RICE UNIVERSITY

AN INTERACTIVE COMPUTER SOLUTION
FOR THERMAL SYSTEMS

by

FRANK FISHER REYNOLDS III

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

MASTER OF SCIENCE

APPROVED, THESIS COMMITTEE:

William F. Walker,
Thesis Director and Professor
of Mechanical Engineering

Alan J. Chapman,
Professor of Mechanical
Engineering

F. A. Wierum,
Professor of Mechanical
Engineering

Houston, Texas

May 1984

3 1272 00289 0851
An Interactive Computer Solution for Thermal Systems

by

F. Fisher Reynolds III

ABSTRACT

The Thermal Analysis System is an attempt to automate the analysis of a closed thermodynamic system. A first law analysis is performed on each device in the system using the properties initially entered by the user at selected states.

The known data for a problem to be solved is entered via interactive prompts. The program then manipulates the data and attempts to solve the system. If enough data is not known to solve the system completely, all known data is printed out and an error message will appear.
TABLE OF CONTENTS

Introduction .................................................. 1
General Outline of Program ................................. 2
Detailed Usage of Program ................................. 5
Program Modeling ............................................ 39
Sample Runs and Results ................................... 61
Conclusions .................................................. 74
Appendix A: Sample Network Description ............... 75
Appendix B: Sample Network Data Entry .................. 76
Appendix C: Detailed Description of Network Devices .. 82
Appendix D: File Layouts .................................. 91
Appendix E: Units for the Various Properties .......... 102
Appendix F: Listing and Description of Subroutines ... 103
Appendix G: Sample Output for Subroutine FIXNOD ...... 134
Appendix H: Sample Output for Subroutine ORDER ...... 136
Appendix I: Adding New Devices to the Program ...... 138
Glossary of Terms .......................................... 139
INTRODUCTION

The Thermal Analysis System is a set of programs used to solve a closed thermodynamic system. The program has been set up to work with steam cycles only, but may be easily expanded to handle any type of thermodynamic cycle with multiple working fluids.

The entire analysis system is broken down into two main sections: input routines, to facilitate the entry of the system network and any known properties, and the processing routine which does the actual analysis. Additional programs are provided to print out the input.data as well as the final processed results. All of these programs are controlled by a main menu.
This manual describes the implementation of the Thermodynamic Systems Analyzer Program. A general outline of menu execution is presented below. Each menu item is then briefly described on the pages that follow. A detailed explanation of each menu item is then presented, complete with sample entries and screen displays. Finally, an in depth discussion of the actual program as well as descriptions of each network device may be found in the various appendices.

GENERAL OUTLINE

(1) Sign on to TSO using the appropriate procedure.

(2) Call up the Main Processing Menu by typing in EX THERMAL(MENU).

(3) Select option 1 to enter the initial network and property data

(4) Select option 3 to begin the processing of the data.

(5) If any errors occurred, use option 1 to enter the corrections.

(6) Save the session data if desired by using option 'S'.

(7) When the session is no longer needed, remove it from the disk using option 'R'.

(8) Select option 'Q' to return to TSO.
OUTLINE OF PROGRAM MENUS

MAIN MENU
A. Add/Change/Delete Data Items
ADD/CHANGE/DELETE MENU
1. Network Add/Change/Delete
NETWORK MAIN MENU
   a. Add New Devices to the System
      NETWORK INPUT MENU
      1. Steam Turbine
      2. Steam Generator
      3. Condenser
      4. Pump
      5. Feedwater Heater
      6. Branch
      7. Mixer
      8. Throttle
      9. 'T' Connection
     10. Dummy Node
     11. Return to Network Main Menu
   b. Change Device Data
      NETWORK CHANGE MENU
      1. Change Data for a Given Node
      2. Change Data for a Given Record Number
      3. Return to Network Main Menu
   c. Delete a Device from the System
      NETWORK DELETE MENU
      1. Delete a Given Node
      2. Delete a Given Record Number
      3. Return to Network Main Menu
   d. Display All Records in File
   e. Quit

2. Property Add/Change/Delete
PROPERTY MAIN MENU
   a. Add New Properties to the System
   b. Change any Existing Properties
      PROPERTY CHANGE MENU
      1. Change Data for a Given State
      2. Change Data for a Given Record Number
      3. Return to Property Main Menu
   c. Delete any State Records
      PROPERTY DELETE MENU
      1. Delete a Given State
      2. Delete a Given Record Number
      3. Return to Property Main Menu
   d. Display all Records in File
   e. Quit

3. Return to Main Menu
B. Print Initial Data

C. Run Analysis

D. Re-Print Final Data

E. Save Session Data on Disk

F. Remove Session Data From Disk

G. Quit
DISPLAYING THE MAIN PROCESSING MENU

To display the Main Processing Menu the user must first be logged onto TSO. When the computer signals READY type the following

EX THERMAL(MENU)

and press the RETURN key. The Main Menu will then appear on the display screen.

<table>
<thead>
<tr>
<th>MAIN MENU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - ADD/CHANGE/DELETE DATA ITEMS</td>
</tr>
<tr>
<td>2 - PRINT INITIAL DATA</td>
</tr>
<tr>
<td>3 - RUN ANALYSIS</td>
</tr>
<tr>
<td>Q - QUIT</td>
</tr>
</tbody>
</table>

ENTER SELECTION .....  

Many of the options listed in the menu depend on the previous selection of another option. For example option 1 (Add/Change/Delete Initial Data) must have been performed before any other option may be selected. If one were to select option 3 (Run Analysis) before option 1, an error message would be displayed indicating that no data exists for an analysis.
MAIN MENU OPTIONS

OPTION 1  ADD/CHANGE/DELETE INITIAL DATA

This allows the initial entry of network and property data for a
given session. It may be selected at any time to create a new session
file or to add data to an existing file. It may also be used to change
or delete specific data items in an existing session file.

OPTION 2  PRINT INITIAL DATA

This option will print a hard copy of all the data entered in
option 1 above. This is especially useful in double checking the data
to be certain it was properly entered.

OPTION 3  RUN ANALYSIS

Once all the necessary data has been entered the analysis may
begin by selecting option 3. This option will first edit the data to
determine if there are any geometry errors in the network. This type of
error would arise if an invalid node connection was entered during the
entry phase. If any errors occur they are displayed on the screen and
printed out and the user is returned to the Main Menu. Corrections
should then be made using option 1 before another attempt is made to run
the analysis.

Should there be no errors during this edit phase the actual
processing of the data begins. There is no user intervention once this
step has begun. At the completion of the analysis a hard copy of all
data is printed as well as any error or warning messages that may have
occurred. Informational messages may also be printed out indicating any assumptions that the program made.

**OPTION 4  RE PRINT FINAL DATA**

This option may be selected at any time after the completion of option 3. It merely re-prints all of the processed data and error messages without having to go through the entire analysis step.

**OPTION S  SAVE DATA ON DISK**

Option 'S' will permanently save on disk all data for a particular session by submitting a batch savecard to the system. It is advisable to use this option whenever any data is entered or changed and after running the analysis (option 3).

**OPTION R  REMOVE SESSION DATA FROM DISK**

When the data for a session is no longer needed it may be deleted from the disk by selecting this option. Only files for a given session are deleted while all others are left intact. CAUTION SHOULD BE EXERCISED IN THE USE OF THIS OPTION TO PREVENT THE ACCIDENTAL DELETION OF ACTIVE FILES.

**OPTION Q  QUIT**

This option may be selected at any time. It will free all currently allocated files and return the user to TSO.

Each of the Main Menu selections will be described in more detail
in the following pages. A sample network will be used to step through the entire process for illustrative purposes. See Appendix A for a complete description and diagram of the sample network.
MAIN MENU OPTION 1 : ADD/CHANGE/DELETE INITIAL DATA

After selecting option 1 on the Main Menu the display screen is cleared and the Add/Change/Delete Menu appears:

```
ADD/CHANGE/DELETE MENU

THIS MENU IS USED TO ADD, CHANGE, OR DELETE ANY DATA FOR A GIVEN SESSION

1 - NETWORK ADD/CHANGE/DELETE
2 - PROPERTY ADD/CHANGE/DELETE
X - RETURN TO MAIN MENU

ENTER SELECTION .....
```

Figure 1

Whenever a new session is entered, the following display will appear:

```
UNITS

1 - S.I. UNITS (METRIC)
2 - BRITISH UNITS

ENTER SELECTION .....
```

Figure 2

This allows one to select the system of units desired. Once a set of units has been selected for a session, they may NOT be changed. See
Appendix E for a listing of the various units applicable to the program.

A/C/D MENU OPTION 1 - ADD, CHANGE, OR DELETE NETWORK DATA

Select option 1 to enter the network data for a new session. The following prompt will then appear at the bottom of the screen:

ENTER SESSION ID (1 - 8 CHARACTERS) ......

The session ID is the unique name given to a particular problem set to distinguish it from other problem sets that may already exist. It should be a letter (A-Z) followed by from zero (0) to seven (7) alphanumeric characters. For the sample network the session ID of TEST will be used.

Enter the session ID and press the RETURN key. The Network Main Menu will then appear on the screen:
Notice the lowercase 'n' at the bottom lefthand corner of the screen with the question mark (?) directly beneath it. The question mark represents the cursor and the 'n' indicates the length of the required response. Because there is only a single 'n', a one character response is required. This same type of indicator is used throughout the program to indicate the correct number of characters to enter for each response.

Since new data will be entered at this time select option 'A' and press the RETURN key. The Network Input Menu will then appear:
This menu is used to enter all the network devices, their efficiencies, and how they are connected to one another. If the work or heat loads for devices such as pumps or condensers are known a priori they may also be entered at this time. A detailed description of each of the devices listed in the menu may be found in Appendix C.

Refering to the sample network in Appendix A it is clear that one must enter a turbine, steam generator, pump and condenser. Begin by selecting option 1 (steam turbine). The screen will clear and the user will be prompted for the following information:
This prompt is asking for the unique number associated with the device. Notice the presence of (I-3) to the right of the prompt. This symbol used in conjunction with the three lowercase n's to the left constitute the input indicators for the prompt. These in particular indicate that an integer(I) value of length three(3) should be entered. In addition, the nodal value entered must be greater than zero.

For the turbine in the sample network, enter the nodal value of 002 and press the RETURN key. The next prompt then appears:

\[ \text{nnn} \quad \text{<----- Enter the \# of INLETS into device (I-3)} \]

\?

The user is next asked to enter the actual number of inlets into the turbine. Since the sample turbine has only one inlet enter 001 and RETURN.

\[ \text{nnn} \quad \text{<----- Enter the \# of OUTLETS from device (I-3)} \]

\?

Enter 001 to indicate that the turbine has only one outlet.

\[ \text{nnnnnnnnnn} \quad \text{<--- Enter WORK output (if known) R 10} \]

\?

The user is now prompted to enter the work output of the turbine if that information is known. If the work is unknown at this stage, enter a value of -1.0 and press the RETURN key. If the work is known apriori
it may now be entered.

Notice that the input indicators are now a series of ten lowercase n's and the code R 10. These indicate that the maximum number of characters entered may be ten but one must include one and only one decimal point. Thus we may enter from one to nine numeric digits and one decimal point. If, for example, the work output was 900.0 we may enter 900. or 900.00 but should not enter simply 900 with no decimal point.

Since we do not know the value of the turbine work for our sample network, enter a value of -1.0 and press the return key.

nnnnnnnnnn <--- Enter Device Efficiency R 10
?

The turbine efficiency is now entered and must be a numeric value between 0.0 and 1.0, decimal included. If no efficiency is entered the computer will default to an efficiency of 1.0.

After pressing RETURN the following will be displayed:

Enter GEOMETRY of inlets/outlets

nnn <--- Enter Connecting Node (1-3)
?

It is here that we enter the nodes to which the turbine is connected. The term GEOMETRY is used for the turbine because of its special nature. In general, turbines may contain several injection or extraction ports. The location of these ports relative to each other is essential for the proper analysis of the turbine. See Appendix C for an in depth discussion of this matter and several illustrative examples.
Since our simple turbine has no intermediate injection or extraction points the GEOMETRY that we enter will merely be the inlet and outlet nodes.

Begin by entering the node preceding the turbine. This happens to be a steam generator with a nodal value of one (1). Therefore enter 001 and RETURN.

\[ n \quad <--- \text{Inlet (1) or Outlet(O) (I-1)} \]

The node we just entered is the inlet to the turbine so we now enter 1 and RETURN.

\[ nnn \quad <--- \text{Enter Connecting Node (I-3)} \]

We now only have one connecting node remaining to be entered - the outlet from the turbine to the condenser. Enter the nodal value for the Condenser (003) and RETURN.

\[ n \quad <--- \text{Inlet (1) or Outlet(O) (I-1)} \]

Enter a 0 indicating the outlet and RETURN.

The turbine has now been completely described and the Network Input Menu (Fig. 4) is re-displayed. The same process is followed for the entry of the steam generator, pump and condenser. Appendix B briefly describes the entry prompts and responses for these other devices as well as for the turbine.

After all of the devices have been entered, select option 'X' to return to the Network Main Menu (Fig. 3).

If you would like to display a non detailed listing of the devices
you have entered select option 'E'. This will display all devices entered and their corresponding nodal values on the display screen and is a handy list to check for erroneous nodes.

Selecting option 'E' for the sample session results in the following display:

<table>
<thead>
<tr>
<th>Node</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Steam Generator</td>
</tr>
<tr>
<td>2</td>
<td>Steam Turbine</td>
</tr>
<tr>
<td>3</td>
<td>Condenser</td>
</tr>
<tr>
<td>4</td>
<td>Pump</td>
</tr>
</tbody>
</table>

End of Network File

press RETURN to continue

Figure 5

Pressing the return key at the end will return control back to the Network Main Menu (Fig. 3).

