</tr>
</tbody>
</table>
### REM RADIAL CONDUCTORS REGION 2, LAYER 1 TO BOUNDARY 2-3

| SIM  | 2, 1, 301, 0, 401, 0, A6204, 4.108946E+02 |
| SIM  | 2, 1, 302, 2, 402, 2, A6204, 4.208772E+02 |
| SIM  | 2, 1, 303, 4, 403, 4, A6204, 5.787063E+02 |

### REM CIRCUMFERENTIAL CONDUCTORS: X-DIRECTION, CONDUCTION

| SIM  | 2, 1, 4001, 23, 4002, 23, A6204, 2.493440E-01, A6204, 7.440972E-01 |
| SIM  | 2, 1, 4002, 21, 4003, 21, A6204, 7.440972E-01, A6204, 1.227115E+00 |

### DIM CIRCUMFERENTIAL CONDUCTORS REGION 2, LAYER NUMBER 1

| SIM  | 2, 1, 5003, 20, 5004, 20, A6202, 3.345383E+02 |
| SIM  | 2, 1, 5005, 16, 5006, 16, A6202, 5.800801E+02 |

### REM RADIAL CONDUCTORS REGION 3, LAYER 1 TO BOUNDARY 3-2

| SIM  | 2, 1, 6001, 24, 6002, 24, A6202, 6.797668E+01 |
| SIM  | 2, 1, 6002, 22, 6003, 22, A6202, 2.028573E+02 |

### DIM RADIAL CONDUCTORS REGION 2, LAYER 1 TO LAYER 2

| SIM  | 2, 1, 6004, 18, 6005, 18, A6202, 4.853926E+02 |
| SIM  | 2, 1, 6005, 16, 6006, 16, A6202, 8.758328E+02 |

### REM RADIAL CONDUCTORS REGION 3, LAYER 2 TO BOUNDARY 3-4

| SIM  | 2, 1, 7001, 0, 7002, 0, A6202, 1.082588E+03 |

### DIM RADIAL CONDUCTORS REGION 2, LAYER 2 TO BOUNDARY 2-3

| SIM  | 2, 1, 8001, 0, 8002, 0, A6202, 1.084617E+03 |

### DIM RADIAL CONDUCTORS REGION 3, LAYER 1 TO BOUNDARY 3-4

| SIM  | 2, 1, 9001, 0, 9002, 0, A6202, 1.036466E+03 |

### DIM RADIAL CONDUCTORS REGION 4, LAYER 4 TO BOUNDARY 4-5

| SIM  | 2, 1, 1001, 0, 1002, 0, A6202, 1.079590E+03 |

### DIM CIRCUMFERENTIAL CONDUCTORS REGION 3, LAYER 1 TO BOUNDARY 3-4

| SIM  | 2, 1, 1003, 0, 1004, 0, A6202, 1.044122E+03 |
DIM 322, 2.1, 6001, 23, 6002, 23, A6202, 9.973752E-02, A6202, 2.976390E-01
DIM 324, 2.1, 6002, 21, 6003, 21, A6202, 2.976390E-01, A6202, 4.908462E-01
DIM 326, 2.1, 6003, 19, 6004, 19, A6202, 4.908462E-01, A6202, 6.763123E-01
DIM 328, 2.1, 6004, 17, 6005, 17, A6202, 6.763123E-01, A6202, 8.511126E-01
DIM 330, 2.1, 6005, 15, 6006, 15, A6202, 8.511126E-01, A6202, 1.012489E+00
DIM 332, 2.1, 6006, 13, 6007, 13, A6202, 1.012489E+00, A6202, 1.157900E+00
DIM 334, 2.1, 6007, 11, 6008, 11, A6202, 1.157900E+00, A6202, 1.285050E+00
DIM 336, 2.1, 6008, 9, 6009, 9, A6202, 1.285050E+00, A6202, 1.391332E+00
DIM 338, 2.1, 6009, 7, 6010, 7, A6202, 1.391332E+00, A6202, 1.476656E+00
DIM 340, 2.1, 6010, 5, 6011, 5, A6202, 1.476656E+00, A6202, 1.538506E+00
DIM 342, 2.1, 6011, 3, 6012, 3, A6202, 1.538506E+00, A6202, 1.575884E+00
DIM 344, 2.1, 6012, 1, 6013, 1, A6202, 1.575884E+00, A6202, 1.588408E+00

REM CIRCUMFERENTIAL CONDUCTORS, REGION 3, LAYER NUMBER 2
DIM 346, 2.1, 6026, 23, 6027, 23, A6202, 9.973752E-02, A6202, 2.976390E-01
DIM 348, 2.1, 6027, 21, 6028, 21, A6202, 2.976390E-01, A6202, 4.908462E-01
DIM 350, 2.1, 6028, 19, 6029, 19, A6202, 4.908462E-01, A6202, 6.763123E-01
DIM 352, 2.1, 6029, 17, 6030, 17, A6202, 6.763123E-01, A6202, 8.511126E-01
DIM 354, 2.1, 6030, 15, 6031, 15, A6202, 8.511126E-01, A6202, 1.012489E+00
DIM 356, 2.1, 6031, 13, 6032, 13, A6202, 1.012489E+00, A6202, 1.157900E+00
DIM 358, 2.1, 6032, 11, 6033, 11, A6202, 1.157900E+00, A6202, 1.285050E+00
DIM 360, 2.1, 6033, 9, 6034, 9, A6202, 1.285050E+00, A6202, 1.391332E+00
DIM 362, 2.1, 6034, 7, 6035, 7, A6202, 1.391332E+00, A6202, 1.476656E+00
DIM 364, 2.1, 6035, 5, 6036, 5, A6202, 1.476656E+00, A6202, 1.538506E+00
DIM 366, 2.1, 6036, 3, 6037, 3, A6202, 1.538506E+00, A6202, 1.575884E+00
DIM 368, 2.1, 6037, 1, 6038, 1, A6202, 1.575884E+00, A6202, 1.588408E+00

REM RADIAL CONDUCTORS, REGION 4, LAYER 1 TO BOUNDARY
REM RADIAL CONDUCTORS, LAYER 1 TO BOUNDARY 4 - 1

REM CIRCUMFERENTIAL CONDUCTORS, LAYER 1 TO BOUNDARY
REM CIRCUMFERENTIAL CONDUCTORS; Y-DIRECTION, CONDUCTION

REM CIRCUMFERENTIAL CONDUCTORS REGION 4, LAYER NUMBER 1

DIV 520, 8001, 8002, A6101, 3.720487E+00
DIV 522, 8002, 8003, A6101, 6.166777E+01
DIV 524, 8003, 8004, A6101, 6.165999E+01
DIV 526, 8004, 8005, A6101, 6.316599E+01
DIV 528, 8005, 8006, A6101, 6.315599E+01
DIV 530, 8006, 8007, A6101, 6.315599E+01
DIV 532, 8007, 8008, A6101, 6.315599E+01
DIV 534, 8008, 8009, A6101, 6.315599E+01
DIV 536, 8009, 8010, A6101, 6.315599E+01
DIV 1350, 10363, 10364, A6301, 1.57588E+01, A6301, 1.58840E+01
REM CONVECTION CONDUCTORS; ATMOSPHERE TO OUTER SURFACE
GEN 1352, 2,1,20301, 0, 7001,24,0.00001E-79,0.00000E+00, 4.80, 0.00
GEN 1354, 2,1,20301, 0, 7002,22,0.00001E-79,0.00000E+00,14.33, 0.00
GEN 1356, 2,1,20301, 0, 7003,20,0.00001E-79,0.00000E+00,23.62, 0.00
GEN 1358, 2,1,20301, 0, 7004,18,0.00001E-79,0.00000E+00,32.55, 0.00
GEN 1360, 2,1,20301, 0, 7005,16,0.00001E-79,0.00000E+00,40.96, 0.00
GEN 1362, 2,1,20301, 0, 7006,14,0.00001E-79,0.00000E+00,48.73, 0.00
GEN 1364, 2,1,20301, 0, 7007,12,0.00001E-79,0.00000E+00,55.73, 0.00
GEN 1366, 2,1,20301, 0, 7008,10,0.00001E-79,0.00000E+00,61.85, 0.00
GEN 1368, 2,1,20301, 0, 7009, 8,0.00001E-79,0.00000E+00,67.00, 0.00
GEN 1370, 2,1,20301, 0, 7010, 6,0.00001E-79,0.00000E+00,71.08, 0.00
GEN 1372, 2,1,20301, 0, 7011, 4,0.00001E-79,0.00000E+00,74.05, 0.00
GEN 1374, 2,1,20301, 0, 7012, 2,0.00001E-79,0.00000E+00,75.85, 0.00
GEN 1376, 1,1,20301, 0, 7013, 0,0.00001E-79,0.00000E+00,76.45, 0.00
REM RADIATION CONDUCTORS; ATMOSPHERE TO OUTER SURFACE
GEN 1377, 2,1,20302, 0, 7001,24,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1379, 2,1,20302, 0, 7002,22,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1381, 2,1,20302, 0, 7003,20,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1383, 2,1,20302, 0, 7004,18,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1385, 2,1,20302, 0, 7005,16,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1387, 2,1,20302, 0, 7006,14,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1389, 2,1,20302, 0, 7007,12,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1391, 2,1,20302, 0, 7008,10,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1393, 2,1,20302, 0, 7009, 8,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1395, 2,1,20302, 0, 7010, 6,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1397, 2,1,20302, 0, 7011, 4,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1399, 2,1,20302, 0, 7012, 2,0.00001E-79,0.00000E+00, 0.00, 0.00
GEN 1401, 1,1,20302, 0, 7013, 0,0.00001E-79,0.00000E+00, 0.00, 0.00
REM 3CONSTANTS DATA
REM NTHETA NBETAS BETAN RIN TVOL
1=  25, 2=  1, 3= 1000, 4= 20000, 5= 19.393
REM K10=SINDA TEMP UNITS; K10=1(DEG F); K10=2(DEG R)
REM 10=  1
REM TMIN (MIN) TMIN (MIN) TMIN (MIN) TMIN (MIN)
REM 0.000000E+00 100.00 0.100000E-02 0.25000
101-0.000000E+00, 102- 2.0000, 103-0.16667E-04, 104=0.41667E-02
NLDUR= 500, NRLXCA=0.001000, ARCLXCA=0.001000
END
REM 3ARRAY DATA
REM CONDUCTIVITY BTU/(INCH.HR.F) FOR ALUMINUM 2219
6204
-442., 0.44400E+00, -424., 0.89712E+00, -406., 0.12954E+01
-408., 0.37243E+01, -370., 0.21491E+01, -352., 0.28238E+01
-334., 0.26988E+01, -316., 0.32488E+01, -298., 0.33986E+01
-280., 0.39465E+01, -190., 0.43482E+01, -100., 0.50479E+01
-10., 0.56476E+01, 80., 0.61474E+01, 170., 0.65472E+01
260., 0.68471E+01, 350., 0.70470E+01, 440., 0.72968E+01
530., 0.75968E+01, 620., 0.77967E+01, 710., 0.79466E+01
800., 0.81467E+01, 890., 0.76967E+01,END
REM SPECIFIC HEAT BTU/(LB.F) FOR ALUMINUM 2219
2219
-442., 0.10356E+00, -424., 0.10362E+00, -406., 0.10322E+00
-388., 0.10318E+00, -370., 0.10312E+00, -352., 0.10308E+00
-334., 0.10305E+00, -316., 0.10301E+00, -298., 0.10298E+00
-280., 0.10296E+00, -190., 0.10278E+00, -100., 0.10260E+00
-10., 0.10224E+00, 80., 0.10188E+00, 170., 0.10152E+00
260., 0.10116E+00, 350., 0.10079E+00, 440., 0.10043E+00
530., 0.10007E+00, 620., 0.99718E+00, 710., 0.99169E+00

END
<table>
<thead>
<tr>
<th>REM DENSITY</th>
<th>LB/(CUBIC INCH) FOR STAINLESS 347</th>
</tr>
</thead>
<tbody>
<tr>
<td>320</td>
<td></td>
</tr>
<tr>
<td>420</td>
<td></td>
</tr>
<tr>
<td>388.</td>
<td></td>
</tr>
<tr>
<td>316.</td>
<td></td>
</tr>
<tr>
<td>310.</td>
<td></td>
</tr>
<tr>
<td>300.</td>
<td></td>
</tr>
<tr>
<td>440.</td>
<td></td>
</tr>
<tr>
<td>980.</td>
<td></td>
</tr>
<tr>
<td>1520.</td>
<td></td>
</tr>
<tr>
<td>1860.</td>
<td></td>
</tr>
<tr>
<td>2060.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REM CP * RHO FOR ALUMINUM 2219</th>
</tr>
</thead>
<tbody>
<tr>
<td>1204</td>
</tr>
<tr>
<td>-424. 0.3698E-04, -424. 0.20516E-03, -406. 0.76488E-03</td>
</tr>
<tr>
<td>-388. 0.18675E-02, -370. 0.34031E-02, -352. 0.52881E-02</td>
</tr>
<tr>
<td>-334. 0.7131E-02, -316. 0.86215E-02, -298. 0.10236E-01</td>
</tr>
<tr>
<td>-280. 0.11532E-01, -190. 0.16445E-01, -100. 0.18776E-01</td>
</tr>
<tr>
<td>-100. 0.20468E-01, -300. 0.21192E-01, -170. 0.21319E-01</td>
</tr>
<tr>
<td>260. 0.21951E-01, 350. 0.22175E-01, 440. 0.22899E-01</td>
</tr>
<tr>
<td>530. 0.23471E-01, 620. 0.23731E-01, 710. 0.23800E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REM CONDUCTIVITY BTU/(INCH.HR.F) FOR STAINLESS 347</th>
</tr>
</thead>
<tbody>
<tr>
<td>6202</td>
</tr>
<tr>
<td>-424. 0.3998E-01, -424. 0.10995E+00, -406. 0.15939E+00</td>
</tr>
<tr>
<td>-388. 0.22490E+00, -370. 0.26989E+00, -352. 0.27613E+01</td>
</tr>
<tr>
<td>-316. 0.41232E+00, -280. 0.45481E+00, -190. 0.55526E+00</td>
</tr>
<tr>
<td>-100. 0.62973E+00, 80. 0.73969E+00, 260. 0.82465E+00</td>
</tr>
<tr>
<td>440. 0.89466E+00, 620. 0.95595E+00, 800. 0.10246E+01</td>
</tr>
<tr>
<td>980. 0.10945E+01, 1160. 0.11595E+01, 1340. 0.12295E+01</td>
</tr>
<tr>
<td>1520. 0.12945E+01, 1700. 0.13644E+01, 1880. 0.14294E+01</td>
</tr>
<tr>
<td>2060. 0.14994E+01, 2240. 0.15643E+01, END</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REM SP VISCOSITY LB/(INCH.HR) FOR HYDROGEN AT P = 49.0 PSIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>4101</td>
</tr>
<tr>
<td>-430. 0.37236E-02, -428. 0.34263E-02, -428. 0.33369E-02</td>
</tr>
<tr>
<td>-427. 0.37101E-02, -426. 0.30174E-02, -425. 0.28775E-02</td>
</tr>
<tr>
<td>-424. 0.27481E-02, -423. 0.26282E-02, -422. 0.25165E-02</td>
</tr>
</tbody>
</table>
-421., 0.2412E-02, -420., 0.2314E-02, -419., 0.2221E-02
-418., 0.2134E-02, -417., 0.2051E-02, -416., 0.1972E-02

REM ENTHALPHY  BTU/(lb. F) FOR HYDROGEN  AT P= 49.0 PSIA
5101
-430., -0.1222E+03, -428., -0.1195E+03, -426., -0.1180E+03
-425., -0.1165E+03, -423., -0.1107E+03, -422., -0.1056E+03
-421., -0.1031E+03, -420., -0.1005E+03, -419., -0.0970E+02
-418., -0.9510E+02, -417., -0.9220E+02, -416., -0.8920E+02

REM CONDUCTIVITY BTU/(INCH. HR. F) FOR HYDROGEN  AT P= 49.0 PSIA
6101
-430., 0.4414E+02, -428., 0.4540E+02, -426., 0.4574E+02
-425., 0.4611E+02, -423., 0.4739E-02, -422., 0.4831E+02
-421., 0.4848E-02, -420., 0.4866E-02, -419., 0.4869E-02
-418., 0.4854E-02, -417., 0.4835E-02

REM CP * RHO FOR HYDROGEN AT P= 49.0 PSIA
1101
-408., 0.0250E+04, -407., 0.2615E+04, -406., 0.2655E+04
-405., 0.2736E+04, -404., 0.2821E+04, -403., 0.2908E+04
-402., 0.2994E-04, -401., 0.3083E-04, -400., 0.3172E+04
-401., 0.3263E-04, -400., 0.3356E-04, -399., 0.3454E+04
-400., 0.3553E-04, -399., 0.3668E-04, -398., 0.3789E+04

REM SPECIFIC HEAT BTU/(INCH. HR. F) FOR HYDROGEN AT P= 49.0 PSIA
2301
-408., 0.9220E+03, -407., 0.9350E+03, -406., 0.9006E+03
-405., 0.8895E+03, -404., 0.8796E+03, -403., 0.8706E+03
-402., 0.8625E+03, -401., 0.8553E+03, -400., 0.8490E+03
-401., 0.8430E+03, -399., 0.8376E+03, -398., 0.7789E+03
-399., 0.8263E+03, -398., 0.8241E+03, -397., 0.8202E+03

REM DENSITY BTU/LB FOR HYDROGEN AT P= 49.0 PSIA
3301
-408., 0.2050E+00, -407., 0.1990E+00, -406., 0.1940E+00
-405., 0.1890E+00, -404., 0.1840E+00, -403., 0.1800E+00
-402., 0.1760E+00, -401., 0.1720E+00, -400., 0.1680E+00
-399., 0.1650E+00, -398., 0.1620E+00, -397., 0.1580E+00
-398., 0.1550E+00, -397., 0.1520E+00, -396., 0.1500E+00

REM VISCOSITY Lb/(INCH. HR) FOR HYDROGEN AT P= 49.0 PSIA
4301
-408., 0.3593E+03, -407., 0.3722E+03, -406., 0.3870E+03
-405., 0.4047E+03, -404., 0.4268E+03, -403., 0.4570E+03
-402., 0.5036E+03, -401., 0.5928E+03, -400., 0.7233E+03
-399., 0.7211E+03, -398., 0.7309E+03, -397., 0.7348E+03
-398., 0.7187E+03, -397., 0.7426E+03, -396., 0.7465E+03

REM ENTHALPHY BTU/(LB. F) FOR HYDROGEN AT P= 49.0 PSIA
5301
-408., 0.1107E+03, -407., 0.1138E+03, -406., 0.1168E+03
-405., 0.1198E+03, -404., 0.1227E+03, -403., 0.1256E+03
-402., 0.1285E+03, -401., 0.1314E+03, -400., 0.1342E+03
-399., 0.1370E+03, -398., 0.1398E+03, -397., 0.1426E+03
-398., 0.1454E+03, -397., 0.1481E+03, -396., 0.1509E+03

REM CONDUCTIVITY BTU/(INCH. HR. F) FOR HYDROGEN AT P= 49.0 PSIA
6301
-408., 0.1347E-02, -407., 0.1405E-02, -406., 0.1471E-02
-405., 0.1540E-02, -404., 0.1627E-02, -403., 0.1734E-02
-402., 0.1882E-02, -401., 0.2153E-02, -400., 0.2592E-02
-399., 0.2618E-02, -398., 0.2643E-02, -397., 0.2669E-02
-398., 0.2695E-02, -397., 0.2720E-02, -396., 0.2745E-02

REM CP * RHO FOR HYDROGEN AT P= 49.0 PSIA
1301
-408., 0.1929E+03, -407., 0.1817E+03, -406., 0.1747E+03
-405., 0.1681E+03, -404., 0.1618E+03, -403., 0.1567E+03
-402., 0.1518E+03, -401., 0.1471E+03, -400., 0.1426E+03
-399., 0.1391E+03, -398., 0.1356E+03, -397., 0.1315E+03
-398., 0.1283E+03, -397., 0.1252E+03, -396., 0.1230E+03

END
M NTHETA= K1
M NBETAS= K2
M BETA = XX3
M RIN = XX4
M TVOL = XX5
M NTUNIT= K10
M TIME0 = XX101
M TIMEND= XX102
M OUTPUT= XX104
M DTIME1= XX103
FWDBCK
END
BCD 3VARIABLES 1
F COMMON/USER1/ NTHETA, NBETAS, NTUNIT, BETA, RIN, TVOL
END
BCD 3VARIABLES 2
END
BCD 3OUTPUT CALLS
TPRNTF
END
BCD 3END OF DATA
EOF
coasinda model
ja -clif # GET ACCOUNTING INFO
APPENDIX C

"CryoTran Model" Files Part 2

Spherical Models with no Nodes in Regions 4 & 5
Sample sphere models where regions 4 and 5 are not nodalized.

```plaintext
# USER-userid   PWpassword
# QSUB -r sphere2   # jobname
# QSUB -eo       # Combine error and standard output
# QSUB -lT 59   # CPU time
# QSUB-1M 1.5Mw   # Memory requested
# rem         # End NQS statements
set -x       # set echo
ja

cat > model << EOF   # SINDA MODEL TO FOLLOW

BCD THERMAL LPCS
C
REM THIS SINDA MODEL WAS GENERATED BY CRYOTRAN
C
REM SPHERE --- 2D WEDGE SHELL - THICK WALL FILL ANALYSIS
C
REM WEDGE ANGLE-BETA = 1.0 RADIANS
BCD 9SAMPLE RUN OF NO NODES IN TANK, CALLING
BCD 9SUBROUTINES
END

BCD NODE DATA
REM NODE TEMPERATURES ARE IN (DEG R)
REM DIMENSIONS ARE IN (IN.), TIME IS IN (SECS)
REM SURFACE NODES, INSIDE TANK WALL
GEN 1001, 25, 1, 540.0, -1.000000 $ SURFACE NODES
REM DIFFUSION NODES, REGION 1, TANKWALL
REM REGION 1, LAYER NO. 1
SIM 2001, 2, 24, 540.0, A1201, 2.610795 $ STAINLESS 304A
SIM 2002, 2, 22, 540.0, A1201, 7.791187 $ STAINLESS 304A
SIM 2003, 2, 20, 540.0, A1201, 12.848701 $ STAINLESS 304A
SIM 2004, 2, 18, 540.0, A1201, 17.703583 $ STAINLESS 304A
SIM 2005, 2, 16, 540.0, A1201, 22.279266 $ STAINLESS 304A
SIM 2006, 2, 14, 540.0, A1201, 26.503586 $ STAINLESS 304A
SIM 2007, 2, 12, 540.0, A1201, 30.309937 $ STAINLESS 304A
SIM 2008, 2, 10, 540.0, A1201, 33.638306 $ STAINLESS 304A
SIM 2009, 2, 8, 540.0, A1201, 36.436127 $ STAINLESS 304A
SIM 2010, 2, 6, 540.0, A1201, 38.659363 $ STAINLESS 304A
SIM 2011, 2, 4, 540.0, A1201, 40.272919 $ STAINLESS 304A
SIM 2012, 2, 2, 540.0, A1201, 41.251343 $ STAINLESS 304A
SIM 2013, 540.0, A1201, 41.579208 $ STAINLESS 304A
REM REGION 1, LAYER NO. 2
SIM 2026, 2, 24, 540.0, A1201, 2.510389 $ STAINLESS 304A
SIM 2027, 2, 22, 540.0, A1201, 7.491554 $ STAINLESS 304A
SIM 2028, 2, 20, 540.0, A1201, 12.354561 $ STAINLESS 304A
SIM 2029, 2, 18, 540.0, A1201, 17.022736 $ STAINLESS 304A
SIM 2030, 2, 16, 540.0, A1201, 21.422440 $ STAINLESS 304A
SIM 2031, 2, 14, 540.0, A1201, 25.484314 $ STAINLESS 304A
SIM 2032, 2, 12, 540.0, A1201, 29.344287 $ STAINLESS 304A
SIM 2033, 2, 10, 540.0, A1201, 32.344635 $ STAINLESS 304A
SIM 2034, 2, 8, 540.0, A1201, 35.034866 $ STAINLESS 304A
SIM 2035, 2, 6, 540.0, A1201, 37.172607 $ STAINLESS 304A
SIM 2036, 2, 4, 540.0, A1201, 38.724106 $ STAINLESS 304A
SIM 2037, 2, 2, 540.0, A1201, 39.649902 $ STAINLESS 304A
SIM 2038, 540.0, A1201, 39.980148 $ STAINLESS 304A
REM REGION 1, LAYER NO. 3
SIM 2051, 2, 24, 540.0, A1201, 2.411952 $ STAINLESS 304A
SIM 2052, 2, 22, 540.0, A1201, 7.197796 $ STAINLESS 304A
SIM 2053, 2, 20, 540.0, A1201, 11.870125 $ STAINLESS 304A
SIM 2054, 2, 18, 540.0, A1201, 16.355255 $ STAINLESS 304A
SIM 2055, 2, 16, 540.0, A1201, 20.382428 $ STAINLESS 304A
SIM 2056, 2, 14, 540.0, A1201, 24.485031 $ STAINLESS 304A
SIM 2057, 2, 12, 540.0, A1201, 28.001480 $ STAINLESS 304A
SIM 2058, 2, 10, 540.0, A1201, 31.076340 $ STAINLESS 304A
SIM 2059, 2, 8, 540.0, A1201, 33.661987 $ STAINLESS 304A
SIM 2060, 2, 6, 540.0, A1201, 35.714996 $ STAINLESS 304A
SIM 2061, 2, 4, 540.0, A1201, 37.205658 $ STAINLESS 304A
SIM 2062, 2, 2, 540.0, A1201, 38.109558 $ STAINLESS 304A
SIV 2063, 540.0, A1201, 38.412460 $ STAINLESS 304A
REM REGION 1, LAYER NO. 4
```
REM SURFACE NODES, OUTSIDE SURFACE, REGION 1, TANKWALL
GEN 1001, 25, 1, 540.0, 0.0, 0.00000 $ INSIDE TANK 1
END

BCD 3CONDUCTOR DATA
REM RADIAL CONDUCTORS, CONDUCTION
REM RADIAL CONDUCTORS REGION 1, LAYER 1 TO BOUNDARY 1- 4
SIM 2076, 2, 24, 540.0, A1201, 2.315484 $ STAINLESS 304A
SIM 2077, 2, 22, 540.0, A1201, 6.909912 $ STAINLESS 304A
SIM 2082, 2, 20, 540.0, A1201, 11.395363 $ STAINLESS 304A
SIM 2079, 2, 20, 540.0, A1201, 15.701111 $ STAINLESS 304A
SIM 2085, 2, 18, 540.0, A1201, 19.759216 $ STAINLESS 304A
SIM 2081, 2, 16, 540.0, A1201, 23.505732 $ STAINLESS 304A
SIM 2072, 2, 14, 540.0, A1201, 26.889331 $ STAINLESS 304A
SIM 2078, 2, 12, 540.0, A1201, 29.833420 $ STAINLESS 304A
SIM 2084, 2, 10, 540.0, A1201, 32.314789 $ STAINLESS 304A
SIM 2085, 2, 8, 540.0, A1201, 34.286545 $ STAINLESS 304A
SIM 2086, 2, 6, 540.0, A1201, 35.717575 $ STAINLESS 304A
SIM 2087, 2, 4, 540.0, A1201, 36.585342 $ STAINLESS 304A
SIM 2088, 2, 2, 540.0, A1201, 36.876114 $ STAINLESS 304A
REM CONSTANT VALUE BOUNDARY NODES; REGION 4, INSIDE OF TANK
GEN 1002, 25, 1, 540.0, 0.0, 1.000000 $ INSIDE TANK 1
END
DIM 216, 2, 21, 2085, 5, 2086, 5.2081, 1, 7.384322+00, 6.2081, 7.692526+00
DIM 218, 2, 21, 2086, 3, 2087, 3, 6.2081, 7.692526+00, 6.2081, 7.879434+00
DIM 220, 2, 21, 2087, 1, 2088, 1, 6.2081, 7.879414+00, 6.2081, 7.942040+00
REM CONVECTION CONDUCTORS; INSIDE TANK TO TANK WALL
GEN 18001, 25, 1, 18001, 1, 100, 100.000000E+00, 0.000000E+00, 1.00, 1.00
END
REM 3 CONSTANTS DATA
REM NTHETA NBETAS BETA RIN TVOL
K=10-SINDA
NBETAS I,2=1,18001,
BCD 3 ARRAY DATA
REM TIMEO(MIN) TIMEND(MIN) DTIMEI(MIN) OUTPUT(MIN)
101=1.000000E+00, 102=6.0000E+00, 103=0.20833E+03, 104=0.41667E+02
NLOOK=300, DALXCA=0.001000, ARLXCA=0.001000
END
REM 3 REGION 1, (TANKWALL)
",.INSIDE SURFACE AREAS (IN**2)
4.53598E+00, 1.35553E+01, 2.23232E+01, 3.07581E+01, 3.87078E+01
4.60471E+01, 5.26603E+01, 5.84429E+01, 6.30039E+01, 6.71665E+01
6.99969E+01, 7.16698E+01, 7.22394E+01, 7.16998E+01, 6.99969E+01
6.71665E+01, 6.30039E+01, 5.84429E+01, 5.26603E+01, 4.60471E+01
3.87078E+01, 3.07581E+01, 2.23232E+01, 1.35536E+01, 4.53598E+00
END
REM 2 REGION 1, (TANKWALL)
",.OUTSIDE SURFACE AREAS (IN**2)
5.32347E+00, 1.58664E+01, 2.61988E+01, 3.60980E+01, 4.54279E+01
5.40441E+01, 6.18027E+01, 6.85893E+01, 7.42941E+01, 7.88274E+01
8.21174E+01, 8.41125E+01, 8.47810E+01, 8.41125E+01, 8.21174E+01
7.88274E+01, 7.42941E+01, 6.85893E+01, 6.18027E+01, 5.40441E+01
4.54279E+01, 3.60980E+01, 2.61988E+01, 1.58664E+01, 5.32347E+00
END
REM CONDUCTIVITY BTU/(INCH.HR.F) FOR STAINLESS 304A
6201
36., 0.10346E+00, 72., 0.21616E+00, 108., 0.35610E+00
144., 0.42607E+00, 180., 0.47480E+00, 270., 0.57476E+00
360., 0.64972E+00, 450., 0.79790E+00, 540., 0.75968E+00
630., 0.80866E+00, 720., 0.84964E+00, 810., 0.88462E+00
900., 0.91691E+00, 990., 0.94960E+00, 1080., 0.98998E+00
1170., 0.10246E+01, 1260., 0.10596E+01, 1350., 0.10895E+01
1440., 0.11245E+01, 1530., 0.11595E+01, 1620., 0.11945E+01
1800., 0.12845E+01, 1980., 0.13344E+01, 2160., 0.14044E+01
2340., 0.14744E+01, 2520., 0.15443E+01, 2700., 0.16143E+01, END
REM SPECIFIC HEAT BTU/(LB.F) FOR STAINLESS 304A
2201
36., 0.10200E-02, 72., 0.19600E-02, 108., 0.25800E-01
144., 0.45500E-01, 180., 0.58500E-01, 270., 0.81000E+00
360., 0.93000E-01, 450., 0.10000E+00, 540., 0.10800E+00
630., 0.11200E+00, 720., 0.11700E+00, 810., 0.12150E+00
900., 0.12600E+00, 990., 0.12950E+00, 1080., 0.13300E+00
1170., 0.13500E+00, 1260., 0.13800E+00, 1350., 0.14000E+00
1440., 0.14200E+00, 1530., 0.14500E+00, 1620., 0.14800E+00
1800., 0.15000E+00, 1980., 0.15200E+00, 2160., 0.15400E+00
2340., 0.15700E+00, 2520., 0.16000E+00, 2700., 0.16200E+00, END
REM DENSITY LB/(CUBIC INCH) FOR STAINLESS 304A
3201
36., 0.28473E+00, 72., 0.28858E+00, 108., 0.28837E+00
144., 0.28808E+00, 180., 0.28782E+00, 270., 0.28725E+00
360., 0.28663E+00, 450., 0.28600E+00, 540., 0.28537E+00
630., 0.28468E+00, 720., 0.28396E+00, 810., 0.28324E+00
900., 0.28255E+00, 990., 0.28179E+00, 1080., 0.28107E+00
1170., 0.28032E+00, 1260., 0.27951E+00, 1350., 0.27883E+00
1440., 0.27784E+00, 1530., 0.27709E+00, 1620., 0.27655E+00
1800., 0.27478E+00, 1980., 0.27294E+00, 2160., 0.27102E+00
2340., 0.26960E+00, 2520., 0.26687E+00, 2700., 0.26463E+00, END
REM CP + RHO FOR STAINLESS 304A
1201
36., 0.29450E-03, 72., 0.20085E-02, 108., 0.74398E-02
144., 0.12819E-01, 180., 0.16083E-01, 270., 0.23267E-01

102
| Temp (°F) | Density (lb/Lbf) | Conductivity (BTU/(in/hr/ft)) | Viscosity (lb/(in/hr)) | Specific Heat (BTU/(in/hr/°F)) | Enthalpy (BTU/ft³) | CP (RHO)
|-----------|------------------|-------------------------------|------------------------|-------------------------------|-------------------|------
| 0         | 0.3370E+03      | 42, 0.4190E+03                | 43, 0.4140E+03         | 44, 0.4085E+03                | END               |
| 30        | 0.3370E+03      | 42, 0.4190E+03                | 43, 0.4140E+03         | 44, 0.4085E+03                | END               |
| 33        | 0.3370E+03      | 42, 0.4190E+03                | 43, 0.4140E+03         | 44, 0.4085E+03                | END               |
| 36        | 0.3370E+03      | 42, 0.4190E+03                | 43, 0.4140E+03         | 44, 0.4085E+03                | END               |
| 39        | 0.3370E+03      | 42, 0.4190E+03                | 43, 0.4140E+03         | 44, 0.4085E+03                | END               |
| 42        | 0.3370E+03      | 42, 0.4190E+03                | 43, 0.4140E+03         | 44, 0.4085E+03                | END               |
| 45        | 0.3370E+03      | 42, 0.4190E+03                | 43, 0.4140E+03         | 44, 0.4085E+03                | END               |
| 50        | 0.3370E+03      | 42, 0.4190E+03                | 43, 0.4140E+03         | 44, 0.4085E+03                | END               |
| 55        | 0.3370E+03      | 42, 0.4190E+03                | 43, 0.4140E+03         | 44, 0.4085E+03                | END               |
| 60        | 0.3370E+03      | 42, 0.4190E+03                | 43, 0.4140E+03         | 44, 0.4085E+03                | END               |

For hydrogen at P = 49.0 psia.
REM ENTHALPHY BTU/(LB.F) FOR HYDROGEN AT P= 49.0 PSIA
5301
52., 0.11078E-03, 53., 0.11380E+03, 54., 0.11680E+03
55., 0.11986E+03, 56., 0.12270E+03, 57., 0.12560E+03
58., 0.12850E+03, 59., 0.13140E+03, 60., 0.13420E+03
61., 0.13700E+03, 62., 0.13980E+03, 63., 0.14260E+03
64., 0.14540E+03, 65., 0.14810E+03, 66., 0.15090E+03, END
REM CONDUCTIVITY BTU/(INCH.HR.F) FOR HYDROGEN AT P= 49.0 PSIA
6301
52., 0.13473E-02, 53., 0.13473E-02, 54., 0.14710E-02
55., 0.15430E-02, 56., 0.16279E-02, 57., 0.17342E-02
58., 0.18823E-02, 59., 0.21532E-02, 60., 0.25921E-02
61., 0.26439E-02, 62., 0.26439E-02, 63., 0.26695E-02
64., 0.26951E-02, 65., 0.27205E-02, 66., 0.27457E-02, END
REM CP * RHO FOR HYDROGEN AT P= 49.0 PSIA
1301
53., 0.19028E+03, 54., 0.19028E+03, 55., 0.19028E+03
56., 0.19028E+03, 57., 0.19028E+03, 58., 0.19028E+03
59., 0.19028E+03, 60., 0.19028E+03, 61., 0.19028E+03
62., 0.19028E+03, 63., 0.19028E+03, 64., 0.19028E+03
65., 0.19028E+03, 66., 0.19028E+03, END

END

BCD 3EXECUTION
F COMMON/USER1/ NTHETA,NBETAS,NTUNIT,BETA,RIN,TWOL
F COMMON/USER2/ PTIME, DELTIM, XC1, XC2, XC3, XC4
F COMMON/INSA /SARIN { 25)
F COMMON/OUTSA/SAROUT ( 25)
F COMMON/SURFT/TSURF { 25)
F COMMON/BNDYT/TBDY ( 25)
F COMMON/HTRCO/HCOEF ( 25)
F COMMON/SURFQ/QSURF ( 25)
F DIMENSION X( 800)
F NDIM= 800
M NTHETA= K1
M NBETAS= K2
M BETA = X3
M RIN = X4
M TVOL = X5
M XC1 = X6
M XC2 = X7
M XC3 = X8
M XC4 = X9
M NTUNIT = K10
F DO 120 I=1,NTHETA
M SARIN(I) =A (1+I)
M SAROUT(I) =A (2+I)
F 120 CONTINUE
F CALL THWSE1
M TIMEO = XK101
M TIMEND= XK102
M OUTPUT= XK104
M DTIME2= XK103
F CONTINUE
F CALL THWSE2
END

BCD 3VARIABLES 1
F COMMON/USER1/ NTHETA,NBETAS,NTUNIT,BETA,RIN,TWOL
F COMMON/USER2/ PTIME, DELTIM, XC1, XC2, XC3, XC4
F COMMON/INSA /SARIN { 25)
F COMMON/OUTSA/SAROUT ( 25)
F COMMON/SURFT/TSURF { 25)
F COMMON/BNDYT/TBDY ( 25)
F COMMON/HTRCO/HCOEF ( 25)
F COMMON/SURFQ/QSURF ( 25)
F PTIME -TIMEO
F DELTIM =DTIME2
F DO 270 I=1,NTHETA
M I1=I-1
M TSURF(I)=T(1001+I1)
F 270 CONTINUE
CALL THWSV1
DO 271 I=1,NTHETA
   CALL THWSV2
   IM(I)=I-NTHETA
   M T(I)=IM(I)-T(I)
   M Q(I)=Q(I)+QSURF(I)
   M G(I)=G(I)+HCEF(I)*SARIN(I)
271 CONTINUE END

BCD 3VARIABLES 2
CALL THWSV2
END BCD

BCD 3OUTPUT CALLS
CALL THWSOU
END BCD

CC* SUBROUTINES CALLED BY SINDA CRYOTRAN PROGRAMS NTYP = 1, MAN = 3.
CC* THESE SUBROUTINES ARE CALLED FROM THE EXECUTION AND VARIABLES BLOCKS
CC* OF THE THICK WALL FILL ANALYSIS OF A SPHERE. THIS IS TAKEN FROM THE
CC* THE UNITS USED IN THE ORIGINAL PROGRAM WERE:
CC* DEG, IN., MIN., LBS., BTU
CC* CRYOTRAN USES DEGF OR DEGR, IN., HR., LBS., BTU
CC* THE FOLLOWING LISTED COMMON BLOCKS ARE DEFINED BY CRYOTRAN TO
CC* COMMUNICATE BETWEEN CRYOTRAN AND THESE SUBS.
CC* BLOCKS USER1, USER2, SURFA, SURFT ARE INPUT TO THIS PGM.
CC* BLOCKS BNDYT, HTRCF, SURFG, SURFQ ARE OUTPUT TO SINDA
CC* IF NTUNIT=1--> SINDA TEMP= DEGF; IF NTUNIT=2--> SINDA TEMPS=DEGR
CC* SUBROUTINE THWSV:
F COMMON /USER1/ NTUNIT, NTHETA, NBF_TAS, NTUNIT, BETA, R[N, TVOL
F COMMON /USER2/ TME0, DTIMEU, FLOW, TLIQ, TGAS, PCTFIL
F COMMON /NDTY/ TDY(I)
F COMMON /QOA/ A100(26), A101(26)
F COMMON /PLTSAV/ NOUT, NCOND(200), VOLCM(200), TOTVIN(200)
F COMMON /CONSTS/ PI, FORPI, TWOPIR, TWOR, CONI, TBTETA, THTARC, ARCO2, NTVI
F COMMON /OUTVAR/ TOTW, TOTVOL, ARCOLD, ARCNEW, HOLD, NCOLD, SRFOLD, VTOTIN
F COMMON /LIQST/ NC, BAKING, TKVTST, FULL, DELTMP(100)
F COMMON /HFCLC/ HFSUM, SUMH, VNEWR, DENS, RHOH2, HVA
F COMMON /DIAMS/ DIAM(100)
F COMMON /BG3S/ BGL3(100)
F COMMON /ARCUM/ CUMARC(100)
F COMMON /LATT/ PHICLAC(100)
F COMMON /DEBUG/ DEBUG
F COMMON /FINOUT/ TTST, ARCWET, ARCHLB, ARCHUB
EQUIVALENCE(IAI00, A100(1)), (IAI01, A101(1))
D DIMENSION FBDT(25), QOA(25)
L LOGICAL BAKING, FULL
LOGICAL DEBUG
C C
C DEUBG=.FALSE.
C
TAS, TLIQ (DEGF), FLOW(LB/SEC) * 3600= (LB/HR)
D DATA FBDT /0., 94.1, 26.1, 44.1, 61.8, 2.0, 80.3, 3.8, 4.9, 4.68,
F 1 5.0, 5.399, 5.4, 7.2, 10.8, 14.4, 16.0,
F 2 36.0, 72.0, 108.1, 360.1, 720.1, 1080.1
C Q/A (FT2/ST2-HR)
D DATA QOA / 0.172/0.1.172E2, 6.92E2, 9.52E2, 1.76E3, 5.08E3, 9.83E3,
F 1 1.28E4, 1.62E4, 2.00E4, 2.35E4, 2.95E4,
F 2 6.03E3, 7.14E3, 9.2E3, 1.05E3, 1.22E3,
F 3 2.00E3, 3.81E3, 3.74E3,
F 4 7.61E1, 9.52E3, 1.98E4, 4.60E4, 8.09E4
C
C DATA GVAN/32.2/, #I/3., 4159265/
F IA100=25
F IA101=25

105
C INITIALIZE PROPERTIES FOR SS AND LH2
F
DENS=0.29
F
RHOLH2=0.0024722
F
HVAP=186.5
C
F
X9=60.
F
XE=2.0
F
XKII=TLIQ*XX10
F
NTH=0
F
NOUT=0
F
VTST=0.0
F
BAKING=.FALSE.
F
TOTWT=0.0.
F
TOTVOL=0.
F
VTSTIN=0.0.
F
SRFOLD=0.
F
ARCOLD=0.
F
HOLD=0.
F
NCOIL=0
F
NTVI=0
F
NRCOLD=0
F
FLOW=FLOW/60.
C
TNKVOL(IN3)=TVOL(FT3)*17,28
F
TNKVOL=TVOL*1728.
F
TBETA=1./60.
F
FORP=4.*PI
F
TMU2 =9.2*RIN
F
TMUPF=TMU*PI
F
CON1 =3./2.///PI/RIN**3
CC
PUT DELTEMP (DEG) AND Q/A (BTU/FT2-HR) INTO ARRAYS 100,101
F
DO 1 I=1, 25
F
A100(I+1)=FCTFIL*TNKVOL
F
FULL=.FALSE.
F
ANG=NTHETA
F
THETA=PI/ANG
F
THTARC=RIN*THETA
F
ARCO2=THTARC/2.
F
PHI=1.
F
NT=NTTHETA
F
NTP1-NTTHETA+1
CC
COMPUTE DIAMETER (FT), THEN COMPUTE BETAG* (L**3) AT EACH STATION
CC
COMPUTE INSIDE ARC LENGTHS OF SPHERE
F
TWOIN=2.*RIN
F
NNN=(NT-1)/2
F
PRINT 1002, TVOL, FLOW, XK9
F102 FORMAT (' CT. FT. TANK WITH LIQ FLOW=',F10.5,
F
1 ' (LB/MIN),  BETA=1/' ,F7.2)
F
PRINT 1003, TNKVOL, TKVTST
F1003 FORMAT(' TANK VOL(IN**3), VOLTEST',IP2GI4.7)
F
IF (DEBUG) PRINT 1001
F1001 FORMAT('//10X,'9X,'II',7X,'ANGLE1',5X,'ANGLE2',5X,'SIN(ANGLE)',
F
1 2X,'RADIUS',6X,'PHIARC',5X,'DIAM',3X,'SURF AREA')
F
DO 69 I=1, NNN
F
ANGLE=I*THETA
F
II+1=NTHETA
F
SINANG=SIN(ANGLE)
F
RAD=RIN*SINANG
F
PHIARC(I)=RAD*PHI
F
PHIARC(I)=PHIARC(I)
F
DIAM(I)=RAD*2
F
DIAM(I)=DIAM(I)
F
ANG2=PI-ANGLE
F
IF (DEBUG) PRINT 1000, I, II, ANGLE, ANG2, SINANG, RAD, PHIARC(I),
F
1 DIAM(I), SAREA(I)
F1000 FORMAT ('EXECN1'//1X,2112,8G12.5)
F
69 CONTINUE

106
F TPHI=THEIA*PHI
F DO 68 I=1,NT
F EL3=(DIAM(I)/12.)**3
F BGL3(I)=GRAV*TBETA*EL3
F TBOY(I)= TGA3
F IF (NTUNIT .EQ. 1) TBOY(I)=TBOY(I)-460.
F 68 CONTINUE
F CUMARC (I)=ARCO2
F DO 74 I=2,NT
F CUMARC (I)=CUMARC (I-1)+THTARC
F 74 CONTINUE
F CUMARC (NT)=CUMARC (NT)-ARCO2
F
C COMPUTE LOWER AND UPPER BOUNDS FOR FINE OUTPUT AT EQUATOR
F ARCHAF=RIN*PI/2.
F ARCHLB=ARCHAF-THTARC
F ARCHUB=ARCHAF+THTARC
F NTO2=NTHETA/2
F TTEST=TBOY (NTO2)
F IF (DEBUG) PRINT 2000,
F 1 1 (I,NA(I),NB(I),ELA(I),ELB(I),FAREA(I),I=1,NCOND)
F2000 FORMAT ('EXECN3',318,3G13.5)
F RETURN
F END
F SUBROUTINE TMSWE2
F COMMON /USER1/ NTHETA, NBETAS, NTUNIT, BETA, RIN, TNKVOL
F COMMON /FLTSAV/ NOUT, NNCOV (200), VOLCUM (200), TOTVIN (200)
C F WRITE (23,2001) NOUT, (NNCOV(I),I=1,NOUT)
F WRITE (23,2002) NOUT, (VOLCUM(I),I=1,NOUT)
F DO 101 I=1,NOUT
F 101 VOLCUM(I)=VOLCUM(I)/TNKVOL
F WRITE (23,2002) NOUT, (VOLCUM(I),I=1,NOUT)
F WRITE (23,2002) NOUT, (TOTVIN(I),I=1,NOUT)
F2001 FORMAT (16(/(2016)})
F2002 FORMAT (16(/(1P10E12.5))
F RETURN
F END
F SUBROUTINE TMSVSI
C COMMON BLOCKS TO COMMUNICATE WITH SINDA
F COMMON /USER1/ NTHETA, NBETAS, NTUNIT, BETA, RIN, TNKVOL
F COMMON /USER2/ TIME0, DTIMEU, FFLOW, TLIQ, TGAS
F COMMON /SURFA/ SAREA (1)
F COMMON /SURFT/ TSURF(1)
F COMMON /SWDTT/ TBOT(1)
F COMMON /HTRCF/ NCOF(1)
F COMMON /SURFG/ GSURF(1)
F COMMON /SURFO/ QSURF(1)
C F COMMON /CGDATA/ NDN, NAM, NFB, NIG, NISG, NSBG, NINTGS
F COMMON /CONSIS/ PI, FORPI, TMOPIR, TQOR, CON1, TBETA, THTARC, ARCO2, NTV1,
  F 1 XXL1
F COMMON /FINOUT/ TTEST, FOWRI, FARCHW, FCARCH, FCHUB
F COMMON /OUTVAR/ TOTWT, TOTVL, ARCW, ARCW, HCOF, NCOF, SFACTOR, VOTIN
F COMMON /NEWOLD/ NCO, NW, TOTVM, SFORW, ARCONW, DELVOL
F COMMON /HCLC/ HSG, SUM, SUM, VENW, DENS, RHOLH2, HVAP
F COMMON /ROQA/ QOA (100)
F COMMON /TQOA/ A100 (26), A101 (26)
F COMMON /DIAMS/ DIAM(1)
F COMMON /BGL3S/ BGL3 (1)
F COMMON /AKCUM/ CUMARC (1)
F COMMON /LATTD/ SHARC (1)
F COMMON /LIQST/NC, BAKING, TKVTST, FULL, DELTMP (100)
F COMMON /DBG/ DEBUG
C F LOGICAL BAKING,FULL
F LOGICAL DEBUG
CC
CC DIMENSION TTBL(27), SH(27), AKT(27)
F DIMENSION TTBL(27)
CC Dimensionalization table for stainless steel
CC Density, Density of H2, VAP (BTU/LB)
C DENS, DENS H2, VAP/ 0.29, 0.024722, 186.5/ These variables
C are initialized in SUB DWEXCI
CC TEMPS FOR PROPERTIES DEGR
F DATA TTBL / 20., 40., 60., 80., 100., 120., 140., 160., 180., 200.,
F 2 460., 480., 500., 520., 540.
CC CP STAINLESS BTU/LB
CC DATA SH / .001, .002, .005, .012, .021, .032, .041, .049, .057, .064,
CC 1 .071, .077, .082, .086, .090, .099, .102, .103, .106,
CC 2 .1075, .109, .110, .113, .114 /
CC THERMAL COND STAINLESS BTU/HR-FT-DEGR
CC DATA AKT /
CC 0.80, 1.70, 2.45, 3.15, 3.78, 4.35, 4.85, 5.26, 5.62,
CC 1 5.98, 6.31, 6.63, 6.92, 7.20, 7.44, 7.71, 7.93, 8.16, 8.37, 8.58,
CC 2 8.78, 8.97, 9.16, 9.33, 9.52, 9.69, 9.87 /
CC
CC PROPERTY TABLES FOR O-H2, USE TEMP TABLE IN VBLI, TTBL(I) 20-540/20
CC
CC GAS H2 CP (BTU/LB)
F DATA GPCP / 2.46, 2.47, 2.46, 2.48, 2.515, 2.60, 2.74, 2.92, 3.17,
F 1 3.44, 3.62, 3.76, 3.86, 3.88, 3.87, 3.85, 3.83, 3.80, 3.76, 3.72,
F 2 3.67, 3.64, 3.61, 3.58, 3.54 /
CC GAS H2 RH0 (LB/FT3)
F DATA GFRHO / .0762, .0762, .0832, .0956, .0823, .0236, .0202,
F 1 .0176, .0157, .0141, .0128, .0117, .0108, .0101, .0094, .0088, .0083,
F 2 .0078, .0074, .0060, .0064, .0061, .0059, .0056, .0054, .0052 /
CC GAS H2 MI (LB/FT-SEC)
F DATA GPMU / 0.70E-6, 0.70E-6, 1.0E-6, 1.43E-6, 1.70E-6, 2.00E-6,
F 1 2.20E-6, 2.54E-6, 2.75E-6, 3.00E-6, 3.22E-6, 3.44E-6, 3.62E-6,
F 2 3.83E-6, 4.00E-6, 4.22E-6, 4.39E-6, 4.56E-6, 4.74E-6, 4.90E-6,
F 3 5.00E-6, 5.24E-6, 5.39E-6, 5.53E-6, 5.70E-6, 5.87E-6, 6.02E-6 /
CC GAS H2 K (BTU/FT-HR-DEGR)
F DATA GPK / .0100, .0100, .0140, .0190, .0230, .0270, .0310, .0355,
F 1 .0395, .0435, .0475, .0515, .0555, .0595, .0630, .0670, .0705, .0750,
F 2 .0780, .0810, .0850, .0889, .0913, .0950, .0985, .1020, .1050 /
CC
CC
CC COMPUTE GS AND QS FOR BOUNDARY CONDITIONS
CC
F DELTIM = DTIM*60.
F TIMO = TIM*60.
F NTVI = NTVI+1
F VNENRE = 0.
F IF (FULL) GO TO 87
F WTIM = FLOW*DELTIM
F DEVAR = WTIM/ROH2
F VOLEW = TOTVL+DEVAR
F COSGAM = 1., CONJ1VOLNEW
F IF (ABS (COSGAM ) .GE. 1.0) COSGAM=SIGN(1., C lag)
F GMACU = ACGOS (COSGAM)
F H2 = RIN+WarosGAM (GMACU+ ORPI)/3.0)
F SRFINEM = WOPFIH
F CARG = (RIN-HT)/816
F IF (DEBUG) PRINT 9999,
F 1 NTVI, GMACU, RIN, HT, TWOPF, SRFINEM,
F 2 CARG, WTIM, DEVAR, VOLEW, COSGAM
F 9999 FORMAT ('VARBLI', 16(6E12.4))
F IF (ABS(CARG ) .GE. 1.0) CARG=SIGN(1., CARG)
F ARCLN = ACOS(CARG)
F ARCNEM = INN* ARCLN
F ARCWET = ARCNEM
F DELEC = ARCNEM-ARCLEO
F DELSRF = SRFINEM- SRFOLD
F

108
CC NC IS NUMBER OF NODES COMPLETELY COVERED BY LIQUID
F HNC=ARCHNM/ARCO2
F WETNEW=(HNC+1.1)/2.
F NC=WETNEW
F NNWN=WETNEW+1.
F NCHM=NC-NNWN
F SUMN=0.
F SUM=0.
CC GET AVERAGE SURFACE TEMP OF NEWLY WETTED NODES
F NCOPI=NNWN+1
F NC2=NNWN
F IF(NCOPI.GT.NC2) NCOPI=NC2
F DO 81 1=NCOPI,NC2
F SUM=SUM+TSURFR
F TAVG=SUM/SUMN
F XK104=TAVG-THLIQ
CC FILM BOILING COEF. IN XK105 (BTU/IN2-MIN)
F CALL DIDA(XK104,A100,A101,XK105)
F DTIME=BTUAVA/XK105/DELSRF
CC VOL OF NEW LIQ REMAINING AT END OF TIME STEP
F BTUTR=BTUAVA
F IF(DTIME.LT.DELTIM) GO TO 85
F BTUTRA-XK105*DELSRF*DELTIM
F BTUREM=BTUAVA-BTUTRA
F IF(BTUREM) 85,85,86
F 86 CONTINUE
F WREM=BTUREM/HVAP
F VNEWRE=WREM/RHOLH2
F 85 CONTINUE
F IF(DEBUG) PRINT 9998,
F 1 NC, NCHM, NCOPI, NC2, NNWN, FULL,
F 2 FLOW, WETIN, DELVOL, TOTVOL, VOLNEW, COSGAM,
F 3 HT, SRFRM, CARG, ARCLEN, ARCHM, DELARC,
F 4 ARC02, DELSRF, BTUAVA, HNC, WETNEW, SUM,
F 5 SUMN, TAVG, XK104, XK105, DTIME, DELTIM,
F 6 BTUTR, BTUREM, WREM, VNEWRE, TIMEO, TIMEND,
F 7 TIMEM, TIMEN, DTIMEU, TNKVOL, TKVTST, TOTVOL,
F 8 TOTWT, TVOL
F9998 FORMAT('VARBLI2',618,L6/(6EI4.5))
CC COMPUTE Q FOR WETTED NODES Q=QOA*SURFAC
CC COMPUTE Q(FILM BOILING, BTU/IN2-MIN) FOR LIQ COVERED NODES
F 87 SUMN=NCOPI
F HFSUM=0.
F IF(NC.GE.0) GO TO 92
F IF(BAKING) GO TO 94
F IM=I-1
F TSURFR=TSURF(I)
F IF(NTUNIT.EQ.1) TSURFR=TSURFR+460.
F DELTMP(I)=TSURFR-THLIQ
F 95 CONTINUE
F 94 CONTINUE
F DO 91 I=1,NC
F QOA(I)=0.
F IM=I-1
F IF(TSURFR.LT.XK11) GO TO 90
F XK104=DELTMP(I)
F CALL DIDA(XK104,A100,A101,XK105)
F QOA(I)=XK105
F IF(I.LE.NCOLD) HFSUM=HFSUM+XK105
F QSURF(I)=XK105*SAREA(I)*60.
F GSURF(I)=0.
F IF(TIMEO.GT.75. .AND. DEBUG) PRINT 9997,
F 1 I,XK105,SAREA(I),QSURF(I)
F9997 FORMAT('VARBLI3',I8,3E14.5)
GO TO 91
90 CONTINUE
QOA(I)=0.0
QSURF(I)=0.0
QSURF(I)=1.815
TBDY(I)=TLDG+0.001
IF(INTUNIT .EQ. 1) TBDY(I)=TBDY(I)-460.
91 CONTINUE
QMULT=1.
GMULT=0.
IF(DTIME .GE. DELTIM) GO TO 75
QMULT=DTIME/DELTIM
GMULT=1.-QMULT
IF(NCOLD .EQ. NC) GO TO 75
CONTINUE
QMULT=0.
GMULT=1.
CONTINUE
QMULT=DTIME/DELTIM
GMULT=1.-QMULT
IF(NCOLD .EQ. NC) GO TO 75
DO 74 I=1-NC,NC
QSURF(I)=QSURF(I)*QMULT*60.
CALL GETH(I,NFILM)
NGHI=I
GO TO 5000
GSURF(I)=HFILM*SAREA(I)*GMULT*60.
74 CONTINUE
Q AND G-HA FOR PARTIALLY COVERED NODE
CONTINUE
75 CONTINUE
NC=NO NODES COMPLETELY COVERED
92 NPC=NC+1
TSURF=TSURF(NPC)
IF(NCOLD .EQ. 1) TSURF=TSURF+460.
XM104=TSURF
CONTINUE
GET QOA FOR PARTIALLY COVERED NODE
CALL DIDIQA(XM104,A100,A101,XM105)
CONTINUE
NSUM=NSUM+1
SUMN=SUMN+I
WSAREA=(ARCNEW-CUMARC(NC))*PHIARC(NPC)
GSAREA=(CUMARC(NPC)-ARCNEW)*PHIARC(NPC)
QSURF(I)=XM105*WSAREA*60.
CONTINUE
NC=NO NODES COMPLETELY COVERED
93 NPC=NC+1
GETH(I,NFILM)
NGHI=I
GO TO 5000
CONTINUE
FS002 CONTINUE
GSURF(NC+1)=HFILM*GSAREA*60.
CONTINUE
NOW : DTME .LT. DELTIM CORRECT Q AND G
IF(DTIME .LT. DELTIM) GO TO 97
GSURF(NC+1)=GSURF(NC+1)*GMULT*60.
CONTINUE
GSURF(NC+1)=GSURF(NC+1)+HFILM*WSAREA*GMULT*60.
97 NPC=NC+1:
98 REMAINDER OF NODES ALL GAS COVERED
CONTINUE
ASSIGN 5003 TO NGATRN
CONTINUE
DO 93 I=1-NG1,NBD
CONTINUE
93 NPC=NC+1
CALL GETH(I,NFILM)
NGHI=I
GO TO 5000
FS003 CONTINUE
CONTINUE
GSURF(I)=HFILM*SAREA(I)*GMULT*60.
CONTINUE
93 CONTINUE
IF(DEBUG) PRINT 9996, NCOLD, NC, NG1, NPC,
CONTINUE
1 TIME0, TOTVOL, WTIN, DELVOL, VMER4, HFLM, HFLM,
CONTINUE
2 XM104, XM105, HFSUM, SUMN, GMULT, QMULT,
SUBROUTINE GUTH(NGHI, HFILM)

CC INTERNAL SUBROUTINE TO CALCULATE H-FILM HEAT TRANS COEF.
CC (BTU/MIN-IN2-DEGR)

TSURFR-TSURF (NGHI)

IF(NTUNIT .EQ. i) TSURFR-TSURF/460.

TWALL-TSURFR

DELT-ABS (TWALL-TGAS)

TFLMAV- (TWALL+TGAS)/2.

TFLNT-TFLMAV/20.

NT-TFLNT

NTPI-NT+1

FRACT- (TFI24AV-TTBL (NT))/20.

CP- GPCP (NT) +FRACT* (GPCP (NTPI) -GPCP (NT))

RHO-GPRHO (NT) +FRACT* (GPRHO (NTPI) -GPRHO (NT))

AKHR-GPK (NT) +FRACT* (GPK (NTPI) -GPK (NT))

AKSEC-AKRH/3600.

GRPR- BGL3 (NGHI)*DELT*RHO*RHO*CP/AMU/AKSEC

IF(NGHI .LE. 18 .AND. DEBUG) PRINT 1000,

QOA (NGHI)-HFILM

IF(NGHI .LE. 18 .AND. DEBUG) PRINT 1000,

IF(VTEST .GT. 0.0) GO TO 99

IF(.NOT. FULL) GO TO 99

RETURN

END
COMMON /FINOUT/ TTTEST, ARCWET, ARCHLB, ARCHUB
COMMON /DEBUG/ DEBUG
LOGICAL DEBUG
LOGICAL BAKING, FULL
DATA NTV2/0/
DATA NBUP/0/
CC
DELTIM=DTIMEU*60.
NTV2=NTV2+1
BAKING=.FALSE.
DO 10 I=1,NC
TSURFR=TSURF(I)
IF (NTUNIT .EQ. 1) TSURFR=TSURFR+460.
IF (TSURF .GE. TLIQ) GO TO 10
DELTMP(I)=DELTMP(I)/2.
BACKUP=0.
I
IF (XKON(I2) .GT. BACKUP
NNN=1000+I
10 CONTINUE
IF (BAKING .GT. 0.) JO TO 100
VOLOST=0.
IF (TOTVOL .LE. 0.) JO TO 98
HF AVG+HFSU M/SUMN
BOLOST-HF AVG*SRFOLD-DELTIM
WOLOST=BOLOST/HF AVG
VOLOST=WOLOST/HNOL2
VTOTIN=VTOTIN+DETVOL
IF (DEBUG) PRINT 9999, HFAVG, BOLOST, WOLOST, VOL OST, HFSUM, SUMN, V TotIN
9999 FORMAT ("JEALZ", 6E14.5)
IF (VOLOST .GT. TOTVOL) VOLOST=TOTVOL
98 TOTVOL=TOTVOL-VOLOST+VNEWRE
TOTWT=TOTWT+HNOIN2
COSGAM=1.-CON1*TOTVOL
IF (ABS(COSGAM) .LT. 1.0) THEN
GAMCU=0.0
ELSE
GAMCU=ACOS(COSGAM)
ENDIF
HLND-RIN*TWOR*COS((GAMCU*FORPI)/3.)
SRFOLD=WOPRI*HOLD
ARG=1.-HOLD/RIN
IF (ABS(ARG) .GE. 1.0) THEN
ARCLE N 0.
ELSE
ARCLE=ACOS(ARG)
ENDIF
ARCOLD=ARCCARC
ARCARC=ARCOLD/ARCC
NCOLD=ARCARC
NCOLD=(NCOLD+1)/2
IF (TOTVOL .LE. TKVTST) GO TO 90
IF (FULL) GO TO 90
PRINT 1000, TOTVOL, NC, HOLD, TOTWT
FULL=.TRUE.
CC
CALL OUTCAL
90 CONTINUE
C TEST FOR FINE OUTPUT AT EQUATOR
IF (ARC WET .LT. ARCHLB .OR. AR CWET .GT. ARCHUB) GO TO 100
C
IF (TTEST-T50049 .LT. 20.) GO TO 100
C
TTEST=T50049
IF (TTEST-TTHETA/2)
F IF (TTEST-T90D(TTEST) .LT. 20.) GO TO 90
F TTEST=T90D(TTEST)
F CALL OUTCAL
F 100 CONTINUE
IF (BAKING) NBUP=NEUP+1
999 FORMAT (BACKUP U, NTV2, I, NOUE, TEMP, DT, Q=",317,F9.2,2F12.5)
1000 FORMAT ("TANK FULL TO WITHIN .05 OF TANK VOLUME,"/
P 1 ' TOTVOL, NO. NODES COV., HEIGHT OF LIQ., WT. OF LIQ."
112
SUBROUTINE TIMESOU
COMMON /USER1/ NTMTHETA, NBETAS, NTUNIT, BETA, SIN, TNKVL
COMMON /USER2/ NTMTHETA, NTUNIT, BFLOW, TLIK, TGAS
COMMON /SURFA/ SAREA(1)
COMMON /SURFY/ TSURF(1)
COMMON /BNDYT/ TBDY(1)
COMMON /HTRCF/ NCOF(1)
COMMON /SURFG/ GSURF(1)
COMMON /SURFQ/ QSURF(1)
COMMON /OUTVAR/ TOTWT, TOTVOL, ARCOLD, ARCNEW, SRFOLD, VTOTIN
COMMON /HFCLC/ HFSUM, SUMN, VNEMRE, DENS, RHOLH2, HVAP
COMMON /QOA/ QA(1)
COMMON /LSTD/ NFHVG, BOLOST, WOLOST, VOLST
COMMON /LIGST/ MC, BAKING, TRUST, FULL, DELTEMP(100)
COMMON /HBUG/ DEBUG
COMMON /PLTSAV/ NOUT, NCOLD, VOLV(200), VCMC(200), TOTV(200)
COMMON /FIXCON/ XKON(50)
LOGICAL DEBUG
TIMEN=XKON(1)
BACKUP=XKON(12)
IF (BACKUP .GT. 0.) GO TO 49
CALL WRTEMP(T1,0.)
NOUT=NOUT+1
NCOLD=NCOLD+1
CALL STN(1),13)
PRINT 2009, (QSURF(I), I=1, NTHETA)
NTHET I=NTHETA+1
PRINT 2(i12, (DELTM(I), I=1, NTHETA)
PRINT 2010, (QOA(I), I=1, NTHETA)
PRINT 2008, HFSUM, SUMN, HFAVG, BOLOST, WOLOST, VOLST, VNEMRE
PRINT 2007, TOTWT, TOTVOL, ARCOLD, ARCNEW, SRFOLD, VTOTIN
IF (BACKUP .GT. 0.) PRINT 2002, NOUT, (TSURF(J), J=1,13)
IF (TINHRS .LE. 0.) OUTPUT=10.
IF (DEBUG) CALL FPRI
IF (DEBUG) CALL FPRI
50 CONTINUE
51 CONTINUE
F2001 FORMAT (1DF8.3)
F2002 FORMAT (1,1DF10.3)
F2003 FORMAT (1D8.1, LQ0 IN TANK AT THIS TIME'/
   5X,'WEIGHT', 6X,'VOLUME', 6X,'ARC ALONG WALL (IN.)', 3X,'DEPTH OF',
   3X,'SURFACE', 5X,'NO. NODES', 5X,'NO. NODES'/
F2004 3X,'(LBS)', 7X,'(IN**3)', 5X,'LIQUID', 6X,'WETTED',
F2005 4X,'AREA COV', 4X,'COVERED', 6X,'WETTED'/
F2006 5 6G12.5, 2I10)
F2007 FORMAT (4X,'HFSUM', 8X,'SUMN', 7X,'HFAVG', 6X,'BOLOST', 6X,'WOLOST',
F2008 1 6X,'VOLOST', 6X,'VNEMRE')
SUBROUTINE WRTMP(TT, TINC)

TO SAVE THE TIME AND TEMPERATURES AT THAT TIME
FOR LATER PLOTTING OR POSTPROCESSING.

FORMAT OF THE FILE IS AS FOLLOWS,
LINE 1. TITLE UP TO 120 CHARACTERS.
LINE 2. NO. NODES, -99, -99.0, -99.0, DATE/TIME OF RUN
WITH FORMAT(218,2F8.2,AB,IX, A8)
LINE 3. NODE NUMBERS WITH FORMAT (2016)
LINE 4. ETC. TIME, TEMPS OF ALL NODES USING FORMAT (IOE12.6).
THE FINAL TIME AND TEMPS ARE REPEATED WITH THE TIME AS A NEGATIVE NO.

COMMON /DIMENS/ NNA, NND, NNT, NGL, NNG, NCH, NARY, LSEQ
COMMON /FIXCON/KOM(1)
COMMON /TITLE/N
COMMON /TEMP/T
COMMON /XSPACE/ NDIM, NTH, X
COMMON /POINTN/ LNODE, LCOND, LCONS, LARRY, ICOMP
DIMENSION HEADER(20), H(1), T(1), CON(50)
DIMENSION X(1), NX(1)
EQUIVALENCE (KON(1),CON(1))
EQUIVALENCE (X,NX)
DATA KK/0/
IF (LNODE .EQ. 0) CALL NqREAD(1)
DT = CON(2)
NSL = NNT
IF (KK .LT. 0) CALL DADTIME
IF (KK .GT. 0) CALL JDADTIME
WRITE (. i,2001) HEADER, N;L, LL, EI, ELL, CDATE, CTIME
WRITE (23,2002) (_X (I +LNODE), I-l, NSL)
TIME2 - TIME2 + DT
IF (CON(1)*I.0000CI .LT. CON(3)) GO TO 12
IF (TIME2 .LT. TINC) GO TO 50
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
CONTINUE
TIME1 - CON(1)
TIME1 - CON(13) + TINC
WRITE (23,2003) CON(1), CON(13), (T(I),I-I,NSL)
WRITE (23,2003) CON(1), CON(13), (T(I),I-I,NSL)
IF (CON(1)*I.0000CI .LT. CON(3)) GO TO 50
TIME1 - CON(1)
TIME1 - CON(13)
WRITE (23,2003) TIME1, (T(I),I-I,NSL)
WRITE (23,2003) TIME1, (T(I),I-I,NSL)
IF (CON(1)*I.0000CI .LT. CON(3)) GO TO 50
TIME1 - CON(1)
TIME1 - CON(13)
WRITE (23,2003) TIME1, (T(I),I-I,NSL)
WRITE (23,2003) TIME1, (T(I),I-I,NSL)
CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
WRITE (.3,2003) CON(1), (T(I),I-I,NSL)
WRITE (.3,2003) CON(13), (T(I),I-I,NSL)
IF (CON(1) .LT. TIME1) GO TO 50
IF (CON(13) .LT. TIME1) GO TO 50
15 CONTINUE
F115 TIME1 - CON(1)
TIME1 - CON(13) + TINC
TIME2 - 0.
END

END

END

EOF

COSLNDs model
ja -sclf GET ACCOUNTING INFO

# USER=userid
# QSUB -r sphere3
# QSUB -ao
# QSUB -lm 1.5Mw
# QSUB -e
# QSUB -IT 59
# QSUB -eM 1. SMw
# set -x
ja

set echo
model
<<
EDF # SINDA MODEL TO FOLLOW
BCD 3 THERMAL LPCS
REM THIS SINDA MODEL WAS GENERATED BY CRYOTRAN
REM SPHERE --- 2D WEDGE SHELL - NO NODES INSIDE OF TANK
REM WEDGE ANGLE=ßETA = 1.0 RADIANS
REM SAMPLE OF SPHERE NOT NODALIZED IN TANK
REM 9
END

REM NODE TEMPERATURES ARE IN (DEG R)
REM DIMENSIONS ARE IN (IN.), TIME IS IN (SECS)
REM SURFACE NODES, INSIDE TANK WALL
GEN 1001, 40, 1, 540.0, -1.000000 $ SURFACE NODES
DIFFUSION NODES, REGION 1, TANKWALL
REM REGION 1, LAYER NO. 1
REM DIFFUSION NODES, REGION 1, TANKWALL
REM REGION 1, LAYER NO. 1
SIM 2001, 2, 39, 540.0, A1204, 0.58764 $ ALUMINUM 2219
SIM 2002, 2, 37, 540.0, A1204, 1.76025 $ ALUMINUM 2219
SIM 2003, 2, 35, 540.0, A1204, 2.92169 $ ALUMINUM 2219
SIM 2004, 2, 33, 540.0, A1204, 4.06512 $ ALUMINUM 2219
SIM 2005, 2, 31, 540.0, A1204, 5.18348 $ ALUMINUM 2219
SIM 2006, 2, 29, 540.0, A1204, 6.26989 $ ALUMINUM 2219
SIM 2007, 2, 27, 540.0, A1204, 7.31764 $ ALUMINUM 2219
SIM 2008, 2, 25, 540.0, A1204, 8.32027 $ ALUMINUM 2219
SIM 2009, 2, 23, 540.0, A1204, 9.27160 $ ALUMINUM 2219
SIM 2010, 2, 21, 540.0, A1204, 10.15809 $ ALUMINUM 2219
SIM 2011, 2, 19, 540.0, A1204, 10.99728 $ ALUMINUM 2219
SIM 2012, 2, 17, 540.0, A1204, 11.76097 $ ALUMINUM 2219
SIM 2013, 2, 15, 540.0, A1204, 12.45216 $ ALUMINUM 2219
SIM 2014, 2, 13, 540.0, A1204, 13.06657 $ ALUMINUM 2219
SIM 2015, 2, 11, 540.0, A1204, 13.60042 $ ALUMINUM 2219
SIM 2016, 2, 9, 540.0, A1204, 14.05043 $ ALUMINUM 2219
SIM 2017, 2, 7, 540.0, A1204, 14.41381 $ ALUMINUM 2219
SIM 2018, 2, 5, 540.0, A1204, 14.88324 $ ALUMINUM 2219
SIM 2019, 2, 3, 540.0, A1204, 14.87272 $ ALUMINUM 2219
SIM 2020, 2, 1, 540.0, A1204, 14.96453 $ ALUMINUM 2219
REM REGION 1, LAYER NO. 2
SIM 2041, 2, 39, 540.0, A1204, 0.58527 $ ALUMINUM 2219
SIM 2042, 2, 37, 540.0, A1204, 1.75220 $ ALUMINUM 2219
SIM 2043, 2, 35, 540.0, A1204, 2.90833 $ ALUMINUM 2219
SIM 2044, 2, 33, 540.0, A1204, 4.08532 $ ALUMINUM 2219
SIM 2045, 2, 31, 540.0, A1204, 5.15981 $ ALUMINUM 2219
SIM 2046, 2, 29, 540.0, A1204, 6.24122 $ ALUMINUM 2219
SIM 2047, 2, 27, 540.0, A1204, 7.31764 $ ALUMINUM 2219
SIM 2048, 2, 25, 540.0, A1204, 8.32027 $ ALUMINUM 2219
SIM 2049, 2, 23, 540.0, A1204, 9.27160 $ ALUMINUM 2219
NLOOP- 2000, DRLXCA- 0.001000, ARLXCA- 0.001000
IO1-0.00000E+00, 102- 6.0000 , I03-0.20833E-03, I04-0.41667E-02
3CONSTANTS
CIRCUMFERENTIAL CONDUCTORS REGION
KIO-2(DEG
2.542553E+00,A6204, 2.542553E.00
2.526876E+00,A6204, 2.542553E+00
2.310783E+00,A6204, 2.448982E+00
2.307833E+00,A6204, 2.433058E+00
2.295625E+00,A6204, 2.433058E+00
2.207080E+00,A6204, 2.542553E+00
2.115686E+00,A6204, 2.542553E+00
2.013210E+00,A6204, 2.542553E+00
1.910734E+00,A6204, 2.542553E+00
1.808258E+00,A6204, 2.542553E+00
1.705782E+00,A6204, 2.542553E+00
1.603306E+00,A6204, 2.542553E+00
1.500830E+00,A6204, 2.542553E+00
1.398354E+00,A6204, 2.542553E+00
1.295878E+00,A6204, 2.542553E+00
1.193402E+00,A6204, 2.542553E+00
1.090926E+00,A6204, 2.542553E+00
9.887350E-01,A6204, 2.542553E+00
8.807000E-01,A6204, 2.542553E+00
7.726654E-01,A6204, 2.542553E+00
6.646309E-01,A6204, 2.542553E+00
5.566064E-01,A6204, 2.542553E+00
4.485819E-01,A6204, 2.542553E+00
3.405574E-01,A6204, 2.542553E+00
2.325329E-01,A6204, 2.542553E+00
1.245084E-01,A6204, 2.542553E+00
0.000000E+00,A6204, 2.542553E+00

BCD 3CONSTANTS DATA
REM NTHETA, NPHI, N1, N2, N3, N4
1- 40, 2- 3, 3- 1.000, 4- 43.534, 5- 200.000
REM K10=SINDA TEMP UNITS; K10=1(DEG F) K10=2(DEG K)
10= 2
REM TIMEO(MIN) TI0ME1(MIN) DTI0ME1(MIN) OUTPUT(MIN)
REM 0.000000E+00 360.00 0.125000E-01 0.250000
101=0.000000E+00, 102= 6.0000 , 103=0.203833E-03, 104=0.416607E-02
NLOOP= 2000, DR1XCA= 0.000100, ARLXCA= 0.000100

BCD 3ARRAY DATA
1 REGION 1, (TANKWALL ), INSIDE SURFACE AREAS (IN**2)

117
F N0IN= 900
M NTHETA= K1
M NBETAS= K2
M BETA =XK3
M RIN =XR4
M TVOL =XR5
M NTUNIT= K10
F DO 120 I=1,NTHETA
M SARIN(I) =A(I+I)
M SAROUT(I)=A(2+I)
F 120 CONTINUE
F HL= 2400.00  /144.
F HV= 200.000  /144.
F DO 272 I=1, 26
F IMI=I-1
M G(18001+IMI)= HL*SARIN(I)
F 272 CONTINUE
F DO 273 I= 27, 40
F IMI=I-1
M G(18001+IMI)= HV*SARIN(I)
F 273 CONTINUE
STDSTL
END
Bcd 3 variables
F COMMON/USERI/ NTHETA, NBETAS, NTUNIT, BETA, RIN, TVOL
END
Bcd 3 variables
END
Bcd 3 output calls
END
Bcd 3 end of data
EOF
cossinda model
ja -scif # GET ACCOUNTING INFO
APPENDIX C

"CryoTran Model" Files Part 3

Sample Case of SOLA-ECLIPSE

- Input Data Requirements
- Input Screens for Sample Case
- Model to submit to CRAY
- Run Output
- Plots
**Input Data Requirements**

**file management system**

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Input file</td>
</tr>
<tr>
<td>6</td>
<td>Printed output file including debug output</td>
</tr>
<tr>
<td>7</td>
<td>History file</td>
</tr>
<tr>
<td>8</td>
<td>Formatted plotfile</td>
</tr>
<tr>
<td>9</td>
<td>Unformatted plotfile (now formatted)</td>
</tr>
<tr>
<td>10</td>
<td>Restart input file</td>
</tr>
<tr>
<td>11</td>
<td>Restart output file</td>
</tr>
<tr>
<td>53</td>
<td>Printed debug file</td>
</tr>
<tr>
<td>59</td>
<td>Debug file (now unit 6)</td>
</tr>
</tbody>
</table>

The CRAY will automatically dispose units 6, 7, and 9.

The input data to run the code is to be provided either in an input file or may be keyed in within CryoTran. The prepared input file may be resident on the VM computer or on the CRAY. The user will be interrogated for the name of the file in CryoTran. The only additional means of providing input is through the restart file if the user is restarting an analysis. All input values except the problem title, (NAME), is input by use of NAMELIST data. All reading of the input file is performed by Subroutine READER. If a restart of an analysis is being performed, only the title and the first namelist, CNTRL, is required. Many variables have default values given in the program. If a value for a variable is provided in the input file, it replaces the default value.

The names of the namelists are: CNTRL, HYDRO, MESHES, ASETIN, THERMS, TURB, and FEATS. The input variable names are listed below grouped by namelist. Each subgroup has a brief description of its contents followed by a listing of the input variables contained in it. An explanation of each variable and the default value, if any, is provided. The default values are enclosed in brackets, [ ].

### Variable Name Description

**NAME**

- problem identification, (title), 48 characters maximum.

**Namelist CNTRL**

These variables are primarily associated with controlling execution of the code.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTIND</td>
<td>automatic adjustment of time step adjustment</td>
<td>[1.0]</td>
</tr>
<tr>
<td>BUGB</td>
<td>print plotfile data to unit 6 in time step</td>
<td>[.false.]</td>
</tr>
<tr>
<td>DELT</td>
<td>time step</td>
<td>[0.001]</td>
</tr>
<tr>
<td>DTCRMX</td>
<td>maximum delt using conjugate residual solution method</td>
<td>[0]</td>
</tr>
<tr>
<td>IDIV</td>
<td>divergence correction flag</td>
<td>[1]</td>
</tr>
<tr>
<td>IEQIC</td>
<td>flag used to activate equilibrium free surface</td>
<td>[none]</td>
</tr>
<tr>
<td>ISOR</td>
<td>pressure iteration solution method</td>
<td>[0]</td>
</tr>
<tr>
<td>NDUMP</td>
<td>ndump=0 new case</td>
<td>[0]</td>
</tr>
<tr>
<td>NPACK</td>
<td>flag to activate packing; 0-off, 1-on</td>
<td>[0]</td>
</tr>
<tr>
<td>PLTDT</td>
<td>time increment between plots and/or prints to be output on film</td>
<td>[none]</td>
</tr>
<tr>
<td>TRFIN</td>
<td>problem time to end calculation</td>
<td>[none]</td>
</tr>
<tr>
<td>UHSC</td>
<td>selects execution of hydrodynamics</td>
<td>[none]</td>
</tr>
<tr>
<td>UHET</td>
<td>selects execution of heat transfer</td>
<td>[none]</td>
</tr>
<tr>
<td>LHERM</td>
<td>not used</td>
<td>[none]</td>
</tr>
<tr>
<td>LTURB</td>
<td>selects turbulence model in hydro sol.</td>
<td>[none]</td>
</tr>
</tbody>
</table>

**Namelist HYDRO**

The variables in this namelist fall into several categories: fluid properties, constants used in the numerical method, gravitational environment, and problem initial state. For dimensional variables the burden of maintaining a consistent set of units falls to the user. One possible consistent set of units is provided in parentheses after the description of each variable where appropriate.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>controls amount of donor cell fluxing</td>
<td>[1.0]</td>
</tr>
</tbody>
</table>
## Namelist MESH

All variables in this group are required for generation of the computational mesh. All lengths must be expressed in units consistent with the fluid properties in the HYDRO namelist. For array variables, the number of entries required is enclosed in [ ] after the description.

### Variables:

- **DXMN**, **DYMN**: minimum cell width (delta-x) in submesh n (N)
- **ICTL**: mesh geometry indicator (0)
- **NKK**, **NKY**: number of submesh regions in x-direction and y-direction respectively (none)
- **NXI(NKX)**, **NXR(NKX)**: number of cells between locations xi(n) and xc(n) in submesh n (NKX)
- **NYI(NKY)**, **NYR(NKY)**: number of cells between locations yi(n) and yc(n) in submesh n (NKY)
- **XCI(NKX)**, **XLI(NKX)**, **YCI(NKY)**, **YLI(NKY)**: x- and y-coordinates of the convergence point in submesh n

### Namelist ASETIN

All variables in this namelist are associated with solid boundary definition within the computational mesh. The boundary modeling procedure is detailed elsewhere in ref.??????? with only brief descriptions of the variables included in this section.
Select effect of obstacle function
-1 to add obstacle "inside" function
0 to subtract obstacle "inside" function

Number of obstacle functions to be defined

Coefficient of \( x^1 \) in obstacle function
Coefficient of \( y^1 \) in obstacle function
Coefficient of \( x^2 \) in obstacle function
Coefficient of \( y^2 \) in obstacle function
Coefficient of \( xy \) in obstacle function
Constant term in obstacle function

Namelist THERMS
The variables in this namelist provide the thermal properties of the fluid being modeled.

CPL cp
FLK k
GASVOL volume of ullage (cm³) [none]
Gsurf surface area of ullage (cm²) [none]
PGAS initial gas pressure (Pa) [none]
PGEND gas pressure to terminate execution (Pa) [none]
PSAT liquid saturation pressure (Pa) [0.0] if nonzero code simulates phase change in nf - 5 cells
QFLBC heat flux at boundary [none]
TFLD initial liquid temperature (K) [none]
TGAS initial gas temperature (K) [none]

Namelist TURB
The variables in this list will provide constants required by the turbulence model. To date, the turbulence models being used do not require input data. This namelist has not been used but its "place" is being held by a dummy variable.

Namelist FEATS
Certain aspects of the code execution and model building require modification and/or addition of FORTRAN coding. Although these areas are clearly identified within the program, new coding obviously requires a higher level of effort than simply changing input variables. FEATS has been established to simplify the inclusion and exclusion of features which commonly occur, such as tank inlets and outlets. Each feature typically requires a logical variable which selects activation of the feature and associated dimensional variables which define the attributes of the feature.

LDRAIN selects activation of tank outlet at bottom [.false.]
DRAINID inside diameter (cm) [none]
DRAINOD outside diameter (cm) [none]
DRAHQV volume flow rate (cm³/s) [none]
VDRAIN fluid velocity (cm/s) [none]
VTJET size of outlet and suction [.false.]
BOTJET y-value for jet suction (cm) [0.0]
EPSJET jet turbulence energy dissipation rate (cm³/s³) [0.0]
RADJET jet radius (cm) [0.0]
TKEJET jet turbulence kinetic energy (cm³/s³) [0.0]
TMFJET jet fluid temperature (K) [0.0]
TOPJET y-value of jet issue (cm) [0.0]
VELJET jet fluid velocity (cm/s) [0.0]
VOLJET volume flow rate, (cm³/s) [0.0]

LQBOT selects heat flux at bottom of tank [.false.]
QBOT heat flux for bottom half of tank [none]
LQTOP selects heat flux at top of tank [.false.]
QTOP heat flux for top half of tank [none]
LQUNI selects uniform heat flux at tank walls if .true., uniform heat flux [.false.]
QUNI non-uniform heat flux or adiabatic [none]
Sample Case of SOLA-ECLIPSE; Input Screens

WELCOME TO CRYOTRAN
YOU WILL BE PROMPTED FOR ALL NECESSARY INPUT.
READ THE INSTRUCTIONS CAREFULLY.
TYPE IN THE INPUT DATA CAREFULLY TO AVOID TROUBLE,
YOU MAY QUIT THE PROGRAM AT ANY INPUT PROMPT BY TYPING A "Q" (QUIT)

ENTER THE NUMBER FOR THE DESIRED PROBLEM TYPE
THE PROBLEM TYPES ARE AS FOLLOWS:

1 - THERMO/THERMAL SINDA ANALYSIS ON A SPHERE.
2 - THERMO/THERMAL SINDA ANALYSIS ON A CYLINDER.
3 - RUN A PRESTORED ANALYSIS PROGRAM

CHOOSE THE ANALYSIS PROGRAM YOU WISH TO USE.
TYPE IN THE NUMBER OF THE DESIRED ANALYSIS.

1 NOVENT FILL
2 CHILL TO TEMP
3 TARGET FOR NVFILL
4 SOLA-ECLIPSE
5 CSAM

THIS TASK IS BEING SET UP FOR THE CRAY,
NOW INPUT NECESSARY CRAY INFO.

WHICH CRAY SYSTEM CGS OR UNICOS
TYPE IN C OR U

TYPE IN YOUR CRAY USERID.
userid

TYPE IN YOUR CRAY PASSWORD.
password

TYPE IN NO. OF CRAY CPU SECONDS TO BE USED.
IF NUMBER OF SECONDS REQUESTED IS < 10, 60 WILL BE USED.

NOW GIVE YOUR JOB A NAME, TYPE IN THE NAME,
1 - 7 ALPHABETIC CHARACTERS.
ecltest

THE CRAY JCL THAT WAS INPUT IS AS FOLLOWS:
USERID = userid
PASSWORD = password
CPU TIME REQUEST = 59 SECS.
MEMORY REQUEST = 1500000 words
JOB NAME = ecltest

ARE THESE ALL CORRECT? TYPE Y OR N OR Q TO QUIT

NOW WE NEED THE INPUT DATA FOR THE ANALYSIS
THIS INPUT DATA CAN BE:
1 STORED ON CRAY
2 STORED ON VM
3 TYPED IN NOW
4 NO INPUT DATA FOR THIS ANALYSIS
   TYPE IN 1 2 3 OR 4

ANALYSIS INPUT DATA IS STORED ON VM
NOW WE NEED FILE NAME; FILE TYPE; FILE MODE
   TYPE IN FN FT FM

solaecl testia a
JCL COMMAND - IRC-FILEDEF VMADATA DISK SOLAECL TESTIA A
JCL COMMAND - IRC-FILEDEF VMADATA CLEAR

THE INPUT DATA IS NOW ALL IN.
END OF CRYOTRAN PREPROCESSOR PROGRAM,
THE OUTPUT FILE IS CALLED "CRYOTRAN MODEL".

DO YOU WANT TO GO TO BEGINNING OF SYSTEM OR QUIT?
   TYPE Y TO GO BACK TO BEGINNING OF SYSTEM,
   OR TYPE N TO QUIT.

ON TO ANALYSIS PROGRAM
THE OUTPUT FILE IS CALLED "CRYOTRAN MODEL".
THIS "CRYOTRAN MODEL" FILE IS A SOLA-ECLIPSE MODEL.

USER MAY NOW SUBMIT THE FILE 'CRYOTRAN MODEL'
TO THE CRAY COMPUTER FOR EXECUTION,
OR MAKE ANY DESIRED MODIFICATIONS WITH AN EDITOR
PRIOR TO SUBMITTING IT TO THE CRAY.

TO SUBMIT THE FILE TO CRAY,
ON THE VM SYSTEM TYPE: CR-submit CRYOTRAN MODEL

UPON COMPLETION OF THE CRAY EXECUTION OF SOLA-ECLIPSE
USER MAY PLOT THESE RESULTS BY TYPING:

DOECLPLT SOLA PLOTFILE

DOECLPLT SOLA PLOTFILE *
produce plots from SOLA output

THE FOLLOWING VARIABLES ARE AVAILABLE TO CONTROL GRAPHICAL OUTPUT

CURRENT VALUE  

DEBUG = T/F ------- WRITE DEBUG OUTPUT TO FILE <DEBUG OUTPUT> F
MODE = INTEGER- 1 NO PAUSES, NO PROMPTS -------------- 1
LVEC = T/F ------ PRODUCE VELOCITY FIELD VECTOR PLOTS------ T
LCON = T/F ---- PRODUCE TEMPERATURE FIELD CONTOUR PLOTS--- F
LSYMPL = T/F ---- ABBREVIATED/FULL DISPLAY --------------- F
LSYM2 = T/F ---- SYMMETRIC/HALF-FIELD DISPLAYS ---------- T

WANT TO CHANGE THE VALUE OF ANY OF THE CONTROL VARIABLES? (Y/N)

CEGDIN100I Graphics device NOT assigned.
CEGDIN100I ENTER desired device name or HELP HELP.
CEGDIN100R DEFAULT to CANCEL.

Name of plot device to print plots

CEGDIO3001 Device LGA0S attached.

SOLA-ECL GRAPHICS PROGRAM, NORMAL TERMINATION

File 2480 TO RSCS COPY 001 NOHOLD

CEGDIO3001 Device DISCONNECTED from VM/GRAPH3D.

Ready; T-**/**.** 16:21:40

DMTRGXI701 FROM MVSLERCI: 16.24.15 JOB 2480 HASP546 RSCS2480 (JOB 2480 FROM MVSLERCI ) SYSTEM OUTPUT RECEIVED AT MVSLERCI
Sample Case of SOLA-ECLIPSE; Model File, Output of Preprocessor

```plaintext
# USER = vvglenn  PW = dendrob
# QSUB -r ecltest  # jobname
# QSUB -eo  # Combine error and standard output
# QSUB -IT 59  # CPU time
# QSUB -IM 1.5Mw  # Memory requested
# #  # End MOS statements
# set -x  # set echo
#
## This SOLA file, (model), was generated by CRYOTRAN.

cat > model << EOF  # DATA FROM VM, FN FT FM= SOLAECL TESTI
OTVJ: LH2, 50%, 4.5 CM/S
&CTRL
NDUMP= 0,
TWFIN= 0.1,
DELT= 0.01,
ISRN= 1,
IDEFM= 0,
LHYDRO= .TRUE.,
LTHEM= .FALSE.,
LSTIN= .FALSE.,
LHEAT= .TRUE.,
PLDT= 60.,
AUTOT= 1,
NTDFX= 1.0,
IEQIC= 1,
IDEF= 0,
NPACK= 0,
IDIV= 0,
&END
&HYDRO
NLHR= 0.0701,
ISRF0= 1,
RDNF= 1.803E-3,
SIGMF= 1.813,
EPS1= 1.E-5,
ALPHA= 1.0,
GXI= 0.0,
GY= 0.0,
Vl= 0.0,
V2= 0.0,
KL= 1,
MA= 1,
FLU= 0.0,
PERFUL = 50.0,
&END
&MESHES
ICYL= 1,
NKX= 1,
XL= 0.0, 210.0,
XC= 20.0,
NXK= 2,
NXR= 10,
DYM= 10.0,
NKY= 1,
YL= 0.0, 1020.0,
YC= 30.0,
NYK= 1,
NYL= 33,
DYMN= 30.0,
&END
&ASETIN
NON= 4,
IOH(1)= 1,
OA2(1)= 0.0,
OB2(1)= 0.0,
OC2(1)= 0.0,
OA1(1)= 0.0,
OC1(1)= 210.0,
IOH(2)= 0,
OA2(2)= 1.0,
OB2(2)= 1.0,
OC2(2)= 0.0,
OA1(2)= 0.0,
OC1(2)= 0.0,
IOH(3)= 1,
OA2(3)= 0.0,
OB2(3)= 0.0,
OC2(3)= 0.0,
OA1(3)= 0.0,
OC1(3)= 0.0,
IOH(4)= 0,
OA2(4)= 1.0,
OB2(4)= 1.0,
OC2(4)= 0.0,
OA1(4)= 0.0,
OC1(4)= 0.0,
&END
&THERMS
PSAT= 0.0,
PGAS= 20.0,
TFLD= 21.0,
TGAS= 21.0,
GSURF= 0.0,
GASVOL= 0.0,
PGEND= 0.0,
QFLBC= 0.0,
CPL= 9.6E+7,
FLK= 1.0E+04,
&END
&TURB
```

127
DUMMY=0.0,

&FEATS
LDRAIN= .FALSE.,
LQUNI = .FALSE.,  QUNI= 0.0,
LQTOP = .TRUE.,  QTOP= 1.3,
LQBOT = .TRUE.,  QBOT= 1.3,
LJET = .TRUE.,
RADJET= 10.0,  BDTJET= 60.0,  TOPJET= 120.0,
VELJET= 4.5,  VOLJET= 0.0,  TMPJET= 18.0,

&END

&FLOBC
NFLBC = 0,
XEND1(1) = 0.0,  YEND1(1) = 0.,  XVEL(1) = 0.,
XEND2(1) = 0.00,  YEND2(1) = 0,  YVEL(1) = 0.,

&END

EOF

Ja -sclf # GET ACCOUNTING INFO
Sample case of SOLA-ECLIPSE; CRAY Output of Run
This output file from the CRAY has been edited and some of the output deleted.

BEGIN EXECUTION OF SOLA ECLIPSE
NAME = OTVJ1. LHR, 50%, 4.5 CM/S
4CTRL NDUMP = 0, TFWIN = 0.1, FRTDT = 10000000000., PLTDT = 60., DELT = 1.E-2, AUTOT = 1., ISGR = 1.,
CON = 0.35.,
4DEPM = 0, NPACK = 0, DTCRMS = 1., DIV = 0, IEQIC = 1, LHYDO = .T., LHEAT = .T., LTERM = .F.,
4LURB = .T.,
DEBUG = .F., BUGB = .F., &END
4HYDRO RHOP = 7.016E-2, XNU = 1.803E-3, EPSI = 1.E-5, GX = 0., GY = 0., UI = 0., VI = 0., OMG = 1.7,
ALPHA = 1., KL = 1.,
KR = 1., KT = 1., KB = 1., FLRT = 0., XI = 1.813, ISRFL1 = 1., CANCLE = 5., PERFUL = 50., &END
4MESHES YDL = 1., NXX = 1., KL = 0., 210., 180.0, XC = 20., 190., NXL = 2, 190, NXA = 10, 19.0,
DXNN = 10., 19.0,
NXY = 1, YL = 0., 1020., 180., YC = 30., 19.0, NYL = 1, 19.0, NYR = 33, 19.0, DYMN = 30., 19.0,
&END
4ASETN HNBS = 4, OA2 = 0., 1., 1., 16.0, OA1 = 20.0, OB2 = 0., 1., 1., 16.0, OB1 =
1., -120., -1.,
-1620., 16.0, GC2 = 20.0, OCL = -210., 0., 810., 612000., 16.0, IOH = 1, 0, 0, 17.0, &END
4THERMS PSAT = 0., PGAS = 20., TFLD = 21., TQAS = 21., CFLBC = 0., GRDIFF = 0., GASUL = 0., &END
CFL = 96000000.,
FLK = 10000., &END
4TURB DUMMY = 0., &END
4FEATS LDRAIN = .F., DRANID = 0., DRANOD = 0., VDRAIN = 0., DRANOV = 0., LQUNI = .F., QUNI = 0., LQTOP =
.T.,
QTOP = 1.3, LQBOT = .T., QBOT = 1.3, LQET = .T., LQET = 10., QTJET = 60., TOPJET = 120., VELOJET =
4.5, VOLJET = 0.,
4MESHES FCYL = 18., TKEJET = 0., EPSJET = 0., &END
4FLBNC NXLBC = 0., XEND1 = 10.0, XEND2 = 10.0, YEND1 = 10.0, YEND2 = 10.0, XVEL = 10.0, YVEL =
10.0, TBC = 10.0,
&END
1 FOLLOWING VALUES COMPUTED & PRINTED IN <MSHSET>
X (1) = 0.00000E+00 RX (1) = 0.00000E+00 DELX (1) = 1.00000E+01 RDX (1) = 1.00000E+01 XI (1) = 5.00000E+00 RXI (1) =
1.00000E+01
X (2) = 1.00000E+01 RX (2) = 1.00000E+00 DELX (2) = 1.00000E+00 RDX (2) = 1.00000E+00 XI (2) = 5.00000E+00 RXI (2) =
1.00000E+01
X (3) = 2.00000E+00 RX (3) = 5.00000E-02 DELX (3) = 1.00000E+00 RDX (3) = 1.00000E+00 XI (3) = 1.50000E+01 RXI (3) =
1.00000E+01
X (4) = 3.00000E+00 RX (4) = 3.33333E+00 DELX (4) = 1.00000E+00 RDX (4) = 1.00000E+00 XI (4) = 2.50000E+01 RXI (4) =
1.00000E+01
X (5) = 4.00000E+00 RX (5) = 2.38095E+02 DELX (5) = 1.20000E+01 RDX (5) = 8.33333E+02 XI (5) = 3.60000E+01 RXI (5) =
1.00000E+01
X (6) = 5.00000E+00 RX (6) = 1.78571E+02 DELX (6) = 1.40000E+00 RDX (6) = 7.14286E+02 XI (6) = 4.90000E+01 RXI (6) =
1.00000E+01
X (7) = 6.00000E+00 RX (7) = 1.38889E+02 DELX (7) = 1.60000E+01 RDX (7) = 6.25000E+02 XI (7) = 6.40000E+01 RXI (7) =
1.00000E+01
X (8) = 7.00000E+00 RX (8) = 1.11111E+02 DELX (8) = 1.80000E+01 RDX (8) = 5.55556E+02 XI (8) = 8.10000E+01 RXI (8) =
1.00000E+01
X (9) = 8.00000E+00 RX (9) = 9.09091E+03 DELX (9) = 2.00000E+01 RDX (9) = 5.00000E+00 XI (9) = 1.00000E+02 RXI (9) =
1.00000E+01
X (10) = 9.00000E+00 RX (10) = 7.57576E-03 DELX (10) = 2.20000E+01 RDX (10) = 4.54545E-02 XI (10) = 1.21000E+00 RXI (10) =
1.00000E+00
RXI (10) = 8.26436E+03 RXI (10) = 1.86000E+03 RDX (11) = 6.41026E-03 DELX (11) = 2.40000E+01 RDX (11) = 4.16667E-02 XI (11) = 1.44000E+00
RXI (11) = 6.94444E-03 DELX (12) = 5.94516E-03 RDX (12) = 2.60000E+01 RDX (12) = 3.84615E-02 XI (12) = 1.69000E+00
RXI (12) = 5.91716E-03 DELX (13) = 4.76190E-03 RDX (13) = 2.80000E+01 RDX (13) = 3.57143E-02 XI (13) = 1.96000E+00
RXI (13) = 2.10000E+03 DELX (14) = 2.38000E+02 RDX (14) = 8.00000E+00 RDX (14) = 3.57143E-02 XI (14) = 2.24000E+00
RXI (14) = 4.44429E-03
1 FOLLOWING VALUES COMPUTED & PRINTED IN <MSHSET>
Y (1) = 0.00000E+00 DELY (1) = 3.00000E+01 RDY (1) = 3.33334E-02 YJ (1) = -1.50000E+01 RYJ (1) = -6.66667E-02
Y (2) = 3.00000E+01 DELY (2) = 3.00000E+01 RDY (2) = 3.33334E-02 YJ (2) = -1.50000E+01 RYJ (2) = -6.66667E-02
Y (3) = 6.00000E+01 DELY (3) = 3.00000E+01 RDY (3) = 3.33333E-02 YJ (3) = 4.50000E+01 RYJ (3) = 2.22222E+00
Y (4) = 9.00000E+01 DELY (4) = 3.00000E+01 RDY (4) = 3.33333E-02 YJ (4) = 7.50000E+01 RYJ (4) = 1.33333E+01
129
Y(5) = 1.20000E+02  DELY(5) = 3.00000E+01  RDY(5) = 3.33333E-02  YJ(5) = 1.05000E+02  RYJ(5) = 9.52381E-03

... data deleted

Y(34) = 9.90000E+02  DELY(34) = 3.00000E+01  RDY(34) = 3.33333E-02  YJ(34) = 9.75000E+02  RYJ(34) = 1.02564E-03

Y(35) = 1.02000E+03  DELY(35) = 3.00000E+01  RDY(35) = 3.33333E-02  YJ(35) = 1.00500E+03  RYJ(35) = 9.95025E-04

Y(36) = 1.05000E+03  DELY(36) = 3.00000E+01  RDY(36) = 3.33333E-02  YJ(36) = 1.03500E+03  RYJ(36) = 9.66184E-04

--- CONSTANTS COMPUTED IN <SETUP> ---

BOND NUMBER = 0.00000E+00

--- data has been deleted

1 1 0.00000E+00 1.00000E+00 1.00000E+00 1.00000E+00 0.00000E+00 0.00000E+00
2 1 0.00000E+00 1.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00

ITER= 0  TIME= 0.00000E+00  DELT= 1.00000E-02  CYCLE= 0  VCHGT= 1.13791E+00

OTVJI: LH2, 50%, 4.5 CM/S

ITER= 88  TIME= 1.02000E-02  DELT= 1.02000E-02  CYCLE= 1  VCHGT= 1.03037E+00

130
<table>
<thead>
<tr>
<th>P</th>
<th>J</th>
<th>I</th>
<th>V</th>
<th>U</th>
<th>D</th>
<th>F</th>
<th>P.S</th>
<th>N.F</th>
<th>CYCLE</th>
<th>VOL(K)</th>
<th>PR(K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>3.45791E-01</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>6.08956E+00</td>
<td>ON CYCLE 1</td>
<td></td>
</tr>
</tbody>
</table>

FLUID VOLUME = 6.08956E+00

ITER= 76  TIME= 2.05020E-02  DELT= 1.03020E-02  CYCLE= 2  VCHGT= 1.13801E+00

ITER= 45  TIME= 3.07010E-02  DELT= 1.01999E-02  CYCLE= 3  VCHGT= 1.24622E+00

... data has been deleted
<table>
<thead>
<tr>
<th>ITER</th>
<th>TIME</th>
<th>DELT</th>
<th>CYCLE</th>
<th>VCHGT</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>4.07900E-02</td>
<td>1.00970E-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>5.09959E-02</td>
<td>1.01980E-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6.12959E-02</td>
<td>1.02999E-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>7.16988E-02</td>
<td>1.04029E-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>8.22058E-02</td>
<td>1.05070E-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>9.28178E-02</td>
<td>1.06120E-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.07182E-02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* EXITQ CALLED FROM <MAIN>: NORMAL TERMINATION *

IN EXITQ, IO= 6, DEBUG= F

******************************************************************************
OTVJ1: LH2, 50%, 4.5 CM/S
TIME = 0.000 CYCLE = 0
APPENDIX C

"CryoTran Model" Files Part 4

Output File (Model) to Run CSAM
This CRCSAM file, (model), was generated by CRYOTRAN.

NAME

# JOB DAT
NPROP(1) = 1, NPHSS(1) = 1,

KTCN=2,
GPogl) = 0.0001, 1.0E10,
TVPNI) = 22,
F(1) = 25,
TVNPGl) - 10 , 1.0E10,
TVNPGl) - 10, 1.0E10,
TVRATE(1) = 0.050,0.175,
XNSTRAC(1) = 0.99999,
OUTPGl) = 0.,120.,70.00,1:00,0.,1.0E10,
HETMPG l) = 500, 1.0E9,
FRATE(1) = 2,
FLOT = 1,
DSTV(1) = 4, 4,
TEND = 810.0,
TSN = 0.5, TSN = 5.000,
DMPMAX = 1,
NDTX = 2,
TPF() = 30.,120.,5.,150.,60.,810.,
NQPF = 10,
ISTST = 1, &END

*NODE
LIG-1, 7, 40.000, 23.50, 2, 1
LIG-2, 7, 40.000, 23.500, 1, 1
GAS-1, 6, 40.000, 2, 1, 1
TANK-A, 4, 36.392, .001
TANK-B, 4, 36.474, .001
TANK-C, 4, 36.565, .001
TANK-D, 4, 36.511, .001
TANK-E, 4, 36.718, .001
TANK-F, 4, 36.507, .001
VCS-A, 4, 40.0, .001
VCS-B, 4, 120.0, .001
VCS-C, 4, 130.0, .001
VCS-D, 4, 140.0, .001
VCS-E, 4, 145.0, .001
VCS-F, 4, 140.0, .001
MLI-A, 5, 360.0, .001
MLI-B, 5, 370.0, .001
MLI-C, 5, 370.0, .001
MLI-D, 5, 371.0, .001
MLI-E, 5, 373.0, .001
MLI-F, 5, 377.0, .001
VAC-JKT, 0, 330.000
BOTT-PEN-1, 3, 34.0, .0001
BOTT-PEN-2, 3, 44.0, .0001
BOTT-PEN-3, 3, 154.0, .0001
TOP-PEN-1, 3, 164.0, .0001
TOP-PEN-2, 3, 164.0, .0001
BOTT-PEN-3, 3, 454.0, .0001
SUPPT-A, 11, 144.0, .001
SUPPT-B, 11, 144.0, .001
TORSLINK, 11, 44.0, 0.3
HX-1-1, 120, 37.000, 40., .1175, .001, 4, 1, 7.5, 0.0,
HX-1-2, 120, 37.000, 5., .1175, .001, 4, 1, 7.5, 1.0,
HX-1-3, 120, 137.000, 144., .1175, .001, 4, 1
HX-1-4, 120, 137.000, 72., .1175, .001, 4, 1
HX-1-5, 120, 157.000, 12., .1175, .001, 4, 1
HX-1-6, 120, 37.000, 4.0, .1175, .001, 4, 2, 7.5, 1, 0.
HX-2-1, 120, 107.000, 8.0, .1175, .001, 4, 2
HX-2-2, 120, 137.000, 36.0, .1175, .001, 4, 2
HX-2-3, 120, 137.000, 144., .1175, .001, 4, 2
HX-2-4, 120, 137.000, 72., .1175, .001, 4, 2
HX-2-5, 120, 137.000, 12., .1175, .001, 4, 2
HX-2-6, 120, 437.000, 12., .1175, .001, 4, 2
HX-2-7, 120, 437.000, 12., .1175, .001, 4, 2

*CONDUCTOR
LIQ-1, LIQ-2, 7, 5.277, 0.579, 1
LIQ-2, GAS-1, 6, 5.277, 0.29, 1
TANK-A, LIQ-1, 7, 6.33010, .80000, 1
TANK-B, LIQ-2, 7, 6.33010, .80000, 1
TANK-C, LIQ-2, 7, 6.33010, .80000, 1
TANK-D, LIQ-2, 7, 6.33010, .80000, 1
TANK-E, GAS-1, 6, 6.33010, .80000, 1
TANK-F, GAS-1, 6, 6.33010, .80000, 1

TANK-A, TANK-B, 4, .0305, 1.
TANK-B, TANK-C, 4, .03975, .9
TANK-C, TANK-D, 4, .0409, .75
TANK-D, TANK-E, 4, .03975, .9
TANK-E, TANK-F, 4, .0305, 1.

VCS-A, VCS-B, 4, .0129, 1.043
VCS-B, VCS-C, 4, .0168, .939
VCS-C, VCS-D, 4, .01727, .7824
VCS-D, VCS-E, 4, .0168, .939
VCS-E, VCS-F, 4, .0129, 1.043

MLI-A, VCS-A, 5, 7.18, .0521
MLI-B, VCS-B, 5, 7.18, .0521
MLI-C, VCS-C, 5, 7.18, .0521
MLI-D, VCS-D, 5, 7.18, .0521
MLI-E, VCS-E, 5, 7.18, .0521
MLI-F, VCS-F, 5, 7.18, .0521

VAC-JKT, MLI-A, 5, 7.68, .0521
VAC-JKT, MLI-B, 5, 7.68, .0521
VAC-JKT, MLI-C, 5, 7.68, .0521
VAC-JKT, MLI-D, 5, 7.68, .0521
VAC-JKT, MLI-E, 5, 7.68, .0521
VAC-JKT, MLI-F, 5, 7.68, .0521
VAC-JKT, SUPPT-A, 11, .0179, .26
VAC-JKT, SUPPT-B, 11, .0179, .26
SUPPT-A, TANK-C, 11, .0179, .07
SUPPT-B, TANK-D, 11, .0179, .07

BOT-PEN-1, BOT-PEN-11, 3, 0.001088, 0.75
BOT-PEN-11, TANK-A, 4, 0.04, 1.0
BOT-PEN-2, BOT-PEN-A, 3, 0.001088, 0.75
BOT-PEN-A1, TANK-A, 4, 0.00565, 1.0
HX-2-1, HX-2-5, 4, 0.000116, 0.67
HX-1-5, HX-1-4, 4, 0.000116, 0.62
VAC-JKT, HX-1-5, 4, 0.000116, 1.30
TOP-PEN-1, HX-2-2, 4, 0.003, 0.17
TOP-PEN-1, TANK-F, 3, 0.000116, 0.21
TOP-PEN-2, TOP-PEN-1, 3, 0.000116, 0.5
VAC-JKT, TOP-PEN-2, 3, 0.000116, 0.13
TOP-PEN-2, HX-1-5, 4, 0.003, 0.17

SUPPT-B, VCS-D, 12, 0.0004, .229
SUPPT-A, VCS-C, 12, 0.0004, .229
BOT-PEN-1, HX-2-1, 4, 0.001088, 0.75
BOT-PEN-A1, HX-1-1, 4, 0.208, 0.0833

VCS-G, TANK-D, 11, .11, 1.0
VCS-C, TANK-C, 11, .11, 1.0
VCS-A, TANK-A, 11, .05, 1.0
VCS-A, TANK-A, 99, .25
VCS-B, TANK-B, 99, .25
VCS-C, TANK-C, 99, .25
VCS-D, TANK-D, 99, .25
VCS-E, TANK-E, 99, .25
VCS-F, TANK-F, 99, .25
APPENDIX D

SUBROUTINE DESCRIPTIONS
SUBROUTINE DESCRIPTIONS

MAIN PROGRAM (No. 0)
The Main Program calls on a subroutine to initialize data values and then calls on the menu subroutines for the model definition.

SUBROUTINE CLEARS (no. 01)
Called from various routines.
Subroutine to clear the terminal screen. This routine calls the system dependent subroutine CLEAR.
CALL CMSCMD('VMFCLEAR', 16, IRT)
CALL CLEAR

SUBROUTINE READAL (No. 02)
Called from various routines.
Subroutine to read the input responses from the user and do some validity checks.
READAL...Read n alphabetic characters, n= 1,2,3
This routine has additional entry points:
ENTRY READLC...Read alphabetic characters, no test.
ENTRY READIN...Read integer and test upper and lower bounds.
ENTRY READRE...Read a real number, test for alphabetic characters

SUBROUTINE DOJCL(COMAND) (No. 03)
Called from MAIN, INDAT2 and INSERT.
Subroutine to execute VM system JCL commands from inside a FORTRAN program.
This routine is system dependent; see note above.
This routine calls "CALL SYSCMD(COMAND,IRC)"

SUBROUTINE BLHDRS (No. 04)
Called from SINTRU, CONDRS, SNBLKS, NODES.
Read SINDA block header information and write block headers for each SINDA block.
This subroutine has 14 entry points.
ENTRY BLHDRS
ENTRY RDTITL
ENTRY BLOTrL
ENTRY BL1TrL
ENTRY BL2TrL
ENTRY BL3TrL
ENTRY BL4TrL
ENTRY BL5TrL
ENTRY BL6TrL
ENTRY BL7TrL
ENTRY BL8TrL
ENTRY BL9TrL
ENTRY BLKEND
ENTRY ENDDAT

SUBROUTINES (No. 05)
Called from VMINTR.
A collection of routines (programs) that will be executed interactively on VM from within the system.

SUBROUTINE INITAL (no. 1)
Called from the MAIN.
Subroutine to initialize common data blocks.
SUBROUTINE MENU1 (No. 2)
Called from MAIN.
This subroutine has the user specify the problem type. The problem types are:
1. Thermal/Thermo analysis on a sphere (SINDA).
2. Thermal/Thermo analysis on a cylinder (SINDA).
3. Run an analysis program without generating a SINDA model for the two geometries described above.
This subroutine may need to be changed as new capabilities are added to the program.

SUBROUTINE MENU2 (No. 3)
Called from MAIN.
This subroutine requests input from the user to specify the analysis program that is to be executed, based on the problem type input in MENU1.
This subroutine will need to be changed as new capabilities are added to the program.

SUBROUTINE SINTRU (No. 4)
Called from MAIN.
This subroutine is used to define the geometric regions and then to generate the SINDA model.

SUBROUTINE GETJCL (No. 41)
Called from SINTRU, NOSIND, INDAT1.
Subroutine to obtain CRAY JCL information from the user and then to write this JCL as file 1 on unit 10 (model file).
The other entry points are called from SINTRU and NOSIND to generate the JCL file for the application requested by the user. The entry points are:
ENTRY RITJCL
ENTRY RITJC2
ENTRY RITJC3
If a different computer system is used this subroutine must be changed to reflect the proper JCL of the system used.

SUBROUTINE NOCHRS (No. 411)
Called from GETJCL, NODES.
This subroutine has 2 entry points.
ENTRY NOCHRS
ENTRY NBCD
Entry NOCHRS computes the number of characters in a character string.
Entry NBCD converts an integer into character form.

SUBROUTINE TOLOWC (No. 412)
Called from GETJCL.
Converts character data to lower case.

SUBROUTINE REGN1 (No. 42)
Called from SINTRU.
This subroutine calls the proper subroutine to get region 1 information for the proper geometry.

SUBROUTINE SFEERE (No. 421)
Called from REGN1.
Reads data to define region 1 of a sphere.

SUBROUTINE CYLNDR (No. 422)
Called from REGN1.
Reads the measurements for a cylindrical tank.
SUBROUTINE RGNGL (No. 423)
Called from RGN1, RGN2T5.
Subroutine to input general information for each region: region width,
temperature, material, number of layers through the region.

SUBROUTINE MATMNU (No. 4231)
Called from RGNGL, RGN2T5, SPLINP.
Displays a menu of materials and prompts the user for a material number
for each defined region.

SUBROUTINE RGN2T5 (No. 43)
Called from SINTRU.
This subroutine prompts the user to obtain input data to define regions
spherical or cylindrical geometry.

SUBROUTINE ULLINP (No. 431)
Called from RGN2T5, ULLGET.
This subroutine prompts the user for ullage information, such as where
the tank and what percent of the tank is filled with liquid.
There are 2 entry points in this subroutine.
ENTRY ULLINP
ENTRY ULLIN2

SUBROUTINE ULLGET (No. 432)
Called from RGN2T5, SPHNDs.
If there is ullage in the tank determine where the ullage is and which nodes are ullage and which are
the ullage is positioned in liquid.

SUBROUTINE CUBIC (No. 4321)
Called from ULLGET.
Finds the real roots of a cubic equation.

SUBROUTINE READHX (No. 433)
Called from RGN2T5.
This subroutine reads in heat exchanger information if there are any. There may be up to 10 heat
exchangers in the model.

SUBROUTINE NODES (No. 44)
Called from SINTRU.
This subroutine calls the proper routines to generate diffusion nodes for the SINDA model. This
routine then generates arithmetic and boundary nodes.

SUBROUTINE SPHNDs (No. 441)
Called from NODES.
Subroutine to generate node data on a sphere.

SUBROUTINE SETUPA (No. 4411)
Called from SPHNDs.
Setup data for arithmetic nodes. Checks for heat exchangers; then puts out arithmetic nodes to
SINDA model file.

SUBROUTINE SPHDIF (No. 4412)
Called from SPHNDs.
Compute diffusion nodes for all nlay layers of a spherical wedge.

SUBROUTINE ULLCHK (No. 44121)
Called from SPHDIF, RADCON, CIRCON.
Checks the type of ullage for region NR and computes where it starts; i.e. when c1='1', at which
theta angle, counting from the south pole, does the ullage start for the current layer LN.

143
When ct='c', is the current layer ullage or not.

**SUBROUTINE CYLND$ (no. 442)**
Called from NODES.
Calls one of the following subroutines to generate nodes.
- **FEND** to generate the nodes for a flat end.
- **SEND** to generate the nodes for a spherical end.
- **FEND** to generate the nodes for an elliptical end.
- **CYLSEC** to generate the nodes in the cylindrical section.

**SUBROUTINE FEND (No. 4421)**
Called from CYLND$.
Subroutine to generate the nodes for a flat end.

**SUBROUTINE SEND (No. 4422)**
Called from CYLND$.
Subroutine to generate the nodes for a spherical end.

**SUBROUTINE FEND (no. 4423)**
Called from CYLND$.
Subroutine to generate nodes for an elliptical end.

**SUBROUTINE CYLSEC (No. 4424)**
Called from CYLND$.
Subroutine to generate nodes in the cylindrical section.

**SUBROUTINE RITNDS (No. 443)**
Called from NODES, SPHDFIF, FEND, SEND, EEND, CYLSEC, SETUPA.
Writes the node lines to the SINDA model file.

**SUBROUTINE SRCDAT (No. 45)**
Called from SINTRU.
This subroutine generates the source data block of the SINDA model.

**SUBROUTINE AREASP(Nol 451)**
Called from SRCDAT, SNBLKS, CIRCON, SPHDFIF, SPHCDS.
Computes areas on a sphere.
The call parameter NAREA determines which type of area.
If NAREA=1, computes radial area, surface areas.
If NAREA=2, computes circumferential area.

**SUBROUTINE CONDRS (No. 46)**
Called from SINTRU.
This subroutine calls on the sphere or cylinder conductor generation routine.

**SUBROUTINE SPHCDS (No. 461)**
Called from CONDRS.
Generates all conductor data for a spherical wedge.

**SUBROUTINE RADCON (No. 4611)**
Called from SPHCDS
Generates radial conductor data for sphere wedge.

**SUBROUTINE SETARY (No. 46111)**
Called from RADCON.
Checks for vapor nodes in conductor data and sets switches NYA and NYB equal to 0 or to 200 to change the property table array numbers for a node. This is only done when NR >= 4 and NLGR=1.
SUBROUTINE CIRCON (No. 4612)
Called from SPHCDS.
Generates circumferential conductor for sphere wedge

SUBROUTINE RITCND (No. 4613)
Called from SPHCDS, RADCON, CIRCON, CYLALL, FCND, SCND, ECND.
Writes conductor cards to SINDA model file, unit 10.

SUBROUTINE CYLCDS (No. 462)
Called from CONDRS.
Calls one of the following subroutines to generate conductors:
FCND to generate the conductors for a flat end.
SCND to generate the conductors for a spherical end.
ECND to generate the conductors for an elliptical end.
CYLALL to generate the conductors for the cylindrical section.

SUBROUTINE HXARR (No. 4621)
Called from CYLCDS.
Generates all conductors that involve a heat exchanger.

SUBROUTINE CYLLA (No. 4622)
Called from CYLCDS.
Generates all conductors in the cylindrical section.

SUBROUTINE FCND (No. 4623)
Called from CYLCDS.
Generates all conductors in a flat end.

SUBROUTINE SCND (No. 4624)
Called from CYLCDS.
Generates all conductors in a spherical end.

SUBROUTINE ECND (No. 4625)
Called from CYLCDS.
Generates all conductors in an elliptical end.

SUBROUTINE SNBLKS (No. 47)
Called from SINTRU.
This subroutine reads SINDA constants data.
Generates the constants, array, execution, Variables 1, Variables 2 and output blocks in the SINDA model and writes these blocks to the model file, unit 10.

SUBROUTINE SPLINP (No. 471)
Called from SNBLKS.
Subroutine to read special input data for analyses where ntyp=1 or 2, and regns(4)=false. This data is fluid data inside the tank where the analysis program is solving the thermo problem and there are no SINDA nodes. The data that is prompted for are:
Liquid temperature
Vapor temperature
Liquid flow rate
Liquid fill level, percent full.

SUBROUTINE AREACY (No. 472)
Called from SNBLKS, FCND, CYLSEC, ECND, SCND.
Computes areas on a cylinder.
The call parameter NAREA determines which type of area. If NAREA=1, compute radial area.
If NAREA=2, compute circumferential area.

145
SUBROUTINE PRPTBL (No. 473)
Called from SNBLKS.
Put property tables into array data block.

SUBROUTINE MATUSR (No. 4731)
Called from PRPTBL.
Subroutine that gives the user the choice of creating his/her own property.

SUBROUTINE INSBT (No. 474)
Called from SNBLKS.
Subroutine to insert the source code of fluid subroutines into the SINDA model. This is done by:
Filedefing the proper unit using DOJCL.
This file will be called 'CRYOSUBS "XCU1"'.
This source file must be LRECL=80, RECFM=F
Then open fortran unit 59 on that file.
Read 59 and write into MODU (unit 10).

SUBROUTINE GEOPLT (No. 48)
Called from SINTRU.
This subroutine controls the calls to the plotting routines to produce geometry plots of the SINDA models. This is not for plots of the analysis output, only the geometry. If the geometry is a sphere this routine calls subroutine PLTSPH. If the geometry is cylindrical subroutine PLTCYL is called.

SUBROUTINE PLTSPH (No. 481)
Called from GEOPLT.
Plots the geometry generated for a sphere.

SUBROUTINE PLTCYL (No. 482)
Called from GEOPLT.
Plots the geometry generated for a cylinder.

SUBROUTINE VMINTR (No. 5)
Called from MAIN.
This subroutine is the entry to execute analysis routines on VM interactively.
This subroutine must be changed whenever a new interactive analysis program is added to the system.
The names of these programs will be added to the array MAINM and the corresponding value in array NSRUNN will be set = 2. For these programs the output will go to both the screen and a disk file named "program output", fortran unit 17.

SUBROUTINE NOSIND (No. 6)
Called from MAIN.
This subroutine sets up the CRAY JCL in a file and then submits it to CRAY to execute an analysis program that is prestored on the CRAY as part of this system.

SUBROUTINE INDAT1 (No. 61)
Called from NOSIND.
Subroutine to get input data for an analysis program. This subroutine has two entry points:
ENTRY INDAT1
ENTRY INDAT2
Entry indatl interrogates the user as to the source of the input data. If the data is on CRAY, write an access to this data in the model file.
Entry INDAT2 is called if the data is on VM or if the data is to be typed in at the terminal. In this case the data is written inline into the model file.

DISSPLA
Called from PLTSPH, PLTCYL.
Plotting package on the LeRC VM computer used to plot sphere and cylinder SINDA models.
APPENDIX E

CryoTran Program Listings

Part I  CRYOTRAN FORTRAN
PROGRAM CRTRAN

READ INPUT DATA, DETERMINE GEOMETRY TYPE.

THE INPUT DATA IS ECHOED TO FORTRAN UNIT INPEKO

FILENAME = CRYOTRAN INPUTEKO

GENERATE A SINDA MODEL ON UNIT MODU. MODE SET BELOW.

INCLUDING CALLS TO SUBROUTINES FROM EXECUTION BLOCK,

VARIABLES 1, VARIABLES 2, AND OUTPUT BLOCKS.

OR GENERATE A RUNSTREAM TO RUN AN ANALYSIS PROGRAM WITHOUT

A SINDA THERMAL ANALYSIS.

A LIBRARY OF SUBROUTINES WILL RESIDE ON CRAY.

THIS LIBRARY WILL CONTAIN ONLY SUBROUTINES, NO MAIN PROGRAMS.

THE MAIN PROGRAM FOR ANY ANALYSIS WILL BE GENERATED EITHER BY

THIS PROGRAM OR BY THE SINDA PREPROCESSOR.

THIS PROGRAM MAY BE ACCESSED AND PUT INTO EXECUTION BY

LINKING TO THE D DISK OF USERLIB CRYOLIB, AND THEN

INVOKING THE VM EXEC 'RUNCryo', AS FOLLOWS:

LINK CRYOLIB 200 NNN RR (NNN MAY BE ANY NO. THE USER
ACCESS NNN M DOES NOT HAVE DEFINED)

RUNCryo (ON THE ACCESS NNN MUST BE THE "M" DISK)

USER MAY EDIT THE SINDA MODEL AT ANY TIME TO TAILOR IT TO A

SPECIFIC NEED; TO ADD A CAPABILITY NOT AVAILABLE IN

THIS PROGRAM; OR TO RUN PARAMETRIC STUDIES.

A FORTRAN CALL TO CLEAR THE SCREEN 'CALL CLEAR' IS USED IN

THIS PROGRAM. THIS ROUTINE IS ON THE AMDahl/VM SYSTEM AT LERC.

THIS ROUTINE, (CLEAR), IS CALLED FROM A SUBROUTINE IN THIS

PROGRAM CALLED CLRS (CLEAR SCREEN).

ON ANOTHER SYSTEM THAT DOES NOT HAVE THIS ROUTINE THE USER
MAY COMMENT OUT THE CALL TO CLEAR IN SUBROUTINE CLRS.

OR ACCESS A SUBSTITUTE ROUTINE.

TO USE AT LERC, USER MUST DO 'FTNLIB' PRIOR

TO LOAD IN ORDER TO ACCESS THE ROUTINE;

OR DO 'ADDLIB FVNLIB', (LERC LOCAL COMMAND).

SEE VM EXEC 'RUNCRYO'

NOTE: CALL SYSCMD ... USED IN SUBROUTINE DOJCL (C03)

AND MAIN (O)

IS A LOCAL LERC SUBROUTINE TO PERFORM VM JCL REQUESTS FROM

INSIDE A FORTRAN PROGRAM.

ON ANOTHER SYSTEM THAT DOES NOT HAVE THIS ROUTINE THE USER
MAY COMMENT OUT THE CALL TO SYSCMD IN SUBROUTINE DOJCL,

OR ACCESS A SUBSTITUTE ROUTINE.

MAIN PROGRAM

CALL MAINPG
CALL PLTDUN
STOP
END
SUBROUTINE MAINPG
COMMON/G_OMTY/ NTYP, NAN, GEOM(2)
COMMON/UNITS/ MODU, INPEKO, ISCRCH, SINDA
COMMON/SUBRTS/ SPLIPT, XCUTI, XCUT2, VBLBL1, VBLBL2, OUTBLK
LOGICAL SPLIPT
LOGICAL SINDA
CHARACTER*1 YN
CHARACTER*6 XCUTI, XCUT2, VBLBL1, VBLBL2, OUTBLK, MAINNM
CHARACTER*12 EXMOD
CHARACTER*20 FNFTFM
CHARACTER*27 CHGN1, CHGN2, CHGN3
CHARACTER*47 RENAME
CHARACTER*1 YN
CHARACTER*6 XCUTI, XCUT2, VBLBL1, VBLBL2, OUTBLK, MAINNM
CHARACTER*12 EXMOD
CHARACTER*20 FNFTFM
CHARACTER*27 CHGN1, CHGN2, CHGN3
CHARACTER*47 RENAME
DATA CHGN1/"RENAME CRYOTRAN INPUTEKO A "/
DATA CHGN2/"RENAME CRYOTRAN MODEL A "/

148
CALL INITAL
CALL INITAL
C
CALL GET ANALYSIS SUBROUTINE (VARIABLES 1 OR STAND ALONE) MENU2 (3)
CALL INITIAL (NRUNON)
EXMOD='SINDA'
IF(NITYF .EQ. 3) THEN
EXMOD(1:16) = XGETI
EXMOD(7:12) = VBLBLI
ENDIF
C
IF(SINDA) THEN
CALL SINTRU
CALL CLEARS
PRINT 2010
PRINT 2011
CLOSE UNIT MODU, REWIND, AND EXIT PREPRO PROGRAM.
PRINT 2012
PRINT 2013, EXMOD
PRINT 2050
PRINT 2060
ELSE
IF(NRUNON .EQ. 2) THEN
MAINNM=XGETI
CALL VMINTR(MAINNM)
PRINT 2020
PRINT 2015
CALL READEL(1, YN)
IF(YN .EQ. 'N') THEN
GO TO 999
ELSE
GO TO 997
ENDIF
ELSE
CALL NOSIND(NRUNON)
ENDIF
C
CALL CLEARS
PRINT 2010
PRINT 2012
ENDIF
ENDFILE MODU
REWIND
C
GO TO TOP OR STOP???
998 PRINT 2015
CALL READEL(1, YN)
997 IF(YN .EQ. 'Y') THEN
CALL CLEARS
PRINT 2016
PRINT 2017
CALL READEL(1, YN)
IF(YN .EQ. 'Y') THEN
PRINT 20181
PRINT 2017
CALL READAL(1,YN)
IF(YN .EQ. 'Y') THEN
PRINT 2019
CALL READAL(3,FNFTFM)
RENAME=CHGN2//FNFTFM
ENDIF
CLOSE UNIT INPEKO, REMIND, CHANGE NAME
ENDFILE INPEKO
RENAME INPEKO
CALL DOJCL(RENAME)
ENDIF
IF(NRUNON .EQ. 1) THEN
CALL CLEARS
PRINT 20182
PRINT 2017
CALL READAL(1,YN)
IF(YN .EQ. 'Y') THEN
PRINT 2019
CALL READAL(3,FNFTFM)
RENAME=CHGN3//FNFTFM
CLOSE UNIT MODU, REMIND, CHANGE NAME
CALL DOJCL(RENAME)
ENDIF
IF(NRUNON .EQ. 2) THEN
CALL CLEARS
PRINT 20183
PRINT 2017
CALL READAL(1,YN)
IF(YN .EQ. 'Y') THEN
PRINT 2019
CALL READAL(3,FNFTFM)
RENAME=CHGN3//FNFTFM
CLOSE UNIT MODU, REMIND, CHANGE NAME
CALL DOJCL(RENAME)
ENDIF
GO TO _
ENDIF
CALL CLEARS
PRINT 2013
PRINT 2012
PRINT 2013, EXMOD
PRINT 2050
PRINT 2060
IF(NTYP .EQ. 3 .AND. NA_ .EQ. 4) PRINT 2061, EXMOD
IF(SINDA) THEN
EXMOD='SINDA MODEL'
PRINT 2062, EXMOD
PRINT 2062, EXMOD
ENDIF
999 RETURN
C
C FORMATS
C
2005 FORMAT(' NOW GET INPUT DATA FOR THE SELECTED ANALYSIS PROGRAM'/CRY01990
1 ' IS THE ANALYSIS INPUT DATA:/' CRY02000
2 ' 1 STORED ON THE CRAY COMPUTER.//' CRY02010
3 ' 2 STORED ON THE VM COMPUTER.//' CRY02020
4 ' 3 TO BE TYPED IN NOW.//' CRY02030
5 ' TYPE 1, 2, OR 3') CRY02040
2010 FORMAT(' END OF CRYOTRAN PREPROCESSOR PROGRAM,') CRY02050
2011 FORMAT(' ON TO ANALYSIS PROGRAM') CRY02060
2012 FORMAT(' THE OUTPUT FILE IS CALLED "CRYOTRAN MODEL",') CRY02070
2013 FORMAT(' THIS "CRYOTRAN MODEL" FILE IS A ',AI2,' MODEL.') CRY02080
2015 FORMAT(' DO YOU WANT TO GO TO BEGINNING OF CRYOTRAN OR QUIT?') CRY02090
1 ' TYPE Y TO GO BACK TO BEGINNING OF CRYOTRAN,' CRY02100

DO YOU WANT TO CHANGE THE NAME OF ANY OF YOUR OUTPUT FILES FROM THIS RUN BEFORE CONTINUING?

TYPE IN Y OR N

CHANGE THE NAME OF THE FILE "CRYOTRAN INPUT"?

CHANGE THE NAME OF THE FILE "CRYOTRAN MODEL"?

CHANGE THE NAME OF THE FILE "PROGRAM OUTPUT"?