In order to obtain a more detailed display for each node, or to change any data for a node select option 'C' and the Change Menu will be displayed.
Selecting option 'X' will return control back to the Network Main Menu. Option 'N' is used to display the data for a specified node, while option 'R' is used when the nodal value is possibly unknown but the physical location of the record in the network file is known. Option 'N' will be used exclusively in most situations and will be used here.

Suppose we wish to check the data we entered for each node in our sample network. We first select option 'N' and are then prompted to enter a node

```
nnn  <--- Enter Node Number
```

Enter 003 to review the data for node number three and the data will be displayed as follows:
All numbers in parentheses denote either line or column numbers, depending on their placement. Most of the quantities are fairly self explanatory except for rows six and seven. Row six lists all the nodal connections to node number three. Directly below each node is a value of one for an inlet into node three, or a zero for an outlet from node three.

The quantity described as Work/Heat in row three must be interpreted in relation to the actual device listed. For a steam turbine this quantity represents the total work output of the turbine, whereas for a pump this would indicate the compression work added to the pump. For a condenser this quantity represents the heat rejected. The sign of Work/Heat is thus always positive when known except for the special case where the device is a feedwater heater. In this instance, Work/Heat represents the Terminal Temperature Difference of the heater and may be positive, negative, or zero.
Any quantities that are unknown are given a value of negative one. As we did not know the heat rejected by the condenser when we entered the initial data, it is reflected by the negative Work/Heat value.

Two items on the screen which may not be altered are the device description in row one and the relative record number (RRN) of the device in row two. All other items may be changed as is necessary by selecting option 'C'. We will then be prompted to enter the row number of the item we would like to change

n   Enter the ROW number of the Item ?

Suppose we want to change the efficiency from 1.0 to 0.75. Efficiency is located in row two so we would enter 2 and RETURN. We would then be prompted as follows:

nnnnnnnnnnn Enter the Efficiency (R 10) ?

We would then enter 0.75 and the new data will be displayed.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Node:</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Efficiency:</td>
<td>0.750000E+00</td>
</tr>
<tr>
<td>3</td>
<td>Work/Heat:</td>
<td>-.100000E+01</td>
</tr>
<tr>
<td>4</td>
<td># Inlets:</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td># Outlets:</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Nodal Connections</td>
<td>2 4</td>
</tr>
<tr>
<td>7</td>
<td>Inlet/Outlet</td>
<td>1 0</td>
</tr>
</tbody>
</table>

D - Delete Record    C - Change an Item    X - Leave As Is

n   <---- Enter Selection ?

Figure 8

If we had wanted to change any of the nodal connections we would
have entered a 6 in response to the row number prompt. We would then be prompted to enter the column number of the data and then make the correction as before.

To remove an entire record from the network file we would select option 'D'. After entering a 'D' and RETURN a safety prompt will appear:

DELETE this Record? (Enter Y or N)

Enter an 'N' if you have no desire to delete the record, or enter a 'Y' if you would like it to be deleted. If a 'Y' is selected, the record will be flagged for deletion and the node number will be changed to negative one. Thus all data for the node previously numbered three(3) in our example would become unaccessible.

If there are no changes (or deletions) to be made, you may leave the data record unscathed and return to the Change Menu (Fig. 6) by selecting option 'X'.

Select option 'X' to leave the Change Menu and return to the Network Main Menu (Fig. 3).

The only option remaining to be tried is option 'D' - delete a device from the system. Select this option and the Delete Menu will appear on the screen.
The Delete Menu is nearly identical to the Change Menu (Fig. 6) in both form and function. The obvious difference is that the Delete Menu is used to delete selected records given either the nodal value for the record or the actual record number in the file.

Select option 'N' and we are prompted to enter the node we would like to delete.

\[ nnn \quad \text{<--- Enter Node Number to Delete} \]

Although at present we have no node to delete, enter a value of 001 to display the actual delete screen.
The same data is displayed here as in the change screen but notice that on this screen the only option is whether or not to actually delete the record. This feature allows a brief review of data for a record to determine if it is, in fact, the record we want to delete. Since we do not want this record to be deleted enter an 'N' and press the return key and the Delete Menu (Fig. 9) is re-displayed.

Select option 'X' to return control to the Network Main Menu.

All of the options have now been covered in the entry of our sample network. If there is no more data to enter or changes to make, end the session and return to the Add/Change/Delete Menu (Fig. 1) by selecting option 'Q'.

<table>
<thead>
<tr>
<th>Node: 1</th>
<th>Steam Generator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficiency: 0.100000E+01</td>
<td>RRN : 2</td>
</tr>
<tr>
<td>Work/Heat: -.100000E+01</td>
<td></td>
</tr>
<tr>
<td># Inlets: 1</td>
<td></td>
</tr>
<tr>
<td># Outlets: 1</td>
<td></td>
</tr>
<tr>
<td>Nodal Connections --&gt; 2 4</td>
<td></td>
</tr>
<tr>
<td>Inlet(1)/Outlet(0)--&gt; 1 0</td>
<td></td>
</tr>
</tbody>
</table>

DELETE this Record? (Y or N)?

Figure 10
A/C/D MENU OPTION 2: ADD, CHANGE, OR DELETE PROPERTY DATA

We are now ready to enter the various known state properties. These include pressure, temperature, enthalpy, entropy and specific volume. Mass flow rate and quality are also included in the state description. A state may be described as a region in the network that is bounded by two devices. In other words, each flowpath from device to device is a separate state.

It should be noted that the entry of network and property data need not be performed in the order followed here. We could have just as easily entered the property data first. In any case, select option 2 and we will once again be prompted for the session id.

ENTER SESSION ID (1 - 8 CHARACTERS) .......

The session ID is now entered using the same rules as described earlier. For our sample network enter TEST and press the return key. The Property Main Menu will then be displayed.
Select option 'A' to add the known properties to the file. For our sample network the turbine inlet pressure and temperature, outlet pressure and mass flow rate are all known. Each state in the network is cataloged as (node1, node2) where the mass flow for the state flows from 'node1' to 'node2'. Thus the state where both pressure and temperature are known is (1, 2), since the mass flows from the steam generator (node 1) to the turbine (node 2). After selecting option 'A' the following prompt appear:

```
nnn  <--- Enter Preceding Node (1-3).

?  
```

This is looking for the 'node1' value associated with the state we are refering to. Because we need to enter the known properties for state (1, 2), respond to the prompt by entering 001 and then pressing the return key. A second, similar prompt will appear:
nnn  <--- Enter Succeeding Node (I-3).

This is looking for the 'node2' value of our state. Enter 002 (for state 1, 2), press the return key, and the following will be displayed:

<table>
<thead>
<tr>
<th>State:</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Pressure</td>
<td>(5) Quality</td>
<td></td>
</tr>
<tr>
<td>(2) Temperature</td>
<td>(6) Specific Volume</td>
<td></td>
</tr>
<tr>
<td>(3) Specific Enthalpy</td>
<td>(7) Mass Flow Rate</td>
<td></td>
</tr>
<tr>
<td>(4) Specific Entropy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter the Number of the Property (Properties) you would like to Add

n n n n  <-------------

Figure 12

Each state property is identified by a number from one to seven. We may enter up to three properties by entering the corresponding number below each lowercase 'n'. To add the pressure, temperature and mass flow for our sample state, enter the numbers one(1), two(2) and seven(7) so that each number is directly below an 'n'. The lower portion of Figure 12 will then appear as follows before the return key is pressed:

Enter the Number of the Property (Properties) you would like to Add

n n n n  <-------------
1 2 7

After entering the numbers in the form indicated above press the return key and you will be prompted for each property. Respond to each
prompt by entering the requested property in the format indicated. The prompts and entries should appear as follows:

```
nnnnnnnnnn  <----- Enter Pressure (R 10)
  100.0
nnnnnnnnnn  <----- Enter Temperature (R 10)
     500.
nnnnnnnnnn  <----- Enter Mass Flow Rate (R 10)
       1.000
```

When the properties have been entered you will be asked if there are any more to enter.

```
n  <----- Any more Properties to Enter ? (enter Y or N)
?
```

Because we still have the turbine outlet pressure to enter (state 2,3), enter a 'Y' and press RETURN. The entire entry procedure will then be repeated.

Caution should be exercised when selecting the properties to enter. Above all, never enter redundant data for a state. For example, do not enter pressure, temperature and quality. Rather enter pressure and quality or temperature and quality. Failure to comply with this rule may result in execution errors at the time of processing.

When all known properties have been entered, respond to the above prompt by entering an 'N' and pressing the return key. Control will then pass to the Property Main Menu (Fig. 11).

Options 'C' and 'D' perform exactly the same functions as their counterparts in the Network Main Menu except that now, changes or
deletions are made using the property file instead of the network file. Because of this similarity, no detailed examples will be given here.

Option 'E' will display all states and selected properties currently active in the property file. The display for our sample network appears as follows:

<table>
<thead>
<tr>
<th>Pred Node</th>
<th>Succ Node</th>
<th>Pressure</th>
<th>Temperature</th>
<th>Mass Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.100000E+03</td>
<td>0.50000E+03</td>
<td>0.100000E+01</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.200000E+02</td>
<td>-.100000E+01</td>
<td>-.100000E+01</td>
</tr>
</tbody>
</table>

End of STATE File
press RETURN to continue

Figure 13

As always, any items with a value of negative one are presumed to be unknown.

After all properties have been entered select option 'Q' to return to the Add/Change/Delete Menu (Fig. 2). You now have the option to enter more data or to quit. If there is no more data to enter, select option 'X' to return to the Main Menu.
MAIN MENU OPTION 2 : PRINT INITIAL DATA

Whenever you would like to obtain a hard copy of all the data in a particular session file select option 2 on the Main Menu. The display screen will clear and the following prompt will appear:

```
PRINT INITIAL DATA
=====================================
ENTER SESSION ID TO PRINT ..... 
```

Enter the session ID for the session that you would like to print out. All data for that session will be sorted by node number and printed in a clear manner.

After a few moments an informational message will appear on the screen:

```
BUILDING PRINT FILE FOR SESSION =========> session ID
```

This message indicates that there were no file errors for the session and the printout is under way. There are two possible errors that will bypass the above message.

The first, and probably most common, error occurs when the files do not exist for the specified session. That is, no data of any kind has been entered under that session name. If this is the case, the following message will be displayed:
THIS SESSION ==> session ID <=== IS NOT ON THE DISK

PRESS RETURN KEY TO RETURN TO MAIN MENU

When this happens, simply press the return key and the Main Menu will be re-displayed. You may now either enter data into the correct session, or select option 2 again to print a different session.

The second error that may occur is a little more complicated. The message associated with this error is as follows:

```
DATASET session.LENGTH.DATA IS NOT ALLOCATED
```

PRESS RETURN KEY TO END JOB

To correct this error you must manually build the LENGTH.DATA file and insert into it, through the use of one of the computer editors, the file lengths of the initial session files. See Appendix D for a complete explanation of session files.

If neither of these errors has occurred, the print file will build until all active records have been processed. Any records that have been flagged for deletion are now removed from the session files. When the print file is complete, the following prompt appears:

```
ENTER OUTPUT BIN NUMBER ....
```

Here you enter the bin number where your output will be placed when printed.

Finally, you are informed that the printing is complete by the
following message:

PRINT OUT IS NOW SCHEDULED

----- PRESS RETURN KEY TO CONTINUE -----

This indicates that the session listing is scheduled to be printed at the line printer. Pressing the return key will then free all allocated files and return you to the Main Menu.
After all the network and property data has been entered for a session, you may then select option 3 to begin the analysis of the system. You will first be prompted to enter the ID for the session which is to be analyzed.

Enter the correct session ID (enter TEST for our sample network) and press the return key. The program then takes over and begins the analysis procedure.

The actual analysis program is divided into four major sections:

A simple edit routine to determine if any of the nodes or states were incorrectly input,

A pre-processor to manipulate the input data into a form that the solution routine can easily handle,

A processor to step through the modified data files and solve for the unknown properties,
A post-processor to reassemble the final data into a form similar to that of the initial input files.

The edit routine checks two things. First it makes sure that all nodes are accountable. In other words it determines if there are any branches to non-existent nodes. It then checks to be certain that all states entered have valid nodes as their keys.

If either of the aforementioned errors should occur, the nodes in question are listed on the printer as well as the display screen and no further processing takes place. This will allow you to correct any menial errors quickly and almost immediately and continue with the major processing.

If the edit routine produces no errors, execution passes to the pre-processor. Once this stage has been reached there is no further user interaction. The pre-processor, processor, and post-processor are all dedicated programs which are designed to run without intervention.

Should any errors arise in any of these steps or should the session problem be insoluble, a listing, separate from the final data listing, will be printed. This additional listing will contain any error or warning messages that occurred during execution.

When the processing has ended, the following message will be displayed:
***** END OF PROCESSING *****

xxx Errors occurred
press RETURN key to continue

If any errors did occur, the value of 'xxx' will be greater than zero(0). Press the return key after this message and the following prompt will be displayed

WOULD YOU LIKE TO PRINT A HARD COPY
AT THIS TIME? ( Y OR N ) .....................

Respond with an 'N' if you do not want to print the output generated by the program. Control will then return to the main menu. If you answer with a 'Y' the following will appear

<table>
<thead>
<tr>
<th>OUTPUT DESTINATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>(1) - LINE PRINTER</td>
</tr>
<tr>
<td>(2) - SERVICE PRINTER</td>
</tr>
</tbody>
</table>

ENTER SELECTION ......

This allows you to send your output to either the main line printer or to the matrix service printer. After you have made your selection the output file will be printed on the appropriate printer and the main menu will be displayed.
MAIN MENU OPTION 4 : RE-PRINT FINAL DATA

Option 4 may be selected at anytime after the completion of the analysis routine (option 3). It will reprint the final data files and error messages that were printed at the end of the analysis routine.