TYPE IN THE NEW FILE NAME; FILE TYPE; FILE MODE. YOU MUST TYPE IN ALL THREE PARTS OF NAME FN FT FM

TO SUBMIT THE FILE TO CRAY, ON THE VM SYSTEM TYPE: CRSUBMIT CRYOTRAN MODEL

UPON COMPLETION OF THE CRAY EXECUTION OF THE NEW MODEL, USER MAY PLOT THE RESULTS BY TYPING:

DOECLPLT SOLA

IF USER HAS REQUESTED A GEOMETRY PLOT OF THE MODEL, THE PLOT DATA IS IN FILE NAMED "QMS PLOTDATA"

USER MAY PLOT THESE RESULTS BY TYPING: PLOTQA

SUBROUTINE CLEARS
CALLED FROM VARIOUS ROUTINES
SUBROUTINE TO CLEAR THE TERMINAL SCREEN
THIS ROUTINE IS SYSTEM DEPENDENT; SEE NOTE IN MAIN PROGRAM.
CALL CMSCMD (' VMFCLEAR ' , 16, IRT)
CALL CLEAR
RETURN
END

SUBROUTINE READAL(N,ALF)
CALLED FROM VARIOUS ROUTINES
DO SOME VALIDITY CHECKING
COMMON/UNITS/ MODU, INPEKO, ISCRCH, SINDA
N = 1 ALPHABETIC CHARACTER
N = 2 ALPHABETIC CHARACTERS
N = 3 ALPHABETIC CHARACTERS NO CHECKING FOR Q, AND CONVERT TO LOWER CASE. CALLED VIA ENTRY READLC
COMMON/UNITS/ MODU, INPEKO, ISCRCH, SINDA
CHARACTER* (*) ALF, ALF2
CHARACTER*1 ALFIN(25), ALFIC(25), INTLO, INTHI
CHARACTER*25 ALPHA, ALOWC
EQUIVALENCE (ALPHA,ALFIN(1)), (ALOWC,ALFIC(1))
DATA INTLO/'0'/, INTHI/'9'/
M=-3
GO TO 10
ENTRY READLC(ALF,ALF2)
READ N ALPHABETIC CHARACTERS, NO CHECK FOR Q, CONVERT TO LOWER CASE. M=-3
10 GO TO (100,200,700), M
1 CHARACTER ALPH INPUT
100 READ(5,1001)ALFIN
WRITE(INPEKO,1001) ALFIN(1)
IF(ALFIN(1) .EQ. 'Q') GO TO 999
ALF=ALFIN(1)
200 CONTINUE
C N CHARACTER ALPHABETIC, TEST THAT INPUT IS NOT BLANK
210 READ(5,1001) ALFIN
   IF(ALFIN(1) .EQ. 'Q'.AND. ALFIN(2) .EQ. ' ' .AND. M .NE. 3) THEN
      GO TO 999
   ENDIF
   IF(ALPHA .EQ. ' ') THEN
      PRINT 2001
      GO TO 210
   ENDIF
   WRITE(INPEKO, IO01) ALFIN
   IF(M .EQ. 3) THEN
      CALL tolowc(25,ALPHA,ALOWC)
      ALFIN = ALLOWC
   ENDIF
   RETURN
C ENTRY READIN(INT, LL, LU)
310 READ(5,1001) ALFIN
   IF(ALFIN(1) .EQ. 'Q') GO TO 999
   DO 315 I=1,25
      IF(ALFIN(I) .EQ. ' ') GO TO 315
      IF(ALFIN(I) .LT. INTLO .OR. ALFIN(I) .GT. INTHI) THEN
         PRINT 3001, ALPHA, LL, LU
         PRINT 3000
         GO TO 310
      ENDIF
   315 CONTINUE
   REWIND ISCRCH
   WRITE(ISCRCH,1001) ALFIN
   REWIND ISCRCH
   READ(ISCRCH,*) INT
   IF(INT .LT. LL .OR. INT .GT. LU) THEN
      PRINT _001, INT, LL, LU
      PRINT 3000
      GO TO 310
   ENDIF
   RETURN
C ENTRY READRE(VAL)
410 READ(5,1001) ALFIN
   IF(ALFIN(1) .EQ. 'Q') THEN
      CALL CL:ARS
   ENDIF
   CONTINUE
   RETURN
C READ REAL NUMBER TEST FOR ALPHABETIC CHARACTERS
410 READ(5,1001) ALFIN
   IF(ALFIN(1) .EQ. 'Q') GO TO 999
   DO 415 I=1,25
      IF(ALFIN(I) .EQ. ' ') GO TO 415
      IF(ALFIN(I) .EQ. ' ') GO TO 415
      IF(ALFIN(I) .EQ. ' ' ) GO TO 415
      IF(ALFIN(I) .LT. INTLO .OR. ALFIN(I) .GT. INTHI) THEN
         PRINT 3002,ALPHA, ALFIN(I),I
         PRINT 3000
         GO TO 410
      ENDIF
   415 CONTINUE
   REWIND ISCRCH
   WRITE(ISCRCH,1001) ALFIN
   REWIND ISCRCH
   READ(ISCRCH,*) VAL
   WRITE(INPEKO, ') VAL
   RETURN
999 CALL CL:ARS
   PRINT 2015
   READ(5,1001) ALFIN
   IF(ALFIN(1) .EQ. 'Q') THEN
      RETURN
STOP
ELSE
REWIND MODU
REWIND INPEKO
CALL MAINPG
ENDIF

C FORMAT STATEMENTS

C
1001 FORMAT (25A1)
2001 FORMAT (/*** ERROR*/
  1 ' THE INPUT TYPED IN IS BLANK, IT MUST NOT BE BLANK.'/
  2 ' RETYPE THE LAST INPUT.'/
2015 FORMAT(/*** ERROR*/
  1 ' DO YOU REALLY WANT TO QUIT CRYOTRAN? OR'/
  1 ' TYPE Q TO QUIT CRYOTRAN,'/
  1 ' OR TYPE Y TO GO BACK TO BEGINNING OF CRYOTRAN '
3000 FORMAT (RE-ENTER THIS NUMBER'.$)
3001 FORMAT(/*** ERROR*/
  1 ' INPUT VALUE OUT OF RANGE, INPUT VALUE = ',A25/
  2 ' THIS INPUT VALUE MUST BE AN INTEGER BETWEEN ',I4,
  3 ' AND ',I4)
3002 FORMAT(/*** ERROR*/
  1 ' INPUT VALUE CONTAINS AN ILLEGAL CHARACTER'/
  2 ' THIS INPUT VALUE MUST BE A REAL NUMBER'/
  3 ' THE INPUT VALUE - ',A25/
  4 ' THE ILLEGAL CHARACTER IS ',AI, ' AT POSITION ',I2)
END

C03SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SUBROUTINE DOJCL(COMAND)
  CALLED FROM MAIN (O0) INDAT2 (61) INSERT (474)
  C SUBROUTINE TO EXECUTE VM SYSTEM JCL COMMANDS FROM INSIDE FORTRAN
  C THIS ROUTINE IS SYSTEM DEPENDENT; SEE NOTE IN MAIN PROGRAM.
  C
  COMMON (*) COMAND
  CALL SYSCMD(COMAND,IRC)
  WRITE(6,'JCL COMMAND - IRC-',COMAND,IRC
  RETURN
END

C04SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SUBROUTINE BLHDRS
  CALLED FROM SINTRU (4)
  CALLED FROM NODES (44)
  CALLED FROM CONDRS (46)
  CALLED FROM SNBLKS (47)
  C SUBROUTINE TO READ INFO FOR SINDA MODEL BLOCK HEADERS
  C AND TO WRITE THE SINDA MODEL BLOCK HEADERS, OTHER BLOCK INFO
  C AND END STATEMENTS.
  C
  COMMON /REGION/ NTHETA, NBETAS, BETA, RIN, TVOL, ROUT(9),
  1 REGNS (9), NLAYRS (9), TEMPS (9), THICK (9),
  2 THKLAY (9), MATRLS (9), MATNMS (9), RGNNMS (9)
  COMMON /UNITS/ MODU, INPEKO, ISCRCH, SINDA
  COMMON /TITL / TITLE, TITLE0
  COMMON /GEO1T/ NTP, HNP, GOM(2)
  C
  LOGICAL REGNS, SINDA
  C
  CHARACTER*16 MATNMS
  CHARACTER*25 RGNNMS
  CHARACTER*8 GEOM
  CHARACTER*50 TITLE0
  CHARACTER*80 TITLE
  CHARACTER*40 TITL,TITL2
  CHARACTER BLANK
  C
  EQUIVALENCE (TITL,TITLE), (TITL2,TITLE(41: )
  DATA BLANK/' '/
  C
  RETURN
ENTRY RDTITL
READ TITLE FOR SINDA BLOCK 0
GET THE TITLE LINE
CALL CLEARS
9 PRINT 1998
READ(5,1001,ERR=10,END=10) TITLE
RETURN
10 PRINT 1111
1111 FORMAT(' ERROR MADE IN INPUTTING THE TITLE, TRY AGAIN')
GO TO 9

ENTRY BLOTTL
C BLOCK 0, ID BLOCK
WRITE BLANK CARD, ID BLOCK, NODE BLOCK TITLE CARDS
WRITE(MODU,2001)
WRITE(MODU,20011)
WRITE(MODU,20012) GEOM(NTYP), TITLE
WRITE(MODU,20013) BETA
WRITE(MODU,2002) TITLE1, TITLE2
WRITE(MODU,3000)
RETURN

ENTRY BLITTL
WRITE TITLE FOR BLOCK 1, NODE DATA BLOCK
WRITE(MODU,2101)
WRITE(MODU,2102) MATNMS(9)
RETURN

ENTRY BL2TTL
WRITE TITLE FOR BLOCK 2, SOURCE DATA BLOCK
WRITE(MODU,2201)
RETURN

ENTRY BL3TTL
WRITE TITLE FOR BLOCK 3, CONDUCTOR DATA BLOCK
WRITE (MODU, 2301)
RETURN

ENTRY BL4TTL
WRITE TITLE FOR BLOCK 4, CONSTANTS DATA BLOCK
WRITE(MODU, 2401)
RETURN

ENTRY BL5TTL
WRITE TITLE FOR BLOCK 5, ARRAY DATA BLOCK
WRITE (MODU, 2501)
RETURN

ENTRY BL6TTL
C WRITE TITLE FOR BLOCK 6, EXECUTION DATA BLOCK
C WRITE(MODU, 2601)
RETURN

ENTRY BL7TTL
C WRITE TITLE FOR BLOCK 7, VARIABLES I BLOCK
WRITE(MODU, 2701)
C
C ENTRY BLBTTL
C WRITE TITL FOR BLOCK 8, VARIABLES 2 BLOCK
WRITE(MODU,2801)
RETURN
C ENTRY BLgTTL
C WRITE TITL FOR BLOCK 9, OUTPUT BLOCK
C WRITE (MODU, 2901)
RETURN
C ENTRY BLKEND
C ENTRY TO WRITE END -- FOR END OF BLOCK.
C WRITE(MODU,3000)
RETURN
C ENTRY ENDDAT
C WRITE END OF DATA LINE
C WRITE{MODU,3001)
RETURN
C FORMAT STATEMENTS
C I001 FORMAT (A80)
C 1998 FORMAT(///
C 2001 FORMAT(BOX
C 20011 FORMAT ('C' ,6X,
C 20012 FORMAT (' C' , 6X,
C 20013 FORMAT (' C' , 6X,
C 2002 FORMAT (7X, ' BCD
C 2101 FORMAT (7X, ' BCD
C 210_ FORMAT (TX, 'REM
C 2201 FORMAT (7×,'BCD
C 2301 FORMAT (7X,'BCD
C 2401 FORMAT(VX,'BCD
C 2501 FORMAT(7X, 'BCD
C 2601 FORMAT(7X,
C 2701 FORMAT(7X, 'BCD
C 2801 FORMAT(7X,
C 2901 FORMAT(7X, 'BCD
C NOW A TITLE FOR THIS PROBLEM.'//
The TITLE LINE MAY BE
TO 80 CHARACTERS LONG.'/
TYPE IN THE TITLE.')
DIMENSIONS ARE IN
(IN.);
SOURCE DATA;
CONDUCTOR DATA;
CONSTANTS DATA;
ARRAY DATA;
EXECUTION;
VARIABLES 1;
VARIABLES 2;
OUTPUT CALLS;
FOMP.MAT(7X, 'END')
3END
OF DATA')
COMMON/TITL/
COMMON/GEOMTY/
COMMON/UNITS/
COMMON/REGION/
COMMON/STUFF/
1 COMMON/ULLAGE/
2 COMMON/HTXGRS/

SUBROUTINE INITIAL
CALLED FROM MAIN (00)
C SUBROUTINE TO INITIALIZE COMMON DATA BLOCKS
COMMON/TITL / TITLE,TITLED
COMMON/GEOMTY/ NTYP, NAN, GEOM(2)
COMMON/UNITS/ MODU, INPEKO, ISCRCH, SINDA
COMMON/REGION/ NTYP, NTHETA, NBETAS, BETA, RIN, TVOL, ROUT(9),
REGNS(9), NLAYRS(9), THICK(9),
1 TEMPS(9), Thickness(9), THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
COMMON/STUFF/ NHTT, PI, CONVY, CONVR, THETA0, DTHETA, NBASOS,
1 BNCOF(2)
COMMON/ULLAGE/ NLU4, NLUI5, NTHU41, RIMMHH, PCTFUL, RADUIC, TVULFT,
1 CT, LG(3), LQVAP(3)
COMMON /HTXGRS/ NHX, HXTEM(10), NANT(10), NNX(10), NLFX(10),

CRY04910
CRY04920
CRY04930
CRY04940
CRY04950
CRY04960
CRY04970
CRY04980
CRY04990
CRY05000
CRY05010
CRY05020
CRY05030
CRY05040
CRY05050
CRY05060
CRY05070
CRY05080
CRY05090
CRY05100
CRY05110
CRY05120
CRY05130
CRY05140
CRY05150
CRY05160
CRY05170
CRY05180
CRY05190
CRY05200
CRY05210
CRY05220
CRY05230
CRY05240
CRY05250
CRY05260
CRY05270
CRY05280
CRY05290
CRY05300
CRY05310
CRY05320
CRY05330
CRY05340
CRY05350
CRY05360
CRY05370
CRY05380
CRY05390
CRY05400
CRY05410
CRY05420
CRY05430
CRY05440
CRY05450
CRY05460
CRY05470
CRY05480
CRY05490
CRY05500
CRY05510
CRY05520
CRY05530
CRY05540
CRY05550
CRY05560
CRY05570
CRY05580
CRY05590
CRY05600

155
1 NTHHX(10), LNGTHX(10)
COMMON/ SUBRTS/ SPLIT, XCUT1, XCUT2, VBLBL1, VBLBL2, OUTBLK

C
LOGICAL REGNS, SINDA
LOGICAL SPLIT
C
CHARACTER*1 CT, LG
CHARACTER*6 LIQVAP
CHARACTER*6 XCUT1, XCUT2, VBLBL1, VBLBL2, OUTBLK
CHARACTER*8 GEOM
CHARACTER*16 MATNMS
CHARACTER*25 RGNNMS
CHARACTER*50 TITLE
CHARACTER*80 TITLEO
C
C SET UNIT NUMBER FOR MODU, MODEL OUTPUT UNIT.
MODU= 10
INPEKO= 9
ISCRCH= 35
C
C USE UNIT ISCRCH, (35) AS A SCRATCH FILE FOR INPUT TESTING
C AND FORMAT CONVERSION IN SUBROUTINE READAL.
C
C UNIT NO. 36 IS USED IN SUBROUTINES INSERT AND INDA1.
C THIS UNIT IS USED TO READ DATA FROM A VM FILE.
C USER WILL BE ASKED THE NAME OF THE FILE, PROGRAM THEN
C DOES A FILEDEV ON THAT FILE, THEN OPENS THE FILE AS UNIT 36.
C THE CLOSE IS DONE WHEN THE READ IS COMPLETED.
C
PI= 3.14159265
TITL0= ' 
TITLE = ' 
NTHETA= 0
THETAO= PI/2.
NBETAS= 1
BETA= 1.
RIN= 0.
DO 10 I= 1, 9
REGNS(I)=.FALSE.
NLAYRS(I)= 0
MATNMS(I)= 0
ROUT(I)= 0.0
THICK(I)= 0.0
TEMS(I)= 0.0
MATNMS(I)= 0.
RGNNMS(I)= 0.
10 CONTINUE
C
BNCOEF(1)= 0.0
BNCOEF(2)= 0.0
C
C INITIALIZE REGION NAMES IN ARRAY RGNNMS(I)
RGNNMS(1)= 'TANKWALL'
RGNNMS(2)= 'OUTSIDE LAYER 1'
RGNNMS(3)= 'OUTSIDE LAYER 2'
RGNNMS(4)= 'INSIDE TANK AT WALL'
RGNNMS(5)= 'INSIDE TANK AT CENTER'
C
C INITIALIZE ULLAGE VARIABLES
C
LG(1)= 'L'
LG(2)= ' 
LG(3)= 'G'
LIQVAP(1)= 'LIQUID'
LIQVAP(2)= 'LIQUID'
LIQVAP(3)= 'VAPOR'
NLUL4= 0
NLUL5= 0
NTHU41= 0
PCTFUL= 0.0
C
CRY05610
CRY05620
CRY05630
CRY05640
CRY05650
CRY05660
CRY05670
CRY05680
CRY05690
CRY05700
CRY05710
CRY05720
CRY05730
CRY05740
CRY05750
CRY05760
CRY05770
CRY05780
CRY05790
CRY05800
CRY05810
CRY05820
CRY05830
CRY05840
CRY05850
CRY05860
CRY05870
CRY05880
CRY05890
CRY05900
CRY05910
CRY05920
CRY05930
CRY05940
CRY05950
CRY05960
CRY05970
CRY05980
CRY05990
CRY06000
CRY06010
CRY06020
CRY06030
CRY06040
CRY06050
CRY06060
CRY06070
CRY06080
CRY06090
CRY06100
CRY06110
CRY06120
CRY06130
CRY06140
CRY06150
CRY06160
CRY06170
CRY06180
CRY06190
CRY06200
CRY06210
CRY06220
CRY06230
CRY06240
CRY06250
CRY06260
CRY06270
CRY06280
CRY06290
CRY06300
CT-''
C INITIALIZE HEAT EXCHANGER COMMON BLOCK TO 0.0
C
NMX=0
DO 30 I=1,10
HXTEMPl)=0.0
NRX(I) = 0
NLX(I) = 0
NRHMX(I) = 0
LNGTHX(I) = 0
30 CONTINUE
C INITIALIZE SUBROUTINE NAMES COMMON BLOCK TO BLANKS
C
XCUT1= '
XCUT2= '
VBBL1= '
VBBL2= '
OUTBLK= '
C
GEOM(1)='SPHERE'
GEOM(2)='CYLINDER'
C RETURN
END
C CRY06310
CRY06320
CRY06330
CRY06340
CRY06350
CRY06360
CRY06370
CRY06380
CRY06390
CRY06400
CRY06410
CRY06420
CRY06430
CRY06440
CRY06450
CRY06460
CRY06470
CRY06480
CRY06490
CRY06500
CRY06510
CRY06520
CRY06530
CRY06540
CRY06550
CRY06560
CRY06570
CRY06580
SUBROUTINE MENU
CALLED FROM MAIN (00)
C THIS SUBROUTINE MAY NEED TO BE CHANGED AS
C NEW CAPABILITIES ARE ADDED TO THE PROGRAM.
C MAIN MENU -- MENU FOR PROBLEM TYPE -- CALLED FROM MAIN
C
COMMON/UNITS/ MODU, INPEKO, ISCRLCH, SINDA
COMMON/GEOMTY/ NTYP,NAN, GEOM(2)
C CHARACTER*8 GEOM
C LOGICAL SINDA, LTEST
C
DATA NTYPES/3/

C 100 CALL CLS
PRINT 2000
200 PRINT 2001
PRINT 2002
PRINT 2003
PRINT 2004
CALL READIN(NTYP, I, NTYPES)
SINDA=.TRUE.
IF(NTYP .EQ. 3) SINDA=.FALSE.
RETURN
C FORMATS
2000 FORMAT(/**/ WELCOME TO CRYOTRAN/**/
1 ' YOU WILL BE PROMPTED FOR ALL NECESSARY INPUT.'/
2 ' READ THE INSTRUCTIONS CAREFULLY.'/
3 ' TYPE IN THE INPUT DATA CAREFULLY TO AVOID TROUBLE.'/
4 ' YOU MAY QUIT THE PROGRAM AT ANY INPUT PROMPT BY',
5 ' TYING A "Q" (QUIT)' )
2001 FORMAT(/**/ ENTER THE NUMBER FOR THE DESIRED PROBLEM TYPE'/
1 ' THE PROBLEM TYPES ARE AS FOLLOWS:' )
2002 FORMAT(/**/ 1 - THERMO/ THERMAL SINDA ANALYSIS ON A SPHERE,' )
2003 FORMAT(/**/ 2 - THERMO/ THERMAL SINDA ANALYSIS ON A CYLINDER,' )
2004 FORMAT(/**/ 3 - RUN A PRESTORED ANALYSIS PROGRAM' )
2050 FORMAT(***** ERROR, THE PROBLEM TYPE YOU INPUT IS OUT OF RANGE' )
1 ' ***** PROBLEM TYPE MUST BE FROM 1 TO ',I3/
2 ' ***** RE-ENTER PROBLEM TYPE *****' )
END
C CRY06590
CRY06600
CRY06610
CRY06620
CRY06630
CRY06640
CRY06650
CRY06660
CRY06670
CRY06680
CRY06690
CRY06700
CRY06710
CRY06720
CRY06730
CRY06740
CRY06750
CRY06760
CRY06770
CRY06780
CRY06790
CRY06800
CRY06810
CRY06820
CRY06830
CRY06840
CRY06850
CRY06860
CRY06870
CRY06880
CRY06890
CRY06900
CRY06910
CRY06920
CRY06930
CRY06940
CRY06950
CRY06960
CRY06970
CRY06980
CRY06990
CRY07000
SUBROUTINE MENU2(NRUNON) CALLED FROM MAIN (OO) THIS SUBROUTINE WILL NEED TO BE CHANGED AS NEW CAPABILITIES ARE ADDED TO THE PROGRAM. MENU 2 ANALYSIS SUBROUTINES FOR NTYP, (FROM MENU1), = 1,2; A SINDA MODEL WILL BE GENERATED. NTP= 1 (SPHERE) NTP= 2 (CYLINDER) FOR NTYP = 3 NO SINDA MODEL GENERATED BY THIS PROGRAM. THE CURRENT ANALYSIS PROGRAMS ARE: ANAL1 OR ANALTS = 1 A SINDA MODEL OF A 1 RADIAN WEDGE. UP TO 5 REGIONS MAY BE DEFINED. REGIONS 4 AND 5 ARE USUALLY LIQUID OR VAPOR AND ARE FULLY MODALIZED WITH SINDA NODES. CURRENTLY THESE REGIONS HAVE CONDUCTION CONNECTORS, BUT CONVECTION CONNECTORS WILL BE ADDED SO THAT ONE MAY HAVE CONDUCTION ONLY, CONVECTION ONLY, OR BOTH CONDUCTION AND CONVECTION IN REGIONS 4 AND 5. ANAL1 OR ANALTS = 2 A SINDA MODEL OF A 1 RADIAN WEDGE. ONLY REGIONS 1, 2, AND 3 WILL BE MODALIZED WITH SINDA NODES. REGIONS 4 AND 5 ARE NOT MODELED WITH SINDA NODES, BUT ARE MODELED BY ANALYTICAL SUBROUTINES. THESE SUBROUTINES ARE CALLED FROM THE EXECUTION AND VARIABLES 1 AND 2 BLOCKS IN SINDA. THE WRITER OF THESE ANALYTICAL ROUTINES IS SUPPLIED WITH THE FOLLOWING COMMON BLOCKS AND ARRAYS. COMMON/USER1/ NTHETA, NBETAS, NTUNIT, BETA, RIN, TNKVOL COMMON/USER2/ TIMEO, DTIMEU, FFLOW, TLIQ, TGAS COMMON/INSA/ SAIN (NN), INSIDE TANK SURFACE AREA, INPUT COMMON/OUTSA/ SAROUT (NN), OUTSIDE SURFACE AREA, INPUT COMMON/SURFT/ TSURF (NN), INSIDE TANK SURFACE TEMP, INPUT COMMON/BNDYT/ TBDY (NN), TANK LIQ OR VAP TEMP, OUTPUT COMMON/HTRCO/ HCOEF (NN), HEAT TR COEFF, NOT SURE NEEDED COMMON/SURFG/ GSURF (NN), TANK TO SURF G VALUE, OUTPUT COMMON/SURFQ/ QSURF (NN), INSIDE SURF Q, OUTPUT WHERE NN IS THE DIMENSION NTHETA PRESENTLY THE PROGRAM IS SETUP TO HANDLE UP TO 15 ANALYSIS PROGRAMS (99) FOR EACH OF THE SIX (6) GEOMETRIES BELOW NTYP=1 (SPHERE) ANALTS - NAN= 1-15 - SPHERE MODELED BY WEDGES, RADIAL MESH, 2D ANALTS - NAM=16-30 - SPHERE MODELED BY WEDGES, RADIAL MESH, 3D ANALTS - NAM=31-45 - SPHERE MODELED BY WEDGES, RECTANGULAR MESH, 2D ANALTS - NAM=46-60 - SPHERE MODELED BY WEDGES, RECTANGULAR MESH, 3D NTYP=2 (CYLINDER) ANALTS - NAM= 1-15 - CYLINDER WEDGE MODEL, RADIAL MESH, 2D ANALTS - NAM=16-30 - CYLINDER WEDGE MODEL, RADIAL MESH, 3D ANALTS -- A CANNED ANALYSIS PROGRAM WITHOUT SINDA, -- ARE CYLINDER MODELS, HAVING A MESH OF WEDGES RADially AND TOP TO BOTTOM, A 3D MODEL WOULD BE A SERIES OF WEDGES CIRCUMFERENTIALLY WHERE USER CAN SPECIFY ANY COMBINATIONS OF SPHERICAL ENDS SORTZ ELLIPTICAL ENDS FLAT ENDS OPEN ENDS THE GEOMETRY MAY BE DEFINED BY AS MANY AS 5 REGIONS AS FOLLOWS: REGION 1. TANK WALL 2. A LAYER ON OUTSIDE OF TANK WALL; E.G. INSULATION 3. A 2ND LAYER OUTSIDE, ON TOP OF REGION 2.
4. The first layer inside of the tank, adjacent to the tank wall. (See description below).

5. The 2nd layer inside of the tank, measured from layer 4 toward the center of the tank. (See description below)

The inside of the tank is determined by the analysis routine. Some analysis routines will have the interior of the tank nodalized with SINDA nodes, some will not. This is specified by the variable REG45. If REG45 = TRUE, the interior will be nodalized; if REG45 = FALSE, the interior will not be nodalized and the thermodynamics of the interior of the tank will be completely managed by the analysis subroutines.

When the inside of the tank is defined (nodalized), it may be 1 or 2 regions. These are defined as regions 4 and 5. Region 5 is optional. Region 4 may be defined alone or both regions 4 and 5 may be defined. This may be used to define 2 different materials inside of the tank, or, in order to have two mesh spacings of a single material inside of the tank in the radial direction.

REG45 is a logical variable such that,

REG45(I,J) = TRUE if the corresponding ANALT(J)(I) is an analysis routine that wants the inside of the tank (regions 4 & 5), to be nodalized with SINDA nodes.

REG45(I,J) = FALSE if no SINDA nodes are needed for the corresponding ANALT(J)(I).

Global variables defined in common statements

COMMON/UNITS/MODU, INPEKO, ISCRCH, SINDA
COMMON/TITLE/TITLE, TITLE0
COMMON/GEOM/NHTYP, NAM, GEOM(2)
COMMON/REGION/NTHETA, NBETAS, BETA, RIN, TVOL, ROUT(9), 1
REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
2 THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
COMMON/SUBRTS/SPLIFT, XCDTI, XCUT2, VBLBL1, VBLBL2, OUTBLK

Logical SPLIFT, LOGICAL REGNS

CHARACTER*8 GEOM
CHARACTER*50 TITLE0
CHARACTER*80 TITLE
CHARACTER*16 MATNMS
CHARACTER*25 RGNNMS
CHARACTER*6 XCBTI, XCUT2, VBLBL1, VBLBL2, OUTBLK

Local variables

LOGICAL SINDA
LOGICAL REG45(15,6)
LOGICAL SPECIN(15,2)

CHARACTER*50 ANALT1(15), ANALT2(15), ANALT3(15), ANALT4(15)
CHARACTER*50 ANALB5(15), ANALB6(15), ANALNS(15)
CHARACTER*6 EXEC1(15,6), EXEC2(15,6), VBL1(15,6),
1 VBL2(15,6), CUT(15,6), MAINNM(15)
CHARACTER*1 INF

DIMENSION NRSUM(15)

NTYP will be = 1 (SPHERE), 2 (CYLINDER) or 3 (NO SINDA)

Combination of variables (NTYP and NAM) will determine the geometry and mesh to be generated.

Data for NTYP=1, SPHERE

DATA ANALT1/2D WEDGE WITH INSIDE OF TANK NODALIZED',
2 '2D WEDGE SHELL - NO NODES INSIDE OF TANK',
3 '2D WEDGE SHELL - THICK WALL FILL ANALYSIS',
C
DATA ANALT2/ 15* /  
DATA ANALT3/ 15* /  
DATA ANALT4/ 15* /  
DATA NALT3/ , NALT2/0/, NALT4/0/  
DATA (SPECIN(I,1),I=1,15)/.FALSE.,.FALSE.,.TRUE., 12* .FALSE./
C
DATA FOR NTYP=2, CYLINDER
DATA ANALT5/ '2D WEDGE WITH INSIDE OF TANK NODALIZED',  
G 13* /  
DATA ANALT6/ 15'' /  
DATA NALT5/2/, NALT6/0/  
DATA (SPECIN(I,2},I=1,15)/ 15*.FALSE./
C
DATA STATEMENT FOR REGIONS 4 AND 5, REG45(I,N),N-I,2,3,4 (SPHERE)  
REG45(I,N),N-5,6 (CYLINDER)  
C IF REGIONS 4/5 ARE NODALIZED WITH SINDA NODES, REG45=TRUE  
C IF REGIONS 4/5 ARE NOT NODALIZED WITH SINDA NODES, REG45=FALSE  
DATA REG45 / .TRUE., .FALSE., .FALSE., 12*.FALSE.,  
2 15*.TRUE.,  
3 15*.FALSE.,  
4 15*.FALSE.,  
5 .TRUE., .FALSE., 13*.FALSE.,  
6 15*.FALSE./
C
NAMES OF ANALYSIS SUBROUTINES FOR SINDA RUNS TO BE CALLED FROM  
EXECUTION, VARIABLES, AND OUTPUT BLOCKS  
THESE ROUTINES ARE IN FILES ON THE VM COMPUTER AND ARE COPIED  
INTO THE BLOCKS (INTO THE MODEL) BY THE SYSTEM. THEY WILL THEN  
BE COMPILED WITH THE GENERATED SINDA ROUTINES AND EXECUTED.  
THEY ARE IN THE MODEL SO THE USER MAY MODIFY THEM IF DESIRED,  
DATA EXEC1/ ' ', 'THWSE1', 12'' ,  
2 15*' ,  
3 15*' ,  
4 15*' ,  
5 ' ' , ' ', 13*' ,  
6 15*' /  
C
DATA EXEC2/ ' ', 'THWSE2', 12'' ,  
2 15*' ,  
3 15*' ,  
4 15*' ,  
5 15*' ,  
6 15*' /  
C
DATA VBL1 / ' ', 'THWSV1', 12'' ,  
2 15*' ,  
3 15*' ,  
4 15*' ,  
5 ' ' , ' ', 13*' ,  
6 15*' /  
C
DATA VBL2 / ' ', 'THWSV2', 12'' ,  
2 15*' ,  
3 15*' ,  
4 15*' ,  
5 ' ' , ' ', 13*' ,  
6 15*' /  
C
DATA OUT / ' ', 'THWSOU', 12'' ,  
2 15*' ,  
3 15*' ,  
4 15*' ,  
5 ' ' , ' ', 13*' ,  
6 15*' /
DATA STATEMENTS THAT NEED TO BE CHANGED WHEN A NEW PROGRAM OF NTYP = 3 IS ADDED TO THE SYSTEM. THAT IS A PROGRAM PRESTORED ON CRAY OR VM.

DATA FOR NTYP = 3, SPECIAL PROGRAMS

DATA ANALNS/NOVENT FILL', 'CHILL TO TEMP', 'TARGET FOR NVFILL', 1 'SOLA-ECLIPSE', 'CSAM', 10*/
DATA ANALNS/5/

NAMES OF NOSINDA ANALYSIS 'MAIN' SUBROUTINES
AND WHICH COMPUTER THEY ARE DESIGNED TO RUN ON.

IF NSFUUM = 1 RUN ON CRAY
2 RUN ON VM IN INTERACTIVE MODE,
3 RUN ON VM IN BATCH MODE,

DATA MAINNM/'NVFILL', 'CHILL', 'TARGET',
2 'SOLECl', 'CRCSAM',
3 10*/
DATA NSFUUM/2, 2, 1, 1, 10*

CALL CLEARS SPLIPT=.FALSE.
PRINT 2001
PRINT 2002
IF(SINDA) THEN
IF(NTYP .EQ. 1) THEN
IF(NALT1 .GT. 0) PRINT 2003, (I ,ANALT1(I),I-I,NALT1)
IF(NALT2 .GT. 0) PRINT 2003, (I+15,ANALT2(I),I-I,NALT2)
IF(NALT3 .GT. 0) PRINT 2003, (I+30,ANALT3(I),I-I,NALT3)
IF(NALT4 .GT. 0) PRINT 2003, (I+45,ANALT4(I),I-I,NALT4)
ENDIF
IF(NTYP .EQ. 2) THEN
IF(NALT5 .GT. 0) PRINT 2003, (I+5,ANALT5(I),I-I,NALT5)
IF(NALT6 .GT. 0) PRINT 2003, (I+15,ANALT6(I),I-I,NALT6)
ENDIF
ELSE
PRINT 2003, (I ,ANALNS(I),I-I,NALNS)
ENDIF
NRUNON=O
CALL READIN(NAN, I,60)
FOR NTYP = 1 SPHERE ALL MODELS ARE WEDGES
MESHING RAD = RADIALY RECT = RECTANGULAR
2D-RAD 3D-RAD 2D-RECT 3D-RECT
NAN = 1,2, ..., 15;16, ..., 30;31, ..., 45;46, ..., 60; ...
NC WILL BE = 0, 1, 2, 3
NANAL = 1, 2, 3, 4
NPROG = 1,2, ..., 15
FOR NTYP = 2 CYLINDER ALL MODELS ARE WEDGES
2D-RAD 3D-RAD
NAN = 1,2, ..., 15
NC WILL BE = 0, 1
NANAL = 1, 2
NPROG = 1,2, ..., 15
NC=NAN/16
NANAL=NC+1
IF(NTYP .EQ. 2) NANAL=NANAL+4
NPROG=NAN-NC+15
PUT ANALTNAN INTO TITLE0
IF(NANAL .EQ. 1) TITLE0=ANALT1(NPROG)
IF(NANAL .EQ. 2) TITLE0=ANALT2(NPROG)
IF (NANAL .EQ. 3) TITLED-ANALT3(NPROG)
IF (NANAL .EQ. 4) TITLED-ANALT4(NPROG)
IF (NANAL .EQ. 5) TITLED-ANALT5(NPROG)
IF (NANAL .EQ. 6) TITLED-ANALT6(NPROG)

C SET BETA=1 RADIANS AND NBETAS=1 FOR VERSION 1.0 OF PROGRAM

IF (NC .EQ. 0) THEN
  BETA=1.0
  NBETAS=1
ENDIF

CALL GET THE TITLE LINE 'TITL' (04)
CALL GET CRAY JCL INFO 'GETJCL (41)
CALL WRITE CRAYJCL TO UNIT MODU 'RITJCL (41)

C FORMAT STATEMENTS
1001 FORMAT(A1)
2001 FORMAT(///' CHOOSE THE ANALYSIS PROGRAM YOU WISH TO USE.')
2002 FORMAT(' TYP_1 IN THE NUMBER OF THE DESIRED ANALYSIS.')
2003 FORMAT(1b, 4x, ASO)
END

COMMON/CRAYJCL/ NTYP, NAN, GEOM(2)
CALL CONI(45)
CALL SNLKS

CRY09810
CRY09820
CRY09830
CRY09840
CRY09850
CRY09860
CRY09870
CRY09880
CRY09890
CRY09900
CRY09910
CRY09920
CRY09930
CRY09940
CRY09950
CRY09960
CRY09970
CRY09980
CRY09990
CRY10000
CRY10010
CRY10020
CRY10030
CRY10040
CRY10050
CRY10060
CRY10070
CRY10080
CRY10090
CRY10100
CRY10110
CRY10120
CRY10130
CRY10140
CRY10150
CRY10160
CRY10170
CRY10180
CRY10190
CRY10200
CRY10210
CRY10220
CRY10230
CRY10240
CRY10250
CRY10260
CRY10270
CRY10280
CRY10290
CRY10300
CRY10310
CRY10320
CRY10330
CRY10340
CRY10350
CRY10360
CRY10370
CRY10380
CRY10390
CRY10400
CRY10410
CRY10420
CRY10430
CRY10440
CRY10450
CRY10460
CRY10470
CRY10480
CRY10490
CRY10500
SUBROUTINE GETJCL(UNICO$)

CALLED FROM SINTRU (4)

C SUBROUTINE TO GET JCL INFO FROM USER AND THEN TO WRITE THE
C JCL AS FILE 1 ON UNIT 10 (MODEL FILE) TO EXECUTE PROBLEM.
C
C GET CRAY UID, ETC. TO GENERATE CRAY JCL FILE
C
C IF A DIFFERENT COMPUTER SYSTEM IS USED THIS SUBROUTINE
C MUST BE CHANGED TO REFLECT THE PROPER JCL OF THE SYSTEM USED.
C
C THIS ROUTINE HAS 2 ENTRY POINTS,
C THE DATA THEN CAN REMAIN LOCAL.
C
COMMON/UNITS/ MODU, NPEKO, ISCRCH, SINDA
COMMON/SUBRTS/ SPLIP',, XCUTI,XCUT2,VBLBL1,VBLBL2,OUTBLK
C
LOGICAL SPLIPT
LOGICAL SINDA
LOGICAL UNICOS
C
CHARACTER*6 XCUTI, XCUT2, VBLBL1, VBLBL2, OUTBLK
CHARACTER*15 CRAUID, CRAAPW, JOBNAM
CHARACTER*15 UIDLC, APWLC, JNAMLC
CHARACTER*2 NCBCD
CHARACTER*1 YN
CHARACTER*48 FMTJOB, FMTACC
C
DATA FMTJOB/'(''JOB, JN-''',A7,'',T-''',IE,'',MFL-''',IT,''.'') '/
DATA FMTACC/'(''ACCOUNT, AC-''',A15,'',APW-''',A15,''.'') '/
C
101 CALL CLEAR (YN)
100 PRINT 2000
106 CALL READAL (YN)
105 IF (YN .EQ. 'U') UNICOS=.TRUE.
102 CALL CLEAR
101 PRINT 2001
111 CALL READLC(CRAUID, UIDLC)
115 CALL READLC(CRAAPW, /PWLC)
120 CALL READIN(KRAMFL,1,4500000)
125 IF (SINDA) THEN
120 PRINT 2004
124 CALL READIN(KRAMFL,1,4500000)
130 ENDIF
135 IF (KRAMFL .LT. 150000) KRAMFL=150000
140 PRINT 2005
145 CALL READLC(JOBNAM, JNAMLC)
150 CALL CLEAR
155 IF (UNICOS) THEN
150 PRINT 1996, UIDLC, APWLC, KRAMFL, JNAMLC
160 ELSE
165 PRINT 1996, CRAUID, CRAAPW, KRAMFL, JOBNAM
170 ENDIF
175 CALL READAL(YN)
180 IF (YN .EQ. 'N') GO TO 101
185 RETURN
ENTRY TO WRITE THE FIRST PART OF CRAY JCL TO UNIT 10, FILE 1

ENTRY RITJCL
REWRIND MODU

ENTRY RITJCL(NINPD)1
ENTRY RITJCL2
ENTRY RITJCL3(NINPD,FILNAM)
ENTRY RITJCL4(NINPD)

END OF CRAY COS FILE 1 (JCL) GENERATION.
ENTRY RITJCL
ENTRY RITJCL(NINPD)2
ENTRY RITJCL3(NINPD)
IF(SINDA) THEN
  WRITE(MODU,3020)
  WRITE(MODU,3021)
ELSE
  IF(NINPD .GE. 2) THEN
    WRITE(MODU,3020)
  ENDIF
  CALL TOLOWC(6,XCUTI,FILNLC)
  WRITE(MODU,3023) FILNLC
ENDIF
WRITE(MODU,3022)

END OF CP_Y UNICOS JCL GENERATION FOR SINDA MODEL.
ENDIF
RETURN

FORMAT STATEMENTS
1001 FORMAT(A15)
  1996 FORMAT(///' THE CRAY JCL THAT WAS INPUT IS AS FOLLOWS://
                1 ' USERID = ',A15/
                2 ' PASSWORD = ',A15/
                3 ' CPU TIME REQUEST = ',19, ' SECS./
                4 ' MEMORY REQUEST = ',19,' words'/
                5 ' JOB NAME = ',A15/
                6 ' ARE THESE ALL CORRECT? TYPE Y OR N,'
                7 ' OR Q TO QUIT')
  2000 FORMAT(///' THIS TASK IS BEING SET UP FOR THE CRAY,'/
                1 ' NOW INPUT NECESSARY CRAY INFO./')
  2001 FORMAT(/' TYPE IN YOUR CRAY USERID.')
  2002 FORMAT(/' TYPE IN YOUR CRAY PASSWORD.')
  2003 FORMAT(/' TYPE IN NO. OF CRAY CPU SECONDS TO BE USED.'/
                1 ' IF NUMBER OF SECONDS REQUESTED IS < 10, 60 WILL BE
                2 USED.')
  2004 FORMAT(/' TYPE AMOUNT OF CRAY MEMORY TO BE REQUESTED,'/
                1 ' IF AMOUNT REQUESTED IS < 1,500,000, 1,500,000 WILL BE
                2 USED.')
  2005 FORMAT(/' WHICH CRAY SYSTEM COS OR UNICOS/'
                1 ' TYPE IN C OR U')
C FORMAT STATEMENTS TO GENERATE CRAY COS JCL
  3003 FORMAT( 'ACCESS, DN=CRYOLIB, PDN=CRTLIB, ID=CFTO, OWN=CRTOLIB.' )
  3004 FORMAT( 'ACCESS, DN=PROCG, PDN=SINDA, ID=SINDA, OWN=TXCRAY.' )
  3005 FORMAT( 'RUNPRE.' '/RUNEXEC.' )
  3006 FORMAT( '*************************************************/
                1 ' **
                2 ' NORMAL JOB TERMINATION
                3 ' **
                4 ' *****************************************************
                5 ' EXIT.' )
  3007 FORMAT( '***************************************************/
                1 ' **
                2 ' JOB BOMBED!!!!!!!
                3 ' **
                4 ' *****************************************************
                5 ' DUMPJOB. '/
                6 ' DEBUG. ')
  3008 FORMAT( 'EOF.' )
C FORMAT STATEMENTS TO GENERATE CRAY UNICOS JCL
  3010 FORMAT('# USER=',A15,3X,'#PW=',A15)
  3011 FORMAT('# QSUB -r ',A15, ' # jobname')
  3012 FORMAT('# QSUB -eo # Combine error and',
                1 'standard output')
  3013 FORMAT('# QSUB -T ',19, ' # CPU time')
  3014 FORMAT('# QSUB -lm ',F4.1, ' # Memory requested')
  3015 FORMAT('# $ # End NQS statements/'
                1 'set -x # set echo/'
                2 'ja ')
  3016 FORMAT('cat > mocal < EOF # SINDA MODEL TO FOLLOW')
30161 FORMAT('cat > model << EOF # DATA FROM VM, FN FT FM- ',A15) CRY12610
30162 FORMAT('cat > model << EOF # DATA FROM THE TERMINAL') CRY12620
3017 FORMAT('cat HOME',A15, ' >> model # DATA FROM CRAY, FILENAME- ',A15) CRY12630
1 A15) CRY12640
3018 FORMAT('### This file (.mod), was generated by CRYOTRAN.') CRY12650
3020 FORMAT('EOF') CRY12660
3021 FORMAT('cosinda model}') CRY12670
3022 FORMAT('/space/cryollb/',A15,' model') CRY12680
3023 FORMAT('Ja -self # GET ACCOUNTING INFO') CRY12690
3024 FORMAT(EOF) CRY12700
3025 FORMAT('cosinda model}') CRY12710
3026 FORMAT('Ja -self # GET ACCOUNTING INFO') CRY12720
3027 FORMAT('/space/cryollb/',A15,' model') CRY12730
3028 FORMAT('Ja -self # GET ACCOUNTING INFO') CRY12740
3029 FORMAT(EOF) CRY12750
3030 FORMAT('cosinda model}') CRY12760
3031 FORMAT('Ja -self # GET ACCOUNTING INFO') CRY12770
3032 FORMAT('/space/cryollb/',A15,' model') CRY12780
3033 FORMAT('Ja -self # GET ACCOUNTING INFO') CRY12790
3034 FORMAT(EOF) CRY12800
3035 FORMAT('cosinda model}') CRY12810
3036 FORMAT('Ja -self # GET ACCOUNTING INFO') CRY12820
3037 FORMAT('/space/cryollb/',A15,' model') CRY12830
3038 FORMAT('Ja -self # GET ACCOUNTING INFO') CRY12840
3039 FORMAT(EOF) CRY12850
3040 FORMAT('cosinda model}') CRY12860
3041 FORMAT('Ja -self # GET ACCOUNTING INFO') CRY12870
3042 FORMAT('/space/cryollb/',A15,' model') CRY12880
3043 FORMAT('Ja -self # GET ACCOUNTING INFO') CRY12890
3044 FORMAT(EOF) CRY12900
3045 FORMAT('cosinda model}') CRY12910
3046 FORMAT('Ja -self # GET ACCOUNTING INFO') CRY12920
3047 FORMAT('/space/cryollb/',A15,' model') CRY12930
3048 FORMAT('Ja -self # GET ACCOUNTING INFO') CRY12940
3049 FORMAT(EOF) CRY12950
3050 FORMAT('cosinda model}') CRY12960
3051 FORMAT(EOF) CRY12970
3052 FORMAT('cosinda model}') CRY12980
3053 FORMAT(EOF) CRY12990
3054 FORMAT('cosinda model}') CRY13000
3055 FORMAT(EOF) CRY13010
3056 FORMAT('cosinda model}') CRY13020
3057 FORMAT(EOF) CRY13030
3058 FORMAT('cosinda model}') CRY13040
3059 FORMAT(EOF) CRY13050
3060 FORMAT('cosinda model}') CRY13060
3061 FORMAT(EOF) CRY13070
3062 FORMAT('cosinda model}') CRY13080
3063 FORMAT(EOF) CRY13090
3064 FORMAT('cosinda model}') CRY13100
3065 FORMAT(EOF) CRY13110
3066 FORMAT('cosinda model}') CRY13120
3067 FORMAT(EOF) CRY13130
3068 FORMAT(EOF) CRY13140
3069 FORMAT(EOF) CRY13150
3070 FORMAT(EOF) CRY13160
3071 FORMAT(EOF) CRY13170
3072 FORMAT(EOF) CRY13180
3073 FORMAT(EOF) CRY13190
3074 FORMAT(EOF) CRY13200
3075 FORMAT(EOF) CRY13210
3076 FORMAT(EOF) CRY13220
3077 FORMAT(EOF) CRY13230
3078 FORMAT(EOF) CRY13240
3079 FORMAT(EOF) CRY13250
3080 FORMAT(EOF) CRY13260
3081 FORMAT(EOF) CRY13270
3082 FORMAT(EOF) CRY13280
3083 FORMAT(EOF) CRY13290
3084 FORMAT(EOF) CRY13300

CALLED FROM GETJCL (41) CRY12730
CALLED FROM NODES (44) CRY12740

C ENTRY INTO THIS SUBROUTINE TO CONVERT AN INTEGER TO CHARACTER FORM
ENTRY NHCD(NC,NCBCD)

NC=+1
END
NCBCD=NUMS(N)(11+)
RETURN

C SUBROUTINE TO CONVERT ALPHABETIC DATA TO LOWER CASE
CHARACTER (*) ALF, ALFLOW
CHARACTER*2 ALFIN(25), ALFLC(25)
CHARACTER*25 ALPHA, ALOWC

EQUIVALENCE (ALPHA,ALFIN(1)), (ALOWC,ALFLC(1))

C ALPHA=ALF
DO 220 I=1,NN
NCCHAR=ICHAR(ALFIN(I))
IF(NCCHAR .LT. 193 .OR. NCCHAR .GT. 233) THEN
ALFLC(I)=ALFIN(I)
ELSE
NCCHAR=NCCHAR-64
END
SUBROUTINE REGNI

CALLED FROM C

CALLED FROM C

COMMON/GEOMTY/ NTYP,NAN, GEOM(2)

CALL IF(NTYP .EQ. I) CALL SPHERE

CALL IF(NTYP .EQ. 2) CALL CYLNDR(NAN)

CALL RGNCNL(1)

RETURN

END

SUBROUTINE RGNCNL(NR)

CALLED FROM C

REGION WIDTH, TEMP, MATERIAL, NO. LAYERS THRU TEMPERATURES ... TEMPS(I-5) TEMP OF REGION 1-5

C TEMPS(6) OUTSIDE ATMOSPHERE TEMP

C TEMPS(7) OUTSIDE ATMOSPHERE TEMP

C TEMPS(8) INSIDE OF TANK WHEN REGNS(4) = FALSE

C TEMPERATURE UNITS DESIGNATOR, NTUNIT, SAVED IN MATRLS(9)

C AND THE VALUE OF THESE UNITS, ('F' OR 'R'), IN MATNMS(9)

C IF INPUT TEMPERATURES ARE TO BE DEG F MATRLS(9) - 1

C IF INPUT TEMPERATURES ARE TO BE DEG R MATRLS(9) - 2

C DESIGNATOR FOR OUTSIDE BOUNDARY COEFFICIENTS WILL BE SAVED IN NLAYRS(8) AND NLAYRS(9) FOR OUTSIDE TO BOUNDARY CONDUCTORS 1 AND 2 RESPECTIVELY.

C IF NLAYRS(8/9) = 1 BNCOEF(1/2) IS A CONVECTION CONDUCTOR

C IF NLAYRS(8/9) = 2 BNCOEF(1/2) IS A RADIATION CONDUCTOR

C IF EITHER OF THE ABOVE COEFFICIENTS IS A RADIATION CONDUCTOR AND IF THE NTUNIT IS DEG R, THEN CHANGE ALL TEMPERATURES TO DEG F.

C THICKNESS (WIDTH) FOR REGIONS 2, 3, 4 AND 5

COMMON /REGION/ NTHETA, NBETAS, BETA, RIN, TVOL, ROUT(9), REGNS(9),NLAYRS(9),TEMPS(9),THICK(9), THKLAY(9),MATRLS(9),MATNMS(9),HGNNMS(9)

COMMON/GEOMTY/ NTYP,NAN, GEOM(2)

LOGICAL REGNS

C

CHARACTER*8 GEOM

CHARACTER*16 MATRLS

CHARACTER*25 RGNNMS

CHARACTER*1 RF

C

COMMON/GEOMTY/ NTYP,NAN, GEOM(2)

C

REMAINING INPUT FOR REGION 1, AND INPUT FOR REGIONS 2, 3, 4 AND 5.

100 IF(NR .GT. 1) CALL CLEARS

PRINT 2000, NR, RGNNMS(NR)

NTUNIT = C

IF(NR .GT. 1) THEN

C THICKNESS (WIDTH) FOR REGIONS 2, 3, 4 AND 5
IF (NR .LE. 5) THEN
THICK(5) = RIN - THICK(4)
ROUT(5) = THICK(5)
IF (THICK(5) .LE. 0.0) THEN
PRINT 3003
CALL READIN (IGO, 1, 2)
ENDM
ENDIF
IF (IGO .EQ. 1) THEN
NR = 4
GO TO 100
ENDIF
IF (IGO .EQ. 2) THEN
REGNS(5) = .FALSE.
RETURN
ENDIF
GO TO 200
ENDIF
IF (NR .EQ. 4) THEN
ROUT(4) = RIN
IF (REGNS(5)) THEN
PRINT 2005, GEOM(NTYP), GEOM(NTYP), GEOM(NTYP), RIN
ELSE
THICK(4) = RIN
GO TO 200
ENDIF
ENDIF
C DOES USER WANT TO INPUT REGION THICKNESS OR LAYER THICKNESS?
C DETERMINE THIS THEN READ THE APPROPRIATE VALUE.
150 PRINT 2007, NR
CALL READIN (MT, 1, 2)
CALL CLEARS
IF (MT .EQ. 1) THEN
PRINT 2001, NR
CALL READRE (THICK(NR))
ENDIF
IF (MT .EQ. 2) THEN
PRINT 2008, NR
CALL READRE (THKLAY(NR))
ENDIF
ENDIF
C GET NO. OF LAYERS, (NO. OF NODES THRU THIS REGION)
200 PRINT 202, NR
CALL READIN (NLAY, 1, 10)
NLAYRS(NR) = NLAY
ENLAY = NLAY
C COMPUTE EITHER THE LAYER THICKNESS OR THE REGION THICKNESS.
IF (MT .EQ. 1) THEN
THKLAY(NR) = THICK(NR) / ENLAY
IF (MT .EQ. 2) THEN
THKLAY(NR) = THKLAY(NR) * ENLAY
IF (NR .LE. 1) THEN
ROUT(2) = ROUT(1) + THICK(2)
ENDIF
IF (NR .LE. 2) THEN
ROUT(3) = ROUT(2) + THICK(3)
ENDIF
IF (NR .LE. 3) THEN
ROUT(4) = ROUT(3) + THICK(4)
ENDIF
IF (NR .EQ. 4) THEN
IF (THICK(4) .GT. RIN) THEN
PRINT 3002, RIN
GO TO 150
ENDIF
ENDIF
C GET MAX RADIUS OF THE MODEL, -- STORE IT IN ROUT(6)
IF (ROUT(NR) .GT. ROUT(6)) THEN
ROUT(6) = ROUT(NR)
ENDIF
C NOW GET INITIAL TEMPERATURE FOR THIS REGION
45 PRINT 2006
CALL READAL (1, 1, 2)
IF (RF .EQ. 'R') THEN
RETURN
ENDIF
CALL CLEARS
PRINT 3001
GO TO 45
ENDIF
MATRIS(9) = 1
168
CALL MATNMS (9)
IF (RF .EQ. 'R') THEN
    MATRLS (9) = 2
    MATNMS (9) = 'R'
ENDIF
ENDIF
PRINT 2003, MATNMS (9)
CALL READE (TEMP) (NR)
NOW GET MATERIAL NUMBER FOR PROPERTIES
CALL CLEARS
CALL MATMNU (4231)
RETURN
I001 FORMAT (AI)
I002 FORMAT (II)
2000 FORMAT ('INPUTTING DATA FOR REGION', I3, ', ', A25)
2001 FORMAT ('TYPE IN THICKNESS (WIDTH) OF REGION', I2, ' (IN.)')
2002 FORMAT ('TYPE IN THE NO. OF LAYERS THRU REGION', I2)
2003 FORMAT ('TYPE IN THE INITIAL TEMPERATURE FOR THIS REGION',
            '(DEG , A1, \')
            1 ' (DEG \', A1, \')
            2 ' FROM THE INSIDE TANK WALL TOWARD THE CENTER OF THE \', A8, \',
            3 ' WHERE RIN, (THE INSIDE \', A8, \', RADIUS) = \', F8.3)
2004 FORMAT ('TEMPERATURES MAY BE IN DEGF OR DEGR IF NO',
            ' RADIATION IS PRESENT.'/
            ' THE TEMPERATURES WILL BE INPUT IN WHAT UNITS F OR R?'/
            ' TYPE IN F OR R')
2005 FORMAT ('NOW NEED TO SPECIFY THICKNESS OF REGION', I2/
            ' AND THE NUMBER OF LAYERS THRU THE REGION.'/
            ' TO DEFINE THE REGION THICKNESS THE INPUT MAY BE:'/
            1 ' THE REGION THICKNESS (IN.)'/
            2 ' OR 2. THE THICKNESS OF EACH LAYER IN THE REGION' /
            3 ' TYPE IN 1 OR 2')
2006 FORMAT ('TYPE IN THICKNESS (WIDTH) OF EACH LAYER OF REGION'
            I , I2, ' (IN.)'}
3001 FORMAT ('ERROR IN TYPING TEMPERATURE UNITS,'/
            ' THEY MUST BE F OR R' /
            2 ' TRY AGAIN')
3002 FORMAT ('ERROR IN REGION 4 THICKNESS, INPUT VALUE IS TOO',
            ' LARGE,' /
            1 ' MUST BE < INSIDE RADIUS, RIN = \', F7.1) /
            ' RETYPE REGION 4 THICKNESS')
3003 FORMAT ('ERROR IN REGION 5 THICKNESS, REGION 5 IS DEFINED/'
            ' BUT REGION 4 THICKNESS = RIN.' /
            2 ' THE OPTIONS ARE:'/
            1 ' 1. CHANGE THE WIDTH OF REGION 4.' /
            2 ' 2. ELIMINATE REGION 5 FROM THE GEOMETRY.' /
            3 ' TYPE IN 1 OR 2')
END

C SUBROUTINE MATMNU(NR)
CALLED FROM RONGNL (423)
CALLED FROM SPLINF (471)
C THIS ENTRY IS TO PRINT THE MATERIAL NAMES AS A MENU FOR THE USER
C TO PICK MATERIAL NUMBERS OF THE MODEL.
C THIS ROUTINE IS NOW IN EHABS LIB OF ROUTINES
C LEAVE THIS IN AS DOCUMENTATION ONLY; USE SUBROUTINE FROM FILE
C CRY2.
C COMMENT OUT ALL LINES OF THIS ROUTINE......
C C THE NAMES AND MATERIAL NUMBERS IN THIS ROUTINE MUST CORRESPOND
C TO THE NAMES AND NUMBERS IN THE DATA BASE.
C THE MATERIAL NUMBERS CORRESPOND TO THE ORDER IN WHICH THE NAMES
C ARE LISTED IN THIS PROGRAM, IN VECTOR MENU.
C C THE MATERIAL NUMBERS ARE:
C IXX LIQUID MATERIALS

CRY14710
CRY14720
CRY14730
CRY14740
CRY14750
CRY14760
CRY14770
CRY14780
CRY14790
CRY14800
CRY14810
CRY14820
CRY14830
CRY14840
CRY14850
CRY14860
CRY14870
CRY14880
CRY14890
CRY14900
CRY14910
CRY14920
CRY14930
CRY14940
CRY14950
CRY14960
CRY14970
CRY14980
CRY14990
CRY15000
CRY15010
CRY15020
CRY15030
CRY15040
CRY15050
CRY15060
CRY15070
CRY15080
CRY15090
CRY15100
CRY15110
CRY15120
CRY15130
CRY15140
CRY15150
CRY15160
CRY15170
CRY15180
CRY15190
CRY15200
CRY15210
CRY15220
CRY15230
CRY15240
CRY15250
CRY15260
CRY15270
CRY15280
CRY15290
CRY15300
CRY15310
CRY15320
CRY15330
CRY15340
CRY15350
CRY15360
CRY15370
CRY15380
CRY15390
CRY15400

169
THE ARRAY DATA FOR MATERIAL PROPERTIES ARE:

<table>
<thead>
<tr>
<th>ARRAY NO.</th>
<th>DESCRIPTION</th>
<th>SYMBOLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SPECIFIC HEAT * DENSITY OF MATERIAL</td>
<td>$C_p \rho$</td>
</tr>
<tr>
<td>2</td>
<td>SPECIFIC HEAT OF MATERIAL</td>
<td>$C_p$</td>
</tr>
<tr>
<td>3</td>
<td>DENSITY OF MATERIAL</td>
<td>$\rho$</td>
</tr>
<tr>
<td>4</td>
<td>VISCOSITY OF MATERIAL</td>
<td>$\mu$</td>
</tr>
<tr>
<td>5</td>
<td>ENTHALPY OF MATERIAL</td>
<td>$h$</td>
</tr>
<tr>
<td>6</td>
<td>THERMAL CONDUCTIVITY OF MATERIAL</td>
<td>$k$</td>
</tr>
</tbody>
</table>

SUBROUTINE RGN2T5

CALLED FROM SINTRU(4)

SUBROUTINE TO DEFINE THE GEOMETRY FOR REGIONS 2, 3, 4 AND 5.

INPUT FOR REGIONS 2, 3, 4, 5

KEG 2 -- LAYER ON OUTSIDE OF TANK.
PEG 3 -- 2ND LAYER ON OUTSIDE OF TANK, ON TOP OF LAYER 1 (REG 2).
REG 4 -- INSIDE TANK, STARTING AT TANK WALL TOWARD CENTER.
REG 5 -- INSIDE TANK, STARTING AT REG 3 TO CENTER.

COMMON /REGION/ NTHET, NBTAS, BETA, RIN, TVOL, ROUT(9),
               REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
               NTHXGRS(9), NXH, HXTEMP(10), NRHX(10), NLHX(10),
               MTHX(10), LNTHX(10)
COMMON/STUFF/ NHTT, PI, CONVY, CONVR, THETA0, DTHETA, NBASOS, ROUTSF,
               BMCOF(2)
COMMON/VULGAE/ NUL4, NUL5, NTM4, RINMH, FCFUL, RADULG, TVULFT,
               CT, LG(3), LIQVAP(3)

LOGICAL REGNS, VPCSHD

CHARACTER*1 YN, CT, LG
CHARACTER*6 LIQVAP
CHARACTER*16 MATNMS
CHARACTER*25 RGNMNS

NTUNIT=MATRLS(9)

GET INPUT INFO FOR REGION 2, (INSULATION) IF ANY

CALL CLEARS
PRINT 2201
PRINT 2001
CALL READAL(1,YN)
IF (YN .EQ. 'Y') THEN
   REGNS(2) = .TRUE.
CALL RGN2NL(2)
ELSE
   GO TO 400
ENDIF

CALL CLEARS
PRINT 2301
PRINT 2001
CALL READAL(1,YN)
IF (YN .EQ. 'Y') THEN
   REGNS(3) = .TRUE.
CALL RGN2NL(3)
ELSE
   GO TO 400
ENDIF

CALL CLEARS
PRINT 2401
CALL READIN(MINSR, 1, 2)

170
CALL RGNGNL(4)
IF(MATRLS(4) .LT. 200) THEN
  CT='L'
ELSE
  IF(MATRLS(4) .LT. 300) THEN
    CT='F'
    PCTFUL=100.
  ELSE
    CT='0'
    PCTFUL=0.0
  ENDIF
ENDIF
ENDIF
501 IF(REGNS(5)) THEN
  NR=5
  CALL RGNGNL(NR)
  IF(NR .NE. 5) GO TO 501
  CT=' '
  509 IF(MATRLS(4) .LT. 300 .AND. MATRLS(5) .LT. 300) THEN
    BOTH SOLID
    CT='F'
    PCTFUL=100.0
  ELSE
    IF(MATRLS(4) .GE. 300 .AND. MATRLS(5) .GE. 300) THEN
      BOTH VAPOR
      CT='0'
      PCTFUL=0.0
    ELSE
      IF(MATRLS(4) .LT. 300 .AND. MATRLS(5) .LT. 300) THEN
        BOTH LIQUID AND VAPOR
        CT='M'
        PCTFUL=100.
      ELSE
        BOTH LIQUID AND SOLID
        CT='M'
        PCTFUL=100.
      ENDIF
    ENDIF
  ENDIF
ENDIF
ENDIF
ENDIF
C CHECK ON MATERIALS OF REGIONS 4 AND 5.
C IF BOTH ARE NOT LIQUID, THEY COULD BE:
C 1. BOTH SOLID, CT='F', PCTFUL=100.
C 2. BOTH VAPOR, CT='0', PCTFUL=0.0
C 3. 1 LIQUID AND 1 VAPOR, CT='M', PCTFUL=100.
C 4. 1 LIQUID AND 1 SOLID, CT='M', PCTFUL=100.
C 5. 1 VAPOR AND 1 SOLID, CT='M', PCTFUL=100.
C IF(CT .NE. 'L') THEN
  IF(MATRLS(4) .LT. 300 .AND. MATRLS(5) .LT. 300) THEN
    BOTH SOLID
  CT='F'
  PCTFUL=100.0
  ELSE
    IF(MATRLS(4) .GE. 300 .AND. MATRLS(5) .GE. 300) THEN
      BOTH VAPOR
  CT='0'
  PCTFUL=0.0
  ELSE
    CALL CLEARS
    CALL RGNGNL(4,0,45)
    IF(N45 .EQ. 0) THEN
      CALL MATMN(4)
      GO TO 509
    ELSE
      IF(N45 .EQ. 4) THEN
        CALL MATMN(5)
        GO TO 509
      ELSE
        IF(N45 .EQ. 5) THEN
          CALL MATMN(4)
          CALL MATMN(5)
          GO TO 509
        ELSE
          PRINT 3008
          GO TO 520
        ENDF
      ENDF
    ENDF
    ENDF
    ENDF
    ENDF
ENDIF
ENDIF
C THIS IS DONE ABOVE.
C IF BOTH ARE LIQUID; SET CT='L', AND CALL ULLAGE ROUTINES.
C THIS IS DONE ABOVE.
MATERIALS ARE MIXED
CT=’M’
PCTFUL=100.0
ENDIF
ENDIF
ENDIF
ENDIF

CHECK ON HEAT TRANSFER MECHANISMS FOR REGIONS 4 AND 5.
PROMPT USER FOR 1. CONDUCTION ONLY; 2. CONVECTION ONLY;
OR 3. BOTH COND. AND CONV.
THEN IF RESPONSE IS 2 OR 3, PROMPT USER FOR CONV. COEFFICIENTS;
CIRCUMFERENTIAL AND/OR RADIAL
NHTT=0
CALL CLEARS
PRINT 2402
CALL READIN(NHTT,1,3)

IF(NHTT .GE. 2) THEN
CONVY=0.0
PRINT 2403
PRINT 2001
CALL READAL(1,YN)
IF(YN .EQ. ‘Y’) THEN
PRINT 2404
CALL READRE(CONVY)
CONVY=CONVY/144.
ENDIF
CONVR=0.0
PRINT 2405
PRINT 2001
CALL READAL(1,YN)
IF(YN .EQ. ‘Y’) THEN
PRINT 2404
CALL READRE(CONVR)
CONVR=CONVR/144.
ENDIF
ENDIF
END OF IFBLOCK
NHTT .GE. 2
GET % TANK IS FILLED, & WHERE IS ULLAGE.
IF PCTFUL = 100 CT=’F’ TANK IS FULL NO ULLAGE
IF PCTFUL = 0 CT=’O’ TANK IS EMPTY, ALL NODES ARE VAPOR
IF 0 < PCTFUL < 100 TANK HAS ULLAGE, WHERE IS IT?
CT=’L’ 1-G ANALYSIS, FLAT ULLAGE ON TOP
CT=’C’ 0-G ANALYSIS, ULLAGE AT CENTER
CT=’T’ 0-G ANALYSIS, ULLAGE AT TOP W/ FILM
IF(CT .EQ. ‘L’) THEN
CALL ULLINF
CALL ULLINF
ENDIF
ENDIF
END OF IFBLOCK REGNS(4) STARTING AT IFN 400

INPUT INFO ABOUT HEAT EXCHANGERS, MAX NO. = 10
CALL CLEARS
MAX=0
PRINT 2601
PRINT 2001
CALL READAL(1,YN)
700 IF(YN .EQ. ‘Y’) THEN
MAX=MAX+1
CALL CLEARS
PRINT 2602
N=MAX
CALL READHX
710 NERR=0
CALL READHX(N, NERR)
IF(NERR .GT. 0) THEN
YN = 'N'
GO TO 715
ENDIF
CALL CLEARS
PRINT 26081, NHX, NLHX(NHX), NRHX(NHX), NTHHX(NHX),
1 LNGTHX(NHX), HXTEMP(NHX)
PRINT 2001
CALL READAL(1, YN)
IF(YN .EQ. 'N') THEN
YNX=NHX-1
GO TO 770
ELSE
ENDIF
ENDIF

C
C TEST THIS NEW HX FOR : OTHER HX'S IN SAME REGION,
OTHER HX'S IN SAME LAYER OF SAME REGION,
C
NHXMI=NHX-1
750 DO 760 I=1, NHXMI
C HX(NHX) IS IN SAME REGION AS HX(I)
C IF(NLHX(NHX) .EQ. NLHX(I)) THEN
C HX(NHX) ALSO ON SAME LAYER AS HX(I)
NOVLAP=0
C IF(NTHHX(NHX) .LT. NTHHX(I)) THEN
C TEST FOR OVERLAP
C IF(NTHHX(NHX)+LNGTHX(NHX)-1 .GE. NTHHX(I)) THEN
C THESE 2 HX'S OVERLAP; ERROR
C WRITE ERROR MESSAGE AND CHANGE AN HX OR DELETE A HX.
NOVLAP=1
ENDIF
ELSE
C HX(NHX) STARTS AT LARGER THETA THAN HX(I)
C TEST FOR OVERLAP
C IF(NTHHX(I)+LNGTHX(I)-1 .GE. NTHHX(NHX)) THEN
C THESE 2 HX'S OVERLAP; ERROR 2
C WRITE ERROR MESSAGE AND CHANGE AN HX OR DELETE HX.
NOVLAP=2
ENDIF
IF(NOVLAP .GT. 0) THEN
NHXEND=NTHHX(NHX)+LNGTHX(NHX)-1
IEND =NTHHX(I) +LNGTHX(I)-1
IF(NOVLAP .EQ. 1) PRINT 3001, NHX, I, NHTHX(NHX),
1 NHXEND, I, NTHHX(I), IEND
1 IF(NOVLAP .EQ. 2) PRINT 3001, I, NHX, I, NTHHX(I), IEND,
1 NHX, NTHHX(NHX), NHXEND
CALL READAL(1, YN)
ENDIF
ENDIF

173
N=N+NH
GO TO 715
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
CONTINUE
760 CONTINUE
C
770 IF(NHX .LT. I0) THEN
PRINT 26083
CALL READAL(I,YN)
IF(YN .EQ. 'Y') GO TO 700
ENDIF
ENDIF
VAPOR COOLED SHIELDS INPUT
9/29 RE
PRESENTLY THE VAPOR COOLED SHIELDS OPTION IS NOT IN THE SYSTEM.
VAPOR COOLED SHIELDS ARE NOT WELL DEFINED.
HEAT EXCHANGERS CAN BE SUBSTITUTED FOR VAPOR COOLED SHIELDS
FOR THE PRESENT. THIS CAPABILITY MAY BE WORKED ON AT A LATER DATE.
C
OUTSIDE ATMOSPHERE BOUNDARY NODES, 2 POSSIBLE
C
OUTSIDE BOUNDARY NODES WILL BE CONVECTION OR RADIATION
C NLAYSRS(8/9) WILL BE SET TO 1 FOR CONVECTION OR TO 2 FOR RADIATION
C IF EITHER OF THESE NODES IS DEFINED AS A RADIATION NODE
C AND IF THE INITIAL TEMPERATURES ARE IN DEG R,
C THEN THE INITIAL TEMPERATURES WILL BE CONVERTED TO DEG F.
C
TEMPs(6)--9999.9
TEMPs(7)--9999.9
IBN=0
CALL CLEARS
PRINT 2901
CALL READAL(I,YN)
IF(YN .EQ. 'Y') THEN
PRINT 2902, MATNMS(9)
IBN=IBN+1
CALL READRE(TEMPS(IBN+5))
920 PRINT 2902, MATNMS(9)
IBN=IBN+1
CALL READRE(TEMPS(IBN+5))
930 PRINT 2903
CALL READAL(I,YN)
IF(YN .EQ. 'C') THEN
PRINT 2904
CALL READRE(BNCOEF(IBN))
BNCOEF(IBN)=BNCOEF(IBN)/144.
NLAYSRS(7+IBN)=1
ELSE
IF(YN .EQ. 'R') THEN
C OUTSIDE BOUNDARY NODE, RADIATION TO SURFACE
C TYPE IN RADIATION COEF (EPS*FORMF)
C PROGRAM WILL MULTIPLY BY STEFAN-BOLTZMAN AREA TO
C CONVERT H TO BTU/HR-IN2-DEG, H=H/144
C COMPUTE G=H*A
C SET NLAYSRS(8/9) = 2 DENOTING RADIATION.
PRINT 2905
CALL READRE(BNCOEF(IBN))
BNCOEF(IBN)=BNCOEF(IBN)*0.1712E-8/144.
NLAYSRS(7+IBN)=2
ELSE
C WRONG REPLY, REPLY MUST BE 'C' OR 'R', TRY AGAIN
PRINT 3005
GO TO 930
ENDIF
ENDIF
CALL CLEARS
IF(IN .LT. 2) THEN
  PRINT 2906
  PRINT 2001
  CALL READAL(1,YN)
  IF(YN .EQ. 'Y') THEN
    GO TO 920
  ELSE
    ENDIF
  ENDIF
ENDIF

C CHECK FOR RADIATION CONDUCTOR AND DEGR.
C INDICATOR 'F' OR 'R' IS IN MATNMS(9)
C IF TRUE CHANGE THE INPUT TEMPERATURES TO DEG F.
C THOSE TO BE CHANGED: TEMPS(1-9), HXTEMP(1-10)
C IF(MATNMS(9) .EQ. 'F') THEN
  IF(NLAYRS(9) .EQ. 3) THEN
    CALL CLEARS
  PRINT 2909
  MATNMS(9) = 'F'
  MATRLS(9) = IHN+5
  DO 950 I=1,IBNT
    TEMPS(I) = TEMPS(I) - 460.0
  950 CONTINUE
  DO 955 I=1,NHX
    HXTEMP(I) = HXTEMP(I) - 460.0
  955 CONTINUE
  ELSE
    ENDIF
  ENDIF
RETURN

C FORMAT STATEMENTS
C
1001 FORMAT(A1)
  2001 FORMAT(' TYPE IN Y OR N')
  2201 FORMAT(///' IS THERE TO BE A REGION ON THE OUTSIDE',
    1 ' OF THE TANK?'/
    2 ' EG. INSULATION.' )
  2301 FORMAT(///' IS THERE TO BE A 2ND REGION OUTSIDE OF',
    1 ' THE TANK?'/
    2 ' EG. MORE OR DIFFERENT INSULATION.')
  2401 FORMAT(///' FOR THIS ANALYSIS THE INSIDE OF THE TANK',
    1 ' WILL BE NODALIZED']/
    2 ' HOW MANY REGIONS INSIDE OF THE TANK? 1 OR 2')
  2402 FORMAT(///' THE HEAT TRANSFER MECHANISM INSIDE THE TANK,'/
    1 ' I.E., REGIONS 4 AND 5, IS TO BE:']/
    2 ' 1. CONDUCTION ONLY'/
    3 ' 2. CONVECTION ONLY'/
    4 ' 3. CONDUCTION AND CONVECTION'/
    5 ' TYPE IN 1 2 OR 3'
  2403 FORMAT(' CONVECTION IN THE Y, (CIRCUMFERENTIAL), DIRECTION?')
  2404 FORMAT(' CONVECTION IN THE RADIAL DIRECTION?')
  2405 FORMAT(' HEAT EXCHANGER INFO, MAX NO. =10')
  2601 FORMAT(/// ' ARE THERE TO BE ANY HEAT EXCHANGERS?')
  2602 FORMAT(/// ' HEAT EXCHANGER INFO, MAX NO. =10')
  2603 FORMAT(/// ' HEAT EXCHANGER NO.,13,' SPECIFIED '/
    1 ' ON TOP OF LAYER,13,' OF REGION,12,
    2 ' STARTING AT THETA ANGLE,13,' FOR ,13,' NODES','
    3 ' WITH TEMPERATURE =',F7.2,'
    4 ' IS THIS CORRECT?')
  2604 FORMAT(' DO YOU WANT TO RE-SPECIFY OR DELETE',
    1 ' HEAT EXCHANGER,13,'?'/
    2 ' TYPE Y TO RE-SPECIFY OR N TO DELETE.')
  2605 FORMAT(' MORE HEAT EXCHANGERS? TYPE Y OR N')
  2701 FORMAT(/// ' ARE THERE TO BE VAPOR COOLED SHIELDS?')
  2702 FORMAT(/// ' IN NUMBER OF SHIELDS 1 OR 2')
  2703 FORMAT(/// ' IN THE REGION NUMBER WHERE THE 1ST SHIELD GOES')
  2704 FORMAT(' THE SHIELD IS ON TOP OF WHICH LAYER OF THE REGION?/
    TYPE IN THE LAYER NO.')
  2705 FORMAT(' IN THE THETA ANGLE WHERE THE VCS STARTS,'
COUNT UP FROM THE SOUTH POLE.

Type in the number of thetas they're.

VCS is to cover.

Type in the vapor cooled shield 1.

Temperature (deg 'A1').

Type in the region number where the 2nd shield goes.

Type in the vapor cooled shield 2.

Temperature (deg 'A1').

There may be up to two boundary nodes on the 'E'.

Outside of the tank wall.

EC. Outside atmosphere.

Do you want one or more of these boundary nodes?

Type in the outside atmosphere.

Temperature (deg 'AA').

Type in the vapor cooled shield.

Temperature (deg 'AA').

Type in 0 (zero) for OK, continue; 1 to change material 1; 5 to change material 5; 45 to change both materials."

***ERROR*** HX 'I' AND HX '2' OVERLAP; HX'S CANNOT OVERLAP.

Type in which HX do you want to respecify? 'I', '2', or 'I2'.

Wrong reply, reply must be C or R, try again.

***ERROR*** ERROR IN TYPING INPUT, THE ONLY VALID RESPONSES ARE '0', '4', '5', OR '45'.

One or type Q to quit.

176
IF (CT .EQ. 'O' .OR. CT .EQ. 'O') THEN
   PRINT 2502
   CALL READAL (1, CT)
   IF (CT .EQ. 'C' .OR. CT .EQ. 'T') THEN
      ELSE
      PRINT 3004
      GO TO 610
      ENDIF
   ELSE
      IF (CT .NE. 'I') THEN
         PRINT 3005
         GO TO 605
     ENDIF
   ELSE
      TANK HAS REGION(S) 4, (5) DEFINED BUT %FULL-0.
      HENCE THE TANK IS EMPTY. ALL NODES SHOULD BE DEFINED AS VAPOR NODES.
      CT-' O'
      ENDIF
   ELSE
      TANK HAS REGION(S) 4, (5) DEFINED BUT %FULL-100.
      HENCE THE TANK IS COMPLETELY FULL. ALL NODES ARE LIQUID NODES.
      CT=' F'
      ENDIF
   END

C FORMAT STATEMENTS
C
2501 FORMAT (///' TYPE IN % TANK IS FULL OF LIQUID.'
2502 FORMAT (///' WHERE IS THE ULLAGE? CENTER OR TOP?'/
2503 FORMAT (///' IS THIS ANALYSIS A LOW-G OR 1-G ANALYSIS?'/
2504 FORMAT (///' ULLAGE MUST BE SPECIFIED AS CENTER C OR TOP T'/
2505 FORMAT (' WRONG REPLY, REPLY MUST BE 0 OR 1, TRY AGAIN.'
C
SUBROUTINE READH(NNHX,NERR)
CALLED FROM RGN2T5 (43)
COMMON /REGION/ NTHETA, NBETAS, BETA, RIN, TVOL, ROUT (9),
1 REGNS (9), NLAYRS (9), TEMPS (9), THICK (9),
2 THKLAY (9), MATRLS (9), MATNMS (9), RGNNMS (9)
COMMON /HTXGRS/ NHX, NHXTEM (10), NRMX (10), NMLX (10),
1 NTHMX (10), LNMTHX (10)
C
LOGICAL REGNS
C
CHARACTER*16 MATNMS
CHARACTER*25 RGNNMS
CHARACTER*1 YN
C
CALL CLEARS
PRINT 26022, NNHX
PRINT 2603
CALL READIN (NNHX, NNHX)
C
HX NO. NNHX IS IN REGION NNHX; TEST IF REGN NNHX EXISTS
NNHX=NNHX (NNHX)
IF( NOT. REGNS (NNHNN)) THEN
C REGN NNHX IS FALSE, ERROR, CANNOT BE A HX, REGN DOES NOT EXIST;
C UNLESS NNHX = 4 AND NLAYERS=4, (INSIDE TANK WALL)
C IF(NNNN .EQ. 4) THEN
C THIS HX IS ON THE INSIDE TANK WALL
NMLX (NNHX) = 1
GO TO 730
ELSE
   PRINT 3002, NNHX, NNNH
   NNN=1
   RETURN
ENDIF
C GET THE LAYER NO. OF THE DESIGNATED REGION THAT CONTAINS THE HX.
C THE HX IS ON TOP OF THE SPECIFIED LAYER.
C COUNT LAYER FROM OUTSIDE TOWARD CENTER OF SPHERE FOR ALL REGIONS.

NREGHX-NRHX {NNHX)

IF{NLAYRS(NREGHX) .LE. I) THEN
NLHX(NNHX) = 1
ELSE
PRINT 2604, NRHX(NNHX)
CALL READIN(NLHX(NNHX),I,NLAYRS(NREGHX))
ENDIF
PRINT 2605
CALL READIN(NTHHX(NNHX),I,NTHETA)
PRINT 2606
CALL READIN(LNGTHX(NNHX),I,NTHETA)
IF(NTHHX(NNHX)+LNGTHX(NNHX)-I .GT. NTHETA) THEN
PRINT 3003,NNHX, NTHHX(NNHX),LNGTHX(NNHX),NTHETA
NERR=10
RETURN
ENDIF
PRINT 2607, MATNMS(9)
CALL READDATAHXTEMP(NNHX)
RETURN

C FORMAT STATEMENTS
C
26022 FORMAT(' INPUT FOR HEAT EXCHANGER NO.',I2)
2603 FORMAT(' TYPE IN THE REGION NUMBER WHERE THE',
1 ' HEAT EXCHANGER GOES.')
2604 FORMAT(' THE HEAT EXCHANGER IS ON TOP OF',
1 ' WHICH LAYER OF REGION',f3,'?'/
2 ' TYPE IN THE LAYER NO., COUNT LAYERS FROM OUTSIDE'/
3 ' TOWARD THE CENTER.')
2605 FORMAT(' TYPE IN THE THETA ANGLE WHERE',
1 ' THE HEAT EXCHANGER STARTS'/
2 ' COUNT UP FROM THE SOUTH POLE.')
2606 FORMAT(' TYPE IN THE NUMBER OF THETAS THAT',
1 ' THE HEAT EXCHANGER COVERS.')
2607 FORMAT(' TYPE IN THE HEAT EXCHANGER TEMPERATURE (DEG ',A1,')'
3001 FORMAT(**** ERROR ****/
1 ' YOU SPECIFIED LAYER NO. ',I3, ' FOR THE HEAT EXCHANGER'/
2 ' BUT REGION',I2,' H AS ONLY',I3,' LAYERS.'/
3 ' RETYPE THE LAYER NO.(')
3002 FORMAT(**** ERROR ****/
1 ' FOR HEAT EXCHANGER NUMBER',I3/)
2 ' THE STARTING POSITION, (THETA ',I3,' ) PLUS'/
3 ' THE LENGTH OF THE HX, (',I3,' )'/
4 ' EXCEEDS THE NUMBER OF THETAS IN THE SPHERE, (',I3,' )'/
5 ' RESPECIFY OR DELETE THIS HEAT EXCHANGER'
C
END

C44SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SUBROUTINE NODES
CALI, ED FROM SINTRU(4)
C GENERATE NODE DATA BLOCK
C GENERATE ALL THE NODES FOR THE SINDA MODEL
C 1. REG 1 TANK -- 2 SURFACES (ARITH NODES), N1 LAYERS(DIFF NODES)
C 2. REG 2 IF ANY -- 1 OUTSIDE SURF (ARITH NODES), N2 LAYERS(DIFF NODES)
C 3. REG 3 IF ANY -- 1 OUTSIDE SURF (ARITH NODES), N3 LAYERS(DIFF NODES)
C 4. REG 4 IF ANY -- N4 LAYERS(DIFF NODES)
C 5. REG 5 IF ANY -- N5 LAYERS(DIFF NODES)
C 6. INSIDE TANK WHEN REGNS(4) .FALSE. --1 BNDY NODE AT EACH THETA
C 7. HEAT EXCHANGERS ---1 BNDY NODE FOR EACH HEAT EXCHANGER
C 8. OUTSIDE ATMOSPHERE ---1 BNDY NODE
C 9. VAPOR COOLED SHIELDS ---1 BNDY NODE FOR EACH SHIELD(NOT IN PGM)
C
C GENERATED NODE NUMBER SERIES, (NODEBASE), FOR EACH REGION IN MODEL
C
1. REG 1 TANK -- SURF1 - 1000; DIF - 2000; SURF2 - 3000
C 2. REG 2 IF ANY -- DIF - 4000; SURF - 5000
C 3. REG 3 IF ANY -- DIF - 6000; SURF - 7000
C 4. REG 4 IF ANY -- DIF - 8000
C 5. REG 5 IF ANY -- SURF - 9000; DIF - 10000
C 6. INSIDE TANK WHEN REGNS(4).F. BDY - 18000
C 7. HEAT EXCHANGERS BDY - 20001, 20002, ETC.
C 8. OUTSIDE ATMOSPHERE BDY - 20301
C 9. VAPOR COOLED SHIELDS BDY - 20101, 20102, ETC.

C
CALL CO_4ON/UNITS/
CALL MODU, INPEKO, ISCRCH, SINDA
COMMON /REGION/ NTHETA, NBETAS, BETA, RIN, TVOL, ROUT (9),
1 REGNS (9), NLAYRS (9), TEMPS (9), THICK (9),
2 THKLAY (9), MATRLS (9), MATNMS (9), RGNNMS (9)
COMMON /HTXGRS/ NHX, HXTEMP (IO), NRHX (10), NLHX (10),
1 NTHHX (I0), LNGTHX (I0)
COMMON/STUFF/ NHTT, PI, CONVY, CONVR, THETA0, DTHETA, NBASOS, ROUTSF,
1 BNCOEF (2)
COMMON/SUBRTS/ SPLIPT, XCUTI, XCUT2, VBLBL1, VBLBL2, OUTBLK
COMMON/UULLAGE/ NLUL4, NLULS, NTHU41, RINMHH, PCTFUL, RADULG, TVULFT,
1 CT, LG (3), LIQVAP (3)
COMMON/GEOMTY/ NTYP, NAN, GEOM (2)
LOGICAL REGNS, SINDA
LOGICAL SPLIPT

C
CHARACTER*1 CT, LG
CHARACTER*2 NUMBR
CHARACTER*6 LIQVAP
CHARACTER*6 XCUT1, XCUT2, VBLBL1, VBLBL2, OUTBLK
CHARACTER*8 GEOM
CHARACTER*15 HXGBCD
CHARACTER*16 MATNMS
CHARACTER*17 HXLABL
CHARACTER*25 RGNNMS

C
DATA HXGBCD/'HEX THERMAL EXCHANGER'/

CALL WRITE NODE BLOCK TITLE
CALL BLITTL
C NTPY=1, SPHERE
CALL IF(NTPY .EQ. 1) CALL SPHNS
C NTPY=2, 2-D CYLINDER
CALL IF(NTPY .EQ. 2) CALL CYLNS
C TEST FOR BNDY NODES (18000) INSIDE TANK
C IF SUB VBLBL EXISTS OR IF USER WANTS THESE NODES, GENERATE THEM HERE
C USE REGNS(9) AS A FLAG FOR NODES 18000
C
IF(.NOT. REGNS(4)) THEN
IF(VBLBL .NE. ' ') THEN
REGNS(9)=-.TRUE.
ELSE
CALL CLEARS
PRINT 3001
CALL READIN(ICT, 1, 3)
IF(ICT .GT. 1) THEN
REGNS(9)=.TRUE.
CALL CLEARS
IF (ICT .EQ. 2) PRINT 3002, MATNMS (9)
IF (ICT .EQ. 3) PRINT 3002, MATNMS (9)
CALL READRE(TEMSPS (9))
IF(ICT .EQ. 3) THEN
PRINT 3003, MATNMS (9)
CALL READRE(THICK(9))
C 3003
CALL READRE(PCTFUL)

179
IF(PCTFUL .LE. 0.0 .OR. PCTFUL .GE. 100.) THEN
  PRINT 3000
  GO TO 301
ENDIF
CT=' I'
THKLAY(4)=RIN
IF(NTYP .EQ. 1) CALL ULLGET
IF(NTYP .EQ. 2) CALL ULLIG
ENDIF
ENDIF
IF(REGNS(9)) THEN
  WRITE (MODU, 2000)
  IF(THICK(9) .EQ. 0.0) THEN
    CALL RITNDS(443)
    CALL RITNDS(1,3,NODNO,1,1,HXTEMP(I),1.0,HXLABL)
  CONTINUE
ENDIF
ENDIF
C
C IF NHX > 0, DEFINE HEAT EXCHANGER NODES, NOE. 20001, 20002, ETC.
IF(NHX .GT. 0) THEN
  DO 600 I=1,NHX
    NODNO=20000+I
    CALL NBCD(I,NUMB)
    HXLABL=HXMLBCD/NUMB
    CALL RITNDS(1,3,NODNO,1,1,HXTEMP(I),1.0,HXLABL)
  CONTINUE
ENDIF
C
GET NODEBASE AND RADIUS FOR OUTSIDE SURFACE
C
NS=1
DO 101 J=2,3
  IF(REGNS(J)) NS=J
  CONTINUE
ROUTSF=ROUT(NS)
NBASOS=2000*NS+1000
C
OUTPUT NODES FOR VAPOR COOLED SHIELDS
C
THIS SECTION PRESENTLY NOT ACTIVATED, VAPOR COOLED SHIELDS
C
NOT WELL DEFINED. PRESENTLY USE HEAT EXCHANGERS AS SUBSTITUTE.
C
OUTSIDE ATMOSPHERE NODE (BOUNDARY NODE)
C
RITNDS(443)
IF(TEMPS(6) .GE. -9999.9) THEN
  CALL RITNDS(1,3,20301,1,1,TEMPS(6),1.0,'OUTSIDE ATMOS 1 ')
ENDIF
IF(TEMPS(7) .GE. -9999.9) THEN
  CALL RITNDS(1,3,20302,1,1,TEMPS(7),1.0,'OUTSIDE ATMOS 2 ')
ENDIF
CALL BLKEND
RETURN

C FORMAT STATEMENTS

2000 FORMAT(7X,'REM CONSTANT VALUE BOUNDARY NODES; REGION 4, ',
1 'INSIDE OF TANK')
2001 FORMAT(///,'FOR THIS MODEL, REGION 4 (INSIDE OF TANK),'/'
1 'IS NOT MODELIZED WITH SINDA NODES;'/
2 'DO YOU WANT CONSTANT TEMPERATURE BOUNDARY NODES?/
3 'TO CONNECT TO INSIDE OF TANK WALL, OR NOT?/
4 'YOU MAY HAVE:'/
5 '1. NO CONSTANT TEMPERATURE BOUNDARY NODES.'/
6 '2. A SINGLE SET OF CONSTANT TEMPERATURE NODES.'/
7 '2 SETS OF CONSTANT TEMPERATURE NODES TO '/

C
TO SIMULATE LIQUID AND VAPOR IN 1-G.

2001 FORMAT(/' TYPE IN 1 2 OR 3')
2002 FORMAT(/' TYPE IN THE TEMPERATURE OF THESE BNDY NODES',
  1 ' DEG('AI,'')')
2003 FORMAT(/' TYPE IN THE TEMPERATURE OF THE LIQUID BNDY NODES',
  1 ' DEG('AI,'')')
2004 FORMAT(/' TYPE IN THE TEMPERATURE OF THE VAPOR BNDY NODES',
  1 ' DEG('AI,'')')
2005 FORMAT(/' % TANK IS FULL,'//
  1 ' % FULL MUST BE > 0.0 AND < 100.0; TRY AGAIN.')

COMMON/UNITS/ MODU, INPEKO, ISCRCH, SINDA

C CHARACTER*16 MLABL
C LOGICAL SINDA
C
NNODE=NNO
IF(NODTYP .EQ. 4) GO TO 400
IF(NODTYP .EQ. 1) THEN
C DIFFUSION NODES
IF(NN .EQ. 1) THEN
WRITE(MODU, 2101) NNODE, TEMP, VOL, MLABL
ELSE
WRITE(MODU, 2102) NNODE, NN, NDEL, TEMP, VOL, MLABL
ENDIF
ELSE
IF(NODTYP .EQ. 2) THEN
C ARITHMETIC NODES
CVOL=1.0
ELSE
C BOUNDARY NODES
NNODE=NNO
CVOL=1.0
ENDIF
IF(NN .EQ. 1) THEN
WRITE(MODU, 2101) NNODE, TEMP, CVOL, MLABL
ELSE
WRITE(MODU, 2102) NNODE, NN, NDEL, TEMP, CVOL, MLABL
ENDIF
ENDIF
199 RETURN
C WRITE SOURCE CARDS IN SOURCE BLOCK
400 IF(NN .EQ. 1) THEN
WRITE(MODU, 2201) NNODE, TEMP, VOL, MLABL
ELSE
WRITE(MODU, 2202) NNODE, NN, NDEL, TEMP, VOL, MLABL
ENDIF
RETURN
C
SUBROUTINE SRCDAT
Cantino FROM SIMTRU (4)
C
BLOCK 2, SOURCE DATA BLOCK -- GENERATE NS SOURCE LINES
C
NODE NO., Q(BTU/HR.), QIN-(BTU/HR.-IN2)
C
COMMON /REGION/ NTHETA, NBETAS, BETA, RIN,TYOL, ROUT (9),
1 REGNS (9), MLAYS (9), TEMPS (9), THICK (9),
2 THKLAY (9), MTRLS (9), MATNMS (9), RGNNMS (9)
COMMON /STUFF/ NTHT, PI, CONVY, CONVR, THETA0, DTHETA, NSASOS,
1 ROUTSF, BNCOEF (2)
COMMON /OUTSRC/ NQIN, QEFF
COMMON /GEOM/ NTYP, NAN, GEOM (2)
LOGICAL REGNS
CHARACTER*8 GEOM
CHARACTER*16 MATNMS
CHARACTER*25 RGNNMS
CHARACTER*16 QUNITS (3)
CHARACTER*1 YN, YES
DATA QUNITS/'(BTU/(HR-FT2))', '(BTU/(HR-IN2))'
1 'BTU/HR ON SPHERE'/
DATA YES/'Y'/
C
CHECK TO SEE IF THERE IS A Q TERM INTO THE OUTSIDE SURFACE.
C
QIN=0
CALL CLEARS
PRINT 2001
CALL READAL (I,YN)
IF (YN .EQ. 'Y') THEN
100 PRINT 2002
CALL READIN (NQIN,1,3)
QIN=0.0
PRINT 2003, QUNITS (QIN)
CALL READRE (QIN)
C
WRITE THE HEADER FOR THE SOURCE DATA BLOCK
C
CALL BL2TTL
C
WRITE THE SOURCE DATA LINES TO UNIT 10
C
NNODE=NBASOS
R=ROUT (6)
C
CONVERT QIN TO BTU/(HR-IN2)
GO TO (110,120,130), NQIN
110 QEFF=QIN/144.
GO TO 140
120 QEFF=QIN
GO TO 140
130 QEFF=QIN/4./PI/R/R
140 CONTINUE
C
IF (NTYP .EQ. 1) THEN
DO 200 J=1,NTHETA, 2
NNODE=NNODE+1
JJJ=JJJ+2
CALL AREAESP (1, JJJ, R, 1., AREA)
QQ=QEFF*AREA
C
END
NFNY=2
NJ=NTHETA-J
IF(NJ.LE.0) NMANY=1
CALL RITNDS(NMANY,4,NNODE,NJ,1,QQ,QIN,QUNITS(NQIN))
CONTINUE
ELSE
NMANY=1
NJ=1
DO 205 J=1,NTHETA
NNODE=NNODE+1
CALL AREAACYL(472)
205
CALL AREACYL(I,J,0.0,0,AREA,-2)
QQ=QEPF*AREA
CALL RITNDS(NMANY,4,NNODE,NJ,I,QQ,QIN,QUNITS(NQIN))
CONTINUE
ENDIF
CALL BLKEND
ENDIF
RETURN
I001 FORMAT(A1)
2001 FORMAT(//'
IS THERE TO BE A CONSTANT Q INPUT, (SOURCE TERM)'/
1
'INTO THE OUTSIDE SURFACE OF THE MODEL?'/
2
'TYPE Y OR N')
2002 FORMAT('
THE VALUE OF Q MAY BE SPECIFIED IN 3 WAYS:'/
1
'1 CONSTANT Q PER UNIT AREA, (BTU/(HR-FT2)'/
2
'2 CONSTANT Q PER UNIT AREA, (BTU/(HR-IN2)'/
3
'3 Q BASED ON BTU/HR OVER THE ENTIRE SPHERE SURFACE'/
4
'TYPE 1, 2, OR 3")
2003 FORMAT('
TYPE IN THE VALUE OF Q IN ','AI6)
END
C45ISSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SUBROUTINE AREASP(NAREA,JPOS,R,TH,AREA)
CALLED PROM SRCDAT (45)
SNBLKS (47)
SPHCDS (461)
CALLED FROM SPHDIP (4412)
CIRCON (4612)
SUBROUTINE TO COMPUTE AREAS ON SPHERE.
AREAS FOR NODES TO COMPUTE NODAL VOLUMES.
OUTSIDE AREAS FOR SOURCE TERMS (IF ANY)
AREAS FOR CONDUCTOR PATHS.
WHERE:
NAREA = 1, 2 RADIAL AREA, CIRCUMFERENTIAL AREA
JPOS - POSITION OF THETA ANGLE COUNTING FROM SOUTH POLE
R = RADII TO AREA SURFACE
TH = LAYER THICKNESS
AREA = VALUE RETURNED TO CALLING PROGRAM
COMMON/STUFF/ NHTT,PI,CONVY,CONVR,THETAO,DTHETA,NBASOS,ROUTSF,
1 BNCOEF(2)
COMMON /REGION/ NTHETA,NBETAS,BETA,RN,TVAL,TROUT (9),
1 RECS (9),NLAYRS (9),TEMPS (9),THICK (9),
2 THKLAY (9),MATRIS (9),MATHIS (9),RNBNM (9)
LOGICAL REGNS
CHARACTER*9 GEOM
CHARACTER*16 MATNMS
CHARACTER*25 RNBNMS
THETA1-THETAO-JPOS*DTHETA
THETA2-THETAI-DTHETA
IF (NAREA .EQ. 1) THEN
AREA =BETA*R* (COS(THETA1)+COS(THETA2))*DTHETA/2.
ELSE
AREA IN Y DIRECTION, (CIRCUMFERENTIAL)
AREA =BETA*R* (COS(THETA1)+COS(THETA2))*TH/2.
ENDIF
RETURN
END
C46SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SUBROUTINE CONDRS
CRY24510
CRY24520
CRY24530
CRY24540
CRY24550
CRY24560
CRY24570
CRY24580
CRY24590
CRY24600
CRY24610
CRY24620
CRY24630
CRY24640
CRY24650
CRY24660
CRY24670
CRY24680
CRY24690
CRY24700
CRY24710
CRY24720
CRY24730
CRY24740
CRY24750
CRY24760
CRY24770
CRY24780
CRY24790
CRY24800
CRY24810
CRY24820
CRY24830
CRY24840
CRY24850
CRY24860
CRY24870
CRY24880
CRY24890
CRY24900
CRY24910
CRY24920
CRY24930
CRY24940
CRY24950
CRY24960
CRY24970
CRY24980
CRY24990
CRY25000
CRY25010
CRY25020
CRY25030
CRY25040
CRY25050
CRY25060
CRY25070
CRY25080
CRY25090
CRY25100
CRY25110
CRY25120
CRY25130
CRY25140
CRY25150
CRY25160
CRY25170
CRY25180
CRY25190
CRY25200
SUBROUTINE TO GENERATE CONDUCTOR DATA

CALL BL3TTL
CALL SPHCDS (461)
CALL CYLCDS

C
REGNB(4) = FALSE; GENERATE NTHETA BOUNDARY CONDUCTORS, FOR
CONVECTION FROM TANK TO INSIDE TANK WALL, (NODES 1800N TO 100N)

C

IF(REGNS(9)) THEN
WRITE(MODU,2006) LABLE='TANK TO WALL'
ENDIF

CALL RITCND (4613)
CALL RITCND(3,18001,NTHETA, I,IB001,I,1001,1,0,0,1.0,1.0,
LABLE)
ENDIF

CALL BLKEND
RETURN

2006 FORMAT(7X,'REM CONVECTION
CONDUCTORS;
INSIDE TANK TO TANK WALL')
IF (NTP .EQ. 4) THEN
WRITE(MODU, 2304) NG, NA, NB, NAA, FA
ELSE
IF (NTP .EQ. 5) THEN
WRITE(MODU, 2305) NG, NGS, IG, NA, IA, IB, NAA, FA
ELSE
IF (NTP .EQ. i) THEN
WRITE(MODU, 2301) NG, NA, NB, RNAA, RNAB, FA, FB
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ELSE
IF (NTP .EQ. 2) THEN
WRITE(MODU, 2302) NG, NA, NB, KNAA, RNAB, FA, FB
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
RETURN

2301 FORMAT (11X, I6, ', ', I6, ', ', I6, ', ', I6, ', ', 1PE13.6)
2302 FORMAT (7X, 'CAL ', I5, ', ', 2X, I5, ', ', 1PE13.6, 1
   ', 1PE13.6, ', ', 1PE13.6, ', ', 1PE13.6)
2303 FORMAT (7X, 'GEN ', I5, ', ', 3X, I1, ', ', I5, ', ', I2, ', ',
   15, ', ', I2, ', ', 2X, IPEI1.5, ', ', 2PE11.5, ', ', 1PE11.5, ', ', 1PE11.5)
2304 FORMAT (7X, 'SIM ', I5, ', ', 3X, I5, ', ', 3X, I5, ', ', 3X, I5, ', ', 3X,
   1', A, ', I4, ', ', 1PE13.6, 1
   ', 1PE13.6, ', ', 1PE13.6, ', ', 1PE13.6)
2305 FORMAT (7X, 'SIV ', I5, ', ', 3X, I5, ', ', 3X, I5, ', ', 3X, I5, ', ', 3X,
   1', A, ', I4, ', ', 1PE13.6, 1
   ', 1PE13.6, ', ', 1PE13.6, ', ', 1PE13.6)
2306 FORMAT (7X, 'DIV ', I5, ', ', 3X, I5, ', ', 3X, I5, ', ', 3X, I5, ', ', 3X,
   1', A, ', I4, ', ', 1PE13.6, 1
   ', 1PE13.6, ', ', 1PE13.6, ', ', 1PE13.6)
2307 FORMAT (7X, 'DIM ', I5, ', ', 3X, I5, ', ', 3X, I5, ', ', 3X, I5, ', ', 3X,
   1', A, ', I4, ', ', 1PE13.6, 1
   ', 1PE13.6, ', ', 1PE13.6, ', ', 1PE13.6)
END

SUBROUTINE SNBLKS
C
C SUBROUTINE TO READ SINDA CONSTANTS AND GENERATE THE REMAINING
C BLOCKS. (CONSTANTS, ARRAY, EXECUTN, VBLES1, VBLES2, OUTPUT)
C
COMMON/UNITS/ MODU, INPEKO, ISCRCH, SINDA
COMMON/GEOM/ NTYP, NAN, GEOM(2)
COMMON/SUBTS/ SPLIPT, XCUT1, XCUT2, VBLBL1, VBLBL2, OUTBLK
COMMON/VULAGE/ NLU14, NLU15, NTHU41, RINMHN, FCTFUL, RADULG, TVULFT,
   1 CT, LG(3), LIQVAP(3)
COMMON /REGION/ NTHTA, NBETAS, BETA, RIN, TVOL, ROUT(9),
   1 REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
   2 THKLAY(9), MATRLS(9), MATNMS(9), RGNMNS(9)
C
LOGICAL SPLIPT
LOGICAL REGNS, SINDA
LOGICAL TRANSNT, STDYST
C
CHARACTER*1 YN
CHARACTER*1 CT, LG
CHARACTER*3 TUNIX
CHARACTER*6 LIQVAP
CHARACTER*6 XCUT1, XCUT2, VBLBL1, VBLBL2, OUTBLK
C
RETURN
C GET VALUES FOR THE SINDA CONSTANTS
C TIME, TIMEND, DTIMEI, DRLXCA, ARLXCA, NLOOP, OUTPUT
C
C GET TYPE(S) OF EXECUTION ROUTINES TO BE USED. SS OR TRAN.
C USE (FWDACK) FOR TRANSIENT, AND (STDSTL) FOR STEADY STATE
C
C UNITS TO BE USED IN SINDA -- DEGR, IN., HRS., LBS., BTU
C
C COMMON / DATA/ RIN, ROUT, NLAY, NTHETA, TIMEND, OUTPUT, FFLOW,
C 1 TGAS, TLIQ, TWALL, DTIMEI, DRLXCA, ARLXCA, NLOOP
C
C DIMENSION SAREA(200)
C
C DATA NLOOP,DTIMEI,DRLXCA,ARLXCA/ 1000, .005, .005, .005/
C DATA TIMEO,TIMEND,OUTPUT / 0.0, 24., .5 /
C
C DATA BLNK/' '/
C DATA EXECNS/'STDSTL', "FWDACK", 'CNFRDL' /
C
C DATA ARRY12 /'INSIDE', 'OUTSIDE' /
C
C PROMPT USER FOR USER CONSTANTS AND EXECUTION ROUTINES
CALL CLEARS
C
101 PRINT 2001
   CALL READIN(NEXE, 1,4)
   TRANS= .TRUE.
   STDST=.TRUE.
   IF(NEXE .EQ. 1) TRANS=.FALSE.
   IF(NEXE .EQ. 2) STDST=.FALSE.
   IF(TRANS) THEN
      PRINT 2002
      PRINT 20021
   ENDIF
   ENDIF
   ENDIF
   CALL CLEARS
   GO TO 201
   ENDIF
   ENDIF
   CALL CLEARS
   PRINT 20021
   C GET FROM USER WHICH UNITS THE TIME VARIABLES ARE TO BE INPUT
   CALL READAL(1,YN)
   IF(YN .EQ. 'H') THEN
      TCNV=1.0
      TUNITS='HRS'
   ELSE
      IF(YN .EQ. 'M') THEN
         TCNV=60.0
         TUNITS='MIN'
      ELSE
         IF(YN .EQ. 'S') THEN
            TCNV=3600.0
            TUNITS='SEC'
         ELSE
            PRINT 3004
            CALL CLEARS
            GO TO 201
         ENDIF
      ENDIF
   ELSE
      PRINT 3004
      CALL CLEARS
      GO TO 201
      ENDIF
      ENDIF
      ENDIF
      CALL CLEARS
      PRINT 20021
      CALL READE(TOIN)
      TOHRS=TOIN/TCNV
      PRINT 2003, TUNITS
      CALL READE(TEIN)
      TEHRS=TEIN/TCNV
      PRINT 2004, TUNITS
      CALL READE(DTIN)
      DTHRS=DTIN/TCNV
      PRINT 2005, TUNITS, TUNITS
CALL READRE (OUTIN)
OUTHRS=OUTIN/TCNV
IF (OUTHRS.LT. 0.0)
  THEN
    OUTHRS=.25
    OUTIN=OUTHRS*TCNV
ENDIF
ENDIF
CALL CLE_S
PRINT 2006
CALL READRE (DRLXIN)
PRINT 2007
CALL READIN (LOOPIN, I, 99990)

ANY SPECIAL INPUT DATA FOR NTYP = 1, 2
THIS WILL BE KNOWN WHEN THE SUBROUTINES ARE ENTERED INTO THE SYSTEM
BASED ON NTYP AND NAN

IF (SPLIPT) THEN
  CALL SPLINP (471)
  CALL SPLINP (XI, X2, XS, X4)
ENDIF

GENERATE CONSTANTS DATA BLOCK
USER CONSTANTS (1, NLAY), (2, NTHETA), (3, NN1), (4, RIN), (5, ROUT)
CALL BL4TTL
WRITE (MODU, 2402) NTHETA, NBETAS, BETA, RIN, TVOL
IF (SPLIPT) THEN
  WRITE (THE SPECIAL INPUT TO CONSTANTS DATA. USER SUBROUTINE WILL GET THE VALUES FROM THE CONSTANTS BLOCK.
M & K6, KT, KS, K9
WRITE (MODU, 2403) XI, X2, X3, X4
ENDIF
WRITE (MODU, 2404) MATRLS (9)
WRITE (MODU, 2406) TUNITS, TUNITS, TUNITS, TUNITS, I TOIN, TEIN, DTIN,
OUTIN, 2 TOHRS, TEHRS, DTHRS, OUTHRS
WRITE (MODU, 2407) LOOPIN, DRLXIN, DRLXIN
CALL BLKEND

GENERATE ARRAY DATA BLOCK
CALL BL5TTL
ARRAY 1 IS ARRAY OF INSIDE TANK SURFACE AREAS, AT RIN.
ARRAY 2 IS ARRAY OF OUTSIDE SURFACE AREAS, OUTSIDE SURFACE 1, 2 OR 3
IF (.NOT. REGNS (4)) THEN
  RAD=RIN
  NRAD=1
  NARAY=1
  NREGN=1
500 DO 501 J=1, NTHETA
  IF (NTYP .EQ. 1) THEN
    IF (MOD (J, 2) .EQ. 0) GO TO 502
    JJJ=J/2
    CALL AREASP (I, JJJ, RAD, 0., SAREA (JJJ+1))
    NJ=NTHETA-J
    SAREA (NTHETA-JJ)=SAREA (JJJ+1)
  ENDIF
502 CONTINUE
ENDIF
IF (NTYP .EQ. 2) THEN
  AREACYL (472)
ENDIF
501 CONTINUE
WRITE (MODU, 2501) NARAY, NREGN, RNDRMS (NREGN), ARRY2 (NARAY)
DO 510 I=1, NTHETA, 5
  NI=1
  N2=NI+4
  IF (N2 .LE. NTHETA) THEN
    WRITE (MODU, 2502) (SAREA (II), II=NI, N2)
  ELSE
```
NN NTHETA N1 + 1
N2 NTHETA
IF (NN .EQ. 4) WRITE (MODU, 25024) (SAREA (II), II-N1, N2)
IF (NN .EQ. 3) WRITE (MODU, 25023) (SAREA (II), II-N1, N2)
IF (NN .EQ. 2) WRITE (MODU, 25022) (SAREA (II), II-N1, N2)
IF (NN .EQ. 1) WRITE (MODU, 25021) (SAREA (II), II-N1, N2)
ENDIF
510 CONTINUE
WRITE (MODU, 2503)
IF (NARY .EQ. 1) THEN
PUT OUT ARRAY 2: OUTSIDE SURFACE AREAS
RAD ROUT (6)
NRA= 2
NREGN = 1
IF (RENS (2)) NREGN = 2
IF (RENS (3)) NREGN = 3
GO TO 500
ENDIF
ENDIF
C CALL SUB PRPTBL TO GET PROPERTIES FROM THE DATABASE FOR EACH
C MATERIAL, TO PUT THE PROPERTIES INTO DOUBLET ARRAYS, THEN TO
C OUTPUT THE ARRAYS TO THE ARRAY BLOCK OF THE MODEL.
C FOR DIFFUSION NODES MAKE ARRAYS FOR K AND (CP*RHO)
C IF THE PROPERTY FOR A REGION IS LIQUID, AND, THE ULLAGE .GT. 0
C THEN GET PROPERTIES AND OUTPUT TABLES FOR BOTH THE LIQUID AND
C GAS FORMS FOR THAT MATERIAL.
C DO 601 LL = 1, 5
CALL PRPTBL (LL)
601 CONTINUE
IF (SPLPT) CALL PRPTBL (6)
CALL BLKEND
C GENERATE EXECUTION BLOCK
C USER MAY HAVE 2 SUBROUTINE CALLS FROM EXECUTION BLOCK
C GET SUBROUTINE NAMES FROM VECTORS EXECI AND EXEC2, IN SUB MENU2
C CALL B16TTL
C WRITE COMMON STATEMENTS INTO MODEL, EXECUTION BLOCK.
WRITE (MODU, 2602)
IF (.NOT. REGNS (4)) THEN
WRITE (MODU, 2603)
WRITE (MODU, 2604) NTHETA, NTHETA, NTHETA, NTHETA, NTHETA, NTHETA
ENDIF
C COMPUTE DIMENSIONS FOR X ARRAY AND NDIM, AND OUTPUT TO EXEC. BLOCK
C THIS IS BASED ON NO. NODES, NO. CONDS, AND ROUTINES USED, ETC.
C NON- NLAYRS (1)
NARN = 2
DO 550 I = 2, 5
IF (RENS (1)) THEN
NDN = NDN + NLAYRS (1)
NARN = NARN + 1
ENDIF
550 CONTINUE
NDN = NDN + NTHETA
NARN = NARN + NTHETA
NX = 5 * (NDN + NARN)
NX = (NX/100+1)*100
WRITE (MODU, 2605) NX, NX
C OUTPUT FORTRAN CODE TO INITIALIZE THE COMMON BLOCKS
```

WRITE (MODU, 2610)
IF (SPLIPT) WRITE (MODU, 2611)
WRITE (MODU, 2612)
IF (.NOT. REGNS(4)) THEN
WRITE (MODU, 2613)
ENDIF
C IF REGNS=1 AND REGNS=2 AND VBLBL= 5 THEN THE USER ASKED FOR
C TEMPERATURE NODES INSIDE TANK, (NODES 18001),
C IF (FTCTU=0) THEN CHECK TO SEE IF USER
C WANTS TO INPUT A CONSTANT H; THEN CALCULATE G(18000+I)= H*A
C IF (FTCTU=0) THEN USER INPUT 2 TEMPS, (TL & TV), CHECK TO SEE IF USER
C WANTS TO INPUT CONSTANT H'S (HL & HV)
C THEN CALCULATE G(18000+I)= HL*A AND G-HV*A
C IF (.NOT. REGNS(4) .AND. REGNS(9) .AND. VBLBLL.EQ. ' ') THEN
IF (FTCTU .LE. 0.0 OR. FTCTU .GE. 100.0) THEN
PRINT 30021, TEMPS(9), MATNMS(9)
CALL READAL (YN)
ENDIF
WRITE (MODU, 26141) HH
ELSE
PRINT 30022, TEMPS(9), MATNMS(9), THICK(9), MATNMS(9)
CALL READAL (YN)
ENDIF
NLN=NTHU41-I
WRITE (MODU, 26142) HL, HV, NLQN,NTHU41, NTHETA
ENDIF
ENDIF
USER SUBROUTINE CALL GOES HERE, 1 OR 2.
C IF (XCTU1 .NE. BLNK) WRITE (MODU, 6789) XCTU1
GO TO (610, 620, 630, 640), NEXE
C 610 STEADY STATE ANALYSIS
610 WRITE (MODU, 2630) EXECNS(1)
GO TO 660
C 620 TRANSIENT ANALYSIS
620 NEXT=1
621 IF (DTIN .GT. 0.) THEN
C FWDBCK, NEED -- TO, TEND, OUTPUT AND DTIMEI
WRITE (MODU, 26312)
WRITE (MODU, 26312)
WRITE (MODU, 2630) EXECNS(2)
ELSE
C CNFRDL, NEED -- TO, TEND, OUTPUT
WRITE (MODU, 26311)
WRITE (MODU, 2630) EXECNS(3)
ENDIF
GO TO (660, 641), NEXT
C 630 STEADY STATE FOLLOWED BY TRANSIENT
630 WRITE (MODU, 2630) EXECNS(1)
GO TO 620
C 640 TRANSIENT FOLLOWED BY STEADY STATE
640 NEXT-2
GO TO 621
641 WRITE (MODU, 2632)
WRITE (MODU, 2630) EXECNS(1)
660 IF (XCTU2 .NE. BLNK) WRITE (MODU, 6789) XCTU2
CALL BLKEND
C C GENERATE VARIABLES I BLOCK
C CALL BL7TTL
C WRITE COMMON STATEMENTS INTO MODEL, VBL$1 BLOCK.
WRITE(MODU, 2602)
IF(.NOT. REGNS(4) .AND. VBL$1 .NE. BLNK) THEN
WRITE(MODU, 2603) NTHETA, NTHETA, NTHETA, NTHETA, NTHETA
WRITE(MODU, 2701)
ENDIF
IF(VBL$1 .NE. BLNK) WRITE(MODU, 6789) VBL$1
IF(.NOT. REGNS(4) .AND. VBL$1 .NE. BLNK) THEN
WRITE(MODU, 2702)
ENDIF
CALL BLKEND
GENERATE VARIABLES 2 BLOCK
CALL BLBTTL
IF(VBL$2 .NE. BLNK) WRITE(MODU, 6789) VBL$2
CALL BLKEND
GENERATE OUTPUT BLOCK
CALL BLBTTL
IF(OUTBLK .NE. BLNK) WRITE(MODU, 6789) OUTBLK
ELSE
WRITE(MODU, 2901)
ENDIF
IF THIS IS A SINDA MODEL THAT CALLS SUBROUTINES FROM THE BLOCKS
INSERT THESE SUBROUTINES INTO THE MODEL FILE FOLLOWING THE OUTPUT BLOCK. THIS WILL GET THE ROUTINES COMPILED WITH THE PREPROCESSOR OUTPUT.
CALL ENDDAT
C FORMAT STATEMENTS
C
2001 FORMAT('NOW INPUT THE SPECIFIC DATA FOR SINDA/'
1 ' ' THIS SINDA ANALYSIS MAY BE,'
2 ' ' 1 A STEADY STATE ANALYSIS'/'
3 ' ' 2 A TRANSIENT ANALYSIS'/'
4 ' ' 3 STEADY STATE FOLLOWED BY A TRANSIENT'/'
5 ' ' 4 A TRANSIENT FOLLOWED BY STEADY STATE'/'
6 ' ' TYPE IN 1, 2, 3, OR 4')
2002 FORMAT('A TRANSIENT ANALYSIS IS TO BE DONE,/'
1 ' ' THE EXECUTION SUBROUTINE WILL BE EITHER FWDBCK ',
2 ' ' OR CNFRDL'/'
3 ' ' THIS WILL BE DETERMINED BY THE VALUE OF THE TIME',
4 ' ' STEP, (DELTIME)/'
5 ' ' WHICH WILL BE INPUT BELOW.'/')
2003 FORMAT('THE NEXT 4 INPUT VALUES INVOLVE PROBLEM TIME,/'
1 ' ' THESE 4 VALUES MAY BE INPUT IN UNITS OF/',
2 ' ' SECONDS, MINUTES, OR HOURS'/')
2004 FORMAT('NOW TYPE IN THE PROBLEM START TIME ('A3',')
3 ' ' NOW TYPE IN THE PROBLEM END TIME ('A3',')
4 ' ' IF DELTIME IS UNKNOWN, OR IF YOU TYPE ZERO (0 ),'/')
2005 FORMAT('TYPE IN THE TIME STEP, (DELTIME), ('A3',' TO BE USED,,'/
1 ' ' WILL BE USED AND DELTIME WILL BE COMPUTED BY THE PROGRAM')
2006 FORMAT('TYPE IN THE CONVERGENCE CRITERIA, DELTA TEMPERATURE'/
1 ' ' TYPE IN THE CONVERGENCE CRITERIA, DELTA TEMPERATURE'/
2 ' ' TYPE IN THE CONVERGENCE CRITERIA, DELTA TEMPERATURE'/
3 ' ' TYPE IN THE CONVERGENCE CRITERIA, DELTA TEMPERATURE'/
4 ' ' TYPE IN THE CONVERGENCE CRITERIA, DELTA TEMPERATURE'})
SUGGESTED VALUE RANGE .01 TO .001/

IF INPUT VALUE .LE. .005 WILL BE USED.

TYPE IN NLOOP, THE NUMBER OF ITERATION LOOPS,

ALLOWED/

SUGGESTED RANGE OF VALUES 100 TO 1000 /

IF INPUT VALUE IS .LE. 100 WILL BE USED.

NOTE: SOME STEADY STATE CASES MAY NEED NLOOP > 1000.

C 2007 FORMAT(/' TYPE IN NLOOP')

THE NUMBER OF ITERATION LOOPS

' ALLOWED/

SUGGESTED RANGE OF VALUES 100 TO 1000 /

IF INPUT VALUE .LE. 0 >> .005 WILL BE USED.

C 2008 FORMAT(/' TYPE IN NLOOP')

THE NUMBER OF ITERATION LOOPS

' ALLOWED/

SUGGESTED RANGE OF VALUES 100 TO 1000 /

IF INPUT VALUE .LE. 0 >> .005 WILL BE USED.

C 2009 FORMAT(/' TYPE IN NLOOP')

THE NUMBER OF ITERATION LOOPS

' ALLOWED/

SUGGESTED RANGE OF VALUES 100 TO 1000 /

IF INPUT VALUE .LE. 0 >> .005 WILL BE USED.

C 2010 FORMAT(/' TYPE IN NLOOP')

THE NUMBER OF ITERATION LOOPS

' ALLOWED/

SUGGESTED RANGE OF VALUES 100 TO 1000 /

IF INPUT VALUE .LE. 0 >> .005 WILL BE USED.

C 2011 FORMAT(/' TYPE IN NLOOP')

THE NUMBER OF ITERATION LOOPS

' ALLOWED/

SUGGESTED RANGE OF VALUES 100 TO 1000 /

IF INPUT VALUE .LE. 0 >> .005 WILL BE USED.
1 'M TIMEND= XM102'
2 'M OUTPUT= XM104'
26312 FORMAT('M DTIME= XM103')
3 'M TIMEND= 0.0 '
2 'M DTIME= 0.0 '
3 'M OUTPUT= 0.0 '
2701 FORMAT('F',6X,'PTIME- TIMEO'/' 1 'F',6X,'DELTIM-DTIMEU'/' 2 'F',6X,'DO 271 I=I,NTHETA'/' 3 'F',6X,'IMI-I-I'/' 4 'M',6X,'TSURF(I)-T(1001+IMI)'/' 5 'F 270 CONTINUE')
2702 FORMAT('F',6X,'DO 271 I=I,NTHETA'/' 1 'F',6X,'IMI-I-I'/' 2 'M',6X,'T(18001+IMI)-TBDY (I)'/' 3 'M',6X,'Q(18001+IMI)-Q(18001+IMI)+QSURF(I)'/' 4 'M',6X,'G(18001*IMI)-HCOEF(I)*SARIN(I)'/' 5 'F 271 CONTINUE')
2901 FORMAT(IIX,'TPRNTF')
3001 FORMAT(' THE INPUT VALUE FOR TYPE OF ANALYSIS IS OUT OF RANGE'/' 1 'M MUST BE FROM 1 TO 4; TRY AGAIN.')
30021 FORMAT(//' TEMPERATURE INSIDE OF TANK IS DEFINED -',F8.2,' DEG ',AI/ 2 ' WANT TO INPUT HL AND HV TO COMPUTE CONVECTION COEFICIENTS G'H*A?'// 3 ' TYPE IN Y OR N')
30022 FORMAT(//' TYPE IN FILM COEFFICIENT H (BTU/HR-FT2-R)')
3003 FORMAT(//' TYPE IN FILM COEFFICIENT HL (BTU/HR-FT2-R)')
6789 FORMAT('F CALL ',A6)
END

SUBROUTINE SPLINP(XI,X2,X3,X4)
CALLED FROM SNBLS(47)
C SPECIAL INPUT DATA FOR NTYP=1,(SPHERE} OR NTYP=2,(CYLINDER)
C THIS SUBROUTINE MAY NEED TO BE CHANGED WHEN NEW FILL PROCEDURES ARE ADDED TO THE SYSTEM IF INPUTS OTHER THAN THE CURRENT ONES ARE REQUIRED.
C PRESENT INPUTS ARE:
C NTYP=1 AND NAN=3---- SPHERE, THICK WALL FILL
C
COMM/REGION/ MHTHA,NBETAS, BETA, RIN, TVOL, ROUT(9),
1 REGNS(9),NLAYRS(9), TPS(9),THICK(9),
2 THKLAY(9),MATRLS(9), MATNMS(9), RGNNMS(9)
COMMON/GEOMTY/ NTYP,NAN,GEOM(2)

C LOGICAL REGNS
C
CHARACTER*1 YN
CHARACTER*8 GEOM
CHARACTER*16 MATNMS
CHARACTER*25 RGNNMS
C
IF (NTYP .EQ. 1) THEN
IF (NAN .EQ. 3) THEN
A FILL ON A THICK WALL SPHERE NTYP=1, NAN=3
C LIQUID FLOW RATE (LB/HR)
C LIQUID TEMPERATURE (DEG F/R)
C VAPOR TEMPERATURE (DEG F/R)
C FILL THE TANK HOW FULL? (%)
NAME OF LIQUID (FROM MATERIAL DATA BASE)

CALL CLEAR
PRINT 208
CALL READRE(FFLOW)
PRINT 209,MATNMS(9)
CALL READRE(TLIQ)
PRINT 210,MATNMS(9)
CALL READRE(TGAS)
PRINT 211
CALL READAL(I,YN)
IF(YN .LE. 'Y') THEN
   CALL MATMNU(6)
ENDIF
X1 = FFLOW
X2 = TLIQ
X3 = TGAS
X4 = PCTFIL
ENDIF
ENDIF
RETURN

C FORMATS
2008 FORMAT('SPECIAL INPUT FOR TANKFILL PROCEDURES/
   TYPE IN FLUID FLOW RATE (LB/HR)'
   TYPE IN FLUID TEMPERATURE (DEG R)'/
   TYPE IN VAPOR TEMPERATURE (DEG R)'
   DO YOU NEED MATERIAL PROPERTIES FOR THE LIQUID?/
   TYPE Y OR N')
2009 FORMAT('NOTE: TEMPERATURE UNITS MUST BE DEG R.')
2010 FORMAT('')
2011 FORMAT('FILL THE TANK HOW FULL? TYPE IN PERCENT TO FILL')
2012 FORMAT('')
END

SUBROUTINE INSERT
CALLED FROM

FILEDEF THE PROPER SOURCE FILE
FILDEF-FILDI//XCUTI//EM
CALL DOJCL(FILDEF)
OPEN(UNIT-36,FILE-'CRYSUBS')

READ 35,1001,END-500) LI,LINE
IF(LI .EQ. ' ') THEN
   WRITE(MODU, 2001)LINE
WRITE FORTRAN "END" FOLLOWING LAST LINE OF OUTPUT BLOCK
WRITE(MODU,2000)
FILEDEF THE PROPER SOURCE FILE
FILEDEF-FILDI//XCUTI//EM
FILEDEF-FILDI//XCUTI//' A'
CALL DOJCL(FILDEF)
OPEN(UNIT-36,FILE='CRYOSUBS')
READ(36,1001,END=500) LI,LINE
   WRITE(MODU,2001)LINE
SUBROUTINE GEOPLT

CALLED FROM SIMTRU(4)

C PLOT ROUTINES FOR CRYOTRAN

C THE PLOT ROUTINES ARE CALLED FROM SUBROUTINE SIMTRU (22)
C THEY PRODUCE PLOTS OF THE GEOMETRY, SPHERE OR CYLINDER,
C THAT THE USER HAS DEFINED WITHIN THE CRYOTRAN SYSTEM.
C THE CURRENT PLOTTING SYSTEM BEING USED ARE:
C 1. THE DISPLA SYSTEM TO PLOT SPHERE GEOMETRIES
C 2. THE DISPLA SYSTEM TO PLOT CYLINDER GEOMETRIES
C THE PLOTTING SUBROUTINES ARE IN THE FILE CRYOPLOT FORTRAN.
C
C SUBROUTINE TO PLOT THE GEOMETRY
C DISPLA PLOTTING CALLS.
C COMMON/UNITS/ MODU, INPEKO, ISCREH, SINDA
C COMMON/GEOMTY/ NTYP, NAN, GEOM(2)
C CHARACTER*1 YN, YES
C CHARACTER*8 GEOM
C LOGICAL PLOTSS
C DATA YES '/Y'/
C DATA PLOTSS '/.FALSE./'
C
CALL CLEARS
PRINT 2001
READ(5,1001) YN
WRITE(INPEKO, IO01) YN
IF(YN .EQ. YES) THEN
PLOTSS '.TRUE.'
CALL PLTSPH (481)
ENDIF
RETURN
ENTRY PLTDUN

CALLED FROM MAIN PROGRAM WHEN EXITING CRYOTRAN
C TO CALL DONEPL (DISPLA SUBROUTINE IN PLOT SUBROUTINE)
C IF PLOTS HAVE BEEN PRODUCED
C
C ENTRY PLTDUN

1001 FORMAT(A1)
2001 FORMAT(/' IN THE PLOTTING ROUTINE, NTYP=1; 2D SPHERE WEDGE'/
1 ' DO YOU WANT A PLOT OF THIS GEOMETRY?'/
2 ' TYPE Y OR N'/)
C SUBROUTINE VMINTR (MAINNM)
C CALLED FROM MAIN (00)
C ENTRY TO EXECUTE ANALYSIS ROUTINES ON VM INTERACTIVELY
C THIS SECTION OF THE CODE MUST BE CHANGED WHENEVER A NEW
INTERACTIVE ANALYSIS PROGRAM IS ADDED TO THE SYSTEM. THE
NAMES OF THESE PROGRAMS WILL BE ADDED TO THE ARRAY MAINNM
AND THE CORRESPONDING VALUE IN ARRAY NSRUNM WILL BE - 2.
FOR THESE PROGRAMS THE OUTPUT WILL GO TO BOTH THE SCREEN
AND A DISK FILE NAMED "PROGRAM OUTPUT", THE FILE PROGRAM OUTPUT
IS FORTRAN UNIT 17 AND MUST BE DEFINED PRIOR TO EXECUTION
OF THE PREPROCESSOR PROGRAM. IN THIS INSTANCE THIS FILE IS
FILEDEF'ED IN THE VM EXEC THAT STARTS THE EXECUTION. THE EXEC
FILE NAME IS "RUNCRYO EXEC".

FOR THESE PROGRAMS THE OUTPUT WILL GO TO BOTH THE SCREEN
AND A DISK FILE NAMED "PROGRAM OUTPUT". THE FILE
IS FORTRAN UNIT 17 AND MUST BE DEFINED PRIOR TO EXECUTION
OF THE PREPROCESSOR PROGRAM. IN THIS INSTANCE THIS FILE IS
FILEDEF'ED IN THE VM EXEC THAT STARTS THE EXECUTION. THE EXEC
FILE NAME IS "RUNCRYO EXEC".

INSERT ENOUGH INPUT INFORMATION TO DETERMINE MATERIAL PROPERTIES
NEEDED FOR THE REQUESTED ANALYSIS ROUTINE,
THEN CALL MATERIAL SUBROUTINES TO GET THE NECESSARY MATERIAL
PROPERTIES FROM THE DATABASE AND INTO THE REQUIRED FORM.
THE ANALYSIS ROUTINES MUST BE CHANGED TO CALL PROPERTY ROUTINES
WITHIN THE SYSTEM.

COMMON/GEOMTY/ NTYP,NAN, GEOM(2)
CHARACTER*6 MAINNM
CHARACTER*8 GEOM

CALL CLEARS
IF(NAN .EQ. 1) CALL NVFILL
IF(NAN .EQ. 2) CALL CHILL
IF (NAN .EQ. 3) CALL TARGET
PRINT 2004,MAINNM
RETURN

C FORMAT STATEMENTS
2004 FORMAT///' END OF PROGRAM ',A6/
1 ' THE OUTPUT IS IN A FILE NAMED "PROGRAM OUTPUT",'/
2 ' RETURNING TO MAIN PROGRAM OF SYSTEM.'/
END

COMMON/GEOMTY/ NTYP,NAN, GEOM(2)
COMMON/UNITS/ MODU, INPEKO, ISCRCH, SINDA
COMMON/SUBRTS/ SPLIPT, XCUTI, XCUT2, VBLBLI, VBLBL2, OUTBLK
LOGICAL SPLIPT
CHARACTER*6 XCUTI, XCUT2, VBLBLI, VBLBL2, OUTBLK
CHARACTER*8 GEOM
LOGICAL UNICOS

RUN AN ANALYSIS WITHOUT A SINDA MODEL GENERATED BY CRYOTRAN.
TEST VARIABLE NRUNON.
IF NRUNON = 1 RUN ANALYSIS ON CRAY.
IF NRUNON = 2 RUN ANALYSIS ON VM IN INTERACTIVE MODE.
THE ANALYSIS IS A SUBROUTINE OF THIS SYSTEM,
AND WILL BE CALLED FROM AN ENTRY (VMINTR) IN
SUBROUTINE MENU2. VMINTR IS CALLED FROM MAIN.
IF NRUNON = 3 RUN ANALYSIS ON VM IN BATCH MODE.

IF(NRUNON .EQ. 1) THEN
THIS ANALYSIS IS TO GO ON THE GRAY
PREPARE A FILE NAMED CRYOTRAN MODEL WHICH WILL
CONTAIN CRAY JCL AND INPUT DATA TO SUBMIT TO CRAY
TO RUN AN ANALYSIS WITH NO SINDA.
CALL GET THE CRAY JCL INFO FROM THE USER
GETJCL (41)
CALL GETJCL(UNICOS)
CALL WRITE FIRST PART OF FILE 1 ON UNIT MODU, CRAJCL FILE. RITJCL(41)
CALL RITJCL(UNICOS)
IF (UNICOS) THEN
C THIS BLOCK FOR UNICOS JCL
CALL INDATI TO DETERMINE WHERE THE DATA FILE RESIDES
C WHEN DATA FILE RESIDENCE IS FOUND, IF NOT STORED ON CRAY
C
INDAT2 WILL DETERMINE LOCATION AND WRITE JCL TO GET IT.
CALL INDAT1 (61)
ELSE
C THIS BLOCK FOR COS JCL
C WRITE CRAY COS JCL TO UNIT MODU TO COMPILE MAIN PROGRAM
WRITE(MODU,3001)
WRITE(MODU,3002)
NOW GET INPUT DATA FOR THE ANALYSIS PROGRAM INDATI (39A)
CALL INDATI(UNICOS)
WRITE(MODU, 3003)
X_UTI
WRITE LAST PART OF CRAY COS JCL
CALL RITJC2
WRITE A MAIN PROGRAM TO UNIT MODU WHICH CALLS ON THE DESIRED
ANALYSIS SUBROUTINE
WRITE (MODU,3010) XCUT2, XCUT1
CALL INDAT2
ENDIF
ENDIF
C IF(NRUNON .EQ. 2) THEN
C THIS ANALYSIS IS RUN ON THE VM SYSTEM IN INTERACTIVE MODE.
C CALL VMINTR THIS CALL IS IN THE MAIN PROGRAM.
C ENDIF
C IF(NRUNON .EQ. 3) THEN
C THIS ANALYSIS IS RUN ON THE VM SYSTEM IN BATCH MODE.
C THIS FEATURE IS NOT YET IN THE SYSTEM.
C ENDIF
RETURN
C FORMAT STATEMENTS
C FORMAT STATEMENTS TO PRODUCE COS RUNSTREAM
3001 FORMAT ('CFT77.')
3002 FORMAT ('ACCESS,DN=CRPROC,PDN=CRYOTRANPROCS,ID=CFTO,OWN=CRYOLIB.'/
1 'LIBRARY,DN=":CRPROC."')
3003 FORMAT (A6,')
3010 FORMAT ('C MAIN PROGRAM - THIS ANALYSIS IS','A20/
2 6X,'CALL ',A6,/ 3 6X,'STOP'/ 6X,'END'/ 'EOF.')
END
C COMMON/UNITS/ MODU, INPEKO, ISCRCH, SINDA
SUBROUTINE INDAT1(UNICOS)
CALLED FROM NOSIND (6)
C SUBROUTINE TO GET THE INPUT DATA FOR AN ANALYSIS PROGRAM.
C SUBROUTINE HAS 2 ENTRY POINTS -- INDAT1 AND INDAT2
C ENTRY INDAT1 INTERROGATES USER AS TO SOURCE OF INPUT DATA
C IF DATA IS ON CRAY, WRITE ACCESS AND ASSIGN STATEMENTS (CRAY JCL).
C ENTRY INDAT2 USED WHEN DATA IS ON VM OR TYPED IN AT RUN TIME
C AND WRITES FILE 3 TO UNIT 10
C IF DATA IS ON VM IT MUST!!! BE LRECL=80, AND REC FM=F;
C THAT IS, FIXED FORMAT AND LINE LENGTH =80.
C COMMON/UNITS/ MODU, INPEKO, ISCRCH, SINDA
CHARACTER*2 NCBCD
CHARACTER*20 FNFTFM
CHARACTER*15 PDN,PID,POWN
CHARACTER*20 FILCOM
CHARACTER*40 FILDEF
CHARACTER*63 FMTOC
CHARACTER*80 DLINE
C
CHARACTER*46 XEDITF
CHARACTER*6 XEDITI
CHARACTER*20 XEDIT2
DATA XEDITI/'XEDIT '/
DATA XEDIT2/"
1A15,'/'"
DATA FILCOM/'FILEDEF VMDATA DISK '/

CALL CLEAR
11 PRINT 2005
CALL READIN(NINPD, 1, 4)
IF(NINPD .EQ. 4) RETURN
GET THE INPUT DATA IF NINPD = 1, DATA STORED ON CRAY
IF(NINPD .EQ. 1) THEN
INPUT DATA STORED ON CRAY, ACCESS FILE, COPY TO UNIT INPUT
IF(UNICOS) THEN
PRINT 2007
CALL READAL(2, PDN)
CALL RITJC3(6, PDN)
ELSE
PRINT 2010
CALL READAL(2, PDN)
PRINT 2011
CALL READAL(2, PID)
PRINT 2012
CALL READAL(2, POWN)
CALL NOCHR3(PDN, 'PDN', 15, NC, NCBCD)
FTOC(27:28)=NCBCD(1:2)
CALL NOCHR3(PID, 'ID', 8, NC, NCBCD)
FTOC(38:39)=NCBCD(1:2)
CALL NOCHR3(POWN, 'OWN', 15, NC, NCBCD)
FTOC(50:51)=NCBCD(1:2)
WRITE(MODU, FTOMC) PDN, PID, POWN
WRITE(MODU, 3002)
ENDIF
CALL RITJC3(NINPD, PDN)
CALL RITJC4(NINPD)
ENDIF
RETURN

ENTRY INDAT2
IF(NINPD .EQ. 2) THEN
INPUT DATA STORED ON VM, ACCESS FILE, READ AND COPY TO UNIT 10
52 PRINT 2020
CALL READAL(2, FNFTFM)
CALL DOJCL(FILDEF)
C
FILEDEF=FILCOM//FNFTFM
CALL READAL(FILDEF)
C
READ UNIT 36, FNFTFM
OPEN UNIT 36, FNFTFM
OPEN UNIT 36, FILE='VMDATA ', IOSTAT=105, STATUS='OLD ', ERR=65
C
PRINT *, OPEN UNIT 36 VMDATA , IOSTAT=105, I05
INQUIRE(FILE='VMDATA ', IOSTAT=105, EXIST=EXS, OPENED=OPN, RECL=IRCL)
PRINT *, INQUIRE ON VMDATA FILE ', 105, EXS, OPN, IRCL
C
PRINT *, ERROR WHEN ATTEMPTING TO OPEN UNIT 36, I05=, I05
PRINT *, FILE ('FNFTFM,' ) DOES NOT EXIST
CLOSE (UNIT=36)
CALL DOJCL('FILEDEF VMDATA CLEAR')
PRINT *, 'TYPE IN NAME OF DATA FILE ON VM'
GO TO 52
75 CONTINUE
CALL RITJC4(NINPD)
CLOSE (UNIT=36)
CALL DOJCL('FILEDEF VMDATA CLEAR')
ENDIF
IF(NINPD .EQ. 3) THEN
C INPUT DATA TO BE TYPED IN AT THIS POINT AND WRITTEN TO UNIT 10
PRINT 2030
CALL RITJC3(NINPD, 'TERMINAL ')
101 READ(5,1003) DLINE
WRITE(INPEKO, 1003) DLINE
IF(DLINE(1:9).NE.'ENDOFMYDATA') THEN
WRITE(MODU, IO03) DLINE
GO TO 101
ENDIF
ENDIF
CALL RITJC4(NINPD)
ENDIF
PRINT 2040
RETURN
C FORMATS
1003 FORMAT (A80)
2005 FORMAT ('NOW WE NEED THE INPUT DATA FOR THE ANALYSIS'/
1 'THIS INPUT DATA CAN BE:'/
2 '1 STORED ON CRAY'/
3 '2 STORED ON VM'/
4 '3 TYPED IN NOW'/
5 '4 NO INPUT DATA FOR THIS ANALYSIS'/
6 'TYPE IN 1 2 3 OR 4')
2007 FORMAT ('ANALYSIS INPUT DATA IS STORED ON CRAY'/
1 'IT MUST BE IN YOUR HOME DIRECTORY'/
2 'TYPE IN THE FILE NAME OF THE INPUT DATA')
2010 FORMAT ('ANALYSIS INPUT DATA IS STORED ON CRAY'/
1 'TYPE IN (PDN) THE PERMANENT DATASET NAME'/
2 'TYPE IN (ID) THE ID OF PDN.')
2012 FORMAT ('ANALYSIS INPUT DATA IS STORED ON VM'/
1 'NOW WE NEED FILE NAME; FILE TYPE; FILE MODE'/
2 'TYPE IN FN FT FM')
2030 FORMAT ('THE ANALYSIS INPUT DATA IS TO BE TYPED IN NOW'/
1 'START TYPING IN THE INPUT DATA.'/
2 'WHEN ALL THE DATA IS TYPED IN THE NEXT LINE MUST'/
3 'BE ENDOFMYDATA'/
4 'THIS WILL STOP THE READING AND CONTINUE THE PROGRAM.')
2040 FORMAT ('THE INPUT DATA IS NOW ALL IN.')
3002 FORMAT ('ASSIGN,DN=INPUTD,A=FT05.')
APPENDIX E