Upon selecting option 4 the following prompt appears:

```
RE-PRINT FINAL DATA

Enter session ID to print ......
```

Enter the session ID to print and press the return key.
Select option 'S' at anytime to save the initial and final files for a session. Option 'S' may be selected at any time after the initial entry of data for a session. Upon selection, the following will be displayed:

```
SAVE SESSION DATA
=================================
ENTER SESSION ID TO SAVE ..... 
```

Enter the session ID that you would like to have saved and press the return key. The following prompt will then appear:

```
ENTER OUTPUT BIN NUMBER ..... 
```

The Save routine saves the session files by submitting a batch savecard to the system. Because of the batch submission, a listing of all files saved is printed. The bin number entered above indicates where this listing is to be placed when printing is complete. Enter a bin number and press the return key. A final message is then displayed:

```
ALL DATA FOR SESSION session.ID HAS BEEN SAVED

PRESS THE RETURN KEY TO RETURN TO MAIN MENU
```
This message will verify that a savecard has been submitted for the correct session. Press the return key and the Main Menu will be re-displayed.
MAIN MENU OPTION R: REMOVE DATA FROM DISK

When the data for a particular session is no longer needed, it may be deleted from the disk by selecting option 'R'. The display screen will then prompt you as follows:

```
DELETE SESSION FILES

THIS PROCEDURE WILL DELETE ALL FILES FOR A PARTICULAR SESSION
WOULD YOU LIKE TO CONTINUE?
ENTER Y OR N ....
```

This prompt allows you to return to the Main Menu without anything being deleted by answering with an 'N'. If you actually want to delete the files for a session, respond to the prompt with a 'Y'. You will then be asked to enter the session you would like deleted:

```
ENTER SESSION ID TO DELETE (1 - 8 CHARACTERS) ....
```

After entering the correct session ID the procedure will then delete ALL the files associated with that session. That is, both the initial and final data files will be deleted.

At the completion of the process the following message will be displayed:
ALL FILES FOR SESSION session.ID HAVE BEEN DELETED

PRESS THE RETURN KEY TO RETURN TO MAIN MENU

This will verify that all the files for the session have been deleted from the disk. Press the return key and control will pass to the Main Menu.
The analysis routine uses the First Law of Thermodynamics to model each of the devices in the network. A heat and mass balance are performed for each device individually to solve for any unknown quantities.

Entropy comes into play through the device's efficiency. That is, for efficiencies less than unity, the outlet entropy of a device will be different from its inlet entropy. Whether the outlet entropy is larger or smaller than the inlet entropy depends on the device in question. For devices where work is added or heat rejected, such as a pump or a condenser, the outlet entropy will be less than the inlet entropy. The situation will be reversed for devices where work is produced or heat is added, such as turbines or steam generators.

All devices that do work or have work done to them (turbines or pumps) are considered to be adiabatic. Similarly, all devices that possess a heat flux are free from any work input or output. Some devices, such as mixers and branches, are considered to be both adiabatic and free of work.

In any device, the value of the work or heat is represented by its absolute value. The program determines the sign of the quantity in the appropriate solution routine for each device.

The closed feed water heater is a device that differs from the above descriptions. The heater is assumed to be adiabatic and free of
work. It is given the characteristic value of terminal temperature difference. The terminal temperature difference (TTD) is the difference in temperature between the 'cold' flow outlet temperature and the saturation temperature of the 'hot' flow. The TTD may take on any value: positive, negative, or zero.

As stated earlier, an attempt is made to solve for the unknowns of each device in an isolated fashion, using only the device in question. This is not always possible when feed water heaters and mixers are involved. If, after one initial pass, either of these devices cannot be solved, a special routine is used to solve, simultaneously, a feed water heater and its associated mixer. The concept of entropy is not used when solving these types of devices.

The program will make various assumptions for certain devices if they prove to be insoluble. A saturated liquid state will be assumed for the outlet of a condenser as well as for the inlet of a pump if these states are unknown after one pass. The same will be assumed for the outlet of the 'hot' flow of a feed water heater.

The term 'pass' mentioned above refers to the number of times the program has looped around the network. The program mimics the human brain in solving the network. That is, it steps through the network, logically moving from one device to the next. A pass has been completed when the program has stepped through the entire network.

The program will make several passes through the network until all devices and states have been solved. To prevent the program from
looping indefinitely for an insoluble network, an upper bound is defined to limit the number of passes the program may take. This upper bound is expressed as a function of the number of mixers in the network. This function is based primarily on intuition but appears to generate reasonable upper bounds for the various networks tested.

The mathematics used in modeling the various devices is presented in the following pages.
**VARIABLE NOMENCLATURE**

- $h$ - Specific Enthalpy
- $m$ - Mass Flow Rate
- $\eta$ - Efficiency
- $P$ - Pressure
- $Q$ - Heat
- $s$ - Specific Entropy
- $T$ - Temperature
- $W$ - Work

\[ \delta_k = +1 \quad \text{Inlet} \]
\[ -1 \quad \text{Outlet} \]

subscript of '1' - inlet state

subscript of '2' - outlet state

subscript of 'i' - ideal case ($\eta = 1.0$)

subscript of 'f' - saturated liquid state

\[ \sum^f \] Sum over all inlets and outlets

TTD - Terminal Temperature Difference
STEAM TURBINE

Case 1 - Given state 1, state 2, m --> find W, γ

\[ W = m(h_1 - h_2) \]

\[ s_{z_i} = s_1, \quad p_{z_i, s_{z_i}} \rightarrow h_{z_i} \]

\[ W_i = m(h_1 - h_{z_i}) \]

\[ \gamma = \frac{W}{W_i} \]

Case 2 - Given state 1, W, γ, m --> find state 2

\[ h_2 = h_1 - \frac{W}{m} \]

\[ h_{2i} = h_1 - \frac{W}{m/\gamma} \]

\[ s_{z_i} = s_1, \quad s_{z_i, h_{2i}} \rightarrow p_2 \]

\[ p_2, h_2 \rightarrow \text{state 2} \]
Case 3 - Given state 2, \( W, \eta, m \) \( \rightarrow \) find state 1

\[
\begin{align*}
    h_1 &= \frac{W}{m} + h_2 \\
    h_{2i} &= h_1 - \frac{W}{m}/\eta \\
    P_2, h_{2i} &\rightarrow s_{2i} \\
    s_1 &= s_{2i} \\
    h_1, s_1 &\rightarrow \text{state 1}
\end{align*}
\]

Case 4 - Given state 1, \( P_2, \eta \) \( \rightarrow \) find state 2

\[
\begin{align*}
    s_{2i} &= s_1 \\
    P_2, s_{2i} &\rightarrow h_{2i} \\
    W/m &= \eta \cdot (h_1 - h_{2i}) \\
    h_2 &= h_1 - W/m \\
    P_2, h_2 &\rightarrow \text{state 2}
\end{align*}
\]
Case 5 - Given state 1, \( P_2, \eta, \ m \rightarrow \) find state 2, \( W \)

\[
S_{2i} = S_1
\]

\[
P_{2, s_{2i}} \rightarrow h_{2i}
\]

\[
W = \eta \cdot m \cdot (h_1 - h_{2i})
\]

\[
h_2 = h_1 - \frac{W}{m}
\]

\[
P_{2, h_2} \rightarrow \text{state 2}
\]
Case 1 - Given all $m$, all $h$ --> find $Q$

$$Q = - \left( \sum s_k m_k h_k \right)$$

Case 2 - Given all $m$, all but 1 $h$, $Q$ --> find unknown $h$

$$s_a m_a h_a + \sum_{k \neq a} s_k m_k h_k = -Q$$

$$h_a = - \left( \frac{Q + \sum_{k \neq a} s_k m_k h_k}{s_a m_a} \right)$$
Case 3 - Given all h, all but 1 m, Q --> find unknown m

\[ \delta_a m_a h_a + \delta_b m_b h_b + \sum_{k \neq a, b} \delta_k m_k h_k = -Q \]

but \( m_b = m_a \), \( \delta_b = -\delta_a \)

\[ \delta_a m_a (h_a - h_b) = -\left( Q + \sum_{k \neq a, b} \delta_k m_k h_k \right) \]

\[ m_a = -\left( \frac{Q + \sum_{k \neq a, b} \delta_k m_k h_k}{\delta_a (h_a - h_b)} \right) \]
CONDENSER

Case 1 - Given state 1, state 2, m --> find Q

\[ Q = m (h_1 - h_2) \]

Case 2 - Given state 1, Q --> find state 2

\[ h_2 = h_1 - \frac{Q}{m} \]

\[ p_2 = p_1 \]

\[ p_2, h_2 \rightarrow \text{state 2} \]
Case 5 - Given state 1 --> find state 2

\[ P_2 = P_1 \]

assume \( h_2 = h_f \) @ \( P_2 \)

\( P_2, h_2 \rightarrow \text{state 2} \)
PUMP

Case 1 - Given state 1, state 2, m \(\rightarrow\) find \(W, \eta\)

\[
W = m (h_2 - h_1)
\]

\[
S_{2i} = S_i, P_i S_{2i} \rightarrow h_{2i}
\]

\[
W_i = m (h_{2i} - h_i)
\]

\[
\eta = W/W_i
\]

Case 2 - Given state 1, \(W, \eta, m\) \(\rightarrow\) find state 2

\[
h_2 = h_1 + W/m
\]

\[
h_{2i} = h_1 + W/m/\eta
\]

\[
S_{2i} = S_i, S_{2i}, h_{2i} \rightarrow P_i
\]

\[
P_i, h_2 \rightarrow \text{state } 2
\]
Case 3 - Given state 2, \( W, \eta, m \) \( \rightarrow \) find state 1

\[
\begin{align*}
    h_1 &= -W/m + h_2 \\
    h_{2i}' &= h_i + W/m/\eta \\
    P_2, h_{2i}' &\rightarrow S_{2i}' \quad S_1 = S_{2i}' \\
    h_1, s_1 &\rightarrow \text{state 1}
\end{align*}
\]

Case 4 - Given state 1, \( P_2, \eta \) \( \rightarrow \) find state 2

\[
\begin{align*}
    S_{2i}' &= S_1 \\
    P_2, S_{2i}' &\rightarrow h_{2i}' \\
    W/m &= \eta \cdot (h_{2i}' - h_i) \\
    h_2 &= h_1 + W/m \\
    P_2, h_2 &\rightarrow \text{state 2}
\end{align*}
\]
Case 5 - Given state 1, $P_2, \eta, m \rightarrow$ find state 2, $W$

\[
S_{2i} = S_i
\]

\[
P_{2i} S_{2i} \rightarrow h_{2z}
\]

\[
W = \eta \cdot m \cdot (h_{2z} - h_i)
\]

\[
h_2 = h_i + W
\]

\[
P_{2z} h_2 \rightarrow \text{state 2}
\]

Case 6 - Given $P_1$ \rightarrow find state 1

assume \quad h_i = h_f @ P_1

\[
P_{1i} h_i \rightarrow \text{state 1}
\]
FEED WATER HEATER

Case 1 - Given all h, one m --> find other m

\[ \sum s_k m_k h_k = 0 \]

\[ s_a m_a h_a + s_b m_b h_b = - \sum_{k \neq a,b} s_k m_k h_k \]

but \( m_b = m_a \), \( s_b = -s_a \)

\[ s_a m_a (h_a - h_b) = - \sum_{k \neq a,b} s_k m_k h_k \]

\[ m_a = -\left( \frac{\sum_{k \neq a,b} s_k m_k h_k}{s_a (h_a - h_b)} \right) \]

Case 2 - Given all m, all but 1 h --> find unknown h

\[ s_a m_a h_a = - \sum_{k \neq a} s_k m_k h_k \]

\[ h_a = -\left( \frac{\sum_{k \neq a} s_k m_k h_k}{s_a m_a} \right) \]
Case 3 - Given hot flow $P_1, h_1$ --> find hot flow $h_2$, cold flow $T_2$

hot flow $\rightarrow P_2 = P_1$

assume $h_{\text{hot},2} = h_f @ P_{\text{hot},2}$

$P_{\text{hot},2}, h_{\text{hot},2} \rightarrow$ hot flow State $Z$

$T_{\text{cold},2} = T_{\text{hot},2} - TTD$

Case 4 - Given hot flow $T_2$ --> find cold flow $T_2$

$T_{\text{cold},2} = T_{\text{hot},2} - TTD$
Case 5 - Given all h, no m --> solve simultaneously with mixer

Setup A

\[ m_A(h_1 - h_2) = m_B(h_4 - h_3) \]

\[ R = \frac{m_A}{m_B} = \frac{h_4 - h_3}{h_1 - h_2} \]

Mixer conservation of energy:

\[ m_A h_5 + m_B h_4 = m_c h_6 \]

But \[ m_c = m_A + m_B \]

\[ m_A h_5 + m_B h_4 = (m_A + m_B) h_6 \]

Divide by \[ m_B : R \cdot h_5 + h_4 = (R+1) h_6 \]

\[ h_6 = \frac{R \cdot h_5 + h_4}{R+1} \]
conservation of energy: heater

\[ m_A(h_1 - h_2) = m_B(h_4 - h_3) \]

\[ R = \frac{m_A}{m_B} = \frac{h_4 - h_3}{h_1 - h_2} \]

conservation of energy: mixer

\[ m_c h_6 + m_A h_5 = m_B h_7 \]

but \[ m_c = m_B - m_A \]

\[ (m_B - m_A) h_6 + m_A h_5 = m_B h_7 \]

divide by \( m_B \): \( (1 - R)h_6 + R h_5 = h_7 \)
Case 1 - Given all but 1 m → find other m

Conservation of mass: \( \sum s_k m_k = 0 \)

\[ s_a m_a = - \sum_{k \neq a} s_k m_k \]

\[ m_a = - \frac{1}{s_a} \sum_{k \neq a} s_k m_k \]
Case 1 - Given all h, all but 2 m --> find unknown m

\[ \delta_b m_b = - \left( \sum_{K \neq a} \delta_K m_K + \delta_a m_a \right) - (1) \ (mass) \]

\[ \delta_a m_a + \delta_b m_b h_b = - \sum_{K \neq a} \delta_K m_K h_K \ (2) \ (energy) \]

Substitute 1 $$\rightarrow$$ 2:

\[ \delta_a m_a - h_b \left[ \sum_{K \neq a} \delta_K m_K + \delta_a m_a \right] = - \sum_{K \neq a} \delta_K m_K h_K \]

\[ \delta_a m_a (h_a - h_b) = - \sum_{K \neq a} \delta_K m_K h_K + \sum_{K \neq a} \delta_K m_K h_b \]

\[ m_a = \frac{\sum_{K \neq a} \delta_K m_K (h_b - h_K)}{\delta_a (h_a - h_b)} \]

\[ m_b = - \left( \frac{\sum_{K \neq b} \delta_K m_K}{\delta_b} \right) \]
Case 2 - Given all but 1 m, all but 1 h --> find unknown m, h

\[ s_a m_a = - \sum_{k+a} s_k m_k \quad - (1) \quad \text{(mass)} \]

\[ s_b m_b h_b = - \sum_{k+b} s_k m_k h_k \quad - (2) \quad \text{(energy)} \]

\[ m_a = - \frac{\sum s_k m_k}{s_a} \quad \text{(from 1)} \]

\[ h_b = - \frac{\sum s_k m_k h_k}{s_b m_b} \quad \text{(from 2)} \]
SAMPLE RUNS AND RESULTS

Six different networks of varying difficulty were used to test the operation and accuracy of the Thermal Analyzer Program. Each of these networks is presented in the following pages with the results obtained using the analysis program. Beside each result, enclosed in parenthesis, is the value obtained by hand calculations.
I. SIMPLE RANKINE CYCLE

Given:

- State (1,2), \( P = 400 \text{ psi} \)
- State (1,2), \( x = 1.0 \)
- State (1,2), \( m = 1.0 \text{ lbm/sec} \)
- State (2,3), \( P = 10 \text{ psi} \)
<table>
<thead>
<tr>
<th>state</th>
<th>Enthalpy (Btu/lbm)</th>
<th>Quality</th>
<th>Mass Flow (lbm/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>1205.49 (1205.0 )</td>
<td>1.00 (1.00)</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(2,3)</td>
<td>946.04 ( 944 )</td>
<td>0.799 (0.80)</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(3,4)</td>
<td>161.19 (161.23 )</td>
<td>0.00 (0.00)</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(4,1)</td>
<td>162.46 (162.43 )</td>
<td>-1.0 (-1.0)</td>
<td>1.000 (1.000)</td>
</tr>
</tbody>
</table>

Expansion Work = 259.454 Btus/sec ( 261.5 )
Compression Work = 1.26563 Btus/sec ( 1.2 )
Heat Addition = 1043.03 Btus/sec (1043.1 )
II. SIMPLE REHEAT CYCLE

Given:
- State (1,2), $P=\text{1000 psi}$
- State (1,2), $T=\text{960 }^\circ\text{F}$
- State (1,2), $m=\text{1.0 lbm/sec}$
- State (2,1), $P=\text{450 psi}$
- State (3,4), $P=\text{2 psi}$
<table>
<thead>
<tr>
<th>state</th>
<th>Enthalpy (Btu/lbm)</th>
<th>Quality</th>
<th>Mass Flow (lbm/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>1482.88 (1482.9)</td>
<td>-1.0 (-1.0)</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(1,3)</td>
<td>1500.61 (1500.61)</td>
<td>-1.0 (-1.0)</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(2,1)</td>
<td>1374.01 (1373. )</td>
<td>-1.0 (-1.0)</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(3,4)</td>
<td>1007.54 (1007. )</td>
<td>.894 (.894)</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(4,5)</td>
<td>94.0164 ( 94.02 )</td>
<td>0.00 (0.00)</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(5,1)</td>
<td>97.0834 ( 97.02 )</td>
<td>-1.0 (-1.0)</td>
<td>1.000 (1.000)</td>
</tr>
</tbody>
</table>

Expansion Work = 601.941 Btus/sec ( 603.5 )
Compression Work = 3.06703 Btus/sec ( 3.00 )
Heat Addition = 1512.40 Btus/sec (1513.5 )
III. ONE STAGE REGENERATION CYCLE

Given:
- State (7,1), $P=450$ psi
- State (7,1), $T=800$ F
- State (7,1), $m=1.0$ lbm/sec
- State (1,4), $P=60$ psi
- State (1,2), $P=2$ psi
- Node 4, $TTD=10$ F
<table>
<thead>
<tr>
<th>state</th>
<th>Enthalpy (Btu/lbm)</th>
<th>Quality</th>
<th>Mass Flow (lbm/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>969.771 (970. )</td>
<td>0.857 (0.86)</td>
<td>0.8561 (.8564)</td>
</tr>
<tr>
<td>(1,4)</td>
<td>1197.81 (1198. )</td>
<td>-1.0 (-1.0)</td>
<td>0.1439 (.1436)</td>
</tr>
<tr>
<td>(2,3)</td>
<td>94.0164 (94.02 )</td>
<td>0.00 (0.00)</td>
<td>0.8561 (.8564)</td>
</tr>
<tr>
<td>(3,4)</td>
<td>95.433 (95.37 )</td>
<td>-1.0 (-1.0)</td>
<td>0.8561 (.8564)</td>
</tr>
<tr>
<td>(4,5)</td>
<td>262.310 (262.25 )</td>
<td>0.00 (0.00)</td>
<td>0.1439 (.1436)</td>
</tr>
<tr>
<td>(4,6)</td>
<td>252.752 (252.25 )</td>
<td>-1.0 (-1.0)</td>
<td>0.8561 (.8564)</td>
</tr>
<tr>
<td>(5,6)</td>
<td>263.625 (263.5 )</td>
<td>-1.0 (-1.0)</td>
<td>0.1439 (.1436)</td>
</tr>
<tr>
<td>(6,7)</td>
<td>254.317 (253.9 )</td>
<td>-1.0 (-1.0)</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(7,1)</td>
<td>1414.35 (1414.4 )</td>
<td>-1.0 (-1.0)</td>
<td>1.000 (1.000)</td>
</tr>
</tbody>
</table>

Expansion Work = 411.756 Btus/sec (411.7)
Compression Work = 1.40223 Btus/sec (1.34)
Heat Addition = 1160.03 Btus/sec (1160.5)
IV. TWO STAGE REGENERATION CYCLE

Given:
- State (1,2), $P=1250$ psi
- State (1,2), $T=960$ F
- State (1,2), $m=1.0$ lbm/sec
- State (2,8), $P=260$ psi
- State (2,5), $P=25$ psi
- State (2,3), $P=0.5$ psi
- Nodes 5,8, TTD=12 F

Program 3

Heat In

Work In

Heat Out
<table>
<thead>
<tr>
<th>state</th>
<th>Enthalpy (Btu/lbm)</th>
<th>Quality</th>
<th>Mass Flow (lbm/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>1474.49 (1474.5)</td>
<td>-1.0 (-1.0)</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(2,3)</td>
<td>864.847 (864.)</td>
<td>.779 (0.81)</td>
<td>.7217 (.7222)</td>
</tr>
<tr>
<td>(3,5)</td>
<td>1086.33 (1086.)</td>
<td>.922 (0.92)</td>
<td>.1214 (.1195)</td>
</tr>
<tr>
<td>(2,8)</td>
<td>1280.01 (1280.)</td>
<td>-1.0 (-1.0)</td>
<td>.1568 (.1583)</td>
</tr>
<tr>
<td>(3,4)</td>
<td>47.6649 (47.65)</td>
<td>-1.0 (0.00)</td>
<td>.1568 (.1583)</td>
</tr>
<tr>
<td>(4,5)</td>
<td>51.3756 (51.37)</td>
<td>-1.0 (-1.0)</td>
<td>.7217 (.7222)</td>
</tr>
<tr>
<td>(5,6)</td>
<td>208.527 (208.52)</td>
<td>0.00 (0.00)</td>
<td>.1214 (.1195)</td>
</tr>
<tr>
<td>(5,7)</td>
<td>199.045 (196.52)</td>
<td>-1.0 (-1.0)</td>
<td>.7217 (.7222)</td>
</tr>
<tr>
<td>(6,7)</td>
<td>212.370 (212.35)</td>
<td>-1.0 (-1.0)</td>
<td>.1214 (.1195)</td>
</tr>
<tr>
<td>(7,8)</td>
<td>200.964 (198.76)</td>
<td>-1.0 (-1.0)</td>
<td>.8432 (.8417)</td>
</tr>
<tr>
<td>(8,9)</td>
<td>379.940 (380.0)</td>
<td>0.00 (0.00)</td>
<td>.1568 (.1583)</td>
</tr>
<tr>
<td>(8,10)</td>
<td>368.342 (368.0)</td>
<td>-1.0 (-1.0)</td>
<td>.8432 (.8417)</td>
</tr>
<tr>
<td>(9,10)</td>
<td>383.409 (383.43)</td>
<td>-1.0 (-1.0)</td>
<td>.1568 (.1583)</td>
</tr>
<tr>
<td>(10,1)</td>
<td>370.705 (370.4)</td>
<td>-1.0 (-1.0)</td>
<td>1.000 (1.000)</td>
</tr>
</tbody>
</table>

Expansion Work = 517.631 Btus/sec (518.1)
Compression Work = 3.68889 Btus/sec (3.68)
Heat Addition = 1103.78 Btus/sec (1104.1)
V. THREE STAGE REGENERATION CYCLE

Given:

- State (1,2), \( P = 1250 \text{ psi} \)
- State (1,2), \( T = 960 \text{ F} \)
- State (1,2), \( m = 1.0 \text{ lbm/sec} \)
- State (2,11), \( P = 425 \text{ psi} \)
- State (2,8), \( P = 100 \text{ psi} \)
- State (2,5), \( P = 15 \text{ psi} \)
- State (2,3), \( P = 0.5 \text{ psi} \)
- Nodes 5, 8, 11, \( \text{TTD} = 12 \text{ F} \)
<table>
<thead>
<tr>
<th>state</th>
<th>Enthalpy (Btu/lbm)</th>
<th>Quality</th>
<th>Mass Flow (lbm/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>1474.49 (1474.5 )</td>
<td>-1.0 (-1.0)</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(2,3)</td>
<td>864.847 (865.1 )</td>
<td>0.779 (0.80)</td>
<td>0.6815 (.6813)</td>
</tr>
<tr>
<td>(2,5)</td>
<td>1051.86 (1052. )</td>
<td>0.898 (0.90)</td>
<td>0.0944 (.0922)</td>
</tr>
<tr>
<td>(2,8)</td>
<td>1191.38 (1191. )</td>
<td>-1.0 (-1.0)</td>
<td>0.0997 (.1001)</td>
</tr>
<tr>
<td>(2,11)</td>
<td>1333.75 (1333. )</td>
<td>-1.0 (-1.0)</td>
<td>0.1244 (.1264)</td>
</tr>
<tr>
<td>(3,4)</td>
<td>47.6649 (47.65 )</td>
<td>0.00 (0.00)</td>
<td>0.6815 (.6813)</td>
</tr>
<tr>
<td>(4,5)</td>
<td>51.3756 (51.37 )</td>
<td>-1.0 (-1.0)</td>
<td>0.6815 (.6813)</td>
</tr>
<tr>
<td>(5,6)</td>
<td>181.159 (181.19 )</td>
<td>0.00 (0.00)</td>
<td>0.0944 (.0922)</td>
</tr>
<tr>
<td>(5,7)</td>
<td>171.914 (169.19 )</td>
<td>-1.0 (-1.0)</td>
<td>0.6815 (.6813)</td>
</tr>
<tr>
<td>(6,7)</td>
<td>185.036 (185.01 )</td>
<td>-1.0 (-1.0)</td>
<td>0.0944 (.0922)</td>
</tr>
<tr>
<td>(7,8)</td>
<td>173.510 (171.08 )</td>
<td>-1.0 (-1.0)</td>
<td>0.7759 (.7735)</td>
</tr>
<tr>
<td>(8,9)</td>
<td>298.668 (298.61 )</td>
<td>0.00 (0.00)</td>
<td>0.0997 (.1001)</td>
</tr>
<tr>
<td>(8,10)</td>
<td>288.202 (286.61 )</td>
<td>-1.0 (-1.0)</td>
<td>0.7759 (.7735)</td>
</tr>
<tr>
<td>(9,10)</td>
<td>302.500 (302.39 )</td>
<td>-1.0 (-1.0)</td>
<td>0.0997 (.1001)</td>
</tr>
<tr>
<td>(10,11)</td>
<td>289.830 (288.42 )</td>
<td>-1.0 (-1.0)</td>
<td>0.8756 (.8736)</td>
</tr>
<tr>
<td>(11,12)</td>
<td>430.792 (430.9 )</td>
<td>0.00 (0.00)</td>
<td>0.1244 (.1264)</td>
</tr>
<tr>
<td>(11,13)</td>
<td>418.143 (418.9 )</td>
<td>-1.0 (-1.0)</td>
<td>0.8756 (.8736)</td>
</tr>
<tr>
<td>(12,13)</td>
<td>433.808 (433.9 )</td>
<td>-1.0 (-1.0)</td>
<td>0.1244 (.1264)</td>
</tr>
<tr>
<td>(13,1)</td>
<td>420.092 (420.8 )</td>
<td>-1.0 (-1.0)</td>
<td>1.000 (1.000)</td>
</tr>
</tbody>
</table>

Expansion Work = 501.097 Btus/sec (500.4 )
Compression Work = 3.65206 Btus/sec (3.64 )
Heat Addition = 1054.40 Btus/sec (1053.7 )
VI. REGENERATION WITH FLASH TO THE CONDENSER