CryoTran Program Listings

Part II  CRYOSPHR FORTRAN
SUBROUTINE SFEERE
CALLED FROM C
C
SUBROUTINE TO INPUT SPECIFIC DATA FOR A SPHERE. (NTYP=1)
THIS ROUTINE IS USED FOR REGION 1 ONLY.
C
REGION WIDTH, TEMP, MATERIAL, NO. LAYERS THRU
COMMON /REGION/ NTETHA,NBETAS,BETA,RIN,TVOl,ROUT(9),
1 REGNS(9),NLAYRS(9),TEMPS(9),THICK(9),
2 THKLAY(9),MATRLS(9),MATNMS(9),RGNNMS(9)
COMMON/STUFF/ NHTH,PI,CONVY,
CONVR,
THETA0,DTHETA, NBASOS,ROUTSF,
1 BNCOEF(2)
COMMON/GEOMTY/ NTYP,NAN,
GOEM(2)
LOGICAL REGNS
CHARACTER*8 GEOM
CHARACTER*16 MATNMS
CHARACTER*25 RGNNMS
NOW INPUT THE DATA TO GET SPHERE RADII, VOLUME AND WALL THICKNESS.
CALL CLEARS
PRINT 2001
PRINT 2002
CALL TO READIN, READRE, READAL (02)
CALL READIN(NN,1,5)
GO TO (51,52,51,53), NN
PRINT 2004
CALL READRE(RIN)
TVOLN=4./3.*PI*RIN**3
TVOl=TVOLN/1728.
GO TO 60
PRINT 2005
CALL READRE(TVOL)
RIN=((.75*TVOL/PI)**(1./3.))*12.
TVOLN=TVOL/1728.
GO TO 60
PRINT 2006
CALL READRE(ROUT(1))
GO TO (61,62,61,62,64), NN
PRINT 2007
CALL READRE(WTHICK)
GO TO (65,65,65,65,65), NN
PRINT 2008
CALL READRE(THKLAY(NL) )
GO TO 70
THICK(NL)=WTHICK
IF(ROUT(1).LE.RIN .OR. TVOL .LE. 0.0 .OR. WTHICK .LE. 0.0)THEN
CALL CLEARS
PRINT 3002
GO TO 45
ENDIF
WE NOW HAVE RIN, ROUT, TMKVOL AND WALLTHICK FOR TANKWALL, (REGION 1)
CALL CLEARS
PRINT 3003
PRINT 3004
PRINT 3005
PRINT 3006
ROUTE(6)=RIN
THICK(1)=WTHICK
IF(ROUT(1).LE.RIN .OR. TVOL .LE. 0.0 .OR. WTHICK .LE. 0.0)THEN
CALL CLEARS
PRINT 3007
GO TO 45
ENDIF
200
The following code snippet is from a FORTRAN program. It involves reading input data for a sphere and determining where ullage (empty space) is located within the sphere. The code also checks if certain conditions are met and handles them appropriately. Here is the text converted to a readable format:

```fortran
C NOW GET NTHETA, NO. OF NODES ALONG CIRCUMFERENCE, SOUTH POLE TO NORTH
C SOME OF THIS CODE WILL NEED TO BE CHANGED WHEN WE GO TO A 3D CONFIG.

PRINT 2008
CALL READIN(N, theta, 1, 100)
IF(N .LT. 5) N = 20
THETA = PI / N

CALL CLEAR
PRINT 2010
CALL READIN(N, bE, 1, 8)
IF(N .LT. 0 OR. N .GT. 8) N = 6
BE = 2 * PI / N

CALL CLEARS
PRINT 2010
CALL READIN(N, bE, 1, 8)
IF(N .EQ. 1) THEN
C ENDIF
C
C THE REMAINING INPUT FOR REGION 1 IS OBTAINED FROM SUB RGNL(42)
C (CALLED FROM SUB REGN1 (31) )
C RETURN

2001 FORMAT(/' NOW INPUT SPECIFIC DATA FOR THIS SPHERE.'/,
'1' ' INPUT DATA TO DEFINE THE SPHERE MAY BE ANY ONE OF:'/,
'2' ' RIN (IN.) AND ROUT (IN.)'/'
'3' ' TANK VOL. (CU.FT.) AND WALL THICKNESS (IN.)'/'
'4' ' TANK VOL. (CU.FT.) AND ROUT (IN.)'/'
'5' ' RIN (IN.) AND WALL THICKNESS (IN.)'/'
'6' ' ROUT (IN.) AND WALL THICKNESS (IN.)'')
2002 FORMAT(/' ENTER A NUMBER 1 - 5')
2004 FORMAT(/' ENTER INSIDE TANK RADIUS, RIN(IN.).')
2005 FORMAT(/' ENTER TANK VOLUME (CU.FT.).')
2006 FORMAT(/' ENTER ROUT (IN.).')
2007 FORMAT(/' ENTER WALL THICKNESS (IN.).')
2008 FORMAT(/' ENTER NUMBER OF NODES ALONG CIRCUMFERENCE.'/,
'1' ' OF THE SPHERE.'/,
'2' ' SOUTH POLE TO NORTH POLE.'/,
'3' ' IF VALUE INPUT IS < 5, 20 WILL BE USED AS A DEFAULT.')
C2009 FORMAT(/' THIS IS A 3D CONFIGURATION,'/,
'1' ' INPUT NO. OF WEDGES AROUND THE SPHERE, DEFAULT = 6.')
2010 FORMAT(/' THIS IS A 2D ANALYSIS, THE WEDGE ANGLE = 1 RAD.')
3001 FORMAT(/' THE GEOMETRY FOR THIS ANALYSIS IS A SPHERE WITH'/,
'1' ' RIN = F8.3, ROUT = F8.3, IN., AND',
'2' ' WALL THICKNESS F7.4, IN.')
3002 FORMAT(/' ERROR IN TYPING DATA TO DEFINE THE SPHERE.'/,
'1' ' INCONSISTENT VALUES WERE INPUT, TRY AGAIN.')
END

C432SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SUBROUTINE ULLGET
CALLED FROM RGN2T5 (43) SPHNS(441)
C
C IF THERE IS ULLAGE DETERMINE
C WHERE THE ULLAGE IS
C WHICH NODES ARE ULLAGE
C AND WHICH NODES ARE LIQUID
C
C IF PCTFUL < 100%, THEN
C IF REGNS 4 & 5 ARE TRUE AND IF PCTFUL < 100% THEN SOME NODES ARE VAPOR
C COMPUTE WHICH NODES ARE ULLAGE (VAPOR) NODES
C ULLAGE MAY BE AT THE CENTER OR AT THE TOP.
C COMPUTE NLUL4, NLUL5 AND NTHU41
C
C COMMON /REGION/ NTHETA, NBETAS, BE, RIN, T, VOL, ROUT(9),
1 REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
2 TOLAY(9), MATLMS(9), RNMS(9), RINGNS(9)
C COMMON/StUFF/ NHMT, PI, CONVY, CONVR, THETA0, DTHETA, DBASOS, ROUTSF,
1 BNGOF(2)
C COMMON /ULLAGE/ NLUL4, NLUL5, NTHU41, RINGH, PCTFUL, RADULG, TVULFT,
1 CT, LG(3), LIQVF(3)
C
C LOGICAL REGNS
C CHARACTER*1 CT, LG

201
CHARACTER*6 LIQVAP
CHARACTER*16 MATNMS
CHARACTER*25 RGNMMS
C
DIMENSION H(3)
C
10 NLUL4=0
NLUL5=0
C
PRINT *, ' IN ULLGET, NR, THICK(4)=',NR,THICK(4)
IF(CT .NE. 'F') THEN
TVLQFT-TVOL*PCTFUL/100.
TVULFT-TVOL-TVLQFT
IF(CT .EQ. '0' .OR. CT .EQ. 'O' ) THEN
C TANK IS EMPTY, ALL NODES ARE VAPOR NODES.
NLUL4= NLAYRS(4)
NLUL5= NLAYRS(5)
RETURN
ENDIF
IF(CT .EQ. 'C') THEN
FOR THE ULLAGE AT THE CENTER, CT='C', ENTIRE LAYERS WILL BE VAPOR
C THIS SECTION DETERMINES WHICH LAYERS ARE VAPOR
C RADULG=((.75*TVULFT/PI)**(1./3.))5*I2.
IF(RADULG .LE. ROUT(5)) THEN
RATIO=RADULG/THKLAY(5)
FIXRAT=NULUL
IF(RATIO-FIXRAT .GE. 0.5) NLUL5=NULUL+1
IF(NLUL5 .GT. NLAYRS(5)) NLUL5=NLAYRS(5)
ELSE
PRINT 3001, RADULG, ROUT(5), TVOL, TVLQFT
ENDIF
ENDIF
C END IF BLOCK FOR CT = 'C'
ELSE
IF(CT .EQ. 'T') THEN
C ULLAGE IS AT THE TOP OF THE SPHERE AND THIS IS A (ZERO), G CASE
C COMPUTE NODES THAT ARE VAPOR, FOR THIS CASE, ULLAGE AT THE TOP,
C A THIN LAYER OF LIQUID WILL BE AT THE WALL.
C THIS OPTION IS NOT YET AVAILABLE
PRINT 3002, CT
CALL ENTRY ULLIN2 IN ULLINP (431)
GO TO 10
ELSE
C END IF BLOCK FOR CT = 'T'
ENDIF
C
CALL ENTRY ULLIN2 IN ULLINP (431)
GO TO 10
ELSE
C END IF BLOCK FOR CT = 'T'
ENDIF
C
CALL CUBIC(P,Q,R,NROOTS,H)
C FIND THE CORRECT ROOT

RTEST=RIN
IF(PCTFUL .LE. 50) RTEST=RIN
DO 50 I=1,3
IF(H(I) .GT. 0.0 .AND. H(I) .LT. RTEST) GO TO 55
50 CONTINUE
PRINT *, 'ERROR, ROOT OF CUBIC NOT FOUND'
STOP

55 H=H(I)
RIN=H=H
C COMPUTE NTHU41, NUL4 AND NUL5 FOR THIS ULLAGE
IF(HH .LT. RIN) THEN
C %FULL > 50
PHI=ACOS(RADULG/RIN)
NTHU41=(PI-PHI)/DTHETA+1.5
IF(REGS(5) .AND. RADULG .LE. ROUT(5)) THEN
NUL4=NLAYRS(5)-RADULG/THKLAY(5)
NUL4=NLAYRS(4)
ELSE
RTMP=RADULG-ROUTE(5)
NUL4=NLAYRS(4)-RTMP/THKLAY(4)+1
IF(NUL4 .LT. NULAYRS(4)) NUL4=0
ENDIF
ELSE
C %FULL <= 50 AND HH >= RIN
RADULG=HH-RIN
PHI=ACOS(RADULG/RIN)
NTHU41=(PI-PHI)/DTHETA+0.5
IF(REGS(5) .AND. RADULG .LE. ROUT(5)) THEN
NUL4=NLAYRS(4)
TEMP=ROUTE(5)-RADULG
NUL4=TEMP/THKLAY(5)
ELSE
RTMP=ROUTE(4)-RADULG
NUL4=RTMP/THKLAY(4)
NUL4=0
ENDIF
ENDIF
RADULG=PHI
C END IF BLOCK FOR CT = 'I'
ELSE
CT IS NOT 'C', 'T', NOR 'I', AND %FULL < 100.
C SOMETHING IS WRONG, INPUT ULLAGE INFO AGAIN.
PRINT 3003
CALL ENTRY ULLIN2 IN ULLINP (431)
GO TO 10
ENDIF
ENDIF
ENDIF
ENDIF
C PRINT *, 'ULLAGE CALCULATIONS'
C PRINT *, PCTFUL, TVOL, TVLQFT, TVULFT, RADULG
RETURN
C FORMAT STATEMENTS

3001 FORMAT(' ** ERROR ****')
1 ' RADIUS OF ULLAGE IS GREATER THAN TANK RADIUS'/
2 ' RADIUS RADIUS', F8.2, ' IN.; TANK RADIUS=', F8.2, ' IN.'/CRYO2690
3 ' TANK VOL=', F8.2, '; LIQ VOL=', F8.2, '; VAP VOL=', F8.2, ' (FT**3)'/CRYO2700
3002 FORMAT(' ** ERROR ****')
1 ' THE TYPE OF ULLAGE REQUESTED, (', A1, ') '/CRYO2710
2 ' LOW-G CASE WITH ULLAGE AT TOP IS NOT YET AVAILABLE.'/CRYO2730
3 ' RE-INPUT THE ULLAGE INFORMATION'/CRYO2740
4 ' OR TYPE IN Q TO STOP')
3003 FORMAT(' ** ERROR ****')
1 ' THE POINTER DESIGNATING THE TYPE OF ULLAGE REQUESTED'/CRYO2770
2 ' IS NOT ONE OF THE ACCEPTABLE VALUES.'/CRYO2780
3 ' RE-INPUT THE ULLAGE INFORMATION'/CRYO2790
SUBROUTINE CUBIC (P, Q, R, NROOTS, X)

DIMENSION X(3)

TOPIO3 = 3.14159/3.0
DATA TOPIO3/2.094395102/

PO3 = P/3.
PSQ = P*P
A = (3.*Q-PSQ)/3.
B = (2.*P*PSQ-9.0*P*Q+27.*R)/27.0
ACO27 = A*A*A/27
BSO4 = B*B/4
RADD = BSO4+ACO27

IF (RADD .LT. 0.) THEN
  C KADD < 0, 3 REAL UNEQUAL ROOTS.
  COSPHI = SQRT(BSO4/(-ACO27))
  PHI = ACOS(COSPHI)
  PHI03 = PHI/3.
  XX = SIGN(2.*SQRT(-A/3.), -B)
  X(1) = XX*COS(PHI03)-PO3
  X(2) = XX*COS(PHI03+TOPIO3)-PO3
  X(3) = XX*COS(PHI03+2.*TOPIO3)-PO3
  NROOTS = 3
ELSE
  C KADD > 0, 1 REAL, 2 IMAGINARY ROOTS. RETURN ONLY 1 REAL ROOT.
  TANPHI = (0.5*ATAN(I./SQRT(RADD)))**0.3333333
  COT2PH = (I. - TANPHI* TANPHI) / (2.*TANPHI)
  X(1) = SIGN(2.*SQRT(A/3.)*COT2PH,B)
  NROOTS = 1
ENDIF
ELSE
  IF (RADD .EQ. 0.0) THEN
    C RADD = 0, 3 EQUAL ROOTS.
    X(1) = -2.*SQRT(-A/3.)
    X(2) = X(1)
    X(3) = X(1)
    NROOTS = 3
  ELSE
    C RADD > 0, 1 REAL, 2 IMAGINARY ROOTS. RETURN ONLY 1 REAL ROOT.
    TANPHI = (0.5*ATAN(1./SQRT(RADD)))**0.3333333
    COT2PH = (1.-TANPHI*TANPHI)/(2.*TANPHI)
    X(1) = SIGN(2.*SQRT(A/3.)*COT2PH,B)
    NROOTS = 1
  ENDIF
ENDIF
ENDIF
ENDIF
RETURN
END
C CHARACTER*16 MAINMS
C CHARACTER*25 RGNMMS
C
C DATA ULTYPE /'LOW-G','LOW-G','1-G'/
C DATA ULWERE /'CENTER','TOP','TOP & FLAT'/
C
C IF THERE IS ULLAGE DETERMINE
C WHERE THE ULLAGE IS
C WHICH NODES ARE ULLAGE
C AND WHICH NODES ARE LIQUID
C
C IF PCTFUL < 100%, THEN
C IF REGNS 4/5 ARE TRUE AND IF PCTFUL < 100% THEN SOME NODES ARE VAPOR
C COMPUTE WHICH NODES ARE ULLAGE (VAPOR) NODES
C ULLAGE MAY BE AT THE CENTER OR AT THE TOP.

CALL

IF(REGNS(4)) CALL ULLGET

GENERATE THE NODES FOR SPHERICAL WEDGE(S); 5 REGIONS /WEDGE;
FOR A 2D MODEL NBETAS=1; FOR A 3D MODEL NBETAS>1.
DO 100 NR=1,NBETAS

NBMINT=(NR-1)*NTHETA

C GENERATE THE NODES AND WRITE THEM TO UNIT 10 FOR ALL REGIONS.
DO 101 NR=1,5

C COMPUTE NODBASE FOR CURRENT REGION.
NODBAS=2000*NR+ NBMINT
TMPTR=TEMPS(NR)

C SURFACE (ARITHMETIC) NODES, REGION 1, INSIDE TANK WALL.
NMI=NODBAS-1000
IF(NR .EQ. 1) THEN
WRITE (MODU, 2001)

CALL

CALL SETUPA(NR,NM1)

ENDIF

C SURFACE (ARITHMETIC) NODES BETWEEN REGION 4 AND REGION 5.
WRITE (MODU, 2001)

CALL

CALL SETUPA(NR,NR,NM1)

ENDIF

C SURFACE (ARITHMETIC) NODES, REGIONS 1 TO 5
WRITE (MODU, 2002) NR,RGNMMS(NR)
NLG=MATRLS(NR)/100
IF(NR .EQ. 4 .AND. NLG .EQ. 1) THEN
NGT=1

C IF CT .EQ. 'T' THEN

ENDIF

C DIFFUSION NODES, REGIONS 1 TO 5
WRITE (MODU, 2003) NR,RGNMMS(NR)
NLG=MATRLS(NR)/100
IF(NR .EQ. 4 .AND. NLG .EQ. 1) THEN
NGT=1

C IF CT .EQ. 'T' THEN

ENDIF

C SURFACE (ARITHMETIC) NODES, REGIONS 1, 2 OR 3, OUTSIDE SURFACE.
IF(NR .EQ. 4) THEN
WRITE (MODU, 2004) NR,RGNMMS(NR)
NP1=NODBAS+1000

C CALL SETUPA(NR,NR,NP1)

ENDIF

RETURN
2001 FORMAT(7X,'REM SURFACE NODES, INSIDE TANK WALL')
2002 FORMAT(7X,'REM SURFACE NODES, OUTSIDE SURFACE, REGION ',I2,
1,','A25)
2003 FORMAT(7X,'REM DIFFUSION NODES, REGION ',I2,','A25)
2004 FORMAT(7X,'REM THIS MODEL; TANK IS ','F4.0','% FULL, A'.
1,')
I A6,' CASE, ULLAGE AT ',A10)

2005 FORMAT(7X,'REM NO. OF LAYERS THAT ARE ULLAGE IN REGION 5 = ',I3,
            ',13,' OF ',I3,' LAYERS.'/
            ',13,' OF ',I3,' LAYERS.'/
2006 FORMAT(7X,'REM ULLAGE STARTS AT TANK WALL AT THETA POSITION',
            1 ' NO. ',I3/,
            27X,'REM IN REGION 4 - ',I3,' OF ',I3,' LAYERS.'/
            27X,'REM (COUNTING FROM SOUTH POLE)').

END

C441 ISSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SUBROUTINE SETUPA(I,NR,NBASE}
C CHECK FOR HEAT EXCHANGERS, THEN PUT OUT ARITHMETIC NODES
C COMMON /REGION/ NTHETA, NBASEAG, BETA, RIN, T VOL, ROUT (9),
   1 REGNS (9), NLAYRS (9), TEMPS (9), THICK (9),
   2 THK2LAT (9), MATRLS (9), MATNMS (9), RGNMMS (9)
C COMMON /HTXGRS/ NHX, HXTEMP (10), NRHX (10), NNLHX (10),
   1 NTHHX (10), lengthy (10)
C COMMON /UNITS/ MODU, INPEKO, ISCRCH, SINDA
C LOGICAL REGNS, SINDA
C DIMENSION NCHX (10), NNODES (10), NNODEO (10), NODTYP (10), NRHX (10)
C CHARACTER*16 LABEL1, LABEL2, LABEL3
C DATA LABEL1/'SURFACE NODES'/, LABEL2/'HEAT EXCHANGER'/
C NSECT=1
NODENO (1)=1
NNODES (1)=NTHETA
NODTYP (1)=2
TMPTR=TEMPS (1)
IF(NHX .GT. 0) THEN
   THERE ARE HX' S IN THIS GEOMETRY.
   NC=0
C CHECK FOR HX'S IN LAYER 1 OF THIS REGION (NR)
   IF ANY, GET HOW MANY HX'S IN THIS LAYER,
   THEN PUT THE HX NO. INTO ARRAY NCHX
   DO 100 K=1,NHX
      IF(NRHX(K) .EQ. NR) THEN
         IF(NLHX(K) .EQ. 1) THEN
            NC=NC+1
            NCHX (NC) =K
         ENDIF
      ENDIF
   100 CONTINUE
   NPOS =1
   IF(NC .GT. 0) THEN
      THERE ARE NC HX'S IN LAYER 1 OF REGION NR,
      SORT THE HX'S INTO ASCENDING ORDER BY NTHHX
      IF(NC .GE. 2) THEN
         NCMT=NC-1
         DO 152 K=1,NCM1
            K1=NCHX(K-1)
            K2=NCHX(K)
            IF(NTHHX(K2) .LT. NTHHX(K1)) THEN
               NSAV=NCHX(K1)
               NCHX (K1)=NCHX(K2)
               NCHX (K2)=NSAV
            ENDIF
         151 CONTINUE
         152 CONTINUE
      ENDIF
C FIND POSITIONS IN THE LAYER, AND PUT OUT THE ARITHMETIC NODES.
C SURROUNDING THE HX'S.
DO 200 K=1,NC
C HEAT EXCHANGER NO.
NUMHX=NCHX(K)
C GET START THETA AND LENGTH OF THIS HX
NTHO=MTNHX(NUMHX)
LENGTH=MNGTHX(NUMHX)
IF(NTHO .LE. NPOS) THEN
NODENO(NSECT)= NPOS
NNODES(NSECT)= LENGTH
NOHX(NSECT)= NUMHX
NODTYP(NSECT)= 7
NPOS= NPOS+NNODES(NSECT)
ELSE
NNODES(NSECT)= NTHO-1
NPOS= NPOS+NNODES(NSECT)
NSECT=NSECT+1
NODENO(NSECT)= NTHO
NNODES(NSECT)= LENGTH
NOHX(NSECT)= NUMHX
NODTYP(NSECT)= 7
NPOS= NPOS+NNODES(NSECT)
ENDIF
IF(NPOS .LE. NTHETA) THEN
NSECT=NSECT+1
NODENO(NSECT)= NPOS
NNODES(NSECT)= NTHETA-NPOS+1
NODTYP(NSECT)= 2
ELSE
NEW POS > NTHETA; LAST HX ENDED AT NTHETA; STOP AND PRINT
GO TO 250
ENDIF
200 CONTINUE
ENDIF
ENDIF
C 250 DO 300 J=1,NSECT
NNO=NBASE+NODENO(J)
NNOPN=NNO+NNODES(J)-1
IF (NODTYP(J) .EQ. 7) THEN
WRITE(MODU,2001) NOHX(J),NNO,NNOPN
ENDIF
300 CONTINUE
RETURN
END

SUBROUTINE SPHDIF(NR,NODBAS)
CALLED FROM SPHNDS (441)
C COMPUTE DIFFUSION NODES FOR ALL NLAY LAYERS OF A SPHERICAL WEDGE.
C IF A 2D PROBLEM DO ONCE, -- IF 3D PROBLEM DO MORE WEDGES.
C INPUT TO THIS SUBROUTINE
C NTHETA - NO. OF ANGLES, SOUTH POLE TO NORTH POLE ALONG CIRCUMFERENCE
C NNODES - BEGINNING NODE NO.
C RI - RADIUS FROM CENTER OF WEDGE TO OUTSIDE SURFACE OF REGION
C SIGN - MULTIPLIER TO CHANGE RADIUS FROM LAYER TO LAYER, (+1 OR -1)
C THICK - THICKNESS OF REGION
C NLAY - NO. OF LAYERS THRU THE REGION
C TSOX - INITIAL TEMPERATURE OF THE REGION
C MATN - MATERIAL NO. FOR THIS REGION
C MATNAM - NAME OF MATERIAL
C TH - THICKNESS OF THE LAYER
C THETA1, THETA2 = ANGLES FROM HORIZONTAL TO FARTHEST SIDE OF LAYER
C AND TO NEAREST SIDE RESPECTIVELY
THETA1 is \( \gt \) THETA2

\[ R = \text{radius from center of sphere to outside surface of layer.} \]

NARY = NMAT + 1000, SINDA array no. for CP*RHO table

NMAT = 1XX for liquid material number.

NMAT = 2XX for solid material number.

NMAT = 3XX for vapor material number.

For vapor corresponding to the liquid above.

COMMON /REGION/ NTHETA, NBETAS, BETA, RIN, TVOL, ROUT(9),

REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),

THKLAY(9), MATRLS(9), MATNMS(9), RNGNNMS(9)

COMMON /STUFF/ NHIT, PI, CONVR, CONVY, THETA0, DTHERA, DBASE, ROUTSF,

COMMON /ULLAGE/ NLUL4, NLUL5, NTHU4, RINMH, PCTFUL, RADULG, TVULFT,

COMMON /UNITS/ MODU, INPEKO, ISCRCH, SINDA

CHARACTER*1 CT, LG

CHARACTER*1 DASH

CHARACTER*6 LIQVAP

CHARACTER*16 MATNMS

CHARACTER*16 NAME

CHARACTER*25 RGNNMS

LOGICAL REGNS, SINDA

DIMENSION VOLLAY(2,20)

DATA VOLSUM/0.0/, SUMVL1/0.0/

DATA DASH/'-'/

NSELECT=1

NLAY=NLAYRS(NR)

TH=THKLAY(NR)

IL=NR-3

C SET NXX FOR ARRAY NO. = MATERIAL NUMBER

NXX=MATRLS(NR)

NXLG=NXX/100

IF (NLGR .EQ. 0 .OR. NLGR .GT. 3) NLGR=2

PRINT 9898, NR, NXX, NL, CT, MATRLS, MATNMS

C SINGLE NODES IN REGION 4 ARE CONSTANT BOUNDARY NODES

NNODE=MODNAS+1

IF (NLG .EQ. 2) THEN

NAME=MATNMS(NR)

ELSE

NAME=1C(NL)/DASH/MATNMS(NR)

ENDIF

CALL RITNDS (NTHETA, 3, NNODE, 1, 1, TEMPS(NR), 1, NAME)

RETURN
ENDIF
ENDIF
C END IF BLOCKS(COL 11);RS=*,FL,CONVR>0,GEN BDY NODES IN RGN 4, RETURN.
ENDIF
CALL ULLCHK(NR, NL, NLVG, NYL)
IF(NLVG .EQ. 0) THEN
IF(NLGR .EQ. 1 .AND. NYL .EQ. 200) THEN
NY=NYL
NLG=3
ENDIF
ELSE
NTHU=NYL
ENDIF
ENDIF
C END IF BLOCK(COL 7);NR>. 4, GEN BDY NODES IN RGN 4 OR CALLS ULLCHK
NSEC=NSEC+1
IF(NR .LT. 4) ELSE-NLAY-NL+1
IF(NR .GE. 4) ELSE-NL
ROUT(NR)=TH*(EL-.5)
node=NODE+THETA*(NSEC)
ELSE
VOLLAY(IL, NL)=0.
BEGIN
130 LOOP, WRITE OUT THE NODES FOR THE LAYER NL(125)
DO 130 J=1,NTHETA, 2
NNODE=NNODE+1
JJJ=J/2
C SET NARY FOR ARRAY NO.
CALL AREASPA
NNY=NNY+2
NJ=NTHETA-J
IF(NLGE .LT. 0) THEN
C IF(NR .GE. 4) VOLLAY(IL, NL)=0.
C BEGIN 130 LOOP, WRITE OUT THE NODES FOR THE LAYER NL(125 LOOP)
ENDIF
ELSE
NAME=LG(NL)///DASH//MATNMS(NR)
ENDIF
C PRINT ",NLGI",NR, JJJ, NTHU, NLGR, NLGI, LG(NLGI),MATNMS(NR),NAME
CALL RITNDS (443)
CALL RITNDS(NMN1,1,NODE,NJ,NARY,TEMPS(NR),VOL,NAME)

IF(NMN1 .EQ. 2) THEN
  NODE2=NODE+NJ
  CALL RITNDS(NMN1,1,NODE2,NJ,NARY,TEMPS(NR),VOL,NNAME)
ENDIF

IF(NMN1 .EQ. 2) THEN
  NODE2=NODE+NJ
  IF(CT .EQ. 'I') THEN
    NAME='0'
    IF(JJ+J+1 .GE. NTHI .AND. NLGR .EQ. I) THEN
      NAME='I'
      NLGR='3'
    ENDIF
  ENDIF
  NARY=NNM1+NX+NYY
  IF(NLG2 .EQ. 2) THEN
    NAME=MATNMS(NR)
  ELSE
    NAME='LG'//NLG2//DASH//MATNMS(NR)
  ENDIF
  PRINT 'NLG2',NR,JJ,NTHI,NLGR,NLG2, LG(NLG2),MATNMS(NR),NAME
  CALL RITNDS(NMN1,1,NODE2,NJ,NARY,TEMPS(NR),VOL,NNAME)
ENDIF

130 CONTINUE

C IF(NR .GE. 4) THEN
  C
  SUMVLI=SUMVLI+VOLLAY(IL,NL)
  NSUML=NSUML+I
  C
  ENDIF
125 CONTINUE

C IF(NR .GE. 4) THEN
  C
  SMRGNF=SMRGN1/1728.
  VOLSUM=VOLSUM+SMRGN1
  VOLSUM=VOLSUM+SMRGN2
  WDGFT=WDGFT+1728.
  C
  PRINT 9999,NR, TVOL, WDGVFT, WDGFT, VOLQIN, IN3, RADUG, NULL
  C
  PRINT 9997,NR, (VOLLAY(IL,LL),LL-I,NLAY)
  C
  SUMVLF=SUMVLI/1728.
  C
  PRINT 9998,NSUML, SUMVLI,SUMVLF
C ENDIF
RETURN

C FORMATS

2001 FORMAT('THIS MODEL HAS ONLY 1 ',A6,' NODE AT EACH THETA',
1 ' IN REG 4,/
2 ' AND THE HEAT TRANSFER TO THE TANK WALL IS/',
3 ' CONVECTION, ARE THE NODES IN REGION 4 TO BE/',
4 ' I. CONSTANT VALUE BNDY NODES?/
5 ' OR 2. DIFFUSION NODES THAT VARY?/
6 ' TYPE IN 1 OR 2')

2002 FORMAT ('REGION ',I2,' VOL OF EACH LAYER'/
1 (5F15.4))
C9997 FORMAT(' REGION ',I2,' VOL OF EACH LAYER'/
1 TVOL(ET3)-',FI0.4,
2 WEDGE VOL(FT3)-',FI0.4,
3 LIQUID VOL(IN3)-',FI0.4,
4 ULLAGE VOL(IN3)-',FI0.4,' COMP RGN VOL(IN3)-',FI0.4,
5 COMP WDG VOL(FT3)-',FI0.4,' COMP WDG VOL(IN3)-',FI0.4)/
C9998 FORMAT('NSUM=',I4,' VSUMLAYERS=',FI0.4,'; VSUM FT3=',FIO.4)
END
NLVG AND NYY ARE RETURNED TO THE CALLING PROGRAM.

NLAY = NO OF LAYERS IN REGION NR

COMMON /REGION/ /NTM, NLMATS, BETA, RIN, TVOL, ROUT(9),
1 REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
2 THLAY(9), MLMS(9), RLMMS(9), RGNMS(9)

COMMON /ULLAGE/ /NLUL4, NLUL5, NTHU41, RNMHH, PCTFUL, RADULG, TVULFT,
1 CT, LG(3), LIQVAP(3)

COMMON/STUFF/ /NHHT, PI, CONVY, CONVR, THTFIL, DTHETA, DTHFIL,
1 NTH, NLAYRS, NLAYRS, NLAYRS, NLAYRS

LOGICAL REGNS

CHARACTER*1 CT, LG
CHARACTER*16 MLMS
CHARACTER*16 LABLE
CHARACTER*25 RLMMS
CHARACTER*6 LIQVAP

NYY = 0
NLVG = 0
IF(LN .LE. 0) RETURN
NLAY = NLAYRS(NR)

IF (CT .EQ. 'F') RETURN
IF (CT .EQ. '0' .OR. CT .EQ. 'O') THEN
  CT = 0, TANK EMPTY, ALL NODES ARE VAPOR
  NYY = 200
  RETURN
END IF

CT = -C, I.E. ULLAGE IS AT CENTER OF SPHERE
IF (NR .EQ. 4) THEN
  IF (NLAY - LN - NLUL4 .LT. 0) THEN
    NYY = 200
  END IF
ELSE
  IF (NLUL5 .GT. 0) THEN
    IF (NLAY - LN - NLUL5 .LT. 0) THEN
      NYY = 200
    END IF
  END IF
END IF
RETURN

END OF IF BLOCK FOR CT = 'C'

IF (CT .EQ. 'I') THEN
  -I ANALYSIS, ULLAGE ON TOP AND FLAT
  FOR THIS CONDITION NODES IN A LAYER MAY BE DIFFERENT MATERIALS
  THE ANGLE PHI IS THE ANGLE FROM THE VERTICAL HEMISPHERE TO THE
  RADIUS DEFINING NTHU41, THE THETA WHERE THE ULLAGE STARTS IN
  LAYER 1 OF REGION 4.
  PHI IS IN THE COMMON VARIABLE RADULG WHEN CT = 'I' AND NLVG = 1.
  SET NLVG = 1, MATERIALS NOT SAME FOR ENTIRE LAYER
  COMPUTE NTHU FOR THIS LAYER.
  NTHU = NO. OF THETA WHERE MATERIAL CHANGES FROM LIQUID TO VAPOR

NLVG = 1
  IF (NTHU41 .GT. 0) THEN
    IF (NR .EQ. 4 .AND. LN .EQ. 1) THEN
      NTHU = NTHU41
    ELSE
      NTHU = NTHU41
    END IF
  ELSE
    IF (NR .EQ. 4 .AND. LN .EQ. 1) THEN
      PHI = RADULG
      IF (RNMHH .EQ. 0.) THEN
        NTHU = NTHU41
      ELSE
        NTHU = NTHU41
      END IF
    END IF
  END IF
RETURN

END OF IF BLOCK FOR CT = 'I'

211
IF(RINMHH .GT. 0) THEN
  YULL = RINMHH
  ISIGN = 1
  NDO = NTHETA
  INC = 1
ELSE
  YULL = RINMHH
  ISIGN = -1
  NDO = 1
  INC = -1
ENDIF

ENDIF

IF(NR .EQ. 4) R = RIN - (LN-I) * THKLAY(4)
IF(NR .EQ. 5) R = ROUT(5) - (LN-I) * THKLAY(5)
IF(R .LT. YULL) THEN
  NTHU = 0
ELSE
  UANG = PI/2. - PHI + DTHETA/2.
  DO I = NTHU41, NDO, INC
    NCOUNT = NTHU41
    Y = R * SIN(UANG)
    IF(Y .GE. YULL) THEN
      NTHU = NTHU41 + NCOUNT
      GO TO 101
    ELSE
      UANG = UANG + DTHETA
    ENDIF
  ENDDO
  NTHU = NTHU + INC
ENDIF

GO TO 101

C END OF IF BLOCK FOR CT = 'I'

C END OF IF BLOCK FOR CT = 'T'

RETURN

ENDI?

END OF IF BLOCK FOR CT - 'I'

IF(CT .EQ. 'T') THEN
  CT = T;
  ULLAGE IS AT THE TOP OF THE SPHERE
ENDIF

C THIS OPTION IS NOT YET AVAILABLE.

C ENDIF

C END OF IF BLOCK FOR CT = 'T'

RETURN

END

C SUBROUTINE SPHCDS

CALLED FROM CONDRS (46)

C SUBROUTINE TO GENERATE CONDUCTOR DATA FOR SPHERE WEDGE.

C COMMON /REGION/ NTHETA, NBETAS, BETA, RIN, TVOL, ROUT(9),
  \ 1  REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
  \ 2  THKLAY(9), MATMCS(9), MATNMS(9), RGNNMS(9)
C COMMON /HTXGRS/ NHX, HXTEMP(10), NRHX(10), NIXH(10),
  \ 1  NTHHX(10), LNGTHX(10)
C COMMON/STUFF/ NHTT, PI, CONVY, CONVR, THETA0, DTHETA, NBASOS, ROUTSF,
  \ 1  BNCOEF(2)
C COMMON /ULLAGE/ NUL4, NUL5, NTHU41, RINMHH, PCTFUL, RADIUS, TVULFT,
  \ 1  CT, LG(3), LIQVAP(3)
C COMMON /UNITS/ MODU, INPEKO, ISCRCH, SINDA

CRY08400
DIMENSION NHXADL(IO)
LOGICAL REGNS, SINDA
CHARACTER*1 CT, LG
CHARACTER*6 LIQVAP
CHARACTER*10 ATOSL(2).
CHARACTER*6 MATNMS
CHARACTER*f6
CHARACTER*25 RGNNMS

DATA ATOSL / 'CONVECTION', 'RADIATION'/

IF NHTT = 1 CONDUCTION ONLY IN REGIONS 4 AND 5
IF NHTT = 2 CONVECTION ONLY IN REGIONS 4 AND 5
IF NHTT = 3 CONDUCTION AND CONVECTION IN REGIONS 4 AND 5

NG=0
NS=1

GENERATE RADIAL CONDUCTORS, (CONDUCTION)
DO REGIONS IN ORDER, 1, 2, 3 FROM RIN TO OUT
4, 5 FROM RIN INWARD

DO 100 NR=1,5
RADIAL AND CIRCUMFERENTIAL CONDUCTORS, (CONDUCTION), ALL REGIONS
IF(REGNS(NR)) THEN
IF(NR .GE. 4 .AND. NHTT .EQ. 2) GO TO 110
WRITE(MODU, 2001)
RADCON(4611)
WRITE(MODU, 2002)
CALL CIRCON(4612)
CONTINUE
ENDIF
110 CONTINUE
END OF IF BLOCKS FOR CONDUCTION CONDUCTORS IN ALL REGIONS.

NOW GENERATE CONVECTION CONDUCTORS IN REGIONS 4,5 IF NHTT >= 2.
IF(NHTT .GE. 2) THEN
IF(CONVR .NE. O.O) THEN
RADIAL CONDUCTORS, (CONVECTION)
WRITE(MODU, 2003)
DO 330 NR=4,5
IF(REGNS(NR)) THEN
RADCON(4611)
CONTINUE
ENDIF
330 CONTINUE
ENDIF
C
IF(CONVR .NE. O.O) THEN
CIRCUMFERENTIAL CONDUCTORS, (CONVECTION)
WRITE(MODU, 2004)
DO 340 NR=4,5
IF(REGNS(NR)) THEN
CIRCON(4612)
CONTINUE
ENDIF
340 CONTINUE
ENDIF
C
END OF IF BLOCKS FOR CONVECTION CONDUCTORS IN REGIONS 4,5.
C
C CONDUCTORS FROM OUTSIDE 'ATMOSPHERE' NODE (20301) TO SPHERE
C OUTER SURFACE, IF ANY. IF(TEMPS(B) .NE. -9999.9) THEN THERE
C IS A NODE OUTSIDE, GENERATE THE CONDUCTORS.
C4611SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
LOGICAL REGNS, SINDA

DO REGIONS IN ORDER, 1, 2, 3 FROM RIN TO OUT

4, 5 FROM RIN INWARD

SIGN=-1.0
IF (NR .GE. 4) SIGN=-1.0
NLAY=NLAYS(NR)
NLPI=NLAY+1
TH=THKLAY(NR)
EL=TH/2.

C GENERATE RADIAL CONDUCTORS.

DO REGIONS IN ORDER, 1, 2, 3 FROM RIN TO OUT

4, 5 FROM RIN INWARD

NLAY=NLAYRS(NR)
NLPI=NLAY+1
TH=THKLAY(NR)

C RAD = INSIDE RADIUS OF REGION NR.

RAD=RIN
IF (NR .EQ. 2) RAD=RIN+THICK(1)
IF (NR .EQ. 3) RAD=RIN+THICK(1)+THICK(2)
IF (NR .EQ. 5) RAD=RIN+THICK(4)

C SET ARRAY NOS. FOR K, MATERIAL NO. + 6000. PROPERTY WMAT.

NARAY=MATRLS(NR)+6000
NLGR=MATRLS(NR)/100
DO 110 I=1,NLPI
IF (NR .EQ. 5 .AND. L .EQ. NLPI) GO TO 110

C CONDUCTORS LAYER 1 TO REGION BOUNDARY; NA=BOUNDARY, NB=INTERIOR.

C TEST FOR HEAT EXCHANGERS ON TOP OF LAYER CORRESPONDING TO

C NODES NA, (NODBAS-1000). IF THERE ARE ANY GET NO. OF HX'S ON THIS

C LAYER, AND WHICH HX'S THEY ARE.

C GET REGION NO. WHICH CONTAINS THE HX ADJACENT TO THIS LAYER (1).

C NR=1, ADJR=4; NR=2, ADJR=1; NR=3, ADJR=2; NR=4, ADJR=4; NR=5, ADJR=5.

C NR, L, NR, NEXT

C GET REGION NO. WHICH CONTAINS THE HX ADJACENT TO THIS LAYER (2).

C NR=1, ADJR=4; NR=2, ADJR=1; NR=3, ADJR=2; NR=4, ADJR=4; NR=5, ADJR=5.

C NR, L, NR, NEXT

C GET REGION NO. WHICH CONTAINS THE HX ADJACENT TO THIS LAYER (3).

C NR=1, ADJR=4; NR=2, ADJR=1; NR=3, ADJR=2; NR=4, ADJR=4; NR=5, ADJR=5.
RADA - RAD+S I G*N (THICK(NR) - TH/4.)
RADB = RADA
RADAB = RADA
ENDIF
LTOUCA = NLAY
LTOUCH = NLAY
NEXT = NADJR
IF (NR .LT. 4) NEXT = NADJR + 1
WRITE (MODU, 2002) NR, NLAY, NR, NEXT
ELSE
C CONDUCTORS LAYER TO LAYER INSIDE OF REGION.
NCT = 2
NTP = 7
NADJR = NR
IF (NR .LE. 3) THEN
NLTEST = NLAY - L + 2
ELSE
NLTEST = L
ENDIF
LM1 = L - 1
ELM1 = L - 1
ELM2 = ELM1 - 1.
NA = NA + NTHERA
NB = NA + NTHERA
RADAB = RAD + SIGN*TH*ELMI
RADA = RAD + SIGN*(TH*ELMI - TH/4.)
RADAB = RAD + SIGN*(TH*ELMI - TH/4.)
LTOUCA = L - 1
LTUCH = L
WRITE (MODU, 2003) NR, LM1, L
ENDIF
ENDIF
C TEST FOR HX(S) ON THE BOUNDARY OF THE LAYER TO LAYER CONDS.
C OR FOR HX(S) ON THE LAYER TO THE REGION BOUNDARY CONDS.
C OR FOR HX(S) ON THE BOUNDARY OF THE LAYER TO LAYER CONDS.
C OR FOR HX(S) ON THE LAYER TO THE REGION BOUNDARY CONDS.
CHX = 0
DO 120 K = 1, NHX
IF (NRHX(K) .EQ. NADJR) THEN
C TEST FOR HX IN APPROPRIATE LAYER OF ADJACENT REGION.
IF (NLHX(K) .EQ. NLTEST) THEN
HXS = HXS + 1
NHXADL(HXS) = K
ENDIF
ENDIF
120 CONTINUE
C PRINT *, NR, L, NADJR, NLTEST, NCT, NHX, HXS, (NHXADL(K), K=1, HXS),
1 NLVGA, NLVGB, NYYA, NYYB
ENDIF
C GENERATE THE CONDUCTORS FOR THE CURRENT VALUE OF L (LOOP 110)
L1 = 1, LAYER 1 TO BOUNDARY
L2 = 2 TO NLAY, LAYER (L-1) TO LAYER L
L = NLAY + 1, LAYER L TO BOUNDARY
DO 130 J=1,NTHETA,2
JJJ=J/2
NA=NA+1
NB=NB+1
NMANY=2
MJ=MTHETA-J
IF (NJ .LE. 0) THEN
  NMANY=1
  NL=0
ENDIF
IA=NJ
IB=IA
IG=I
MA=NA
MB=NB
NC=NMANY
NT=NTP
NJ=0
IF (NC .EQ. 1) NT=NTP-1
C
IF (NC .EQ. 1) THEN
C 1 CONDUCTOR TO BE PUT OUT, USE CAL
NT=2
ENDIF
ENDIF
C
TEST IF THERE ANY HEAT EXCHANGERS ON THIS LAYER.
IF (NHXS .GT. 0) THEN
C
C THERE ARE HEAT EXCHANGERS ON THIS BOUNDARY
C SET CONTROLS TO PUT OUT 1 CONDUCTOR AT A TIME FOR THETA1 AND
C THETA2(MIRROR THETA)
NC=1
IF (NC .EQ. 1) THEN
  NT=4
ELSE
  NT=2
ENDIF
C
TEST FOR THIS THETA IN A RANGE OF AN HX ON THIS BOUNDARY
NTK=1
NTEST=(J+1)/2
145 CONTINUE
DO 140 KK=1,NHXS
  NKK=NHXLADL(KK)
  NLO=NTHXH(NKK)
  NH1=NLO+LAGTHX(NKK)-1
  IF (NTEST .GE. NLO .AND. NTEST .LE. NH1) THEN
    IF (NCT .EQ. 3) THEN
      MB=20000+NKK
    ELSE
      MA=20000+NKK
      FA=FB
    ENDIF
  ENDIF
C
C THIS THETA IS WITHIN THE RANGE OF THE HX
ENDIF
C
END OF IF BLOCK TO TEST FOR THETA WITHIN THE RANGE OF THE HX
140 CONTINUE
IF(NCC .EQ. 1) THEN
CALL SETARY(NR, JJJ, NNJ, NAA, NAB)
CALL RITCND(NT, NC, IG, MA, IA, MB, IB, XAA, XAB, FA, FB, LABLE)
ELSE
CALL RITCND(NT, NG, NC, IG, MA, IA, MB, IB, XAA, XAB, FA, FB, LABLE)
ENDIF
NG=NG+NC
C
IF(NC .EQ. 1) THEN
IF(NCT .EQ. 1) THEN
MB=MA
MA=NA
ENDIF
CALL SETARY(NR, JJJ, NNJ, NAA, NAB)
CALL RITCND(NT, NG, NC, IG, MA, IA, MB, IB, XAA, XAB, FA, FB, LABLE)
ENDIF
END OF 1ST HALF OF IF BLOCK ON NHX > 0.
ELSE
C NO HEAT EXCHANGERS ON THIS BOUNDARY
IF(NCC .EQ. 1) THEN
CALL RITCND(NT, NG, NC, IG, MA, IA, MB, IB, XAA, XAB, FA, FB, LABLE)
ENDIF
C END OF IF BLOCK (COL 9) IS NCC = 1?
ENDIF
IF0 CONTINUE
RETURN
FORMAT STATEMENTS
218
C4611 FORMAT (7X,'REM RADIAL CONDUCTORS REGION',I2),
1   ' ', LAYER ','12,' TO BOUNDARY ','12, ',12)
C4612 FORMAT (7X,'REM RADIAL CONDUCTORS REGION',I2),
1   ' ', LAYER ','12,' TO LAYER ','12)
END
SUBROUTINE SETARY(NR, JJJ, NNJ, NAA, NAB)
C THIS SUBROUTINE CHECKS FOR VAPOR NODES IN CONDUCTOR DATA.
C AND SETS NYA AND NYB = 0 OR 200 TO CHANGE THE PROPERTY TABLES
C FOR A NODE. THIS IS ONLY DONE WHEN NR >= 4 AND NLGR=1.
C
COMMON /REGION/ NTHETA, NBETAS, BETA, RIN, TVOL, ROUT (9),
1   REGN (9), NLAYRS (9), THICK (9),
2   THKLAY (9), MATLDS (9), MATNMS (9), RGNMNS (9)
COMMON /ULLAGE/ NLUL4, NLUL5, NTHU4, RINMHH, PCTFUL, RADULG, TVULFT,
1   CT, LG (3), LIQVAP (3)
COMMON /CONDUL/ MA, MB, NC, NARA¥, NLVGA, NLVGB, NYYA, NYYB
CHARACTER*1 CT, LG
CHARACTER*16 LABEL
LOGICAL REGNS
CHARACTER*16 MATNMS
CHARACTER*25 RGNNMS
NAA-NARAY
NAB=NARAY
IF(NR .LE. 3) RETURN
IF(NLVGA+NLVGB .EQ. 0) THEN
NLVG = 0, ALL NODES IN BOTH LAYERS SAME MATERIAL.
   NAA=NARAY+NYYA
   NAB=NARAY+NYYB
   NC=2
   LABEL=LG(NLG)//MATNMS(NR)
ELSE
NLVG > 0, SOME NODES IN ONE OF THE LAYERS MAY BE DIFFERENT MATERIALS
   IF(CT .EQ. 'I') THEN
C CT = 'I', 1-G CASE AND ULLAGE ON TOP AND FLAT
      NC=1
      NYYA=0
      NYYB=0
      NTHUA=NYYA
      NTHUB=NYYB
      IF(NTHUA .GT. 0 .AND. MA .LT. 20000) THEN
         IF(JJJ+NNJ+I .GE. NTHUA) THEN
            NYA=200
            NLG=3
         ENDIF
      ENDIF
      IF(NTHUB .GT. 0 .AND. MB .LT. 20000) THEN
         IF(JJJ+NNJ+I .GE. NTHUB) THEN
            NYB=200
            NLG=3
         ENDIF
      ENDIF
   ENDIF
   NAA=NARAY+NYYA
   NAB=NARAY+NYYB
   LABEL=LG(NLG)//MATNMS(NR)
ENDIF
SUBROUTINE CIRCON(NR, NCC, NG)
C THIS SUBROUTINE CHECKS FOR VAPOR NODES IN CONDUCTOR DATA.
C AND SETS NYA AND NYB = 0 OR 200 TO CHANGE THE PROPERTY TABLES
C FOR A NODE. THIS IS ONLY DONE WHEN NR >= 4 AND NLGR=1.
C
COMMON /REGION/ NTHETA, NBETAS, BETA, RIN, TVOL, ROUT (9),
1   REGN (9), NLAYRS (9), THICK (9),
2   THKLAY (9), MATLDS (9), MATNMS (9), RGNMNS (9)
COMMON /ULLAGE/ NLUL4, NLUL5, NTHU4, RINMHH, PCTFUL, RADULG, TVULFT,
1   CT, LG (3), LIQVAP (3)
COMMON /CONDUL/ MA, MB, NC, NARA¥, NLVGA, NLVGB, NYYA, NYYB
CHARACTER*1 CT, LG
CHARACTER*16 LABEL
LOGICAL REGNS
CHARACTER*25 MATNMS
CHARACTER*25 RGNNMS
NAA-NARAY
NAB-NARAY
IF(NR .LE. 3) RETURN
IF(NLVGA+NLVGB .EQ. 0) THEN
NLVG = 0, ALL NODES IN BOTH LAYERS SAME MATERIAL.
   NAA=NARAY+NYYA
   NAB=NARAY+NYYB
   NC=2
   LABEL=LG(NLG)//MATNMS(NR)
ELSE
NLVG > 0, SOME NODES IN ONE OF THE LAYERS MAY BE DIFFERENT MATERIALS
   IF(CT .EQ. 'I') THEN
C CT = 'I', 1-G CASE AND ULLAGE ON TOP AND FLAT
      NC=1
      NYYA=0
      NYYB=0
      NTHUA=NYYA
      NTHUB=NYYB
      IF(NTHUA .GT. 0 .AND. MA .LT. 20000) THEN
         IF(JJJ+NNJ+I .GE. NTHUA) THEN
            NYA=200
            NLG=3
         ENDIF
      ENDIF
      IF(NTHUB .GT. 0 .AND. MB .LT. 20000) THEN
         IF(JJJ+NNJ+I .GE. NTHUB) THEN
            NYB=200
            NLG=3
         ENDIF
      ENDIF
   ENDIF
   NAA=NARAY+NYYA
   NAB=NARAY+NYYB
   LABEL=LG(NLG)//MATNMS(NR)
ENDIF
CALL IF NCC -1, COMPUTE CONDUCTION CONDUCTORS IN REGIONS 4 AND 5.
CALL IF NCC = 2, COMPUTE CONVECTION CONDUCTORS IN REGIONS 4 AND 5.

NARY = 6000 + NXX, SINDA ARRAY NO. FOR K, (THERMAL CONDUCTIVITY).
WHERE NXX = MATERIAL NO. AS INPUT BY USER.
NXX = 1XX, LIQUID MATERIAL NO.
NXX = 2XX, SOLID MATERIAL NO.
NXX = 3XX, VAPOR MATERIAL NO.

COMMON /REGION/ NTHETA, NBETAS, BETA, RIN, TVOL, ROUT(9),
1 REGNS(9), NLYRS(9), TEPMS(9), THICK(9),
2 THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
COMMON /HTXGRS/ NXX, HXTEMP(10), NHBX(10), NULX(10),
1 NTHBX(10), LNTHBX(10)
COMMON/STUFF/ NHIT, PI, CONVY, CONVR, THETA0, DTHETA, NBASOS, ROUTSF,
1 BNCOEF(2)
COMMON /VILAGE/ NLUL4, NLUL5, WTHU41, RINHHH, PCTFUL, RADULG, TVULFT,
1 CT, LG(3), LIQVAP(3)
COMMON /UNITS/ MODU, INPEKO, ISCRCH, SINDA

DIMENSION NHXADL(10)
LOGICAL REGNS, SINDA

CHARACTER*1 CT, LG
CHARACTER*6 LIQVAP
CHARACTER*16 MATNMS
CHARACTER*25 RGNNMS

GENERATE CONDUCTORS ALONG CIRCUMFERENCE (Y DIRECTION)
NTMI=NTHETA-1
SIGN=1.0
IF (NR .GE. 4) SIGN=-1.0
NLAY=NLYRS(NR)
TH=THKLAY(NR)
RAD=RIN
IF (NR .EQ. 2) RAD=RIN+THICK(1)
IF (NR .EQ. 3) RAD=RIN+THICK(1)+THICK(2)
IF (NR .EQ. 5) RAD=RIN-THICK(4)
RAD=RAD-SIGN*TH/2.
NDBAS=2000+NR
NARAY=MATRLS(NR)+6000
NLGR=MATNMS(NR)/100
DO 325 L=1,NLAY
NYT=0
NLVG=0
NLGR=NLGR
IF (NR .GE. 4) THEN
IF (NLGR .EQ. 1) THEN

CALL ULLCHK (NR, L, NLVG, NYT)

ENDIF
WHITE(MODU, 2005) NR, L
RAD=RAD+SIGN*TH
NL =RAD+DTHETA/2.
NA=NTHETA*(L-1)+NDBAS
DO 330 J=1,NTMI, 2
JJJ=J/2
NA=NA+1
NB=NA+1
NMANY=2
MA=NA
MB=MB+1
NC=MANY
NJ=NTHETA-J-1
NTF =-7

CALL AREASP (2, JJJ, RAD, TH, AREA)
XAA=AREA
FA = AREA/EL
IF (NJ .EQ. 0) THEN
NC=1
NTP = 6
ELSE
CALL AREAERP(2, JJJ+1, RAD, TH, AREA)
FB = AREA/EL
ENDIF
IA=NJ
IB=IA
IG=1
IF (NCC .EQ. 1) THEN
ENDIF
IA=NJ
IB=IA
IF (NLVG .EQ. 0) THEN
CALL AREASP(2, JJJ+1, PAD, TH, AREA)
FB = AREA/EL
ENDIF
IA=NJ
IB=IA
IG=1
IF (NCC .EQ. 1) THEN
IF (NLVG .EQ. 0) THEN
CALL AREASP(2, JJJ+1, PAD, TH, AREA)
FB = AREA/EL
ENDIF
IA=NJ
IB=IA
IG=1
IF (NCC .EQ. 1) THEN
C NLVG > 0, SAME MATERIAL FOR THIS LAYER.
NAA=NARAY+NYT
NAB=NAAA
CALL RITCND (4613)
CALL RITCND (NTP, NG, NC, IG, MA, IA, IB, NAA, NAB, FA, FB, MATNMS (NR))
NG=NG+NC
ELSE
C NLVG > 0, MATERIAL MAY BE DIFFERENT FOR SOME NODES IN THIS LAYER
NTHU-NYT
NTP=6
IG0=0
NAA=NARAY
NAB=NAAA
IF (CT .EQ. '1') THEN
IF (JJJ .GE. NTHU) THEN
NAA=NARAY+200
NC=1
ENDIF
IF (JJJ+1 .GE. NTHU) THEN
NAB=NARAY+200
NC=1
NLC=3
ENDIF
IF(MANY .EQ. 2 .AND. NC .EQ. 1) THEN
MA=MA+NJ
MB=MB+NJ
IF (CT .EQ. '1') THEN
IF (JJJ+1 .GE. NTHU) THEN
NAA=NARAY+200
IF (JJJ+1 .GE. NTHU) THEN
NAB=NARAY+200
ENDIF
ENDIF
C LABEL=LG (NLG)//MATNMS (NR)
CALL RITCND (NTP, NG, NC, IG, MA, IA, MB, NAA, NAB, FA, FB, MATNMS (NR))
NG=NG+NC
ENDIF
ENDIF
ELSE
XAB=CONVY
NTP=NTP-4
CALL RITCND (NTP, NG, NC, IG, MA, IA, MB, XAA, XAB, 1, 1, MATNMS (NR))
NG=NG+NC
ENDIF
330 CONTINUE
325 CONTINUE
RETURN
C FORMAT STATEMENTS
2005 FORMAT (7X,'REM CIRCUMFERENTIAL CONDUCTORS REGION', I2, 1
       ',LAYER NUMBER ', I2)
       END

CRY14700
CRY14710
CRY14720
CRY14730
CRY14740
CRY14750
CRY14760
CRY14770
CRY14780
CRY14790
CRY14800
CRY14810
CRY14820
CRY14830
CRY14840
CRY14850
CRY14860
CRY14870
CRY14880
CRY14890
CRY14900
CRY14910
CRY14920
CRY14930
CRY14940
CRY14950
CRY14960
CRY14970
CRY14980
CRY14990
CRY15000
CRY15010
CRY15020
CRY15030
CRY15040
CRY15050
CRY15060
CRY15070
CRY15080
CRY15090
CRY15100
CRY15110
CRY15120
CRY15130
CRY15140
CRY15150
CRY15160
CRY15170
CRY15180
CRY15190
CRY15200
CRY15210
CRY15220
CRY15230
CRY15240
CRY15250
CRY15260
CRY15270
CRY15280
CRY15290
CRY15300
CRY15310
CRY15320
CRY15330
CRY15340

221
APPENDIX E

CryoTran Program Listings

Part III CRYOCYL FORTRAN
SUBROUTINE CYLINDER(NAN)

COMMON/REGION/NTHETA, NBETAS, BETA, RIN, TVOL,
* ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
* THKLAY(9), MATLS(9), MATNMS(9), RGNMS(9),
COMMON/TOPBOT/NBOT, NFTLAY, NSTLAY, NFSLAY, NSBLAY,
* NEBLAY, ETRAT, EBRAT, FTTHK, FRTHK
COMMON/VOLUME/ NUL4, NUL5, NTHU41, RINMHH, PCTFUL, RADULG, TVULFT,
* CT, LG(3), LIQVAP(3)
COMMON/CYDATA/CYLHT, NCYLAY
COMMON/NXGRS/NHX, HXTEMP(i0), NRHX(i0), NLHX(i0), NTHHX(i0), LNGTHX(i0)
COMMON/UNITS/MODU, SINDA

LOGICAL REGNS
CHARACTER*16 RGNMS, MNAME, MATNMS
CHARACTER*1 CT, LG, YORN
CHARACTER j 6 LIQVAP

CALL CLEARS
WRITE (6,44)
WRITE (6,*) 'NOW ENTER YOUR CHOICE OF HOW THE CYLINDRICAL'
WRITE (6,*) 'SECTION OF THE TANK IS GOING TO BE DEFINED:'
WRITE (6,*) '1. HEIGHT ; WALL THICKNESS ; INSIDE RADIUS (INCH).'
WRITE (6,*) '2. HEIGHT ; OUTSIDE RADIUS ; INSIDE RADIUS (INCH).'
WRITE (6,*) '3. HEIGHT ; OUTSIDE RADIUS ; WALL THICKNESS (INCH).'
CALL READIN(NMENU, I,3)
IF (NMENU.EQ.I) THEN
  CALL CLEARS
  WRITE (6,44)
  WRITE (6,*) 'ENTER CYLINDRICAL HEIGHT (INCHES).'
  CALL READRE(CYLHGT)
  CALL CLEARS
  WRITE (6,44)
  WRITE (6,*) 'ENTER THE THICKNESS OF THE WALL (INCHES).'
  CALL READRE(THICK(1))
  CALL CLEARS
  WRITE (6,44)
  WRITE (6,*) 'ENTER THE INSIDE RADIUS OF THE TANK (INCHES).'
  CALL READRE(RINI)
  CALL CLEAR
  ROUT(1)=RIN+THICK(1)
ENDIF
IF (NMENU.EQ.2) THEN
  CALL CLEARS
  WRITE (6,44)
  WRITE (6,*) 'ENTER CYLINDRICAL HEIGHT (INCHES).'
  CALL READRE(CYLHGT)
  CALL CLEARS
  WRITE (6,44)
  WRITE (6,*) 'ENTER THE OUTSIDE RADIUS OF THE TANK (INCHES).'
  CALL READRE(RADIUS)
  CALL CLEARS
  WRITE (6,44)
  WRITE (6,*) 'ENTER THE INSIDE RADIUS OF THE TANK (INCHES).'
  CALL READRE(RIN)
  CALL CLEAR
  THICK(1)=RADIUS-RIN
  ROUT(1)=RIN+THICK(1)
ENDIF
IF (NMENU.EQ.3) THEN
  CALL CLEARS
  WRITE (6,44)
  WRITE (6,*) 'ENTER CYLINDRICAL HEIGHT (INCHES).'
  CALL READRE(CYLHGT)
  CALL CLEARS
  WRITE (6,44)
  WRITE (6,*) 'ENTER THE OUTSIDE RADIUS OF THE TANK (INCHES).'
  CALL READRE(RADIUS)
  CALL CLEARS
  WRITE (6,44)
  WRITE (6,*) 'ENTER THE WALL THICKNESS (INCHES).'
  CALL READRE(THICK)
  CALL CLEARS
  WRITE (6,44)
  WRITE (6,*) 'ENTER THE INSIDE RADIUS OF THE TANK (INCHES).'
  CALL READRE(RINI)
  CALL CLEAR
  ROUT(1)=RIN+THICK(1)
ENDIF
CALL READRE (THICK(1))
CALL CLEAR
RIN=RADIUS-THICK(1)
ROUT(1)=RIN+THICK(1)
ENDIF
CALL CLEAR
WRITE (6,44)
WRITE (6,*) 'ENTER NUMBER OF LAYERS TO DIVIDE THE HEIGHT INTO'
CALL READIN (NCTLAY,0,9999)
CALL CLEAR
WRITE (6,44)
WRITE (6,*) 'ENTER THE NUMBER CORRESPONDING TO THE TOP SHAPE: ' 
WRITE (6,*) '(1=NO TOP, 2=FLAT TOP, 3=SPHERICAL TOP, 4=ELIPTICAL TOPICAL BOTTOM)'
CALL READIN (NTOP,1,4)
CALL CLEAR
WRITE (6,44)
WRITE (6,*) 'ENTER THE NUMBER CORRESPONDING TO THE BOTTOM SHAPE ' 
WRITE (6,*) '(1=NO BOTTOM, 2=FLAT BOTTOM, 3=SPHERICAL BOTTOM, 4-ELIPTICAL BOTTOM)'
CALL READIN (NBOT,1,4)
CALL CLEAR
IF (NTOP.EQ.1) GOTO 7
WRITE (6,44)
WRITE (6,*) 'ENTER THE LAYERS TO DIVIDE THE TOP INTO'
IF (NTOP.EQ.2) THEN
CALL READIN (NFTLAY,0,9999)
CALL CLEAR
WRITE (6,44)
WRITE (6,*) 'ENTER THE THICKNESS OF THE FLAT TOP (INCHES).'
CALL READRE (FTTHK)
ENDIF
IF (NTOP.EQ.3) CALL READIN (NSTLAY,0,9999)
IF (NTOP.EQ.4) THEN
CALL READIN (NETLAY,0,9999)
CALL CLEAR
WRITE (6,44)
WRITE (6,*) 'DO YOU WANT A SQRT(2.) ELIPSE I.E. A:B=SQRT(2.)?' 
CALL READAL (YORN)
IF (YORN.EQ.'Y') ETAR=1./SQRT(2.)
IF (YORN.EQ.'N') THEN
WRITE (6,*) 'ENTER THE RATIO OF A (MAJOR AXIS) TO B ' 
WRITE (6,*) ' (MINOR AXIS) I.E. A/B FOR THE TOP'
CALL READRE (ETRAT)
ETAR=1./ETAR
CALL CLEAR
ENDIF
ENDIF
IF (NBOT.EQ.1) GOTO 99
CALL CLEAR
WRITE (6,44)
WRITE (6,*) 'ENTER THE LAYERS TO DIVIDE THE BOTTOM INTO'
IF (NBOT.EQ.2) THEN
CALL READIN (NFBLAY,0,9999)
CALL CLEAR
WRITE (6,44)
WRITE (6,*) 'ENTER THICKNESS FOR FLAT BOTTOM SHAPE (INCHES).'
CALL READRE (FBTHK)
ENDIF
IF (NBOT.EQ.3) CALL READIN (NSBLAY,0,9999)
IF (NBOT.EQ.4) THEN
CALL READIN (NEBLAY,0,9999)
CALL CLEAR
WRITE (6,44)
WRITE (6,*) 'DO YOU WANT A SQRT(2.) ELIPSE I.E. A:B-SQRT(2.)?' 
CALL READAL (YORN)
IF (YORN.EQ.'Y') EBRAT=1./SQRT(2.)
IF (YORN.EQ.'N') THEN
WRITE (6,*) 'ENTER THE RATIO OF A (MAJOR AXIS) TO B ' 
WRITE (6,*) ' (MINOR AXIS) I.E. A/B FOR THE BOTTOM'
CALL READRE (EBLAT)
EBLAT=1./ETRAT
CALL CLEAR
ENDIF
ENDIF
ENDIF
CALL READRE (EBRAT)
EBRAT=1/EBRAT
CALL CLEARS
ENDIF
ENDIF
99 NTHTA=NCYLA+NETLAY+NETLAY+NLAY+NFBLAY+NEBLAY+NSBLAY
44 FORMAT (///)
RETURN
END

SUBROUTINE MATMNU (IREG)
COMMON /REGION/ NTHTA, NBETAS, BETA, RIN, TVOL,
 ROUT (9), REGNS (9), NLAYRS (9), THICK (9),
 THKLAY (9), MATRLS (9), MATNMS (9), RGNNMS (9)
COMMON /UNITS/ MODU, SINDA
DIMENSION P(100)
LOGICAL REGNS
CHARACTER*16 MNAME, MATNMS, RGNNMS
CALL CLEARS
12 REWIND 4
WRITE (6,1)
30 READ (4,2) MNUM, MNAME
IF (MNUM.EQ.0) GOTO 10
IF (MNUM.GE.100.AND.MNUM.LT.200) WRITE (6,133) MNUM, MNAME
IF (MNUM.GE.200.AND.MNUM.LT.300) WRITE (6,3) MNUM, MNAME
IF (MNUM.GE.300) WRITE (6,134) MNUM, MNAME
READ (4,*) SPRES, EPRES, NINC
GOTO 30
10 MNUM=999
MNAME='USER DEFINED'
WRITE (6,3) MNUM, MNAME
CALL READIN (MATRLS(IREG), I00, 1000)
NTEST=0
REWIND 4
20 READ (4,2) MNUM, MNAME
IF (MNUM.EQ.0) GOTO 11
READ (4,*) SPRES, EPRES, NINC
IF (MATRLS(IREG).EQ.MNUM.OR.MATRLS(IREG).EQ.999) NTEST=1
IF (MATRLS(IREG).EQ.MNUM) MATNMS(IREG)=MNAME
GOTO 20
11 IF (NTEST.EQ.0) THEN
CALL CLEARS
WRITE (6,‘‘) ‘‘MATERIAL NUMBER DOES NOT EXIST’‘
WRITE (6,‘‘) ‘‘PLEASE ENTER A MATERIAL NUMBER FROM THE’‘
WRITE (6,‘‘) ‘‘MENU BELOW OR ENTER 999 TO ENTER’‘
WRITE (6,‘‘) ‘‘OWN PROPERTY DATA.’‘
GOTO 12
ENDIF
IF (MATRLS(IREG).EQ.999) THEN
CALL CLEARS
WRITE (6,7) IREG
CALL READLC (MATNMS(IREG))
ENDIF
REWIND 4
1 FORMAT (///, ‘ENTER MATERIAL NUMBER FOR REGION ‘, I1, ‘:)’
2 FORMAT (13, A16)
3 FORMAT (1X, 13, 5X, A16)
133 FORMAT (1X, 13, 5X, ‘LIQUID ‘, A16)
134 FORMAT (1X, 13, 5X, ‘GAS ‘, A16)
7 FORMAT (///, ‘ENTER MATERIAL NAME FOR REGION ‘, I1, ‘:)’
RETURN
END

SUBROUTINE PRPTBL(IREG)
COMMON /REGION/ NTHTA, NBETAS, BETA, RIN, TVOL,
 ROUT (9), REGNS (9), NLAYRS (9), THICK (9),
 THKLAY (9), MATRLS (9), MATNMS (9), RGNNMS (9)
COMMON /UNITS/ MODU, SINDA
LOGICAL REGNS
CHARACTER*16 MNAME, MATNMS, RGNNMS

225
CHARACTER*15 TABUNT(10)
CHARACTER*1 T8
CHARACTER*20 PRPUNT(10)
CHARACTER*13 PROP
DIMENSION P(100), T(10, 700), NAMTAB(10), PRTOUT(1000), CONFAC(10)
NRPT=0
DO 581 I=1,IREG
   IF (I.NE.IREG) THEN
      IF (MATRLS(I).EQ.MATRLS(IREG)) NRPT=1
   ENDIF
581 CONTINUE
IF (NRPT.EQ.1) GOTO 100
IF (MATRLS(IREG).EQ.999) THEN
   CALL MATUSR(IREG)
   GOTO 100
ENDIF
CALL CLEARS
IF (MATRLS(IREG).GE.200.AND.MATRLS(IREG).LE.299) THEN
   MNUM=I
   REWIND 4
   PRES=0.0
   GOTO 35
ENDIF
12 REWIND 4
WRITE (6,2)
WRITE (6,3) MATNMS(IREG),IREG
NTEST=0
10 READ (4,4) MNUM,MNAME
   IF (MNUM.EQ.0) GOTO 11
   IF (MNUM.NE.0) READ(*) SPRES,EPRES,PINC
   IK=1
   IF (MATRLS(IREG),EQ.MNUM) THEN
      WRITE (6,56) SPRES,EPRES,PINC
      NTEST=1
      P(IK)=SPRES
      P(IK)=P(IK-1)+PINC
      IF (P(IK).LT.EPRES) GOTO 1
   ENDIF
   IF (NTEST.EQ.1) GOTO 11
   GOTO 10
11 CALL READRE (PRES)
   NTEST=0
   DO 5 IK=1,NP
      IF (ABS(PRES-P(IK)).LE.O.O1) NTEST=1
5 CONTINUE
   IF (NTEST.EQ.0) THEN
      CALL CLEARS
      WRITE (6,*) ' THIS PRESSURE IS NOT IN THE DATA BASE.'
      GOTO 12
   ENDIF
12 KTEMP=0
35 IF (MNUM.EQ.0) GOTO 51
36 IF (MNUM.NE.0) READ (4,4) MNUM,MNAME
   IF (MNUM.NE.0) READ (4,*) SPRES,EPRES,PINC
   GOTO 36
51 IF (MNUM.NE.0) GOTO 36
   READ (4,6,END=9) MNUM,MNAME,NTABLE,NSETS,NPSIA
   DO 71 IK=1,NTABLE+2
      READ (4,74) NAMTAB(IK),TABUNT(IK),CONFAC(IK),PRPUNT(IK)
71 CONTINUE
   DO 97 J=1,NTABLE+2
      READ (4,*) (T(J,IK),J=1,NTABLE+2)
      DO 97 IK=1,NTABLE+2
97 CONTINUE
197 CONTINUE
9 DO 73 IK=3,NTABLE+2
IF (NAMTAB(IK).EQ.2) PROP='SPECIFIC HEAT'
IF (NAMTAB(IK).EQ.3) PROP='DENSITY'
IF (NAMTAB(IK).EQ.4) PROP='VISCOITY'
IF (NAMTAB(IK).EQ.5) PROP='ENTHALPHY'
IF (NAMTAB(IK).EQ.6) PROP='CONDUCTIVITY'
IF (MNUM.GE.200.AND.MNUM.LE.299) THEN
  WRITE (MODU,29) PROP,PRPUNT(IK),MNAME
ELSE
  WRITE (MODU,19) PROP,PRPUNT(IK),MNAME,PRES,TABUNT(2)
ENDIF
JJ=0
DO 75 IJ=1,NTSETS*NPSIA
  IF (ABS(T(I,IJ)-PRES).LE.0.01) THEN
    JJ=JJ+1
    T(JJ)=T(I,IJ)
  ENDIF
  IF (T(JJ).EQ.0) THEN
    WRITE (MODU,91) (PRTOUT(KK),KK=IJ+5)
  ELSE
    WRITE (MODU,91) (PRTOUT(KK),KK=IJ+5)
  ENDFINDO 75
LINES=JJ/6
DO 81 IK=3,NTABLE+2
  IF (NAMTAB(IK).EQ.2) NSEP=IK
  IF (NAMTAB(IK).EQ.3) NCND=IK
ENDIF
DO 81 JJ=0,NTSETS*NPSIA
  IF (ABS(T(I,IJ)-PRES).LE.0.01) THEN
    JJ=JJ+1
    T(JJ)=T(I,IJ)
  ENDIF
  IF (MNUM.GE.200.AND.MNUM.LE.299) THEN
    WRITE (MODU,84) MNAME
  ELSE
    WRITE (MODU,85) MNAME,PRES,TABUNT(2)
ENDIF
  IF (M.EQ.0) THEN
    WRITE (MODU,92) NAMTAB(IK),MATRLS(IREG),PRTOUT(I),PRTOUT(2)
  ELSE
    WRITE (MODU,92) NAMTAB(IK),MATRLS(IREG),PRTOUT(I),PRTOUT(2),PRTOUT(3),PRTOUT(4),PRTOUT(5),PRTOUT(6)
  ENDIF
  IF (M.EQ.2) THEN
    WRITE (MODU,93) NAMTAB(IK),MATRLS(IREG),PRTOUT(I),PRTOUT(2),PRTOUT(3),PRTOUT(4),PRTOUT(5),PRTOUT(6)
  ELSE
    WRITE (MODU,93) NAMTAB(IK),MATRLS(IREG),PRTOUT(I),PRTOUT(2),PRTOUT(3),PRTOUT(4),PRTOUT(5),PRTOUT(6)
  ENDIF
  IF (M.EQ.4) THEN
    WRITE (MODU,94) NAMTAB(IK),MATRLS(IREG),PRTOUT(I),PRTOUT(2),PRTOUT(3),PRTOUT(4),PRTOUT(5),PRTOUT(6)
  ELSE
    WRITE (MODU,94) NAMTAB(IK),MATRLS(IREG),PRTOUT(I),PRTOUT(2),PRTOUT(3),PRTOUT(4),PRTOUT(5),PRTOUT(6)
  ENDIF
  IF (M.EQ.6) THEN
    WRITE (MODU,95) NAMTAB(IK),MATRLS(IREG),PRTOUT(I),PRTOUT(2),PRTOUT(3),PRTOUT(4),PRTOUT(5),PRTOUT(6)
  ELSE
    WRITE (MODU,95) NAMTAB(IK),MATRLS(IREG),PRTOUT(I),PRTOUT(2),PRTOUT(3),PRTOUT(4),PRTOUT(5),PRTOUT(6)
  ENDIF
75 CONTINUE
lines=JJ/6
DO 81 IK=3,NTABLE+2
  IF (M.EQ.0) THEN
    WRITE (MODU,96) (PRTOUT(KK),KK=IJ+5)
  ELSE
    WRITE (MODU,96) (PRTOUT(KK),KK=IJ+5)
  ENDIF
76 CONTINUE
81 CONTINUE
80 CONTINUE

IF (ABS(T(1,I) - PRES) .LE. 0.01) THEN
JJ = JJ + 1
PRTOUT(JJ) = T(2, IJ)
TB = TARBUT(1)
!
IF (TB.EQ.'R' .AND. MATNSM(9) .EQ. 'F')
   PRTOUT(JJ) = PRTOUT(JJ) + 459.69
IF (TB.EQ.'C' .AND. MATNSM(9) .EQ. 'F')
   PRTOUT(JJ) = (1.8 * PRTOUT(JJ)) + 32
IF (TB.EQ. 'C' .AND. MATNSM(9) .EQ. 'R')
   PRTOUT(JJ) = ((PRTOUT(JJ) - 32) / 1.8) + 459.69
!
IF (TB.EQ.'K' .AND. MATNSM(9) .EQ.'F')
   PRTOUT(JJ) = (1.8 * (PRTOUT(JJ) - 273.16)) + 32
!
IF (TB.EQ. 'K' .AND. MATNSM(9) .EQ. 'R')
   PRTOUT(JJ) = ((PRTOUT(JJ) - 273.16) / 1.8) + 459.69
JJ = JJ + 1
!
ENDIF
CONTINUE
LINES = LINES - JJ / 6
K = I
!
IF (JJ.EQ.2) WRITE (MODU, 92) K, M_TRLS(IREG), PRTOUT(I), PRTOUT(2)
IF (JJ.EQ.4) WRITE (MODU, 93) K, M_TRLS(IREG), PRTOUT(I), PRTOUT(2), PRTOUT(3), PRTOUT(4)
IF (JJ.EQ.6) WRITE (MODU, 94) K, M_TRLS(IREG), PRTOUT(I), PRTOUT(2), PRTOUT(3), PRTOUT(4), PRTOUT(5), PRTOUT(6)
IF (JJ.GT.6) WRITE (MODU, 95) K, M_TRLS(IREG), PRTOUT(I), PRTOUT(2), PRTOUT(3), PRTOUT(4), PRTOUT(5), PRTOUT(6)
!
M = MOD(JJ, 6)
!
DO 86 II = 2, LINES
!
L = LINES - II * 6 + 1
!
IF (M.EQ.0 .AND. II.NE.1) WRITE (MODU, 91) (PRTOUT(KK), KK = IJ + 5)
!
IF (M.EQ.2) WRITE (MODU, 96) PRTOUT(IJ), PRTOUT(IJ + 1)
!
IF (M.EQ.4) WRITE (MODU, 97) PRTOUT(IJ), PRTOUT(IJ + 1), PRTOUT(IJ + 2), PRTOUT(IJ + 3)
!
GOTO 36
!
CONTINUE
!
M = MOD(JJ, 6)
!
DO 86 II = 2, LINES
!
L = LINES - II * 6 + 1
!
IF (M.EQ.0 .AND. II.NE.1) WRITE (MODU, 91) (PRTOUT(KK), KK = IJ + 5)
!
IF (M.EQ.2) WRITE (MODU, 96) PRTOUT(IJ), PRTOUT(IJ + 1)
!
IF (M.EQ.4) WRITE (MODU, 97) PRTOUT(IJ), PRTOUT(IJ + 1), PRTOUT(IJ + 2), PRTOUT(IJ + 3)
!
GOTO 36
!
86 CONTINUE
!
M = MOD(JJ, 6)
!
DO 86 II = 2, LINES
!
L = LINES - II * 6 + 1
!
IF (M.EQ.0 .AND. II.NE.1) WRITE (MODU, 91) (PRTOUT(KK), KK = IJ + 5)