Given:

- State (1,2), $P=450$ psi
- State (1,2), $T=800$ °F
- State (1,2), $m=1.0$ lbm/sec
- State (2,5), $P=60$ psi
- State (2,3), $P=2$ psi
- Node 5, $T_{TD}=10$ °F

---

Diagram:

- Heat In
- Work In
- Work Out
- Heat Out
<table>
<thead>
<tr>
<th>State</th>
<th>Enthalpy (Btu/lbm)</th>
<th>Quality</th>
<th>Mass Flow (lbm/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>1414.35 (1414.4)</td>
<td>-1.0</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(2,5)</td>
<td>1197.81 (1198.)</td>
<td>-1.0</td>
<td>1.682 (.1677)</td>
</tr>
<tr>
<td>(2,99)</td>
<td>969.771 (970.)</td>
<td>.857</td>
<td>.8318 (.8323)</td>
</tr>
<tr>
<td>(3,4)</td>
<td>94.0164 (94.02)</td>
<td>0.00</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(4,5)</td>
<td>95.4332 (95.37)</td>
<td>-1.0</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(5,1)</td>
<td>252.752 (252.25)</td>
<td>-1.0</td>
<td>1.000 (1.000)</td>
</tr>
<tr>
<td>(5,6)</td>
<td>262.310 (262.25)</td>
<td>0.00</td>
<td>1.682 (.1677)</td>
</tr>
<tr>
<td>(6,3)</td>
<td>262.310 (262.25)</td>
<td>.165</td>
<td>1.682 (.1677)</td>
</tr>
<tr>
<td>(99,3)</td>
<td>969.771 (970.)</td>
<td>.857</td>
<td>.8318 (.8323)</td>
</tr>
</tbody>
</table>

Expansion Work = 406.231 Btus/sec (406.2)
Compression Work = 1.41673 Btus/sec (1.35)
Heat Addition = 1161.60 Btus/sec (1162.15)
CONCLUSIONS

The results presented indicate that the program does indeed solve a thermal system with excellent results. Any discrepancies that do arise may be due to inaccuracies in the steam table data that resides in the computer.

The program could be expanded easily to include other types of devices as well as other working fluids.
APPENDIX A

Below is a diagram of the sample network TEST used in the manual and the list of known properties.

<table>
<thead>
<tr>
<th>node</th>
<th>device</th>
<th># inlets</th>
<th># outlets</th>
<th>efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Steam Generator</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>Steam Turbine</td>
<td>1</td>
<td>1</td>
<td>0.75</td>
</tr>
<tr>
<td>3</td>
<td>Condenser</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>Pump</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>state</th>
<th>Pressure(psi)</th>
<th>Temperature(F)</th>
<th>Mass Flow(lbm/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>100.0</td>
<td>500.0</td>
<td>1.0</td>
</tr>
<tr>
<td>2-3</td>
<td>20.0</td>
<td>---</td>
<td>--</td>
</tr>
</tbody>
</table>
APPENDIX B

Below are listed the various prompts and responses for the entry of the sample network TEST shown in Appendix A.

NETWORK ENTRY

STEAM GENERATOR

nnn <--- Enter Nodal Value for Device (I-3)
001

nnnnnnnnnn <--- Enter HEAT input (if known) R 10
-1.0

nnnnnnnnnn <--- Enter Device Efficiency R 10
1.0

nn <--- Enter the number of Separate Flows for the device (I-2)
01

For Flow # 1

nnn <--- Enter the Nodal Connection to the INLET (I-3)
004

nnn <--- Enter the Nodal Connection to the OUTLET (I-3)
002

STEAM TURBINE

nnn <--- Enter Nodal Value for Device (I-3)
002

nnn <--- Enter the # of INLETS into device (I-3)
001

nnn <--- Enter the # of OUTLETS from device (I-3)
001

nnnnnnnnnn <--- Enter WORK output (if known) R 10
-1.0
nnnnnnnnnn
0.75

<--- Enter Device Efficiency R 10

Enter GEOMETRY of inlets/outlets

nnn
001

<--- Enter Connecting Node (I-3)

n
1

<----- Inlet (1) or Outlet (0) (I-1)

nnn
003

<--- Enter Connecting Node (I-3)

n
0

<----- Inlet (1) or Outlet (0) (I-1)


CONDENSER


nnn
001

<--- Enter Nodal Value for Device (I-3)

nnnnnnnnnn
-1.0

<--- Enter HEAT rejected (if known) R 10

nnnnnnnnn
1.0

<--- Enter Device Efficiency R 10

nnn
001

<--- Enter the number of INLETS into device (I-3)

nnn
002

<--- Enter Nodal Connection for Inlet # 1 (I-3)

nnn
004

<--- Enter Nodal Connection to the Outlet (I-3)


PUMP

nnn
004

<--- Enter Nodal Value for Device (I-3)

nnnnnnnnnn
-1.0

<--- Enter WORK input (if known) R 10
--- Enter Device Efficiency  \( \text{R 10} \)

--- Enter the Nodal Connection to the INLET (I-3)  \( \text{003} \)

--- Enter the Nodal Connection to the OUTLET (I-3)  \( \text{001} \)

**PROPERTY ENTRY**

**STATE 1-2**

--- Enter Preceding Node (I-3).  \( \text{001} \)

--- Enter Succeeding Node (I-3).  \( \text{002} \)

State: 1 2

<table>
<thead>
<tr>
<th>State</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Pressure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2) Temperature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3) Specific Enthalpy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4) Specific Entropy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5) Quality</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(6) Specific Volume</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(7) Mass Flow Rate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter the Number of the Property (Properties) you would like to Add

---

<table>
<thead>
<tr>
<th>1 2 7</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

--- Enter Pressure  \( \text{R 10} \) 100.0

--- Enter Temperature  \( \text{R 10} \) 500.0

--- Enter Mass Flow Rate  \( \text{R 10} \) 1.0

--- Any more Properties to Enter? (enter Y or N)  \( \text{Y} \)
STATE 2-3

nnn  <--- Enter Preceding Node (1-3).
002

nnn  <--- Enter Succeeding Node (1-3).
003

State:  2  3

(1) Pressure    (5) Quality
(2) Temperature (6) Specific Volume
(3) Specific Enthalpy  (7) Mass Flow Rate
(4) Specific Entropy

Enter the Number of the Property (Properties)
you would like to Add

nnnn          
1

nnnnnnnnnnn  <--- Enter Pressure (R 10)
20.0

n  <--- Any more Properties to Enter ? (enter Y or N)
N
SAMPLE INPUT PROMPTS FOR OTHER DEVICES

Sample input prompts for other devices are presented below. These samples are indicative of the types of prompts to be expected for the various devices.

**FEEDWATER HEATER**

nnnnnnnnnnn  <--- Enter Terminal Temperature Difference

For the "HOT" Flow ......

nnn  <--- Enter The Nodal Connection to the Inlet

nnn  <--- Enter The Nodal Connection to the Outlet

For The "COLD" Flow ......

nnn  <--- Enter The Nodal Connection to the Inlet

nnn  <--- Enter The Nodal Connection to the Outlet

**BRANCH**

nnn  <--- Enter The Nodal Connection to the Inlet

nnn  <--- Enter # of Outlets from Device

nnn  <--- Enter The Nodal Connection to the Outlet

**MIXER**

nnn  <--- Enter # of Inlets Into Device

nnn  <--- Enter the Nodal Connection for Inlet # n

nnn  <--- Enter the Nodal Connection for the Outlet
THROTTLE and DUMMY NODE

nnn  <--- Enter the Nodal Connection to the Inlet
nnn  <--- Enter the Nodal Connection to the Outlet

"T" CONNECTION

nnn  <--- Enter # of Inlets Into Device
nnn  <--- Enter # of Outlets from Device
nnn  <--- Enter the Nodal Connection to the Inlet
nnn  <--- Enter the Nodal Connection to the Outlet
APPENDIX C

DETAILED DESCRIPTION OF NETWORK DEVICES

An in depth description of each network device and its associated terminology is discussed below.

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steam Generator</td>
<td>1</td>
</tr>
<tr>
<td>Steam Turbine</td>
<td>2</td>
</tr>
<tr>
<td>Condenser</td>
<td>3</td>
</tr>
<tr>
<td>Pump</td>
<td>4</td>
</tr>
<tr>
<td>Feedwater Heater</td>
<td>5</td>
</tr>
<tr>
<td>Branch</td>
<td>6</td>
</tr>
<tr>
<td>Mixing Bin</td>
<td>7</td>
</tr>
<tr>
<td>Throttle</td>
<td>8</td>
</tr>
<tr>
<td>Dummy Node</td>
<td>9</td>
</tr>
<tr>
<td>'T' Connection</td>
<td>*</td>
</tr>
</tbody>
</table>

Note: During the input routines the 'T' connection is determined to be either a branch or a mixer and is assigned the appropriate code.

STEAM GENERATOR

The steam generator, as the name implies, is a device used to generate steam. This could be anything from a simple boiler to a nuclear reactor. In general, a steam generator may have several independent flowpaths through it but must have at least one.
These multiple flowpaths are differentiated in the program by the relative positions of the nodal connections in the boiler input record. Each flowpath has but one inlet and one outlet. In the input record, the inlet number for any flowpath is given first, followed by the outlet number for that same flowpath. This alternating property allows the computer to handle multiple flowpaths at varying pressures. See the diagram below for examples.

```
Node  Connections  Inlet/Outlet  Node  Connections  Inlet/Outlet
3     1, 2        1, 0          5     1, 2, 3, 4    1, 0, 1, 0
```

**STEAM TURBINE**

The Steam Turbine is a device used to convert high pressure, high temperature steam into work. This process will cause the outlet pressure to be lower than the inlet pressure.

In general, a turbine may have several injection or extraction
ports at intermediate pressures. Because of the way the turbine is solved, all inlets and outlets must be entered in the order of their physical placement on the turbine beginning with the initial inlet and ending with the final outlet.

The pre-processor of the program will take the data for a multiple injection/extraction turbine and create several single inlet/outlet turbines with branches and mixers placed throughout the flowpath to represent an injection or extraction port.

Never directly connect a multistage turbine to a condenser that has multiple inlets. Rather, place a dummy node in the flowpath that would ordinarily connect the two devices. Unless this procedure is strictly adhered to, errors may arise in the pre-processing phase. These errors are due to the fact that the data for both a multistage turbine and multiple inlet condenser are modified during this stage and the original connections between the two devices may be lost.
Some examples of both the data representation and the preprocessor action for a turbine are given below.

**INITIAL DESCRIPTION**

<table>
<thead>
<tr>
<th>Node</th>
<th>Connections</th>
<th>Inlet/Outlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1, 2, 3, 5, 4</td>
<td>1, 0, 0, 1, 0</td>
</tr>
</tbody>
</table>

**AFTER PRE-PROCESSING**

<table>
<thead>
<tr>
<th>Node</th>
<th>Connections</th>
<th>Inlet/Outlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>1, 1002</td>
<td>1, 0</td>
</tr>
<tr>
<td>1002</td>
<td>1001, 2, 1003</td>
<td>1, 0, 0</td>
</tr>
<tr>
<td>1003</td>
<td>1002, 1004</td>
<td>1, 0</td>
</tr>
<tr>
<td>1004</td>
<td>1003, 3, 1005</td>
<td>1, 0, 0</td>
</tr>
<tr>
<td>1005</td>
<td>1004, 1006</td>
<td>1, 0</td>
</tr>
<tr>
<td>1006</td>
<td>1005, 5, 1007</td>
<td>1, 1, 0</td>
</tr>
<tr>
<td>1007</td>
<td>1006, 4</td>
<td>1, 0</td>
</tr>
</tbody>
</table>
The condenser will take several inflow streams and reject heat in such a way that the single outflow stream will be saturated liquid at the condenser pressure. At least one inlet must be specified. If the condenser contains more than one inlet, the pre-processor will modify the condenser data such that all the inlets will flow into an ideal mixing bin. The outflow from this mixer will then flow into a single inlet condenser. The outlet from this condenser will be the same as the outlet from the original condenser. An example is shown below.

*Never directly connect a multiple inlet condenser to a multistage turbine.* Rather, place a dummy node in the flowpath that would ordinarily connect the two devices. Unless this procedure is strictly adhered to, errors may arise in the pre-processing phase. These errors are due to the fact that the data for both a multistage turbine and
multiple inlet condenser are modified during this stage and the original connections between the two devices may be lost.

All inflow streams must be at the same pressure or errors may arise. Unless otherwise specified, the program will assume that the outlet conditions are that of a saturated liquid at the condenser pressure.

**PUMP**

The pump is used simply to raise the pressure of the fluid flowing through it. There may be only one inlet and one outlet associated with a pump.

**FEEDWATER HEATER**

The feedwater heater is a closed device allowing two separate flows to pass through it so that some heat transfer may occur from one flow to the other. Because the two flows do not come into direct contact, each may be at a different pressure from the other.

When describing a heater it is necessary to include the Terminal
Temperature Difference between the two outlet streams. This quantity is the difference in temperature between the 'hot' flow and the 'cool' flow. This quantity may have a positive, negative, or zero value. Unless otherwise specified, the terminal temperature difference is given the value of zero.

This program will only handle heaters where the outlet stream for the 'hot' flow is either pumped directly back into the line or flashed to the condenser.

A branch is used to split a single flow into several flows. Hence the branch may have only one inlet but may have several outlets.
All state properties are equal across a branch; only the mass flow differs.

**MIXER**

A mixer is used to mix several flows at the same pressure, and discharge the mixture from a single outlet. The mixing process is assumed to be adiabatic. If the pressures of the mixing flow vary, an error message will be printed at the end of the run.

**THROTTLE**

A throttle is used to decrease the pressure from the inlet to the outlet while the specific enthalpy remains constant throughout. The process is also assumed to be adiabatic.