!
IF (M.EQ.2) WRITE (MODU, 96) PRTOUT(IJ), PRTOUT(IJ + 1)
!
IF (M.EQ.4) WRITE (MODU, 97) PRTOUT(IJ), PRTOUT(IJ + 1), PRTOUT(IJ + 2), PRTOUT(IJ + 3)
!
GOTO 36
!
831 FORMAT (1X, A1)
2 FORMAT (/,' THE FOLLOWING IS THE RANGE OF PRESSURES IN THE')
3 FORMAT (' MATERIAL DBASE FOR ', AI6, ' IN REGION ', I1, ': /')
4 FORMAT (1X, A16)
SUBROUTINE MATUSR(IREG)

COMMON/REGION/NTBETA, NBETA, BETA, RIN, TVOL,
* ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
* TNKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)

COMMON/UNITS/MODU, SINDA

LOGICAL REGNS

CHARACTER*f6 MNAME, MATNMS, RGNNMS

CHARACTER*f5 TABUNT(10)

CHARACTER*I TB

CHARACTER*f3 PROP

DIMENSION P(100), T(IO, 700), NAMTAB(10), PRTOUT(1000)

CALL CLEARS

WRITE (6, *) ' PLEASE ENTER THE NUMBER CORRESPONDING TO THE STATE'

WRITE (6, *) ' OF THE MATERIAL TO CREATE:'

WRITE (6, *)

I - LIQUID

2 - SOLID

3 - GAS

CALL READIN (LSG, I, 3)

LSG = LSG * 100

MAX = 0

DO I = 1, 5

IF (1.NE.IREG) THEN

IF (MATRLS(1) - LSG .LT. 100 .AND. MATRLS(I) - LSG .GT. MAX)

MAX = MATRLS(I) - LSG

ENDIF

CONTINUE

MATRLS(IREG) = LSG + MAX + 1

MNUNM = MATRLS(IREG)

IF (LSG.NE.200) THEN

CALL CLEARS

WRITE (6, *) ' ENTER THE PRESSURE FOR REGION # ', IREG, ' (PSIA)'

CALL READIN (PRESS)

TABUNT(2) = 'PSIA'

ENDIF

CALL CLEARS

WRITE (6, *) ' ENTER NUMBER OF TABLES YOU WISH TO INPUT (MIN = 2)'

WRITE (6, *) ' NOTE: SPECIFIC HEAT AND CONDUCTIVITY MUST BE GIVEN'

CALL READIN (NTABLE, I, 999)

NAMTAB(3) = 2

DO 2 I = 2, NTABLE

CALL CLEARS

WRITE (6, *) ' ENTER THE NUMBER CORRESPONDING TO TABLE #', I

WRITE (6, *) ' 1. VISCOSITY'

WRITE (6, *) ' 2. ENTHALPY'

WRITE (6, *) ' 3. DENSITY'

CALL READIN (NTYPE, I, 3)

IF (NTYPE.EQ.1) NAMTAB(I + 2) = 4

CONTINUE

RETURN

END
IF (NTYPE.EQ.2) NAMTAB(I+2)=5
IF (NTYPE.EQ.3) NAMTAB(I+2)=3
CONTINUE
DO 3 I=1,NTEMP
CALL CLEAR
WRITE (6,*) 'ENTER THE NEW TEMPERATURE (R)'
CALL READ
DO 3 I=1,NTEMP
CALL CLEAR
T(I, I) = TEMP
IF (NAMTAB(J+2).EQ.2) PROP = 'SPECIFIC HEAT'
IF (NAMTAB(J+2).EQ.3) PROP = 'DENSITY'
IF (NAMTAB(J+2).EQ.4) PROP = 'VISCOSITY'
IF (NAMTAB(J+2).EQ.5) PROP = 'ENTHALPY'
IF (NAMTAB(J+2).EQ.6) PROP = 'CONDUCTIVITY'
IF (LSG.NE.200) WRITE (6,7)
IF (LSG.EQ.200) WRITE (6,8)
CONTINUE
DO 73 IK=3,NTAB+2
IF (NAMTAB(IK).EQ.2) PROP = 'SPECIFIC HEAT'
IF (NAMTAB(IK).EQ.3) PROP = 'DENSITY'
IF (NAMTAB(IK).EQ.4) PROP = 'VISCOSITY'
IF (NAMTAB(IK).EQ.5) PROP = 'ENTHALPY'
IF (NAMTAB(IK).EQ.6) PROP = 'CONDUCTIVITY'
IF (MNUM.GE.200.AND.MNUM.LE.299) THEN
WRITE (MODU, 29)
ENDIF
JJ=0
DO 75 IJ=1,NTEMP
IF (ABS(T(I,IJ)-PRESS).LE.0.01) THEN
JJ = JJ+1
PRTOUT(JJ) = T(2, IJ)
TB = TABUNT(I)
* PRTOUT(JJ) = T(2, IJ)
TB = TABUNT(I)
IF (TB.EQ.'R'.AND.MATNL(9).EQ.'F')
PRTOUT(JJ) = T(2, IJ)
IF (TB.EQ.'F'.AND.MATNL(9).EQ.'R')
PRTOUT(JJ) = T(2, IJ)
IF (TB.EQ.'C'.AND.MATNL(9).EQ.'F')
PRTOUT(JJ) = T(2, IJ)
IF (TB.EQ.'K'.AND.MATNL(9).EQ.'F')
PRTOUT(JJ) = T(2, IJ)
ELSE
WRITE (MODU, 19)
ENDIF
JJ=JJ+1
PRTOUT(JJ) = T(IK, IJ)
CONTINUE
LINES = JJ/6
IF (JJ.EQ.2) WRITE (MODU, 92)
IF (JJ.EQ.4) WRITE (MODU, 93)
IF (JJ.EQ.6) WRITE (MODU, 94)
ELSE
WRITE (MODU, 95)
ENDIF
M = MOD(JJ, 6)
DO 76 II=2,LINES
I = ((II-1)*6)+1
IF (M.EQ.0.AND.L.IE.11) WRITE (MODU, 91)
IF (M.NE.0.AND.II.EQ.L) WRITE (MODU, 91) (PRTOUT(KK),KK=IJ,IJ+5)  CRY05610
IF (M.NE.0.AND.II.NE.L) WRITE (MODU, 91) (PRTOUT(KK),KK=IJ,IJ+5)  CRY05620
IF (M.EQ.0.AND.II.EQ.L) WRITE (MODU, 98) PRTOUT(IJ),PRTOUT(IJ+1),  CRY05630
* PRTOUT(IJ+2),PRTOUT(IJ+3),PRTOUT(IJ+4),PRTOUT(IJ+5)  CRY05640

76 CONTINUE
  IJ-((LINES)*6)+1  CRY05650
  IF (NTEMP.EQ.11) GOTO 73  CRY05660
  IF (M.EQ.2) WRITE (MODU, 96) PRTOUT(IJ),PRTOUT(IJ+1)  CRY05680
  IF (M.EQ.4) WRITE (MODU, 97) PRTOUT(IJ),PRTOUT(IJ+1),PRTOUT(IJ+2),  CRY05690
* PRTOUT(IJ+3)  CRY05700

73 CONTINUE
  IF (MNUM.GE.200.AND.MNUM.LE.299) THEN
    WRITE (MODU, 84) MATNMS(IREG)  CRY05740
  ELSE
    WRITE (MODU, 85) MATNMS(IREG), PRESS, TABUNT(2)  CRY05770
  ENDIF

DO 81 IK=3,NTABLE+2  CRY05780
  IF (NAMTAB(IK).EQ.2) NSP=IK  CRY05790
  IF (NAMTAB(IK).EQ.6) NCND=IK  CRY05800

81 CONTINUE
  JJ=0  CRY05810
  DO 82 II=1,NTEMP  CRY05820
    IF (ABS(T(I,IJ)-PRESS).LE.0.01) THEN
      JJ=JJ+1  CRY05830
      PRTOUT(JJ)=T(2,IJ)  CRY05840
      TB=TABUNT(I)  CRY05850
      IF (TB.EQ.'R'.AND.MATNMS(9).EQ.'F')  CRY05860
        PRTOUT(JJ)=PRTOUT(JJ)-459.69  CRY05870
      IF (TB.EQ.'F'.AND.MATNMS(9) .EQ.'R')  CRY05880
        PRTOUT(JJ)+459.69  CRY05890
      IF (TB.EQ.'C'.AND.MATNMS(9).EQ.'F')  CRY05900
        PRTOUT(JJ)*(1.8*(PRTOUT(JJ)-32))  CRY05910
      IF (TB.EQ.'C'.AND.MATNMS(9).EQ.'R')  CRY05920
        PRTOUT(JJ)-((PRTOUT(JJ)-32)/1.8)+459.69  CRY05930
      IF (TB.EQ.'K'.AND.MATNMS(9).EQ.'F')  CRY05940
        PRTOUT(JJ)-((PRTOUT(JJ)-273.16)+32)  CRY05950
      IF (TB.EQ.'K'.AND.MATNMS(9).EQ.'R')  CRY05960
        PRTOUT(JJ)-((PRTOUT(JJ)-32)/1.8)+459.69  CRY05970
      JJ=JJ+1  CRY05980
      PRTOUT(JJ)=T(NSP,IJ)*T(NCND,IJ)  CRY05990
    ENDIF

82 CONTINUE
  LINES=JJ/6  CRY06000
  K=1  CRY06010
    IF (JJ.EQ.2) WRITE (MODU, 92) K,MATRLS(IREG),  CRY06020
      PRTOUT(1),PRTOUT(2)  CRY06030
    IF (JJ.EQ.4) WRITE (MODU, 93) K,MATRLS(IREG),  CRY06040
      PRTOUT(1),PRTOUT(2),PRTOUT(3),PRTOUT(4)  CRY06050
    IF (JJ.EQ.6) WRITE (MODU, 94) K,MATRLS(IREG),  CRY06060
      PRTOUT(1),PRTOUT(2),PRTOUT(3),PRTOUT(4),  CRY06070
      PRTOUT(5),PRTOUT(6)  CRY06080
    IF (JJ.GT.6) WRITE (MODU, 95) K,MATRLS(IREG),  CRY06090
      PRTOUT(1),PRTOUT(2),PRTOUT(3),PRTOUT(4),  CRY06100
      PRTOUT(5),PRTOUT(6)  CRY06110
  M=MOD(JJ,6)  CRY06120
  DO 86 II=1,LINES  CRY06130
    L-LINES  CRY06140
    IJ-((LINES)*6)+1  CRY06150
      IF (M.EQ.0.AND.II.NE.L) WRITE (MODU, 91) (PRTOUT(KK),KK=IJ,IJ+5)  CRY06160
      IF (M.EQ.0.AND.II.NE.L) WRITE (MODU, 91) (PRTOUT(KK),KK=IJ,IJ+5)  CRY06170
    IF (M.EQ.2) WRITE (MODU, 96) PRTOUT(IJ),PRTOUT(IJ+1),  CRY06180
    IF (M.EQ.4) WRITE (MODU, 97) PRTOUT(IJ),PRTOUT(IJ+1),PRTOUT(IJ+2),  CRY06190
* PRTOUT(IJ+3)  CRY06200

86 CONTINUE
  M=MOD(JJ,6)  CRY06210
  IJ-((LINES)*6)+1  CRY06220
  IF (NTEMP.GT.I; THEN  CRY06230
    IF (M.EQ.2) WRITE (MODU, 96) PRTOUT(IJ),PRTOUT(IJ+1)  CRY06240
    IF (M.EQ.4) WRITE (MODU, 97) PRTOUT(IJ),PRTOUT(IJ+1),PRTOUT(IJ+2),  CRY06250
* PRTOUT(IJ+3)  CRY06260

231
SUBROUTINE CYLND

COMMON/REGION/NTHETA, NBetas, ETA, RIN, TVOL,
* HOUT (9), REGNs (9), NLAYRS (9), TEMPS (9),
* THICK (9), THKLAY (9), MATRLS (9), RNMS (9)
COMMON/REGINP/MATT, DIST, THK, NLAY, MATN, RGNAM
COMMON/CYDATA/CYLHGT, NCYL
COMMON/HTXGRS/ NHX, HMTEMP (10), NRHX (10), NLHX (10), NTHHX (10),
* LNGTHX (10)
COMMON/UNITS/MODU, SINDA
COMMON/ULLAGE/NLUL4, NLUL5, NLH4, NLH5, THK4, THK5, NL4, NL5
COMMON/NODDAT/NODNUM (100000), VOL (10000)

LOGICAL REGNS
CHARACTER*16 MLABL
CHARACTER*16 RNMS, MMNAME, MATNMS, RGNAM, MATN
CHARACTER*10 TYPE1, TYPE2
CHARACTER*1 CT, LG
CHARACTER*6 LIQVAP

IF (NTOP.EQ.2) TYPE1 = 'FLAT'  
IF (NTOP.EQ.3) TYPE1 = 'SPHERICAL'  
IF (NTOP.EQ.4) TYPE1 = 'ELLIPTICAL'
IF (NBOT.EQ.2) TYPE2 = 'FLAT'
IF (NBOT.EQ.3) TYPE2 = 'SPHERICAL'
IF (NBOT.EQ.4) TYPE2 = 'ELLIPTICAL'
DO 4 I = 1, 5
  IF (I.EQ.1) NUMNOD = 2001
  IF (I.EQ.2) NUMNOD = 4001
  IF (I.EQ.3) NUMNOD = 6001
  IF (I.EQ.4) NUMNOD = 8001
  IF (I.EQ.5) NUMNOD = 1001
  IF (REGNS (I)) THEN
    NLAY = NLAYRS (I)
    TMP = TEMPS (I)
    THK = THICK (I)
    DIST = ROUT (I)
    MAT = MATRLS (I)
    MATNMS = MATNMS (I)
    RGNAM = RNMS (I)
  ELSE
    GOTO 4
  ENDIF

END
CALL CYLSEC(I,NUMNOD)

IF (NTOP.EQ.1) GOTO 4
IF (NTOP.EQ.2) CALL FEND(I,NUMNOD,FTTHK,NFTLAY,2)
IF (NTOP.EQ.3) CALL EEND(I,NUMNOD,E2TRAT,1)
IF (NTOP.EQ.4) CALL EEND(I,NUMNOD,E2TRAT,1)

4 CONTINUE

IF (PCTFUL.GT.0.001.AND.CT.EQ.'1') CALL ULLIG
IF (PCTFUL.GE.0.001) THEN
   DO 91 I=8001,10000
      IF (NLGS(I).LT.100.AND.NLGS(I).LE.199) NLGS(I)=NLGS(I)+200
   CONTINUE
ENDIF
IF (PCTEUL.GT.0.001.AND.CT.EQ.'O') CALL ULLIG
WRITE (MODU, 170)
CALL RITNDS(NTHETA,2,1001,1,0,TEMPS(1),0.0,MATNMS(1))
WRITE (MODU, 171)
NN=I
WRITE (MODU, 172) 'I'
DO 46 I=2001,3000
   IF (MODNUM(I).EQ.0) GOTO 46
   IF (VOL(I).NE.VOL(I+1).OR.NLGS(I).NE.NLGS(I+1)) THEN
      NUMNOD=I-NN+1
      NARY=NLGS(NUMNOD)
      CALL RITNDS(NN,1,NUMNOD,1,NARY,TEMPS(1),VOL(I),MATNMS(1))
      NN=I
   ENDIF
   IF (VOL(I).EQ.VOL(I+1).AND.NLGS(I).EQ.NLGS(I+1)) NN=NN+1
   CONTINUE
WRITE (MODU, 173) 'I'
WRITE (MODU, 170)
CALL RITNDS(NTHETA,2,3001,1,0,TEMPS(1),0.0,MATNMS(1))
WRITE (MODU, 171)
NN=I
IF (REGNS(2)) THEN
   WRITE (MODU,172) '2'
   DO 47 I=4001,5000
      IF (MODNUM(I).EQ.0) GOTO 47
      IF (VOL(I).NE.VOL(I+1).OR.NLGS(I).NE.NLGS(I+1)) THEN
         NUMNOD=I-NN+1
         NARY=NLGS(NUMNOD)
         CALL RITNDS(NN,1,NUMNOD,1,NARY,TEMPS(2),VOL(I),MATNMS(2))
         NN=I
      ENDIF
      IF (VOL(I).EQ.VOL(I+1).AND.NLGS(I).EQ.NLGS(I+1)) NN=NN+1
   CONTINUE
WRITE (MODU, 173) '2'
WRITE (MODU, 170)
CALL RITNDS(NTHETA,2,5001,1,0,TEMPS(2),0.0,MATNMS(2))
WRITE (MODU, 171)
ENDIF
NN=I
IF (REGNS(3)) THEN
   WRITE (MODU,172) '3'
   DO 48 I=6001,7000
      IF (MODNUM(I).EQ.0) GOTO 48
      IF (VOL(I).NE.VOL(I+1).OR.NLGS(I).NE.NLGS(I+1)) THEN
         NUMNOD=I-NN+1
         NARY=NLGS(NUMNOD)
         CALL RITNDS(NN,1,NUMNOD,1,NARY,TEMPS(3),VOL(I),MATNMS(3))
         NN=I
      ENDIF
      IF (VOL(I).EQ.VOL(I+1).AND.NLGS(I).EQ.NLGS(I+1)) NN=NN+1
   CONTINUE
WRITE (MODU, 173) '3'
WRITE (MODU, 170)
CALL RITNDS(NTHETA,2,7001,1,0,TEMPS(3),0.0,MATNMS(3))
WRITE (MODU, 171)
ENDIF
NN=1
IF (REGNS(4)) WRITE (MODU, 172) '4'
DO 49 I=8001,9000
IF (NODNUM(I).EQ.0) GOTO 49
IF (VOL(I).NE.VOL(I+1).OR.NLGS(I).NE.NLGS(I+1)) THEN
  NUMNOD=I-NN+1
  NARY=NLGS(NUMNOD)
  CALL RITNDS(NN,1,NUMNOD,1,NARY,TEMPS(4),VOL(I),MATNMS(4))
  NN=1
ENDIF
49 CONTINUE
IF (REGNS(4)) WRITE (MODU, 173) '4'
NN=1
IF (REGNS(5)) THEN
  WRITE (MODU, 170)
  CALL RITNDS (NTHETA, 2,9001,1,0,TEMPS(5),0.00,MATNMS(I))
  WRITE (MODU, 171)
  WRITE (MODU, 172) '5'
  DO 50 I=9001,10000
    IF (NODNUM(I).EQ.0) GOTO 50
    IF (VOL(I).NE.VOL(I+1).OR.NLGS(I).NE.NLGS(I+1)) THEN
      NUMNOD=I-NN+1001
      NARY=NLGS(NUMNOD)
      CALL RITNDS(NN,1,NUMNOD,1,NARY,TEMPS(5),VOL(I),MATNMS(5))
      NN=1
    ENDIF
  50 CONTINUE
  WRITE (MODU, 173) '5'
ENDIF
IF (NHX.GT.0) WRITE (MODU, 178)

RETURN
END

SUBROUTINE CYLSEC (I,NUMNOD)

COMMON/REGION/NTHETA,NBETAS,BETA,RIN,TVOL,
  ROUT(9),REGNS(9),NLAYRS(9),TEMPS(9),
  THICK(9),THKLAY(9),MATRLS(9),MATNMS(9),RGNNMS(9)
COMMON/CONINP/MATT,DIST,THK,NLAY,MATN
COMMON/REGINP/MATT,DIST,THK,NLAY,MATN
COMMON/NODDAT/NODNUM(IO000),VOL(IO000),NLGS(IO000)
COMMON/CYDATA/CYLHGT,NCYLAY

LOGICAL REGNS
CHARACTER*f6 RGNNMS,MNAME, MATNMS, RGNAM,MATN
HGT=CYLHGT/NCYLAY
DIN=ROUT(I)-THICK(I)
NDIV=NLAYRS(I)
WIDTH=THICK(I)/NDIV
DOUT=DIN÷WIDTH
ANG=BETA
NARY=1000+MATRLS(I)
DO I J=I,NDIV
  RADI=A/G*DOUT
  RA/D2=ANG*DIN
  VOLO=WIDTH*HGT* ((RADI+RAD2)/2.)

RETURN
END

SUBLTINE CYLSEC (I,NUMNOD)
IF (J.LT.NDIV) NL-NCYLAY+NUMNOD
IF (J.EQ.NDIV) NL-NCYLAY+NUMNOD-1
DO 3 IJ-NUMNOD, NL
  NODNUM (IJ)-1
  VOL(IJ)-VOLU
  NLGS (IJ)-NARY
CONTINUE
3 CONTINUE
CALL RITNDS (NCYLAY, 1, NUMNOD, 1, NARY, TEMPS (I), VOLU, MATNMS (I))
NUMNOD=NUMNOD+NCYLAY
DIN=DIN+WIDTH
DOUT=DOUT+WIDTH
CONTINUE
1 CONTINUE
RETURN
END

SUBROUTINE SEND (I, NUMNOD,NFLAY, NWHICH)
COMMON/REGION/NTHETA,NBETAS,BETA,RIN,T_VOL,
  * ROUT (9), REGNS (9), NLAYRS (9), TEMPS (9),
  * THICK (9), THKLAY (9), MATRLS (9), MATNMS (9), RGNNMS (9)
COMMON/TOBOT/NOBOT, NFLAY, NSTLAY, NETLAY, NFSLAY, NSBLAY, NEBLAY, NMLAY, CRY0900
  * ETAT, EBRAT, FTTHK, FBTN
COMMON/REGION/FYMT, DIST, NLAY, MATN, RGNAM
COMMON/NC0DAT/CYHT, KYHT, NGLAY
COMMON/COMM/ACC/ACCLIQ
COMMON/STUFF/CMAT, ACCLIQ, MATN
LOGICAL REGNS
logical+16 RGNNMS, MNAME, MATNMS, RGNAM, MATN
HGT=FTTHK/NFLAY
DIN=ROUT (I)-THICK (I)
NDIV=NLAYRS (I)
ANG=BETA
WIDTH=THICK (I)/NDIV
DOUT=DIN+WIDTH
NARY=1000 + M1RALS (I)
DO 1 J=1, NDIV
  RAD1=ANG*DOUT
  RAD2=ANG*DIN
  VOLU=WIDTH*HGT*{(RAD1+RAD2)/2.}
DO 3 IJ=NUMNOD, NUMNOD+NFLAY
  NODNUM (IJ)-1
  VOL(IJ)-VOLU
  NLGS (IJ)-NARY
CONTINUE
3 CONTINUE
CALL RITNDS (NFLAY, 1, NUMNOD, 1, NARY, TEMPS (I), VOLU, MATNMS (I))
NUMNOD=NUMNOD+NFLAY
DOUT=DOUT+WIDTH
DIN=DIN+WIDTH
1 CONTINUE
RETURN
END

SUBROUTINE FEND (I, NUMNOD,FTHK, NFLAY, NWHICH)
COMMON/REGION/NTHETA,NBETAS,BETA,RIN,TVOL,
  * ROUT (9), REGNS (9), NLAYRS (9), TEMPS (9),
  * THICK (9), THKLAY (9), MATRLS (9), MATNMS (9), RGNNMS (9)
COMMON/TOBOT/NOBOT, NFLAY, NSTLAY, NETLAY, NFSLAY, NSBLAY, NEBLAY, NMLAY, CRY0902
  * ETAT, EBRAT, FTTHK, FBTN
COMMON/REGION/FYMT, DIST, NLAY, MATN, RGNAM
COMMON/NC0DAT/CYHT, KYHT, NGLAY
COMMON/COMM/ACC/ACCLIQ
COMMON/STUFF/CMAT, ACCLIQ, MATN
COMMON/STUFF/ CMAT, ACCLIQ, MATN
LOGICAL REGNS
logical+16 RGNNMS, MNAME, MATNMS, RGNAM, MATN
HGT=FTHK/NFLAY
DIN=ROUT (I)-THICK (I)
NDIV=NLAYRS (I)
ANG=BETA
WIDTH=THICK (I)/NDIV
DOUT=DIN+WIDTH
NARY=1000 + M1RALS (I)
DO 1 J=1, NDIV
  RAD1=ANG*DOUT
  RAD2=ANG*DIN
  VOLU=WIDTH*HGT*{(RAD1+RAD2)/2.}
DO 3 IJ=NUMNOD, NUMNOD+NFLAY
  NODNUM (IJ)-1
  VOL(IJ)-VOLU
  NLGS (IJ)-NARY
CONTINUE
3 CONTINUE
CALL RITNDS (NFLAY, 1, NUMNOD, 1, NARY, TEMPS (I), VOLU, MATNMS (I))
NUMNOD=NUMNOD+NFLAY
DOUT=DOUT+WIDTH
DIN=DIN+WIDTH
1 CONTINUE
RETURN
END
LOGICAL REGNS
CHARACTER*16 RGNMNS,MNAME, MATNMS, RGNAM, MATN

C THIS SECTION CALCULATES THE NODE AREA FOR A NODE IN EITHER THE TOP OR BOTTOM SPHERE.

TH = THKLAY(I)
NARY = 1000 + MATRLS(I)
IF (NWHICH.EQ.1) THEN
  NSLAY=NSTLAY
  THETA0=0
ELSE
  NSLAY=NSBLAY
  THETA0=PI/2.
ENDIF
DTHETA=PI/2./NSLAY
DO 1 I=1,NLAYRS(I)
  IF (I.EQ.4) THEN
    NSLAY=NSLAY+I
    THETA0=0.5
  ELSE
    NSLAY=NSLAY+I
    THETA0=THETA0+DTHETA
  ENDIF
  AREA=BETA*R*R*(COS(THETA0)+COS(THETA2))*DTHETA/2.
  NODNUM(NUMNOD)=NUMNOD
  VOL(NUMNOD)=AREA*TH
  NLGS(NUMNOD)=NARY
  NUMNOD=NUMNOD + 1
CONTINUE
CONTINUE
RETURN
END

SUBROUTINE END(I, NUMNOD, ERAT, NWHICH)

COMMON/REGION/NTHETA, NBETAS, BETA, RIN, TVOL,
  * REGNS(9), REGNS(9), NLAYRS(9), MNAME(9), MATNMS(9), RGNMNS(9)
COMMON/REGINP/MATT, DIST, THK, NLAY, MATN, RGNAM
COMMON/UNITSD/MODU, SINDA
COMMON/VOLUME/VOLLIQ, ACCLIQ
COMMON/STUFF/ NHTT, PI, CONVY, CONVR, THETA0, OTHETA, NBASOS, ROUTSF,
  * BNCOEF(2)

LOGICAL REGNS
CHARACTER*16 RGNMNS, MNAME, MATNMS, RGNAM, MATN

C THIS SECTION CALCULATES THE NODE AREA FOR A NODE IN EITHER THE TOP OR BOTTOM SPHERE.

TH = THKLAY(I)
BETA = 1.0
NARY = 1000 + MATRLS(I)
IF (NWHICH.EQ.1) THEN
  NELAY=NETLAY
  THETA0=0
ELSE
  NELAY=NEBLAY

RETURN
END
THETA0 = PI/2.
ENDIF

DTHETA = PI/2./NELAY

DO 1 M=1,NLAYRS(I)
   IF (I.GE.4) EL=M-I
   IF (I.LE.3) EL=NLAYRS(I)-M+1
   AI=ROUT(I)-TH* (EL)
   AO=AI+TH
   BI=ROUT(I)*ERAT
   BO=BI+TH

PRINT ' EEND, I,NW, RO, TH0,DTH',I,NWHICH, ROUT(I),THETA0,DTHETA
PRINT *,'M,AI,AO, BI,BO',M, AI,AO, BI,BO

DO 2 JPOS=0,NELAY-I
   IF (NWHICH.EQ.I) POS_-I*(JPOS+I)
   IF (NWHICH.EQ.2) POS=JPOS
   THETA2 = THETAO-POS* DTHETA
   THETAI = THETA2-DTHETA

   NODNUM(NUMNOD) = NUMNOD •
   AAVG = (AO+AI)/2.
   BAVG = (BO+BI)/2.
   THAVG = (THETAI+THETA2)/2.
   COSAVG = COS (THAVG)
   SINAVG = SIN (THAVG)

   FIRST = ((BETA*COSAVG)/2.) * ((AAVG*BAVG)/SQRT((BAVG*BAVG*
   COSAVG*COSAVG + (AAVG*AAVG*SINAVG*SINAVG)))
   SND = AO*BO* (ATAN ((AO/BO) *TAN (THETA2)) -ATAN ((AO/BO) *TAN (THETAI)))
   THR = AI*BI* (ATAN ((AI/BI) *TAN (THETA2)) -ATAN ((AI/BI) *TAN (THETAI)))
   VOL(NUMNOD) = FIRST* (SND-THR)
   VOL(NUMNOD) = FIRST, SND,THR,VOL',FRST, SND, THR, VOL(NUMNOD)

NLGS(NUMNOD) = NARY
NUMNOD = NUMNOD + 1
2 CONTINUE
1 CONTINUE
RETURN

END

SUBROUTINE HXARR

COMMON/REGION/NTHETA, NBETAS,BETA, RIN,TVOL,
** ROUT (9), REGMS (9), NLAYRS (9), TEMPS (9), THICK (9),
** THKLAY (9), MTRLMS (9), MATTMS (9), RGNNMS (9)
COMMON/TOPBOT/NBOT, NFTLAY, NSTLAY, NETLAY, NFBLAY, NSBLAY,
** NEBLAY, ETRAT, EBRAT, FTTHK, FSTH
COMMON/CYDATA/CYLHGT,NCYLAY
COMMON/VOLUME/VOLLIQ, ACCLIQ
COMMON/NTHK2/ NTHK,NTHMP (10), NRHK (10), NLHK (10), NTHHK (10),
** LTHHK (10)
COMMON/NK/NDS (1000), NCMD (1000), INDEX
COMMON/UNITS/MODU, SINDA

LOGICAL REGNS
CHARACTER*16 MIABL
CHARACTER*16 RGNNMS, MNAME, MATTMS, RGNN, MATN

NUMNOD=1
INDEX=0

NUMBER=20001
NTEI=NETLAY+NETLAY+NSTLAY
NLAY=NFBLAY+NSBLAY+NLAY

DO 1 I=1,NHK
   IF (NRHK(1).EQ.1) NSTART=2001
   IF (NRHK(1).EQ.2) NSTART=4001
   IF (NRHK(1).EQ.3) NSTART=6001
   IF (NRHK(1).EQ.4) NSTART=8001
   IF (NRHK(1).EQ.5) NSTART=1001
   IF (NRHK(1).EQ.6) NSTART=2001
   IF (NRHK(1).EQ.7) NSTART=3001
   IF (NRHK(1).EQ.8) NSTART=4001
   IF (NRHK(1).EQ.9) NSTART=5001
   IF (NRHK(1).EQ.10) NSTART=6001
   IF (NRHK(1).EQ.11) NSTART=7001
   IF (NRHK(1).EQ.12) NSTART=8001
   IF (NRHK(1).EQ.13) NSTART=9001
   IF (NRHK(1).EQ.14) NSTART=1001
   IF (NRHK(1).EQ.15) NSTART=1101
   IF (NRHK(1).EQ.16) NSTART=1201
   IF (NRHK(1).EQ.17) NSTART=1301
   IF (NRHK(1).EQ.18) NSTART=1401
   IF (NRHK(1).EQ.19) NSTART=1501
   IF (NRHK(1).EQ.20) NSTART=1601

CRY09810
CRY09820
CRY09830
CRY09840
CRY09850
CRY09860
CRY09870
CRY09880
CRY09890
CRY09900
CRY09910
CRY09920
CRY09930
CRY09940
CRY09950
CRY09960
CRY09970
CRY09980
CRY09990
CRY10000
CRY10010
CRY10020
CRY10030
CRY10040
CRY10050
CRY10060
CRY10070
CRY10080
CRY10090
CRY10100
CRY10110
CRY10120
CRY10130
CRY10140
CRY10150
CRY10160
CRY10170
CRY10180
CRY10190
CRY10200
CRY10210
CRY10220
CRY10230
CRY10240
CRY10250
CRY10260
CRY10270
CRY10280
CRY10290
CRY10300
CRY10310
CRY10320
CRY10330
CRY10340
CRY10350
CRY10360
CRY10370
CRY10380
CRY10390
CRY10400
CRY10410
CRY10420
CRY10430
CRY10440
CRY10450
CRY10460
CRY10470
CRY10480
CRY10490
CRY10500
IF (NRHX(I).EQ.4.AND.NLMX(I).EQ.1) NSTART=1001
IF (NRHX(I).EQ.5.AND.NLMX(I).EQ.1) NSTART=8001+
* (NCYLAY*NLAYRS(4)) + (NTLAY*NLAYRS(4)) + (NSLAY*NLAYRS(4))
LEVEL=NLTHX(I)
LEVEL2=0
LEVEL3=0
DO 2 K=1,NLTHX(I)
IF (LEVEL.LE.NBLAY) NUM=NSTART+(NBLAY*(NLHX(I)-1))+
* LEVEL2=LEVEL2+1
* LEVEL2=LEVEL2+1
NDST=LEVEL2+1+NUM
IF (LEVEL.GT.NBLAY.AND.NBLAY.GT.0.AND.LEVEL.LE.NBLAY+NCYLAY) THEN
LEVEL2=LEVEL2+1
NUM=NSTART+(NBLAY*NLAYRS(NRHX(I)))+
* (NCYLAY*(NLHX(I)-1))
ENDIF
NDST=LEVEL2+1+NUM
IF (LEVEL.GT.NBLAY+NCYLAY) THEN
LEVEL3=LEVEL3+1
NUM=NSTART+(NBLAY*NLAYRS(NRHX(I)))+
* (NCYLAY*NLAYRS(NRHX(I)))+
* (NTLAY*(NLHX(I)-1))+LEVEL3
ENDIF
INDEX=INDEX+1
LEVEL=LEVEL+I
NDS(INDEX)=NUM
NCND(INDEX)=NUMBER
2 CONTINUE
NUMBER=NUMBER-1
1 CONTINUE
RETURN
END
SUBROUTINE CYLCHS
COMMON/REGION/NTHETA,NBETAS,BETA,RIN,TVOL,
ROUT(9),REGNS(9),TEMPS(9),THICK(9),
THKLAY(9),MATRLS(9),MATNMS(9),RGNNMS(9)
COMMON/TOPBOT/NTOP,NBOT,NFTLAY,NSTLAY,NETLAY,NFBLAY,NSBLAY,
NEBLAY,ETRAT,EBRAT,FTTHK,FBTHK
COMMON/CYDATA/CYLSHT
COMMON/HXGRS/NHX,HXTEMP(10),NRHX(10),NLHX(10),NTHNX(10),
LNGT(HX(I0))
COMMON/UNITS/MODU,SINDA
COMMON/NANB/NAI(5,3),NBI(5,3)
COMMON/HX/HDS(1000),NCD(1000),INDEX
COMMON/NODDAT/NODNUM(10000),VOL(10000),NLGS(10000)
COMMON/STUFF/NHTT,PI,CONVY,CONVR,TNETAO,DTNETA,NBASOS,ROUTSF,
BNCOEF(2)
LOGICAL REGNS
CHARACTER*16 MATNMS
CALL HXARR
NCON=1
DO 1 I=1,5
IF (.NOT.REGNS(I)) GOTO 1
NB=2000*I+1
NA=NB-1000
IF (I.EQ.4) NA=1001
IF (NBOT.NE.0) THEN
NAI(1,1)=NA
NBI(1,1)=NB
NBI(1,2)=NB(1,1)+NBLAY+NSLAY+NEBLAY
NBI(1,2)=NB(1,1)+NSLAY+NEBLAY)*NLAYRS(I)
ENDIF
IF (NBOT.EQ.1) THEN
NAI(1,1)=0
NBI(1,1)=0
NBI(1,2)=NA
NBI(1,2)=NB
ENDIF
NAI(1,3)=NAI(1,2)+NCYLAY
1 CONTINUE
CALL CYLALL (I, NA, NB, NCON)
IF (NTOP . EQ. 2)
  CALL FCND (I, NA (I), NB (I), NCON, NFBLAY, FBTHK, 1)
IF (NTOP . EQ. 3)
  CALL SCND (I, NA (I), NB (I), NCON, NSTLAY, 2)
IF (NTOP . EQ. 4)
  CALL ECND (I, NA (I), NB (I), NCON, NETLAY, ETRAT, 2)
CONTINUE
RETURN
END

SUBROUTINE CYLALL (I, NA, NB, NCON)
  COMMON/REGION/NTHETA, NBETAS, BETA, RIN, TVOL,
  * ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
  * THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
  COMMON/TOPBOT/NTOP, NBOT, NFTLAY, NSTLAY, NETLAY, NFBLAY, NSBLAY,
  * NEBLAY, ETRAT, ERRAT, FTTHK, FBTHK
  COMMON/CYDATA/CYLHGT, NCYLAY
  COMMON/HTXGRS/ NHX, HXTEMP(10), NRHX(IO), NLHX(10), LNGTHX(10),
  * LNGTHX(IO)
  COMMON/UNITS/MODU, SINDA
  COMMON/HX/NDS(IOOO), NCND(IOOO), INDEX
  COMMON/NODDAT/NODNUM(IOOO), VOL(IOOO), NLGS(IOOO)
  COMMON/STUFF/ WHTT, PI, CONVY, CONVR, THETA(1), NTHH(1), NTHH(IO),
  * NTHH(IO)
  COMMON/REGION/THETA, BETA, RIN, TVOL,
  * ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
  * THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
  COMMON/TOPBOT/NTOP, NBOT, NFTLAY, NSTLAY, NETLAY, NFBLAY, NSBLAY,
  * NEBLAY, ETRAT, ERRAT, FTTHK, FBTHK
  COMMON/CYDATA/CYLHGT, NCYLAY
  COMMON/HTXGRS/ NHX, HXTEMP(10), NRHX(IO), NLHX(10), LNGTHX(10),
  * LNGTHX(IO)
  COMMON/UNITS/MODU, SINDA
  COMMON/HX/NDS(IOOO), NCND(IOOO), INDEX
  COMMON/NODDAT/NODNUM(IOOO), VOL(IOOO), NLGS(IOOO)
  COMMON/STUFF/ WHTT, PI, CONVY, CONVR, THETA(1), NTHH(1), NTHH(IO),
  * NTHH(IO)
  COMMON/REGION/THETA, BETA, RIN, TVOL,
  * ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
  * THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
  COMMON/TOPBOT/NTOP, NBOT, NFTLAY, NSTLAY, NETLAY, NFBLAY, NSBLAY,
  * NEBLAY, ETRAT, ERRAT, FTTHK, FBTHK
  COMMON/CYDATA/CYLHGT, NCYLAY
  COMMON/HTXGRS/ NHX, HXTEMP(10), NRHX(IO), NLHX(10), LNGTHX(10),
  * LNGTHX(IO)
  COMMON/UNITS/MODU, SINDA
  COMMON/HX/NDS(IOOO), NCND(IOOO), INDEX
  COMMON/NODDAT/NODNUM(IOOO), VOL(IOOO), NLGS(IOOO)
  COMMON/STUFF/ WHTT, PI, CONVY, CONVR, THETA(1), NTHH(1), NTHH(IO),
  * NTHH(IO)
  COMMON/REGION/THETA, BETA, RIN, TVOL,
  * ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
  * THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
  COMMON/TOPBOT/NTOP, NBOT, NFTLAY, NSTLAY, NETLAY, NFBLAY, NSBLAY,
  * NEBLAY, ETRAT, ERRAT, FTTHK, FBTHK
  COMMON/CYDATA/CYLHGT, NCYLAY
  COMMON/HTXGRS/ NHX, HXTEMP(10), NRHX(IO), NLHX(10), LNGTHX(10),
  * LNGTHX(IO)
  COMMON/UNITS/MODU, SINDA
  COMMON/HX/NDS(IOOO), NCND(IOOO), INDEX
  COMMON/NODDAT/NODNUM(IOOO), VOL(IOOO), NLGS(IOOO)
  COMMON/STUFF/ WHTT, PI, CONVY, CONVR, THETA(1), NTHH(1), NTHH(IO),
  * NTHH(IO)
  COMMON/REGION/THETA, BETA, RIN, TVOL,
  * ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
  * THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
  COMMON/TOPBOT/NTOP, NBOT, NFTLAY, NSTLAY, NETLAY, NFBLAY, NSBLAY,
  * NEBLAY, ETRAT, ERRAT, FTTHK, FBTHK
  COMMON/CYDATA/CYLHGT, NCYLAY
  COMMON/HTXGRS/ NHX, HXTEMP(10), NRHX(IO), NLHX(10), LNGTHX(10),
  * LNGTHX(IO)
  COMMON/UNITS/MODU, SINDA
  COMMON/HX/NDS(IOOO), NCND(IOOO), INDEX
  COMMON/NODDAT/NODNUM(IOOO), VOL(IOOO), NLGS(IOOO)
  COMMON/STUFF/ WHTT, PI, CONVY, CONVR, THETA(1), NTHH(1), NTHH(IO),
  * NTHH(IO)
  COMMON/REGION/THETA, BETA, RIN, TVOL,
  * ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
  * THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
  COMMON/TOPBOT/NTOP, NBOT, NFTLAY, NSTLAY, NETLAY, NFBLAY, NSBLAY,
  * NEBLAY, ETRAT, ERRAT, FTTHK, FBTHK
  COMMON/CYDATA/CYLHGT, NCYLAY
  COMMON/HTXGRS/ NHX, HXTEMP(10), NRHX(IO), NLHX(10), LNGTHX(10),
  * LNGTHX(IO)
  COMMON/UNITS/MODU, SINDA
  COMMON/HX/NDS(IOOO), NCND(IOOO), INDEX
  COMMON/NODDAT/NODNUM(IOOO), VOL(IOOO), NLGS(IOOO)
  COMMON/STUFF/ WHTT, PI, CONVY, CONVR, THETA(1), NTHH(1), NTHH(IO),
  * NTHH(IO)
  COMMON/REGION/THETA, BETA, RIN, TVOL,
  * ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
  * THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
  COMMON/TOPBOT/NTOP, NBOT, NFTLAY, NSTLAY, NETLAY, NFBLAY, NSBLAY,
  * NEBLAY, ETRAT, ERRAT, FTTHK, FBTHK
  COMMON/CYDATA/CYLHGT, NCYLAY
  COMMON/HTXGRS/ NHX, HXTEMP(10), NRHX(IO), NLHX(10), LNGTHX(10),
  * LNGTHX(IO)
  COMMON/UNITS/MODU, SINDA
  COMMON/HX/NDS(IOOO), NCND(IOOO), INDEX
  COMMON/NODDAT/NODNUM(IOOO), VOL(IOOO), NLGS(IOOO)
  COMMON/STUFF/ WHTT, PI, CONVY, CONVR, THETA(1), NTHH(1), NTHH(IO),
  * NTHH(IO)
  COMMON/REGION/THETA, BETA, RIN, TVOL,
  * ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
  * THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
  COMMON/TOPBOT/NTOP, NBOT, NFTLAY, NSTLAY, NETLAY, NFBLAY, NSBLAY,
  * NEBLAY, ETRAT, ERRAT, FTTHK, FBTHK
  COMMON/CYDATA/CYLHGT, NCYLAY
  COMMON/HTXGRS/ NHX, HXTEMP(10), NRHX(IO), NLHX(10), LNGTHX(10),
  * LNGTHX(IO)
  COMMON/UNITS/MODU, SINDA
  COMMON/HX/NDS(IOOO), NCND(IOOO), INDEX
  COMMON/NODDAT/NODNUM(IOOO), VOL(IOOO), NLGS(IOOO)
  COMMON/STUFF/ WHTT, PI, CONVY, CONVR, THETA(1), NTHH(1), NTHH(IO),
  * NTHH(IO)
NCOV2=NUM+1  
NCOV3=NCLAY-NCOV1-NCOV2  
IF (NPART.EQ.1) THEN  
  NCOV1=0  
  NCOV2=NUM+1  
  NCOV3=NCLAY-NCOV1-NCOV2  
ENDIF  
GOTO 72  
ENDIF  
CONTINUE  
72  
N=NCOV1  
IF (NTSTX.EQ.1) THEN  
  WRITE (MODU, 123) I  
ENDIF  
GOTO 91  
32.  
WRITE (MODU, 124) I  
ENDIF  
IF (NTSTX.EQ.0) THEN  
  CALL RITCND(5,NCON,N,NA,1,NB,1,NARY,0,F,O,MATNMS(I))  
  NCON=NCON+N  
  NA-NA+NCYLAY  
  NB=NB+NCYLAY  
ENDIF  
IF (NDIV.EQ.1.AND.I.EQ.5) GOTO 776  
DIN=DOUT  
DOUT=DIN+(BETA*((THICK(I)/NDIV)/2.))  
F=(((DIN+DOUT)/2.)*CYLHGT/NCLAY)/(((THICK(I)/NDIV)/2.))  
NA=NB  
NB=NB+NCYLAY  
IF (NDIV.EQ.1) THEN  
  NB=3001+(NFBLAY+NSBLAY+NEBLAY)  
  IF (I.EQ.1) THEN  
    NF=5001+(NFBLAY+NSBLAY+NEBLAY)  
    IF (I.EQ.2) THEN  
      NB=7001+(NFBLAY+NSBLAY+NEBLAY)  
      ENDIF  
      ENDIF  
    ENDIF  
  ELSE  
    NF=3001+(NFBLAY+NSBLAY+NEBLAY)  
    IF (I.EQ.1) THEN  
      NB=5001+(NFBLAY+NSBLAY+NEBLAY)  
      IF (I.EQ.2) THEN  
        NB=7001+(NFBLAY+NSBLAY+NEBLAY)  
        ENDIF  
        ENDIF  
      ENDIF  
    ENDIF  
  ENDIF  
  NTSTX=0  
DO 21 K=1,INDEX  
  IF (NDS(K).GE.NA.AND.NDS(K).LE.NA+NCYLAY) THEN  
    NUM=0  
    NPART=0  
    DO 51 IK=K,INDEX  
      IF (NDS(IK).EQ.NDS(IK-1).AND.IK.EQ.K) NPART=1  
      IF (NDS(IK-1).EQ.NDS(IK+1).AND.  
        NCND(IK).EQ.NCND(IK+1).AND.  
        NDS(IK+1).EQ.NDS(IK+1)) NUM=NUM+1  
      ENDIF  
    ENDIF  
  CONTINUE  
21  
CONTINUE  
51  
NTSTX=1  
NAHX=NDS(K)  
NBHX=NCND(K)  
N=ABS(20000-NCND(K))  
NTH=NTSTX(N)=(NFBLAY+NEBLAY+NSBLAY)  
NCOV1=NTH-1  
NCOV2=NUM+1  
NCOV3=NCLAY-NCOV1-NCOV2  
IF (NPART.EQ.1) THEN  
  NCOV1=0  
  NCOV2=NUM+1  
  NCOV3=NCLAY-NCOV1-NCOV2  
ENDIF  
GOTO 776  
240
ENDIF
GOTO 22
ENDIF

21
22
N=NCOVI
IF (NTSTHX.EQ.1) THEN
WRITE (MODU,123) I
ENDIF
IF (N.EQ.0) GOTO 92
CALL RITCND(5,NCON,N,1,NA,1,NB,1,NARY,0,F,O,MATNMS(I))
NP=NA+N
NCON=NCON+N
N=NCOV2
NH=NBHX
CALL RITCND(5,NCON,N,1,NH,1,NF,1,NARY,0,F,O,MATNMS(I))
NCON=NCON+N
CALL RITCND(5,NCON,N,1,NH,1,NARY,0,F,O,MATNMS(I))
ENDIF
CONTINUE
N-NCOVI
IF (NTSTHX.EQ.0) THEN
IF (NLGS(NA).NE.NLGS(NB).AND.J.NE.I) THEN
FT=(DOUT+DOUT+(THICK(I)/NDIV))/2.
F2=(FT*(CYLHGT/NCYLAY))/((THICK(I)/NDIV)/2.)
CALL RITCND(7,NCON,N,1,NA,1,NB,1,NLGS(NA),NLGS(NB),F,F2,MATNMS(I))
ELSE
CALL RITCND(5,NCON,NCYLAY,1,NA,1,NB,1,NARY,0,F,O,MATNMS(I))
ENDIF
ENDIF
ENDIF

776
ENDIF

IF (J.GT.I.AND.J.LT.NDIV) THEN
DIN = DOUT
DOUT = DIN + (BETA*(THICK(I)/NDIV))
F=(DOUT+(DOUT/2.)*CYLHGT/NCYLAY)/{(THICK(I)/NDIV)/2.}
CALL RITCND(7,NCON,N,1,NA,1,NB,1,NLGS(NA),NLGS(NB),F,F2,MATNMS(I))
ENDIF

END IF
END IF
END IF
END IF
END IF
GOTO 24

ENDIF

24

CONTINUE

IF (NTSTHX.EQ.1) THEN
WRITE (MODU, 123) I
IF (N.EQ.0) GOTO 93
IF (NLGS(NA).NE.NLGS(NB)) THEN
FT = (DOUT + DOUT + (THICK(I)/NDIV))/2.
F2 = (FT * (CYLHGTY/NCYLAY))/((THICK(I)/NDIV)/2.)
CALL RITCND(7, NCON, N, 1, NA, 1, NB, 1, NLGS(NA), NLGS(NB), F, F2, MATNMS(I))
ELSE
CALL RITCND(5, NCON, N, 1, NA, 1, NB, 1, NARY, 0, F, 0, MATNMS(I))
ENDIF

93

N = NA + N
NCON = NCON + N
N = NCOV2
NP = NA + N
NCON = NCON + N
NH = NB
CALL RITCND(5, NCON, N, 1, NA, 1, NB, 1, NARY, 0, F, 0, MATNMS(I))
NCON = NCON + N
NCOV3
N = NCON + N
NCON = NCON + N
NARY
CALL RITCND(5, NCON, N, 1, NA, 1, NB, 1, NARY, 0, F, 0, MATNMS(I))
NCON = NCON + N
NCON = NCON + N
IF (NCOV3.EQ.0) GOTO 323
WRITE (MODU, 124) I
ENDIF

IF (NTSTHX.EQ.0) THEN
IF (NLGS(NA).NE.NLGS(NB)) THEN
FT = (DOUT + DOUT + (THICK(I)/NDIV))/2.
F2 = (FT * (CYLHGTY/NCYLAY))/((THICK(I)/NDIV)/2.)
CALL RITCND(7, NCON, N, 1, NA, 1, NB, 1, NLGS(NA), NLGS(NB), F, F2, MATNMS(I))
ELSE
CALL RITCND(5, NCON, N, 1, NA, 1, NB, 1, NARY, 0, F, 0, MATNMS(I))
ENDIF
NCON = NCON + N
ENDIF
NBHX = NCND(K)
N = ABS(20000+NCND(K))
NTH = NBHX(N) - NFBLAY + NEBLAY + NSBLAY
NCOV1 = NTH-1
NCOV2 = NTH
NCOV3 = NCLAY - NCOV1 - NCOV2
IF (NFRANT.EQ.1) THEN
  NCOV1 = 0
  NCOV2 = NTH
  NCOV3 = NCLAY - NCOV1 - NCOV2
ENDIF
GOTO 26
ENDIF

CONTINUE
N = NCOV1
IF (NTHSHX.EQ.1) THEN
  WRITE (MODU, 123) I
  IF (N.EQ.0) GOTO 95
  IF (NGS(NA).NE.NGS(NB)) THEN
    FT = (FT + (THICK(I)/NIVI)) / 2.
    F2 = (F2 + (CYLHGT/NCLAY))/(THICK(I)/NIVI)**2.
    CALL RITCND(7, NCON, N, NA, 1, NB, 1, NGS(NA), NGS(NB), F, F2, MATNMS(I))
  ELSE
    CALL RITCND(5, NCON, N, I, NA, 1, NB, 1, MATNMS(I))
  ENDIF
NCON = NCON + N
N = NCOV2
NH = NBHX
CALL RITCND(5, NCON, N, 1, NA, 1, NB, 1, NH, 1, NARY, 0, MATNMS(I))
NCON = NCON + NH
N = NCOV1
CALL RITCND(5, NCON, N, 1, NA, 1, NB, 1, NCOV1, 0, MATNMS(I))
NCON = NCON + N
IF (NCOV3.EQ.0) GOTO 331
N = NCOV3
NP = NA + N
NCST = NCON + N
NCON = NCON + N
NH = NB + N
CALL RITCND(5, NCON, N, 1, NA, 1, NB, 1, NH, 1, NARY, 0, MATNMS(I))
NCON = NCON + NH
NCON = NCON + N
IF (NTHSHX.EQ.0) THEN
  IF (NGS(NA).NE.NGS(NB)) THEN
    FT = (FT + (THICK(I)/NIVI)) / 2.
    F2 = (F2 + (CYLHGT/NCLAY))/(THICK(I)/NIVI)**2.
    CALL RITCND(7, NCON, N, I, NA, 1, NB, 1, NGS(NA), NGS(NB), F, F2, MATNMS(I))
  ELSE
    CALL RITCND(5, NCON, N, I, NA, 1, NB, 1, MATNMS(I))
  ENDIF
NCON = NCON + NP
NCON = NCON + NP
NCON = NCON + NP
NCON = NCON + NP
N = NCOV2
NH = NB + NCOV2
CALL RITCND(5, NCON, N, I, NA, 1, NB, 1, NH, 1, NARY, 0, MATNMS(I))
NCON = NCON + NP
NCON = NCON + NP
IF (I.EQ.1) NSTART = 2001 + (LAYB*NLAYRS(I))
IF (I.EQ.2) NSTART = 4001 + (LAYB*NLAYRS(I))
IF (I.EQ.3) NSTART = 6001 + (LAYB*NLAYRS(I))
IF (I.EQ.4) NSTART = 8001 + (LAYB*NLAYRS(I))
IF (I.EQ.5) NSTART = 9001 + (LAYB*NLAYRS(I))
ALEN = CYLHGT + NCLAY
DIN = ROUT(I) - THICK(I)
DOUT = DIN + (THICK(I)/NIVI)
DO 31 J = 1, NIVI
  NA = NSTART
  NB = NSTART + 1
  WIDTH = (DOUT + DOUT) / 2.* (THICK(I)/NIVI)
  CONTINUE
F=WIDTH/ALEN
NSAME=0
FLAG=0
DO 131 KI-NA+NCYLAY-1
   IF (.NOT.NLGS(KI).LE.NLGS(KI+I)) FLAG=1
   IF (NLGS(KI).EQ.NLGS(KI+I).AND.FLAG.EQ.0) NSAME=NSAME+I
CONTINUE
IF (NSAME.NE.NCYLAY) NSAME=NSAME+I
IF (NSAME.NE.NCYLAY) THEN
   N1=NCYLAY-NSAME
   CALL RITCND(5, NCON,NSAME-1,1,NA,1,NB,1,NARY,0,F,0,MATNMS(I))
   N1=NA+NSAME-1
   N2=NB+NSAME-1
   NCON=NCON+NSAME-1
   NLA=NLGS(N1)+5000
   NLB=NLGS(N2)+5000
   CALL RITCND(7, NCON, I,I,NI, I, N2,I,NL,I,NB, F, MATNMS (I))
   CALL RITCND(5,NCON,NK,-1,NI+I,I,N2+I,I,NL,0,F,O,MATNMS(I))
ELSE
   CALL RITCND(5,NCON, NCYLAY-1,1,NA,1,NB,1,NARY,0,F,0,MATNMS(I))
ENDIF
NSTART=NSTART+NCYLAY
DIN=DOUT
DOUT=DIN+(THICK(I)/NDIV)
31 CONTINUE
WRITE (MODU, 172) I
123 FORMAT(7X,'REM START OF H/X IN REGION #',1I1)
124 FORMAT(7X,'REM END OF H/X IN REGION #',1I1)
171 FORMAT(7X,'REM CONDUCTOR BLOCK FOR REGION #',1I1,' BEGINS (CYL).')
172 FORMAT(7X,'REM CONDUCTOR BLOCK FOR REGION #',1I1,' ENDS (CYL).')
RETURN
END
SUBROUTINE FCND (I,NA,NB,NCON,NFLAY,FTHK,N WHICH)
COMMON/REGION/NTHETA,NBETAS,BETA,RIN,TVOL,
* ROUT(9),REGNS(9),NLA(9),THKS(9),THICK(9),
* THK(9),MATRLS(9),MATNMS(9)
COMMON/TOPBOT/ NTOP,NBT(9),NBOT(9),NBL(9),NBOT(9),NBL(9),
* NBL,EBOT,EBOT,FTHK,FBTHK
COMMON/CYDATA/CYLHGT,NCYLAY
COMMON/HTXGRS/ HX,HXTEMP(IO),NRHX(IO),NLHX(IO),NTHNX(IO),
* LNGTHX(IO)
COMMON/UNITS/MODU,SINDA
COMMON/HX/NDS(IO),NCND(IO),INDEX
COMMON/NDDAT/NDNUM(IO),VOL(IO),NLGS(IO)
COMMON/STUFF/ NTHT,PI,CONVY,CONVR,THETA0,DTBETA,NBASOS,ROUTSF,
* NCOEF(2)
LOGICAL REGNS
CHARACTER*16 MLABL
CHARACTER*16 RGNMMS,MNAME,MATNMS,RGNAM,MATN
CHARACTER*6 TYPE
IF (NWHICH.EQ.1) TYPE='BOTTOM'
IF (NWHICH.EQ.2) TYPE='TOP'
IF (NWHICH.EQ.1) WRITE (MODU,171) I,TYPE
IF (NWHICH.EQ.2) WRITE (MODU,173) I,TYPE
NARY=6000+MATRLS(I)
NDIV=NLAYRS(I)
IF (NDIV.EQ.I) NTEMP=1
IF (NDIV.GT.1) NTEMP=NDIV-1
DO 1 J=1,NTEMP
  IF (J.EQ.1) THEN
    DIN=BETA* (ROUT(I)-THICK(I))
    DOUT=DIN + (BETA* (THICK(I)/NDIV)/2.)
    F=-(DIN+DOUT)/2.)*((FTHK/NFLAY))/((THICK(I)/NDIV)/2.)
  NTSTHX=0
  DO 71 K=1,NINDEX
    IF (NDS(K).GE.NA.AND.NDS(K).LE.NA+NFLAY) THEN
      NUM=0
      DO 52 IK=1,KINDEX
        IF (NDS(IK+1).EQ. (NDS(IK)+1).AND.
          NCND(IK+1).EQ.NCND(IK)) NUM=NUM+1
        IF (NDS(IK+1).NE. (NDS(IK)+1)) GOTO 62
      CONTINUE
      NTSTHX=1
      NAHX=NDS(K)
      NBHX=NCND(K)
      N=ABS(20000+NCND(K))
      IF (NWHICH.EQ.2) NTH=NTHHX(N)-NCYLAY-(NFBLAY+NEBLAY)
      ELSE
        NTH=NTHHX(N)
        NCOVI=NTH-I
        NCOV2=NUM+1
        NCOV3=NFLAY-NCOVI-NCOV2
        GOTO 72
      ENDIF
    CONTINUE
      N=NUM+1
      IF (NTSTHX.EQ.1) THEN
        WRITE (MODU, 123) I
      ELSE
        CALL RITCND(5,NCON,N,I,NH,BHX,NARY,0,F,F2,MATNMS(I))
        NCON=NCON+N
        NH=NBHX+NCOV1
        CALL RITCND(5,NCON,N,I,NP,BHX,NARY,0,F,F2,MATNMS(I))
        NCON=NCON+N
        NP=NP+N
        CALL RITCND(5,NCON,N,I,NP,BHX,NARY,0,F,F2,MATNMS(I))
        ELSE
          CALL RITCND(5,NCON,N,I,NH,BHX,NARY,0,F,F2,MATNMS(I))
        ENDIF
      WRITE (MODU, 124) I
    ENDIF
  ENDIF
  IF (NTSTHX.EQ.0) THEN
    IF (NLGS(NA).NE.NLGS(NB).AND.J.EQ.I) THEN
      FT= (DOUT+DOUT+ (THICK(I)/NDIV))/2.
      F2= (FT* (FTHK/NFLAY))/((THICK(I)/NDIV)/2.)
      CALL RITCND(7,NCON,N,1,NA, I,NB, I,NLGS(NA),NLGS(NB),F,
      F2,MATNMS(I))
    ELSE
      CALL RITCND(5,NCON,N,1,NA, I,NB, I,NARY,0,F,0,MATNMS(I)}
      NCON=NCON+N
      IF (NCON.EQ.1) AND (1.EQ.5) GOTO 776
  ENDIF
END
DIN-DOUT
DOHT=DIN
+ (BETA * (THICK(I)/NDIV)/2.))
F=((DIN+DOUT)/2.*((THICK(I)/NDIV)/2.)
NA=NB
NB=NB+NFLAY
IF (NDIV.EQ.1) THEN
IF (NWHICH.EQ.1) THEN
IF (I.EQ.3) NB=3001
IF (I.EQ.2) NB=5001
IF (I.EQ.3) NB=7001
ENDIF
IF (NWHICH.EQ.2) THEN
IF (I.EQ.1) NB=3001+NFLAY+NBLAY+NSBLAY+NEBLAY
IF (I.EQ.2) NB=5001+NFLAY+NBLAY+NSBLAY+NEBLAY
IF (I.EQ.3) NB=7001+NFLAY+NBLAY+NSBLAY+NEBLAY
ENDIF
ENDIF
ENDIF
ENDIF
NTSTHX=0
DO 21 K=1, INDEX
IF (NDS(K).GE.NA.AND.NDS(K).LE.NA+NFLAY) THEN
NUM=0
DO 51 IK=K, INDEX
* IF (NDS(IK).EQ.(NDS(IK)+1).AND.
  NCND(IK+1).EQ.NCND(IK)) NUM=NUM+1
IF (NDS(IK).NE.(NDS(IK)+1)) GOTO 32
CONTINUE
51 CONTINUE
32 NTSTHX=1
NAHX=NDS(K)
NBHX=NCND(K)
N=ABS(20000+NCND(K))
IF (NWHICH.EQ.1) NTH=NTHX(N)-NCRYLAY-(NCRYLAY+NCRYLAY)
IF (NWHICH.EQ.2) NTH=NTHX(N)-NCRYLAY-(NCRYLAY+NCRYLAY)
NCOV1=NTH-1
NCOV2=NUM+1
NCOV3=FLAY-NCOV1-NCOV2
GOTO 22
ENDIF
21 CONTINUE
22 N=NCOV1
IF (NTSTHX.EQ.1) THEN
WRITE (MODU, 123) I
IF (N.EQ.0) GOTO 64
IF (NLGS(NA).NE.NLGS(NB).AND.J.NE.I) THEN
FT=DOUT*DOUT+(THICK(I)/NDIV)/2.
F2=FT*(FTHK/NFLAY))/((THICK(I)/NDIV)/2.)
CALL RITCND(7, NCON,N,I,NA, I,NB, I,NLGS(NA),NLGS(NB),F,
F2,MATNMS(I))
ELSE
CALL RITCND(5, NCON,N,I,NH, I,NP, I,NARY, 0, F,0,MATNMS(I))
ENDIF
64 NP=NA+N
NCON=NCON+N
N=NCOVV
NH=NBHX
CALL RITCND(5, NCON,N,I,NH, I,NF, I,NARY, 0, F,0,MATNMS(I))
NCON=NCON+N
NH=NB+NCOV1
CALL RITCND(5, NCON,N,I,NF, I,NH, I,NARY, 0, F,0,MATNMS(I))
NCON=NCON+N
IF (NCOV3.EQ.0) GOTO 391
N=NCOV3
NP=NA+NCOV1+NCOV2
NH=NB+NCOV1+NCOV2
CALL RITCND(5, NCON,N,I,NF, I,NH, I,NARY, 0, F,0,MATNMS(I))
NCON=NCON+N
GOTO 64
391 WRITE (MODU,124) I
ENDIF
IF (NTSTHX.EQ.0) THEN
IF (NLGS(NA).NE.NLGS(NB).AND.J.NE.I) THEN
WRITE (MODU, 124) I
IF (N.EQ.0) GOTO 64
IF (NLGS(NA).NE.NLGS(NB).AND.J.NE.I) THEN
WRITE (MODU, 124) I
IF (N.EQ.0) GOTO 64
IF (NLGS(NA).NE.NLGS(NB).AND.J.NE.I) THEN
WRITE (MODU, 124) I
IF (N.EQ.0) GOTO 64
IF (NLGS(NA).NE.NLGS(NB).AND.J.NE.I) THEN
WRITE (MODU, 124) I
IF (N.EQ.0) GOTO 64
IF (NLGS(NA).NE.NLGS(NB).AND.J.NE.I) THEN
WRITE (MODU, 124) I
IF (N.EQ.0) GOTO 64
IF (NLGS(NA).NE.NLGS(NB).AND.J.NE.I) THEN
WRITE (MODU, 124) I
IF (N.EQ.0) GOTO 64
IF (NLGS(NA).NE.NLGS(NB).AND.J.NE.I) THEN
\[ F_T = \frac{(D_{OUT} + D_{OUT} + (T_{\text{THICK}}(I)/N_{\text{DIV}}))}{2}. \]

\[ F_T = \frac{(T_{\text{THICK}}(I)/N_{\text{DIV}})}{2}. \]

\[ \text{CALL RITCND}(7, N_{\text{CON}}, N_{\text{FLAY}}, 1, N_{A}, 1, N_{B}, N_{\text{LGS}}(N_{A}), N_{\text{LGS}}(N_{B}), F, F_2, \text{MATNMS}(I)) \]

\[ \text{ELSE} \]

\[ \text{CALL RITCND}(5, N_{\text{CON}}, N_{\text{FLAY}}, 1, N_{A}, 1, N_{B}, 1, N_{\text{ARY}}, 0, 0, 0, \text{MATNMS}(I)) \]

\[ \text{ENDIF} \]

\[ N_{\text{CON}} = N_{\text{CON}} + N_{\text{FLAY}} \]

\[ \text{ENDIF} \]

IF \((J.GT.1 . \text{AND.} J.LT.N_{\text{DIV}}) \) THEN

\[ D_{IN} = D_{OUT} \]

\[ D_{OUT} = \frac{(D_{IN} + D_{OUT} + (T_{\text{THICK}}(I)/N_{\text{DIV}}))}{2}. \]

\[ F = \frac{(F_{\text{THK}}/N_{\text{FLAY}})}{(T_{\text{THICK}}(I)/N_{\text{DIV}})/2}. \]

\[ N_{A} = N_{A} + N_{\text{FLAY}} \]

\[ N_{B} = N_{B} + N_{\text{FLAY}} \]

IF \((I.EQ.1) \) THEN

\[ N_{\text{SAVE}} = N_{\text{SAVE}} + N_{\text{FLAY}} \]

\[ N_{\text{TSX}} = 0 \]

DO \(23 \) \( K = 1, I_{\text{INDEX}} \)

IF \((N_{\text{DS}}(K) . G.E. N_{A}, \text{AND.} N_{\text{DS}}(K) \text{. LE.} N_{A} + N_{\text{FLAY}}) \) THEN

\[ N_{U} = 0 \]

DO \(33 \) \( I_{\text{K}} = K, I_{\text{INDEX}} \)

IF \((N_{\text{DS}}(I_{\text{K}} + I) . E Q. (N_{\text{DS}}(I_{\text{K}}) + 1) \) AND.

\[ N_{\text{CN}}(I_{\text{K}} + 1), \text{EQ.} N_{\text{CN}}(I_{\text{K}}) \) \) NUM = NUM + 1

IF \((N_{\text{DS}}(I_{\text{K}}) \text{. NE.} (N_{\text{DS}}(I_{\text{K}}) + 1)) \) GOTO 34

CONTINUE

34

\[ N_{\text{TSX}} = 1 \]

\[ N_{A_{\text{K}}} = N_{\text{DS}}(K) \]

\[ N_{B_{\text{K}}} = N_{\text{CN}}(K) \]

\[ N = \text{ABS}(20000 + N_{\text{CN}}(K)) \]

IF \((N_{\text{WHICH}}. \text{EQ.} 2) \) THEN

\[ N_{\text{TH}}(N_{\text{A}}) = N_{\text{TH}}(N_{\text{A}}) - N_{\text{FLAY}} + N_{\text{FLAY}} \]

\[ N_{\text{CO}} = N_{\text{TH}}(N_{\text{A}}) \]

\[ N_{\text{CO}2} = N_{\text{NUM}} + 1 \]

\[ N_{\text{CO}3} = N_{\text{FLAY}} - N_{\text{CO}1} - N_{\text{CO}2} \]

GOTO 24

ENDIF

CONTINUE

23

IF \((N_{\text{TSX}} . \text{EQ.} 1) \) THEN

WRITE (MODU, 123) I

IF \((M . \text{EQ.} 0) \) GOTO 65

IF \((N_{\text{LS}}(N_{A}) . \text{NE.} N_{\text{LS}}(N_{B}) \) AND \( J . \text{NE.} 1) \) THEN

\[ F_T = \frac{(D_{OUT} + D_{OUT} + (T_{\text{THICK}}(I)/N_{\text{DIV}}))}{2}. \]

\[ F_T = \frac{(T_{\text{THICK}}(I)/N_{\text{DIV}})}{2}. \]

\[ \text{CALL RITCND}(7, N_{\text{CON}}, N_{A}, 1, N_{B}, 1, N_{\text{ARY}}, 0, F, 0, 0, \text{MATNMS}(I)) \]

\[ \text{ELSE} \]

\[ \text{CALL RITCND}(5, N_{\text{CON}}, N_{A}, 1, N_{B}, 1, N_{\text{ARY}}, 0, 0, 0, 0, \text{MATNMS}(I)) \]

ENDIF

65

\[ N = N + N_{\text{A}} \]

\[ N_{\text{CON}} = N_{\text{CON}} + N_{\text{A}} \]

\[ N = N_{\text{CO}}2 \]

\[ N = N_{\text{CO}}2 \]

\[ N = N_{\text{CO}}2 \]

\[ N = N_{\text{CO}}2 \]

\[ N = N_{\text{CO}}2 \]

\[ \text{CALL RITCND}(5, N_{\text{CON}}, N_{A}, 1, N_{B}, 1, N_{\text{ARY}}, 0, 0, 0, 0, \text{MATNMS}(I)) \]

\[ N_{\text{CON}} = N_{\text{CON}} + N_{\text{CO}}2 \]

\[ \text{CALL RITCND}(5, N_{\text{CON}}, N_{A}, 1, N_{B}, 1, N_{\text{ARY}}, 0, 0, 0, 0, \text{MATNMS}(I)) \]

\[ N_{\text{CO}} = N_{\text{CON}} + N_{\text{CO}}2 \]

IF \((N_{\text{CO}}2 . \text{EQ.} 0) \) GOTO 311

\[ N = N_{\text{CO}}2 \]

\[ N = N_{\text{CO}}2 \]

\[ N = N_{\text{CO}}2 \]

\[ N = N_{\text{CO}}2 \]

\[ \text{CALL RITCND}(5, N_{\text{CON}}, N_{A}, 1, N_{B}, 1, N_{\text{ARY}}, 0, 0, 0, 0, \text{MATNMS}(I)) \]

\[ N_{\text{CON}} = N_{\text{CON}} + N_{\text{CO}}2 \]

WRITE (MODU, 124) I

ENDIF

IF \((N_{\text{TSX}} . \text{EQ.} 0) \) THEN

ENDIF
CALL RITCND(5, NCON, N, 1, NH, 1, NF, 1, NARY, 0, F, 0, MATNMS(I))
NCON = NCON + N
NH = NB + NCOV1
CALL RITCND(5, NCON, N, 1, NH, 1, NF, 1, NARY, 0, F, 0, MATNMS(I))
NCON = NCON + N
IF (NCOV3.EQ.0) GOTO 312
N = NCOV3
NP = NA + NCOV1 + NCOV2
NH = NB + NCOV1 + NCOV2
CALL RITCND(5, NCON, N, 1, NH, 1, NF, 1, NARY, 0, F, 0, MATNMS(I))
NCON = NCON + N
WRITE (MODU, 124) I
ENDIF
IF (NTRYH.EQ.0) THEN
IF (NLGS(NA).NE.NLGS(NB).AND.J.NE.I) THEN
FT = (DOUT + DOUT + (THICK(1)/NDIV))/2.
F2 = (F* (THICK(NFLAY))/((THICK(1)/NDIV))/2.)
CALL RITCND(7, NCON, NFLAY, 1, NA, 1, NB, 1, NLGS(NA), NLGS(NB), F, F2, MATNMS(I))
ELSE
CALL RITCND(5, NCON, NFLAY, 1, NA, 1, NB, 1, NARY, 0, F, 0, MATNMS(I))
ENDIF
NCON = NCON + NFLAY
ENDIF
1 CONTINUE
10 LAY = NFLAY + N2BLAY + NFLAY
IF (NWHICH.EQ.1) LAY = 0
IF (I.EQ.1) NSTART = 2001 + (LAY*NLAYRS(I))
IF (I.EQ.2) NSTART = 4001 + (LAY*NLAYRS(I))
IF (I.EQ.3) NSTART = 6001 + (LAY*NLAYRS(I))
IF (I.EQ.4) NSTART = 8001 + (LAY*NLAYRS(I))
IF (I.EQ.5) NSTART = 9001 + (LAY*NLAYRS(I))
ALEN = THICK/NFLAY
DIN = DOUT(I) - THICK(I)
DOUT = DIN + (THICK(I)/NDIV)
DO 31 J = 1, NDIV
NA = NSTART
NB = NSTART + 1
WIDTH = (DIN + DOUT)/2) * (THICK(I)/NDIV)
F = WIDTH/ALEN
IF (NLGS(NA).NE.NLGS(NB).AND.J.NE.1) THEN
CALL RITCND(7, NCON, NFLAY - 1, 1, NA, 1, NB, 1, NLGS(NA), NLGS(NB), F, F, MATNMS(I))
ELSE
CALL RITCND(5, NCON, NFLAY - 1, 1, NA, 1, NB, 1, NARY, 0, F, 0, MATNMS(I))
ENDIF
NCON = NCON + NFLAY - 1
NSTART = NSTART + NFLAY
DIN = DOUT(I)
DOUT = DIN + (THICK(I)/NDIV)
31 CONTINUE
IF (NWHICH.EQ.1) WRITE (MODU, 172) I, TYPE
IF (NWHICH.EQ.2) WRITE (MODU, 174) I, TYPE
123 FORMAT(TX, 'REM START OF H/X IN REGION #', II)
124 FORMAT(TX, 'REM END OF H/X IN REGION #', II)
171 FORMAT(TX, 'REM CONDUCTOR BLOCK FOR REGION #', II, ' BEGINS. ('',A6, **')')
172 FORMAT(TX, 'REM CONDUCTOR BLOCK FOR REGION #', II, ' ENDS. ('',A6, **')')
173 FORMAT(TX, 'REM CONDUCTOR BLOCK FOR REGION #', II, ' BEGINS. ('',A3, **')')
174 FORMAT(TX, 'REM CONDUCTOR BLOCK FOR REGION #', II, ' ENDS. ('',A3, **')')
RETURN
END
SUBROUTINE SCND (I, NA, NB, NCON, NSLAY, NWHICH)

COMMON/REGION/NTHETA, NBTAS, BETA, RIN, TVOL,
   * ROUT (9), REGNS (9), NLAYRS (9), THICK (9),
   * THKLAY (9), MTRLS (9), MATNMS (9), RGNMS (9)
COMMON/TOPBOT/NTOP, NBOT, NFTLAY, NSLAY, NFTLAY, NFTLAY, NSLAY,
   * NEBLAY, ETET, EBRAT, NTHK, NTBH
COMMON/CYDATA/CYLHGT, NCYL
COMMON/HXGS/ NHX, HXTEMP (10), NRHX (10), NLHX (10), LNHX (10),
   * LNGTHX (10)
COMMON/UNITS/MODU, SINDA
COMMON/HX/NDS (1000), NCX (1000), INDEX
COMMON/NCDAT/NCDNUM (10000), VOL (10000), NLGS (10000)
COMMON/STUFF/ NHTT, PI, CONV, CONVR, DTHETA, NBASOS, ROUTSF,
   * BNCOEF (2)

LOGICAL REGNS
CHARACTER*16 MLABL
CHARACTER*16 RGNNMS, MNAME, MATNMS, RGNAM, MATN
CHARACTER*6 TYPE

NOUT=NWB+1000
IF (NWHICH.EQ.1) TYPE='BOTTOM'
IF (NWHICH.EQ.2) TYPE='TOP'

IF (NWHICH.EQ.1) NSLAY = NSLAY
IF (NWHICH.EQ.2) NSLAY = NSTLAY

NC2=NB
NARY=6000+MATRLS (I)
TH=THKLAY (I)
DOUT = RIN+THKLAY(I)/4.0
BASE = NB
DO 1 J=1,NLAYRS(I)
   IF (J.GT.I) WRITE (MODU, 18) I, J-I, J, TYPE
   IF (J.EQ.I) LATEMP = NB
   IF (J.EQ.2) THEN
      NA = LATEMP
   ENDIF
   DO 2 IJ=0,NSLAY-1
      IF (I .EQ. 0.AND. J .EQ. 2) WRITE (MODU, 94) I, TYPE
      NTBH = 0
      NBH = 0
      DO 71 K = 1, INDEX
         IF (NDS (K).EQ.NA) THEN
            NTBH = 1
            NA = NA + NBH
            NBH = NCDNUM (K)
            GOTO 72
         ELSE
            CONTINUE
         ENDIF
91
      CONTINUE
      CALL AREACYL (I, IJ, OUT, DOUT, THICK (I), AREA (I, NWHICH)
      CALL RITCND (5, NCON, 1, 1, NA, 1, NBH, 1, NARY, 0,
         * AREA (I, MATNMS (I))
      NCON=NCON+1
      NA = NA + 1
      DHALF=DOUT-(THICK (I)/MLAYRS (I))
      CALL AREACYL (I, IJ, DHALF, THICK (I), AREA (I, NWHICH)
      CALL AREACYL (I, IJ, DHALF, THICK (I), AREA (I, NWHICH)
      NARY=NBH+5000
      CALL RITCND (5, NCON, 1, 1, NBH, 1, NARY, 0,
         * AREA (I, MATNMS (I))
      NB = NB + 1
      NCON=NCON+1
ENDIF
IF (NWHICH.EQ.0) THEN

250
CALL AREACYL (I, IJ, DOUT, THICK(I), AREA, NWHICH)
AREA = (AREA/TH/2.) * 4.
IF (J.NE.1) THEN
  D2 = DOUT + TH/4.
  CALL AREACYL (I, IJ, D2, THICK(I), AREA2, NWHICH)
  AREA2 = (AREA2/TH2) * 4.
  NL1 = NLGS (NA) + 5000
  NL2 = NLGS (NB) + 5000
  CALL RITCND (7, NCON, 1, I, NA, 1, NB, 1, NL1, NL2, AREA, AREA2, MATNMS(I))
ELSE
  NL1 = NLGS (NB) + 5000
  CALL RITCND (5, NCON, 1, I, NA, 1, NB, 1, NOUT, 1, NOUT, 0, AREA, 0, MATNMS(I))
ENDIF
NCON = NCON + 1
NA = NA + 1
NB = NB + 1
ENDIF
CONTINUE
DOUT = DOUT + TH/4.
1 CONTINUE
IF (1.EQ.5) GOTO 92
IF (NWHICH.EQ.2) NOUT = (2001*I) + 1000 + (NCYLAY + NSLAY + NBLAY + NBLAY)
NIN = NA
WRITE (MODU, 123) I, NLAYRS(I), TYPE
DO 32 J = 0, NSLAY - 1
  D2 = DOUT + TH/4.
  CALL AREACYL (I, J, D2, THICK(I), AREA, NWHICH)
  AREA = (AREA/TH/2.) * 4.
  NL1 = NLGS (NIN) + 5000
  NL2 = NLGS (NIN + 1) + 5000
  CALL RITCND (7, NCON, 1, I, NIN, 1, NOUT, 1, NOUT, 0, AREA, 0, MATNMS(I))
  MIN = NIN + 1
  NOUT = NOUT + 1
  NCON = NCON + 1
ENDIF
32 CONTINUE
92 CONTINUE
DOUT = DOUT + THLAY(I) / 4.0
DO 988 J = 1, NLAYRS(I)
  WRITE (MODU, 987) I, J, TYPE
  DO 989 I = 1, NSLAY - 1
    NC = WC2 + (NSLAY*(J-1)) + J
    IF (I.EQ.NSLAY-1) THEN
      CALL AREACYL (2, IJ, DOUT, THICK(I), AREA, NWHICH)
      IF (NWHICH.EQ.1) THEN
        IF (I.EQ.1) THEN
          CALL AREACYL (2, IJ, DOUT, THICK(I), AREA, NWHICH)
        ELSE
          CALL AREACYL (2, IJ, DOUT, THICK(I), AREA2, NWHICH)
        ENDIF
        NL1 = NLGS (NB) + 5000
        NL2 = NLGS (NB) + 5000
        CALL RITCND (7, NCON, 1, I, NB, 1, NB, 1, NOUT, 0, AREA, 0, MATNMS(I))
      ELSE
        IF (NWHICH.EQ.2) THEN
          CALL AREACYL (2, IJ, DOUT, THICK(I), AREA, NWHICH)
        ELSE
          CALL AREACYL (2, IJ, DOUT, THICK(I), AREA2, NWHICH)
        ENDIF
        NL1 = NLGS (NB) + 5000
        NL2 = NLGS (NB) + 5000
        CALL RITCND (7, NCON, 1, I, NB, 1, NB, 1, NOUT, 0, AREA, 0, MATNMS(I))
      ENDIF
      NCON = NCON + 1
    ENDIF
  ENDIF
989 CONTINUE
988 CONTINUE
251
\[\text{NBTEMP} = (2001^*I) + ((\text{NSBLAY} + \text{NEBLAY} + \text{NFBLAY})*\text{NLAYRS}(I) + (\text{NCYLAY})*J) - 1\]

\[\text{NAT} = (2001^*I) + ((\text{NSBLAY} + \text{NEBLAY} + \text{NFBLAY})*\text{NLAYRS}(I) + (\text{NCYLAY})*J) - 1\]