There may be only one inlet and one outlet associated with a throttle.

**'T' CONNECTION**

A 'T' connection is used to describe where in the network various flows come together. It is modeled as either a branch or a mixer.
depending on the number of inlets and outlets.

This device may have either one inlet and several outlets or several inlets and one outlet. **Multiple inlets and outlets for a single device are not allowed.**

**DUMMY NODE**

The dummy node is simply a device with one inlet and one outlet in which all properties remain constant throughout. This device would be used in a network where two separate states may have identical preceding and succeeding nodes. However, the program requires that each state have a unique combination of preceding and succeeding nodes. The dummy node may then be used as a non-active device to prevent any redundancy of nodal connections that might occur.

This device should also be used to connect a multistage turbine to a multiple inlet condenser. That is, rather than having the outlet flow of a multistage turbine flow directly into a multiple inlet condenser, have the outlet flow into a dummy node and then the dummy's outlet flow will flow to the inlet of the condenser.
APPENDIX D

The naming convention for the various files is described below.

Also given are the record layouts for each file listed.

- **session-ID.INIT.NODE.DATA**: Contains the initial network data for the session.
- **session-ID.INIT.STATE.DATA**: Contains the initial property data for the session.
- **session-ID.WORK.NODE.DATA**: Contains the modified network data for the session.
- **session-ID.WORK.STATE.DATA**: Contains the modified property data for the session.
- **session-ID.FINAL.NODE.DATA**: Contains the final (solved) network data for the session.
- **session-ID.FINAL.STATE.DATA**: Contains the final (solved) property data for the session.
- **session-ID.LENGTH.DATA**: Contains the number of active records in each of the initial and final data files.
- **session-ID.PRINT.FILE**: This is the file that is created when the final data is printed out.

The record layouts for each of these files may be found on the following pages. Each layout supplies the position in the record of each data field, the program name associated with that field, and a description of each field.

All of the files used by the program are direct files (also known as random access files). These differ from sequential files in that any
record may be read from or written to the files in any order, i.e., at random.
The variable 'WORK' indicates the work or heat input to or output from the device. It is also used to represent the terminal temperature difference (TTD) of a feedwater heater. This variable is always positive, except in the case of the TTD where it may take on any value, and its exact meaning is dependant on the device in question. 'EFF' represents the device efficiency and is a value from 0.0 to 1.0. If it is not specified during the input routines, it will default to a value of 1.0. The value of 'KK' will always equal the sum of 'IN' and 'IOUT'. The delete flag 'IDEL' is equal to zero(0) for an active record, and is equal to one(1) for a record that has been deleted.

Note: The placement of the nodal connections and the inlet/outlet specifications is not absolutely defined for each device, but is a function of the value of KK.
The variables 'IFROM' and 'ITO' uniquely determine the state. The delete flag, 'IDEL', is equal to zero(0) for an active record and equal to one(1) for a record that has been deleted. The value of 'ISLV' may range from zero to sixty three and indicates which state properties are known. See the end of this appendix for a listing of the values of 'ISLV'. Any of the other properties listed above which are not known are given a value of negative one.
The variable 'WORK' indicates the work or heat input to or output from the device. This variable is always positive and its exact meaning is dependant on the device in question. 'EFF' represents the device efficiency and is a value from 0.0 to 1.0. If it is not specified during the input routines, it will default to a value of 1.0. The value of 'KK' will always equal the sum of 'IN' and 'IOUT'. 'IOLD' is equal to the value of 'INODE' associated with the device in the initial file.

Note: The placement of the nodal connections and the inlet/outlet specifications is not absolutely defined for each device, but is a function of the value of KK.
The variables 'IFROM' and 'ITO' uniquely determine the state. The value of 'ISLV' may range from zero to sixty three and indicates which state properties are known. 'IFOLD' and 'ITOLD' have the values of 'IFROM' and 'ITO' associated with the same state in the initial file. If there is no corresponding state in the initial file, the values of 'IFOLD' and 'ITOLD' are zero. Any of the other properties listed above which are not known are given a value of negative one.
The variable 'WORK' indicates the work or heat input to or output from the device. This variable is always positive and its exact meaning is dependant on the device in question. 'EFF' represents the device efficiency and is a value from 0.0 to 1.0. If it is not specified during the input routines, it will default to a value of 1.0. The value of 'KK' will always equal the sum of 'IN' and 'IOUT'.

Note: The placement of the nodal connections and the inlet/outlet specifications is not absolutely defined for each device, but is a function of the value of KK.
session-id.FINALSTATE.DATA

File Device #: 22
Record Length: 110

<table>
<thead>
<tr>
<th>from</th>
<th>to</th>
<th>name</th>
<th>format</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>IFROM</td>
<td>I4</td>
<td>Preceding node</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>ITO</td>
<td>I4</td>
<td>Succeeding node</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>ISLV</td>
<td>I4</td>
<td>Code specifying known properties</td>
</tr>
<tr>
<td>13</td>
<td>24</td>
<td>AP</td>
<td>E12.6</td>
<td>Pressure</td>
</tr>
<tr>
<td>25</td>
<td>36</td>
<td>AT</td>
<td>E12.6</td>
<td>Temperature</td>
</tr>
<tr>
<td>37</td>
<td>48</td>
<td>AH</td>
<td>E12.6</td>
<td>Specific Enthalpy</td>
</tr>
<tr>
<td>49</td>
<td>60</td>
<td>AS</td>
<td>E12.6</td>
<td>Specific Entropy</td>
</tr>
<tr>
<td>61</td>
<td>72</td>
<td>AQ</td>
<td>E12.6</td>
<td>Quality</td>
</tr>
<tr>
<td>73</td>
<td>84</td>
<td>AM</td>
<td>E12.6</td>
<td>Mass Flow Rate</td>
</tr>
<tr>
<td>85</td>
<td>96</td>
<td>AV</td>
<td>E12.6</td>
<td>Specific Volume</td>
</tr>
</tbody>
</table>

The variables 'IFROM' and 'ITO' uniquely determine the state. The value of 'ISLV' may range from zero to thirty three and indicates which state properties are known. Any of the other properties listed above which are not known are given a value of negative one.
**session-id.LENGTH.DATA**

File Device #: 18  
Record Length: 17

<table>
<thead>
<tr>
<th>from</th>
<th>to</th>
<th>name</th>
<th>format</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>LOF10</td>
<td>I4</td>
<td>Length of Initial Network File</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>LOF20</td>
<td>I4</td>
<td>Length of Initial Property File</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>LOF12</td>
<td>I4</td>
<td>Length of Final Network File</td>
</tr>
<tr>
<td>13</td>
<td>16</td>
<td>LOF22</td>
<td>I4</td>
<td>Length of Final Property File</td>
</tr>
<tr>
<td>17</td>
<td>17</td>
<td>IUNIT</td>
<td>I1</td>
<td>Units (S.I. or British)</td>
</tr>
</tbody>
</table>

The four 'LOF' variables give the number of active records in the respective file. The variable 'IUNIT' is determined during the initial data entry for a session. Its value is zero(0) if British units are used, and is equal to one(1) if S.I. units are used.
session-id.PRINT.FILE

File Device #: 15
Record Length: 137

This file is a sequential file that contains all the printed output from the analysis run. In addition, any errors that occurred will be written to this file. The contents may be viewed from the terminal by loading the file into one of the terminal editors, such as VFS or QED.
VALUES TAKEN ON BY THE VARIABLE 'ISLV'

1 - Pressure
2 - Temperature
4 - Enthalpy
8 - Entropy
16 - Sp. Volume
32 - Mass Flow Rate

Each of the above properties is given a unique value that corresponds to a certain power of two. The value of 'ISLV' is determined by adding together the values for all the properties that are known.

For example, if pressure alone is known, 'ISLV' would take on the value of one. If pressure and mass flow were both known, 'ISLV' would be equal to 33 (1 + 32). If all properties are known, 'ISLV' will be equal to 63.

Because of the binary nature of the assigned values and the special properties associated with binary numbers, each different combination of values will result in a unique sum.
### APPENDIX E

**VALID UNITS FOR THE ANALYSIS PROGRAM**

#### S.I. UNITS

<table>
<thead>
<tr>
<th>Property</th>
<th>S.I. Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure</td>
<td>Mega Pascals</td>
</tr>
<tr>
<td>Temperature</td>
<td>Degrees Celsius</td>
</tr>
<tr>
<td>Enthalpy</td>
<td>Kilo Joules per Kilogram</td>
</tr>
<tr>
<td>Entropy</td>
<td>Kilo Joules per Kilogram-Degree Celsius</td>
</tr>
<tr>
<td>Specific Volume</td>
<td>Cubic Meters per Kilogram</td>
</tr>
<tr>
<td>Mass Flow Rate</td>
<td>Kilograms per Second</td>
</tr>
<tr>
<td>Work/Heat</td>
<td>Kilo Joules per second</td>
</tr>
</tbody>
</table>

#### BRITISH UNITS

<table>
<thead>
<tr>
<th>Property</th>
<th>British Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure</td>
<td>Pounds per Square Inch (PSI)</td>
</tr>
<tr>
<td>Temperature</td>
<td>Degrees Fahrenheit</td>
</tr>
<tr>
<td>Enthalpy</td>
<td>BTU per Pound Mass</td>
</tr>
<tr>
<td>Entropy</td>
<td>BTU per Pound Mass- Degree Fahrenheit</td>
</tr>
<tr>
<td>Specific Volume</td>
<td>Cubic Feet per Pound Mass</td>
</tr>
<tr>
<td>Mass Flow Rate</td>
<td>Pounds Mass per Second</td>
</tr>
<tr>
<td>Work/Heat</td>
<td>Btus per second</td>
</tr>
</tbody>
</table>
### APPENDIX F

**LISTING AND DESCRIPTION OF SUBPROGRAMS**

#### NETWORK DATA ENTRY

<table>
<thead>
<tr>
<th>Network</th>
<th>Data</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETWRK</td>
<td>INPUT1</td>
<td>CLARA</td>
</tr>
<tr>
<td>CHANGE</td>
<td>SCREEN</td>
<td>DELETE</td>
</tr>
<tr>
<td>DISP1</td>
<td>SORT</td>
<td></td>
</tr>
</tbody>
</table>

#### PROPERTY DATA ENTRY

<table>
<thead>
<tr>
<th>Property</th>
<th>Data</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATIN</td>
<td>SCREEN2</td>
<td>CHANG2</td>
</tr>
<tr>
<td>DELET2</td>
<td>DISP2</td>
<td>SORT</td>
</tr>
</tbody>
</table>

#### PRINT INITIAL DATA

<table>
<thead>
<tr>
<th>Print</th>
<th>Initial</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEAD</td>
<td>PRNODE</td>
<td>PRSTAT</td>
</tr>
<tr>
<td>SORT</td>
<td>REORG</td>
<td></td>
</tr>
</tbody>
</table>

#### PRE-PROCESSOR ROUTINES

<table>
<thead>
<tr>
<th>Pre-Processor</th>
<th>Routine</th>
<th>Routine</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDIT</td>
<td>SRCH1</td>
<td>SRCHST</td>
</tr>
<tr>
<td>BLDNOD</td>
<td>SPLTRB</td>
<td>SPLCON</td>
</tr>
<tr>
<td>BLDSTA</td>
<td>STATE</td>
<td>FIXNOD</td>
</tr>
<tr>
<td>ORDER</td>
<td>EQUATE</td>
<td>EQUIAP</td>
</tr>
<tr>
<td>EQUAH</td>
<td>EQUIAV</td>
<td>EQUIAT</td>
</tr>
<tr>
<td>EQUAS</td>
<td>EQUIAQ</td>
<td>EQUAM</td>
</tr>
<tr>
<td>EQUESTA</td>
<td>GETST1</td>
<td>GETST2</td>
</tr>
</tbody>
</table>

#### PROCESSOR ROUTINES

<table>
<thead>
<tr>
<th>Processor</th>
<th>Routine</th>
<th>Routine</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO</td>
<td>GETNOD</td>
<td>GETSTA</td>
</tr>
<tr>
<td>CKJSTK</td>
<td>TURB</td>
<td>BOILER</td>
</tr>
<tr>
<td>CONDEN</td>
<td>PUMP</td>
<td>FDHTR</td>
</tr>
<tr>
<td>MIXER</td>
<td>BRANCH</td>
<td>THROTTL</td>
</tr>
<tr>
<td>STEP</td>
<td>DUMMY</td>
<td>QUAL</td>
</tr>
<tr>
<td>PUTFIL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### POST-PROCESSOR ROUTINES

<table>
<thead>
<tr>
<th>REASM</th>
<th>PRINTF</th>
<th>HEAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFNOD</td>
<td>PFSTAT</td>
<td>SORT</td>
</tr>
</tbody>
</table>

### RE-PRINT FINAL DATA

<table>
<thead>
<tr>
<th>HEAD</th>
<th>PFNOD</th>
<th>PFSTAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### UTILITY ROUTINES

<table>
<thead>
<tr>
<th>CHAIN</th>
<th>CHECK</th>
<th>CKPROP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEAR</td>
<td>DEVICE</td>
<td>FILSZ</td>
</tr>
<tr>
<td>PAUSE</td>
<td>PUTFZ</td>
<td></td>
</tr>
</tbody>
</table>
Subroutine NETWRK is a very simple and straightforward routine. It displays the Network Input Menu on the screen and sets the device code after a selection has been made. It then calls subroutine INPUT1, passing to it the device code. The routine returns to the main network processor only when option 'X' (Return to Network Main Menu) is selected from the Input Menu.

Subroutine INPUT1 handles all the data entry for each device. There are several entry routines that are used to handle specific prompts for different sets of devices. The devices are grouped into these sets depending on the configuration of their inlets and outlets. For instance, the pump, throttle, and dummy node comprise one group because each of these devices has only one inlet and one outlet. A second group is made up of the devices where the various flows through them must remain separate. This group includes the feedwater heater and steam generator.