\[\text{IF (NWHICH.EQ.1) I1=I+1}\]

\[\text{IF (NWHICH.EQ.2) I1=I-1}\]

\[\text{CALL AREACYL (2, I1, DOUT, THICK(I), AREA2, 0)}\]

\[\text{NL1+NLGS (NAT) + 5000}\]

\[\text{NL2+NLGS (NBTEMP) + 5000}\]

\[\text{CALL RITCND (7, NCON, I, I, NAT, I, NBTEMP, I, NL1, NL2, AREA, AREA2, MATNMS(I))}\]

\[\text{NCON} = \text{NCON} + 1\]

\[\text{END}DEF\]

\[\text{END}\]

\[\text{989 CONTINUE}\]

\[\text{DOUT} = \text{DOUT} + \text{TH}/4.0\]

\[\text{CONTINUE}\]

\[\text{18 FORMAT(7X, 'REM RADIAL CONDUCTORS REGION', I2, ' LAYER', I2, ' TO LAYER ', I2, ' IN ', A6, ' END.' )}\]

\[\text{88 FORMAT(7X, 'REM RADIAL CONDUCTORS REGION', I2, ' LAYER', I2, ' TO LAYER 1 IN ', A6, ' END.' )}\]

\[\text{94 FORMAT(7X, 'REM RADIAL CONDUCTORS REGION', I2, ' LAYER', I2, ' TO BOUNDARY NODES IN ', A6, ' END.' )}\]

\[\text{123 FORMAT(7X, 'REM RADIAL CONDUCTORS REGION', I2, ' LAYER', I2, ' TO BOUNDARY NODES IN ', A6, ' END.' )}\]

\[\text{987 FORMAT(7X, 'REM CIRCUMFERENTIAL CONDUCTORS REGION', I2, ' LAYER', I2, ' NUMBER', I2, ' IN ', A6, ' END.' )}\]

\[\text{RETURN}\]

\[\text{END}\]

\[\text{SUBROUTINE ECND (I, NA, NB, NCON, NELAY, ERAT, NWHICH)}\]

\[\text{COMMON/REGION/NTHETA, NBETAS, BETA, RIN, TVOL, ROUT (9), REGNS (9), NLAYRS (9), THICK (9), THKLAY (9), MATRLS (9), RGNMS (9)}\]

\[\text{COMMON/TOP/BOT/NTOP, NBOT, NFTLAY, NSTLAY, NLAY, NFBLAY, NEBLAY, ETRAT, EBRAT, FTTMK, FBTHK}\]

\[\text{COMMON/CYDATA/CYLHGT, NCYLAY}\]

\[\text{COMMON/HTXGRS/NHX, HXTEMP(10), NRHX(10), NLHX(10), NTHHX(10), LNGTHX(10)}\]

\[\text{COMMON/UNITS/MODU, SINDA}\]

\[\text{CO_MON/HX/NDS(10DO), NCND(1000), INDEX}\]

\[\text{COMMON/MODAT/MODNUM(1000), VOL(1000), NLGS(10000)}\]

\[\text{COMMON/STUFF/NMTT, PT, CONVY, CONVR, THETAO, DTHETA, RBASOS, ROUTSF, BNCOEF (2)}\]

\[\text{LOGICAL REGNS}\]

\[\text{CHARACTER*16 MLABL}\]

\[\text{CHARACTER*16 RNMS, RNAME, MATNMS, RGNM, MATN}\]

\[\text{CHARACTER* TYPE}\]

\[\text{NOUT} = NB + 1000\]

\[\text{IF (NWHICH.EQ.1) TYPE='BOTTOM'}\]

\[\text{IF (NWHICH.EQ.2) TYPE='TOP'}\]

\[\text{IF (NWHICH.EQ.1) NELAY = NEBLAY}\]

\[\text{IF (NWHICH.EQ.2) NELAY = NSTLAY}\]

\[\text{NC2*NB}\]

\[\text{NARY = 6000 + MATRLS(I)}\]

\[\text{TH = THKLAY(I)}\]

\[\text{DOUT = RIN + THKLAY(I)/4.0}\]

\[\text{BASE = NB}\]

\[\text{DO 1 J=1, NLAY(1)}\]

\[\text{IF (J.GT.1) WRITE (MODU, 18) I, J-1, J, TYPE}\]

\[\text{IF (J.EQ.1) NATEMP = NB}\]

\[\text{IF (J.EQ.2) THEN}\]

\[\text{CRY20980}\]

\[\text{CRY20990}\]

\[\text{CRY21000}\]
IF (IJ.EQ.0.AND.J.EQ.1) WRITE (MODU, 94) I,TYPE
NTBHXX = 0
NBXX = 0
DO 71 K = 1, INDEX
   IF (NDS(K).EQ.NA) THEN
      NTBSTXX = 1
      NAHX = NA
      NBHX = NCND(K)
      GOTO 72
   ENDIF
CONTINUE
    IF (NBHX.NE.0) THEN
      CALL AREACYL (2,IJ, DOUT, THICK(1),ARKA, NWHICH)
      AREA=AREA/2.
      CALL RITCND (5,NCON, I,1,NBHX, I,NARY,0, AREA, 0,MATNMS (I))
      NCON+NCON+1
      NA = NA + 1
      DHALF=DOUT-(THICK (I)/NLAYRS (I))
      CALL AREACYL (2,IJ, DHALF, THICK(I),AREA, NWHICH)
      AREA=AREA/2.
      CALL RITCND (5,NCON, I,1,NBHX, I,NB, I,NARY,0, AREA, 0,MATNMS (I))
      NB = NB + 1
      NCON=NCON+I
    ENDIF
  IF (NBHX.EQ.0) THEN
    CALL AREACYL (2,IJ, DOUT,THICK(1),AREA, NWHICH)
    AREA-(AREA/TH/2. ) *4.
    IF (J.NE.I) THEN
      D2=DOUT+TH/4.
      CALL AREACYL (2, J, D2, THICK (I) , AREA, NWHICH)
      AREA- (AREA/TH/2.) *4.
      NLI-NLGS (NIN) +5000
      CALL RITCND(5,NCON, I,1,NIN, I,NOUT,I,NLI,0,AREA, 0,MATNMS (I))
      NIN-NIN+I
      NOUT-NOUT+I
    ENDIF
  NCON=NCON+I
  NA = NA + 1
  NB = NB + 1
ENDIF
DO 988 J=0,NLAYRS(I)
D2=DOUT+TH/4.
CALL AREACYL (2, J, D2, THICK (I) , AREA, NWHICH)
AREA= (AREA/TH/2.) *4.
   NLI-NLGS (NIN) +5000
   CALL RITCND(5,NCON, I,1,NIN, I,NOUT,I,NLI,0,AREA, 0,MATNMS (I))
   NIN=NIN+1
   NOUT=NOUT+1
   NCON=NCON+1
988 CONTINUE
DOUT=RIN+THKLAY (I)/4.0
92 CONTINUE
WRITE (MODU, 987) I,J,TYPE  
DO 989 IJ=0, NELAY-I  
   NC=NC2+(NELAY*J-1)+IJ  
   IF (IJ,NE,NELAY-I) THEN  
      CALL AREACYL (I, IJ,DOUT,THICK(I),AREA, NWHICH)  
      IF (NWHICH.EQ.1) II=IJ+1  
      IF (NWHICH.EQ.2) II=IJ-1  
      CALL AREACYL (I,II,DOUT,THICK(I),AREA2,NWHICH)  
      NLI=NLGS(NC)+5000  
      NL2=NLGS(NC+1)+5000  
      CALL RITCND (7, NCON, I,II,NC, I,NC+I,II,NLI,NL2)  
      AREA, AREA2, MATNMS(I))  
      NCON = NCON + 1  
   ENDIF  
   IF (IJ.EQ.NELAY-I) THEN  
      IF (NWHICH.EQ.1) THEN  
         CALL AREACYL (I, IJ,DOUT,THICK(I),AREA,NWHICH)  
         NB = (2001*I)+(NELAY*J)-I  
         NBT = (2001*I)+(NELAY*NLAYRS(I))+(NCYLAY* (J-1))  
         IF (NWHICH.EQ.1) II=IJ+1  
         IF (NWHICH.EQ.2) II=IJ-1  
         CALL AREACYL (I,II,DOUT,THICK(I),AREA2,0)  
         NLI=NLGS(NB)+5000  
         NL2=NLGS(NBT)+5000  
         CALL RITCND (7, NCON, I,II,NB, I,NBT,II,NLI,NL2)  
         AREA, AREA2, MATNMS(I) )  
         AREA, AREA2, MATNMS(I))  
         NCON = NCON + 1  
      ENDIF  
      IF (NWHICH.EQ.2) THEN  
         CALL AREACYL (I, II,DOUT,THICK(I),AREA, NWHICH)  
         NBTEMP = (2001*I)+((NSBLAY+NEBLAY+NFBLAY)*NLAYRS(I))+(NCYLAY* J) -I  
         NAT = (2001*I)+((NCYLAY+NSBLAY+NEBLAY+NFBLAY)*NLAYRS(I))+(NELAY* (J-1))  
         IF (NWHICH.EQ.1) II=IJ+1  
         IF (NWHICH.EQ.2) II=IJ-1  
         CALL AREACYL (I,II,DOUT,THICK(I),AREA2,0)  
         NLI=NLGS(NAT)+5000  
         NL2=NLGS(NBTEMP)+5000  
         CALL RITCND (7, NCON, I,II,NAT, I,NBTEMP,II,NLI,NL2)  
         AREA, AREA2, MATNMS(I) )  
         AREA, AREA2, MATNMS(I))  
         NCON = NCON + 1  
      ENDIF  
   ENDIF  
CONTINUE  
DOUT=DOUT+TH/4.  
CONTINUE  
989 CONTINUE  
988 CONTINUE  
18 FORMAT (7X,'REM RADIAL CONDUCTORS REGION ',I2,' LAYER ',I2,TO LAYER ',I2, IN ','A6,' END.')  
88 FORMAT (7X,'REM RADIAL CONDUCTORS REGION ',I2,' LAYER ',I2,TO BOUNDARY NODES IN ','A6,' END.')  
94 FORMAT (7X,'REM RADIAL CONDUCTORS REGION ',I2,' BOUNDARY NODES IN ','A6,' END.')  
123 FORMAT (7X,'REM RADIAL CONDUCTORS REGION ',I2,' LAYER ',I2,TO BOUNDARY NODES IN ','A6,' END.')  
987 FORMAT (7X,'REM CIRCUMFERENTIAL CONDUCTORS REGION ',I2,' LAYER ',I2,TO BOUNDARY NODES IN ','A6,' END.')  
   RETURN  
END  
SUBROUTINE AREACYL (NAREA, IJPOS, R, TH, AREA, NW)  
   COMMON REGION/ NTHETA, NBETAS, BETA, RIN, TVOL,  
   ROUT(9), REGHS(9), NLAYRS(9), TEMPS(9), THICK(9),  
   THLAY(9), MATRLS(9), MATNMS(9), RCNMS(9)  
   COMMON TOPBOT/ NTOP, NBOT, NFTLAY, NSLAY, NFTLAY, NBLAY, NSBLAY,  
   NNETLAY, NBLAY, ETBLAY, FTTLAY, FBTLAY  
   COMMON CYDATA/CYLHGT, NCYLAY  
RETURN
SUBROUTINE TO COMPUTE AREAS IN A CYLINDER, SPHERE, ELLIPSE, OR FLAT

C AREAS FOR NODES TO COMPUTE NODAL VOLUMES. OUTSIDE AREAS FOR SOURCE
C TERMS (IF ANY) AREAS FOR CONDUCTOR PATHS.

C NAREA: 1,2 RADIAL AREA (IN & OUT), CIRCUMFERENTIAL AREA (UP & DOWN)
C JPOS: POSITION OF THETA ANGLE COUNTING FROM THE SOUTH POLE.
C R: RADIUS TO AREA SURFACE
C TH: LAYER THICKNESS
C AREA: VALUE RETURNED TO CALLING SUBROUTINES.

C NWHICH=0 THE LAYER IS IN THE CYLINDRICAL SHAPE.
C NWHICH=1 THE LAYER IS IN THE BOTTOM SHAPE.
C NWHICH=2 THE LAYER IS IN THE TOP SHAPE.

JPOS=JPOS
NWHICH=NW
IF (NWHICH .EQ. -1 .OR. NWHICH .EQ. -2) THEN
  IF (NWHICH .EQ. -1) R=RIN
  IF (NWHICH .EQ. -2) R=RIN+THICK(1)+THICK(2)+THICK(3)
  NBL=NSBLAY+NEBLAY+NFBLAY
  IF (JPOS.GE.1.AND.JPOS.LE.NBL) NWHICH=0
  IF (JPOS.GT.NBL.AND.JPOS.LE.NBL+NCYLAY) NWHICH=2
  IF (JPOS.GT.NBL+NCYLAY) NWHICH=2
ENDIF

C THE NEXT SECTION IS FOR THE NODE AREAS OF NODES IN THE CYLINDER PART.

IF (NWHICH .EQ. 0) THEN
  IF (NAREA .EQ. 1)
    AREA=BETA*R*TH
  ELSE
    AREA=((R+(R+TH))/2.)*TH/(CYLHGT/NCYLAY)
  ENDIF
ENDIF

C THE NEXT SECTION IS FOR THE NODE AREAS IN EITHER THE FLAT TOP
C OR FLAT BOTTOM.

IF ((NWHICH .EQ. 1 .AND. NBOT .EQ. 2) .OR.
  (NWHICH .EQ. 2 .AND. NTOP .EQ. 2)) THEN
  IF (NWHICH .EQ. 1) THEN
    NFLAY=NFBLAY
    FTHK=FTHK
    ELSE
      NFLAY=NFLAY
      FTHK=FTHK
  ENDIF
  IF (NAREA .EQ. 1)
    AREA=((R+(R+TH))/2.)*(FTHK/NFLAY)/(TH/2.)
  ELSE
    AREA=((R+(R+TH))/2.)*TH/(FTHK/NFLAY)
  ENDIF
ENDIF

C THIS SECTION CALCULATES THE NODE AREA FOR A NODE IN EITHER THE TOP
C OR BOTTOM SPHERE.

IF ((NWHICH .EQ. 1 .AND. NBOT .EQ. 3) .OR.
  (NWHICH .EQ. 2 .AND. NTOP .EQ. 3)) THEN
  IF (NWHICH .EQ. 1) THEN
    NSLAY=NSBLAY
  ELSE
    NSLAY=NSLAY
  ENDIF
ENDIF

C ENDS.
THETA0=PI/2.
POS/JPOS
ELSE
NSLAY=NSTLAY
THETA0=0.0
POS=1*(JPOS+1)
ENDIF
DTHETA=PI/2./NSLAY

THETA1=THETA0-POS*DTHETA
THETA2=THETA1-DTHETA
IF (NAREA.EQ.1) THEN
AREA=THETA0*R*R*(COS(THETA0)+COS(THETA1))
ELSE
AREA=THETA0*R*R*(COS(THETA1)+COS(THETA2))
ENDIF

IF (NAREA.EQ.2) THEN
AREA=THETA0*R*R*(COS(THETA0)+COS(THETA1))
ELSE
AREA=THETA0*R*R*(COS(THETA1)+COS(THETA2))
ENDIF

C THE NEXT SECTION IS FOR THE NODE AREAS IN EITHER THE ELIPTICAL TOP
C OR ELIPTICAL BOTTOM.

IF ((NWHICH.EQ.1.AND.NBOT.EQ.4) .OR. (NWHICH.EQ.2.AND.NTOY.EQ.4)) THEN
NE=NEBLAY
ERAT=EBRAT
THETA0=PI/2.
POS=JPOS
ELSE
NE=NETLAY
ERAT=ETRAT
THETA0=0.0
POS=1*(JPOS+1)
ENDIF
DTHETA=PI/2./NE

JP=JPOS-1
AI=R-TH
AO=AI+TH
BI=AI*ERAT
BO=BI+TH

THETA2=THETA0-POS*DTHETA
THETA1=THETA2-DTHETA
AAVG=(AO+AI)/2.
BAVG=(BO+BI)/2.
THAVG=(THETA2+THETA1)/2.
COSAVG=COS(THAVG)
SINAVG=SIN(THAVG)
FRST=((BETA*COSAVG)/2.+((AAVG*BAVG)/SQRT((BAVG*BAVG*COSAVG*COSAVG)*(AAVG*AAVG*SINAVG*SINAVG)))
SND=AO*BO*(ATAN((AO/BO)*TAN(THETA1))-ATAN((AO/BO)*TAN(THETA2)))
THR=AI*BI*(ATAN((AI/BI)*TAN(THETA2))-ATAN((AI/BI)*TAN(THETA1)))
IF (NAREA.EQ.2) THEN
RETURN
ENDIF

SUBROUTINE ULLIG

COMMON/UILLAGE/ NLUL4,NLUL5,NTHU41,RINMHH, PCTFUL, RADULG, TVULFT,
CT, LG(3), LIQVAP(3)
COMMON/REGION/NTHETA, NBETAS, BETA, RIN, TVOL,
ROUT(9), REGNS(9), NLAYRS(9), TEMPS(9), THICK(9),
THKLAY(9), MATRLS(9), MATNMS(9), RGNNMS(9)
COMMON/TOP/NBOT, NDOT, NDSLAY, NNSLAY, NFBLAY, NSBLAY,
NEBLAY, ETRAT, EBRAT, FTTHK, FBTTHK

CRY23110
CRY23120
CRY23130
CRY23140
CRY23150
CRY23160
CRY23170
CRY23180
CRY23190
CRY23200
CRY23210
CRY23220
CRY23230
CRY23240
CRY23250
CRY23260
CRY23270
CRY23280
CRY23290
CRY23300
CRY23310
CRY23320
CRY23330
CRY23340
CRY23350
CRY23360
CRY23370
CRY23380
CRY23390
CRY23400
CRY23410
CRY23420
CRY23430
CRY23440
CRY23450
CRY23460
CRY23470
CRY23480
CRY23490
CRY23500
CRY23510
CRY23520
CRY23530
CRY23540
CRY23550
CRY23560
CRY23570
CRY23580
CRY23590
CRY23600
CRY23610
CRY23620
CRY23630
CRY23640
CRY23650
CRY23660
CRY23670
CRY23680
CRY23690
CRY23700
CRY23710
CRY23720
CRY23730
CRY23740
CRY23750
CRY23760
CRY23770
CRY23780
CRY23790
CRY23800
COMMON/CYDATA/CYLHT,NCYLAY
COMMON/HTXGRS/NHX(10),NLHX(10),NTHHX(10),
* LENHX(10)
COMMON/UNITS/MODU,SINDA
COMMON/NODDAT/NODNUM(10000),VOL(IO000),NLGS(10000)
COMMON/VOLUME/VOLLIQ,ACCLIQ

CHARACTER*1 CT, LG
CHARACTER*6 LIQVAP
LOGICAL REGNS

PI=3.141592654
IF (.NOT.REGNS(4)) THEN
  IF (.NOT.REGNS(5)) GOTO 99
ENDIF
NTHU41=0
DO 77 I=8001,10000
  TOTVOL=TOTVOL+VOL(I)
CONTINUE
VOLLIQ=TOTVOL*(PCTFUL/100.)
IF (NBOT.EQ.3.OR.NBOT.EQ.4) THEN
  IF (NBOT.EQ.3) LAYBOT=NSBLAY
  IF (NBOT.EQ.4) LAYBOT=NEBLAY
  NUMNOD=8001
  DO 1 I=1,NLAYRS(4)+NLAYRS(5)
  IF (I.LE.NLAYRS(4)) NSTART=8001
  IF (I.GT.NLAYRS(4)) NSTART=9001
  IF (I.LE.NLAYRS(4)) K=J
  IF (I.EQ.NLAYRS(4)+I) K=J-1
  IF (I.GT.NLAYRS(4)+I) K=J+1
  NUMNOD=NSTART+((K-J)*LAYBOT)
  ACCLIQ=ACCLIQ+VOL(NUMNOD)
  IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD).GT.1100.AND.
     NLGS(NUMNOD).LE.1199) NLGS(NUMNOD)=NLGS(NUMNOD)+200
  IF (NTHU41.EQ.0.AND.NLGS(NUMNOD).GT.1300) NTHU41=1
  CONTINUE
  IF (I.LE.LAYBOT) THEN
    DO 3 J=1,I
    IF (J.LE.NLAYRS(4)) NSTART=8001
    IF (J.GT.NLAYRS(4)) NSTART=9001
    IF (J.LE.NLAYRS(4)) K=J
    IF (J.GT.NLAYRS(4)+I) K=J-1
    IF (J.GT.NLAYRS(4)+I) K=J+1
    NUMNOD=NSTART+((K-J)*LAYBOT)+I-1
    ACCLIQ=ACCLIQ+VOL(NUMNOD)
    IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD).GT.1100.AND.
     NLGS(NUMNOD).LE.1199) NLGS(NUMNOD)=NLGS(NUMNOD)+200
    IF (NTHU41.EQ.0.AND.NLGS(NUMNOD).GT.1300) NTHU41=1
    CONTINUE
    IF (J.LT.LAYBOT) THEN
      DO 2 J=1,LAYBOT
      IF (J.LE.NLAYRS(4)) NSTART=8001
      IF (J.GT.NLAYRS(4)) NSTART=9001
      IF (J.LE.NLAYRS(4)) K=J
      IF (J.GT.NLAYRS(4)+I) K=J-1
      IF (J.GT.NLAYRS(4)+I) K=J+1
      NUMNOD=NSTART+((K-J)*LAYBOT)+I-1
      ACCLIQ=ACCLIQ+VOL(NUMNOD)
      IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD).GT.1100.AND.
       NLGS(NUMNOD).LE.1199) NLGS(NUMNOD)=NLGS(NUMNOD)+200
      IF (NTHU41.EQ.0.AND.NLGS(NUMNOD).GT.1300) NTHU41=1
      CONTINUE
  ENDIF
  IF (NBOT.EQ.2) THEN
    DO 4 I=1,NLAYRS(4)
    2 CONTINUE
    IF (J.LE.LAYBOT) THEN
      DO 3 J=1,1
      IF (J.LE.NLAYRS(4)) NSTART=8001
      IF (J.GT.NLAYRS(4)) NSTART=9001
      IF (J.LE.NLAYRS(4)) K=J
      IF (J.GT.NLAYRS(4)+I) K=J-1
      IF (J.GT.NLAYRS(4)+I) K=J+1
      NUMNOD=NSTART+((K-J)*LAYBOT)+I-1
      ACCLIQ=ACCLIQ+VOL(NUMNOD)
      IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD).GT.1100.AND.
       NLGS(NUMNOD).LE.1199) NLGS(NUMNOD)=NLGS(NUMNOD)+200
      IF (NTHU41.EQ.0.AND.NLGS(NUMNOD).GT.1300) NTHU41=1
      CONTINUE
  ENDIF
  IF (NBOT.EQ.2) THEN
    DO 4 I=1,NLAYRS(4)
  ENDIF
ENDIF
CONTINUE
NUMNOD=NUMNOD+1
ACCLIQ=ACCLIQ+VOL(NUMNOD)
IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD).GT.1100.AND.
 * NLGS(NUMNOD).LE.1199) NLGS(NUMNOD)=NLGS(NUMNOD)+200
IF (NTHU41.EQ.0.AND.NLGS(NUMNOD).GT.1300) NTHU41=1
ENDIF
CONTINUE
NUMNOD=NUMNOD+1
ACCLIQ=ACCLIQ+VOL(NUMNOD)
IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD).GT.1100.AND.
 * NLGS(NUMNOD).LE.1199) NLGS(NUMNOD)=NLGS(NUMNOD)+200
IF (NTHU41.EQ.0.AND.NLGS(NUMNOD).GT.1300) NTHU41=1
ENDIF
CONTINUE
PARAMETER (NSTAT5=9001*, NSTAT4=8001*, NSTAT3=7001*, NSTAT2=6001*, NSTAT1=5001*, NSTAT0=4001*, NSTAT-4001*, NSTAT-5001*, NSTAT-6001*, NSTAT-7001*, NSTAT-9001*)

NSTAT5*9001*(I-1)
NSTAT4*8001*(I-1)

DO 5 J=NLAYRS(5),1,-1
  NNUMMOD=NSTAT5* ((J-1)*NFBLAY)
  ACCLIQ=ACCLIQ+VOL (NNUMMOD)
  IF (ACCLIQ.GT.VOLLIQ.AND.NLGS (NNUMMOD).GT.1100.AND.
   NLGS (NNUMMOD).LE.1199) NLGS (NNUMMOD) = NLGS (NNUMMOD) +200
  IF (NTU+1.EQ.0.AND.NLGS (NNUMMOD).GT.1300) NTU+1=1
  CONTINUE

5 CONTINUE

DO 6 J=NLAYRS(4),1,-1
  NNUMMOD=NSTAT4* ((J-1)*NFBLAY)
  ACCLIQ=ACCLIQ+VOL (NNUMMOD)
  IF (ACCLIQ.GT.VOLLIQ.AND.NLGS (NNUMMOD).GT.1100.AND.
   NLGS (NNUMMOD).LE.1199) NLGS (NNUMMOD) = NLGS (NNUMMOD) +200
  IF (NTU+1.EQ.0.AND.NLGS (NNUMMOD).GT.1300) NTU+1=1
  CONTINUE

6 CONTINUE

ENDIV
VOLIQ=TOTVOL*(PCTFUL/100.)
GASVOL=TOTVOL-VOLIQ
NSTA=9001+((NSBLAY+NFBLAY+NEBLAY)*NLAYRS(5))
NSTB=8001+((NSBLAY+NFBLAY+NEBLAY)*NLAYRS(4))
DO 7 I=1,NCYLAY
  NSTATS=NSTA+(I-1)
  NSTATB=NSTB+(I-1)
  DO 9 J=NLAYS(5),1,-1
      NUMNOD=NSTATS+((J-1)*NCYLAY)
      ACCLIQ=ACCLIQ+VOL(NUMNOD)
      IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD).GT.1100.AND.
          NLGS(NUMNOD).LE.1199) NLGS(NUMNOD)=NLGS(NUMNOD)+200
      IF (NTHU41.EQ.0.AND.NLGS(NUMNOD).GT.1300)
          NTHU41=1+NFBLAY+NSBLAY+NEBLAY
  9 CONTINUE
  CONTINUE
  DO 8 J=NLAYS(4),1,-1
      NUMNOD=NSTATB+((J-1)*NCYLAY)
      ACCLIQ=ACCLIQ+VOL(NUMNOD)
      IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD).GT.1100.AND.
          NLGS(NUMNOD).LE.1199) NLGS(NUMNOD)=NLGS(NUMNOD)+200
      IF (NTHU41.EQ.0.AND.NLGS(NUMNOD).GT.1300)
          NTHU41=1+NFBLAY+NSBLAY+NEBLAY
  8 CONTINUE
  END IF (NBOT.EQ.3.OR.NBOT.EQ.4)
    IF (NBOT.EQ.3)
       LAYBOT=NSBLAY
    IF (NBOT.EQ.4) LAYBOT=NEBLAY
    NUMNOD=8001
    DO I=1,NLAYS(4)+NLAYS(5)
      IF (I.LE.NLAYS(4)) KSTART=8001
      IF (I.GT.NLAYS(4)) KSTART=9001
      IF (I.LE.NLAYS(4)) IK=I
      IF (I.EQ.NLAYS(4)+1) IK=IK+1
      NUMNOD=KSTART+((IK-1)*LAYBOT)
      ACCLIQ=ACCLIQ+VOL(NUMNOD)
      IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD).GT.1100.AND.
          NLGS(NUMNOD).LE.1199) NLGS(NUMNOD)=NLGS(NUMNOD)+200
      IF (NTHU41.EQ.0.AND.NLGS(NUMNOD).GT.1300) NTHU41=IK
      CONTINUE
    DO 2 J=1,13
      NUMNOD=NUMNOD+I
      ACCLIQ=ACCLIQ+VOL(NUMNOD)
      IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD).GT.1100.AND.
          NLGS(NUMNOD).LE.1199) NLGS(NUMNOD)=NLGS(NUMNOD)+200
      IF (NTHU41.EQ.0.AND.NLGS(NUMNOD).GT.1300) NTHU41=IK
  2 CONTINUE
  IF (.NOT.LAYBOT) THEN
    DO 3 J=1,1
      IF (J.LE.NLAYS(4)) KSTART=8001
      IF (J.GT.NLAYS(4)) KSTART=9001
      IF (J.LE.NLAYS(4)) K=J
      IF (J.EQ.NLAYS(4)+1) K=K+1
      NUMNOD=KSTART+((K-1)*LAYBOT)+1-1
      ACCLIQ=ACCLIQ+VOL(NUMNOD)
      IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD).GT.1100.AND.
          NLGS(NUMNOD).LE.1199) NLGS(NUMNOD)=NLGS(NUMNOD)+200
      IF (NTHU41.EQ.0.AND.NLGS(NUMNOD).GT.1300) NTHU41=IK
  3 CONTINUE
ENDIF
END
* NLGS (NUMNOD), LE.1199) NLGS (NUMNOD) =NLGS (NUMNOD) +200
IF (NTHU41.EQ.0.AND.NLGS(NUMNOD) .GT.1300) NTHU41 =I
ENDIF
IF (NBOT.EQ.2) THEN
  DO 4 I=1,NFBLAY
    NSTAT5=NSTAT+((I-1))
    NSTAT4=NSTAT+(I-1)
  DO 5 J=NLAYRS(5),1,-1
    NUMNOD=NSTAT5*((J-I)*NFBLAY)
    ACCLIQ=ACCLIQ+VOL(NUMNOD)
    IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD) .GT.1100.AND.
      NLGS(NUMNOD).LE.1199) NLGS(NUMNOD) =NLGS (NUMNOD) +200
    IF (NTHU41.EQ.0.AND.NLGS(NUMNOD) .GT.1300) NTHU41 =I
  CONTINUE
  DO 6 J=NLAYRS(4),1,-1
    NUMNOD=NSTAT4*((J-I)*NFBLAY)
    ACCLIQ=ACCLIQ+VOL(NUMNOD)
    IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD) .GT.1100.AND.
      NLGS(NUMNOD).LE.1199) NLGS(NUMNOD) =NLGS (NUMNOD) +200
    IF (NTHU41.EQ.0.AND.NLGS(NUMNOD) .GT.1300) NTHU41 =I
  CONTINUE
ENDIF
IF (NTOP.EQ.2) THEN
  NSTA =NSTA+ ((NSBLAY*NFBLAY+EBLAY*NCYLAY)*NLAYRS(5))
  NSTB =NSTB+ ((NSBLAY*NFBLAY+EBLAY*NCYLAY)*NLAYRS(4))
  DO 14 I=1,LAYTOP
    NSTATS=NSTA+(I-1)
    NSTAT4=NSTB+(I-1)
  DO 15 J=NLAYRS(5),1,-1
    NUMNOD=NSTAT5*((J-I)*LAYTOP)
    ACCLIQ=ACCLIQ+VOL(NUMNOD)
    IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD) .GT.1100.AND.
      NLGS(NUMNOD).LE.1199) NLGS(NUMNOD) =NLGS (NUMNOD) +200
    IF (NTHU41.EQ.0.AND.NLGS(NUMNOD) .GT.1300) NTHU41 =I
    NTHU41=I+NFBLAY+NSBLAY+EBLAY*NCYLAY
  CONTINUE
  DO 16 J=NLAYRS(4),1,-1
    NUMNOD=NSTAT4*((J-I)*LAYTOP)
    ACCLIQ=ACCLIQ+VOL(NUMNOD)
    IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD) .GT.1100.AND.
      NLGS(NUMNOD).LE.1199) NLGS(NUMNOD) =NLGS (NUMNOD) +200
    IF (NTHU41.EQ.0.AND.NLGS(NUMNOD) .GT.1300) NTHU41 =I
    NTHU41=I+NFBLAY+NSBLAY+EBLAY*NCYLAY
  CONTINUE
ENDIF
IF (NTOP.EQ.3.OR.NTOP.EQ.4) THEN
  NSTA =NSTA+ ((NSBLAY*NFBLAY+EBLAY*NCYLAY)*NLAYRS(5))
  NSTB =NSTB+ ((NSBLAY*NFBLAY+EBLAY*NCYLAY)*NLAYRS(4))
  IF (NTOP.EQ.3) LAYTOP=NSTAT5
  IF (NTOP.EQ.4) LAYTOP=NSTAT4
  DO 24 I=1,LAYTOP
    NSTATS=NSTA+(I-1)
    NSTAT4=NSTB+(I-1)
  DO 25 J=NLAYRS(5),1,-1
    NUMNOD=NSTAT5*((J-I)*LAYTOP)
    ACCLIQ=ACCLIQ+VOL(NUMNOD)
    IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD) .GT.1100.AND.
      NLGS(NUMNOD).LE.1199) NLGS(NUMNOD) =NLGS (NUMNOD) +200
    IF (NTHU41.EQ.0.AND.NLGS(NUMNOD) .GT.1300) NTHU41 =I
    NTHU41=I+NFBLAY+NSBLAY+EBLAY*NCYLAY
  CONTINUE
  DO 26 J=NLAYRS(4),1,-1
    NUMNOD=NSTAT4*((J-I)*LAYTOP)
    ACCLIQ=ACCLIQ+VOL(NUMNOD)
    IF (ACCLIQ.GT.VOLLIQ.AND.NLGS(NUMNOD) .GT.1100.AND.
      NLGS(NUMNOD).LE.1199) NLGS(NUMNOD) =NLGS (NUMNOD) +200
    IF (NTHU41.EQ.0.AND.NLGS(NUMNOD) .GT.1300) NTHU41 =I
    NTHU41=I+NFBLAY+NSBLAY+EBLAY*NCYLAY
CONTINUE
ENDIF
GOTO 199

ENDIF
GOTO 199

99 IF (NBOT.EQ.2) BOTVOL=(PI*RIN*RIN*FRTHK)/2.
   IF (NBOT.EQ.3) BOTVOL=((4./3.)*PI*RIN*RIN*RIN)/4.
   IF (NBOT.EQ.4) BOTVOL=((4./3.)*PI*RIN*RIN*EBRAT)/4.
   IF (NBOT.NE.1) BOTLAY=(BOTVOL/(NSBLAY+NEBLAY+NFBLAY))
   CYLVOL=(PI*RIN*RIN*CYLHGT)/2.
   CYLLAY=((PI*RIN*RIN*CYLHGT)/NCYLAY)/2.
   IF (NTOP.EQ.2) TOPVOL=(PI*RIN*RIN*FTTHK)/2.
   IF (NTOP.EQ.3) TOPVOL=((4./3.)*PI*RIN*RIN*RIN)/4.
   IF (NTOP.EQ.4) TOPVOL=((4./3.)*PI*RIN*RIN*ETRAT)/4.
   IF (NTOP.NE.1) TOPLAY=(TOPVOL/(NFTLAY+NSTLAY+NETLAY))
   VOLUM=BOTVOL+CYLVOL+TOPVOL
   VOLLIQ=VOLUM*(PCTFUL/100.)
   ACCLIQ=0
   DO 33 I=1,NFBLAY+NSBLAY+NEBLAY
       ACCLIQ=ACCLIQ+BOTLAY
       IF (ACCLIQ.GT.VOLLIQ.AND.NTHU41.EQ.0) NTHU41=I
   33 CONTINUE
   IP=NFBLAY+NSBLAY+NEBLAY
   ACCLIQ=ACCLIQ+CYLHGT
   IF (ACCLIQ.GT.VOLLIQ.AND.NTHU41.EQ.0) NTHU41=I
   34 CONTINUE
   IP=NFBLAY+NSBLAY+NEBLAY+NCYLAY
   ACCLIQ=ACCLIQ+TOPLAY
   IF (ACCLIQ.GT.VOLLIQ.AND.NTHU41.EQ.0) NTHU41=I
   CONTINUE
RETURN
END
APPENDIX E

CryoTran Program Listings

Part IV CRYOPLOT FORTRAN
SUBROUTINE PLTSPH
CALLED FROM
SUBROUTINE TO PLOT THE GEOMETRY
NTYP = 1 SPHERE WEDGE MODEL
USING ISSCO DISPLA

COMMON /TITL/ PTITLE
COMMON /REGION/ NTETA, NLATRS, BETA, RIN, TVOL, ROUT(9),
1 REGNS(9), NLAYRS(9), THICK(9),
2 THKLAY(9), MATTLS(9), MATNNMS(9), AGNMS(9)
COMMON /ULLAGE/ NUL4, NUL5, NTHU41, RINMHH, PCTFUL, RADULG, TVULFT,
1 CT, LG(3), LIQVAP(3)
COMMON /HTXGRS/ NHX, HXTEMP(10), NRHX(10), NLHX(10),
1 NTHHX(I0) , LNGTHX(I0)
COMMON /STUFF/ NHTT, PI, CONVY, CONVR, THETA0, DTHETA, NBASOS, ROUTSF,
1 BNCOEF(2)
COMMON /OUTSRC/ NQIN, QEFF
LOGICAL REGNS
LOGICAL NOSCAL
CHARACTER*1 CT, LG
CHARACTER*6 LIQVAP
CHARACTER*1 REGNO
CHARACTER*16 MATNMS
CHARACTER*25 RGNNMS
CHARACTER*80 REGLAB
CHARACTER*10 RADIUS
CHARACTER*2 NUM
CHARACTER*6 RLABLE, ULLG
CHARACTER*8 REGLAB
CHARACTER*10 RADIUS
CHARACTER*17 NOSMSG
CHARACTER*20 QMSG1, QMSG2, QMSG4, HXMSG1
CHARACTER*4 QMSG3, TEQMSG
CHARACTER*6 DEQMSG
CHARACTER*10 NQMSG
CHARACTER*20 BMMSG1, BMMSG5
CHARACTER*20 CONRAD(2)
CHARACTER*5 TANKEQ
CHARACTER*6 TVUNIT
CHARACTER*7 PCTSI
CHARACTER*12 TNKVEQ

DIMENSION PROUT(5), PTHICK(5)
DIMENSION R(500), THETA(500)

DATA REGLAB/"REGION "/, RADIUS/"RADIUS(IN)"/
DATA ULLG/"ULLG"/
DATA NGMSG/"PLOT NOT TO SCALE"/
DATA DEQMSG/"-5"/
DATA NQMSG/"NOE NO. 5"/
DATA QMSG1/"SOURCE Q INTO ALL$"/
DATA QMSG2/"OUTER SURFACE NODES$"/
DATA QMSG2/"Q=$"/
DATA QMSG4/"(BTU/(HR-IN2))$"/
DATA BMMSG1/"OUTSIDE BNDY NODE $"/
DATA BMMSG5/"TO SURFACES*$"/
DATA CONRAD(1)/"CONVECTIONS"/
DATA CONRAD(2)/"RADIATIONS"/
DATA HXMSG1/"HEAT EXCHANGER NO. $"/
DATA TNKVEQ/"TANK VOLUME$"/
DATA TVUNIT '/ (FT3)'/
DATA TANK1Q '/ TANK'/
DATA PCTSIN '/ % FULL'/

C INITIALIZE OUTPUT TO (1) QMS PRINTER OR TO (2) TERMINAL SCREEN

PRINT 2001

2001 FORMAT(///'IN THE SPHERE PLOTTING ROUTINE'/
1 ' SEND THE GRAPH TO '/*
2 ' 1. THE QMS PRINTER'/
3 ' 2. THE TERMINAL SCREEN'/*
4 ' 3. SOME OTHER DEVICE '/
5 ' TYPE IN 1 2 OR 3')

CALL READIN(IDV, I,3)

IF (IDV .EQ. 1) CALL QMS2
IF (IDV .EQ. 2) CALL IBM52(3179,0,0,0)

C NEW SUBROUTINE CALL FROM DAVE HUBLER TO ALLOW USER TO CHOOSE DEVICE

IF (IDV .EQ. 3) CALL PDEV(' ',ISTAT)

C

C SET PAGE SIZE

CALL PAGE(11., 8.5)

C SET SUBPLOT SIZE

CALL AREA2D(10.5, 8.0)

C SET CHARACTER STYLE TO TRIFLEX

CALL TRIPLX

C SCALE REGION THICKNESSES UP IF REAL SCALE IS TOO SMALL

NOSCAL=.FALSE.
SCALEF=3IN/20.

DO 5 I=1,5
PROUT(I)=PROUT(I)
PTHICK(I)=THICK(I)

IF (REGNS(I)) THEN

IF (PTHICK(I) .LT. 1.) THEN

PTHICK(I)=THICK(I)+SCALEF
NOSCAL=.TRUE.

ENDIF
ENDIF

5 CONTINUE

IF (NOSCAL) THEN
THKSUM=0.0

DO 7 I=1,3

THKSUM=THKSUM+PTHICK(I)
PROUT(I)=RIN+THKSUM

ENDIF

7 CONTINUE

IF (PTHICK(4) .LT. 1.) THEN

PROUT(5)=RIN+PTHICK(4)

PTHICK(5)=PROUT(5)

ENDIF

ENDIF

C

RADMAX= AMAX1 (PROUT(1), PROUT(2), PROUT(3))

C

RSTEP=RADMAX/4.5
PRINT *, 'PLOT -- RADMAX, RSTEP=', RADMAX, RSTEP

CALL POLAR(3.14159/180., RSTEP, 5.5, 0.7)

C

WRITE THE PLOT TITLE

C

CALL ANGLE(90.)
CALL MESSAG(PTITLE(1:40), 40, 0, 0, 0)
CALL MESSAG(PTITLE(41:), 40, 0, 3, 0, 0)

C

THETA(1)= 3.
DO 8 I=2,187

THETA(I)=THETA(I-1)+1.

8 CONTINUE

C

CALL ALNMES(0.0, 0.5)
CALL ANGLE(90)
CALL HEIGHT(9.1)
DO 9 IJ=1,5
IF(REN5(IJ))THEN
NL=MLAYR5(IJ)
TNL=NL
DELHK+THICK(IJ)/TNL
NL=NL
IF(IJ.EQ.1)NL=NL+1
DO 15 IJL=1,NLL
R(IJL)=PROUT(IJL)*(IJL-I)*DELTHK
IF(IJL.EQ.1.AND.IJL.GT.NL)R(I)-RIN
DO 10 I=2,187
R(I)-R(1)
10 CONTINUE
NPTS=181
NTHP=4
IF(IJL.EQ.1.OR.IJL.GT.NL)THEN
NPTS=187
NTHP=1
CALL THKCRV(0.03)
ENDIF
CALL CURVE(THETA(NTHP),R,NPTS,0)
CALL RESET("THKCRV")
15 CONTINUE
C WRITE THE OUTER RADIUS OF THIS REGION IN THE MARGIN, TOP OF PLOT
XPOS1=XPOS(180.0,PROUT(IJ))
CALL REALNO(ROUT(IJ),2,XPOS1,-0.35)
C WRITE THE REGION NUMBER IN THE MARGIN, BOTTOM OF CIRCLE
XPOS2=XPOS(0.0,PROUT(IJ))
CALL MESSAG(RETLAB,8,XPOS2,-0.35)
CALL INTNO(IJ,'ABUT',"ABUT")
C IF REN5(4)=FALSE; EXTRA CALL TO WRITE INSIDE RADIUS, RIN
IF(IJ.EQ.1.AND..NOT.REN5(4))THEN
XPOS1=XPOS(180.0,RIN)
CALL REALNO(RIN,2,XPOS1,-0.35)
ENDIF
9 CONTINUE
C CALL RESET(’ALNMES’)
C WRITE LABEL ’RADIUS’ ABOVE RADIUS VALUES IN MARGIN
XPOS= XPOSN(180.0,RADMAX)
CALL MESSAG(RADIUS,10,XPOS,-0.2,-0.5)
CALL RESET(’HEIGHT’)
C IF SOME THICK CHANGED, WRITE NOSCALE MESSAGE
IF(NOSCAL) CALL MESSAG(NOSMSG,17,9,95,5.5)
C NOW PLOT THE RADII; NO. OF RADII = NTHETA+1
NPTS=2
R(1)=0.
R(2)=RADMAX
THETA(1)=0.
THETA(2)=0.
CALL CURVE(THETA,R,NPTS,0)
THETA=-0.
DELTH=180./NTHETA
IF(.NOT.REN5(4))R(I)-RIN
DO 25 IA=2,NTHETA
THETA(1)+THETA+(IA-1)*DELTH
THETA(2)+THETA(1)
CALL CURVE(THETA,R,NPTS,0)
25 CONTINUE
R(I)=0.
THETA(1)=180.
THETA(2)+THETA(1)
CALL CURVE(THETA,R,NPTS,0)
C PLOT LINES OR A SEMICIRCLE DENOTING THE ULLAGE
C IF(CY.EQ.‘C’) THEN
R(1)=RADULG
DO 111 I=2,187
R(I)-R(1)
111 CONTINUE
C
NPTS=181
NTHP=4
CALL MARKER(16)
CALL THKCRV(.05)
CALL CURVE(THETA(NTHP),R,NPTS,9)
CALL RESET('THKCRV')
XPOS=XPOS(178.0,0.4)
YPOS=YPOS(178.0,0.4)
CALL RIMESS(ULIG,6,XPOS,YPOS)
CALL HEIGHT(0.09)
CALL MESSAG(TNKEQ,12,XPOS,-0.5)
CALL REALNO(TVOL,1,XPOS+.15,-0.5)
CALL MESSAG(TVUNIT,6,'ABUT','ABUT')
CALL MESSAG(TANKEQ,5,XPOS+.35,-0.5)
CALL REALNO(PCTFUL,0,XPOS+.50,-0.5)
CALL MESSAG(PCTSIN,7,'ABUT','ABUT')
ENDIF

C

IF(CT .EQ. 'I') THEN
   R(1)=RINMHH
   R(2)=RIN
   ANGR=(PI/2.-RADULG)
C CONVERT ANGLE TO DEGREES
   ANGD=ANGR*180./PI
   THETA(1)=180.
   THETA(2)=ANGD+90.
   IF(PCTFUL .LE. .50.) THEN
      THETA(1)=0.0
      THETA(2)=RADULG*180./PI
      R(1)=-RINMHH
   ENDIF
   NPTS=2
   CALL MARKER(16)
   CALL THKCRV(.05)
C HORIZONTAL LINE
   CALL CURVE(THETA,R,NPTS,1)
C VERTICAL LINE
   THETA(2)=180.0
   CALL MARKER(16)
   CALL CURVE(THETA,R,NPTS,1)
C CURVE ON REGION 4-1 BOUNDARY
   IF(PCTFUL .LE. .50.) THEN
      THETA(1)=RADULG*180./PI
   ELSE
      THETA(1)=ANGD+90.
   ENDIF
   THTEND=180.
   R(1)=RIN
   DO 205 I=2,181
   NPTS=1
   THETA(1)=THETA(I-1)+1.0
   R(I)=R(I-1)
   IF(THETA(I) .GE. THTEND) GO TO 210
205 CONTINUE
210 THETA(NPTS)=THTEND
   CALL CURVE(THETA,R,NPTS,0)
   CALL RESET('THKCRV')
   CALL RIMESS(ULIG,6,180.0,RINMHH+0.2)
   CALL HEIGHT(0.09)
   XPOS=XPOS(180.0,RINMHH)
   CALL MESSAG(TNKEQ,12,XPOS,-0.5)
   CALL REALNO(TVOL,1,XPOS+.15,-0.5)
   CALL MESSAG(TVUNIT,6,'ABUT','ABUT')
   CALL MESSAG(TANKEQ,5,XPOS+.35,-0.5)
   CALL REALNO(PCTFUL,0,XPOS+.50,-0.5)
   CALL MESSAG(PCTSIN,7,'ABUT','ABUT')
ENDIF

C HEAT EXCHANGERS

C

300 IF(NHX.GT.0) THEN

267
CALL THKCRV(.05)
DTHDEG=DTHETA*180./PI
DO 310 I=1,NHX
THETA(1)= (NTHHX(I)-1)*DTHDEG
THETA(I)=THETA(I)+NTHHX(I)*DTHDEG
NRG=NMXH(I)
RSUB=MXH(I)-1
TNL=NLAYRS(NRG)
DELTHK=PSTICK(NRG)/TNL
R(1)=PROUT(NRG)-RSUB*DELTHK
DO 315 J=2,181
NPTS=J
THETA(J)=THETA(J-1)+1.0
R(J)=R(1)
IF(THETA(J) .GE. THTEND) GO TO 320
315 CONTINUE
320 THETA(NPTS)=THTEND
CALL MARKER(17)
CALL CURVE(THETA,R,NPTS,4)
CALL HEIGHT(0.1)
XPOS=XPOSN(THETA(1),VENDI)
YPOS=YPOSN(THETA(1),VENDI)
NODENO=20000+I
CALL MESSAG(NMMSG,100,XPOS+.17,YPOS+0.4)
CALL INTNO (I,'ABUT','ABUT')
CALL REALNO(QEFF,4,'ABUT','ABUT')
CALL MESSAG(QMSG4,100,'ABUT',' ABUT')
CALL VECTOR(XPOS,YPOS, XPOS+0.20, YPOS-0.5,1001)
CALL VECTOR(XPOS,YPOS, XPOS+0.40,YPOS-0.1,1001)
CALL RESET('HEIGHT')
ENDIF
C PUT MESSAGE OF Q TO OUTSIDE SURFACE
IF(NQIN .GT. 0) THEN
CALL HEIGHT(0.1)
XPOS=1.8
YPOS=5.5
CALL MESSAG(QMSG1,100,XPOS,YPOS)
CALL MESSAG(QMSG2,100,XPOS+0.17,YPOS)
CALL MESSAG(QMSG3,100,XPOS+0.34,YPOS)
CALL REALNO(QEFF,4,'ABUT',' ABUT')
CALL MESSAG(QMSG4,100,'ABUT',' ABUT')
CALL VECTOR(XPOS,YPOS,XPOS+0.20,YPOS+0.5,1001)
CALL VECTOR(XPOS,YPOS,XPOS+0.40,YPOS+0.3,1001)
CALL RESET('HEIGHT')
ENDIF
C PUT MESSAGE OF OUTSIDE BOUNDARY NODES
DO 501 IBND=1,2
IF(TEMPS(IBND) .NE. -9999.9 .AND. NLAYRS(IBND) .GT. 0) THEN
XPOS=0.7
YPOS=3.6+(IBND-1)*2.3
CALL MESSAG(QMSG1,100,XPOS,YPOS)
CALL MESSAG(QMSG2,100,XPOS+0.17,YPOS)
CALL MESSAG(QMSG3,100,XPOS+0.34,YPOS)
CALL REALNO(TEMPS(IBND),2,'ABUT',' ABUT')
CALL MESSAG(QMSG4,100,'ABUT',' ABUT')
CALL VECTOR(XPOS,YPOS,XPOS+0.6,1001)
CALL VECTOR(XPOS,YPOS,XPOS+0.20,YPOS+0.5,1001)
CALL VECTOR(XPOS,YPOS,XPOS+0.40,YPOS+0.3,1001)
CALL RESET('HEIGHT')
ENDIF
CALL MESSAG('H/MSG1',100,'ABUT','ABUT')

IF(NCR.EQ.1) THEN
  CALL BASALF('L/C')
  CALL MESSAG('H -',I00,XPOS+0.68,YPOS)
  CALL RESET('BASALF')
ENDIF

IF(NCR.EQ.2) THEN
  CALL BASALF('L/CGREEK')
  CALL MESSAG('S -',I00,XPOS+0.68,YPOS)
  CALL BASALF('SCRIPT')
  CALL MESSAG('F -',I00,'ABUT','ABUT')
  CALL RESET('BASALF')
ENDIF

CALL TRIPLX
CALL REALNO(BNCOEF(IBND),-5,'ABUT','ABUT')
CALL RESET('HEIGHT')
ENDIF

501 CONTINUE

C PUT NUMBERS ON OUTSIDE OF THE SPHERE AT EACH SECTOR
DANGL=DTHETA*180./PI
ANG=DANGL/2.
RRRR=RADMAX+SCALEF*0.7
CALL HEIGHT(0.I0)
DO 710 I=1,NTHETA
EYEMI=I-I
ANGL=ANG+EYEMI*DANGL
CALL RLINT(I,ANGL,RRRR)
 710 CONTINUE
CALL RESET('HEIGHT')

800 CALL ENDPLO(0)
RETURN

C

ENTRY DUNPLT
CALLED FROM GEOPLT (48)
ENTRY POINT TO CLOSE PLOT FILE WHEN EXITING FROM CRYOTRAN
C

900 CALL DONEPL
RETURN
END

SUBROUTINE PLTCYL

DIMENSION XTEMP(100),YTEMP(100),PREC(5),XLINE(2),YLINE(2)
COMMON /TITL/ PTITLE
COMMON /REGION/ NTHETA, NREPS, BETA, RIN, TVOL, ROUT(N)
  1 REGNS(N), NLAYRS(N), TEMPS(N), THICK(N),
  2 TKLAY(N), MATRLS(N), MATNMS(N), RGNNMS(N)
COMMON /ULLAGE/ NLUL4, NLUL5, NTHU4, RINM, PCTFUL, RADULG,
  TVULFT,C,LG(3),LIQVAP(3)
COMMON /HTXGRS/ NHX, HTTEMP(N), NLHX(10), NLHX(10),
  1 NLTHX(10), LOTHX(10)
COMMON /STUFF/ NHTT, PL, CONV, CONVR, THETAO, DTHETA, NBSOS, ROUTSF,
  1 BNCOEF(2)
COMMON /OUTSRC/ NGM, GEFF
COMMON /TOPBOT/ NTOP, NBOT, NFTLAY, NSTLAY, NFBLAY, NSBLAY,
  * NEBLAY, ETRAT, EBRAT, FTHK, FTHK
COMMON /CYDATA/CYLHGT, NCYLAY

C

LOGICAL REGNS
LOGICAL NOSCAL
C
CHARACTER*1 CT, LG
CHARACTER*6 LIQVAP
CHARACTER*1 REGNO
CHARACTER*16 MATNMS
CHARACTER*25 RGNNMS
CHARACTER*80 TITLE

C

CHARACTER*2 NUM

C
DIMENSION PROUT(5),PThICK(5)
DIMENSION R(500), THETA(500)

DATA REGLAB/'REGION '/, RADIUS/'P_%DIUS(IN)'/
DATA ULLG/'ULLAGE'/
DATA NOSMSG/'PLOT NOT TO SCALE'/
DATA NOHX/'NO HEAT EXCHANGER'/
DATA TEQMSG/'T- $'/
DATA DEGMSG/' DEG $'/
DATA NNOMSG/'NODE NO. $'/
DATA QMSG['/'SOURCE Q INTO ALL$'/
DATA QMSG2['OUTER SURFACE NODES$'/
DATA QMSG3['Q- $'/
DATA QMSG4['(BTU/(HR-IN2))$'/
DATA BNMSG1['OUTSIDE BNDY NODE $'/
DATA BNMSG5['TO SURFACES' /
DATA CONRAD(1)/'CONVECTIONS'/
DATA CONRAD(2)/'RADIATIONS'/
DATA HXMSG1/'HEAT EXCHANGER NO. $'/

C C INITIALIZE OUTPUT TO (1) QMS PRINTER OR TO (2) TERMINAL SCREEN
PRINT 2001
2001 FORMAT(15X,'IN THE SPHERE PLOTTING ROUTINE'/
1 'SEND THE GRAPH TO '/
2 '1. THE QMS PRINTER'/
3 '2. THE TERMINAL SCREEN'/
4 'TYPE IN 1 OR 2'/)
CALL READIN(IDV, I,3)
IF(IDV .EQ. 1) CALL QMS2
IF(IDV .EQ. 2) CALL IBM52(3179,0,0,0)
C NEW SUBROUTINE CALL FROM DAVE HUBLER TO ALLOW USER TO CHOOSE DEVICE
IF(IDV .EQ. 3) CALL PDEV(' ',ISTAT)
C SET PAGE SIZE
CALL PAGE(11.0,8.5)
C SET SUBPLOT SIZE
CALL AREA2D(8.0, 8.0)
CALL GRAP (0.,10.,400.,0.,10.,400.)

CALL TRIPLEX
C SCALE REGION THICKNESSES UP IF REAL SCALE IS TOO SMALL
NOSCAL = .FALSE.
SCALEF=RIN/20.
DO 5 I=1,5
PROUT(I)=ROUT(I)
PThICK(I)=PThICK(I)
IF(REGNS(I)) THEN
  IF(PThICK(I) .LT. 1.) THEN
    PThICK(I)=PThICK(I)+SCALEF
  ENDIF
ENDIF
5 CONTINUE
IF(NOSCAL) THEN
  THKSUM=0.0
  DO 7 I=1,3
    IF(REGNS(I)) THEN
      THKSUM=THKSUM+PThICK(I)
    ENDIF
  PROUT(I)=RIN+THKSUM
ENDIF
ENDIF
 Continue
 IF (THICK(4) .LT. 1.) THEN
 PROUT(5) = RIN - PTINCH(4)
 PTINCH(5) = PROUT(5)
 ENDIF
 ENDIF

 IF (THICK(4) .LT. I.) THEN
 PROUT(5) = RIN - PTINCH(4)
 PTINCH(5) = PROUT(5)
 ENDIF

 ENDIF

 RADMAX = AMAX1 (PROUT(1), PROUT(2), PROUT(3))

 RSTEP = RADMAX/4.5

 WRITE THE PLOT TITLE

 CALL ANGLE(90.)

 CALL MESSAG(NOSMSG, 17, 9, 500)

 IF (NBOT.EQ.2) THEN
 XTEMP(1) = 300
 YTEMP(1) = 150
 XTEMP(2) = 325
 YTEMP(2) = 150
 XTEMP(3) = 325
 YTEMP(3) = 250
 XTEMP(4) = 300
 YTEMP(4) = 250
 CALL THKCRV(0.02)
 CALL CURVE(XTEMP, YTEMP, 4, 0)
 CALL THKCRV(0.01)
 RAD = 25./NFBLAY
 XTEMP(1) = 300
 YTEMP(1) = 150
 XTEMP(2) = 300
 YTEMP(2) = 250
 DO 31 I = 1, NFBLAY - 1
 XTEMP(1) = XTEMP(1) + RAD
 XTEMP(2) = XTEMP(1)
 CALL CURVE(XTEMP, YTEMP, 2, 0)
 CALL THKCRV(0.02)
 CALL CURVE(XTEMP, YTEMP, 4, 0)
 CALL CURVE(XTEMP, YTEMP, 2, 0)

 CONTINUE

 RAD = 250.
 XTEMP(1) = 300
 XTEMP(2) = 325

 DO 32 I = 1, 5
 IF (REGNS(I)) THEN
 IF (1.EQ.1) SCALE = 10
 IF (1.EQ.2) SCALE = 10
 IF (1.EQ.3) SCALE = 10
 IF (1.EQ.4) SCALE = 40
 IF (1.EQ.5) SCALE = 10
 CALL THKCRV(0.02)
 XTEMP(1) = 300
 XTEMP(2) = 325
 XTEMP(3) = 325
 XTEMP(4) = 300
 YTEMP(1) = YTEMP(1) - SCALE
 YTEMP(2) = YTEMP(2) - SCALE
 CALL CURVE(XTEMP, YTEMP, 2, 0)
 CALL CURVE(XTEMP, YTEMP, 2, 0)
 CALL CURVE(XTEMP, YTEMP, 2, 0)
 CALL CURVE(XTEMP, YTEMP, 2, 0)
 CALL CURVE(XTEMP, YTEMP, 2, 0)
 CALL CURVE(XTEMP, YTEMP, 2, 0)

 CONTINUE

 RAD = 250.
 XTEMP(1) = 300
 XTEMP(2) = 325

 DO 32 I = 1, 5
 IF (REGNS(I)) THEN
 IF (1.EQ.1) SCALE = 10
 IF (1.EQ.2) SCALE = 10
 IF (1.EQ.3) SCALE = 10
 IF (1.EQ.4) SCALE = 40
 IF (1.EQ.5) SCALE = 10

 CONTINUE

BAD - RAD - SCALE
YTEMP (1) = RAD
YTEMP (2) = RAD
DO 31 J = 1, NLAYERS (I) - 1
YTEMP (1) = YTEMP (1) + (SCALE / NLAYERS (I) )
YTEMP (2) = YTEMP (2) + (SCALE / NLAYERS (I) )
CALL THKCRV (0.01)
CALL CURVE (XTEMP, YTEMP, 2, 0)
CALL RESET ('THKCRV')
CONTINUE
31 CONTINUE
ENDIF
32 CONTINUE
ENDIF

IF (NTOP.EQ.2) THEN
XTEMP (1) = 75
YTEMP (1) = 150
XTEMP (2) = 100
YTEMP (2) = 150
XTEMP (3) = 100
YTEMP (3) = 250
XTEMP (4) = 75
YTEMP (4) = 250
XTEMP (5) = 75
YTEMP (5) = 150
CALL THKCRV (0.02)
CALL CURVE (XTEMP, YTEMP, 5, 0)
CALL THKCRV (0.01)
RAD = 25. / NFTLAY
XTEMP (1) = 75
YTEMP (1) = 150
XTEMP (2) = 75
YTEMP (2) = 250
DO 733 I = 1, NFTLAY - 1
XTEMP (1) = XTEMP (I) + RAD
XTEMP (2) = XTEMP (I)
CALL CURVE (XTEMP, YTEMP, 2, 0)
733 CONTINUE
YTEMP (1) = 250.
YTEMP (2) = 250.
DO 730 I = 1, 5
IF (REGNS (I)) THEN
IF (I.EQ.1) SCALE = 10
IF (I.EQ.2) SCALE = 10
IF (I.EQ.3) SCALE = 10
IF (I.EQ.4) SCALE = 40
IF (I.EQ.5) SCALE = 30
CALL THKCRV (0.02)
XTEMP (1) = 75
XTEMP (2) = 100
YTEMP (1) = YTEMP (I) - SCALE
YTEMP (2) = YTEMP (2) - SCALE
CALL CURVE (XTEMP, YTEMP, 2, 0)
CALL RESET ('THKCRV')
ENDIF
730 CONTINUE
RAD = 250.
XTEMP (1) = 75
YTEMP (2) = 100
DO 732 I = 1, 5
IF (REGNS (I)) THEN
IF (I.EQ.1) SCALE = 10
IF (I.EQ.2) SCALE = 10
IF (I.EQ.3) SCALE = 10
IF (I.EQ.4) SCALE = 40
IF (I.EQ.5) SCALE = 30
RAD = RAD - SCALE
YTEMP (: ) = RAD
YTEMP (2) = RAD
DO 731 J=1,NLAYRS(I)-1
YTEMP (1) = YTEMP (1) + (SCALE/NLAYRS (I))
YTEMP (2) = YTEMP (2) + (SCALE/NLAYRS (I))
CALL THKCRV (0.01)
CALL CURVE (XTEMP, YTEMP, 2, 0)
CALL RESET ('THKCRV')
731 CONTINUE
ENDIF
732 CONTINUE
ENDIF
RADIUS = 100

IF (NBOT.EQ.3 OR NBOT.EQ.4) THEN
IF (NBOT.EQ.3) NUMLAY = NSBLAY
IF (NBOT.EQ.4) NUMLAY = HEBLAY
NUMINT = INT (100/NUMLAY)
INTER = INT (100/NUMINT)
X1 = 300
Y1 = 150
RADIUS = 100
RADIUS = RADIUS/100.
XTEMP (1) = X1 + RADIUS
YTEMP (1) = (((RADIUS**2.) - ((XTEMP (1) - X1)**2.))** (1./2.)) + Y1
DO 1 I=2,100
XTEMP (I) = XTEMP (I-1) - RADIUS
YTEMP (I) = (((RADIUS**2.) - ((XTEMP (I) - X1)**2.))** (1./2.)) + Y1
1 CONTINUE
CALL THKCRV (0.02)
CALL CURVE (XTEMP, YTEMP, 100, 0)
CALL RESET ('THKCRV')
ANG = 90./NUMLAY
ANG = (PI/180.)*ANG
DO 65 J = 1, NUMLAY-1
TOTANG = J*ANG
XLINE (1) = 300
YLINE (1) = 150
XLINE (2) = 300 + (100 * COS (TOTANG))
YLINE (2) = 150 + (100 * SIN (TOTANG))
CALL CURVE (XLINE, YLINE, 2, 0)
65 CONTINUE
DO 11 J=1,5
IF (FSIZE (J)) THEN
IF (J.EQ.1) SCALE = 10
IF (J.EQ.2) SCALE = 10
IF (J.EQ.3) SCALE = 10
IF (J.EQ.4) SCALE = 40
IF (J.EQ.5) SCALE = 30
RADIUS = RADIUS/SCALE
RADIUS = RADIUS/100.
XTEMP (1) = X1 + RADIUS
YTEMP (1) = (((RADIUS**2.) - ((XTEMP (1) - X1)**2.))** (1./2.)) + Y1
DO 12 I=2,100
XTEMP (I) = XTEMP (I-1) - RADIUS
YTEMP (I) = (((RADIUS**2.) - ((XTEMP (I) - X1)**2.))** (1./2.)) + Y1
12 CONTINUE
CALL THKCRV (0.02)
CALL CURVE (XTEMP, YTEMP, 100, 0)
CALL RESET ('THKCRV')
RADIUS = RADIUS
DO 71 I=1,NLAYRS (J)
RADIUS = RADIUS/100.
XTEMP (1) = X1 + RADIUS
YTEMP (1) = (((RADIUS**2.) - ((XTEMP (1) - X1)**2.))** (1./2.)) + Y1
DO 81 K=2,100
XTEMP (K) = XTEMP (K-1) - RADIUS
YTEMP (K) = (((RADIUS**2.) - ((XTEMP (K) - X1)**2.))** (1./2.)) + Y1
81 CONTINUE
CALL THKCRV(0.01)
CALL CURVE(XTEMP,YTEMP,100,0)
CALL RESET ("THKCRV")

71 CONTINUE
ENDIF

11 CONTINUE
RAD = 100.
XTEMP(1) =300
XTEMP(2) =300 + RAD
XVBOT = XTEMP(2)
YTEMP(1) =150
YTEMP(2) =150
CALL THKCRV (0.02)
CALL CURVE(XTEMP,YTEMP,2,0)
CALL RESET ("THKCRV")

PAD - I00.
XTEMP(I) =-300
XTEMP(2) =-300 + RAD
XVBOT = XTEMP(2)
YTEMP(I) =-150
YTEMP(2) =-150
CALL THKCRV (0.02)
CALL CURVE(XTEMP,YTEMP,2,0)
CALL RESET ("THKCRV")

PAD - I00.
XTEMP(I) =-300
XTEMP(2) =-300
XTEMP(3) =-100
XTEMP(4) =-100
XTEMP(5) =-300
YTEMP(1) =150
YTEMP(2) =150 + RAD
YTEMP(3) =150 + RAD
YTEMP(4) =150
YTEMP(5) =150
CALL THKCRV (0.02)
CALL CURVE(XTEMP,YTEMP,5,0)
CALL RESET ("THKCRV")

XTEMP(I) =100
XTEMP(2) =300
YTEMP(I) =150 + RAD
YTEMP(2) =150 + RAD
DO 13 J = 1, 5
IF (REGNS(J)) THEN
   IF (J.EQ.1) SCALE = 10
   IF (J.EQ.2) SCALE = 10
   IF (J.EQ.3) SCALE = 10
   IF (J.EQ.4) SCALE = 10
   IF (J.EQ.5) SCALE = 10
   YTEMP(I) = YTEMP(I) - SCALE
   YTEMP(2) = YTEMP(2) - SCALE
   YA = YTEMP(I)
   YB = YTEMP(2)
CALL THKCRV (0.02)
CALL CURVE(XTEMP,YTEMP,2,0)
CALL RESET ("THKCRV")
RAD2 = RAD
DO 73 I = 1, NLAYRS(J)
   RAD2 = RAD2 + (SCALE/NLAYRS(J))
   RADINT = SCALE/NLAYRS(J)
   XTEMP(I) = 100.
   YTEMP(I) = YTEMP(I) + RADINT
   XTEMP(2) = 300.
   YTEMP(2) = YTEMP(2) + RADINT
   CALL THKCRV (0.01)
   CALL CURVE(XTEMP,YTEMP,2,0)
   CALL RESET ("THKCRV")
73 CONTINUE
YTEMP(I) = YA
YTEMP(2) = YB
ENDIF

13 CONTINUE
XTEMP(I) = 100
SCALE = 200./NCYLAY
DO 14 I = 1,NCYLAY
   XTEMP(I) = XTEMP(I) + SCALE
   XTEMP(2) = XTEMP(1)
14 CONTINUE
YTEMP(1) = 150
YTEMP(2) = YTEMP(1) + RAD
CALL THKCRV(0.01)
CALL CURVE(XTEMP, YTEMP, 2, 0)
CALL RESET ('THKCRV')
CONTINUE

IF (NTOP.EQ.3.OR.NTOP.EQ.4) THEN
  IF (NTOP.EQ.3) NUMLAY = NSTLAY
  IF (NTOP.EQ.4) NUMLAY = NETLAY
  X1 = 100
  Y1 = 150
  RAD = 100.
  NUMINT = INT (100/NUMLAY)
  RADINT = RAD / 100.
  XTEMP(I) = X1 + RAD
  YTEMP(I) = (((RAD**2.) - ((XTEMP(I) - X1)**2.))**(1./2.)) + Y1
  DO 2 I=2,100
  XTEMP(I) = XTEMP(I-1) - RADINT
  YTEMP(I) = (((RAD**2.) - ((XTEMP(I) - X1)**2.))**(1./2.)) + Y1
  2 CONTINUE
  DO 3 I=1,100
  XTEMP(I) = 200 - XTEMP(I)
  3 CONTINUE
  CALL THKCRV (0.02)
  CALL CURVE (XTEMP, YTEMP, 100, 0)
  CALL RESET ('THKCRV')
  ANG = 90./NUMLAY
  ANG = (PI/180.)*ANG
  DO 66 J = 1, NUMLAY-1
  TOTANG = J*ANG
  XLINE (1) = 100
  YLINE (1) = 150
  XLINE (2) = 100 + (100 * COS (TOTANG))
  YLINE (2) = 150 + (100 * SIN (TOTANG))
  XLINE (1) = 200 - XLINE(1)
  XLINE (2) = 200 - XLINE(2)
  CALL CURVE (XLINE, YLINE, 2, 0)
  66 CONTINUE
  XTEMP(1) = 0
  XTEMP(2) = 100
  CALL THKCRV (0.02)
  CALL CURVE (XTEMP, YTEMP, 2, 0)
  CALL RESET ('THKCRV')
  DO 22 J=1,5
  IF (REIONS(J)) THEN
    IF (J.EQ.1) SCALE = 10
    IF (J.EQ.2) SCALE = 10
    IF (J.EQ.3) SCALE = 10
    IF (J.EQ.4) SCALE = 10
    IF (J.EQ.5) SCALE = 10
    RAD = RAD - SCALE
    RAD2 = RAD - SCALE
    XTEMP(I) = X1 + RAD
    YTEMP(I) = (((RAD**2.) - ((XTEMP(I) - X1)**2.))**(1./2.)) + Y1
    DO 22 I=2,100
    XTEMP(I) = XTEMP(I-1) - RADINT
    YTEMP(I) = (((RAD**2.) - ((XTEMP(I) - X1)**2.))**(1./2.)) + Y1
    22 CONTINUE
    DO 23 I=1,100
    XTEMP(I) = 200 - XTEMP(I)
    23 CONTINUE
    CALL THKCRV (0.02)
    CALL CURVE(XTEMP, YTEMP, 100, 0)
    CALL RESET ('THKCRV')
    RAD2 = RAD
    DO 72 I=1,NLAYRS(J)
    RAD = RAD2 + (SCALE/NLAYRS(J))
RADINT = RAD2 / 100.
XTEMP(1) = X1 + RAD2
YTEMP(1) = ((RAD2**2.) - (XTEMP(1) - X1)**2.)**1./2.) + Y1
DO 82 K=2,100
XTEMP(K) = XTEMP(K-1) - RADINT
YTEMP(K) = ((RAD2**2.) - (XTEMP(K) - X1)**2.)**1./2.) + Y1
CONTINUE
82 DO 53 IJ=1,100
XTEMP(IJ) = 200 - XTEMP(IJ)
CONTINUE
DO 82 K=2,100
XTEMP(K) = XTEMP(K-1) - RADINT
YTEMP(K) = ((RAD2**2.) - (XTEMP(K) - X1)**2.)**1./2.) + Y1
CONTINUE
DO 53 IJ=1,100
XTEMP(IJ) = 200 - XTEMP(IJ)
CONTINUE
CALL TBKCRV(0.01)
CALL CURVE(XTEMP, YTEMP, 100, 0)
CALL RESET('THKCRV')
72 CONTINUE
ENDIF
21 CONTINUE
RAD = 100.
ENDIF
CALL REALNO(CYLHGT/2.,4_300,150)
CALL REALNO(CYLHGT/2.,4_100,150)
IF (NBOT.EQ.2) THEN
VALUE=(CYLHGT/2.)+FTHK
CALL REALNO(VALUE,4_325,150)
ENDIF
IF (NTOP.EQ.2) THEN
VALUE=(CYLHGT/2.)+FTHK
CALL REALNO(VALUE,4_75,150)
ENDIF
IF (NBOT.EQ.3) THEN
VALUE=(CYLHGT/2.)+RIN
CALL REALNO(VALUE,4_300,150)
ENDIF
IF (NTOP.EQ.3) THEN
VALUE=(CYLHGT/2.)+RIN
CALL REALNO(VALUE,4_75,150)
ENDIF
C PUT MESSAGE OF Q TO OUTSIDE SURFACE
IF (NQIN .GT. 0) THEN
CALL HEIGHT(0.1)
XPOS=5.2
YPOS=0.0
CALL MESSAG(MQSG1,100,XPOS,YPOS)
CALL MESSAG(MQSG2,100,XPOS+0.17,YPOS)
CALL MESSAG(MQSG3,100,XPOS+0.34,YPOS)
CALL REALNO(MEFF,4,'ABUT','ABUT')
CALL MESSAG(MQSG4,100,'ABUT','ABUT')
YPOS=0.15
CALL VECTOR(XPOS,YPOS+2.0,XPOS-0.20,YPOS+2.5,1001)
CALL REALNO(TEMPS(IBND+5),2,'ABUT','ABUT')
ENDIF
C PUT MESSAGE OF OUTSIDE BOUNDARY NODES
DO 501 IBND=1,2
IF (TEMPS(IBND+5) .NE. -9999.9 .AND. NLAYRS(IBND+7) .GT. 0) THEN
XPOS=2.5 + (IBND-1) *1.3
YPOS=0.5
CALL HEIGHT(0.1)
NODENO=20300+IBND
CALL MESSAG(BNMSG1,100,XPOS,YPOS)
CALL INTNO (IBND,'ABUT','ABUT')
CALL MESSAG(NNOMSG,100,XPOS+0.17,YPOS)
CALL INTNO (NODENO,'ABUT','ABUT')
CALL MESSAG(TECMSG,100,XPOS+0.34,YPOS)
CALL REALNO(TEMPS(IBND+5),2,'ABUT','ABUT')
ENDIF
CALL MESSAG(DEGMSG, 100, 'ABUT', 'ABUT')
CALL MESSAG(MATNMS(9), 1, 'ABUT', 'ABUT')
NCR = NLAYRS(IBND + 7)
CALL MESSAG(CONRAD(NCR), 100, XPOS + 0.51, YPOS)
CALL MESSAG(BNMSG5, 100, 'ABUT', 'ABUT')
IF (NCR .EQ. 1) THEN
C CONVECTION, PUT OUT L/C H
CALL BASALF('L/CSTD')
CALL MESSAG('H = $', 100, XPOS + 0.68, YPOS)
CALL RESET('BASALF')
ENDIF
IF (NCR .EQ. 2) THEN
C RADIATION, PUT OUT SCRIPT F
CALL BASALF('L/CGREEK')
CALL MESSAG('F = $', 100, 'ABUT', 'ABUT')
CALL RESET('BASALF')
CALL TRIPLX
CALL REALNO(BNCOEF(IBND), -5, 'ABUT', 'ABUT')
ENDIF
501 CONTINUE
IF (NHX.EQ.0) THEN
XPOS = 0.6
YPOS = 5.2
CALL MESSAGE (NOHX, 100, XPOS - 0.17, YPOS)
ENDIF
IF (NHX.GT.0) THEN
DO 503 I = 1, NHX
XPOS = 0.6 + (I - 1) * 1.1
YPOS = 5.2
NODENO = 20000 + I
CALL MESSAGE(NEWMSG1, 100, XPOS - 0.17, YPOS)
CALL INTNO(I, 'ABUT', 'ABUT')
CALL MESSAG(MNMSG, 100, XPOS, YPOS)
CALL INTNO(NODENO, 'ABUT', 'ABUT')
CALL MESSAG(TEMMSG, 100, XPOS + 0.17, YPOS)
CALL REALNO(HXTEMP(I), 2, 'ABUT', 'ABUT')
CALL MESSAG(DEGMSG, 100, 'ABUT', 'ABUT')
CALL MESSAG(MATNMS(9), I, 'ABUT', 'ABUT')
CALL MESSAG('ON LAYER $', 9, XPOS + 0.34, YPOS)
CALL INTNO(NRHX(I), 'ABUT', 'ABUT')
CALL MESSAG(' IN REGION $', I, 'ABUT', 'ABUT')
CALL INTNO(LNTHX(I), 'ABUT', 'ABUT')
CALL MESSAG(' NODE(S).S', 9, 'ABUT', 'ABUT')
CONTINUE
503 CONTINUE
ENDIF
CALL ENDPLO()
RETURN
END
APPENDIX E

CryoTran Program Listings

Part V  VM Exec Files
VM Exec File  RUNCryo EXEC

/* THIS EXECUTES PROGRAM CRYOTRAN */
/* THIS EXEC DOES NOT ACCESS THE PLOTTING ROUTINES */
/* USE WHEN RUNNING NON-SINDA TYPE CASES OR WHEN */
/* PLOTS ARE NOT NEEDED. */
/*FILEDEF FT04F001 DISK MATERIAL DBASE FOR THE MATERIAL DBASE*/
/*FILEDEF FT10F001 DISK FN FT FM*/
FILEDEF FT04F001 DISK MATERIAL DBASE M
FILEDEF FT09F001 DISK CRYOTRAN INPU TekO
FILEDEF FT10F001 DISK CRYOTRAN MODEL
FILEDEF FT17F001 DISK PROGRAM OUTPUT
FILEDEF FT25F001 DISK N2 TABLE M
FILEDEF FT26F001 DISK O2 TABLE M
FILEDEF FT27F001 DISK N2 TABLE M
FILEDEF FT35F001 DISK SCRATCH
FTNLIB
LOAD CRYOTRAN '( CLEAR
INCLUDE CRYOSPHR
INCLUDE CRYOCYL
INCLUDE CRYVMSUB
INCLUDE CRYOPLOT
START
FILEDEF '*' CLEAR
VM Exec RUNCryo

/* THIS EXECUTES PROGRAM CRYOTRAN */
/* ACCESS GRAPH3D OR NECESSARY JCL TO ACCESS DISSPLA */
/* FILEDEF FT04F001 DISK MATERIAL DBASE FOR THE MATERIAL DBASE */
FILEDEF FT10F001 DISK FM FT FM /
SETUP DISSPLA
SETUP GDDM
DRUN CRYOPLOT
FILEDEF FT04F001 DISK MATERIAL DBASE M
FILEDEF FT09F001 DISK CRYOTRAN INPUTeko
FILEDEF FT10F001 DISK CRYOTRAN MODEL
FILEDEF FT20F001 DISK H2 TABLE M
FILEDEF FT21F001 DISK O2 TABLE M
FILEDEF FT22F001 DISK H2 TABLE M
FILEDEF FT17F001 DISK PROGRAM OUTPUT
FTPRLIB
LOAD CRYOTRAN('( 'CLEAR
INCLUDE CRYOSPHER
INCLUDE CRYOCYL
INCLUDE CYRCYACO
START
FILEDEF ' ' CLEAR

VM Exec CRYOLINK

/* THIS EXECUTES PROGRAM CRYOTRAN */
FTPRLIB
LOAD CRYOTRAN('( 'CLEAR
INCLUDE CRYOSPHER
INCLUDE CRYOCYL
INCLUDE CYRCYACO
START
FILEDEF ' ' CLEAR

280
/*TRACE & */ /* for debugging*/
arg fname (' options ')'/*GET FILE NAME AND OPTIONS OR OFFER HELP SCREEN */

if fname = '?' | fname = 'HELP' then do
call Getoptions ?
end

if fname = '' then /*A valid filename must be given*/
do
VMFCLEAR
say;say;say;say;say
SAY 'A valid filename of the form "filename FORTRAN" is required'
say 'please enter the name of your FORTRAN source file'
pull upper pull fname.
state fname fortran
if rc = 0 then do
say 'DISSPLA source file 'fname 'FORTRAN not found''
call exit_exec -93
end
end

'globalv select ISSCOGRP' /*temporary global variables indicate*/
'globalv stack DIS110'/*if user has seen option screen once*/
pull oncethru/'during current IPL'/*
if oncethru = 1 then do
'options = options]' 'fake_option'
end
/*y 'options now are ' options */
call Getoptions options /*parse options entered on command line*/
DISSPOPT = 1
'globalv select ISSCOGRP set DIS110' DISSPOPT

/***** DETERMINE NEED FOR FURTHER STORAGE ACQUISITION ******/
makebuf