The first prompt that is displayed by this routine is to enter the nodal value for the device in question. Before any more data entry is allowed, a check is first made to determine if the nodal value entered already exists in the network file. If it does not, the rest of the input prompts will follow. However, if the node is found in the file, all data associated with that node is displayed on the screen and you
are given the options to change any of the data, delete the record entirely, or leave it as is.

After all data has been entered for a device, it is written to the network file in the position following the last device entered. If this is the first device to be entered, the data will be written to record number one.

A limitation on the devices is that each will only handle up to fifteen (15) connections. That is, the sum of the inlets and the outlets must be less than or equal to fifteen. Also, the nodal value assigned to a device must be between 1 and 999 inclusive. This restriction is necessary due the special processing that takes place with certain devices such as turbines and condensers.

**CLARA**

This routine is used to set both the nodal connection array (IGEO) and the inlet/outlet array (INOUT) to zeros before a record is read from the network file.

**CHANGE**

This subroutine is called so that changes may be made to existing data in the network file. It will display the Network Change Menu from which a specified node or record number may be called up and displayed on the screen. Figures 6 illustrates the appearance of this menu.
SCREEN

This routine will display on the screen all network data for a given record number. The routine also controls the options that are displayed with the data. These options allow one to delete an entire record, change any item associated with the record, or to leave it as is. See Figure 7 for an illustration of this display.

DELETE

This routine is used to display the Network Delete Menu. This menu allows the deletion of either a specified node or record number (see Figure 9). Before the record is deleted, all data associated with it is displayed in a format similar to the change screen (see Figure 10).

A prompt will ask if this node is to be deleted. A negative response will leave the record intact and return control to the Network Delete Menu. If the node is to be deleted, respond to the prompt by answering 'Y'. This will set the delete flag equal to one and the nodal value equal to negative one. The record is not physically deleted from the file until the subroutine REORG is run during the initial print routine.

DISP1

This routine is called from the Network Main Menu by selecting
option 'E' at any time during data entry. It will read the initial network file, sort all records by nodal value and device code, and then display all the node numbers and their descriptions. Only the active records will be displayed.

**SORT**

The SORT subroutine will sort a two dimensional array using two levels of sort. The array has 1000 rows and 3 columns. Column two contains the primary sort keys and column three contains the secondary sort keys. The record numbers for the respective keys are kept in column one.

The routine uses a simple bubble sort to sort the keys into ascending order.

**STATIN**

This routine is used to enter the various known states and properties. It is called by selecting option 'A' on the Property Main Menu. After entering the preceding and succeeding nodes that describe the state, a small menu will appear for the entry of the properties associated with that state (see Figure 12).

When entering known properties do not enter any more data than is necessary to describe a state. For instance, if pressure, temperature, and enthalpy are all known for a state, enter only two of the
quantities, such as pressure and temperature. The program will solve for the rest of the state properties internally.

If the state entered above already exists in the property file, the existing data is displayed and you have the option to change any data, delete the entire record, or leave as is.

After the entry of a state and its properties, a prompt appears asking if there are any more states to enter. Answer with a 'Y' to continue, or an 'N' to halt data entry. If a 'Y' is entered inadvertently when you have no more data, do NOT use the BREAK key. Instead, simply enter a fictitious state. You can then go back using the Property Delete Menu and delete that state. When an 'N' is entered in response to the prompt, control passes back to the Property Main Menu. If more data remains to be entered, simply select option 'A' on this menu.

**SCREN2**

Subroutine SCREN2 is used to display the data for any state record on the screen. It also controls the various options used to change any data items, delete the record, or leave as is.

**CHANG2**

This routine will display the Property Change Menu and is used to change a specified state or record number from the property file. It
calls subroutine SCREN2 to handle the display and any changes that need to be made.

**DELET2**

This routine will display the Property Delete Menu and allows any state or record number from the property file to be deleted. The data for the state is first displayed followed by a prompt asking if this is the record to delete. Respond with an 'N' to bypass the delete process and return to the Delete Menu.

If a 'Y' is entered, the delete flag is set equal to one and both the preceding and succeeding nodes of the state are set equal to negative one. The record is not physically deleted from the file until subroutine REORG is called during the initial print routine.

**DISP2**

This routine is called from the Property Main Menu by selecting option 'E' and is used to display all active states and a few properties of each state. It begins by reading the property file, ignoring all records with a delete flag of one. Each active state is placed into an array which is then sorted. The sorted states are then displayed along with their respective pressures, temperatures, and mass flow rates.
HEAD

This routine will skip to the top of the next page and print the appropriate headings when printing the network and property files.

Because the routine may be called from either PRNODE, PRSTAT, PFNOD, or PFSTAT, a parameter is needed to specify which routine called so that the correct heading will be written. A parameter code of one(1) indicates that the network heading should be written, while a code of two(2) will write out the property heading.

PRNODE

This subroutine is used by the initial data print routine. It reads all the active network records and prints the data in a clear, formatted manner. A call is first made to subroutine SORT to sort the data by node number.

This routine increments a line counting variable every time a line of data is printed. Whenever the line counter reaches the end-of-page value, it is reset and subroutine HEAD is called to skip to a new page and write the headings.

PRSTAT

This subroutine is called by the initial data print routine. It performs a similar function to PRNODE except the records are read from the initial property file. The routine also sorts the data by
and succeeding nodes.

As in PRNODE the line count variable is incremented accordingly. When the end-of-page value has been reached, a call is made to subroutine HEAD.

**REORG**

This routine reorganizes the data in the initial network or property files. It steps through the files and simply rewrites each record not flagged for deletion into a separate file. Any records that are flagged for deletion (a delete code equal to one) are not copied to the new file and are permanently lost.

Since only data in the initial data files may be selectively deleted, this routine is called only by the program that prints the initial data.

**EDIT**

This is the first subroutine called by the pre-processor. It performs a check to be certain that all nodes and states are accountable. If, for instance, you had entered a nodal connection for which there was no nodal record, the network description would be incorrect. Similarly, if a state were entered for which there were no corresponding preceding or succeeding nodal record, an error would arise.
The method used to determine if there are any extraneous nodes is a simple one and is described as follows. Each nodal record is read sequentially from the initial network file. All states associated with that node are then assembled and placed into an array. Because every state must have two nodes associated with it—a preceding and a succeeding node—there should be exactly one duplication for every state in the array once all the nodal records have been read. That is to say there should be two and only two entries in the array for each state.

Once the array has been assembled it is checked through to be certain that the above premise does indeed hold true. If there are more or less than two occurrences of a state in the array, an error message is printed out.

After this first phase is complete the second edit phase begins. Each state is read from the property file one by one. The array is then searched until that state is found and the processing continues. If the state cannot be found in the array, an error message is again generated and the edit continues until all state records have been read.

If no errors occurred the pre-processor will continue on. If any errors were present they are printed out and the processing ends with control passing to the Main Menu.

SRCH1

This routine is called by subroutine EDIT to determine the number
of occurrences of a particular state in the 'edit array'. The state in question is passed as a parameter, and the number of occurrences is returned to the calling program.

**SRCHST**

This routine is called by subroutine EDIT to determine if a given state exists at all in the 'edit array'. A return code of one signals that the state was found in the array, while a return code of zero means that it was not.

**BLDNOD**

The routine BLDNOD reads each record from the initial network file and checks both the delete flag and the device code for that record. If the delete flag is equal to one, the record is ignored and the next record is read.

If the device code does not signify either a steam turbine or a condenser, the record is written to the network work file. Whenever a turbine or a condenser is encountered, special routines are called which manipulate the original data into a form usable to the processor. The two special routines called are SPLTRB for the turbine, and SPLCON for the condenser.
SPLTRB

When called by BLDNOD, this routine first determines if the steam turbine has any intermediate injection or extraction ports. If not, the turbine record is written to the work file without any change.

However, if there are any injection or extraction ports, the original turbine is 'split' into a series of single inlet, single outlet turbines and either branches or mixers. Take, for example, a turbine with one extraction port. A new set of devices will be generated consisting of two single inlet, single outlet turbines separated by a branch.

Since several new devices will be generated, each must be given a nodal value as yet undetermined. The new nodal values given to these new devices are all greater than 1000. The current value for this new node is accessed through a common block. It begins at a value of 1001 and is incremented by one every time a new device is written out.

In addition, the nodal value of the original device is written into a special field of each new device (IOLD). This allows easy reassembly of the original device by the post-processor.

SPLCON

This routine is called by subroutine BLDNOD whenever a condenser record has been read. If the condenser has only one inlet and one outlet, the record is written to the network work file unaltered.
However, if there is more than one inlet, the condenser is broken down into two new devices - a mixing bin and a single inlet, single outlet condenser.

The inlets to the mixer are the inlets to the original condenser. The mixer outlet flows into the new condenser. The outlet of the new condenser and the old condenser are the same.

As in the case of the turbine these two new devices have nodal values greater than 1000. Each record also has a field which contains the nodal value of the original condenser (IOLD).

**BLDSTA**

This routine reads each record from the initial property file. If the delete flag is equal to one the record is ignored and another is read. Otherwise the record is written to the property work file.

When all records have been read, the subroutine STATE is called to write out all possible states for the network.

**STATE**

Subroutine STATE is called by subroutine BLDSTA after all active states in the initial property file have been written to the property work file. This routine will read each record in the initial network file, assemble all states associated with each node, and write these states to the property work file.
Before any state is written out, however, the property work file is searched to determine if that state already exists in the file. If it does, it is not written out and the next state value is checked. Otherwise the new state is added to the end of the property work file.

**FIXNOD**

This routine is called after BLDNOD and BLDSTA have completed. Recall that during the BLDNOD phase several new devices may have been generated, each with a nodal value greater than 1000. However, the connections to these new devices still reflect a connection to the old node, as do all the states.

This routine will step through the work files and alter the necessary records so that these new devices are properly connected to the rest of the network. See Appendix G for an example.

**ORDER**

This routine steps through the network work file and creates an 'ordered' stack which is used to step through the program in a logical fashion. As each node is read, its succeeding node is written to the array ISTACK. If there is more than one succeeding node, those remaining are placed in a 'holding' array until they are needed later on.

This process continues until the next succeeding node is equal to
our original starting node in the first element of ISTACK. At this point the last element in the 'holding' array is moved to ISTACK. The same processing described above then repeats. This continues on until all network records have been read and there are no more elements in the 'holding' array.

See Appendix H for several examples.

**EQUATE**

Subroutine EQUATE is used to 'equate' certain properties across various devices. It is first called during the pre-processor phase to establish equivalent properties at various states given only a few known properties at a few states. It is also called from the processor routine each time a new property is calculated.

This routine reads a state record and determines whether any of the properties are known. If none are known the next record is read. This continues until all property records have been read. If any properties are known at that state, the appropriate solution routines are then called to possibly equate these properties to other states.

**EQUAP**

This routine is called by subroutine EQUATE whenever the pressure for a state is known. It will then step both forward and backward through the network file and equate any unknown pressures to this known
value. Only the pressures that lie across certain devices are equated. These devices include the steam generator, condenser, branch, mixer, feedwater heater, and dummy node.

It should be noted that for devices with multiple flows (feedwater heater and steam generator) only the pressures for an individual flow are equated.

**EQUAH**

This routine is called by subroutine EQUATE whenever the enthalpy for a state is known. It will then step both forward and backward through the network file and equate any unknown enthalpies to this known value. Only the enthalpies that lie across certain devices are equated. These devices include the branch, throttle, and dummy node.

**EQUAV**

This routine is called by subroutine EQUATE whenever the specific volume for a state is known. It will then step both forward and backward through the network file and equate any unknown volumes to this known value. Only the volumes that lie across certain devices are equated. These devices include the branch and dummy node.

**EQUAT**

This routine is called by subroutine EQUATE whenever the
temperature for a state is known. It will then step both forward and backward through the network file and equate any unknown temperatures to this known value. Only the temperatures that lie across certain devices are equated. These devices include the branch and dummy node.

**EQUAS**

This routine is called by subroutine EQUATE whenever the entropy for a state is known. It will then step both forward and backward through the network file and equate any unknown entropies to this known value. Only the entropies that lie across certain devices are equated. These devices include the branch and dummy node.

**EQUAQ**

This routine is called by subroutine EQUATE whenever the quality for a state is known. It will then step both forward and backward through the network file and equate any unknown qualities to this known value. Only the qualities that lie across certain devices are equated. These devices include the branch and dummy node.

**EQUAM**

This routine is called by subroutine EQUATE to equate the mass flow rate at various states. If a device contains multiple flows only the mass flows for each individual flow are equated.
EQUSTA

This routine is called by each of the Equate Property routines. EQUSTA does the actual equating of properties. After the equate has been made, the routine calls subroutine CKPROP to obtain a new property code for the state, and to solve for all other state properties if possible.

GETST1

This routine is called by each of the equating subroutines when assembling the states associated with a device. A node and a single state associated with that node are passed to the routine as arguments. The subroutine will then return all of the other states (excluding the one that was passed down) to the calling routine.

This routine is called when the nodal device in question has only one flow through it, such as a condenser or mixer.

GETST2

This routine is called by each of the equating subroutines and is similar in nature to subroutine GETST1. However, this routine is called whenever the device in question may have multiple flows through it. Such devices include steam generators and feedwater heaters.

In this instance the routine will return all states associated with the flow to which the arguments passed are connected.
This is the driving routine for the processor. It begins by copying the array ISTACK (created by subroutine ORDER) into the array JSTACK. The first column of JSTACK will thus contain all the nodes in the network. The second column will contain a completion code for that node, which will initially be zero.

The code specifies what data is known. A code of zero indicates that not all properties associated with the device are known. A code of one indicates that all properties except mass flow associated with the device are known. Finally, a code of two indicates that all properties associated with the device are known, as well as the work or heat of the device.

The routine then steps through the array JSTACK and checks the completion code. If the code is equal to two, the next array element is read. If the code is equal to zero or one the appropriate solution routine is called for that node.

The return code from the solution routine is then placed in the second column of JSTACK in the row corresponding to the node just processed. Subroutine CKJSTK is then called to determine the occurrence of each code in JSTACK. When the number of code two's is equal to the length of the array, the network is completely solved and the post-processor takes over. Otherwise the next node in JSTACK is read and the process continues.

After the last element of JSTACK has been read, the pointer is
reset to the top of the stack and the entire procedure is repeated. To
prevent the possibility of an infinite loop resulting from an insoluble
problem, a limit of five is set on the number of times the stack pointer
may be reset.

GETNOD

Given the record number for a device, this routine will assemble
all states associated with that device into an array. In addition, the
routine will call subroutine GETSTA to place all the properties for each
state into another array.

There are two sets of each of the above arrays. One contains all
the inlet states and the other contains all the outlet states.

GETSTA

This routine is called by subroutine GETNOD and will place all the
properties for a given state into the appropriate inlet and outlet
arrays.

CKJSTK

This routine is called by subroutine GO to determine the occurrence
of the various completion codes present in the array JSTACK. It begins
by setting all counters to zero and then steps through the array one
element at a time. Whenever a particular code is encountered, the
corresponding counter is incremented by one. This process continues until all elements of the array have been read.

**TURB**

This subroutine is called by subroutine GO to solve a steam turbine. There are several routines that can be used to solve the turbine depending on what data is known.

Given both the inlet and outlet states and the mass flow rate, the expansion work and the turbine efficiency are calculated. If the inlet state is known as well as the outlet pressure, the outlet state may be found. If the mass flow is also known the turbine work may be calculated.

Given either the inlet or outlet state and the expansion work, the other state may be found.

**BOILER**

This subroutine will solve a steam generator and all states associated with it. There are three routines used to solve three types of problems. The first is given the enthalpies and mass flows of all the states to calculate the heat addition to the boiler.

The second routine will calculate one state's enthalpy given the enthalpies for all other states, all mass flows, and the heat added to the boiler.
The final routine used will calculate the remaining mass flow given the mass flow rate at every other state, all state enthalpies, and the heat addition.

CONDEN

This routine will solve the unknowns of a condenser for the following situations: given the inlet and outlet states and the mass flow, calculate the heat rejected; given the inlet and outlet enthalpies and the mass flow, calculate the heat rejected; given the inlet state, the heat rejected, and the mass flow, calculate the outlet state; given the inlet enthalpy, heat rejected, and mass flow, calculate the outlet enthalpy; and finally, given the inlet state and the mass flow, calculate the outlet state and the heat rejected. This last calculation is made on the assumption that the outlet state is a saturated liquid.

PUMP

This subroutine is called by subroutine GO to solve a pump. There are several routines that can be used to solve the pump depending on what data is known.

Given both the inlet and outlet states and the mass flow rate, the compression work and the pump efficiency are calculated. If the inlet state is known as well as the outlet pressure, the outlet state may be found. If the mass flow is also known the compression work may be calculated.
Given either the inlet or outlet state and the compression work, the other state may be found.

**FDHTR**

This subroutine is called to solve a closed feedwater heater. The routine first attempts to solve the device by itself. If this is unsuccessful after several tries, the heater is solved in conjunction with a particular mixing bin.

The correct mixer is found by stepping through the network with each of the heater flows until a common mixer is found. Because of this limitation, only heaters with flows that are pumped back to the line or flows that are flashed to the condenser may be handled.

This routine calls subroutine STEP to do the actual searching for the mixer common to both flows.

Each heater may have only two flows through it, with each flow having only one inlet and one outlet. In the data entry routines and throughout the literature the two flows are referred to as the 'hot' and 'cold' flows. As the names imply, heat transfer occurs from the 'hot' flow to the 'cold' flow.

**MIXER**

This subroutine is called to solve a mixing bin. One solution routine will solve for any two mass flows given all other mass flows and
the enthalpies at every state.

The second solution routine will solve for any enthalpy and any mass flow given the enthalpies and mass flows for all the other states.

**BRANCH**

The BRANCH routine will simply equate the unknown state properties to any known properties. In addition, if all the mass flows but one are known, that unknown mass flow is calculated. This routine assumes an adiabatic branching of fluid, thus the heat in or out is equal to zero.

**THROTL**

This routine will equate the enthalpies of the inlet and outlet states, if either one is known. Again the process is assumed to be adiabatic.

**DUMMY**

The dummy node routine simply equates any unknown properties of either state to the known properties of the other state. There is no energy transfer involed, nor is there any mixing or branching of fluid.

The dummy node is used in the network merely to prevent two separate states from having the same preceding and succeeding node.
QUAL

The routine is a function subprogram used to calculate the quality of a state given the saturation pressure, a state quantity (enthalpy, entropy, or specific volume), and a code specifying which quantity is used. A code of one(1) indicates enthalpy, two(2) indicates entropy, and three(3) indicates specific volume.

This routine is only valid when it has been previously established that the state lies in the saturated region.

PUTFIL

This routine is called after each solution routine to write the device and property data back to the work files. It is called only if the return code from the solution routine is greater than zero.

STEP

This routine is called by subroutine FDHTR to find a mixer (if one exists) that is common to both feedheater flows.

REASM

This is the initial routine in the post-processor, used to reassemble the network and property data into a form similar to that in the initial files. It reads through both the network and property work
files and writes certain records to the corresponding final files.

All nodes greater than 1000 are returned to their original values and any devices which were previously 'split up' are put back together.

PRINTF

Subroutine PRINTF is called after the work files have been reassembled and written to the final files. This routine is simply a driver to control the printing of data from the final files.

PFNOD

This routine is called by PRINTF to print the final network file. It is similar in operation to the subroutine PRNODE.

PFSTAT

This routine is called by PRINTF to print the final property file. It is similar in operation to the subroutine PRSTAT.

CHAIN

This routine is used to find the record number of a given key value. In the case of the network files - initial, work, and final - the key is a nodal value. For the property files - initial, work, and final - there are two keys needed. The first key is the preceding node
for a state, and the second key is the succeeding node. The key(s) and file device number (see Appendix D) are passed to the CHAIN routine whenever it is called.

The subroutine searches the appropriate file until it finds the record specified by the key(s). For example, if the device number of 10 and key value of 3 were passed to the subroutine, the initial network file would be searched until node number 3 was found. The routine then returns the record number for the key(s) and a return code of one to the calling program. If the key value is not found in the file a return code of zero is returned. The search used is a sequential search beginning with record number one.

CHECK

The various state properties - pressure, temperature, enthalpy, entropy, specific volume, and mass flow rate - are passed to this subroutine and a code, specifying which properties are known, is returned. The code results from a binary addition scheme.

Each property is assigned a binary code. For instance, pressure is one, temperature is two, enthalpy is four, and so on down the line. As each property is determined to be known, its corresponding binary code is added to the running total. A property is said to be known if its value is non-negative.

After examining all properties, the value of the final code is returned. It will range from zero, for no properties known, to sixty
three, for all properties known.

**CKPROP**

Whenever enough properties are known for a state, subroutine CKPROP will solve for all the other properties of that state. It first calls subroutine CHECK to determine what properties are known. The appropriate solution routine is then used to solve for the remaining properties. Before returning, a second call is made to subroutine CHECK to determine the new code for the state.

This process occurs whenever new data is found for a state. Thus as soon as two properties are known, all others may be found. Because of this feature only a limited number of the sixty four codes will be used. For example, codes where only four properties are known should never arise because as soon as two of the properties are known, the rest are found.

**CLEAR**

This routine is called to clear the display screen and home the cursor. It does this by writing an ESCAPE 'v' to the terminal. Because FORTRAN does not normally allow special control characters to be written from inside a program, a special ICSA supplied subroutine is used - TOUTCC. This routine is included in the Fortran library and is automatically linked to the CLEAR routine through the use of the XLINK command.
Because each terminal has different characteristics, this routine may not work on any terminal other than a Visual 200.

**DEVICE**

This routine is a character function subprogram with a single parameter – the device code. Given this code the function will return the description of the corresponding device. For example, if a code equal to four (4) is passed to DEVICE, the returning description will be 'Pump'.

**FILSIZ**

This routine will return at any time the number of active records in either the initial or final network and property files. The current size of each of these files is written to the first and only record of the LENGTH file (18).

When called, the subroutine reads this record and returns the appropriate field value to the calling program.

**PAUSE**

This routine will halt processing until the RETURN key is pressed. When called, subroutine PAUSE will display on the screen the message

press RETURN key to continue
and wait for user response, at which time it will return to the
calling program.

PUTSIZ

This routine will write the current length of a specified file
into the appropriate field in the LENGTH file(18).
APPENDIX G

SAMPLE OF OUTPUT GENERATED BY SUBROUTINE FIXNOD

AFTER INITIAL DATA ENTRY:

<table>
<thead>
<tr>
<th>node</th>
<th>connections</th>
<th>inlet/outlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>n1, 10</td>
<td>1, 0</td>
</tr>
<tr>
<td>2</td>
<td>10, n2</td>
<td>1, 0</td>
</tr>
<tr>
<td>3</td>
<td>n3, 10</td>
<td>1, 0</td>
</tr>
<tr>
<td>4</td>
<td>10, n4</td>
<td>1, 0</td>
</tr>
<tr>
<td>...</td>
<td>........</td>
<td>........</td>
</tr>
<tr>
<td>10</td>
<td>1, 2, 3, 4</td>
<td>1, 0, 1, 0</td>
</tr>
</tbody>
</table>

AFTER SUBROUTINE BLDNOD:

<table>
<thead>
<tr>
<th>node</th>
<th>connections</th>
<th>inlet/outlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>n1, 10</td>
<td>1, 0</td>
</tr>
<tr>
<td>2</td>
<td>10, n2</td>
<td>1, 0</td>
</tr>
<tr>
<td>3</td>
<td>n3, 10</td>
<td>1, 0</td>
</tr>
<tr>
<td>4</td>
<td>10, n4</td>
<td>1, 0</td>
</tr>
<tr>
<td>...</td>
<td>........</td>
<td>........</td>
</tr>
<tr>
<td>1001</td>
<td>1, 1002</td>
<td>1, 0</td>
</tr>
<tr>
<td>1002</td>
<td>1001, 2, 1003</td>
<td>1, 0, 0</td>
</tr>
<tr>
<td>1003</td>
<td>1002, 1004</td>
<td>1, 0</td>
</tr>
<tr>
<td>1004</td>
<td>1003, 3, 1005</td>
<td>1, 1, 0</td>
</tr>
<tr>
<td>1005</td>
<td>1004, 4</td>
<td>1, 0</td>
</tr>
</tbody>
</table>
AFTER SUBROUTINE FIXNOD:

<table>
<thead>
<tr>
<th>node</th>
<th>connections</th>
<th>inlet/outlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>n1, 1001</td>
<td>1, 0</td>
</tr>
<tr>
<td>2</td>
<td>1002, n2</td>
<td>1, 0</td>
</tr>
<tr>
<td>3</td>
<td>n3, 1004</td>
<td>1, 0</td>
</tr>
<tr>
<td>4</td>
<td>1005, n4</td>
<td>1, 0</td>
</tr>
<tr>
<td>1001</td>
<td>1, 1002</td>
<td>1, 0</td>
</tr>
<tr>
<td>1002</td>
<td>1001, 2, 1003</td>
<td>1, 0</td>
</tr>
<tr>
<td>1003</td>
<td>1002, 1004</td>
<td>1, 0</td>
</tr>
<tr>
<td>1004</td>
<td>1003, 3, 1005</td>
<td>1, 1, 0</td>
</tr>
<tr>
<td>1005</td>
<td>1004, 4</td>
<td>1, 0</td>
</tr>
</tbody>
</table>
APPENDIX H

OUTPUT EXAMPLES OF SUBROUTINE ORDER:

Network

Value(s) of array ISTACK after processing

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>1</th>
</tr>
</thead>
</table>
Whenever a new device is to be added to the Thermal Analyzer, additions must be made to several of the subroutines. The obvious addition is to add a subroutine that will model the device in question. This subroutine should contain the same common blocks and arguments that are present in all of the device solution routines.

New sections must also be added to the input subroutines to facilitate any special entry routines. The device must also be added to the Network Input Menu (Fig. 4) as well as to the subroutine DEVICE.

Each device is assigned a unique integer code. Changes must be made throughout the entire set of programs so that this new code may be handled.

Additional lines must also be added to most of the subroutines in the pre-processor. These would indicate if the device had several separate flow paths through it, or if all incoming and outgoing flows are mixed together.

If any special data manipulation is needed during the pre-processing phase, the appropriate subroutines must be added.
GLOSSARY OF TERMS

File: A file is a collection of records that resides on disk as opposed to memory. Thus a file is not lost when program execution ends.

Node: This is an integer from 1 to 999 inclusive given to each device in the network. No device may have the same nodal value.

Post-Processor: This describes the collection of routines that first reassemble the manipulated data (created by the pre-processor) into a form similar to the initial data entry files. After assembling the final files, the post-processor prints these files.

Pre-Processor: This is the term used to describe a set of subroutines that manipulate the data entered for a session into a form that may be readily used by the main solution routines.

Processor: The processor is a collection of subroutines used to step through the prescribed network and solve for the unknowns. It uses the work files created by the pre-processor to solve the problem.

Record: A record is a collection of data that are related in some fashion. A collection of records are organized into a file. For example, the network file (INIT.NODE.DATA) contains several records, each of which contain the data for a certain device in the network.

Session: This is the term used to differentiate problems from one another. This allows one to work on several different networks concurrently. In other words, it is not necessary to work only one
problem from start to finish before beginning another problem.
REFERENCES