'query stor [stack]' /*check user's memory = DISSPLA needs 6meg*/
281
pull . . storage .
storage = left(storage, length(storage)-1) /*drop the K*/
dropbuf
if storage < 6144 then do /*test for 6 meg of memory*/
say
say 'There is not enough virtual storage for DISSPLA II.0'
say 'More virtual storage has been allocated.'
say 'Please type IPL CMS, and then restart DISSPLA.'
say 'To permanently change your memory enter:'
say ' ==> VMSECURE MAI STOR 6M '
say 'and then logoff and log back on to make the new'
say ' storage allocation available ' 
'scp def stor 6m'
end

*************** SET UP ACCOUNTING OF DISSPLA USE ************
/* user got filename and options right */
/* and he has enough memory so record usage of software */
set cmstype ht
vmacct pack start disspla
vmacct pack end /*set up vmaccount software to count # of uses*/
set cmstype rt

****** ACQUIRE TEMPORARY STORAGE ******
call qdix
workdisk = result
wd = result /* I am lazy, workdisk is too long & takes too much space*/
say
say 'Assigning temporary storage destination to disk ' workdisk
say
TDISK 10 workdisk

************** SETUP LIBRARIES NEEDED BY ISSCO **************/
GLOBAL T×TLIB DIIOMOD /*ADD LIBRARIES USED FOR PACKAGE*/
/*libraries are added in the same order as the execs supplied by CA-ISSCO*/
if resting='YES' then do /* if TEST option is called*/
addllb disman /*add library of test plots*/
end
select
when gksfile='YES' then do /*if GKS option is called*/
addllb gkslib /*add gks library*/
en
derwise do
addllb diss10a /*otherwise add disspla A-library*/
en
deselect clause*
addllb diss10b /*add disspla B-library in either case*/
addllib intlib /*always need device driver library*/
if rc ^= 0 then say 'DISSPLA product not linked - do SETUP DISSPLII'

/*g loadlib*/
cp sleep 5 sec*/
global loadlib dynlib /*load dld drivers*/
if versatec = 'VERS31' | versatec = 'VERS42' | versatec = 'VERSVU'
then do
addllb clr /*add versatec color random library*/
end
/*addllb clr */ /*add it to resolve references in testing*/
SELECT
when tektrnx = 'TEK' then do
   /* do nothing - do not use gddm library - it will*/
end
   /*cause problems with tek terminals*/
otherwise do
setup gddm /*gddm library interferes with tek calls*/
end
END
setup ftn /*add fortran libraries*/
if tlib = YES then do
say 'You have requested an additional txtlib to be searched for'
say 'programs'
say 'Please enter the name of the library -->'
parse upper pull addlib_name .
addlib addlib_name
end

/**QUERY TXTLIB*/ /*/* USED FOR DEBUGGING*/
/**CP SLEEP 5 SEC*/ */
/***** SAVE ENVIRONMENT AND BEGIN ISSCO INITIALIZATION *****/

/***** SET TERMINAL ENVIRONMENT AS REQUESTED BY ISSCO PRODUCTS *****/
call SaveTerminalEnvironment
makebuf
   'query blip (stack'
pull . . blip_state .
   'query idrtbls (stack'
pull . . loader_tables .
dropbuf
   'set blip off'
   'set idrtbls 15'
   'set msg off'
   'set wng off'
   'terminal linesize 255'
   'terminal escape off'
   'terminal llnend off'
   'terminal chardel off'

/***** ISSUE ISSCO-REQUIRED AND SITE-SPECIFIC FILEDEFS *****/
   'cp sleep 5 sec'
call file definitions
   'cp sleep 5 sec'

/*************** COMPILE PROGRAM AND RUN ***********************
if "cryoflag then do /*this do group created to bypass normal*/
   /*normal way to load and run*/
   /*added for G. Cowgill at Analea*/
   /*end of this loop has commands for runcryo*/
   /*program*/
gennodule=0
call GetFileAge fname fortran
src_age = result
call GetFileAge fname text
txt_age = result
call GetFileAge fname module
mod_age = result
if src_age<0 & txt_age<0 & mod_age<0 then
  do
    SAY 'DISSPLA source program `fname' FORTRAN not found'
call scratch_tdsk workdisk
call exit_exec -93
  end
end

if src_age > txt_age & src_age > mod_age then
  do
genmodule = 1
  /* 'THE SOURCE AGE IS ' SRC_AGE
say 'the txt age is ' txt_age
SAY 'THE MODULE AGE IS ' MOD_AGE
CP SLEEP 5 SEC */
  for:vs fname
  if rc > 4 then do
    say 'there are errors in the source program ' fname
call scratch_tdsk workdisk
call exit_exec rc
  end
  mod_age = -99
end

if txt_age > mod_age then
  do
genmodule = 1
say relinking
load fname
  if rc > 4 then do
    say 'there are errors in the link edit - condition code ' rc
call scratch_tdsk workdisk
call exit_exec rc
  end
  /* create a module if so directed */
  if genmodule then
    do
      say fname ' load module being generated.........'
genmod fname
    end
  end
end
****** RUN DISSPLA MODULE *****/
say
say 'Now loading DISSPLA . . .
set msg off
if txt_age > mod_age then
  do
    start
  end
end
if mod_age > txt_age then
  do
    'run' fname 'module'
  end
end
if rc<>0 then
  do
    say 'non-zero return code from DISSPLA. Return code is ' rc
    return_code = rc
  end
else do
  return_code=0
end
set msg on
end /*end of do if not cryo*/
else do /*must be cryoflag is 1*/
/*y 'into the cryo part for compiling'*/
cryolink return_code = 0
genmodule = 0
eend

//***** OFFLINE GRAPHICS DATA FILE DETECTION AND PROCESSING *****/
if noplot ^= 1 then noplot = 0
if noplot then do
call Scratch_tdsk workdisk
call exit_exec return_code
eend
call SavePrinterEnvironment

/*** QMS FILES ***/
call IsThereFile 'HP7550 or TALARIS, QMS LASER PRINTER', 'STD* DATA A'
if queued() > 0 then do /*if no files in queue */
   /*skip this part of exec*/
valid_response = 0
do until valid_response /*if they don't get it right the lst*/
dropbuf /*time make them answer till they do*/
say say 'Please choose the plotter to route your file to.'
say 'Enter:'
say 'Q - to route the file to a QMS plotter'
say 'T - to route the file to a LIMS Talaris printer'
say 'WP - to save an HPGL file to import into Wordperfect'
say 'H - to route the file to a LIMS HP7550 plotter'
/*
**/ /*DH Remove following comment to enable users Postscript access*/
/* **/
/*say 'P - to route Postscript file to the VAX LPS40 printer'*/
parse upper pull plotter_response
/*SAY 'THE RESPONSE WAS ****PLOTTER_RESPONSE**** */
if plotter_response = Q |,
   plotter_response = T |,
   plotter_response = P |,
   plotter_response = DH |,
   plotter_response = WP |,
   plotter_response = H then valid_response=1
else valid_response = 0
end /*end of do until valid_response loop*/

if PLOTTER_RESPONSE = DH then do /*hubler testing exit*/
   exit -7550
   end /*end of do for HP7550 problems*/

select
when plotter_response = WP then do
   say 'Please enter file mode for HPGL file'
   parse upper pull file_mode
   listfile 'std* data a (stack' /*put all QMS files (STD0000X) files*/
do while queued() > 0 /* on the stack*/
pull filename typemode
   'copyf ' filename typemode filename ' hpgl ' file_mode
   if rc = 0 then erase filename typemode
   end /* hpgl_file=1 */
   end /*end of when WP*/
when plotter_response = Q then do
done = 'false'
do while done = 'false'
say 'Please choose a QMS printer site for your data.'
say ' 1) RAC'
say ' 2) ERB'
say ' 3) LGAOS (Analex)'
say 'Enter the number of your choice:'
pull choice
select
  when choice = 1 then do
    tag dev prt mvsercl rmt7
    done = 'true'
  end
  when choice = 2 then do
    tag dev prt mvsercl rmt10
    done = 'true'
  end
  when choice = 3 then do
    tag dev prt mvsercl rmt34
    done = 'true'
  end
otherwise do
  say 'Improper selection ... Try again.'
end
end
end
end
end
end

dooprt nohold recs
listfile 'std* data (stack' /*put all QMS files (STD0000x) files*/
do while queued() > 0 /* on the stack*/
pull filename
  /*we must reset QMS*/
  /*to landscape mode*/
execlo 1 diskw filename '{ string "PY"--'
/*note that there are no spaces prior to execlo string with cc opt*/
execlo 1 diskw filename '{ string "IOL"FN"--' /*cause ISSCO*/
  /*doesn't*/
print filename '(noco notrc li 00'
erase filename
end
end /*end of the first select clause for plotter_response*/

********************************************
/*This part of the exec sends an ISSCO QMS plot file to the */
/*LIMS Talaris printers or HP 7550 plotters*/
when plotter_response = T | plotter_response = H then do
  if plotter_response = T then do
    /* send to Talaris or HP7550? */
    tp = T
  end
  else do
    tp = P
  end
  /* get the user and device ID's */
makebuf
 'q cons (stack'
pull
pull
pull .. user /*get the user id*/
dropbuf

vrfclear
Say 'Please enter the Talaris printer or HP plotter ID you want your'
say 'plot sent to' 
say 'for example' 
say 'enter B14201 for the Talaris or HP7550 on the RAC second floor' 
say 'or B50081 for the Talaris or HP7550 on the DEB second floor' 
say 'or B50111 for the Talaris or HP7550 in the DEB annex' 
say 'etc.' 
parse upper pull t_d

/** check for Interlink software and link if not found **/
call Access_Interlink

/** send to device **/

 listfile std* data a (stack) /* put list of plot files on stack*/
do while queued() > 0
 pull filename /* take names off stack and send them*/

******************************************************************************
**** Interlink Modification due to VMS upgrade ***********************
******************************************************************************
' nft send lims02::tp::'user'.vm ' filename '/nocc' /*
above is replacement nft command until LIMS bug due to VMS 5.1 */
upgrade of June 25 is fixed. Then below 2 nft commands will also/*
' work. /*
' /*
' ' nft send lims01::'user'.vm ' filename '/nocc' /*
' cp smog decmcs cmd lims01 print/delete/queue='t_d'_tp 'user'.vm'/*
******************************************************************************

erase filename /* clean up after ourselves*/
end /* end of do while*/

if we_linked_to_nft then do
 set cmstype ht
 'release nftmode (det' /* bye bye nft software minidisk*/
 set cmstype rt
end /* end of select clause for plotter_response - T*/

when plotter_response = P then do /* this is for postscript files*/
say 'The file will be sent to the Postscript printer'
set cmstype ht
call Access_Interlink

 'makebuf' /* get user's bin number*/
bufno = rc
'q cons (stack'
pull;pull;pull ... . . bin .
'dropbuf' bufno

 listfile std* data a (stack) /* put list of plot files on stack*/
do while queued() > 0
 pull filename /* take names off stack and send them*/

xxxxx = strip(userid()) ] strip(time('s')) /* make unique filename*/

'MAKEBUF'
bu_no-rc
queue "$ ASSIGN" bin "NASASBIN"
queue "$ LPS40/P" xxxxx*.LPS
queue "$ DELETE" xxxxx*.LPS;*
queue "$ DELETE" xxxxx*.COM;**
queue "$ RENAME" xxxxx*.LOG RACCESS.LOG
"execio 5 DISKX CA-LPS40 COM A (FINIS'
dropbuf bufno

'nf send venus"RACCESS REMACC2":'xxxxx'.lps' filename
'nf send venus"RACCESS REMACC2":'xxxxx'.com CA-LPS40 COM A'
'nf submit venus"RACCESS REMACC2":'xxxxx'.com'
end /*end of queued file on stack*/

/*send the files*/

if we_linked_to_nft then 'release 'nftmode ' (det'
' erase ca-lps40 com a'
' erase ' filename
end /*end of Postscript portion of select clause*/

otherwise /*if user entered anything other than Q or T*/
do
set cmstype ht
' listfile std* data a (stack' /*list files left on user's minidisk*/
set cmstype rt
do while queued() > 0
pull filename
say 'DISSPLA:INVALID plotter_response - file not plotted'
/* say 'plot file ' filename ' saved on disk without plotting' */
end /*end of do while queued>0*/
end /*end of otherwise part of select clause*/

endif /*end of entire large select clause*/

END /*END OF IF QUEUED>0 THEN DO AT TOP OF ROUTINE*/

/******************** 3800 FILES and 3820 FILES ********************/

call IsThereFile 'IBM 3800/3820', 'TAG' ADMIMAGE A'
if queued() > 0 then do
dropbuf
say
say 'Please choose an IBM printer:'
say
say ' 1) IBM MODEL 3800'
say ' 2) IBM MODEL 3820'
say
do until (opt=1 | opt=2 | opt=3)
    say 'Enter 1 or 2:'
pull opt
end
if opt=3 then do /*hubler uses option 3 to bail out*/
call exit_exec -3800
end
if opt=1 then do
say
say 'Your graphic data is being routed to the 3800 printer.'
38XX'A' OPT
end
else do
say
say 'Which IBM 3820 printer should the output be sent to?'
say 'Enter 1, 2, 3, 4, 5 or 6:' /*choose location*/
do until (loc=1 | loc=2 | loc=3 | loc=4 | loc=5 | loc=6)
pull loc
end
if loc = 1 then locname = 'RAC'
if loc = 2 then locname = '10 X 10'
if loc = 3 then locname = 'DEB 1st floor'
if loc = 4 then locname = 'DEB 3rd floor'
if loc = 5 then locname = 'IRT'
if loc = 6 then locname = 'Sverdrup Middleburg Hts. office'
say 'Your graphic data is being routed to the'
say 'IBM 3820 printer at the 'locname
say 'opt = loc+1 /* augment opt for 38XX */
38XX 'A' opt
end
/* MATRIX FILM RECORDER */
call IsThereVile
if queued() > 0 then do
  dropbuf
  say 'Your graphic data is being routed to the matrix film recorder.'
  'datesend'
  wd = workdisk
  'copyf lerc31 header 'wd' lerc31 data 'wd' lerc31 data 'wd' [replace'
  'erase lerc31 header 'workdisk
  'listfile 'laser_offline' [stack'
  set imsg off /*set to off after debugging*/
  detach 00e
  define 3800 as 00e
  'tag dev mvslercl rmtl2'
  'spool prt nohold rscs'
  CALL PRINTSTACKEDFILES ' (NOCC NOTRC LI 00)'
  detach 00e
  define printer as 00e
  set imsg on
end
/* VERSATEC FILES */
call IsThereFile 'Versatec plotter', 'file vrfdata 'workdisk
if queued() > 0 then do /*if true a file exists*/
  dropbuf
  say 'Your graphic data is being routed to the Versatec plotter.'
  randout
  erase file vrfdata workdisk
  /*
  listfile 'file vrfout 'workdisk' [stack'
  set imsg off
  detach 00e
  define 3800 as 00e
  select
    when versatec = "VERS42" then do
    'tag dev prt mvslercl ud'
    end
    when versatec = "VERS11" then do

'tag dev prt mvalercl ul0'
end
when versatec = "VERSIIVU" then do
' tag dev prt mvalercl ul0'
end
otherwise do
say '****Invalid Versatec plotter - 'versatec'****'
end
end
spool prt nohold rscs
call PrintStackedFiles ' (nocc notrc li 00)'
detach 00e
define printer as 00e
set imaa on
end

****** DETECTION OF FILE ERROR ******

set emaa off
state file f00f001 a
if rc=0 then do
say
say 'The plot failed because of invalid options in PLOTPARM DATA'
type file f00f001 a
erase file f00f001 a
end

'state' fname ' module a' /*offer user option of deleting*/
if rc = 0 & genmodule then do /*source file load module*/
say;say;say
say 'A file -- ' fname ' MODULE -- has been created on your A disk'
say 'Would you like to have it deleted to conserve disk space? ' 
say 'Reply Y to have the module deleted'
say ' or just hit enter to keep the file'
parse upper pull delete_response
if delete_response = 'Y' then do
'erase' fname ' module a'
end
'erase load map'
end
call scratch_disk workdisk /*now scratch temporary minidisk*/

set emaa on

****** RESTORE PRINTER AND TERMINAL ENVIRONMENTS PRIOR TO ISSCO ******

set cmstype ht
'filedef * clear' /*we are done - clear all filedefs*/
call RestorePrinterEnvironment /*reset virtual printer settings*/
call RestoreTerminalEnvironment
set blip blip_state
set lortb, loader_tables
set msg on
set wng on
say 'return_code value is ' return_code
erase vscr tmp
set cmstype rt
call exit_exec return_code

****** SUBROUTINE: Access_Interlink ************

Access_Interlink: procedure expose we_linked_to_nft nftmode
set cmstype ht

290
we_linked_to_nft=0
state nft module  /*see whether nft software is linked*/
if rc = 0 then do
  we_linked_to_nft = 1
  call qvirt /*find an unused address for the minidisk*/
  dsdk-result
call qdisk /*find the next available filemode letter*/
nftmode = result
link decrtr 200 dsdk rr /*link to nft minidisk*/
  access dsdk nftmode
end
set cmstype rt
return

/*** SUBROUTINE: SaveTerminalEnvironment ***/

SaveTerminalEnvironment: procedure expose terminal_environment
  /* save the current terminal environment in <terminal_environment> */
terminal_environment = ''
makebuf
query terminal '+ (stack'
do while queued() > 0
  pull tmp
  terminal_environment = terminal_environment ',/ tmp
end
terminal_environment = substr(termnal_environment,2)
dropbuf
return

/*** SUBROUTINE: RestoreTerminalEnvironment ***/

RestoreTerminalEnvironment: procedure expose terminal_environment
  /* restore the terminal environment saved in <terminal_environment> */
do while len_termnal_environment() > 0
  i = index(termnal_environment,',')
  if i = 0 then i = length(termnai_environment) + 1
  strg = left(termnal_environment,i-1)
  if length(strg) > 0 then do
    set emsg off
    terminal strg
    set emsg on
  end
  terminal_environment = substr(termnal_environment,i+1)
end
return

/*** SUBROUTINE: IsThereFile ***/

IsThereFile: procedure /*This subroutine has been commented */
  /*to the point that it is merely a * /
  /*return. We are now sending directly*/
  /*to the device and not allowing the*/
  /*the user the choice to store file on*/
  /*disk*/
arg device, filename
makebuf
set emsg off
setlist file filename +(stack' /*look for the given file name*/
temp = rc
set emsg on
dropbuf
if temp = 0 then do /*the file exists*/
  VMFCLEAR
  say
  say
  say 'There are graphic data file(s) for the offline device
  say 'Do you want to send them to the device? (Y/N)'
pull ans
/* if ans = 'Y' then do /*/  /* put filename on stack and return */  listfile filename '(stack' /* temporarily while within the loop */  return  /* end */ */

/* say 'Do you want to delete them? (Y/N)' pull ans /* after file has been output pull */ if ans = 'Y' then do /*/  /* name off stack and delete files */  listfile filename '(stack' do while queued() > 0  pull name  erase name  return  /* end */ */  /* end of if ans=y */ return  /* end of if temp=0 */

/*** SUBROUTINE: SavePrinterEnvironment ***/

SavePrinterEnvironment: procedure expose tag_text spool_text  /* copied from <print38 exec> */ makebuf  before = queued() execlo '** cp '(' fifo string query virtual ur pull response parse var response device . after = queued() do while device ^= 'PRT' & after > before  pull response parse var response device . after = after - 1 end if after = before then do dropbuf say 'virtual printer missing *error*' call exit_exec -99 end parse var response . . prt.cl prt.cont prt.hold 'COPY' prt.copy . pull . prt.for_to prt.whom . dropbuf makebuf  /* get tag text */ rest = '' tag_text = '' execlo '** cp '(' fifo string tag query dev prt pull . . rest if rest ^= 'NOT SET' then pull tag_text  if prt.for_to ^= 'TO' then spool_text = prt.whom prt.cont copy prt.copy else spool_text = system prt.cont copy prt.copy dropbuf return

/**** SUBROUTINE: RestorePrinterEnvironment *****/

RestorePrinterEnvironment: procedure expose tag_text spool_text  /* copied from print38 */ tag dev prt tag_text spool prt spool_text return

/**** SUBROUTINE: Scratch_tdisk *****/
scratch_tdisk: procedure
/*used if compile or link goes wrong*/
arg workdisk_mode
/*also used at end of program*/

/*say 'scratch tdisk routine entered & workdisk is workdisk_mode*/

'q disk stack'
/*put disks on stack*/
pull
/*pull off header*/
do while queued()>0
   pull . cuuadd mod .
   if (mod - workdisk_mode) then do
      set cmstype ht
      'release ' workdisk_mode' (det'
      /*release mode and detach cuu*/
      set cmstype rt
   end
end
return

/** SUBROUTINE: PrintStackedFiles ***/

PrintStackedFiles: procedure
/* stack has queued filenames, print them all */
arg print_opts

do while queued() > 0
   pull filename
   /*say 'the filename stacked is ' filename */
   print filename print_opts
   erase filename
end
return

/** SUBROUTINE: TestPrintStackedFiles ***/

TestPrintStackedFiles: procedure
/* stack has queued filenames, print them all */
arg print_opts

so while queued() > 0
   pull filename
   say 'the filename stacked is ' filename
   print filename print_opts
   erase filename
end
return

******* HELP SCREEN SUBROUTINE **************/

Helpuser: procedure
VMFCLEAR
say 'This exec is for use in compiling and running your'
say 'DISSPLA program'
say
say 'ENTER:'
say
say 'DISSPLA fname (option'
say
say 'at the CMS prompt'
say
say 'where fname is the filename of your FORTRAN source program'
say 'and option is one of the following output devices:'
say
say 'TEK - a Tektronix model terminal'
say 'VERS42 - the 42 inch Versatec plotter'
say 'VERS11 - the 11 inch Versatec plotter'
say 'VERS11VU - the Versatec viewgraph plotter'
say 'GKS - a GKS standard file'
say 'TEST - to use the library of test plots supplied by CA-ISSCO'
say 'TXTLIB - to add a txtlib'
say 'NOPLOT - to create plot file without sending to plot device'
return
GetOptions: procedure expose versatec tektronx gksfile testing,
    tlib noplot cryoflag

arg options

    versatec = ''
    tektronx = ''
    gksfile = ''
    testing = ''
    tlib = ''
    noplot = 0
    cryoflag = 0 /*cryoflag added for use by Glenn Cowgill at Analex*/

do i = 1 until i > words(options)
    option.i = word(options, i)
end

if option.1 = '' or option.1 = '?' or option.1 = 'HELP' then do
    vmfclear
    SAY;SAY;SAY;SAY;SAY
    SAY ' Some options may be required'
    CP SLEEP 1 SEC
    call Helpuser
    say;say;say 'Would you like to continue?'
    say
    say ' Enter Q to quit the exec'
    say ' or Enter O to supply the options'
    say ' or hit ENTER to continue'
parse upper pull restart_response
select;
    when restart_response = 'Q' then do
        call exit_exec -92 /*exit exec so user can restart*/
    end
    when restart_response = 'O' then do
        say 'Please type all desired options and hit ENTER'
        parse upper pull options
            do i = 1 until i > words(options)
                option.i = word(options, i)
            end
    end
    otherwise do /*do nothing - continue with exec*/
end /*end of if options.1 = '?'*/

end /*end of select clause*/
end /*end of if options.1 = '?'*/

do j = 1 to words(options)
call Identify_option option.j
end /*end of Identify_option loop*/

return

Identify_option: procedure expose versatec tektronx gksfile testing,
    tlib noplot cryoflag

arg option

/*y 'option passed to identify subroutine is ' option*/

select;
    when abbrev('TEK',option) then
tektronx = TEK
    when abbrev('VERS11',option,4) then /*chars 'VERS' will*/
        versatec=VERS11 /*default to 2552 */
    when abbrev('VERS42',option,4) then
        versatec = VERS42
when abbrev('VERS11VU', option) then  
  versatec = VERS11VU
when abbrev('GKS', option) then  
  gksfile = YES
when abbrev('TEST', option) then  
  testing = YES
when abbrev('TXTLIB', option) then  
  tlib = YES
when abbrev('NOPLOT', option) then  
  noplot = 1
when abbrev('CRYO', option) then  
  cryoflag = 1
when abbrev('FAKE_OPTION', option) then  
  nop
otherwise  
  do
    say 'unrecognized option: ' option
    'cp sleep 3 sec'
    call Helpuser
    call exit_exec -94  
  end  
/*end of select clause*/
*/y versatec tektronx gksfile testing tlib */

return

GetFileAge: procedure  /*procedure to determine if compilation*/
arg fname  /*of user program is needed*/
makebuf
bufno = rc
set emsg off
'listfile' fname '* { date stack'
rcode = rc
set emsg on
if rcode = 0 then  
  do
    parse pull fname . . . . mo'/da'/yr hr':mn':sc
    tim = sc + mn*100 + hr*10000 + da*1.0e+6 + mo*1.0e+8 + yr*1.0e+10
  end
else  
  tim = -1
dropbuf bufno
return tim

***** ISSUE ISSCO-REQUIRED AND SITE-SPECIFIC FILEDEFS *****/
/*WARNING WARNING WARNING*******************************************/
/* IF THE USER HAS HIS OWN COPY OF PLOTPARM OR IF SOME OTHER*/
/* MINIDISK HAS ONE THEN RC>0 AND PLOTPARM WILL NOT BE FILEDEFED*/
/* VERSATEC SOFTWARE WILL NOT BE RUN AND EXEC WILL END WITH AN ERROR*/
/* USER MUST NOT HAVE HIS OWN PLOTPARM FILE*/

file_definitions: procedure expose versatec workdisk laser_offline

/*say 'versatec value is ' *****versatec***** /*for debugging*/
if versatec = ' ' THEN DO
  makebuf
  bufno = rc
  'LISTFILE' VERS11 PLOTPARM '* { LIFO AL'
  rclist = rc
  nfiles = queued()
  /*say rclist ' IS THE RETURN CODE*/
  /*say nfiles ' IS THE NUMBER OF TIMES IT WAS FOUND */
  say
  if rclist = 0  
    then do queued()
      parse pull . . fmode
      /*say fmode ' IS THE FILEMODE OF VERS11 PLOTPARM */
end
dropbufbufno
if nfiles > 0
then fm = substr(fmode, l, l)
/*SAY FM IS THE FILEMODE DERIVED FOR VERS11*/

if versatec = 'VERS11' then do
 'FILEDEF PLOTPARM DISK VERS11 PLOTPARM' FM
end
if versatec = 'VERS11VU' then do
 'FILEDEF PLOTPARM DISK VERSVU PLOTPARM' FM
end
if versatec = 'VERS42' then do
 'FILEDEF PLOTPARM DISK VERS42 PLOTPARM' FM
end

'FILEDEF VRFDATA DISK FILE VRFDATA' WORKDISK' (XTENT 65535'
'FILEDEF VRFOUT DISK FILE VRFOUT' WORKDISK
'FILEDEF 23 DISK VSDR TMP WORKDISK' (XTENT 5000) /*COMMENTED
by hubler to solve fortran traceback with A Lain problem*/
'FILEDEF 59 disk vers diag ' WORKDISK' (recfm f lrecl 132 blksize 132'
laser_offline = 'lerc31 data 'workdisk /*QCR film recorder file*/
laser_header = 'lerc31 header ' workdisk /*QCR header info*/
b_offline = 'lerc32 data 'workdisk
/* qms_offline = 'lerc50 data 'workdisk */
/* set emsg off */
/* erase laser_offline */
/* erase b_offline */
/* erase qms_offline */
set emsg on

'FILEDEF 5 term (recfm f lrecl 80 blksize 80'
'FILEDEF 6 term (recfm f lrecl 132 blksize 132'*)

'FILEDEF 31 disk' laser_offline
'FILEDEF 68 disk' laser_header
'FILEDEF 32 disk' b_offline
'FILEDEF 50 disk qms_offline */
return

****** SUBROUTINE: EXIT_EXEC ***************

/* exit can have following return codes
0 - normal completion
-92 - no options provided so user chose to exit
-93 - FORTRAN source file not found
-94 - invalid option
-99 - virtual printer missing
9,12,or 16 - VSFORTAN or link edit errors
-300 - my own personal exit from 3800 part of exec (used to debug)
*/
exit_exec: procedure
arg exit_code
exit exit_code
return

296
/* plotq - print graphic stuff to the qms */
arg file_name '/' destination .
default_name = "std0001 data a"
if length(file_name) = 0 then file_name = default_name
rcode = rc
if rcode <> 0 then exit rcode

/* tellagraf doesn't switch back to portrait orientation
so we must issue the proper qms commands to do that */
execio 1 diskw file_name " ( string ^PY^-"
execio 1 diskw file_name " ( string ^IOL^PN ^-"

/* determine the destination qms printer */
select:
  WHEN DESTINATION = 'ANALEX' THEN DESTINATION_NODE = RMT34
  WHEN DESTINATION = 'RAC' THEN DESTINATION_NODE = RMT7
  WHEN DESTINATION = 'ERB' THEN DESTINATION_NODE = RMT10
  WHEN DESTINATION = 'DEB' THEN DESTINATION_NODE = RMT9
otherwise do
  dropbuf
done = "false"
do while done = "false"
say "Which QMS printer would you like to have your output sent to?"
  SAY " 1) ANALEX"
  SAY " 2) RAC"
  SAY " 3) ERB"
  SAY " 4) DEB"
say "Enter the number of your choice:"
pull choice parse upper
select:
  when choice = 1 then do
    DESTINATION_NODE = RMT34
done = "true"
  end
  WHEN CHOICE = 2 THEN DO
    destination_node = RMT7
done = "true"
  end
  WHEN CHOICE = 3 THEN DO
    destination_node = RMT10
done = "true"
  end
  WHEN CHOICE = 4 THEN DO
    destination_node = RMT9
done = "true"
  end
otherwise
  say "That is not an acceptable choice. Try again."
end
end
tag dev prt mvslercl destination_node
spool prt nohold rscs
print file_name ''' nocc notrc 11 00
erase file_name
VM Exec DOECLPLOT

/* THIS EXECUTES PROGRAM ECLPLOT TO PRODUCE PLOTS FROM ECL PLOTFILE*/
/* WHICH CAME FROM CRAY. NAMED -- SOLAPLOT CARDS -- */
/* THE FILE FN FT IS A BINARY FILE WITH LRECL=2024 */
/* COPY THIS FILE TO B-DISK USING COPYFILE*/
/* COPYFILE DOECLPLOT EXEC A DOECLPLOT EXEC B */
/
ARG FN FT FM .
IF FN = ' ' THEN
   DO
      SAY "ENTER FILENAME, FILETYPE AND FILEMODE"
      PULL FN FT FM
   END
IF FT = ' ' THEN
   DO
      SAY "ENTER FILENAME, FILETYPE AND FILEMODE"
      PULL FN FT FM
   END
SETUP CONVDISK
SETUP FTN
GRAPH3D
FILEDEF 08 CLEAR
FILEDEF 09 CLEAR
FILEDEF 59 CLEAR
FILEDEF 08 DISK ECLPLOT INPTECHO
FILEDEF 09 DISK FN FT FM
FILEDEF 59 DISK ECLPLOT DEBUGOUT ' (LRECL 132'
LOAD ECLGRAPH ' (CLEAR NOMAP ORIGIN 30000'
INCLUDE CHCFTR ' (NOMAP'
START
GTERM
APPENDIX E

CryoTran Program Listings

Part VI  CRAY Script File to Execute SOLA ECLIPSE
#!/SOLA ECLIPSE SCRIPT FILE FOR UNICOS (BOURNE SHELL)

# VERSION 1.0 05/23/89 BY Glenn Cowgill

# SOLA ECLIPSE ** SOLA ECLIPSE ** SOLA ECLIPSE ** SOLA ECLIPSE **

# The invocation of the script is as follows:
# solaecl filename
# where filename contains your input sola deck
# which is on CRAY in the users root directory.

# The exit status is as follows:
# 0 - Successful sola run
# 1 - Unable to create temporary directory or input file does not exist
# 2 - Loading errors from segldr
# 3 - Errors in the execution of sola_eclipse

# The script variables are as follows:
# exe_dir - directory where sola libraries, (solaxxxx.o), exist.
# solaecl.o, solaheat.o, solatherm.o
# user_dir - this is directory from where the job is submitted
# root_name - this is filename prefix of the input filename
# exe_dlr="/space/cryollb"
# user_dir= pwd
# root_name= basename "$I"

# banner SOLA
# banner ECLIPSE
# echo This is the Hochstein Version of SOLA ECLIPSE.
# echo
#
# Let's check to see if the input file exists
#
# IF [ ! -f "$I" ]
# then
# else
# echo file "$I" does not exist in $user_dir.
# echo Try again.
# exit 0
#
# echo Using "pwd" as the temporary directory for all user files.
# echo "user directory is" "$user_dir"
#
# echo "The model is file $1 $2"
#
# fetch model -t'fn='$I',ft='$2',addr=191'
# the fetch command is not in this procedure (shell)
# the shell that the user makes up should fetch the model
# and invoke this shell to execute sola,
# the user then submits this shell to the CRAY
#

#### generate and compile main program

cat >malnpgm.f << EOFM
    program solaecl
    call mainnp
    end
EOFM

cft77 malnpgm.f

#### RUN PROGRAM

# Linking main run

cat >libdir<< libend
lib= $(exe_dlr)/solaecl.o
lib= $(exe_dlr)/solaheat.o
lib= $(exe_dlr)/solatherm.o
libend

segldr -k -Ms -o xqtecl mainnpgm.o libdir >>""$<user_dir>".err

IF [ $? -NE 0 ]
then

echo Unable to link your sola run with the sola libraries.
echo Contact Dave Chato, 216-433-2845
echo your output is in file: "${user_dir}".solarun
cat "$(root_dir).err" >> "${user_dir}".solarun
dispose "${user_dir}".solarun
exit 2
else
echo " begin execution of sola eclipse"
xqtecl < $1 > "${user_dir}".solarun
fi
if
#
#
if Your exit status from the sola run is "$?"
then
echo Errors detected! Exit status = 3
dispose "${user_dir}".solarun for your output.
dispose "${user_dir}".history -t'fn=sola,ft=history'
dispose "${user_dir}".bugfile -t'fn=sola,ft=bugfile'
mv core "${user_dir}".cor
dispose "${user_dir}".solarun
dispose "${user_dir}".cor
dispose "${user_dir}".solarun
exit 3
else
#### Successful run
#### dispose all files to front end
echo " Successful sola eclipse run."
info output is in file: "${user_dir}".solarun
dispose "${user_dir}".solarun
dispose fort.9 -t'fn=sola,ft=plotfile'
## save restart file in Home directory filename= solarestart
cat fort.11 >> $HOME/solarestart
rm mainpgm.f mainpgm.o xqtecl libdir model
rm "$(root_dir).err" "$(user_dir)".solarun fort.7 fort.9 fort.11
exit 0
fi

APPENDIX E

CryoTran Program Listings

Part VII  CRAY Script File to Execute CSAM
#CSAM  SCRIPT FILE FOR UNICOS (BOURNE SHELL)
#VERSION 1.0 05/23/89 BY Glenn Cowgill
# CSAM  ** CSAM  ** CSAM  ** CSAM
#
# The invocation of the script is as follows:
# crcsam filename
# where filename contains your input csam deck
# which is on CRAY in the users root directory.
#
# The exit status is as follows:
# 0 - Successful csam run
# 1 - Unable to create temporary directory or input file does not exist
# 2 - Loading errors from segldr
# 3 - Errors in the execution of csam/sinda
#
# The script variables are as follows:
# csam_dir  - directory where csam library, (csam.o), exists.
# user_dir  - this is directory from where the job is submitted
#
# csam_dir="/space/cryollb"
# user_dir= pwd
# banner CSAM
# banner CSAM
# echo " This is CSAM"
# echo Using "pwd" as the temporary directory for all user files.
# echo " user directory is" $user_dir
# echo " generate the main program"
cat >mainpgm.f << EOFM
program osam
call mpcsam
end
EOFM
echo " compile the main program"
cft77 mainpgm.f
##### RUN PROGRAM
echo " Linking main run"
cat >libdir<< libend
llb= ${csam_dir}/csam.o
libend
#
# echo " segldr"
segldr -k -M,s -o xqtcsam mainpgm.o libdir >>"${user_dir}".err
if [ $? -NE 0 ]
# Linking errors! Exit status = 2
then
  echo Unable to link your csam run with the csam library.
  echo Contact Dave Chato, 216-433-2843
  echo Your output is in file: "$user_dir".csamrun
  cat "$user_dir".err >> "$user_dir".csamrun
  dispose "$user_dir".csamrun
  exit 2
else
  echo " begin execution of csam"
xqtcsam < $1 > "${user_dir}".csamrun
#####
#
fi;
#
echo Your exit status from the csam run is "$?"
if [ $? -NE 0 ]
# Errors detected! Exit status = 3
then
  echo Errors were detected in the csam run.
  echo Check file: "$user_dir".csamrun for your output.
  dispose "$user_dir".csamrun
  exit 3
fi;

else

#### Successful run
#### dispose all files to front end
echo " Successful csam run."

# dispose output files

echo Output is in file: "$user_dir".csamrun
dispose "$user_dir".csamrun

## dispose plots -fBB -t'fn=sola,ft=plotfile'

cd "$user_dir"

rm mainpgm.f mainpgm.o sqrtcsam libdir model

rm "$user_dir".err "$user_dir".csamrun

exit 0

fl
REFERENCES


### Table 1

**Units Used in CryoTran**

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Symbol</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td></td>
<td>Hrs</td>
</tr>
<tr>
<td>Time Increment</td>
<td></td>
<td>Hrs</td>
</tr>
<tr>
<td>All Temperatures</td>
<td>T</td>
<td>°F or °R</td>
</tr>
<tr>
<td>All Lengths</td>
<td>l</td>
<td>in</td>
</tr>
<tr>
<td>Areas</td>
<td>A</td>
<td>in²</td>
</tr>
<tr>
<td>Internal heat source (heat generation in a node or on a surface)</td>
<td>Q</td>
<td>BTU/hr</td>
</tr>
<tr>
<td>Capacitance of each diffusion node</td>
<td>C=C&lt;sub&gt;p&lt;/sub&gt;*ρ</td>
<td>BTU/°F</td>
</tr>
<tr>
<td>Conductor Values</td>
<td>G</td>
<td>BTU/hr-°F</td>
</tr>
<tr>
<td>Conduction Conductor</td>
<td>G=A*k/l</td>
<td></td>
</tr>
<tr>
<td>Convection Conductor</td>
<td>G=Ah</td>
<td></td>
</tr>
<tr>
<td>Radiation Conductor</td>
<td>G=σ<em>e</em>f*A</td>
<td></td>
</tr>
<tr>
<td>(where: cross section area of heat flow)</td>
<td>A</td>
<td>in²</td>
</tr>
<tr>
<td>Thermal conductivity of material</td>
<td>k</td>
<td>BTU/hr-in-°F</td>
</tr>
<tr>
<td>Heat capacity of material</td>
<td>C&lt;sub&gt;p&lt;/sub&gt;</td>
<td>BTU/lb-°F</td>
</tr>
<tr>
<td>Density of material</td>
<td>ρ</td>
<td>lb/in³</td>
</tr>
<tr>
<td>Length of conductor path</td>
<td>l</td>
<td>in</td>
</tr>
<tr>
<td>Film coefficient</td>
<td>h</td>
<td>BTU/hr-ft²-°F</td>
</tr>
<tr>
<td>Stefan-Boltzman Constant</td>
<td>σ=0.173x10&lt;sup&gt;8&lt;/sup&gt;</td>
<td>BTU/hr-ft³-°R⁴</td>
</tr>
<tr>
<td>Surface emissivity</td>
<td>ε</td>
<td></td>
</tr>
<tr>
<td>View Factor</td>
<td>f</td>
<td>0&lt;f&lt;1</td>
</tr>
</tbody>
</table>
## Table 2

### Fortran Files and Units

**FORTRAN units and file names used in CryoTran:**

<table>
<thead>
<tr>
<th>Logical Unit No.</th>
<th>File Name (alias)</th>
<th>Description</th>
<th>Status at end of Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
<td>MATERIAL DBASE M</td>
<td>Material properties database</td>
<td>CRYOLIB M disk</td>
</tr>
<tr>
<td>05</td>
<td></td>
<td>Standard input</td>
<td>User A disk</td>
</tr>
<tr>
<td>06</td>
<td></td>
<td>Standard output</td>
<td>User A disk</td>
</tr>
<tr>
<td>09</td>
<td>CRYOTRAN INPUTEKO</td>
<td>Echo of input typed in by user</td>
<td>User A disk</td>
</tr>
<tr>
<td>10</td>
<td>CRYOTRAN MODEL</td>
<td>Model output</td>
<td>User A disk</td>
</tr>
<tr>
<td>17</td>
<td>PROGRAM OUTPUT</td>
<td>Output of program executed interactively on VM</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>H2 TABLE M</td>
<td>H2 property data</td>
<td>CRYOLIB M disk</td>
</tr>
<tr>
<td>26</td>
<td>O2 TABLE M</td>
<td>O2 property data</td>
<td>CRYOLIB M disk</td>
</tr>
<tr>
<td>27</td>
<td>N2 TABLE M</td>
<td>N2 property data</td>
<td>CRYOLIB M disk</td>
</tr>
<tr>
<td>35</td>
<td>Scratch file</td>
<td>Used in sub READAL</td>
<td>Gone</td>
</tr>
<tr>
<td>36</td>
<td>Scratch file</td>
<td>Used in sub INSERT</td>
<td>Gone</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Used in sub INSERT1</td>
<td>Gone</td>
</tr>
</tbody>
</table>

**FORTRAN units used by SINDA on the Cray:**

<table>
<thead>
<tr>
<th>Logical Unit No.</th>
<th>File Name (alias)</th>
<th>Description</th>
<th>Status at end of Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>05</td>
<td></td>
<td>Standard input</td>
<td></td>
</tr>
<tr>
<td>06</td>
<td></td>
<td>Standard output</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>LUT1</td>
<td>Actual/relative dictionary</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>LB3D</td>
<td>Input data after the SINDA preprocessor has executed</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>LB4P</td>
<td>5 FORTRAN programs generated by the SINDA preprocessor</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>MIN</td>
<td>Matrix input tape</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>LUT3</td>
<td>Parameters runs data</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>MOUT</td>
<td>Matrix output unit</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>LUT7</td>
<td>Recall data file</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>STAPE</td>
<td>Store data file</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>INTERN</td>
<td>Prepro scratch unit</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>NEDIN</td>
<td>EDIT input</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>nedout</td>
<td>EDIT output</td>
<td></td>
</tr>
</tbody>
</table>


CRYOTRAN—AN INTEGRATED CRYOGENIC FLUID SYSTEM MODEL

OBJECTIVE:
DEVELOP AN INTEGRATED ANALYTICAL MODEL OF CRYOGENIC LIQUID TRANSFER SYSTEMS DESIGNED FOR THE LOW GRAVITY ENVIRONMENT OF SPACE

FIGURE 1

INDIVIDUAL CRYOGENIC FLUID ANALYTICAL MODELS
FLOW CHART OF CRYOTRAN

MAIN PROGRAM

READ PROBLEM TYPE

PREPROCESSOR

READ INPUT

EXECUTE AN INTERACTIVE ANALYSIS WITHIN PREPROCESSOR

GENERATE A FILE TO EXECUTE AN ANALYSIS PROGRAM ON CRAY

GENERATE SINDA MODEL

PROPERTIES FROM DATA BASE

READ INPUT

EXECUTE AN INTERACTIVE ANALYSIS WITHIN PREPROCESSOR

GENERATE A FILE TO EXECUTE AN ANALYSIS PROGRAM ON CRAY

GENERATE SINDA MODEL

PROPERTIES FROM DATA BASE

PLOT GEOMETRY

FIGURE 2
FIGURE 3. - SKETCHES SHOWING SPHERICAL WEDGE.
FIGURE 4. - SKETCHES SHOWING CYLINDRICAL WEDGE.
Figure 5. - Sketches showing possible and configurations of cylindrical wedge.
SAMPLE MODEL SPHERE1

OUTSIDE BNDY NODE 1
NODE NO. 20301
T= 140.00 DEG F
CONVECTION TO SURFACE
h = 1.38889*10^6

OUTSIDE BNDY NODE 2
NODE NO. 20302
T= 440.00 DEG F
RADIATION TO SURFACE
\sigma = 2.37778*10^{-12}

SOURCE Q INTO ALL OUTER SURFACE NODES
Q = 0.0023 (BTU/(HR-IN2))

HEAT EXCHANGER NO. 1
NODE NO. 20001
T= -424.00 DEG F

HEAT EXCHANGER NO. 2
NODE NO. 20002
T= -424.00 DEG F

HEAT EXCHANGER NO. 3
NODE NO. 20003
T= -400.00 DEG F

VOLUME = 9.4 (FT^3)
\% FULL = 5.5%

Sample Model Sphere 1
PLOT NOT TO SCALE
OUTSIDE BNDY NODE 1
NODE NO. 20301
T = 23.00 DEG F
CONVECTION TO SURFACE
h = 1.92901 \times 10^{-7}

OUTSIDE BNDY NODE 2
NODE NO. 20302
T = 25.00 DEG F
CONVECTION TO SURFACE
h = 3.85802 \times 10^{-7}

SOURCE Q INTO ALL OUTER SURFACE NODES
Q = 0.0007 (BTU/(HR-IN2))

HEAT EXCHANGER NO. 1
NODE NO. 20001
T = 101.00 DEG F
ON LAYER 1 IN REGION 1
STARTING AT LEVEL 1
AND COVERING 1 NODE(S).

HEAT EXCHANGER NO. 2
NODE NO. 20002
T = 104.00 DEG F
ON LAYER 2 IN REGION 2
STARTING AT LEVEL 2
AND COVERING 2 NODE(S).

figure 8 Plot Sample of Cylinder

PLOT NOT TO SCALE
FLOW DIAGRAM OF CRYOTRAN 1

FIGURE 9-1
FLOW DIAGRAM OF CRYOTRAN 2

FIGURE 9-2
FLOW DIAGRAM OF CRYOTRAN 3

FIGURE 9-3
FLOW DIAGRAM OF CRYOTRAN

FIGURE 9-4
FLOW DIAGRAM OF CRYOTRAN 6

01
CLEARS (clear terminal screen)

02
READAL
entry READLC
entry READIN
entry READRE
(read input from terminal)

03
DOJCL (execute system tasks)

04
BLHDTRS
entry RDTITL
entry BL0TTL
entry BL1TTL
entry BL2TTL
entry BL3TTL
entry BL4TTL
entry BL5TTL
entry BL6TTL
entry BL7TTL
entry BL8TTL
entry BL9TTL
entry BLKEND
entry ENDDAT

05
Collection of routines that will be executed interactively on VM from within the system.

FIGURE 9-6
Abstract

The development of cryogenic fluid management systems for space operation is a major portion of the efforts of the Cryogenic Fluids Technology Office (CFTO) at the NASA Lewis Research Center. Analytical models are a necessary part of experimental programs which are used to verify the results of experiments and are also used as a predictor for parametric studies. The CryoTran computer program is a bridge to obtain analytical results. The object of CryoTran is to coordinate these separate analyses into an integrated framework with a user-friendly interface and a common cryogenic property database. CryoTran is an integrated software system designed to help solve a diverse set of problems involving cryogenic fluid storage and transfer in both ground and low-g environments.
**NASA Scientific and Technical Document Availability Authorization**

To be initiated by the responsible NASA Project Officer, Technical Monitor, or other appropriate NASA official for all presentations, reports, papers, and proceedings that contain scientific and technical information. Explanations are on the back of this form and are presented in greater detail in NMI 2230.1B, "NASA Scientific and Technical Document Availability Authorization."

### I. DOCUMENT/PROJECT IDENTIFICATION

- **Title:** Cryotran User Manual
- **Author:** G. Cowgill, D. Chato, E. Saad
- **Originating NIAA Organization:** Lewis Research Center
- **Contract/Grant/Interagency/Project Number(s):** 591-23
- **Document Number(s):** TM02468
- **Date:**

### II. AVAILABILITY CATEGORY

- **Security Classification:**
  - [ ] Secret
  - [ ] Secret RD
  - [ ] Confidential
  - [ ] Confidential RD
  - [ ] Unclassified
- **FROM:**
  - [ ] SDD
  - [ ] EAR
  - [ ] Other - See Section III
- **Document disclosing an invention:**
  - [ ] Other - See Section III

Publicly Available Document

- [ ] Publicly available documents must be unclassified and may not be export-controlled nor restricted distribution documents.

### III. SPECIAL CONDITIONS

Check one or more of the applicable boxes as the basis for special restricted distribution if the "Other" box under NASA Restricted Distribution Document in Section II is checked.

- [ ] Foreign government information
- [ ] Preliminary information
- [ ] Commercial product test or evaluation results
- [ ] Information subject to special contract provision
- [ ] U.S. Government agencies and U.S. Government agencies contractors only
- [ ] NASA personnel and NASA contractors only
- [ ] U.S. Government agencies only
- [ ] Available only with approval of issuing office

### IV. BLANKET RELEASE (OPTIONAL)

All documents issued under the following contract/grant/project number may be processed as checked in Sections II and III.

The blanket release authorization granted is:

- [ ] Restricted - Future documents must have individual availability authorizations.
- [ ] Modified - Limitations for all documents processed in the STI system under the blanket release should be changed to conform to blocks as checked in Section II.

### V. PROJECT OFFICER/TECHNICAL MONITOR

- **Name:** Joseph D. Gaby, Jr.
- **Title:** Project Officer/Technical Monitor
- **Office Code:** 6210
- **Signature:**
- **Date Signed:** 2-19-89

### VI. PROGRAM OFFICE REVIEW

- **Name:** MARIA LOPEZ - TELLADO
- **Title:** Program Officer
- **Office Code:**
- **Signature:**
- **Date:** 1/9/90

### VII. INTERNATIONAL AFFAIRS DIVISION REVIEW

- [ ] Foreign publication/presentation approved.
- [ ] Export controlled limitation is approved.
- [ ] Export controlled limitation is not applicable.
- [ ] The following Export controlled limitation (ITAR/EAR) is assigned to this document:

### VIII. EXPIRATION OF REVIEW TIME

The document is being released in accordance with the availability category and limitations checked in Section II since no objection was received from the Program Office within 15 days of submission, as specified by NMI 2230.1B.

### IX. DOCUMENTS DISCLOSING AN INVENTION

a. **This document may be released on**

   - **DATE:**

   - [ ] Installation Patent or Intellectual Property Counsel

b. **The document was processed on**

   - **DATE:** in accordance with Section II.

### X. DISPOSITION

- [ ] Printed or reproducible copy of document enclosed.
- [ ] Abstract or standard bibliographic page enclosed. The issuing activity should provide a copy of the document, when completed, to the NASA Scientific and Technical Information Facility at the above listed address.